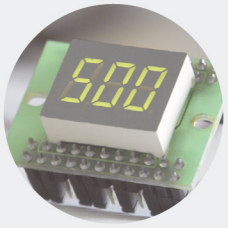
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES






Automatização do processo de comissionamento de um RGV

DIOGO NOGUEIRA BESSA
novembro de 2021

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

AUTOMATIZAÇÃO DO PROCESSO DE COMISSIONAMENTO DE UM RGV

Diogo Nogueira Bessa
Outubro de 2021

Departamento de Engenharia Eletrotécnica
Mestrado em Engenharia Eletrotécnica e de Computadores
Área de Especialização em Automação e Sistemas

Relatório elaborado para satisfação parcial dos requisitos da Unidade Curricular de Tese/Dissertação do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: Diogo Nogueira Bessa, Nº 1160781, 1160781@isep.ipp.pt

Orientação científica: Professor Lino Manuel Baptista Figueiredo, lbf@isep.ipp.pt

Empresa: Körber Supply Chain PT

Supervisão: João Oliveira, joao.oliveira@koerber.com

Departamento de Engenharia Eletrotécnica

Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização em Automação e Sistemas

2021

Agradecimentos

Agradeço ao engenheiro João Oliveira pela ajuda e pelos conhecimentos fornecidos ao longo da realização do estágio, e aos restantes colaboradores da empresa Körber Supply Chain PT que me proporcionaram as melhores condições de trabalho possíveis.

Agradeço ao professor Lino Figueiredo a disponibilidade e os esclarecimentos prestados para o desenvolvimento deste projeto.

Agradeço aos meus amigos, por todos os momentos de diversão e a partilha de ideias durante todo o percurso académico.

Agradeço aos meus pais e ao meu irmão pela educação, suporte, apoio e alegria que me deram ao longo da vida.

Resumo

O mundo está em constante modernização, principalmente na área industrial. Nos últimos anos tem-se verificado uma evolução no desenvolvimento de equipamentos e métodos completamente automáticos. Nos armazéns, a automação das atividades é essencial para obter o melhor desempenho possível. Uma destas atividades é o comissionamento do circuito de RGVs. Atualmente o comissionamento é um processo manual que, além de ser bastante demorado, consome mão-de-obra extra.

Este projeto surge com o objetivo de automatizar as diferentes fases do comissionamento de um circuito de RGVs, através da análise e desenvolvimento de vários métodos. Estas fases consistem no registo dos limites das curvas e das agulhagens, a deteção dos RGVs e o alinhamento dos transportadores.

As implementações das soluções analisadas baseiam-se na aplicação de sensores, como leitores RFID e sensores de distância LiDAR. Ao longo do projeto, verifica-se o desenvolvimento das várias rotinas e a configuração dos sensores.

As soluções são validadas através da realização de vários testes num circuito, adquirindo diferentes valores de posições e confirmando-se o correto desempenho.

Palavras-Chave

RGV, Comissionamento, *Master Controller*, RFID, LiDAR, Automação, PLC

Abstract

The world is in constant modernization, especially in the industrial area. In recent years there has been an evolution in the development of fully automatic equipment and methods. In warehouses, automation of activities is essential to obtain the best possible performance. One of these activities is the commissioning of the RGV circuit. Commissioning is currently a manual process that, in addition to being quite time consuming, consumes extra labour.

This project arises with the objective of automating the different phases of commissioning a circuit of RGVs, through the analysis and development of various methods. These phases consist of recording the limits of curves and switch devices, detecting the RGVs and aligning the conveyors.

The implementations of the analysed solutions are based on the application of sensors, such as RFID readers and LiDAR distance sensors. Throughout the project, the development of the various routines and the configuration of the sensors is verified.

The solutions are validated by carrying out several tests on a circuit, acquiring different position values and confirming correct performance.

Keywords

RGV, Commissioning, *Master Controller*, RFID, LiDAR, Automation, PLC

Índice

Índice de Figuras

Índice de Tabelas

Acrónimos

AGV – *Autonomous Guided Vehicle*

AP – *Access Point*

AS/RS – Sistema de Reaquisição e Armazenamento Automático

DB – *Data Block*

ER – *Enhanced Read*

ERP – *Enterprise Resorce Planning*

EW – *Enhanced Write*

FB – *Function Block*

FC – *Function*

HF – *High Frequency*

IL – Lista de instruções

IP – *Internet Protocol*

LF – *Low Frequency*

LiDAR – *Light Detection And Ranging*

MC – *Master Controller*

PLC – Controlador Lógico Programável

RFID – *Radiofrequency Identification*

RGV – *Rail Guided Vehicle*

SD – *Switch Device*

SFC – *Sequential Function Chart*

SR – *Single Read*

SW – *Single Write*

UDT – *User Defined Type*

UHF – *Ultra High Frequency*

WCS – *Warehouse Control System*

WMS – *Warehouse Management System*

1. INTRODUÇÃO

1.1. CONTEXTUALIZAÇÃO

Com a constante modernização visível mundialmente, tem-se observado cada vez mais organização a nível industrial. O número de pedidos dos clientes aumenta e as empresas sentem uma necessidade de evolução no nível de controlo dos seus produtos. Esta evolução consiste na automatização dos armazéns e centros de distribuição, tendo em vista a diminuição dos tempos operacionais e dos custos totais, e o aumento no desempenho dos trabalhadores. Esta automatização baseia-se na utilização de vários dispositivos, complementados com sensores e atuadores, como os *Rail Guided Vehicles* (RGVs), com o objetivo de auxiliar os operadores presentes nestas instalações.

Assim surgiu este projeto, proposto pela empresa Körber Supply Chain PT, com o propósito de desenvolver soluções de automação nas etapas do comissionamento de circuitos de RGVs, recorrendo ao estudo e implementação de sensores nestes sistemas.

1.2. OBJETIVOS

O objetivo principal deste projeto é a automatização das diferentes fases do processo de comissionamento de um circuito de RGVs.

Tendo em vista este objetivo, o projeto divide-se nas seguintes tarefas:

- Avaliar os elementos do circuito que podem ser alvo de deteção;
- Pesquisar os sensores existentes no mercado que podem ser usados para detetar as posições;
- Avaliar e apresentar os sensores mais indicados para as tarefas, nomeando as vantagens e desvantagens dos mesmos;
- Configurar e testar os sensores;
- Criar uma rotina no programa do Controlador Lógico Programável (PLC) do RGV que permita avaliar e calcular a posição central de cada ponto de carga/descarga;
- Criar uma rotina no programa do PLC do RGV que permita detetar e registar o início e o fim de cada curva e agulhagem;
- Criar uma rotina no *Master* PLC que permita a interação entre dois RGVs numa curva, adquirindo a informação da distância;
- Testar as rotinas criadas e avaliar os resultados.

1.3. BREVE DESCRIÇÃO DA EMPRESA

A Körber AG, fundada em 1946 por Kurt A. Körber, é um grupo multinacional alemão que produz soluções de gestão estratégica. Atualmente conta com cerca de 10 000 trabalhadores, tendo mais de 100 instalações em todo o mundo. Está presente em cinco áreas: digital, farmacêutica, cadeias de abastecimento, produção de papel e tabaco.

Em setembro de 2015, este grupo adquiriu a Efacec Handling Solutions, alterando o seu nome para Consoveyo e integrando-a na sua área de Sistemas Logísticos. Mais tarde, a 1

de setembro de 2020, passou a designar-se Körber Supply Chain PT, nome que mantém até hoje.

A empresa é responsável por fornecer uma gama de tecnologias e soluções relacionadas com as cadeias de abastecimento, de forma a ultrapassar qualquer complexidade nos fluxos de trabalho, desde o desenvolvimento de *software* até ao transporte de materiais e a sua integração.

1.4. CALENDARIZAÇÃO

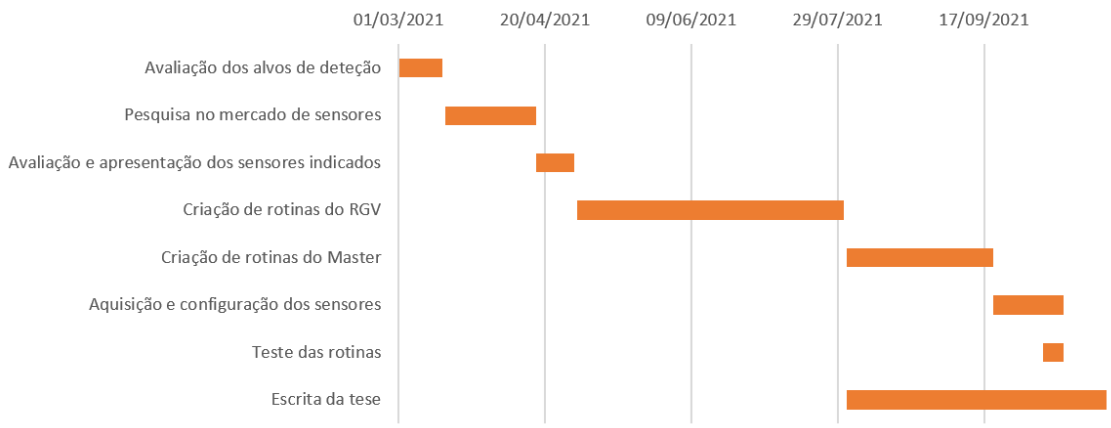


Figura 1 – Calendarização

1.5. ORGANIZAÇÃO DO RELATÓRIO

Com o intuito de melhorar a organização da leitura, este documento encontra-se dividido em sete capítulos.

No Capítulo 1 é abordada a introdução ao projeto, onde se faz uma breve contextualização do problema e onde são realçados os principais objetivos inerentes a este projeto. É feita

uma breve apresentação da empresa, exibindo-se também a calendarização do projeto, e finalmente, a organização do relatório.

Em seguida, no Capítulo 2, é feito um reconhecimento em concreto do problema, analisando o conceito do comissionamento de circuitos de RGVs e possíveis soluções a implementar.

Quanto ao Capítulo 3 é realizado um estudo sobre o estado da arte relacionado ao projeto abordando conceitos como a Automação e armazéns automáticos.

No Capítulo 4 é apresentada a arquitetura dos sistemas, com foco nos RGVs. São também analisadas as diferentes hipóteses de resolução dos problemas existentes, explorando diferentes sensores.

Posteriormente, no Capítulo 5, é apresentada a instalação e a configuração destes sensores, e a implementação das principais rotinas desenvolvidas.

No Capítulo 6, são expostos os resultados obtidos em cada teste e a sua respetiva análise.

Finalmente no Capítulo 7, são retiradas as conclusões sobre o desenvolvimento do projeto e possíveis melhorias a realizar futuramente.

2. ANÁLISE DE REQUISITOS

2.1. PROBLEMA

Atualmente, a implementação dos sistemas automáticos projetados pela Körber Supply Chain PT recorre à necessidade de um comissionamento. O comissionamento é o processo que assegura o bom funcionamento dos sistemas e dos seus respetivos componentes, através do controlo da projeção, instalação, teste e operação, tendo em conta todas as necessidades e requisitos presentes nas instalações do cliente. Este processo é aplicado tanto na implementação em novas instalações como na expansão ou modernização de armazéns já existentes. O comissionamento consiste na aplicação de um conjunto de procedimentos e técnicas, com o objetivo de testar e adaptar o sistema implementado na instalação atual, através da configuração dos equipamentos e instrumentos a utilizar.

O comissionamento de um circuito de RGVs é um processo bastante demorado, uma vez que a maioria das técnicas utilizadas para configurar o sistema são realizadas manualmente. Assim sendo, tendo em conta este desafio, pretende-se com este projeto estudar e automatizar estas técnicas através do uso de sensores.

2.2. SOLUÇÃO PROPOSTA E OPERACIONALIZAÇÃO

O sistema de RGVs desenvolvido pela Körber Supply Chain PT, no seu processo de comissionamento, é necessário reconhecer o circuito para que se possa configurar e colocar em funcionamento. Pretende-se desta forma automatizar este processo de reconhecimento escolhendo os sensores apropriados para detetar os vários elementos. Os elementos que se pretende detetar são os limites das curvas e agulhagens, as posições de alinhamento entre o RGV e os transportadores, e as distâncias possíveis entre RGVs presentes numa curva. Para isso, é necessário programar várias rotinas, com o objetivo de:

- Registar as posições inicias e finais de todas as curvas e agulhagens, para o RGV abrandar;
- Detetar as posições de alinhamento entre os transportadores, para não ocorrer a queda das cargas;
- Substituir o sensor atual por outro sensor que permita uma área de deteção maior, de maneira a que o RGV consiga detetar outro veículo que possa estar à sua frente.

3. ESTADO DA ARTE

Neste capítulo é feito um estudo superficial da área onde este projeto se enquadra, desde o conceito e a sua importância, até aos seus diferentes tipos e aplicações. É também abordado uma noção de armazéns automáticos e dos seus constituintes básicos.

3.1. AUTOMAÇÃO

Definir automação é de certa forma complexo uma vez que este conceito está presente em diferentes áreas. *Oxfords English Dictionary* define automação como:

> *"Automatic control of the manufacture of a product through a number of successive stages; the application of automatic control to any branch of industry or science; by extension, the use of electronic or mechanical devices to replace human labor"*.

Como é descrito nesta definição, automação refere-se geralmente à mecanização e integração de sensorização num sistema [1].

3.1.1. CONCEITO

Automação, ou controlo automático, pode ser definida como a tecnologia pela qual um processo é executado sem assistência humana através de vários sistemas de controlo, isto é, as pessoas podem estar presentes como observadores ou até mesmo como participantes, sendo que o processo opera por si mesmo [2][3].

Com o rápido crescimento das tecnologias de informação e de mecânica, a relevância destes sistemas aumentou. Hoje em dia, o termo automação desenvolveu-se para além do fabrico, transformando o trabalho físico em trabalho cognitivo. Tarefas como a interpretação e o registo de dados, a tomada de decisões e a visualização de informações são consideradas automação uma vez que atualmente são realizadas por computadores. Ao projetar um sistema automático é essencial ter em conta a otimização e a alocação das tarefas entre humanos e máquinas [1].

As atividades de um processo automático são determinadas por sistemas de controlo que executam um programa de instruções. Em processos automáticos mais simples, uma única instrução pode controlar uma certa variável, como regular a temperatura de um forno. Em processos mais complexos, é necessária uma sequência de atividades durante o ciclo de trabalho, onde a ordem e os detalhes destas atividades são definidas pelo programa de instruções.

Em alguns processos automáticos, o programa deve conter instruções capazes de tomar decisões ou reagir a eventos inesperados. Alguns destes eventos podem ser variações nas matérias-primas que precisam do ajuste de certos parâmetros, interações e comunicações com humanos através de informações do estado do sistema, requisitos da monitorização de segurança e mau funcionamento dos equipamentos [2].

3.1.2. TIPOS

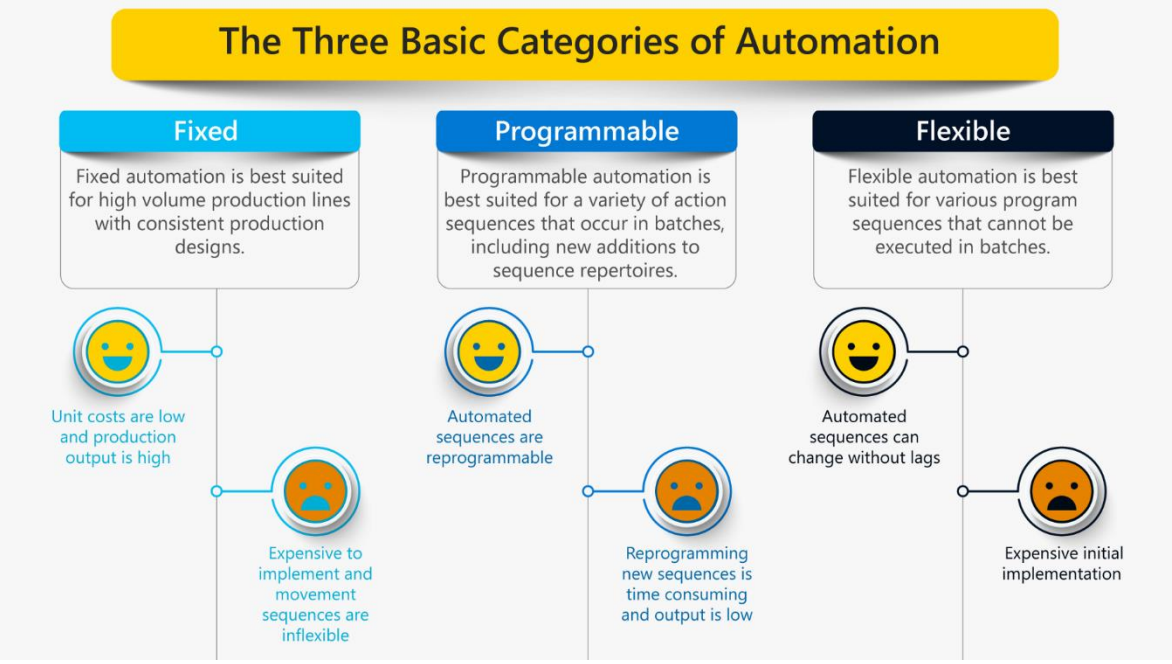


Figura 2 – Tipos de automação [4]

Como podemos ver na Figura 2, os sistemas automáticos podem ser classificados através de três tipos de automação [2][4][5]:

- **Fixa**:

  Também conhecida como *Hard Automation*, é caracterizada pela rigidez da configuração do equipamento, ou seja, o programa de instruções é determinado pela configuração dos equipamentos e não pode ser alterado facilmente. Cada etapa da sequência normalmente envolve uma ação simples. Embora o ciclo de trabalho consista em operações simples, integrar e coordenar as ações pode resultar na necessidade de um sistema de controlo mais sofisticado.

  A automação fixa possui características como um alto investimento inicial em equipamentos especializados, um aumento de produtividade e pouca flexibilidade para acomodar uma grande variedade de produtos.

  Este tipo de automação é mais adequado para produções em massa uma vez que o alto custo de investimento pode ser distribuído por várias unidades, tornando o

custo de cada unidade relativamente baixo. As primeiras linhas de montagem de automóveis nos Estados Unidos são exemplos de automação fixa.

- **Programável:**

  A automação programável permite que os equipamentos sejam projetados no sistema com a capacidade de alterar o programa de instruções para possibilitar a produção de diferentes produtos. Na automação fixa, a automação é projetada com apenas um conjunto de sequências de operação, já na automação programável é permitida a reprogramação para diferentes tarefas.

  Esta automação é caracterizada pelo alto investimento em equipamentos reprogramáveis, menos produtividade do que a automação fixa e uma maior flexibilidade de lidar com uma maior variedade de produtos.

  Estes fatores levam a que esta automação seja adequada para produções com uma grande variedade de produtos.

- **Flexível**:

  A automação flexível é uma extensão da automação programável em que praticamente não há perdas de tempo para alterações na configuração ou reprogramação. Esta automação permite mudanças automáticas e rápidas em sequências programadas sem tempos de inatividade. Um sistema flexível é, portanto, capaz de produzir uma mistura de diferentes produtos, um após o outro, em vez de lotes separados.

  As características inerentes a este tipo de automação vão desde o grande investimento em equipamentos de engenharia e taxas de produção médias, até uma produção contínua de diferentes estilos de produtos.

3.1.3. DILEMAS ÉTICOS

Ao contrário do que maioria da população acredita, a automação não aumenta o desemprego. Tal como a história nos ensinou no século XIX, a mecanização da agricultura apenas veio substituir as tarefas repetitivas e perigosas, criando trabalhos mais qualificados

como a interação com clientes, o desenvolvimento de novos produtos e a própria gestão dos robôs [6].

Para H. James Wilson e Paul R. Daugherty, da *Harvard Business Review*, o impacto da automação é algo positivo: *"While AI will radically alter how work gets done and who does it, the technology's larger impact will be in complementing and augmenting human capabilities, not replacing them"* [7]. Isto significa que para ocorrer a modernização de uma empresa, além de implementar novos processos e melhorar os fluxos de trabalho, é também necessário o investimento em novos trabalhadores.

Em 2019, a empresa *Zebra Technologies* realizou o estudo *"2024 Warehouse Vision Study"* com o objetivo de analisar as tendências e os desafios que influenciam o armazenamento. O estudo contou com 1403 pessoas ligadas à área que descreveram algumas estratégias atuais e futuras para modernizar e aumentar a eficiência e a produtividade dos armazéns, centros de distribuição e centros de abastecimento, de forma a acompanhar o contexto da economia.

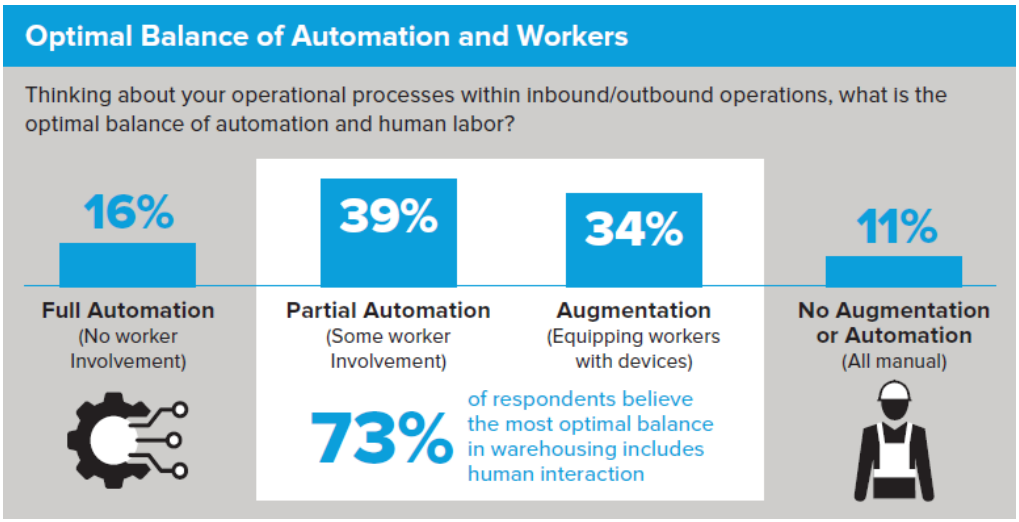


Figura 3 – Estudo sobre o equilíbrio entre automação e trabalhadores [8]

De acordo com a Figura 3, estas estratégias variam entre manter os processos manuais, automatizar de forma parcial ou apenas otimizar a mão-de-obra, até realizar uma automação total. Desta forma foi possível verificar que a maioria dos respondentes concorda que a melhor maneira de automatizar os armazéns é a introdução de mais trabalhadores em conjunto com a tecnologia. Isso já se verifica em alguns casos onde os

trabalhadores usam dispositivos portáteis para ler códigos de barras, etiquetas *Radiofrequency Identification* (RFID) e robôs para aumentar a eficiência e minimizar o risco de acidentes de trabalho. Pode-se dizer então que tanto o aumento da automação quanto o crescimento da mão-de-obra serão soluções essenciais para o desenvolvimento industrial nos próximos anos [8].

3.1.4. APLICAÇÕES

A automação está presente na transformação de uma ampla variedade de funções vitais em diferentes setores, tais como a saúde, agricultura, fabrico, armazenamento, entre outros, sendo que muitos deles já se encontram completamente automáticos.

A saúde, que é uma das grandes preocupações da espécie humana devido ao seu elevado risco e necessidade de precisão, é um setor em constante evolução uma vez que são feitos vários estudos sobre a possibilidade de realização de procedimentos mais fáceis e mais rápidos. A cirurgia é uma das áreas mais desenvolvidas nos últimos anos, onde a automação é empregada na execução de procedimentos com maior precisão, diminuindo o número de incisões e os relativos riscos [8].

Na agricultura, a automação está presente na realização de tarefas difíceis e repetitivas, de forma a poder ajudar os trabalhadores e aumentar a eficiência nos processos. Estas tarefas vão desde a colheita e o transporte de plantas até ao controlo do clima em estufas inteligentes [10].

O constante aumento do consumo leva a um aumento na necessidade de produção de vários tipos de produtos. A introdução da automação no setor do fabrico começou bem cedo com o aparecimento dos primeiros robôs industriais em 1961. Inicialmente estes robôs apenas realizavam tarefas repetitivas, mas com o avançar dos anos estes foram utilizados para conceber produtos com melhor qualidade, através da execução de processos com maior precisão [10].

Quanto ao setor de armazenamento, cada vez é mais comum ouvir falar do conceito de armazéns automáticos. Tal como nos outros setores, a presença de automação nos

armazéns traz muitas vantagens, como a melhoria da qualidade e segurança do trabalho ao realizar automaticamente tarefas mais difíceis. Este setor será analisado mais aprofundadamente no próximo subcapítulo.

3.2. ARMAZÉNS AUTOMÁTICOS

Os armazéns são componentes importantes na maioria das cadeias de abastecimento. Em termos de custo, eles representam aproximadamente 20% dos custos logísticos totais, enquanto em termos de serviço eles são essenciais para alcançar os níveis de exigência do cliente, especialmente porque os centros de distribuição são geralmente o destino das cadeias de abastecimento, onde as encomendas são preparadas e expedidas para o cliente [11].

A automação é bastante comum em grandes armazéns, sobretudo no que diz respeito ao transporte e/ou classificação de produtos e a sistemas de reaquisição e armazenamento automático (AS/RS). No entanto, apesar da sua importância nas cadeias de abastecimento, a automação de armazéns tem recebido pouca atenção na área de investigação [11].

Armazéns automáticos integram múltiplas funções, como o armazenamento, o transporte, a entrega e a gestão de produtos, tendo também várias melhorias, como a capacidade de armazenamento em massa, menor utilização de área, aumento da velocidade de processamento, menor taxa de desgaste, fácil gestão, entre outras. A maioria destes armazéns encontram-se em cadeias de abastecimento e sistemas comerciais tendo sido os principais componentes dos sistemas logísticos modernos [12].

Os AS/RS são sistemas de gestão de *stock* utilizados em instalações de fabrico, centros de distribuição e armazéns. Estes sistemas consistem geralmente em máquinas que se movem em várias direções num ou mais corredores, armazenando e recuperando os produtos que depois serão divididos e exportados para diferentes destinos [13].

3.2.1. VANTAGENS

Os armazéns automáticos possuem várias vantagens, algumas delas são [13]:

- Mais flexíveis para acomodar mudanças nas condições de negócios uma vez que fornecem aos utilizadores um melhor controlo sobre a localização do *stock*;
- Compostos por subsistemas modulares que podem ser facilmente substituídos minimizando assim os tempos de inatividade e prolongando a vida útil do sistema;
- Reduzem os custos de mão-de-obra, aumentando a segurança no trabalho ao remover os trabalhadores de condições precárias, como ambientes frios;
- Reduzem os custos de armazenamento e na área utilizada já que possuem uma maior densidade de armazenamento.

Resumidamente, os AS/RS permitem que os trabalhadores melhorem a produtividade, maximizem a capacidade de armazenamento e reduzam os custos operacionais [14].

As soluções de AS/RS atendem a uma necessidade crescente nas cadeias de abastecimento ao fornecer tecnologia de automação flexível em instalações novas ou já existentes. Estes sistemas ajudam os operadores a otimizar o armazenamento e a recuperação do *stock*, ao mesmo tempo que melhoram a resposta a um aumento de pedidos, de forma mais rápida e mais precisa [14].

Os armazéns e os centros de distribuição normalmente têm dois sistemas principais que auxiliam na preparação das operações a realizar. Estes sistemas têm como funcionalidades a gestão e o controlo das operações. Na Figura 4 podemos observar a conexão entre esses sistemas.

Figura 4 – Ligação entre *Hardware* e *Sofware* num armazém [23]

3.2.2. WMS

Um Sistema de Gestão de Armazém, também conhecido como *Warehouse Management System* (WMS), é a base de uma operação no armazém. É uma solução de software que gere vários processos, projetada para controlar o fluxo de *stock*, desde a sua receção até ao armazenamento, assim como o seu fornecimento e reabastecimento. É o nível mais alto utilizado num armazém, comunicando com os sistemas *Enterprise Resorce Planning* (ERP) das empresas [15]. A implementação deste sistema fornece um aumento na eficácia, uma redução nos custos de mão-de-obra e uma maior capacidade de atendimento ao cliente, reduzindo os tempos de ciclo.

3.2.3. WCS

Um Sistema de Controlo de Armazém, ou *Warehouse Control System* (WCS), é uma solução de software que integra e controla em tempo real vários equipamentos de automação usados num armazém, de maneira que a eficiência dos subsistemas de manuseamento de materiais aumente [15]. Este sistema é a ligação entre o WMS e os equipamentos de automação, fornecendo ao utilizador interfaces para controlar, monitorizar e diagnosticar possíveis problemas [16].

3.2.4. EQUIPAMENTOS DE AUTOMAÇÃO

O equipamento de automação é projetado para otimizar a eficiência e a produtividade dos armazéns e centros de distribuição. Cada equipamento pode ser integrado no WMS ou WCS, que controlam a cadeia de abastecimento, o que significa que é possível obter os dados de cada componente e fazer uma análise individual, de forma a melhorar o controlo das operações. Todas as funções principais de um armazém, nomeadamente o transporte, a recolha, o rastreio e a seleção de produtos, podem ser realizadas por estes equipamentos. Dependendo da funcionalidade, existem vários tipos de equipamento, entre os quais:

Transelevadores

Máquinas automáticas que realizam o armazenamento e a recuperação de produtos, de forma rápida, segura e sem erros, enquanto trabalham em sistemas de estantes, tal como se pode ver na Figura 5. Podem atingir 50 m de altura e velocidades até 6 m/s, operando com cargas entre os 1 kg e os 10 000 kg.

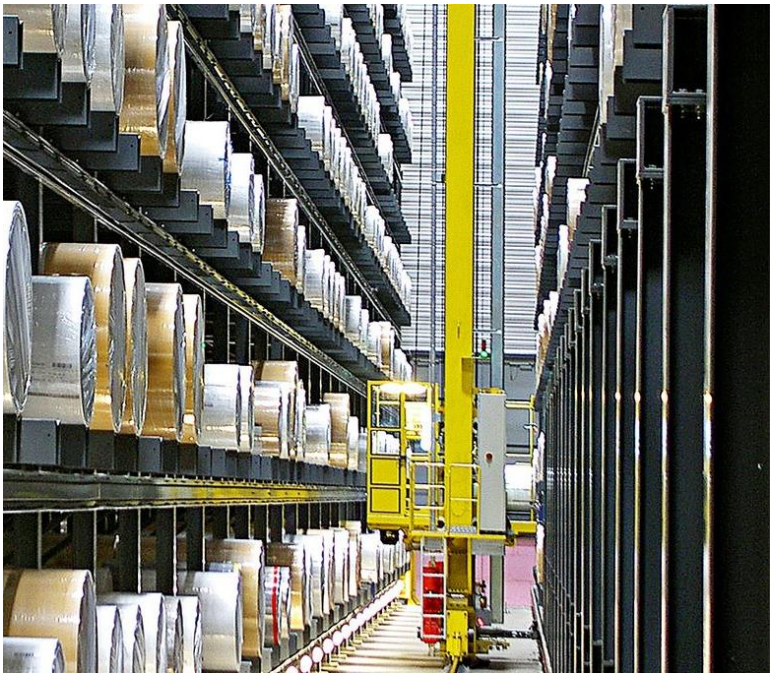


Figura 5 – Transelevador [24]

Transportadores

Os transportadores, presentes na Figura 6, garantem uma transferência de produtos de forma mais precisa, eficiente e rápida, podendo transportar cargas leves, unitárias ou até mesmo pesadas. É uma forma de transportar produtos mais segura do que utilizar empilhadores. Conforme o objetivo e as condições, este equipamento pode variar noutros sistemas, como mesas rotativas que permitem alterar a orientação do movimento da carga, elevadores que possibilitam a colocação de cargas em várias camadas, transportadores gravíticos que acumulam as cargas que irão ser expedidas, entre outros.

Figura 6 – Transportadores [25]

AGVs

Os veículos automaticamente guiados, Figura 7, igualmente conhecidos como *Autonomous Guided Vehicles* (AGV), são veículos projetados para permitir uma movimentação segura de mercadorias de forma totalmente automática, não necessitando de operadores humanos. Estes equipamentos permitem otimizar os sistemas internos de transporte e armazenagem de um armazém, aumentando a eficiência e a produtividade das operações uma vez que não estão propensos a erros dispendiosos. A navegação destes veículos é adquirida através de soluções de reflexão *laser*, sensores, fios indutivos, fitas/pontos magnéticos ou uma combinação de todos os elementos referidos [17].

Figura 7 – AGV [17]

RGV

Um veículo guiado por carril, visível na Figura 8, também conhecido como RGV, é um sistema económico de alta velocidade que conecta vários locais num armazém através de uma infraestrutura leve. Este sistema é uma alternativa mais eficiente relativamente às longas cadeias de transportadores quando se trata de transportar produtos dentro de um armazém uma vez que atinge velocidades superiores [18].

Figura 8 – RGV em andamento [18]

No próximo capítulo é feita uma análise mais aprofundada dos sistemas RGV, desde os seus constituintes até ao funcionamento geral.

4. SISTEMA

Neste capítulo é analisado um sistema RGV, desde um simples veículo e os seus componentes principais até ao seu funcionamento e a constituição do circuito. É apresentado também os diferentes modos de operação assim como a comunicação entre os subsistemas incluídos.

4.1. RGV

Tal como vimos no capítulo anterior, os RGVs são veículos ferroviários rápidos e flexíveis que permitem transportar grandes quantidades de cargas, leves ou pesadas, por longas distâncias com custos baixos e grande eficiência. Estes veículos são essenciais no armazenamento e atendimento de pedidos em vários setores.

Um sistema de RGVs é avaliado pela cadência, ou seja, pelo número de cargas movidas, dentro e fora do sistema, num determinado período de tempo. Este valor depende da

capacidade dos RGVs, das rotinas de controlo, do tipo de operação e da fiabilidade do sistema.

Os RGVs movimentam-se num circuito fechado e são controlados através de um controlador externo que lhes atribui tarefas e novas posições. Estes veículos possuem dois eixos de movimentação: o eixo X, que controla a translação do veículo, e o eixo Z, que controla o movimento do transportador.

4.1.1. VANTAGENS

Os sistemas RGV possuem várias vantagens face a operadores humanos e até mesmo relativamente a outros equipamentos de automação. Dentro destas vantagens podemos observar as seguintes:

- Alto nível de automação
- Fluxo de trabalho eficiente
- Evita trabalho manual, o que diminui a probabilidade de ocorrência de erros
- Compacto, o que leva a uma redução na área ocupada
- Baixos custos de manutenção
- Grandes velocidades de transporte
- Capacidade de transportar cargas pesadas

4.1.2. COMPONENTES PRINCIPAIS

O RGV é constituído por quatro rodas, duas são apoiadas no carril de alimentação e as outras duas são colocadas no piso. Este veículo, tal como se pode ver na Figura 9, é o resultado do conjunto de vários componentes, como sensores e atuadores. Estes componentes permitem que o veículo consiga saber o que está à sua volta e de que forma deve responder.

Figura 9 – RGV

Motores

O veículo está equipado com dois motores, um para a translação do veículo e o outro para mover o transportador, visíveis respetivamente nas Figuras 10 e 11. Estes motores não podem estar a funcionar em simultâneo, logo é utilizado um inversor de frequência que controla os dois motores alternadamente. Todos os motores possuem um termóstato, para detetar temperaturas excessivas, e travões eletromagnéticos de emergência.

Figura 10 – Motor de translação do RGV

Figura 11 – Motor do transportador do RGV

Transportadores

Normalmente, o RGV possui um transportador, podendo ter dois em alguns casos. Estes transportadores podem ter mecanismos diferentes como correntes, rolos e garfos, com o objetivo principal de transportar e transferir as cargas em segurança.

Figura 12 – Transportador do RGV

Sistema de posicionamento

O sistema de posicionamento consiste na utilização de um *encoder*, neste caso o leitor de código de barras presente na Figura 13, que lê continuamente os valores da posição através de uma fita colocada no carril ao longo do circuito. A fita, tal como a Figura 14 apresenta,

é semelhante a uma régua, uma vez que é constituída por uma sequência de códigos de barras que representam o valor absoluto da posição em milímetros. O *encoder* é instalado perto da roda traseira esquerda, de forma a ler esta posição.

Figura 13 – Leitor de código de barras

Figura 14 – Fita com sequência de códigos de barras

Sensor indutivo de referência

O sistema de posicionamento, como obtém valores absolutos da posição, não necessita de pontos de referência. No entanto, no caso de ser necessário substituir o *encoder* por outro

novo, encontra-se instalado no veículo um sensor indutivo de referência de modo a calibrar o “ponto zero” do dispositivo de posicionamento. A Figura 15 representa este sensor.

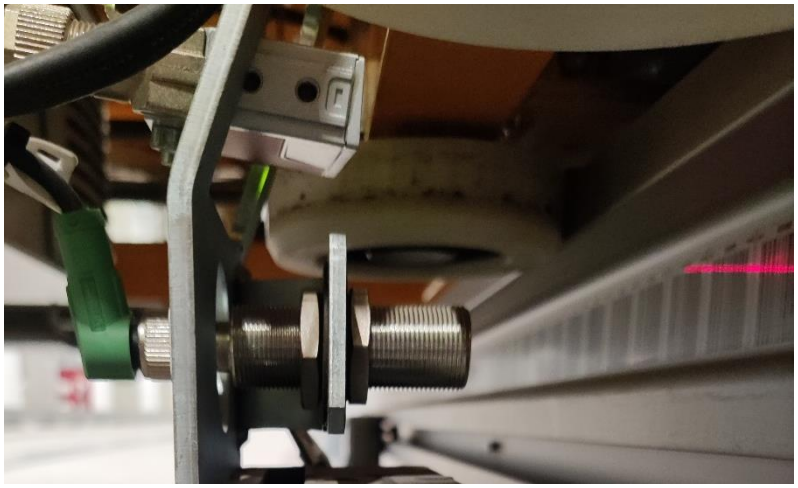


Figura 15 – Sensor indutivo de referência

Sensor magnético de redução de velocidade

Por questões de segurança, um sensor magnético é instalado na roda traseira esquerda. Juntamente com o leitor de código de barras, este sensor permite limitar a velocidade do RGV através de ímanes dispostos em certos pontos no circuito.

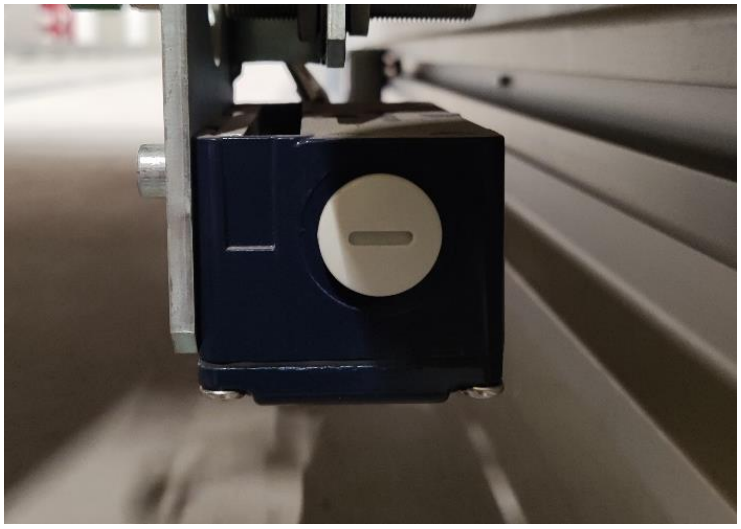


Figura 16 – Sensor magnético de redução de velocidade

Fotocélulas refletoras

No veículo, como se pode verificar pela Figura 17, são instaladas várias fotocélulas: três no transportador do RGV, para alinhar e detetar a presença de cargas, e duas nas extremidades do mesmo transportador, utilizadas por questões de segurança na transferência de cargas entre transportadores.

Figura 17 – Fotocélulas refletoras do transportador do RGV

Além das fotocélulas mencionadas, existem também dois emissores, Figura 18, na dianteira do veículo, um à esquerda e outro à direita, e dois recetores, Figura 19, nos transportadores do circuito. Estas fotocélulas servem para comunicar o estado da transferência de cargas entre o RGV e os transportadores, sendo que a comunicação só é realizada se os dois estiverem corretamente alinhados.

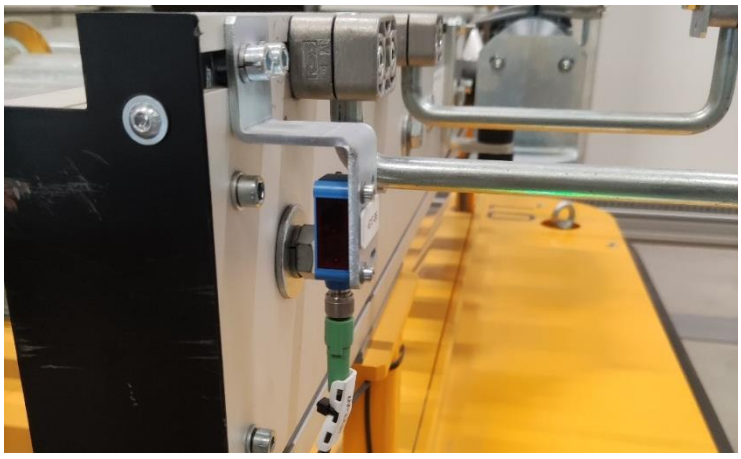


Figura 18 – Fotocélulas de comunicação com os transportadores (emissor)

Figura 19 – Fotocélulas de comunicação com os transportadores (recetor)

Sensor de deteção de RGVs

Atualmente é utilizada uma fotocélula refletora na deteção de RGVs. Na dianteira de cada RGV é instalado um emissor, neste caso o sensor de distância que se pode ver na Figura 20, e na traseira é colocado um espelho. Para evitar colisões, o veículo reduz drasticamente a velocidade quando deteta uma distância de três metros relativamente ao RGV da frente.

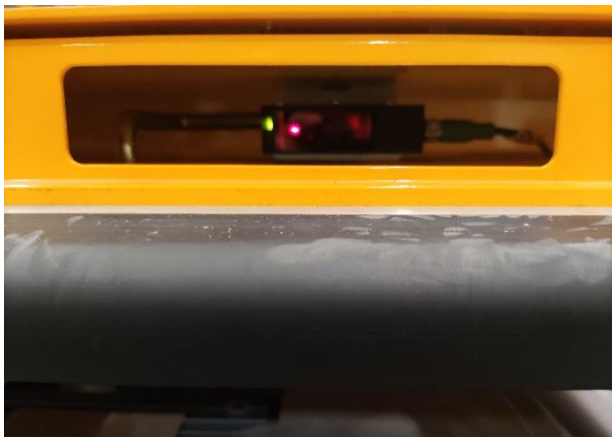


Figura 20 – Sensor de distância

Sensor de pressão de emergência (*bumper*)

Com o objetivo de aumentar a segurança no circuito, os veículos têm um sensor de pressão à sua frente e nos cantos, visíveis na Figura 21. Este sensor é projetado para proteger o RGV de possíveis choques com outros veículos e realizar uma paragem de emergência no caso de colidir com pessoas.

Figura 21 – Sensor de pressão de emergência

Sistema de comunicação *Wireless Ethernet*

A comunicação das operações pode ser realizada através de uma ligação física pelo carril de alimentação ou de forma *wireless*, onde o RGV está equipado com um cliente *Wireless Ethernet* com duas antenas, visivo na Figura 22. Este dispositivo conecta-se automaticamente a um *Access Point* (AP), como o da Figura 23, com melhor intensidade de sinal.

Figura 22 – *Access Point*

Figura 23 – Cliente *Wireless Ethernet*

PLC

Um PLC é um computador utilizado na automação de processos eletromecânicos, como o controlo de máquinas e de luzes. Ao contrário dos computadores mais utilizados atualmente, o PLC é projetado para trabalhar com várias entradas e saídas, e funcionar em áreas com grandes variações de temperatura, vários ruídos elétricos, vibrações e até mesmo impactos. Neste projeto, o PLC utilizado no veículo é da marca Siemens, da série S7-1200.

No RGV, o PLC é o controlador responsável por toda a lógica dentro do veículo. É a unidade de processamento central que tem como funções:

- Processamento de todos os sinais das fotocélulas, sensores indutivos, sensores magnéticos, etc.…;
- Verificação do sentido do movimento atual dos motores (translação e transportador);
- Interface com o operador através de um controlador manual e luzes de sinalização;
- Deteção de erros e falhas.

Apesar de cada RGV ter um controlador próprio responsável pela execução destas funções, ele não reconhece as posições dos outros veículos, apenas a sua posição atual e a posição de destino. Por este motivo, torna-se necessário a existência de um controlador externo, também conhecido como *Master Controller* (MC).

4.2. MASTER CONTROLLER

O *Master Controller* é um *software* inteligente, responsável por gerir todo o sistema de RGVs. Ele controla todos os processos que envolvam o movimento de cargas, as comunicações de dados e o comportamento do sistema. O MC é o cérebro do sistema uma vez que comunica com todos os componentes do circuito, entre os quais os RGVs, as agulhagens, os equipamentos de segurança, e o WCS. Ele recebe informações sobre cada

equipamento podendo depois enviar comandos para carregar e descarregar produtos, mover as agulhagens e inserir ou remover dados no RGV.

4.2.1. TOPOLOGIA MASTER-SLAVE

Um sistema de RGVs baseia-se no modelo de comunicação *Master-Slave*, como o da Figura 24, onde o *Master Controller* atribui tarefas e gere as informações no circuito e o *Slave* (cada RGV) executa os comandos e envia as informações para o *Master*.

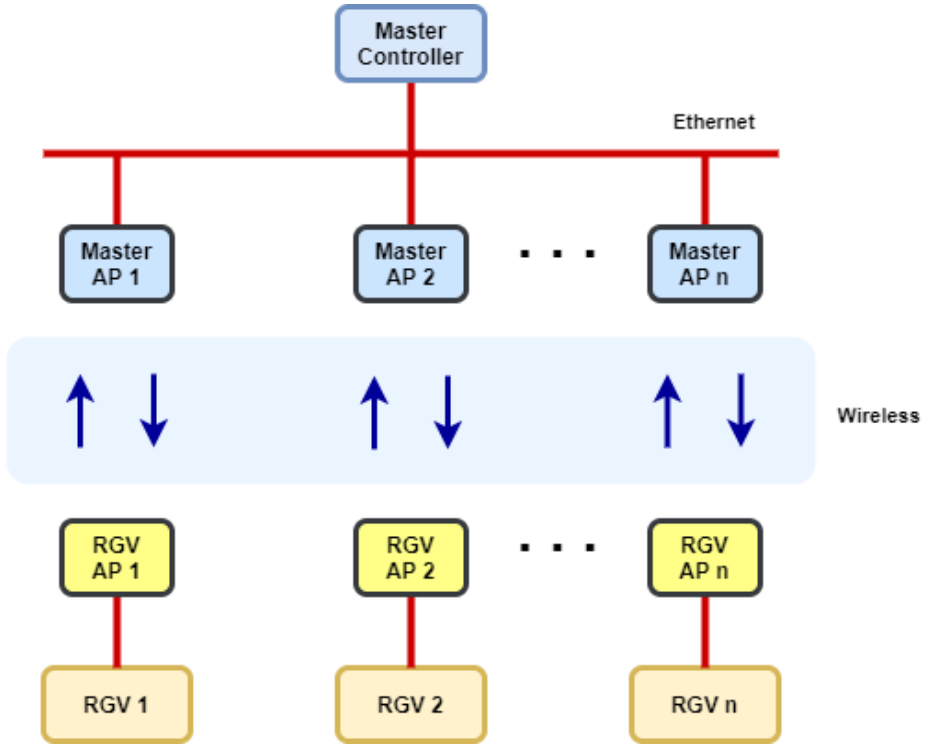


Figura 24 – Modelo de comunicação *Master-Slave*

Neste modelo de comunicação é usado o protocolo ISO-*on*-TCP. Este protocolo, também conhecido como RFC1006, foi desenvolvido pela Siemens numa tentativa de combinar dois métodos de transferência de dados *Ethernet*.

O *Master Controller* tem as seguintes funções principais:

- Inserção e remoção de RGVs no sistema;
- Comunicação com RGVs, transportadores e agulhagens;

- Controlo de tráfego no circuito através da ativação das agulhagens;
- Gestão das posições dos veículos;
- Envio de novas tarefas e comandos para os RGVs;
- Monitorização de erros;
- Comunicação com o WMS;
- Verificação dos sistemas de segurança no circuito;
- Interface com o operador através de uma consola, botões e luzes.

O PLC utilizado neste projeto como *Master Controller* é o modelo S7-1500 da Siemens.

Como já foi referido, quem faz a gestão do sistema é o *Master*, isto é, controla os RGVs, agulhagens, transportadores, e todos os equipamentos presentes no circuito. No caso dos RGVs, é apresentada a arquitetura da comunicação na Figura 25.

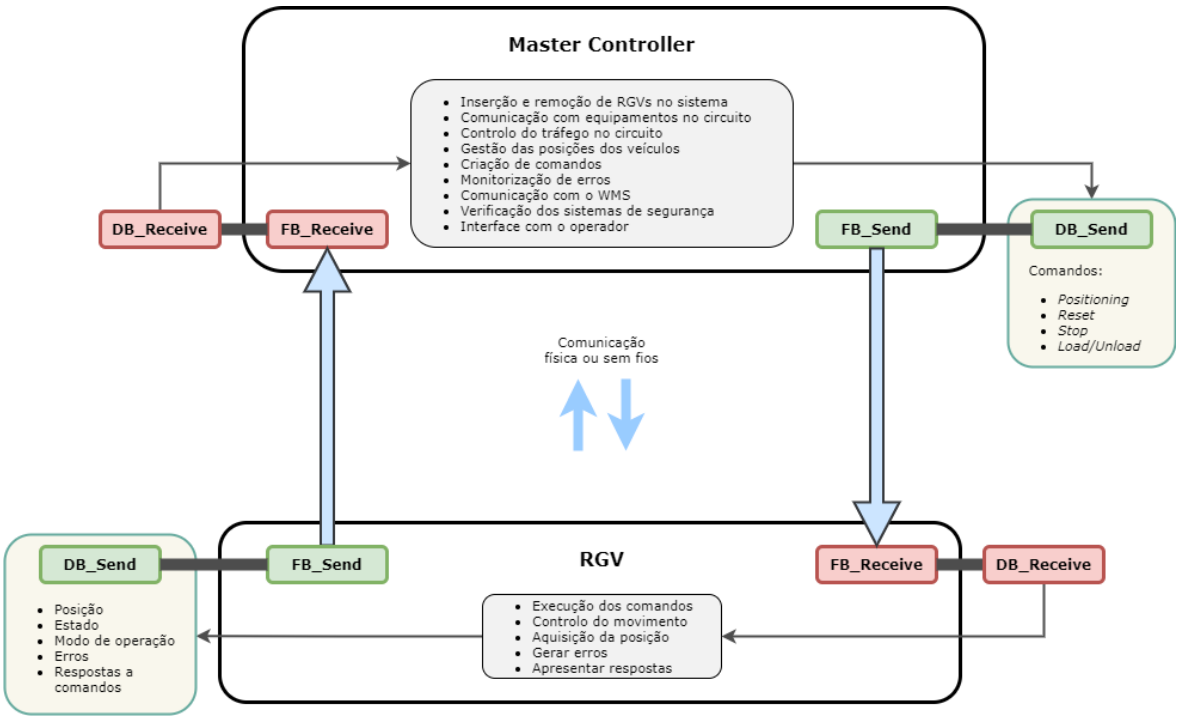


Figura 25 – Arquitetura da comunicação MC-RGV

Basicamente a comunicação entre os dois PLCs baseia-se na troca de bases de dados. O *Master* executa as suas funções e envia os comandos, armazenados na base de dados **DB_Send**, para o RGV através da função **FB_Send**. Estes comandos têm como objetivo posicionar, reiniciar ou parar o veículo, e carregar ou descarregar produtos. O PLC do

veículo recebe e guarda os comandos na base de dados **DB_Receive**, através da função **FB_Receive**. Estes fazem com que o veículo execute as suas tarefas e armazene a informação resultante na **DB_Send**, que consiste na posição, estado e modo de operação do veículo, eventuais erros e o estado da execução dos comandos, sendo depois enviada para a **DB_Receive** do *Master* através da **FB_Send**.

4.3. CIRCUITO

O circuito de RGVs define a área de operação e envolve todos os equipamentos presentes no sistema, como veículos, cabines elétricas, agulhagens, transportadores, entre outros. Para melhor entendimento, será analisado neste subcapítulo um exemplo de *layout* do circuito de um armazém, presente na Figura 26.

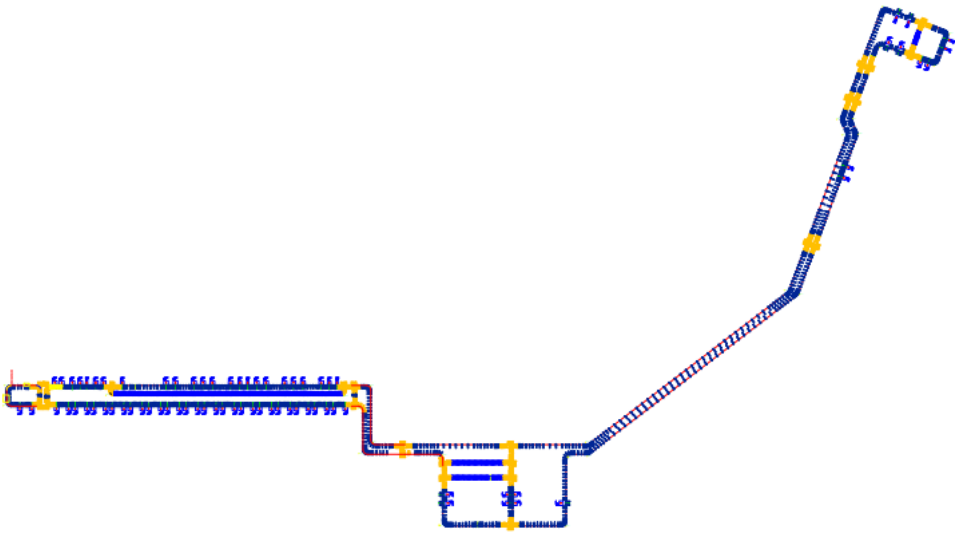


Figura 26 – *Layout* do circuito

4.3.1. ÁREAS

De forma a obter um melhor desempenho dos RGVs é necessário ter um circuito organizado, com áreas bem delimitadas, para isso um sistema encontra-se dividido em três circuitos:

1. Circuito principal
2. Circuito de manutenção
3. Circuito de espera

Estes circuitos agrupam-se em áreas principais e secundárias.

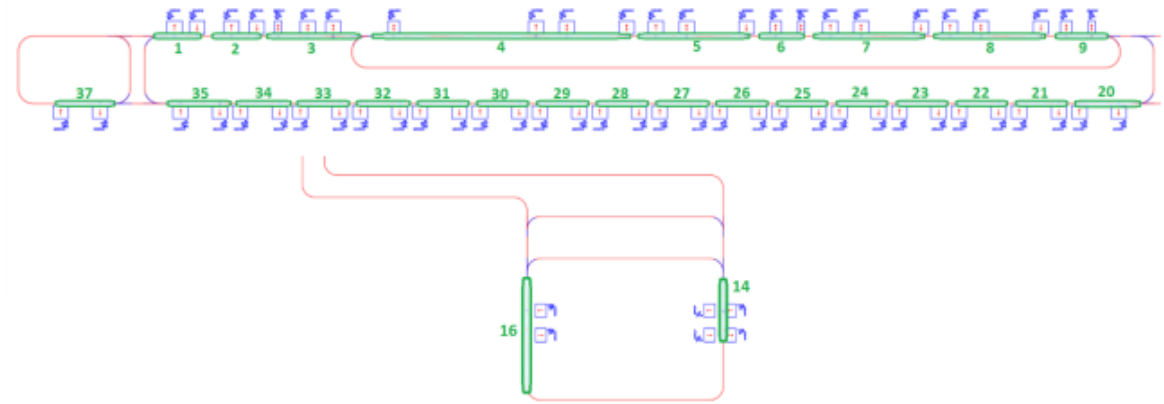


Figura 27 – Exemplo de áreas principais no circuito

As áreas principais, também conhecidas como áreas prioritárias, são espaços menores que incluem o circuito principal que por sua vez cobre um certo número de transportadores. Na Figura 27 é possível observar a verde algumas das áreas principais presentes num circuito.

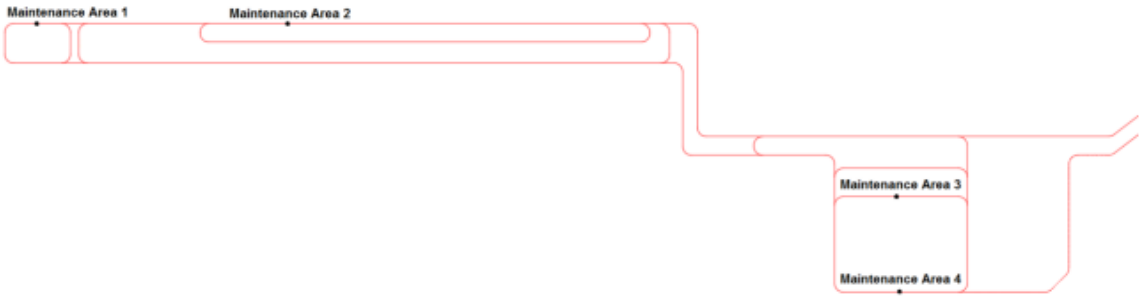


Figura 28 – Exemplo de áreas de manutenção no circuito

As áreas secundárias normalmente são áreas maiores do que as principais, uma vez que contêm várias zonas para estacionar os RGVs: os circuitos de manutenção e os de espera.

Os circuitos de manutenção são criados para facilitar e simplificar o processo de manutenção dos RGVs, reduzindo o impacto no desempenho do sistema. Estes circuitos estão presentes na Figura 28.

Os circuitos de espera são utilizados para acumular os RGVs que não se encontram a ser utilizados. O Master calcula e define um número mínimo de veículos necessários para atender às necessidades dos circuitos principais. Se o número de RGVs presentes nos circuitos principais for menor do que o necessário, o circuito de espera mais próximo fornece veículos. Estes tipos de circuitos são colocados em pontos estrategicamente definidos para melhorar e gerir o fluxo de veículos presentes no sistema.

4.3.2. PARTES PRINCIPAIS

Ao longo do circuito estão presentes vários componentes essenciais ao circuito, tanto a nível funcional, como as agulhagens e os transportadores, como a nível de segurança.

Agulhagens

A agulhagem, visível na Figura 29, também conhecida como *Switch Device* (SD), é um equipamento com capacidade de alterar trajetos lineares em trajetos curvos. Estes equipamentos são controlados através da receção de comandos do MC e permitem calcular o percurso mais curto que cada RGV deverá realizar para chegar à sua posição de destino.

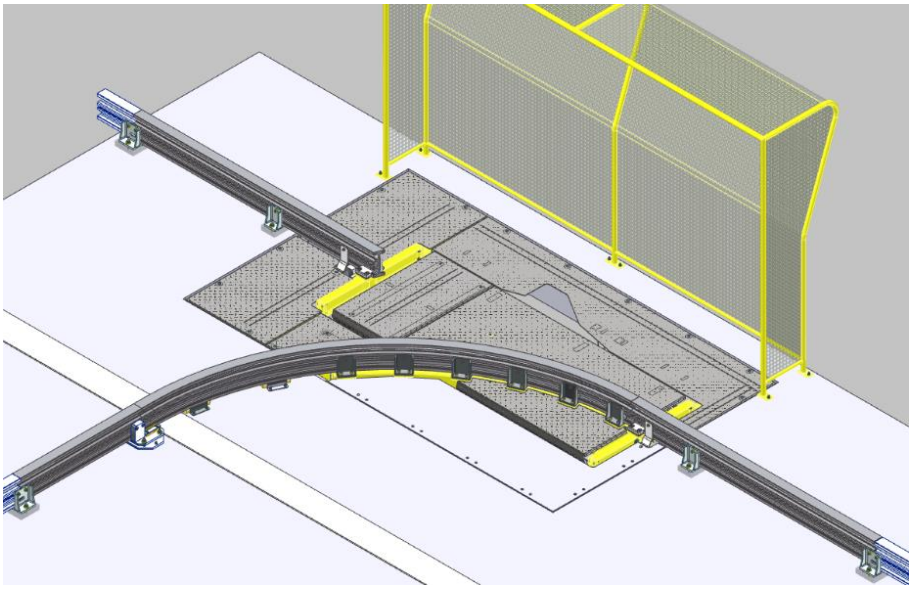


Figura 29 – Modelo de uma agulhagem

A Figura 30 corresponde a uma agulhagem presente no circuito de RGVs da Super Bock Group, sendo possível verificar que possui duas secções de carril, uma curva e outra reta. Sempre que for necessário alterar entre carris a agulhagem terá de rodar.

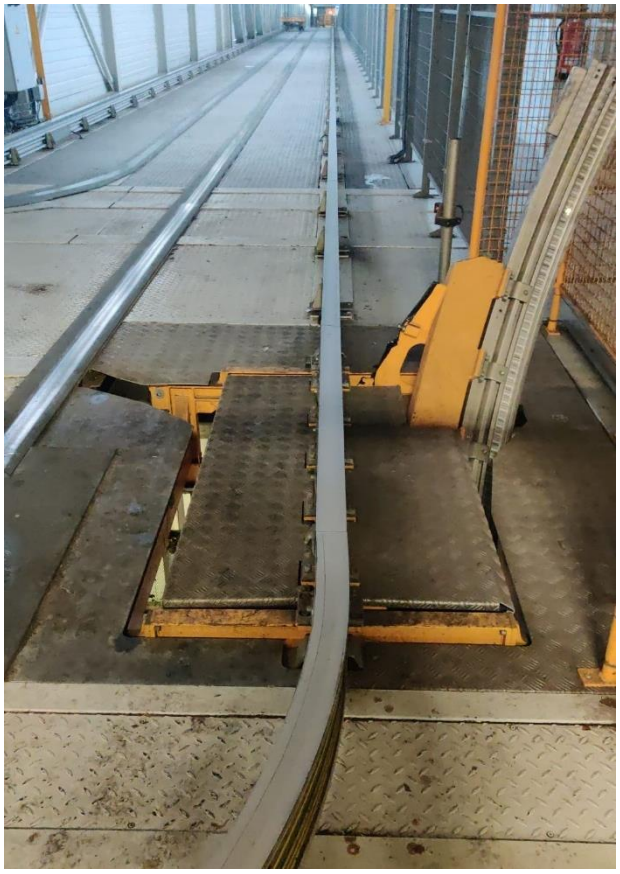


Figura 30 – Agulhagem presente no armazém da Super Bock Group

Transportadores

Os transportadores são projetados para abranger, de forma mais eficiente, a movimentação dos produtos, normalmente em paletes e caixas, entre diferentes áreas dentro de um armazém. A principal função destes equipamentos no circuito de RGVs é a entrega e a receção de cargas dos RGVs, a velocidades reduzidas, comunicando com os outros equipamentos do sistema de forma síncrona para melhorar a eficiência do processo e a não ocorrência de erros. Na Figura 31 é possível verificar um transportador que faz a comunicação com os veículos no circuito.

Figura 31 – Transportador

Relativamente aos RGVs, os transportadores podem ser de três tipos: entrada, saída e mistos. Como entrada, os transportadores fornecem cargas ao circuito, enquanto como saída, estes recebem as cargas dos veículos e enviam-nas para outras áreas. Os transportadores mistos operam um pouco como os outros dois tipos mencionados, funcionando como entrada e saída de cargas no circuito, sendo essencial uma boa comunicação entre o MC e os outros equipamentos.

Zonas de acesso

Por vezes é necessária a intervenção de um operador no circuito, tanto por falhas no sistema como pela manutenção dos veículos. Por estes motivos são criadas várias zonas de acesso distribuídas ao longo do circuito, tendo em conta o fluxo de trabalho e a segurança. Para garantir a segurança do operador, é introduzido no circuito alguns mecanismos, como barreiras de luz, nos limites destas zonas. Se algo for detetado ao passar a barreira, o PLC envia um comando para cortar a energia no carril de alimentação.

4.3.3. LAYOUT

O *layout* de um circuito de RGVs representa toda a instalação do sistema. Um *layout*, como o da Figura 26, é desenhado no AUTOCAD e é utilizado na configuração do MC, traduzindo o circuito físico em múltiplas variáveis, como o formato do circuito, o tipo de transportadores, as posições das curvas, entre outras.

Segmentos

Para controlar os RGVs do circuito, e porque estes veículos podem seguir diferentes caminhos para alcançar o mesmo destino, o circuito é dividido em vários segmentos. Cada segmento é uma fração do circuito que é utilizada para construir a rota do RGV. A rota de cada veículo pode ser definida por um ou mais segmentos.

Cada segmento é caracterizado por ter uma posição inicial e uma posição final. Normalmente, estas posições são definidas num ponto onde dois segmentos convergem ou onde um segmento diverge em dois, ou seja, onde existem agulhagens. Entre a posição inicial e a posição final não existem pontos de convergência ou divergência, pois se existissem seria necessário dividir segmentos.

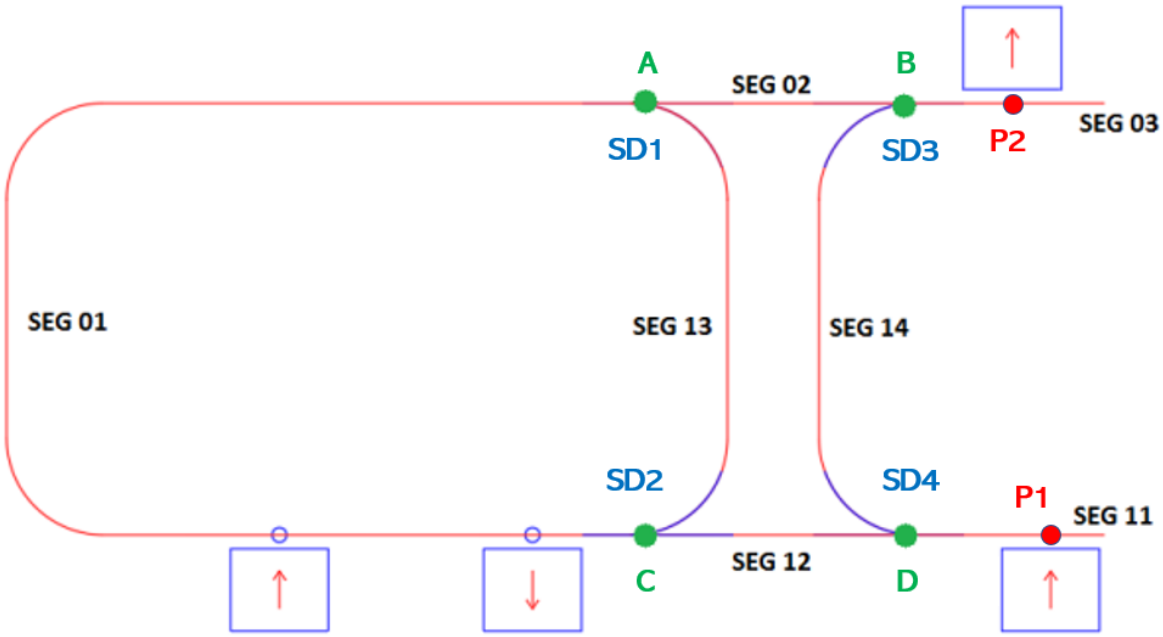


Figura 32 – Segmentos

Na Figura 32 está representado uma parte do circuito presente na Figura 26, onde se podem verificar os segmentos, os pontos de convergência e divergência de segmentos, os transportadores e as agulhagens.

Tabela 1 – Segmentos e posições

Segmento	Posição inicial	Posição final
01	C	A
02	A	B
12	D	C
13	A	C
14	D	B

A Tabela 1 indica os segmentos e as suas respetivas posições. Apesar de existirem segmentos que partilham os mesmos limites, os valores obtidos pelo *encoder* são diferentes.

Rotas

A definição dos segmentos de um circuito é essencial para que o sistema possua um bom desempenho, afetando diretamente a construção das rotas. As rotas são os percursos que cada veículo tem de realizar para chegar à sua posição de destino. Uma rota é constituída por um ou mais segmentos. A definição das rotas dos RGVs é umas das principais funções do MC, sendo responsável por estar constantemente a determinar a rota mais curta possível.

Na Figura 32, ainda é possível observar dois pontos: a posição inicial P1, colocada no segmento 11, e a posição final P2, definida no segmento 03. Após uma análise do estado das agulhagens e do tipo de segmento, o MC iria calcular duas rotas, definidas na Tabela 2.

Tabela 2 – Exemplo de possíveis rotas

Rota 1	Rota 2
11	11
12	14
01	03
02	-
03	-

Dentro das rotas definidas, o MC escolhe a menor possível. Neste caso, como é possível de notar-se, se as agulhagens SD3 e SD4 estivessem definidas como curvas a segunda rota seria a mais curta, sendo esta atribuída ao RGV.

Os segmentos estimados nos circuitos de espera não são considerados no cálculo das rotas.

4.4. MODOS DE OPERAÇÃO

O RGV possui dois modos de operação:

- Automático:
  - Ocorre a comunicação entre o MC, os RGVs, transportadores, agulhagens e o WCS;
  - O *Master Controller* é responsável por controlar remotamente o veículo e as agulhagens;
  - O Master envia tarefas/comandos aos RGVs e às agulhagens, e retorna a confirmação do estado das tarefas/comandos, erros e informação estatística.

- Manual:
  - O controlo é feito pelo operador através de uma consola ou painel de controlo;
  - Utilizado para configuração do sistema e manutenção do veículo;
  - Disponível para inserir e remover RGVs do sistema, cancelar comandos em execução, alterar dados, cargas, etc…

Para alterar entre modos, o utilizador deve utilizar um interruptor presente no veículo ou no painel de controlo perto das zonas de manutenção.

4.5. PROBLEMAS E POSSÍVEIS SOLUÇÕES

Como já foi referido no segundo capítulo, para comissionar um circuito de RGVs é necessário realizar anteriormente uma identificação do circuito. Antes de o sistema funcionar, o *Master*, para ser configurado, tem de ter conhecimento de vários parâmetros do circuito, como os comprimentos, as posições iniciais e finais das curvas e das agulhagens, e as posições de alinhamento entres os RGVs e os transportadores.

4.5.1. REGISTO DE POSIÇÕES DAS CURVAS E AGULHAGENS

Um dos principais problemas na projeção de um circuito de RGVs é o registo das posições das curvas e das agulhagens. Estas posições são muito importantes uma vez que, para além do MC reconhecer os limites destas partes do circuito, são também posições onde deverá ocorrer uma diminuição de velocidade.

Atualmente, o reconhecimento destas posições é feito manualmente pelo operador, o que torna o processo bastante demorado, por vezes durando algumas semanas. Um dos objetivos deste projeto é automatizar este processo através da utilização de sensores, de forma a poupar no tempo e nos custos.

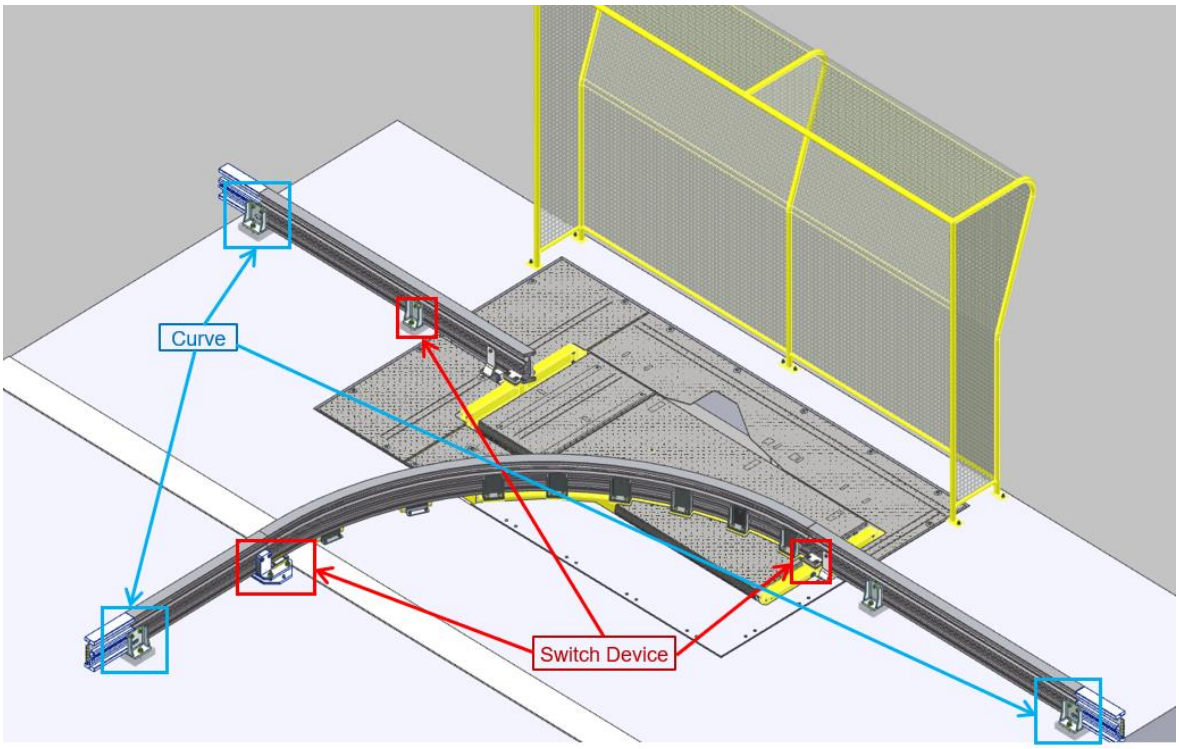


Figura 33 – Pontos a detetar

Num circuito de RGVs, o carril está afixado ao chão através de múltiplos suportes, visíveis na Figura 33. Os formatos destes suportes diferem consoante a área onde estão colocados. Nas Figuras 35 e 34 é possível observar, respetivamente, o suporte que prende as curvas e o seu formato visto por cima.

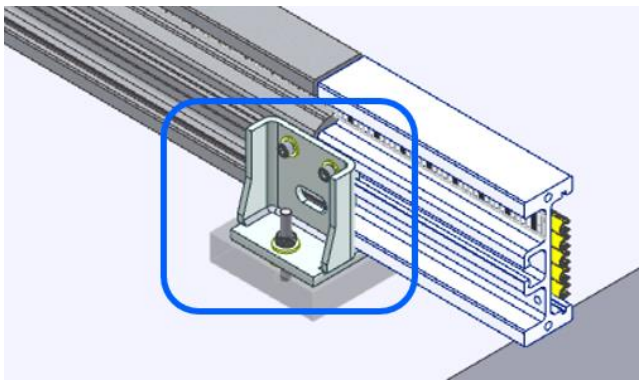


Figura 35 – Suporte das curvas

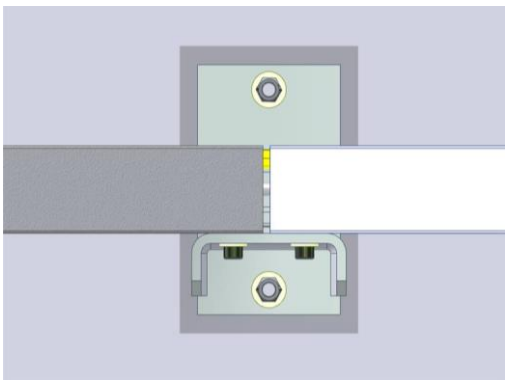


Figura 34 – Formato do suporte das curvas

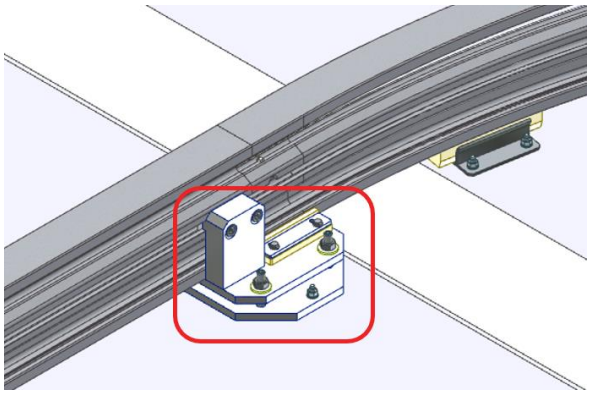


Figura 36 – Suporte das agulhagens

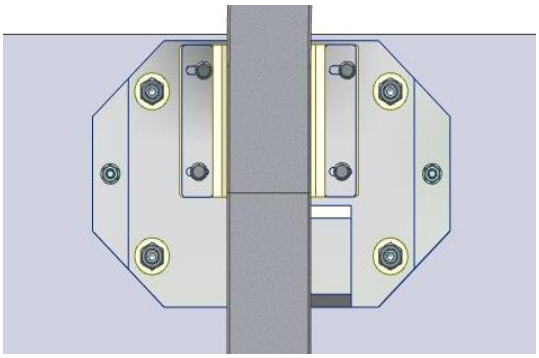


Figura 37 – Formato do suporte das curvas

As Figuras 36 e 37 representam também, respetivamente, o suporte das agulhagens e o seu formato.

Para solucionar o problema de identificar estes pontos no circuito existem várias hipóteses de solução, entre as quais: a utilização de sensores visuais, leitores de códigos 2D, navegação por linha, e etiquetas RFID.

Hipótese 1: Sensores visuais

Uma das hipóteses seria a utilização de sensores visuais. Estes sensores seriam instalados perto da roda da frente que está apoiada no carril e teriam como objetivo a deteção do formato/contorno dos suportes mencionados anteriormente. Para isso, fez-se uma pesquisa e uma análise de alguns sensores existentes atualmente no mercado. Destes sensores, destacam-se o O2D222 e o O3D302, do fabricante IFM.

Figura 38 – Sensor de contorno O2D222 [26]

Figura 39 – Sensor 3D O3D302 [27]

Tabela 3 – Características dos sensores visuais

	O2D222	O3D302
Função	. Controlo de presença, integridade, posição e qualidade de produtos . Deteção de contornos independentemente da posição rotacional . Fornece uma percentagem de correspondência entre objetos	. Medição, sem contacto, de objetos de formato retangular . Deteção do tamanho, rotação e posição de objetos . Saídas de comutação
Resolução de imagem [px]	640 x 480	176 x 132
Taxa de leitura [Hz]	≤ 20	≤ 25
Distância de funcionamento [mm]	50 - 2000	300 - 8000
Preço [€]	873.40	1295.90

A Tabela 3 apresenta algumas das características principais dos sensores visuais analisados.

Um problema deste tipo de sensores é a necessidade de outros equipamentos que dê o "disparo" para ativar o sensor, uma vez que não podem estar constantemente ligados devido ao aquecimento inerente à sua utilização. Estas são as razões pela rejeição desta hipótese.

Hipótese 2: Leitores de códigos 2D

Na hipótese da utilização de leitores de códigos 2D (códigos QR e *Data Matrix*), os leitores seriam instalados perto da roda da frente de forma a detetar os códigos 2D, que seriam distribuídos ao longo do circuito, junto dos suportes que marcam as posições pretendidas. A Figura 41 representa o sensor analisado para esta hipótese. Este sensor, da marca SICK, foi desenvolvido principalmente para a localização de AGVs, como se pode ver na Figura

40, mas neste caso a leitura dos códigos teria como objetivo diferenciar o tipo de posição detetada (início e fim de curva ou agulhagem). As características deste sensor são apresentadas na Tabela 4.

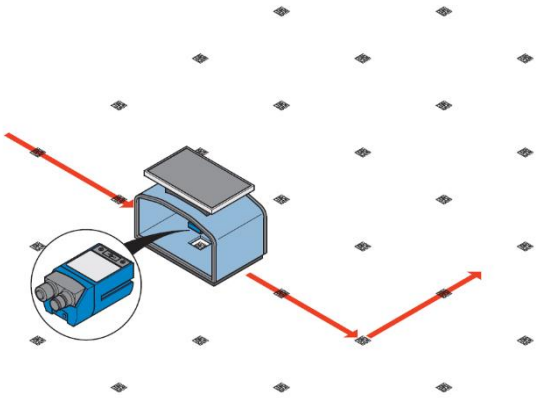


Figura 40 – Esquema da localização de AGVs [28]

Figura 41 – Leitor de códigos 2D GLS6 [28]

Tabela 4 – Características do leitor de códigos 2D

	GLS6 (V2D621G-2MSXBB5)
Função	. Leitura de etiquetas com códigos QR ou *Data Matrix*, colocadas no chão ou nos suportes . Medição de ângulos integrada . Leitura a altas velocidades (até 5 m/s) . Consegue calcular o desvio da posição a detetar
Resolução de imagem [px]	1280 x 1024
Taxa de leitura [Hz]	≤ 50
Distância de leitura [mm]	70 - 500
Preço [€]	2187.70

Embora esta seja uma opção mais precisa do que a anterior, e mais rentável, uma vez que os códigos 2D podem ser facilmente impressos em larga escala pelo técnico, o ambiente num armazém não permite a utilização ideal do mesmo. Variáveis como a luminosidade, ruídos, vibrações e grandes diferenças de temperatura não permitem o bom funcionamento deste tipo de sensores, levando a resultados inconclusivos. Tendo isto em conta, a hipótese da utilização de leitores de códigos 2D foi rejeitada.

Hipótese 3: Navegação por linha

Na hipótese de navegação por linha, com os sensores de navegação por linha, da marca SICK, seria possível a deteção das posições através da constante leitura de uma linha instalada ao longo do circuito, junto ao carril. Este tipo de sensores é maioritariamente utilizado em AGVs.

Figura 42 – Sensor ótico de navegação por linha OLS [29]

Figura 43 – Sensor magnético de navegação por linha MLS [30]

O sensor da Figura 42, o modelo OLS, permite a leitura de uma fita luminescente instalada no chão e a deteção de códigos 1D colocados nas posições pretendidas, como na Figura 44. Já o modelo MLS, da Figura 43, além da deteção da linha magnética, também permite a identificação dos pontos desejados. A comparação entre as características destes dois sensores encontra-se na Tabela 5.

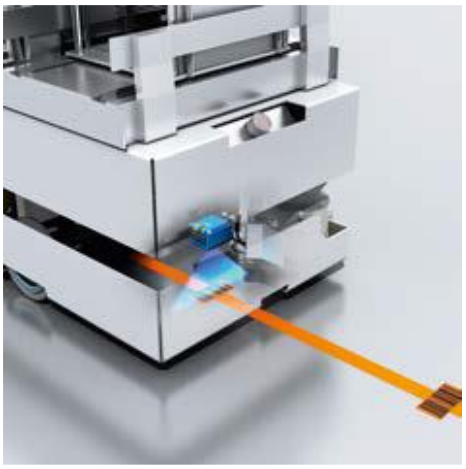


Figura 44 – Deteção de pontos através de códigos 1D [29]

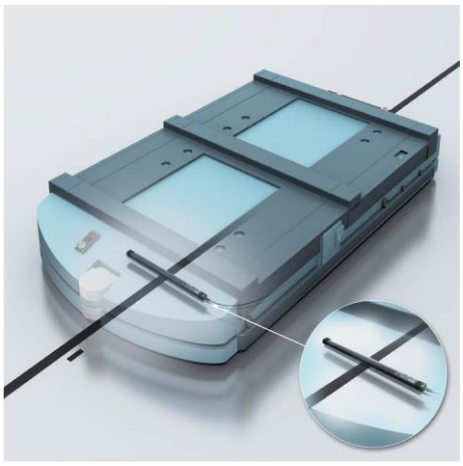


Figura 45 – Navegação através linha magnética [30]

Tabela 5 – Características dos sensores de navegação por linha

	OLS	MLS
Função	. Deteção da fita luminescente . Consegue calcular o desvio do centro da fita . Permite ler códigos 1D (códigos de barra), perpendiculares à fita, que fornecem a posição	. Navegação é feita através do alinhamento do sensor ao centro da linha magnética . Diferencia até 3 linhas, conseguindo navegar através de ramos e junções de linhas . É capaz de medir a posição e a velocidade do veículo
Taxa de leitura [Hz]	≤ 50	≤ 100
Distância de leitura [mm]	100	300
Preço [€]	1486.96	1269.12

Apesar desta hipótese ser válida para alcançar o objetivo final, as desvantagens associadas são muitas. Os custos, a área e o tempo gasto na instalação eliminam esta hipótese das opções consideradas.

Hipótese 4: Etiquetas RFID

A última hipótese considerada foi a utilização de leitores RFID. RFID é uma tecnologia de comunicação sem fios utilizada na identificação de objetos ou pessoas. Está atualmente incorporada em muitas aplicações, entre as quais [19]:

- Monitorização da localização de paletes nas cadeias de abastecimento;
- Sistemas de controlo de acessos;
- Dispositivos de rastreio de animais;
- Monitorização de veículos.

O número de aplicações tem vindo a crescer, abrangendo cada vez mais indústrias. Em todas as aplicações mencionadas, o objetivo é detetar objetos em movimento, ou seja, o leitor é instalado numa zona fixa e os objetos possuem a *tag* a ser detetada. No entanto, nesta hipótese, pretende-se adaptar o funcionamento das aplicações anteriores de forma inversa, onde o leitor estará em movimento e as *tags* estarão fixas ao chão. Assim sendo, os leitores deverão ser instalados no RGV, permitindo assim detetar as diferentes *tags* colocadas nos limites de cada curva e agulhagem.

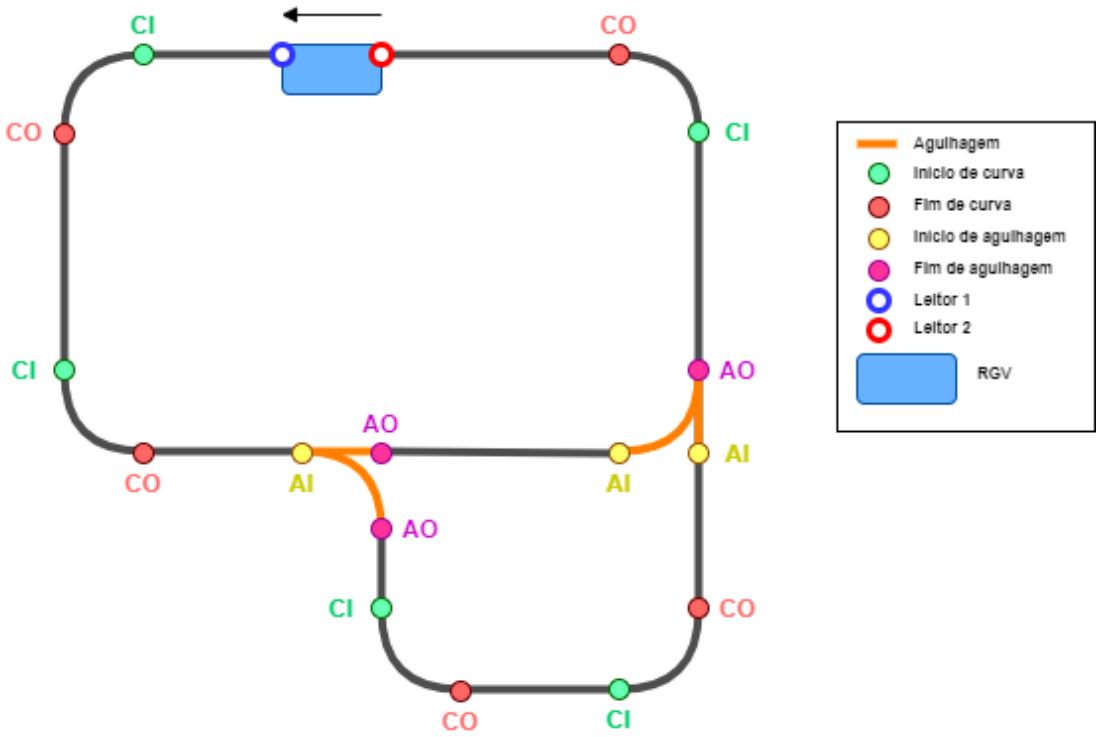


Figura 46 – Esquema da colocação das *tags* e os respetivos códigos

Na Figura 46, observa-se um exemplo do funcionamento desta hipótese. Cada *tag* tem um código associado, escrito previamente pelo programador. Nas entradas e saídas das curvas, as *tags* possuem respetivamente o código "CI" e "CO". Já nas entradas e saídas das agulhagens os códigos são "AI" e "AO".

Nesta hipótese é necessária a utilização de dois leitores RFID, um na dianteira do veículo, para detetar o momento em que o RGV entra na curva ou agulhagem, e o outro na traseira, que capta o instante de saída. O leitor da frente tem como objetivo a deteção dos códigos "CI" e "AI", ou seja, os pontos iniciais de cada curva e agulhagem, já o leitor de trás reconhece os códigos "CO" e "AO", de forma a obter os pontos finais dos mesmos.

Tendo em conta a utilização desta hipótese, foram analisados alguns leitores RFID. Para além da capacidade de leitura de informação, estes necessitam também da gravação, visto que é necessário primeiro escrever os códigos pretendidos nas *tags*. Os leitores analisados estão presentes nas seguintes imagens.

Figura 47 – Leitor RFID HF DTI421 [31]

Figura 48 – Leitor RFID UHF RFU620-10100 [32]

Figura 49 – Leitor RFID UHF IUT-F190-B40-2V1D-FR1-01 [33]

A Tabela 6 apresenta algumas das características principais destes sensores.

Tabela 6 – Características dos leitores RFID

	DTI421	RFU620-10100	IUT-F190-B40-2V1D-FR1-01
Função	. Cabeça de leitura e gravação RFID HF para instalação não nivelada . Deteta as etiquetas ID em suportes de peças de trabalho e produtos	. Habitualmente utilizado em transportadores e transportes móveis . Design robusto de acordo com o protocolo IP67	. Deteta as etiquetas ID em suportes de peças de trabalho e produtos . A leitura de várias etiquetas aumenta a produtividade
Banda de frequências	HF	UHF	UHF
Distância de leitura [mm]	180	2000	2000
Preço [€]	166.40	1052	1100

O leitor da Figura 47, o modelo DTI421 do fabricante IFM, permite ler e escrever informação nas *tags*, porém tem uma desvantagem relativamente aos outros leitores, a operação em *High Frequency* (HF). Embora os leitores de RFID HF, por serem mais baratos e mais práticos, estejam presentes na maioria das aplicações, a utilização de *Ultra High Frequency* (UHF) domina a nível industrial. Apesar de serem mais caros, os leitores UHF permitem um maior alcance de leitura e uma maior velocidade de transferências de dados, assim como as *tags* UHF são geralmente mais baratas e funcionam melhor com metais ao seu redor.

Os leitores HF trabalham globalmente a uma frequência de 13.56 MHz, já os leitores UHF possuem bandas de frequência entre 865 MHz e 928 MHz, dependendo da região onde são instalados. Na Europa, a frequência de funcionamento deverá estar entre 865 MHz e 868 MHz.

Tanto o leitor RFU620-10100 da SICK, Figura 48, como o leitor IUT-F190-B40-2V1D-FR1-01 da Pepperl+Fuchs, Figura 49, funcionariam nesta hipótese. Por questões de quantidade de documentação existente e nível de simplicidade da configuração, a opção escolhida para implementar neste desafio do registo de posições do circuito é o leitor da Pepperl+Fuchs.

4.5.2. DETEÇÃO DE RGVS

Outro grande desafio deste projeto é a análise e a implementação de um sistema de distanciamento entre RGVs, de forma a substituir a solução que se encontra a ser utilizada.

Tal como já foi mencionado, o RGV apenas conhece a sua posição atual e a sua posição de destino, sendo que o *Master Controller* é o responsável pela gestão das posições de todos os RGVs, logo os veículos nunca comunicam diretamente entre si. Por esta razão, cada RGV requer um sensor de distância na sua dianteira, de maneira a que o veículo não choque com o veículo da frente, no caso deste estar parado. Atualmente, a solução a este obstáculo é a utilização de um sensor de distância linear na dianteira de cada veículo. Apesar de funcionar, esta solução também apresenta algumas desvantagens, sendo a principal a não deteção nas curvas, como se pode verificar pelo esquema da Figura 50.

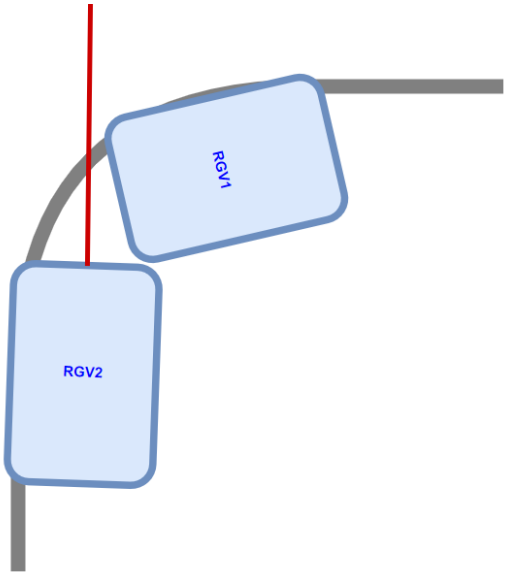


Figura 50 – Problema da utilização do sensor linear

Um dos principais processos no comissionamento de um circuito de RGVs é o registo manual das posições "limite", ou seja, as posições numa curva onde os RGVs poderiam colidir. Este processo baseia-se na divisão de uma curva em várias posições, seguida pelo envio de uma sequência de comandos do *Master Controller*, presente na Figura 51.

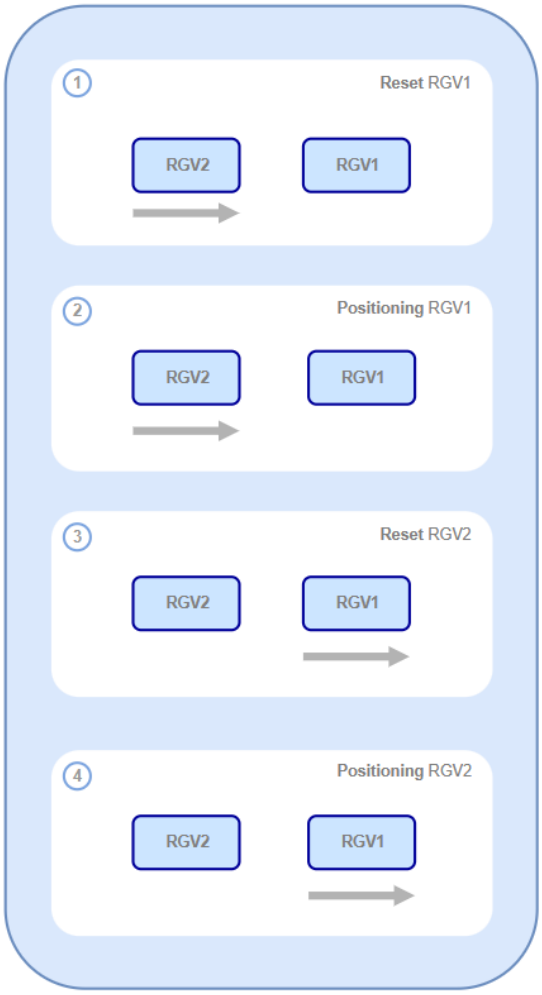


Figura 51 – Sequência de comandos do *Master Controller*

A sequência consiste em 4 passos:

1. *Reset* do RGV da frente (RGV1):
    - O RGV1 é reiniciado assim que parar na primeira posição definida na curva;
2. Posicionamento do RGV da frente:
    - O *Master* envia a nova posição de destino (próxima posição na curva) ao RGV1;

3. *Reset* do RGV de trás (RGV2):
    - O RGV2 é reiniciado assim que parar a uma pequena distância definida do RGV de frente;
    - O RGV1 inicia o movimento para a próxima posição;
4. Posicionamento do RGV de trás:
    - O *Master* envia a nova posição para o RGV2.

Este envio de comandos é cíclico, repetindo-se até se registarem todas as posições configuradas na curva.

Assim sendo, para resolver o problema da não deteção nas curvas do sensor linear, são analisadas várias hipóteses.

Hipótese 1: Utilização de 2 sensores lineares

Uma possível forma de contornar o problema seria a utilização de dois sensores lineares de distância nas extremidades do RGV, de modo a abranger uma área de deteção maior. Para isso foi analisado o sensor ótico de distância O1D100 da IFM, visível na Figura 52.

Na Tabela 7 podem-se verificar algumas características deste sensor.

Figura 52 – Sensor ótico de distância O1D100 [34]

Tabela 7 – Características do sensor O1D100

	O1D100
Função	. Deteção ótica confiável da distância com elevado alcance . Uso em aplicações com supressão de fundo . Duas saídas de comutação, uma delas programável como saída analógica
Taxa de leitura [Hz]	≤ 50
Distância de leitura [mm]	200 - 10000
Preço [€]	314.60

Apesar destes sensores terem as características pretendidas para completar este problema, o grau de sincronismo necessário e a existência de outros sensores mais adequados fazem com que esta hipótese seja descartada.

Hipótese 2: Utilização de sensores LiDAR

Além dos sensores lineares de distância, existe também outro tipo de sensores, os sensores *Light Detection And Ranging* (LiDAR). LiDAR é uma tecnologia semelhante ao radar. Enquanto que os sistemas radar emitem ondas de rádio e medem o que é devolvido, o LiDAR envia e recebe pulsos de luz. Esta tecnologia tem sido amplamente utilizada em múltiplas áreas, incluindo o mapeamento de superfícies e vegetação, estudos urbanos e geociências. LiDAR é uma boa ferramenta para captar informação altamente precisa das coordenadas 2D ou 3D de uma área em estudo [20][21].

Com o objetivo de utilizar este tipo de tecnologia, fez-se um estudo de vários sensores e a sua capacidade de evitar colisões entre os veículos.

Figura 55 – OMD10M [37]

Figura 53 – scanGrid2 [35]

Figura 54 – TIM351 [36]

Tabela 8 – Características dos sensores LiDAR

	OMD10M	scanGrid2	TIM351
Função	. Deteção de objetos e posição . Tamanho compacto . Possui dois planos de deteção	. Deteção de objetos localizados numa área configurável . O alcance de medição é dividido em 2 campos	. Utiliza a tecnologia HDDM+, que leva a um aumento na precisão da medição . Baixo consumo energético . Comissionamento rápido e simples . Possui 3 campos de medição
Distância de leitura [mm]	4000	4000	8000
Imunidade à luz ambiente [klx]	≤ 60	≤ 10	≤ 80
Ângulo de medição	100° - horizontal 9° - vertical	150° horizontal	270° horizontal
Preço [€]	1057,90	1715.40	2389.20

O sensor OMD10M da Pepperl+Fuchs, Figura 55, tal como os outros analisados, consegue detetar objetos a grandes distâncias. A principal vantagem relativamente aos outros é o poder de detetar coordenadas verticalmente e horizontalmente, ou seja, analisar

coordenadas 3D. Apesar desta vantagem, o ângulo de medição horizontal não é suficiente para evitar as colisões entre RGVs, como está exemplificado na Figura 56.

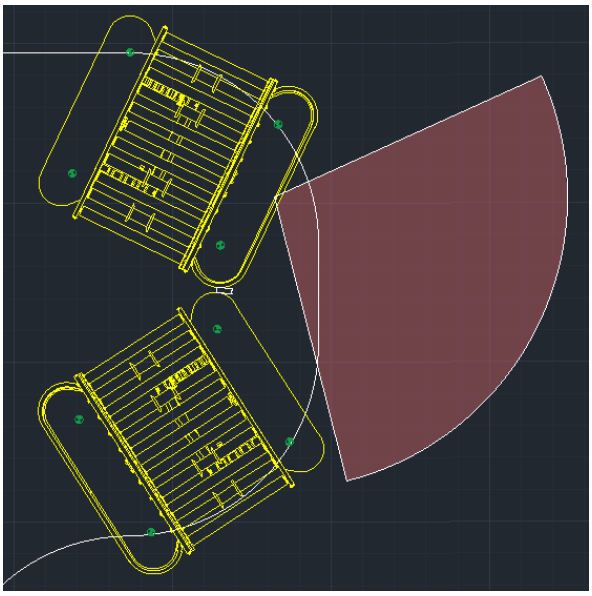


Figura 56 – Sensor com ângulo de medição 100°

O modelo scanGrid2 da SICK, Figura 53, apenas deteta coordenadas 2D, isto é, realiza uma varredura de posições na horizontal, tal como o sensor TIM351, também da SICK. A principal diferença entre estes dois sensores é o ângulo de medição e o número de campos de medição.

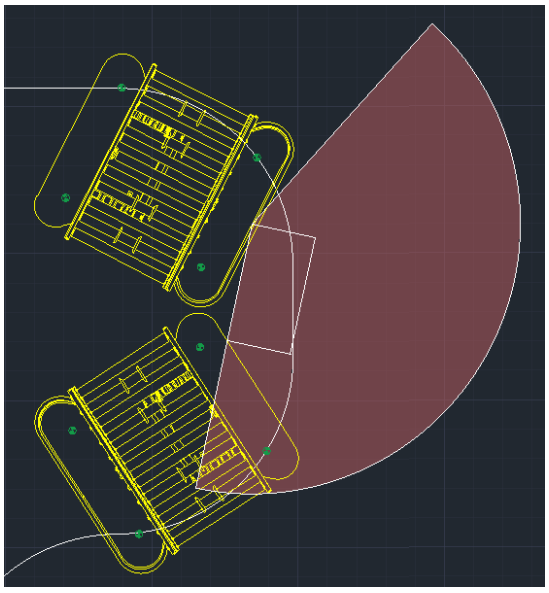


Figura 57 – Sensor com ângulo de medição 150°

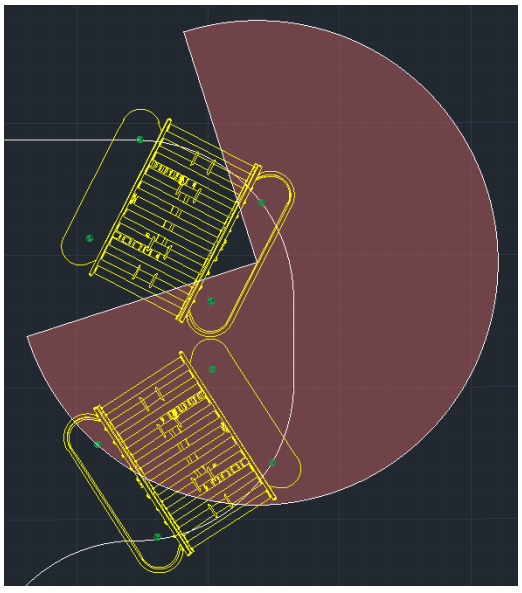


Figura 58 – Sensor com ângulo de medição 270°

Tal como se pode verificar pela Figura 57, um ângulo de medição de 150° chega para detetar os RGVs nas curvas mais apertadas, no entanto, o TIM351 possui um ângulo de medição de 270°, podendo ser configurado livremente em três campos, através do *software* de programação.

Ainda que o TIM351 seja mais caro, todas as suas características são superiores à dos outros sensores analisados, assim como a presença de 3 campos de medição configuráveis torna-se uma mais-valia para este problema e para o próximo objetivo.

4.5.3. ALINHAMENTO COM TRANSPORTADORES

O último desafio deste projeto é o registo automático das posições de alinhamento entre o transportador do RGV e os transportadores do armazém, e tal como nos outros objetivos, este processo é realizado manualmente pelo operador. Por vezes, o comprimento dos transportadores varia, o que pode causar um desalinhamento e, consequentemente, a queda das cargas.

Figura 59 – Alinhamento dos transportadores

Tal como já foi mencionado, o RGV possui um sistema de fotocélulas refletoras que tem por objetivo a comunicação com os outros transportadores no armazém. Porém, este sistema apenas transmite o estado da transferência, isto é, se o RGV está presente ou não.

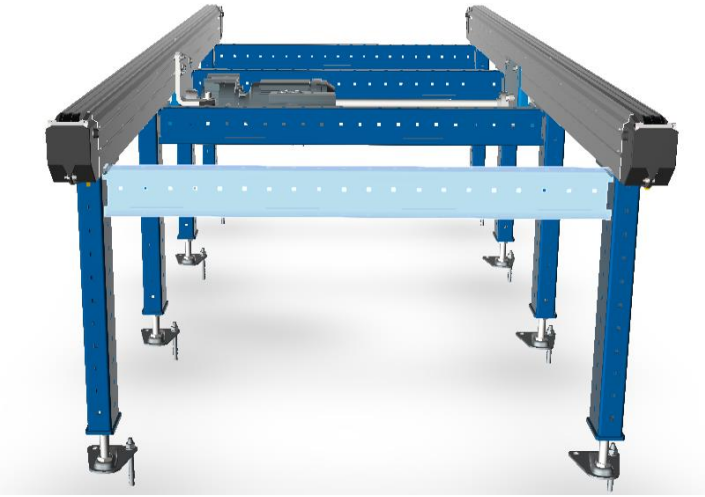


Figura 60 – Transportador a ser detetado

Os transportadores que se pretende detetar possuem um aspeto semelhante ao da Figura 60. Portanto, a forma mais fácil para detetar o alinhamento é a deteção da posição das suas pernas.

Para dar a volta a este problema, são analisadas duas hipóteses de resolução: a utilização de etiquetas RFID ou a utilização de sensores LiDAR.

Hipótese 1: Etiquetas RFID

Na primeira possibilidade analisada, teve-se em conta a reutilização do leitor RFID do registo das curvas e agulhagens, tal como se pode ver na Figura 61. A hipótese consiste na instalação do leitor RFID no RGV, numa posição onde ficaria exatamente à mesma distância das duas pernas do transportador do veículo, isto é, numa posição média. As *tags* seriam colocadas em cada perna do transportador. O leitor quando passasse pelas *tags* registaria as suas posições, sendo que a posição final de alinhamento seria a média das posições registadas.

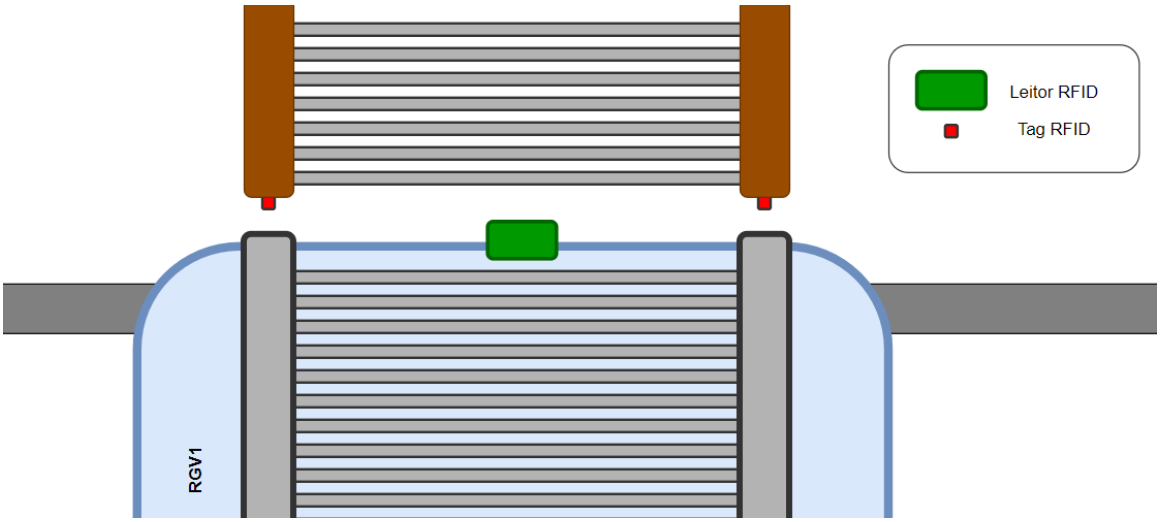


Figura 61 – Esquema de alinhamento por RFID

Uma adversidade a esta hipótese é a falta de precisão. Desvios de cerca de 5 mm provocam a queda das cargas, pelo que estes valores dificilmente são obtidos através de RFID, mesmo com uma boa configuração.

Hipótese 2: Utilização de sensores LiDAR

De modo a reduzir os custos e os tempos de comissionamento, o alinhamento pode ser implementado com auxílio de um sensor já existente. Nesta hipótese é considerada a reutilização do TIM351.

Como já foi mencionado, o TIM351 possui três campos configuráveis de deteção. Tendo esta característica em conta, é possível definir estes campos de forma a detetar as pernas do transportador.

Tal como está representado no esquema da Figura 62, o veículo desloca-se a uma certa velocidade da esquerda para a direita. O terceiro campo tem como objetivo a deteção da primeira perna do transportador de maneira a diminuir a velocidade do RGV. Os outros dois campos têm a função de detetar, alinhar e registar a posição.

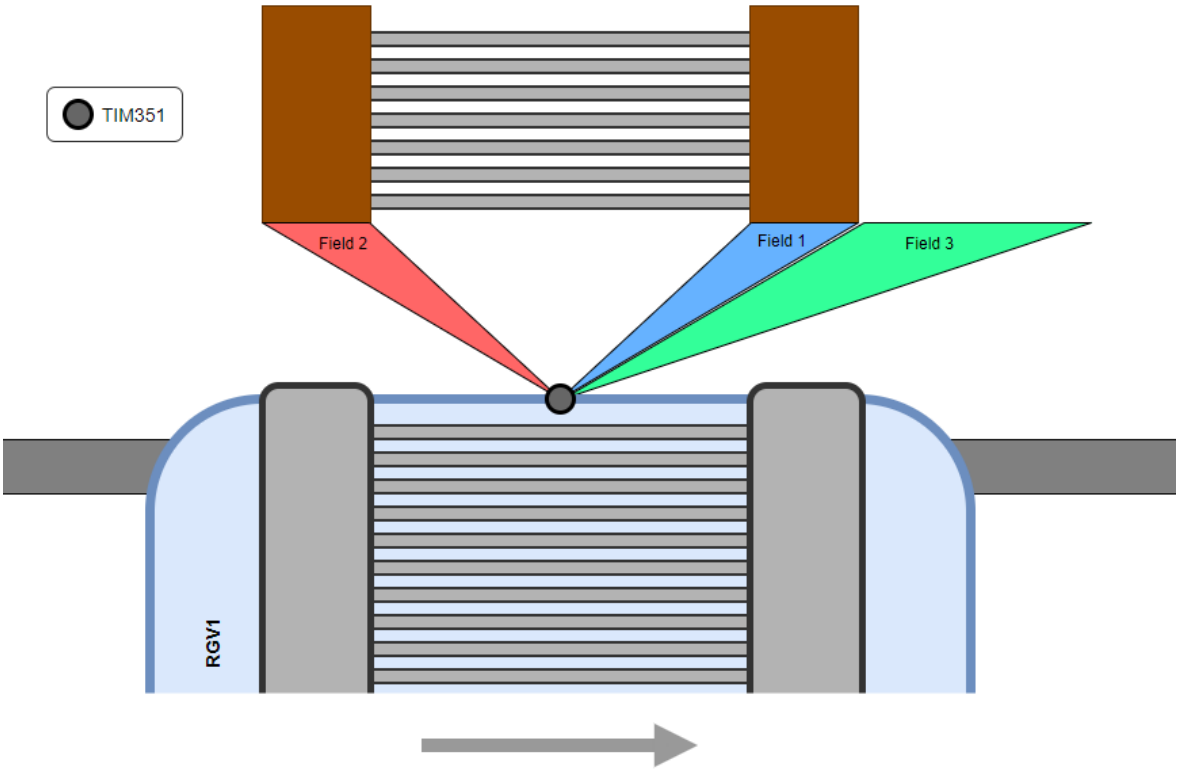


Figura 62 – Esquema de alinhamento através de sensores LiDAR

Em suma, no registo dos limites das curvas e agulhagens será utilizada a tecnologia RFID, e na deteção de RGVs e no alinhamento dos transportadores serão aplicados sensores LiDAR.

Dado como concluída a análise e a escolha das hipóteses ideais, seguem-se no próximo capítulo as suas implementações, desde a configuração dos sensores até ao desenvolvimento das rotinas.

5. IMPLEMENTAÇÃO

5.1. SOFTWARE

Tal como já foi referido, num circuito de RGVs são necessários dois tipos de PLC: o PLC do RGV e o PLC do *Master Controller*. Neste projeto o PLC utilizado para o RGV é o modelo S7-1200 e para o MC é o S7-1500, ambos da Siemens. O programa utilizado no processo de criação das rotinas e na programação dos PLCs é o TIA Portal, fornecido também pela Siemens.

As rotinas são grupos de funções que trabalham em ciclo. Estas consistem em funções e bases de dados. As funções podem ser divididas em dois tipos: o tipo *Functions* (FC) e os *Function Blocks* (FB).

Os FCs são blocos de código ou sub-rotinas que não possuem memória dedicada. Já os FBs são blocos de código que armazenam os seus valores permanentemente numa instância

de bases de dados, de maneira a que estes valores permaneçam disponíveis depois do bloco ser executado. Uma vantagem da utilização dos FBs é a instância *Data Block* (DB). O DB permite ter memória diretamente dedicada ao FB, funcionando como uma base de dados da função. Isto é importante para poder reencaminhar informação de um ciclo para outro. Durante cada ciclo, o FC perde o seu conteúdo enquanto o FB armazena a sua memória no seu bloco associado. O FC é por isso utilizado principalmente para cálculos aritméticos.

5.2. LINGUAGENS DE PROGRAMAÇÃO

Como já foi mencionado nos capítulos anteriores, o PLC é um computador especialmente adaptado ao controlo industrial. Um PLC é composto por uma unidade central que, em ciclo, copia o estado das entradas para a memória, executa as operações definidas no programa do utilizador escrevendo os resultados em memória, e copia os resultados da memória para a unidade de saída [22].

A organização PLCopen, em busca de uniformizar as linguagens de programação para o controlo industrial, desenvolveu a norma IEC-61131. Esta norma inclui a definição do *Sequential Function Chart* (SFC), usado para estruturar os programas, e quatro linguagens de programação:

- Lista de instruções (IL)
- Blocos lógicos
- LADDER
- Texto estruturado

No programa TIA Portal, da Siemens, é permitido apenas a utilização das linguagens LADDER e texto estruturado.

LADDER é uma linguagem gráfica que segue muito de perto os métodos de desenho esquemático. Nesta linguagem as variáveis são representadas por contactos, podendo ser normalmente abertos ou normalmente fechados. Consoante o seguimento lógico de uma

ou várias entradas, podem ser correspondidas saídas. As saídas funcionam como atuadores ou permitem a escrita de valores em áreas de memória, podendo ser lidas como se fossem entradas. O conjunto de entradas e saídas constituem um bloco. Estes são executados de cima para baixo e da esquerda para a direita, só se iniciando um depois de terminado o anterior. Quando o último bloco é executado, o programa regressa ao primeiro e repete, entrando em ciclo.

Texto estruturado é uma linguagem descritiva (textual) que disponibiliza o acesso a todos os recursos definidos na norma IEC-61131. É a linguagem mais sofisticada das propostas sendo que as demais linguagens, quando utilizadas, geram texto estruturado para a compilação final da aplicação. Esta linguagem é composta por *statements* separados por ponto e vírgula. Os *statements* utilizam funções predefinidas e sub-rotinas de programa para alterar as variáveis da aplicação. As variáveis podem ser valores explicitamente definidos, variáveis armazenadas internamente, ou valores de entradas e saídas associadas ao *hardware* do projeto. Texto estruturado é uma solução emergente, ganhando cada vez mais notoriedade uma vez que se assemelha bastante ao C e ao Pascal.

Com o constante aumento da capacidade de cálculo dos PLCs, a linguagem LADDER tem vindo a manifestar a sua fraqueza, assistindo-se à sua substituição pelo texto estruturado. No entanto, LADDER tem a vantagem de ser mais simples na gestão das entradas e saídas. Por estas razões, estas são as duas linguagens utilizadas neste projeto.

5.3. REGISTO DE POSIÇÕES DAS CURVAS E AGULHAGENS

Para registar as posições iniciais e finais das curvas e agulhagens, é implementada a hipótese da utilização de RFID, presente no capítulo anterior. O leitor RFID escolhido foi o IUT-F190-B40-2V1D-FR1-01 da Pepperl+Fuchs, porém, por falta de *stock*, recorreu-se à utilização do modelo IUH-F190-V1-FR1-01, também do mesmo fabricante. Ambos os modelos apresentam características semelhantes (como o funcionamento em UHF, a mesma distância máxima de leitura, preços idênticos, entre outros), sendo que a única diferença entre eles é a interface de comunicação com o PLC. O modelo IUT-F190-B40-2V1D-FR1-01 comunica diretamente com o PLC através de uma comunicação física

Ethernet, enquanto que no caso do leitor IUH-F190-V1-FR1-01, a comunicação entre o leitor e o PLC é mantida através de uma interface de controlo intermédia. Posto isto, foi necessário realizar a compra adicional da unidade de interface de controlo IC-KP2-2HB17-2V1D, visível na Figura 63.

Figura 63 – Interface de controlo IC-KP2-2HB17-2V1D

Assim, para conectar o leitor RFID ao PLC, é necessário fazer as ligações presentes no esquema da Figura 64. O leitor liga-se à interface de controlo através de um conector M12, que vai ser ligado ao PLC via *Ethernet*. Este conector também alimenta o leitor.

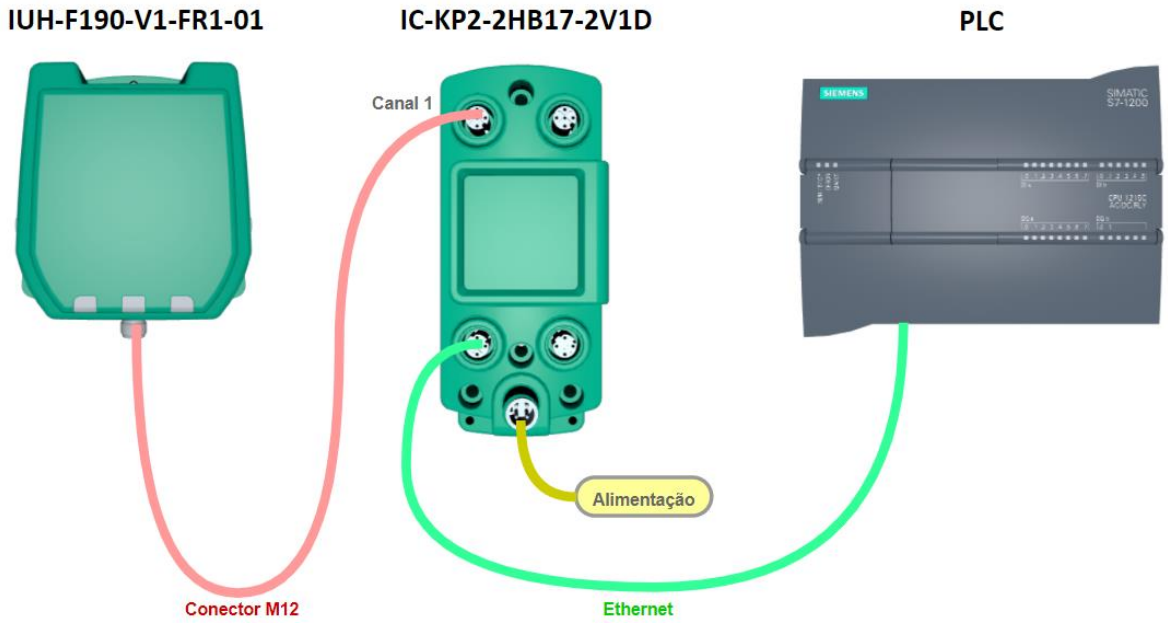


Figura 64 – Esquema de ligações entre o leitor RFID e o PLC

Após feitas as ligações, procede-se para a configuração do sensor.

5.3.1. CONFIGURAÇÃO DO SENSOR

O leitor utilizado opera em UHF. A utilização de UHF traz várias vantagens, entre as quais: o aumento de alcance de deteção, maiores taxas de transferência de dados, e *tags* mais económicas e com mais memória. O leitor IUH-F190-V1-FR1-01 é a versão europeia, ou seja, opera de acordo com a norma EN302208, entre 865 MHz e 868 MHz.

Memória da *tag*

As *tags* UHF utilizadas são o modelo IUC76-F157-T17-M-FR1, também da Pepperl+Fuchs. Cada *tag* tem um custo de 8.60 €.

A estrutura da memória de uma *tag* está dividida em quatro segmentos, como está indicado na Figura 65.

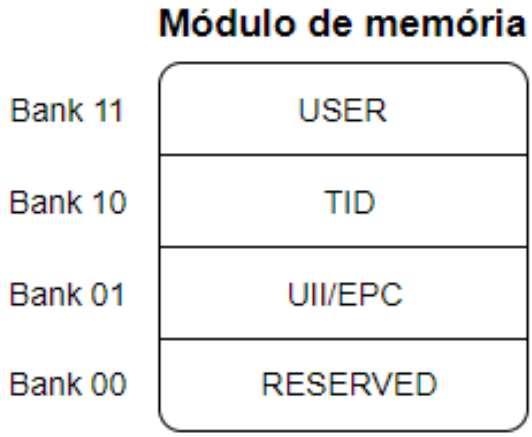


Figura 65 – Estrutura da memória de uma *tag*

O *Bank* 00 é um segmento reservado para a informação da gestão da *password*.

O *Bank* 01 além de conter o identificar único, possui também um conjunto de informações especiais como o tamanho do identificador e a verificação do local dos dados.

O *Bank* 10 contém o identificador da *tag* que consiste no número de série da *tag*. Este identificador é guardado permanentemente sem poder sofrer qualquer alteração.

O último segmento, *Bank* 11, contém uma área de livre acesso ao utilizador. Esta será a zona onde serão guardados os códigos que se pretende detetar.

Comandos de leitura e escrita

Para aceder ao segmento USER, o leitor possui um conjunto de comandos pré-definidos: os comandos únicos e os comandos contínuos.

Nos comandos únicos tem-se o *Single Read* (SR) e o *Single Write* (SW). Estes comandos significam, respetivamente, a leitura e a escrita única da informação na *tag*, ou seja, apenas é feita uma única tentativa de leitura ou escrita.

Nos comandos contínuos, estão presentes o *Enhanced Read* (ER) e o *Enhanced Write* (EW), que significam, também respetivamente, a leitura e a escrita contínua da informação na *tag*, isto é, o leitor está constantemente a ler ou a escrever informação.

Modo de protocolo

O leitor utilizado possui bastante documentação *online*, encontrando-se bibliotecas já definidas para utilizar estes comandos. As bibliotecas variam consoante a aplicação que se pretende implementar, podendo ter dois protocolos: *Singleframe* e *Multiframe*.

O protocolo *Singleframe* é normalmente implementado em sistemas *Low Frequency* (LF) e HF. Neste protocolo, apenas uma *tag* pode estar a comunicar com o leitor de cada vez.

No protocolo *Multiframe*, cada *tag* detetada transmite uma resposta, ou seja, podem ser detetadas várias *tags* ao mesmo tempo.

Uma vez que este leitor permite uma leitura de várias *tags* presentes em simultâneo no campo de deteção, definiu-se a utilização do protocolo *Multiframe*.

Configuração da interface de controlo IC-KP2-2HB17-2V1D

Visto que o leitor não se conecta diretamente ao PLC, a sua configuração é realizada através da interface de controlo. Inicialmente é necessário importar a biblioteca no TIA Portal. A biblioteca além de conter as funções que invocam os comandos, inclui também as suas bases de dados, a definição e a estrutura das variáveis e as tabelas de visualização de dados. Após este processo de importação, configura-se a ligação virtual no TIA Portal, como pode-se ver na Figura 66.

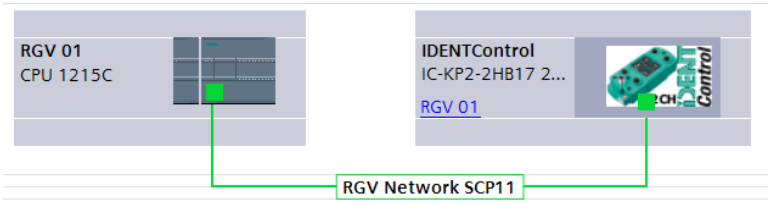


Figura 66 – Ligação virtual entre o PLC e a interface

Para que seja possível realizar uma comunicação de dados entre os dois dispositivos é essencial definir os endereços *Internet Protocol* (IP). O PLC do RGV possui o endereço 192.168.1.2, por isso é necessário atribuir à interface de controlo um endereço IP que esteja incluído na mesma gama de endereços. Assim sendo, a interface pode ficar com o endereço 192.168.1.20, visível na Figura 67.

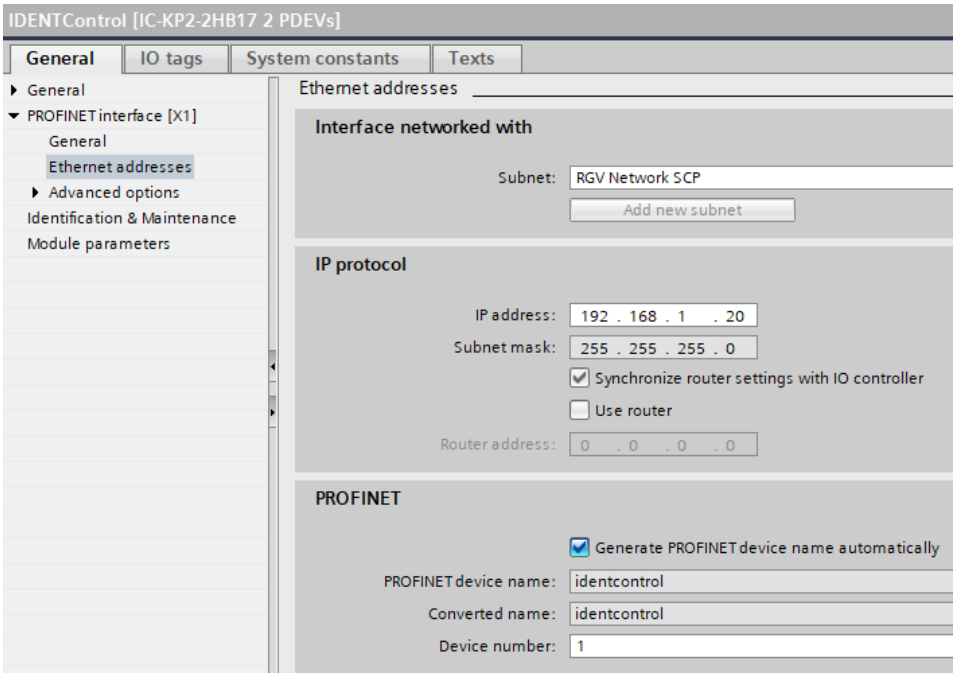


Figura 67 – Atribuição do endereço IP da interface de controlo

Depois de configurar o endereço da interface no TIA PORTAL, é necessário também alterar o endereço através do *Web Browser*.

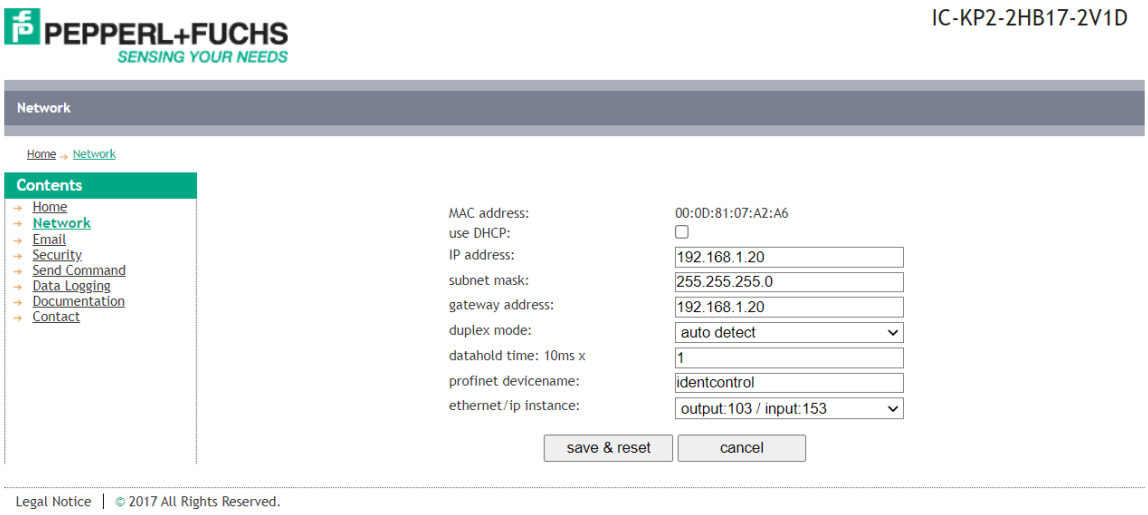


Figura 68 – Alteração do endereço da interface no *Web Browser*

Além do endereço, a interface também deve ter o mesmo nome de dispositivo na ligação criada, neste caso “identcontrol”.

Um parâmetro que deve-se configurar é o *Data Hold Time*. Este parâmetro é o tempo de espera para respostas dos telegramas. Este tempo deve ser o dobro do tempo de ciclo do PLC. O tempo de ciclo do PLC varia entre 1 e 2 ms, no entanto o limite mínimo permitido do *Data Hold Time* é 10 ms, visível na Figura 69.

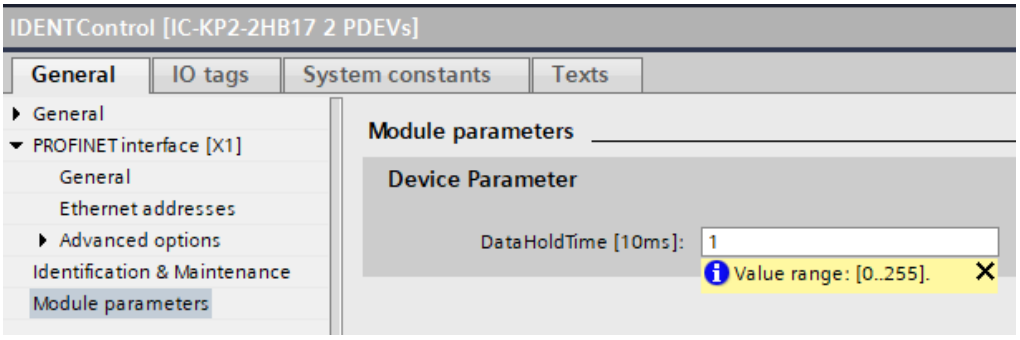


Figura 69 – *Data Hold Time*

Valores muito grandes tornam as transmissões de dados mais lentas, e valores baixos levam a perdas de telegramas.

Para armazenar os valores dos telegramas transportados entre a interface e o leitor, é necessário atribuir zonas de memória à interface. Para isso, recorre-se à importação do módulo "In/Out 64 Bytes" presente no catálogo de *hardware*, como na Figura 70. O tamanho de 64 Bytes é recomendado para transferir uma quantidade máxima de dados por cada telegrama.

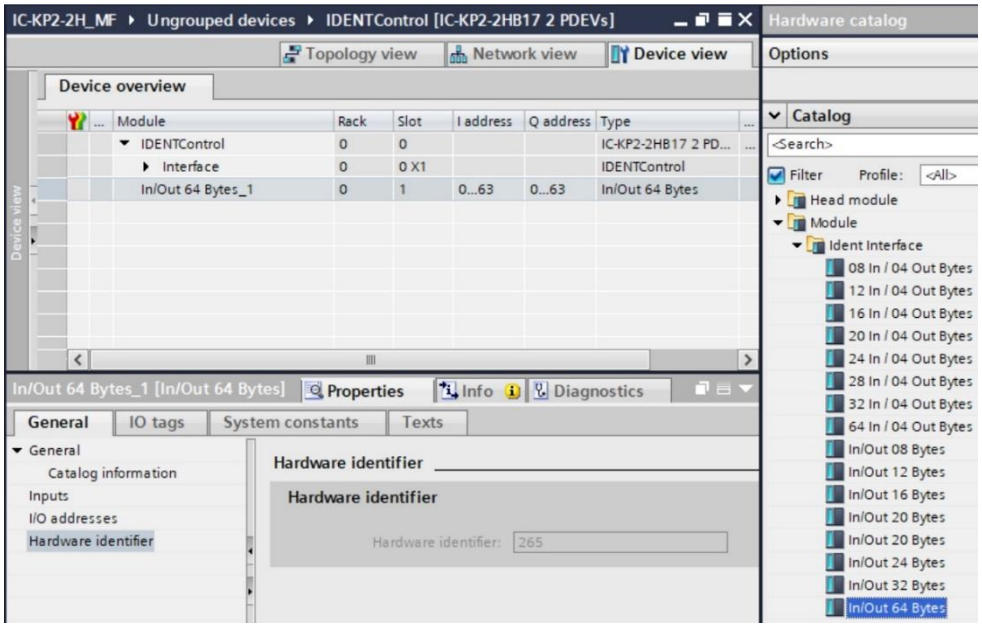


Figura 70 – Configurar zonas de memória de interface

Importação das funções e bases de dados

Após a configuração dos parâmetros, importam-se da biblioteca as funções e as suas bases de dados, Figura 71, assim como o *User Defined Type* (UDT), que é basicamente a estrutura de dados da informação RFID, Figura 72.

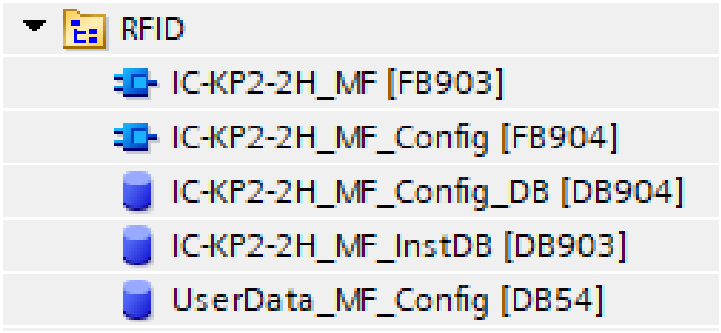


Figura 71 – Funções e bases de dados da biblioteca *Multiframe*

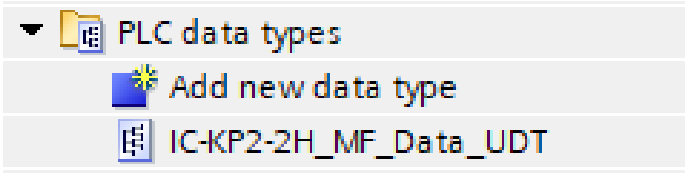


Figura 72 – Importação da UDT

O FB904, IC-KP2-2H_MF_Config, é a função que gere os comandos de leitura e escrita, no modo *Multiframe*, com acesso automático dos parâmetros. O FB903, IC-KP2-2H_MF, possui a utilidade, mas não fornece o acesso automático. Os DB904 e DB903 são, respetivamente, as bases de dados correspondentes às funções mencionadas.

O FB904, além de atribuir os comandos pretendidos, também permite configurar os parâmetros do leitor. Dentro destes parâmetros encontram-se o *Antenna Polarization* (AP), que define o plano de polarização da antena, presente na Figura 73, e o *Power Transmit* (PT), que controla o alcance de deteção conforme a potência de transmissão.

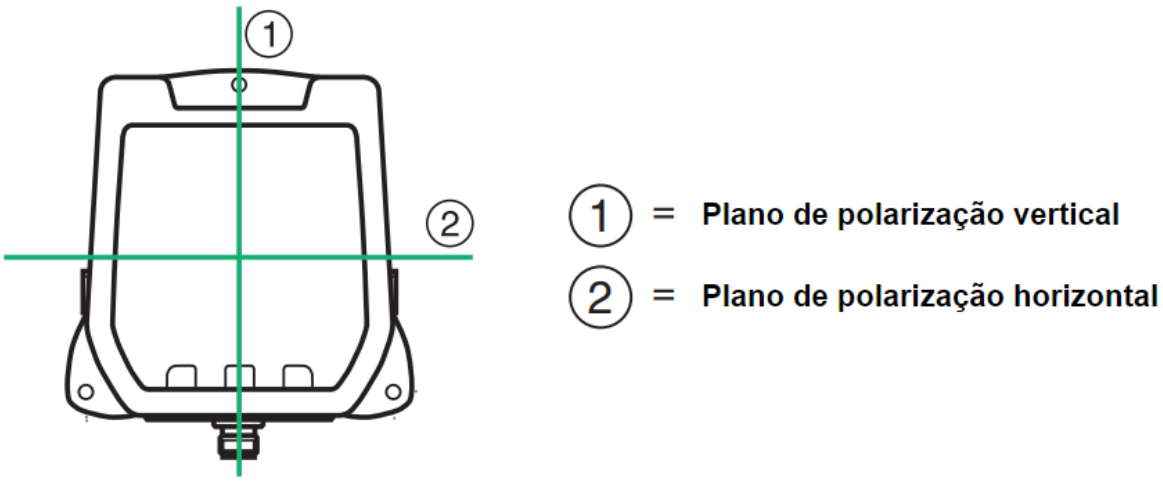


Figura 73 – Parâmetro de configuração

O DB54, UserData_MF_Config, é uma base de dados, baseada no UDT importado, que contém a informação presente no segmento USER da memória de cada *tag*. A informação consiste nos dados de escrita, leitura, comandos especiais, número de *tags* detetadas e informações das *tags*.

A interligação entre as funções e as bases de dados funcionam de acordo com o esquema presente na Figura 74.

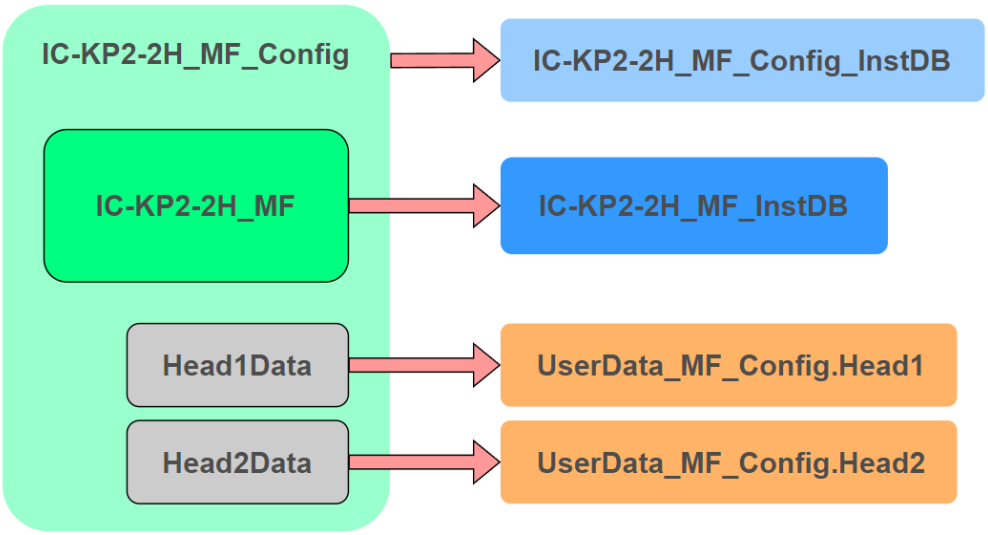


Figura 74 – Esquema do programa com configuração de parâmetros automática

O esquema representa o comportamento do programa numa configuração de parâmetros automática, onde cada função está associada a uma instância de base de dados e os valores resultantes, Head1Data e Head2Data, são armazenados no DB UserData_MF_Config.

A interface de controlo utilizada tem capacidade para trabalhar com dois leitores em simultâneo, logo este DB possui informação proveniente de ambos, o Head1 e o Head2.

UserData_MF_Config

	Name	Data type
1	Static	
2	Head1	"IC-KP2-2H_MF_Data_UDT"
3	WriteData	Array[0..59] of Byte
4	ReadData	Array[0..59] of Byte
5	SpecialCommand	Array[0..63] of Byte
6	Number_of_Tags	Array[0..3] of Byte
7	Tag_out_of_zone	Array[0..59] of Byte
8	Tag_Information	Array[0..4] of Byte
9	Head2	"IC-KP2-2H_MF_Data_UDT"
10	WriteData	Array[0..59] of Byte
11	ReadData	Array[0..59] of Byte
12	SpecialCommand	Array[0..63] of Byte
13	Number_of_Tags	Array[0..3] of Byte
14	Tag_out_of_zone	Array[0..59] of Byte
15	Tag_Information	Array[0..4] of Byte

Figura 75 – Base de dados da informação proveniente dos leitores

5.3.2. LEITURA E ESCRITA CONTÍNUA DE INFORMAÇÃO

Após concluída a configuração do sensor, é necessário criar uma função que permita ativar a leitura ou a escrita de valores nas *tags*, através do envio de comandos e parâmetros para o leitor. Neste projeto, pretende-se manter uma leitura contínua das *tags*, de maneira a que o leitor esteja constantemente a detetar ao longo do circuito. Do mesmo modo, para facilitar o processo da escrita dos códigos de marcação nas *tags*, também se recorre à escrita contínua. Assim, são utilizados os comandos contínuos do leitor, o ER e o EW.

Os comandos são enviados através de uma sequência de ativação e desativação dos parâmetros da Figura 76.

1		"SetRestart"	%M0.2	Bool
2		"InitFinish"	%M0.1	Bool
3		"Start_Configuration_Hea...	%M16.2	Bool
4		"Head1WriteParameter"	%M16.0	Bool
5		"Finish_Configuration_He...	%M16.4	Bool
6				
7		"Head1SingleEnhanced"	%M1.0	Bool
8		"Head1DataFixcode"	%M1.2	Bool
9		"Head1SpecialFixcode"	%M1.4	Bool
10		"Head1SpecialCommand"	%M1.6	Bool
11		"Head1Read"	%M2.0	Bool
12		"Head1Write"	%M2.2	Bool
13		"Head1Quit"	%M2.4	Bool
14				
15		"Head1WordNum"	%MW4	DEC
16		"Head1WordAddress"	%MW8	Hex
17				
18		"Head1NewData"	%M2.6	Bool
19		"Head1Busy"	%M3.0	Bool
20		"Head1Done"	%M3.2	Bool
21		"Head1NoDataCarrier"	%M3.4	Bool
22		"Head1Error"	%M3.6	Bool
23		"Head1Status"	%MB12	Hex
24		"Head1ReplyCounter"	%MB14	Hex
25				
26		"IN_ER"	%M55.0	Bool
27		"IN_EW_CI"	%M55.1	Bool
28		"IN_EW_CO"	%M55.2	Bool
29		"IN_EW_AI"	%M56.5	Bool
30		"IN_EW_AO"	%M55.4	Bool

Figura 76 – Parâmetros de configuração dos comandos

Inicialização do leitor

Sempre que o leitor for utilizado é preciso fazer um *reset* inicial. O *reset* é realizado através da ativação da variável "SetRestart". Se o comando for realizado corretamente, as variáveis "InitFinish" e "Head1Done" são ativadas, e o "Head1Status" fica com o valor 16#00, de acordo com a Figura 77.

Name	Address	Displa...	Monitor ...	Modify ...
"SetRestart"	%M0.0	Bool	FALSE	TRUE
"InitFinish"	%M0.1	Bool	TRUE	
"StartConfiguration_Head1"	%M10.0	Bool	FALSE	
"FinishConfiguration_Head1"	%M10.2	Bool	FALSE	
"WriteParameter_Head1"	%M10.3	Bool	FALSE	
"Head1SingleEnhanced"	%M1.0	Bool	FALSE	
"Head1DataFixcode"	%M1.1	Bool	FALSE	
"Head1SpecialFixcode"	%M1.2	Bool	FALSE	
"Head1SpecialCommand"	%M1.3	Bool	FALSE	
"Head1Read"	%M1.4	Bool	FALSE	
"Head1Write"	%M1.5	Bool	FALSE	
"Head1Quit"	%M1.6	Bool	FALSE	
"Head1WordNum"	%MW2	DEC+/-	0	
"Head1WordAddress"	%MW4	Hex	16#0000	
"Head1NewData"	%M1.7	Bool	FALSE	
"Head1Busy"	%M6.0	Bool	FALSE	
"Head1Done"	%M6.1	Bool	TRUE	
"Head1NoDataCarrier"	%M6.2	Bool	FALSE	
"Head1Error"	%M6.3	Bool	FALSE	
"Head1Status"	%MB7	Hex	16#00	
"Head1ReplyCounter"	%MB8	Hex	16#08	

Figura 77 – Inicialização do leitor

Comando *Enhanced Read Words*

Para utilizar o comando ER é necessário inicialmente ativar a variável "Head1Quit" para travar qualquer operação que estivesse a ser realizada anteriormente. Após esta paragem, é atribuído um número de Bytes à variável "Head1WordNum" (número de *Words* que se pretende ler), e são ativadas as variáveis "Head1SingleEnhanced" e "Head1Read", podendo esta última ser depois desativada. Na Figura 78 é possível verificar este processo.

Name	Address	Displa...	Monitor ...	Modify ...
"SetRestart"	%M0.0	Bool	FALSE	
"InitFinish"	%M0.1	Bool	TRUE	
"StartConfiguration_Head1"	%M10.0	Bool	FALSE	
"FinishConfiguration_Head1"	%M10.2	Bool	FALSE	
"WriteParameter_Head1"	%M10.3	Bool	FALSE	
"Head1SingleEnhanced"	%M1.0	Bool	TRUE	TRUE
"Head1DataFixcode"	%M1.1	Bool	FALSE	
"Head1SpecialFixcode"	%M1.2	Bool	FALSE	
"Head1SpecialCommand"	%M1.3	Bool	FALSE	
"Head1Read"	%M1.4	Bool	FALSE	TRUE
"Head1Write"	%M1.5	Bool	FALSE	
"Head1Quit"	%M1.6	Bool	FALSE	
"Head1WordNum"	%MW2	DEC+/-	2	2
"Head1WordAddress"	%MW4	Hex	16#0000	16#0000

Figura 78 – Leitura contínua de *tags*

Comando *Enhanced Write Words*

A utilização do comando EW é semelhante ao processo da ativação do comando ER. Primeiro é necessário ativar a variável "Head1Quit" para interromper os comandos que se encontram a correr. Depois, para além de atribuir-se o número de Bytes ao "Head1WordNum", é necessário também atribuir a mensagem que se pretende enviar. Esta mensagem está presente na base de dados "UserData_MF_Config".Head1.WriteData.

UserData_MF_Config

	Name	Data type
1	Static	
2	Head1	"IC-KP2-2H_MF_Data_UDT"
3	WriteData	Array[0..59] of Byte
4	WriteData[0]	Byte
5	WriteData[1]	Byte
6	WriteData[2]	Byte
7	WriteData[3]	Byte

Figura 79 – Localização da mensagem que se pretende enviar

Depois de atribuída a mensagem, podem-se ativar as variáveis “Head1SingleEnhanced” e “Head1Write”, como se pode ver na Figura 80.

Name	Address	Displa...	Monitor ...	Modify ...
"SetRestart"	%M0.0	Bool	FALSE	
"InitFinish"	%M0.1	Bool	TRUE	
"StartConfiguration_Head1"	%M10.0	Bool	FALSE	
"FinishConfiguration_Head1"	%M10.2	Bool	FALSE	
"WriteParameter_Head1"	%M10.3	Bool	FALSE	
"Head1SingleEnhanced"	%M1.0	Bool	TRUE	TRUE
"Head1DataFixcode"	%M1.1	Bool	FALSE	
"Head1SpecialFixcode"	%M1.2	Bool	FALSE	
"Head1SpecialCommand"	%M1.3	Bool	FALSE	
"Head1Read"	%M1.4	Bool	FALSE	
"Head1Write"	%M1.5	Bool	FALSE	TRUE
"Head1Quit"	%M1.6	Bool	FALSE	
"Head1WordNum"	%MW2	DEC+/-	2	2
"Head1WordAddress"	%MW4	Hex	16#0000	16#0000

Figura 80 – Escrita contínua de *tags*

Depois da análise das sequências de parâmetros, procede-se para o desenvolvimento da função de controlo de comandos, a FB_RFID_Commands, e a sua respetiva base de dados, DB_RFID_Commands. Também é necessário implementar uma função de conversão de dados para ASCII, uma vez que os códigos são recebidos no formato hexadecimal, e uma base de dados que armazene este valores.

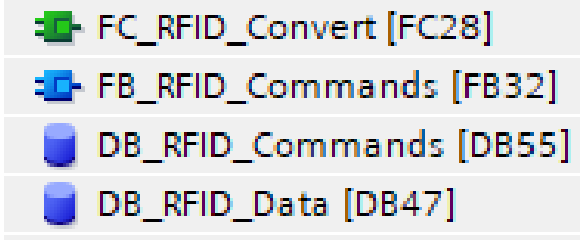


Figura 81 – Blocos criados para as rotinas RFID

Função FB_RFID_Commands

A função FB_RFID_Commands é criada para gerir os controlos de leitura e escrita das *tags*. Inicialmente são implementados seis botões de entrada. Cada botão ativa uma variável através de um pulso, como está representado na Figura 82.

Existe um botão para cada função:

- Parar o comando atualmente a ser decorrido;
- Ler continuamente;
- Escrever continuamente “CI”;
- Escrever continuamente “CO”;
- Escrever continuamente “AI”;
- Escrever continuamente “AO”.

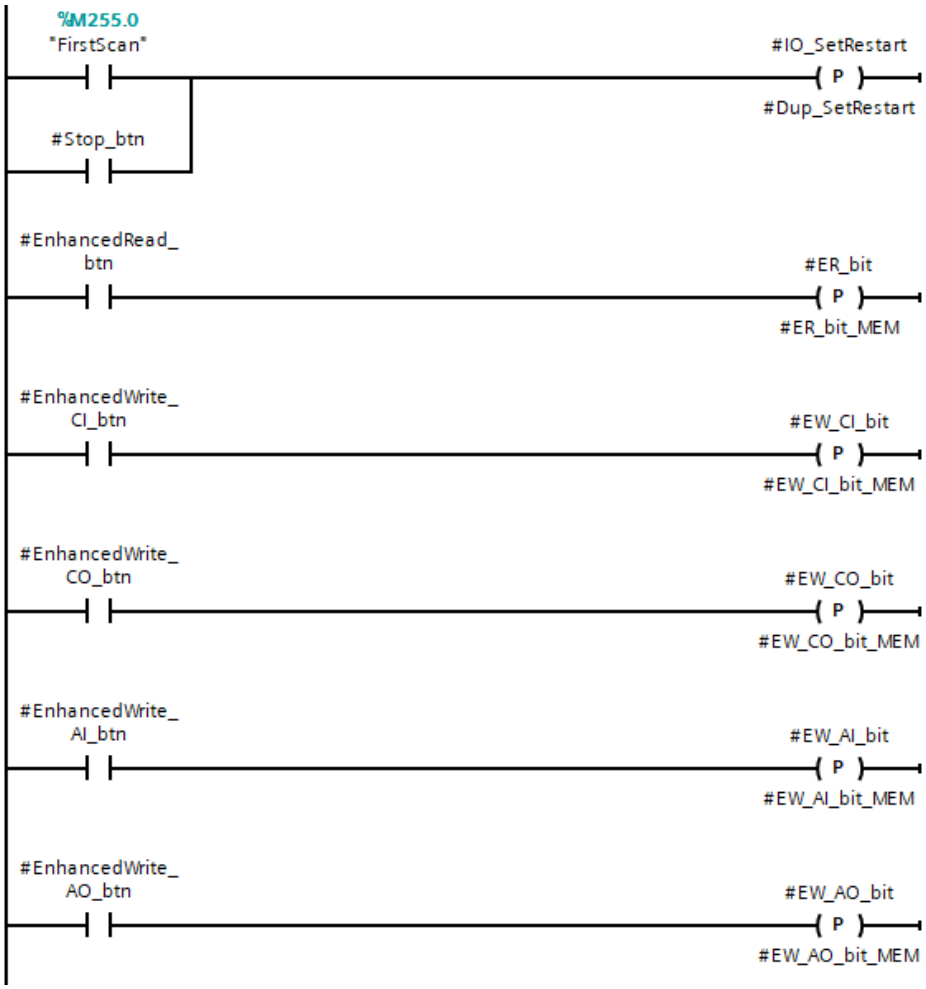


Figura 82 – Implementação dos botões de entrada do sistema RFID

A variável “FirstScan” é corrida sempre que o PLC seja ligado. Na sua saída é ativado o *reset* do leitor, ou seja, sempre que o PLC seja ligado pela primeira vez o leitor é reiniciado.

As sequências das ativações dos parâmetros de leitura e escrita, mencionadas anteriormente, são controladas por variáveis de estado, como está presente na Figura 83. Esta variável de estado “#State” controla a sequência conforme o botão premido:

- State = 1 → Leitura contínua
- State = 2 → Escrita contínua do código “CI”
- State = 3 → Escrita contínua do código “CO”
- State = 4 → Escrita contínua do código “AI”
- State = 5 → Escrita contínua do código “AO”

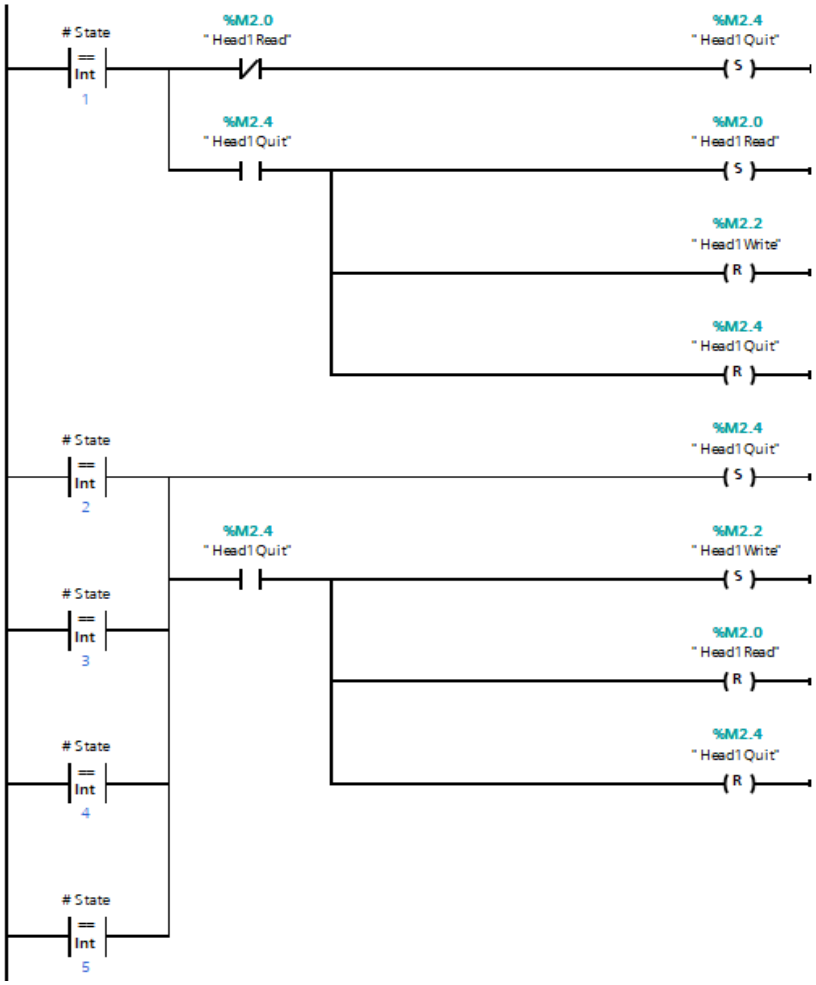


Figura 83 – Implementação das sequências de parâmetros

Na leitura contínua, é ativada a entrada “#EnhancedRead_btn”. Isto leva a um pulso na saída “#ER_bit”. Este processo ativa o parâmetro “Head1SingleEnhanced”, atribui 1 Byte à variável “Head1WordNum”, e coloca o estado a 1.

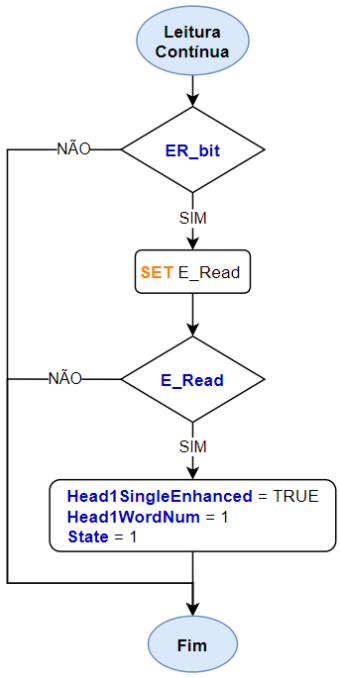


Figura 84 – Parâmetros para leitura contínua

O caso da escrita contínua apresenta uma diferença. Só existe apenas um estado para a leitura contínua, no entanto, a escrita contínua pode conter quatro estados, isto é, quatro códigos. Sempre que se deseja alterar o código de escrita na *tag*, é necessário interromper o processo atual de escrita e iniciar outro. Desta forma, é implementado um novo estado, o estado 0.

Como é possível observar na Figura 85, quando é pressionado um novo comando de escrita e se o estado anterior for diferente de 0, isto é, se existir algum comando em curso, o estado atual fica a 0. Quando o estado é 0, as variáveis de escrita e comando são reiniciadas. Após este *reset*, são atribuídos os códigos e os parâmetros de escrita, e o novo valor de estado.

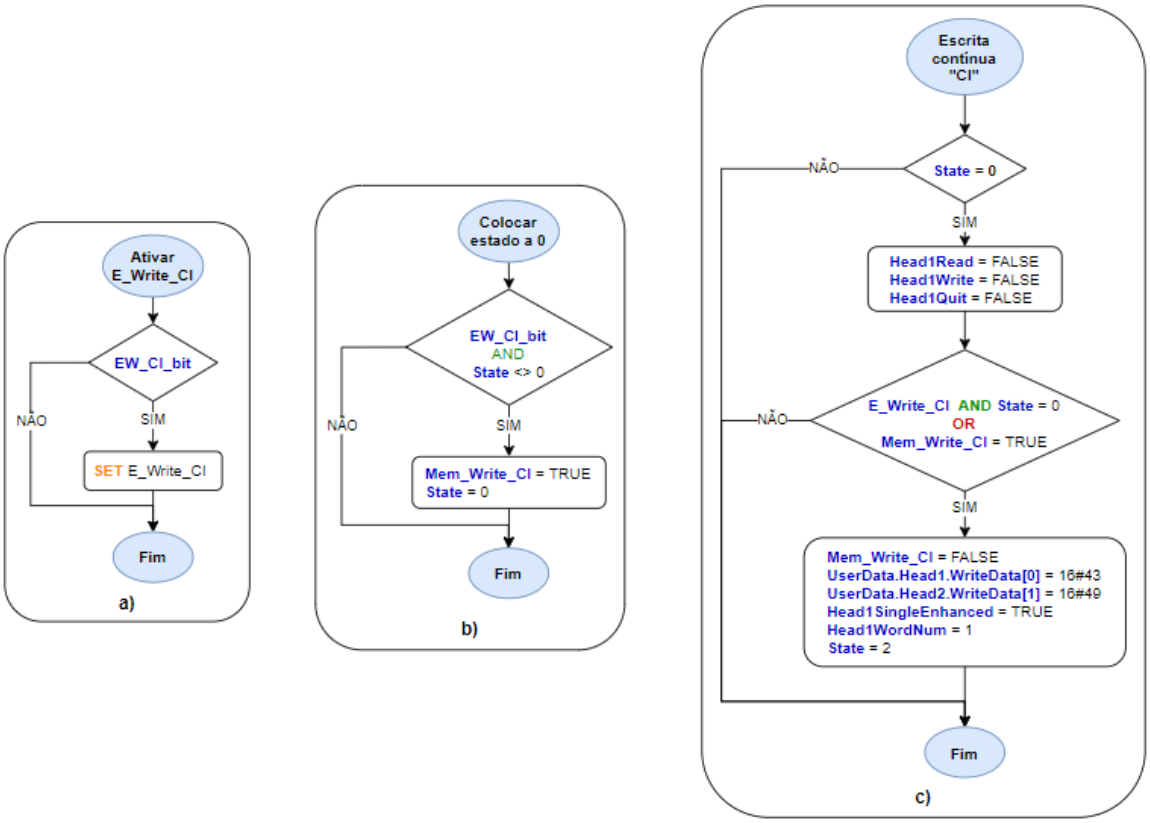


Figura 85 – a) Ativação da variável E_Write_CI
b) Verificação do estado anterior
c) Parâmetros para escrita contínua do código “CI”

Função FB_RFID_Convert

Esta função tem como objetivo a conversão dos valores detetados pelo leitor. Os valores lidos, em hexadecimal, encontram-se no DB UserData_MF_Config. Estes são convertidos para ASCII e armazenados na base de dados DB_RFID_Data.

```
1  "DB_RFID_Data".Read_Data_1[0] := WORD_TO_CHAR("UserData_MF_Config".Head1.ReadData[18]);
2  "DB_RFID_Data".Read_Data_1[1] := WORD_TO_CHAR("UserData_MF_Config".Head1.ReadData[19]);
3
4  "DB_RFID_Data".Write_Data_1[0] := WORD_TO_CHAR("UserData_MF_Config".Head1.WriteData[0]);
5  "DB_RFID_Data".Write_Data_1[1] := WORD_TO_CHAR("UserData_MF_Config".Head1.WriteData[1]);
6
7  IF "Head1Status" = 16#05 THEN
8      "UserData_MF_Config".Head1.ReadData[18] := ' ';
9      "UserData_MF_Config".Head1.ReadData[19] := ' ';
10 END_IF;
```

Figura 86 – Função FB_RFID_Convert

A função também elimina dos registos os códigos lidos pelo leitor. Isto faz com que o programa só guarde o valor enquanto o leitor deteta a *tag*.

5.3.3. ROTINA DO REGISTO DE POSIÇÕES

Tal como já foi mencionado várias vezes ao longo deste documento, um dos principais objetivos deste projeto é a automatização de uma fase do comissionamento onde são registadas as posições limites das curvas e agulhagens. Para isso, é desenvolvida uma rotina que permite executar os processos desta fase de comissionamento, que são atualmente realizados de forma manual.

A rotina criada é constituída pela função FB_RFID_Tag_Data que, de acordo com o código da *tag* detetada, regista a posição atual do RGV na base de dados DB_Circuit_Positions.

DB_Circuit_Positions

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static								
2	Curve	Array[1..24] of Struct							Store the positions of curves
3	Switch	Array[1..24] of Struct							Store the positions of SDs
4	SD_Config	Array[1..24] of Struct							Configuration of each SD
5	Conveyor	Array[1..50] of Struct							Store the positions of conveyors
6	Curve_Config	Array[1..24] of Struct							
7	Segment_Curve	Array[1..36] of Struct							Number of curves in each segment
8	Distance_CI	Array[1..23] of DInt							Distance between CIs
9	Distance_CO	Array[1..24] of DInt							Distance between COs
10	Distance_Curve	Array[1..24] of DInt							Curve distance
11	Num_Of_SDs	Int	24						Number of Switch Devices
12	Num_Of_Curves	Int	24						Number of Curves
13	Num_Of_Segments	Int	36						Number of Segments
14	Position_Offset	DInt	150						Block if less than this distance from an actual ...
15	Actual_Segment	Int	0						Segment where the RGV is
16	Curve_Index	Int	0						
17	Segment_Control	Bool	false						Store the number of curves at first scan
18	Sim_Position	DInt	0						Simulated actual position
19	Curve_IN_Registered	Bool	false						CI Position registered / not registered
20	Curve_OUT_Registered	Bool	false						CO Position already registered / not registered
21	SD_IN_Registered	Bool	false						AI Position already registered / not registered
22	SD_OUT_Registered	Bool	false						AO Position already registered / not registered
23	Save_Position_C_IN	Bool	false						
24	Save_Position_C_OUT	Bool	false						
25	Save_Position_SD_IN	Bool	false						
26	Save_Position_SD_OUT	Bool	false						

Figura 87 – Base de dados DB_Circuit_Positions

Inicialmente, é necessário fazer uma pré-configuração do circuito. A pré-configuração deve conter algumas das características principais que constam a nível do projeto, como o número de curvas, agulhagens e segmentos. A base de dados DB_Circuit_Positions armazena estes valores. Neste caso, os valores utilizados na Figura 87, correspondem aos valores do circuito analisado na Figura 26.

Além destes valores, a pré-configuração também contém outras informações relativas às curvas e agulhagens.

Configuração das curvas

Curve_Config é um vetor do tipo *struct*, com um determinado número de curvas, que possui as informações presentes na Figura 88.

6	Curve_Config	Array[1..24] of Struct	
7	Curve_Config[1]	Struct	
8	Segment	Int	6
9	Curve_IN_Read	Bool	false
10	Curve_OUT_Read	Bool	false
11	Curve_Config[2]	Struct	
12	Segment	Int	6
13	Curve_IN_Read	Bool	false
14	Curve_OUT_Read	Bool	false
15	Curve_Config[3]	Struct	
16	Segment	Int	8
17	Curve_IN_Read	Bool	false
18	Curve_OUT_Read	Bool	false

Figura 88 – Vetor Curve_Config

A variável Segment refere-se ao segmento onde se situa cada curva, e as variáveis Curve_IN_Read e Curve_OUT_Read representam, respetivamente, se a posição de entrada ou de saída já se encontra registada. O número do segmento de cada curva encontra-se no Anexo A.

Configuração das agulhagens

SD_Config é também um vetor do tipo *struct* que possui a configuração de cada agulhagem.

4	SD_Config	Array[1..24] of Struct	
5	SD_Config[1]	Struct	
6	SD_Config[2]	Struct	
7	SD_Config[3]	Struct	
8	IN_1_Segment	Int	2
9	IN_2_Segment	Int	24
10	OUT_1_Segment	Int	3
11	OUT_2_Segment	Int	0
12	Type	Int	2
13	SD_Config[4]	Struct	
14	IN_1_Segment	Int	21
15	IN_2_Segment	Int	0
16	OUT_1_Segment	Int	22
17	OUT_2_Segment	Int	24
18	Type	Int	1

Figura 89 – Vetor SD_Config

No caso das curvas, podem existir várias com o mesmo segmento na entrada e na saída. Já nas agulhagens não existem duas entradas nem duas saídas com o mesmo segmento, por isso é necessário configurar cada uma delas de forma independente, como na Figura 89. A configuração dos segmentos de entrada e saída de cada agulhagem são indicados no Anexo B.

Ao contrário das curvas, as agulhagens são definidas em dois tipos:

- Tipo 1: a agulhagem possui uma entrada e duas saídas;
- Tipo 2: a agulhagem possui duas entradas e uma saída.

Estes tipos também são definidos no vetor SD_Config.

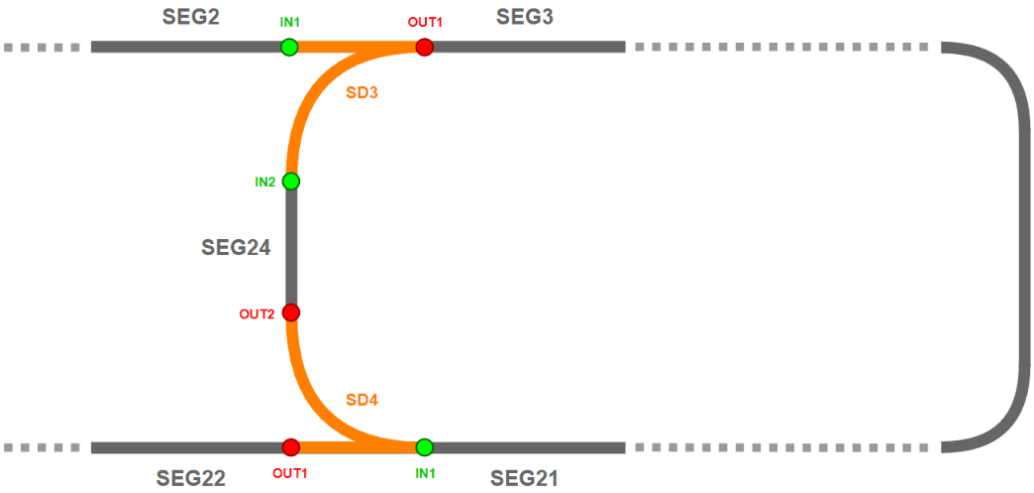


Figura 90 – Esquema de agulhagens

A Figura 90 é um esquema que representa a lógica das agulhagens. É possível observar os dois tipos de agulhagens, o SD3 do tipo 2 e o SD4 do tipo 1. Para detetar ambas as entradas e saídas, o veículo necessita de dar várias voltas ao circuito, passando múltiplas vezes em cada agulhagem.

Função FB_RFID_Tag_Data

A rotina consiste na função FB_RFID_Tag_Data. Através da confirmação de várias condições, esta função guarda a posição atual do RGV de acordo com a informação obtida pelo leitor RFID. A função divide-se em várias *Networks*.

Inicialmente, as duas primeiras *Networks* têm o objetivo de calcular o número de curvas para cada segmento. Quando o PLC é ligado, a variável "FirstScan" é ativada, que por sua vez, coloca a variável Segment_Control como TRUE. Quando o Segment_Control estiver ativo, o vetor Curve_Config, que contém as configurações das curvas, é percorrido, verificando-se o número de vezes que um segmento se repete, através da utilização de dois ciclos FOR. O número de curvas é armazenado no vetor Segment_Curves, na base de dados. A Figura 91 mostra o funcionamento deste cálculo.

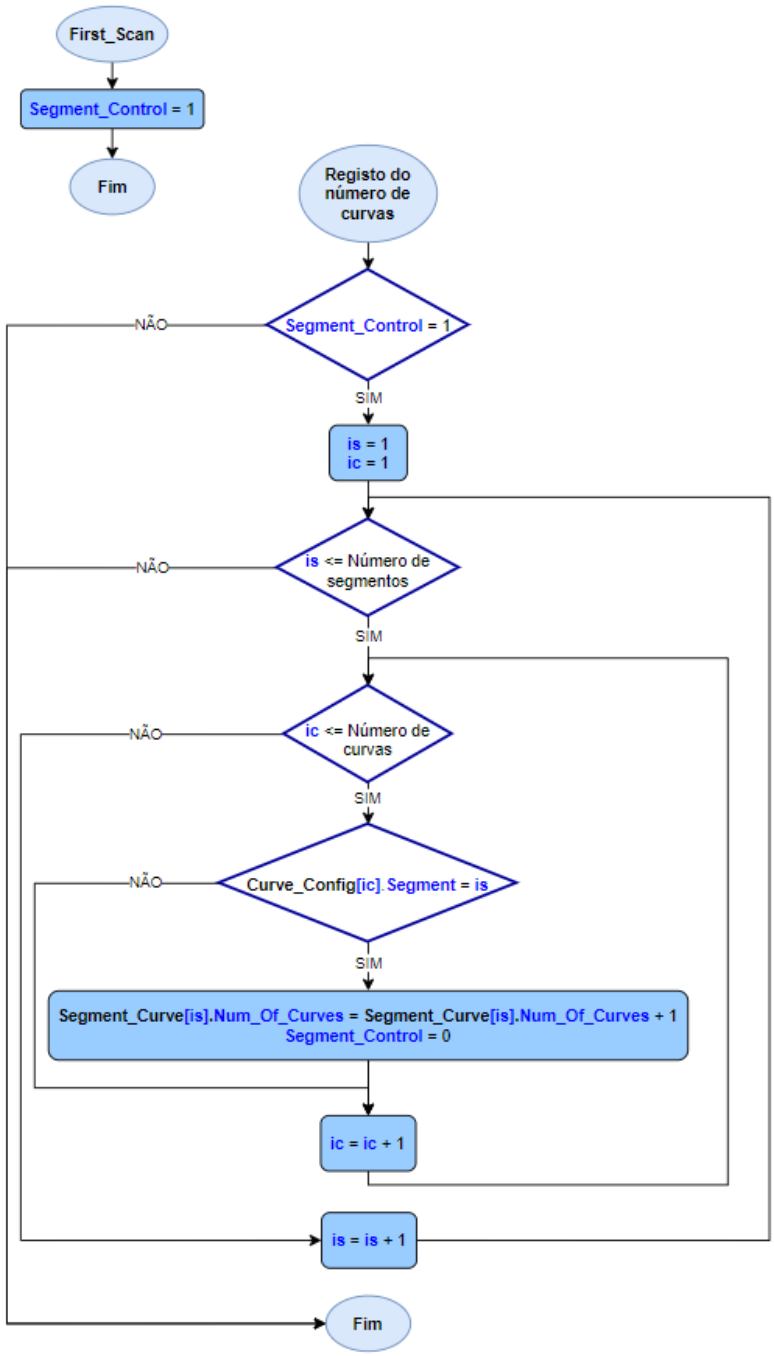


Figura 91 – Cálculo do número de curvas

A terceira e quarta *Network*, têm como finalidade a leitura da mensagem presente na *tag* RFID detetada e a ativação das *flags* de deteção respetivas, como mostra a Figura 92.

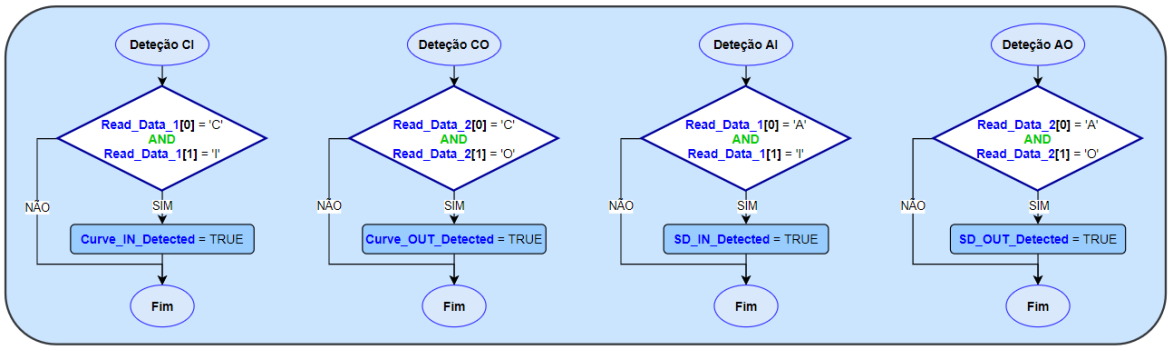


Figura 92 – Ativação das *flags* de deteção de *tags*

O Read_Data_1 refere-se às informações recebidas pelo leitor colocado na dianteira do veículo, e o Read_Data_2 relaciona-se ao leitor da traseira. Ambas as variáveis estão presentes na base de dados DB_RFID_Data, que contém os códigos detetados pelo leitor RFID.

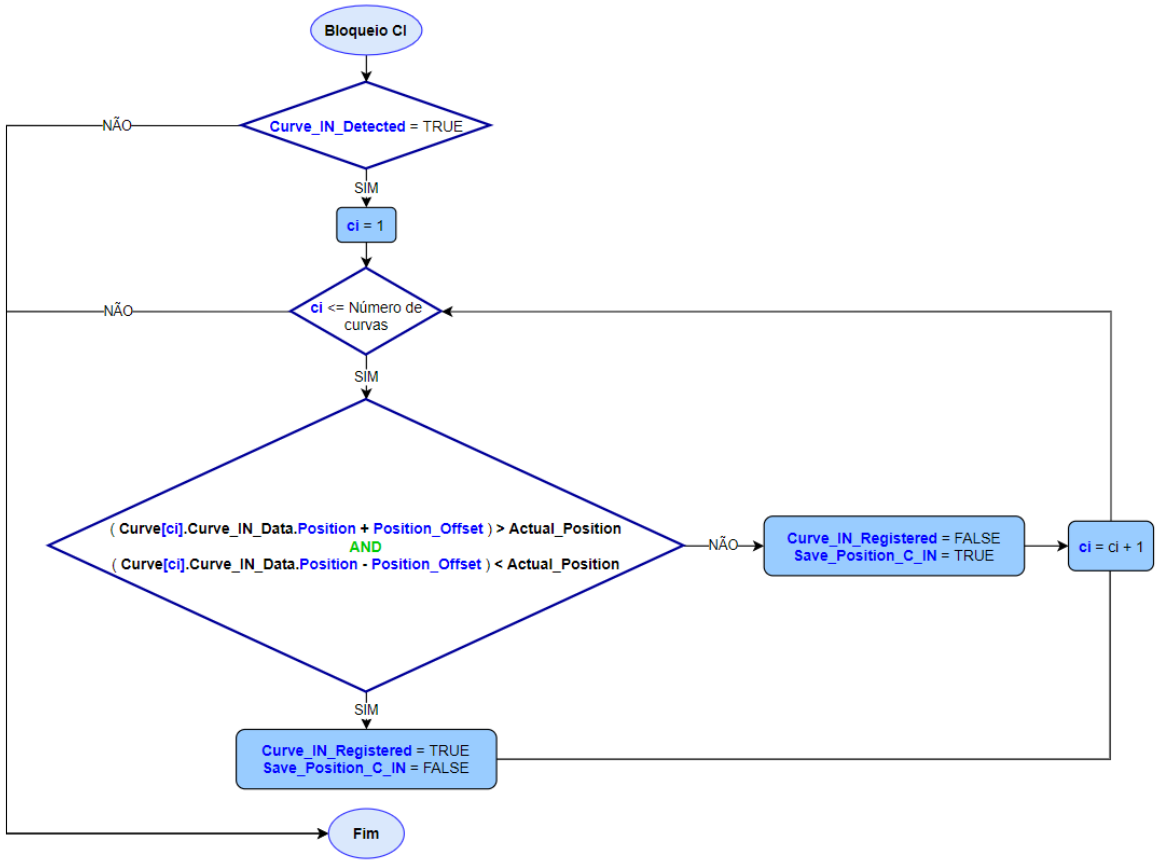


Figura 93 – Bloqueio de registo da posição de entradas de curvas

Depois de uma *flag* de deteção ser ativada, antes da posição ser registada é necessário verificar se a posição está disponível para ser guardada. O fluxograma da Figura 93, atua como um filtro de bloqueio, ou seja, quando é detetada a mensagem da *tag*, verifica se a posição atual se encontra dentro do intervalo de uma posição já registada. Este filtro funciona como um mecanismo de controlo de erros. No caso de o leitor detetar duas vezes a mesma posição, este só regista a primeira vez.

No caso da Figura 93, o vetor Curve, onde são armazenadas as posições das curvas, é percorrido, verificando-se se a posição detetada encontra-se no intervalo de deteção.

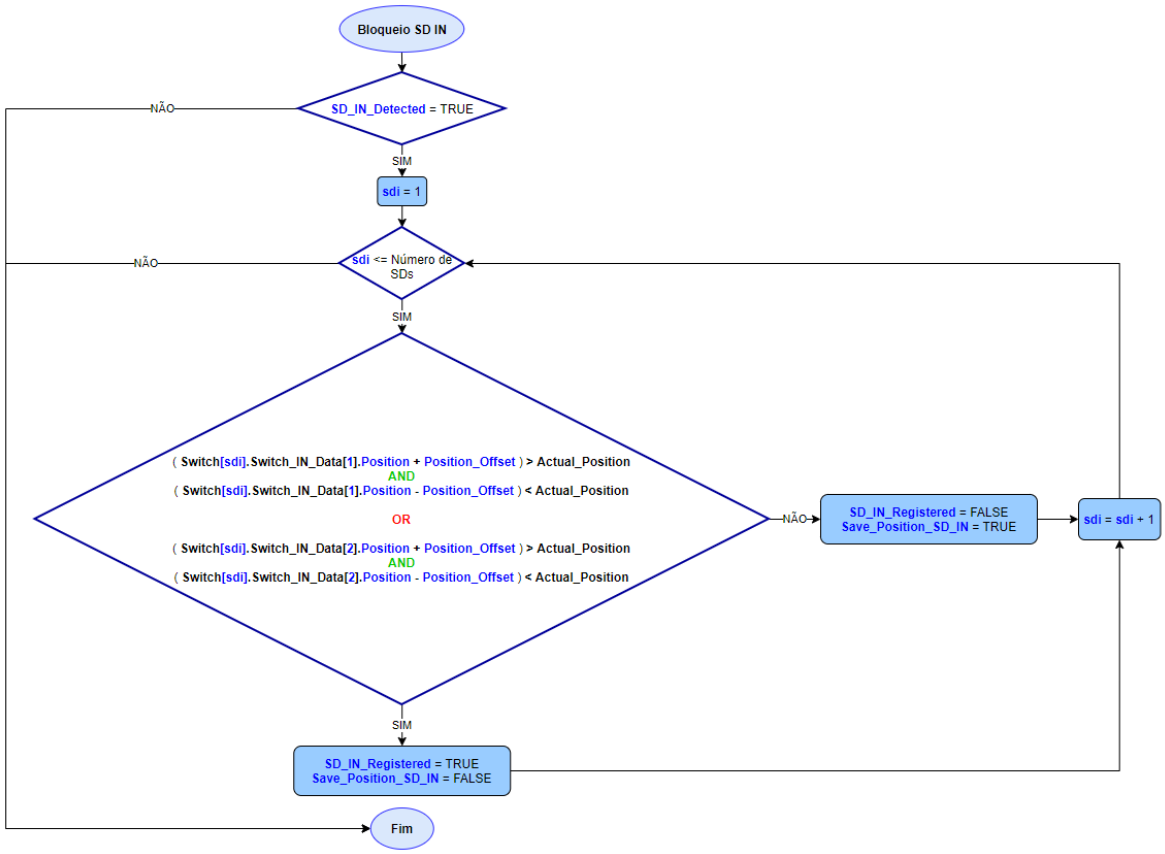


Figura 94 – Bloqueio de registo da posição de entradas de agulhagens

Este filtro é semelhante para a deteção de outros códigos (“CO”, “AI” e “AO”). Nas agulhagens, a diferença é a observação das múltiplas entradas e saídas, como se pode ver na Figura 94.

Após ser validada a possibilidade de registar a posição, é enviado um pulso que regista a posição apenas no instante que começa a detetar, como na Figura 95.

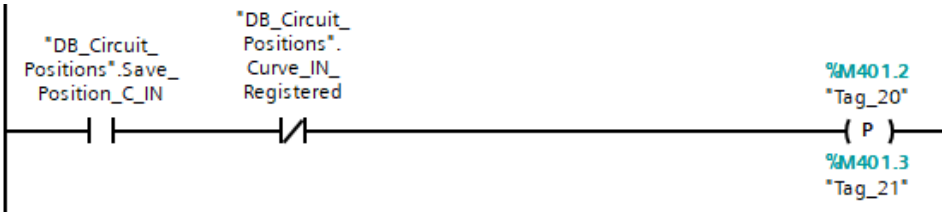


Figura 95 – Validação dos registos de posições de entrada das curvas

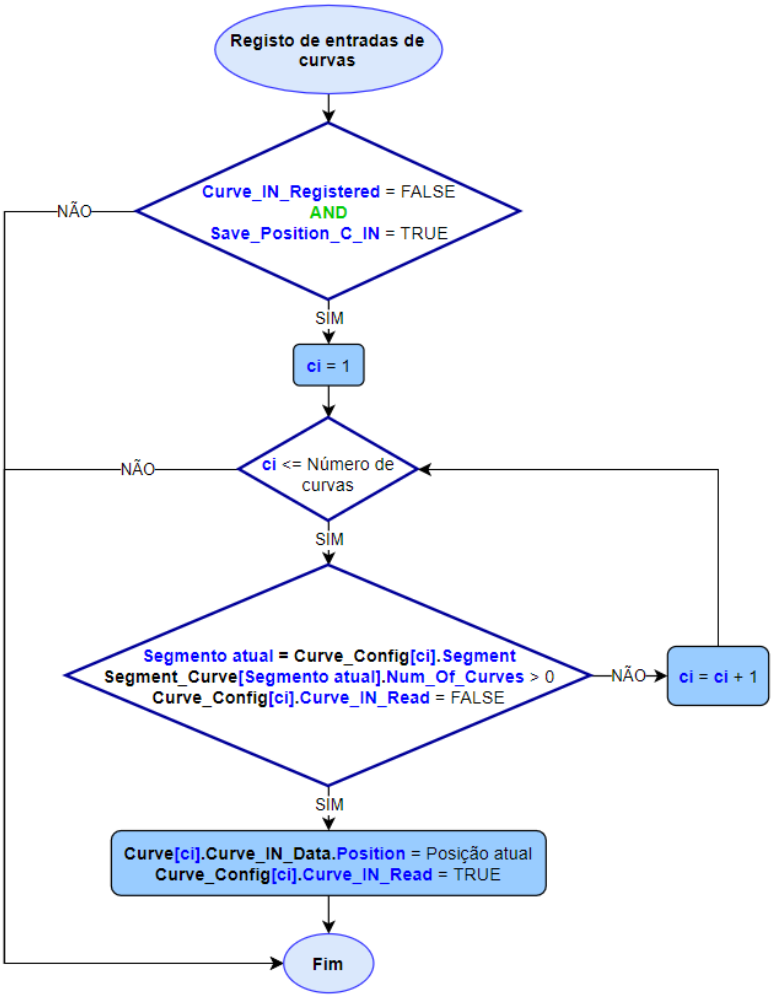


Figura 96 – Registo das posições de entrada das curvas

No caso das posições de entrada das curvas, Figura 96, depois de ser recebida a validação, é guardada a posição da curva pretendida ao confirmarem-se as seguintes condições:

- O segmento atual corresponde a um segmento presente no vetor Curve_Config;
  - Descobre a curva que se pretende registar através da correspondência do segmento configurado com o atual.
- O número de curvas do segmento atual é maior do que 0;
  - Verifica se ainda existem curvas por ler no segmento.
- A curva detetada ainda não foi registada.
  - Verifica o estado da variável Curve_IN_Read, presente no vetor Curve_Config.

Após o registo da posição, é ativada a *flag* Curve_IN_Read, o que previne que a entrada da curva seja registada novamente. A posição é registada no vetor Curve, no número da curva identificada.

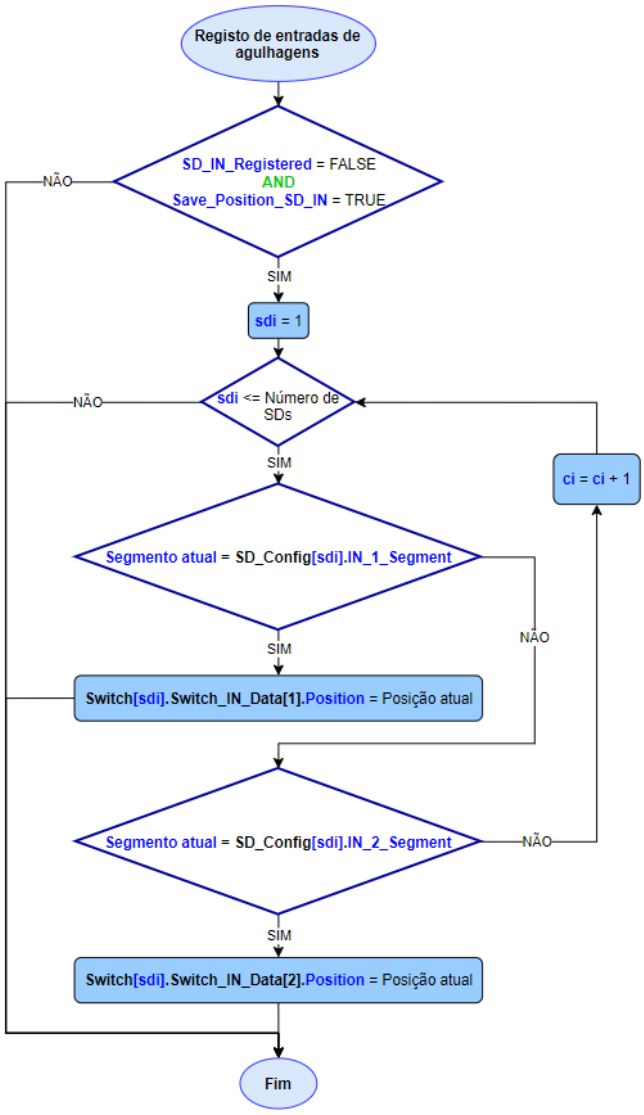


Figura 97 – Registo das posições de entrada das agulhagens

Tal como no ciclo mencionado anteriormente, o registo das posições de entrada das agulhagens, correspondente à Figura 97, também é realizado após receber a validação. A principal diferença relativamente ao registo das curvas é a verificação dos segmentos das múltiplas entradas. Neste caso, a única condição para registar a posição é a correspondência do segmento atual com o segmento pré-configurado das agulhagens, uma vez que, ao contrário das curvas, apenas existe uma agulhagem por segmento.

Adicionalmente ao registo das posições, esta função também calcula a distância de cada curva, a distância entre diferentes entradas e saídas de curvas consecutivas.

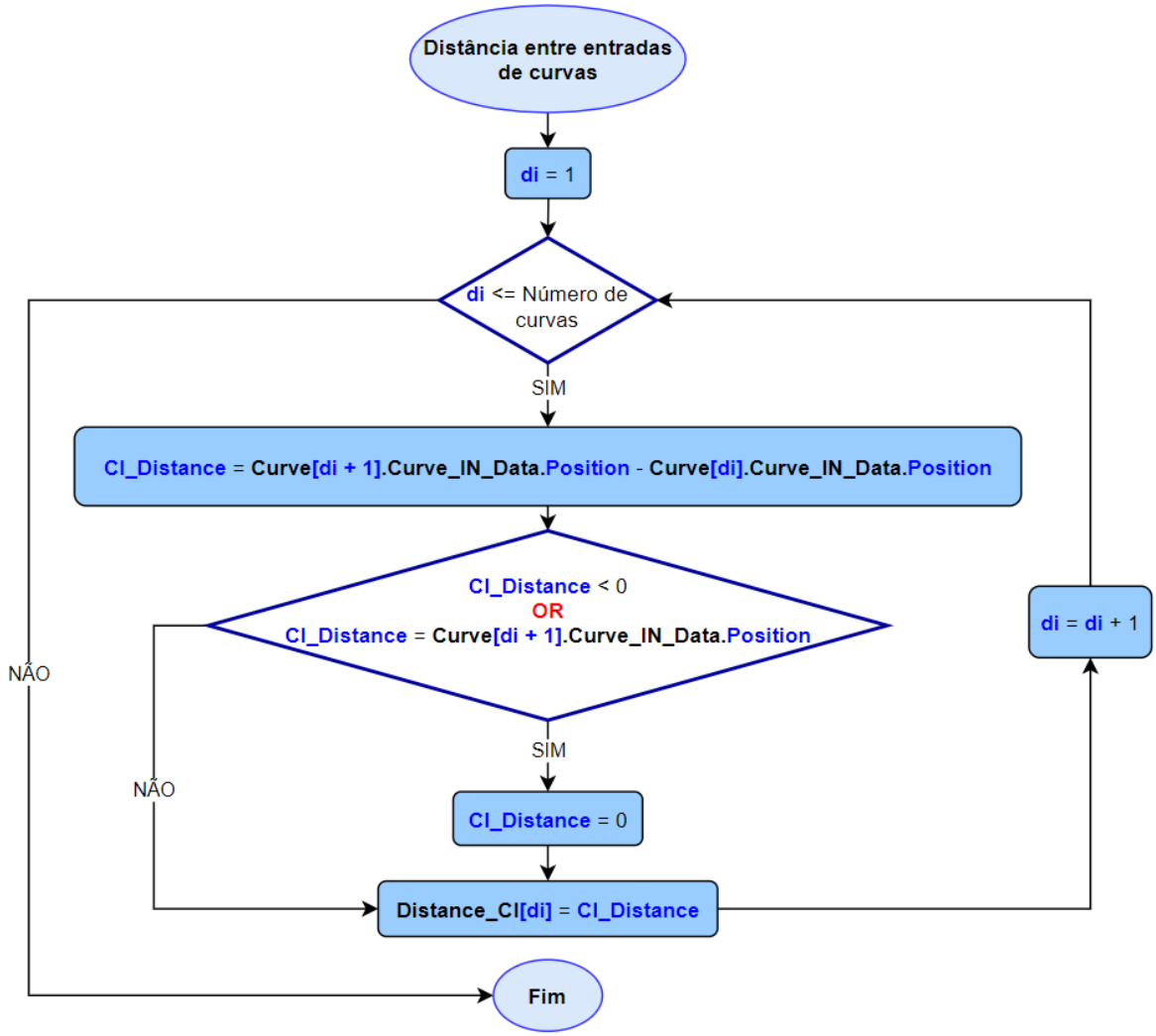


Figura 98 – Cálculo da distância entre entradas de curvas consecutivas

O fluxograma da Figura 98 representa o funcionamento do cálculo das distâncias entre entradas consecutivas. Um ciclo FOR percorre o vetor Curve que armazenas as posições das curvas, subtraindo a entrada de uma curva pela anterior. O valor calculado da distância é guardado no vetor Distance_CI, presente na Figura 99, apenas se o valor for positivo.

8		Distance_CI	Array[1..23] of DInt		☐	☑	☑	☑	☐	Distance between CIs
9		Distance_CI[1]	DInt	0	☐	☑	☑	☑	☐	CI[2] - CI[1]
10		Distance_CI[2]	DInt	0	☐	☑	☑	☑	☐	CI[3] - CI[2]
11		Distance_CI[3]	DInt	0	☐	☑	☑	☑	☐	CI[4] - CI[3]
12		Distance_CI[4]	DInt	0	☐	☑	☑	☑	☐	CI[5] - CI[4]
13		Distance_CI[5]	DInt	0	☐	☑	☑	☑	☐	CI[6] - CI[5]
14		Distance_CI[6]	DInt	0	☐	☑	☑	☑	☐	CI[7] - CI[6]
15		Distance_CI[7]	DInt	0	☐	☑	☑	☑	☐	CI[8] - CI[7]
16		Distance_CI[8]	DInt	0	☐	☑	☑	☑	☐	CI[9] - CI[8]
17		Distance_CI[9]	DInt	0	☐	☑	☑	☑	☐	CI[10] - CI[9]
18		Distance_CI[10]	DInt	0	☐	☑	☑	☑	☐	CI[11] - CI[10]
19		Distance_CI[11]	DInt	0	☐	☑	☑	☑	☐	CI[12] - CI[11]
20		Distance_CI[12]	DInt	0	☐	☑	☑	☑	☐	CI[13] - CI[12]
21		Distance_CI[13]	DInt	0	☐	☑	☑	☑	☐	CI[14] - CI[13]
22		Distance_CI[14]	DInt	0	☐	☑	☑	☑	☐	CI[15] - CI[14]
23		Distance_CI[15]	DInt	0	☐	☑	☑	☑	☐	CI[16] - CI[15]
24		Distance_CI[16]	DInt	0	☐	☑	☑	☑	☐	CI[17] - CI[16]
25		Distance_CI[17]	DInt	0	☐	☑	☑	☑	☐	CI[18] - CI[17]
26		Distance_CI[18]	DInt	0	☐	☑	☑	☑	☐	CI[19] - CI[18]
27		Distance_CI[19]	DInt	0	☐	☑	☑	☑	☐	CI[20] - CI[19]
28		Distance_CI[20]	DInt	0	☐	☑	☑	☑	☐	CI[21] - CI[20]
29		Distance_CI[21]	DInt	0	☐	☑	☑	☑	☐	CI[22] - CI[21]
30		Distance_CI[22]	DInt	0	☐	☑	☑	☑	☐	CI[23] - CI[22]
31		Distance_CI[23]	DInt	0	☐	☑	☑	☑	☐	CI[24] - CI[23]

Figura 99 – Vetor que armazena as distâncias entre entradas de curvas

O funcionamento do cálculo das saídas é semelhante ao da Figura 98, no entanto, no caso do cálculo da distância de cada curva, esquematizado na Figura 100, a principal diferença é a subtração da posição de saída com a posição de entrada de cada curva.

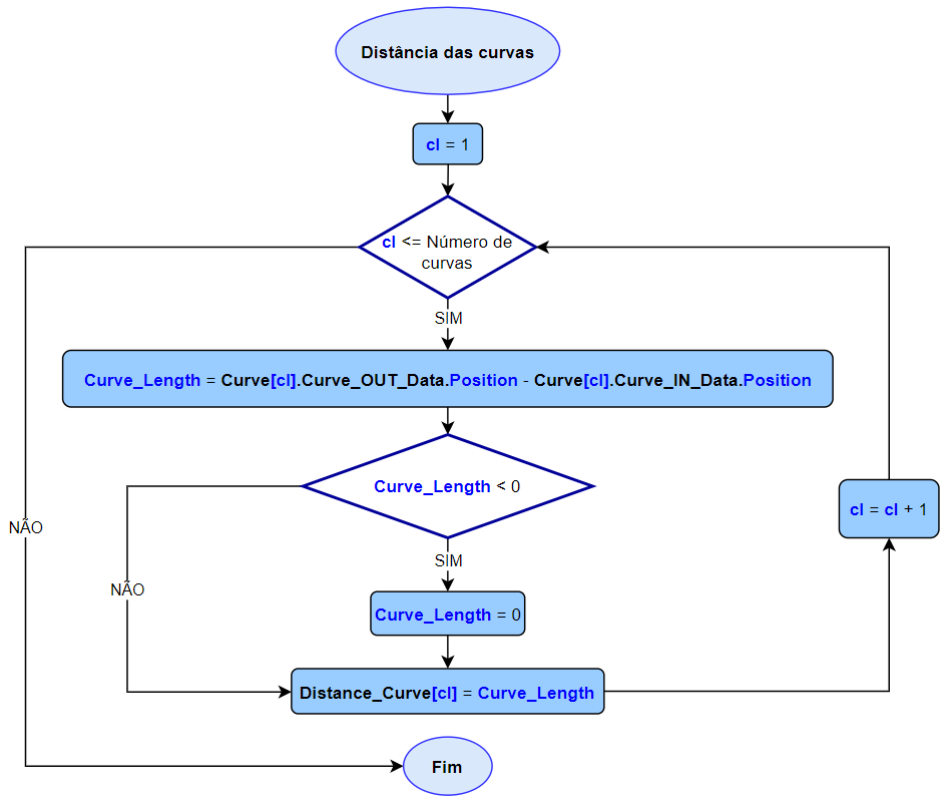


Figura 100 – Cálculo da distância de cada curva

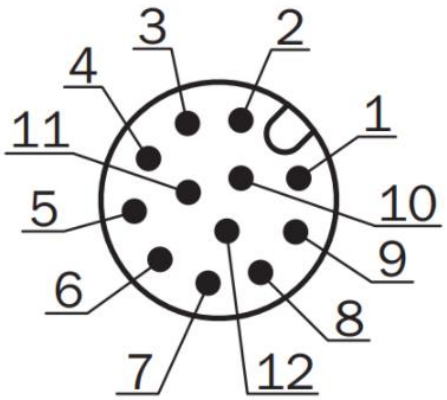
5.4. DETEÇÃO DE RGVS

Uma das soluções a implementar é a deteção de RGVs. Devido à existência de várias vantagens, analisadas no capítulo anterior, é escolhida a hipótese da utilização de sensores LiDAR. O sensor LiDAR selecionado é o TIM351 da SICK.

5.4.1. SENSOR LIDAR

O TIM351 é um sensor de distância LiDAR que permite verificar as coordenadas 2D de um campo de deteção. Este sensor possui uma boa precisão de medição e imunidade à luz ambiente.

Ele pode armazenar 16 configurações diferentes. Cada configuração possui até três campos de deteção. A configuração é selecionada através de quatro entradas. Na Figura 101, além das quatro entradas é possível verificar o *pinout* do sensor.



① GND
② DC 9 V ... 28 V
③ In_1
④ In_2
⑤ OUT1
⑥ OUT2
⑦ OUT3
⑧ OUT4
⑨ PNP: INGND, NPN: IN 9 V ... 28 V
⑩ In3
⑪ In4
⑫ nc

Figura 101 – *Pinout* do TIM351

Neste projeto só são utilizadas as duas primeiras configurações, logo apenas é necessária a utilização de uma entrada, a entrada In1. As restantes entradas são ligadas à massa. Se In1 estiver desativada, a primeira configuração é utilizada, senão utiliza-se a segunda.

As primeiras três saídas do sensor correspondem à resposta de cada campo de deteção. Por exemplo, no caso do primeiro campo detetar objetos é ativada a saída OUT1. A saída OUT4 é usada para avisar se o sensor se encontra em erro.

O modelo do sensor escolhido é o TIM351-2134001. Este modelo é uma versão PNP, ou seja, as saídas possuem 24 V quando estão ativas. Como é uma versão PNP é necessário colocar o pino 9 na massa.

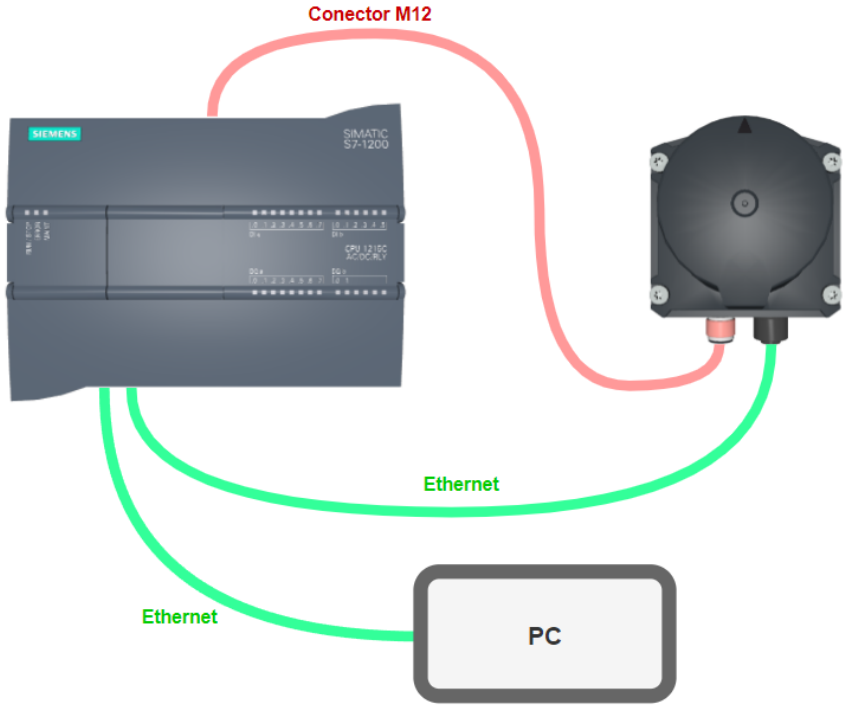


Figura 102 – Esquema de ligações do TIM351

As entradas, as saídas e a alimentação do sensor são feitas através do conector M12. O sensor é configurado pelo PC, através do *software* SOPAS da SICK. Uma vez que o PC apenas possui uma entrada *Ethernet* e esta já se encontra utilizada pelo PLC, pode-se ligar diretamente o cabo *Ethernet* à outra porta do PLC.

Através do SOPAS é possível definir os campos de deteção. Neste projeto, o sensor é instalado na fronteira do veículo, que tem cerca de 2,8 m de largura. Visto que o sensor é colocado a meio, os campos configurados estão de acordo com a Figura 103.

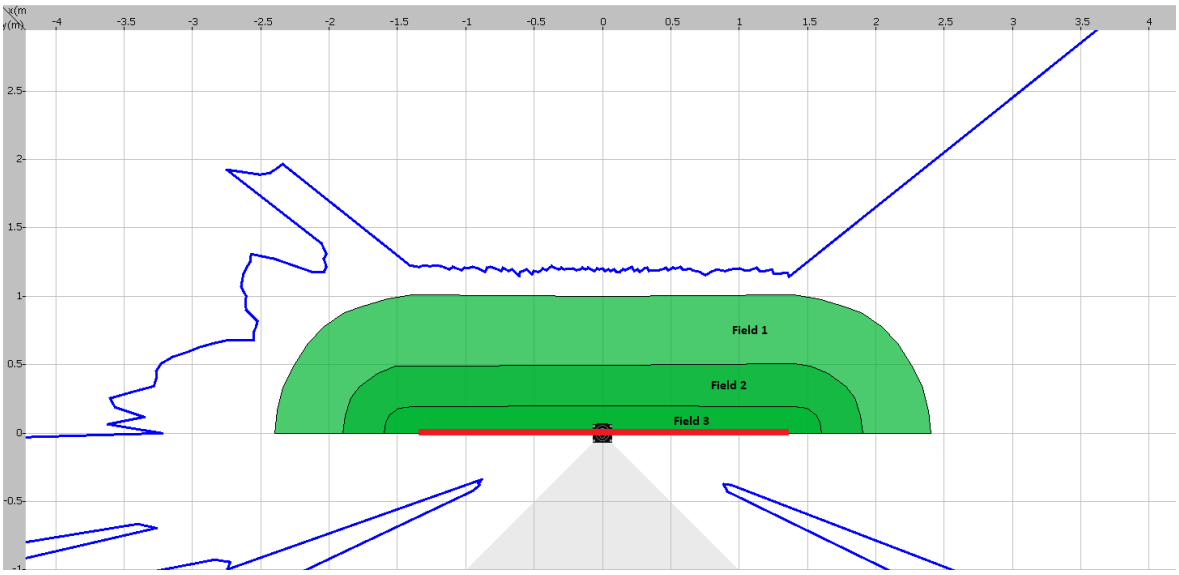


Figura 103 – Configuração dos campos do TIM351 para a deteção de RGVs

O primeiro campo, Field 1, deteta os objetos que estão a 1 m do RGV. O segundo campo, Field 2, está definido a 0,5 m do RGV. Já o terceiro campo, Field 3, está configurado para detetar os RGVs que se encontram a 0,2 m do veículo, parando o veículo e registando a posição.

5.4.2. COMANDOS DO *MASTER-CONTROLLER*

Tal como já foi referido na Figura 25, um sistema de RGVs possui uma topologia de comunicação *Master-Slave*. O *Master Controller* controla as operações de cada RGV através do envio de comandos. Quando um RGV interrompe o movimento, o *Master* envia um comando *Reset*. Este comando reinicia o RGV, colocando-o em modo *ready*. Após o RGV ser reiniciado, o *Master* envia-lhe uma nova posição através do comando *Positioning*.

Toda esta troca de comandos do *Master* para o RGV é efetuada através de funções já existentes no programa. Estas funções encontram-se na linguagem de lista de instruções, sendo necessário convertê-las para LADDER e texto estruturado.

Função FB_MC_COM_RESET

Esta função permite o envio do comando RESET por parte do *Master Controller*. Quando um veículo se encontra parado ele entra em erro, sendo necessário reiniciá-lo. Como já foi mencionado, o RGV está constantemente a enviar informações para o *Master*, como a sua posição atual, o seu modo de operação e a sua disponibilidade.

O esquema da Figura 104 representa os processos realizados no comando RESET, em ambos os PLCs.

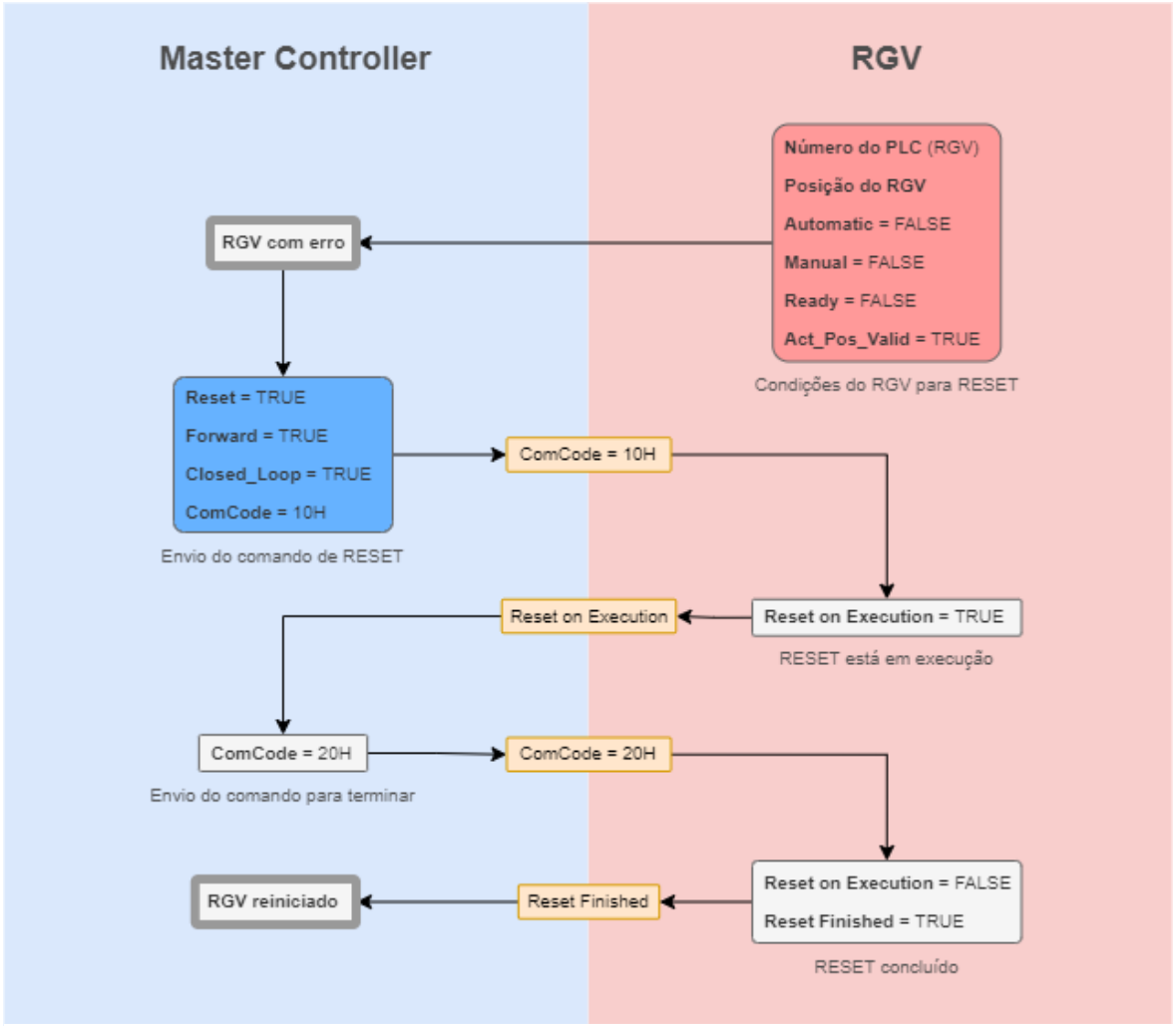


Figura 104 – Comando RESET

O *Master* só envia um comando de RESET conforme a receção de um conjunto de parâmetros provenientes do RGV. Estes parâmetros definem que o RGV se encontra em erro. Após serem validadas as informações recebidas, o *Master* envia o comando de RESET.

De seguida, o RGV envia uma resposta a confirmar que o veículo se encontra em processo de RESET, entretanto, o *Master* espera pela finalização deste processo. Quando o comando estiver concluído, isto é, quando o veículo for reiniciado, o RGV envia uma mensagem a confirmar a conclusão do comando.

A função FB_MC_COM_RESET é concebida através desta troca de informações. O fluxograma presente no Anexo D representa o funcionamento desta função.

Esta função possui como entradas o número do RGV, o sentido de deslocação e o código da resposta do RGV.

Os processos de envio e receção de informações são definidos no programa como passos, ou *steps*. O *step* de cada RGV encontra-se na base de dados "DB_STEP" e indica a parte de um certo comando que o RGV se encontra a processar.

Para a realização do comando RESET, o veículo terá de estar entre o *step* 61 e o *step* 64. Após o *step* ser validado, limpa-se a "DB_SEND". Esta base de dados contém os parâmetros que são enviados para o RGV. Estando a base de dados limpa, procede-se para a ativação do comando RESET através da escrita dos parâmetros Reset, Front (sentido de deslocação) e Closed_Circuit (circuito fechado). Entre o *step* 61 e o *step* 64 são realizadas as transferências dos parâmetros. No final, o RGV é reiniciado e fica com o *step default* 5.

Função FB_MC_COM_POS

Esta função permite o envio do comando POSITIONING por parte do *Master Controller*. Após um veículo ser reiniciado, ele necessita de uma nova posição final. Através deste comando é possível atribuir novas posições a RGVs.

O esquema da Figura 105 representa os processos realizados no comando POSITIONING, em ambos os PLCs.

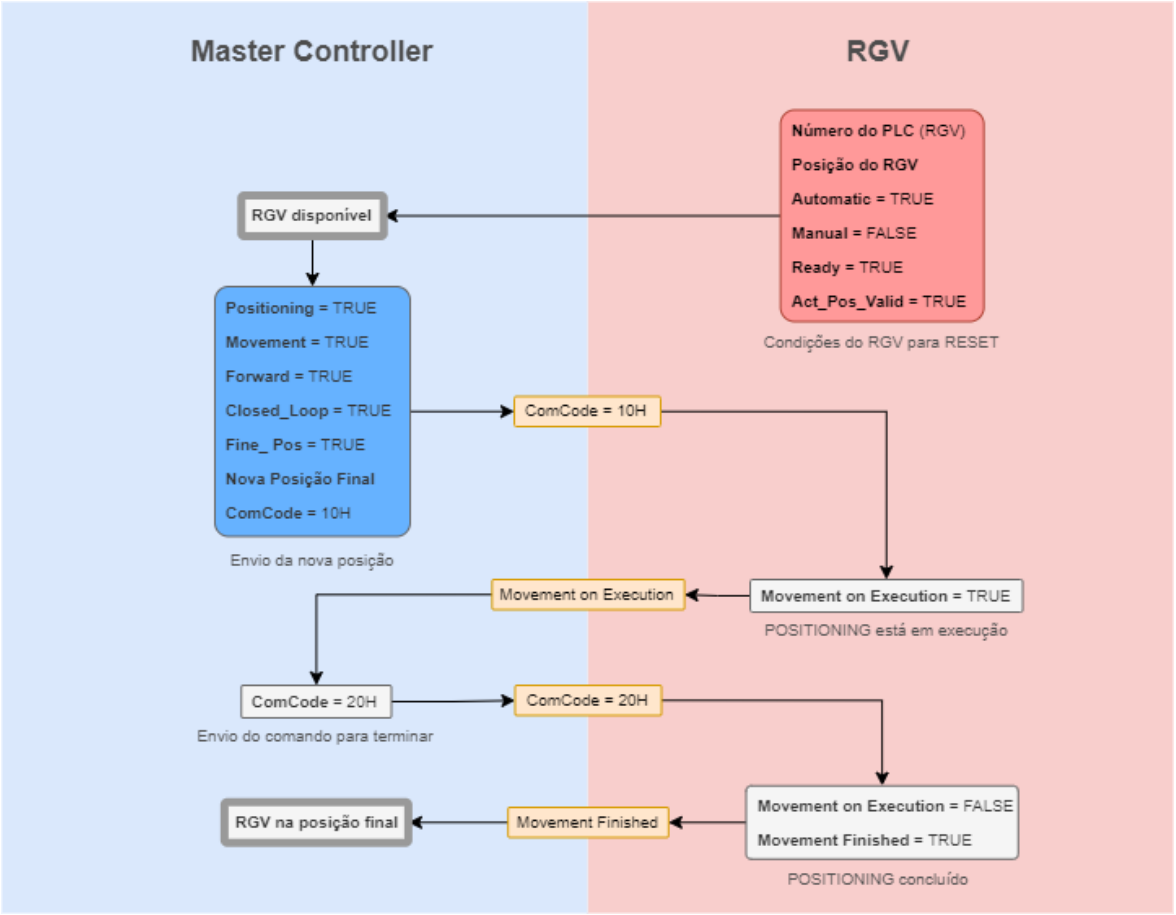


Figura 105 – Comando POSITIONING

Para o comando POSITIONING ser enviado pelo *Master* é necessária a confirmação de que o RGV está disponível, através da receção de um conjunto de parâmetros. Depois de verificar a disponibilidade do RGV, o *Master* envia os parâmetros de ativação do processo POSITIONING, juntamente com a nova posição final, obtendo uma resposta de confirmação do início de movimento do RGV. Enquanto o RGV encontra-se em movimento, o *Master* aguarda pela paragem na posição. Quando o RGV chegar à posição final, ele envia uma mensagem a confirma a conclusão do comando.

A função FB_MC_COM_POS consiste na realização destes processos. O fluxograma presente no Anexo E representa o funcionamento desta função.

As entradas desta função consistem no número do RGV, a nova posição final, o sentido de deslocação, o código da resposta do RGV e o modo de deslocação. Para o comando POSITIONING ser realizado, o veículo terá de estar entre o *step* 21 e o *step* 24. A função começa pela validação do *step* do RGV, ocorrendo de seguida a limpeza da base de dados

"DB_SEND". Após esta limpeza, são enviados os parâmetros que ativam este comando. Dentro destes parâmetros constam a nova posição final, Positioning, Movement, Front, Fine_Pos e Closed_Circuit. As transferências das informações presentes neste comando são realizadas entre o *step* 21 e o *step* 24. O comando termina quando o RGV não estiver em movimento, ou seja, quando chegar à posição final. No final, o RGV permanece com o *step default* 5.

Função FB_MC_RGV_Detection

Uma vez desenvolvidas e analisadas as funções dos envios dos comandos, é necessário criar uma função que envolva todo o processo de deteção e registo de posições, da parte do *Master Controller*.

O processo baseia-se na divisão de uma curva em dez pontos. São colocados dois RGVs na curva. O RGV da frente e o RGV de trás são enviados para o primeiro ponto. Como o RGV da frente é o primeiro a parar, o RGV de trás irá detetar a sua presença, parando também e registando a posição atual. Este processo repete-se até completarem-se as dez posições da curva. Pretende-se com isto registar as posições numa curva onde a distância entre os RGVs é mínima, de forma a evitar possíveis colisões.

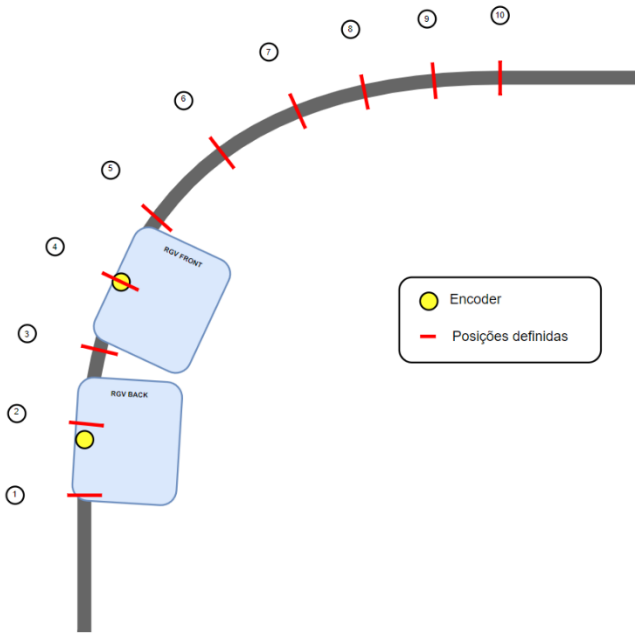


Figura 106 – Esquema da deteção de RGVs

O controlo do envio das posições e dos *resets* dos veículos é feito através das duas funções analisadas anteriormente. A função FB_MC_RGV_Detection baseia-se neste controlo, utilizando uma sequência de envio de comandos. Esta sequência está representada na Figura 51. A sequência possui quatro envios de comandos pelo *Master*. Estes quatro envios foram designados de *Macro Steps*, que são os passos que influenciam esta função.

No Anexo C encontra-se o fluxograma do funcionamento desta função. A função começa pela validação do número de posições de que pretende-se ler, como está representado na Figura 107.

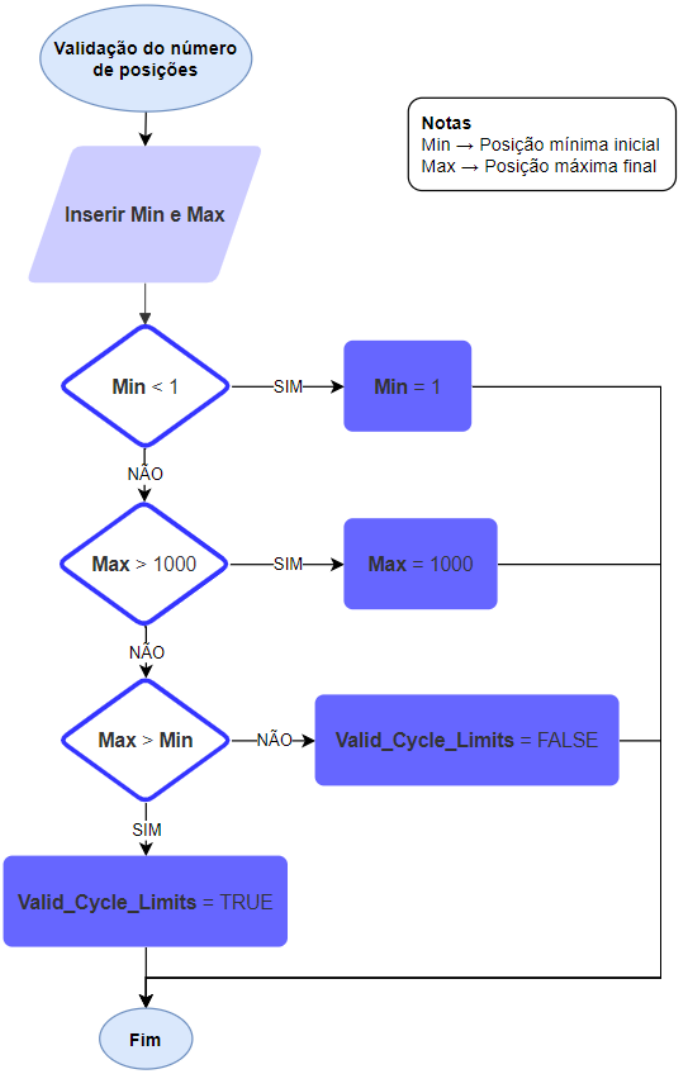


Figura 107 – Validação do número de posições da curva

O número de posições deverá estar entre 1 e 10. A variável Min é o número da primeira posição que se pretende enviar, sendo Max a última posição.

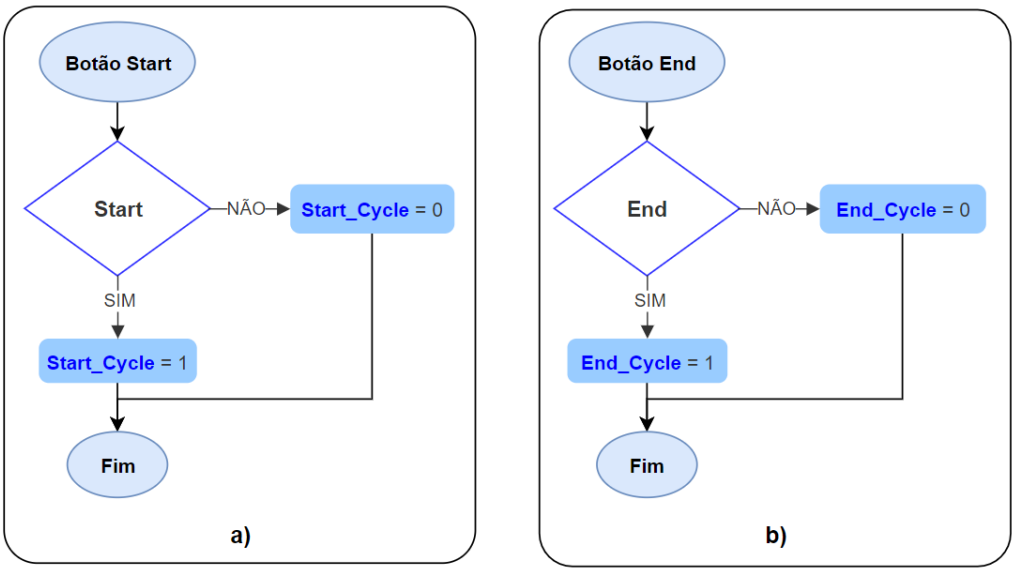


Figura 108 – a) Botão de início
b) Botão de fim

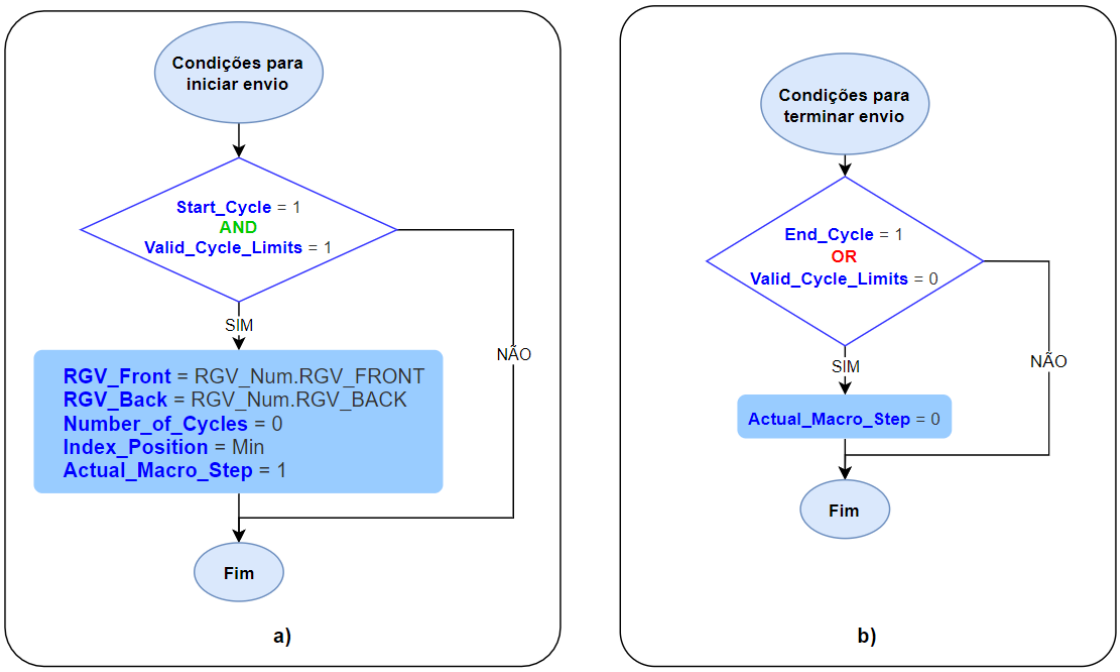


Figura 109 – a) Condições para iniciar o envio de comandos
b) Condições para terminar o envio de comandos

Na Figura 109 é possível verificar as condições para iniciar ou finalizar o processo. Se as posições Min e Max forem válidas e se for pressionado o botão Start, presente na Figura 108, são inicializadas as informações relativas ao processo, como os números do RGVs que são utilizados, o número de ciclo realizados, a primeira posição a registar e o *Macro Step* atual. Estas informações são armazenadas na DB_MC_DATA_CONFIG.

DB_MC_DATA_CONFIG

	Name	Data type
1	Static	
2	RGV_Num	Struct
3	RGV_Front	Int
4	RGV_Back	Int
5	Curve_Positions	Array[1..10] of DInt
6	Curve_Positions[1]	DInt
7	Curve_Positions[2]	DInt
8	Curve_Positions[3]	DInt
9	Curve_Positions[4]	DInt
10	Curve_Positions[5]	DInt
11	Curve_Positions[6]	DInt
12	Curve_Positions[7]	DInt
13	Curve_Positions[8]	DInt
14	Curve_Positions[9]	DInt
15	Curve_Positions[10]	DInt
16	Min	Int
17	Max	Int
18	Index_Position	Int
19	Actual_Macro_Step	Int
20	Previous_Macro_Step	Int
21	Number_of_Cycles	Int
22	Start	Bool
23	End	Bool

Figura 110 – DB_MC_DATA_CONFIG

Se as posições Min e Max não forem válidas ou se for pressionado o botão End, a função é terminada.

Após o processo ser inicializado, é enviado o comando de RESET ao veículo da frente. No final deste comando, é verificada a possibilidade do envio do comando POSITIONING através das condições recebidas do RGV.

Verificadas as condições, o *Macro Step* fica a 2, ou seja, o processo encontra-se na etapa de posicionamento do RGV da frente. Nesta etapa, a nova posição final a enviar encontra-se no vetor Curve_Positions, presente na base de dados DB_MC_DATA_CONFIG. Este vetor inclui as posições dos pontos pré-configurados da curva. Esta etapa termina quando o veículo da frente chegar à nova posição final e forem confirmadas as condições do RESET do RGV de trás, alterando assim o *Macro Step* para 3.

No *Macro Step* 3, o processo de envio do comando RESET ao segundo RGV é semelhante ao envio do veículo da frente. Esta etapa termina com a confirmação dos parâmetros de reinício do segundo RGV.

Na última etapa, *Macro Step* 4, é realizado o posicionamento do RGV de trás. A nova posição final do segundo RGV é a mesma que foi anteriormente enviada para o RGV da frente. No final desta etapa, são verificadas as condições do primeiro RGV para o envio do comando RESET e é incrementado o índice da posição. Se o índice de posição for maior do que a posição Max, o processo do registo das posições encontra-se concluído. Se ainda não ultrapassou este índice, será realizado mais um ciclo.

5.4.3. ROTINA PARA A DETEÇÃO DE RGVS

Esta rotina tem como objetivo o controlo da velocidade do segundo RGV conforme a distância entre os RGVs. No caso do RGV estiver a menos de 0,2 m, a posição é registada.

O sensor possui quatro saídas: três que indicam a deteção do campo e uma para alerta de erros. Estas saídas funcionam como entradas do PLC. O PLC encontra-se constantemente a ler estas entradas, como se pode ver pela Figura 111.

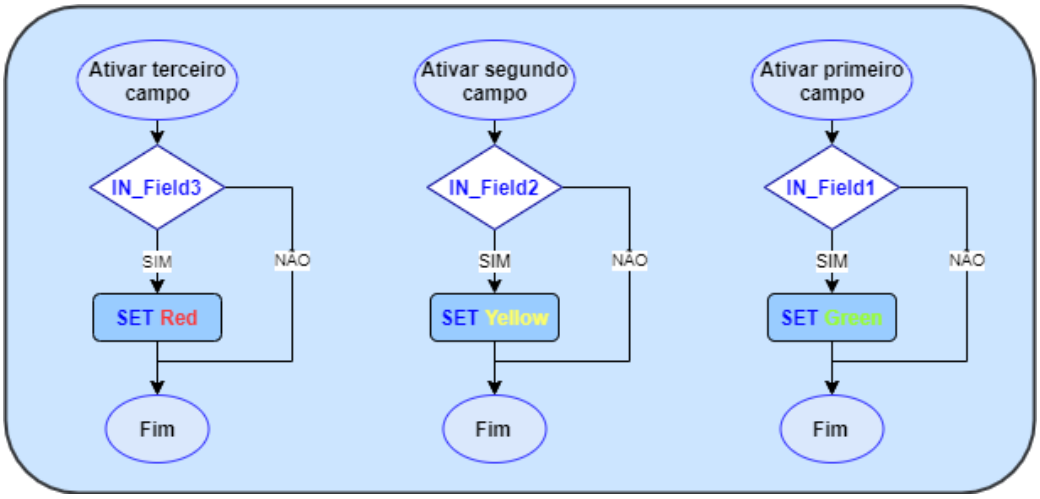


Figura 111 – Ativação dos campos de deteção

No caso de o sensor detetar o terceiro campo, isto é, se estiver a 1 m do veículo da frente, será ativada a variável Red. O mesmo acontece para os outros dois campos.

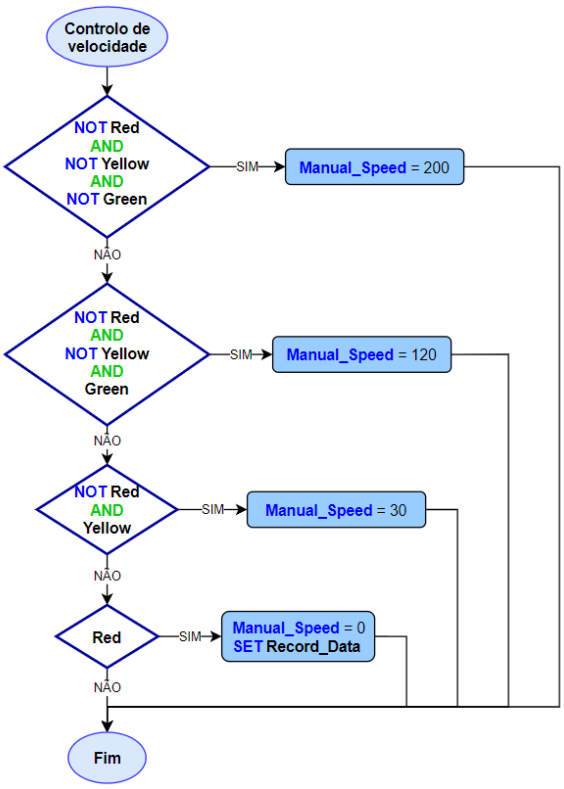


Figura 112 – Controlo da velocidade conforme o campo detetado

A velocidade do RGV muda conforme o campo detetado. No caso de não ter nenhum campo detetado o RGV possui uma velocidade de 200 mm/s. Se a variável Green estiver ativa, isto é, se o veículo da frente estiver a menos de 1 m do RGV de trás, a velocidade é reduzida para 120 mm/s. Se a distância entre os RGVs for menor do que 0,5 m é ativada a variável Yellow, que reduz a velocidade do RGV de trás para 30 mm/s. Por fim, quando esta distância chegar a 0,2 m (variável Red ativa) o segundo RGV para e ativa a variável Record_Data.

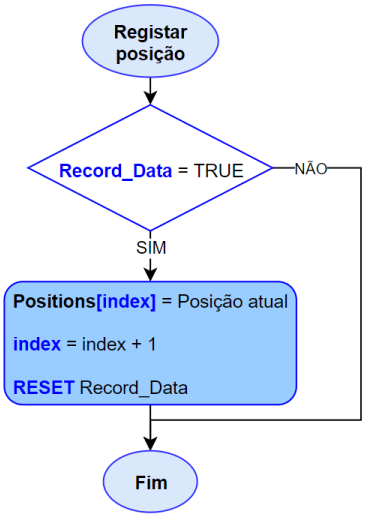


Figura 113 – Registo da posição de deteção do Field1

Como se pode ver pela Figura 113, quando a variável Record_Data é ativada, ocorre o registo da posição atual dentro do vetor Positions. De seguida, é incrementado o índice da posição a armazenar no próximo ciclo, sendo reiniciada a variável Record_Data.

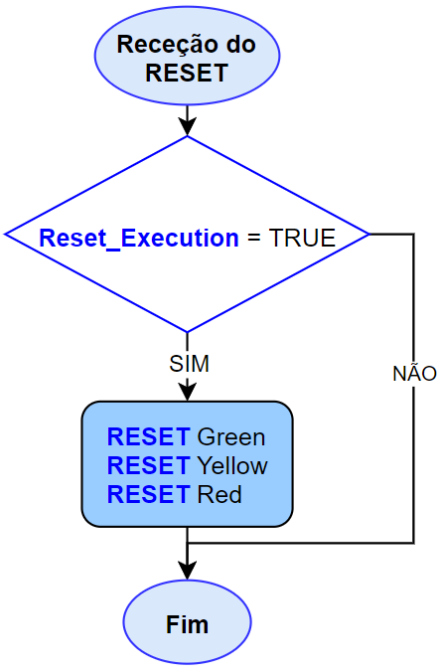


Figura 114 – Receção do comando RESET

Quando o RGV recebe o comando RESET do *Master*, o programa limpa as variáveis que indicam a deteção dos campos, permitindo assim iniciar outro ciclo.

5.5. ALINHAMENTO DOS TRANSPORTADORES

O terceiro objetivo principal deste projeto é o registo das posições de alinhamento entre o transportador do RGV e os transportadores do armazém. Devido às várias vantagens analisadas no capítulo anterior, decide-se implementar a hipótese da utilização de sensores LiDAR. Neste caso, é reutilizado o sensor TIM351.

Tal como já foi mencionado anteriormente, o TIM351 possui 16 configurações controladas pelas quatro entradas do sensor. Em cada configuração é possível definir três campos de deteção.

Pretende-se desta forma, criar uma rotina que permita detetar as pernas do transportador através da utilização destes campos, registando assim a posição onde se verifica o alinhamento.

Inicialmente, é necessário configurar os campos de deteção. Para a rotina implementada são utilizadas duas configurações, sendo apenas necessária uma entrada. As configurações, tal como já foi mencionado, podem ser estabelecidas através do *software* SOPAS.

Neste projeto, supõe-se que o RGV se desloca da esquerda para a direita, a uma velocidade de 200 mm/s, sendo os campos de deteção configurados desse modo.

Na primeira configuração, visível na Figura 115, o campo Field3 tem como objetivo a deteção da primeira perna do transportador. Ao detetar a primeira perna, o RGV reduz a sua velocidade para metade.

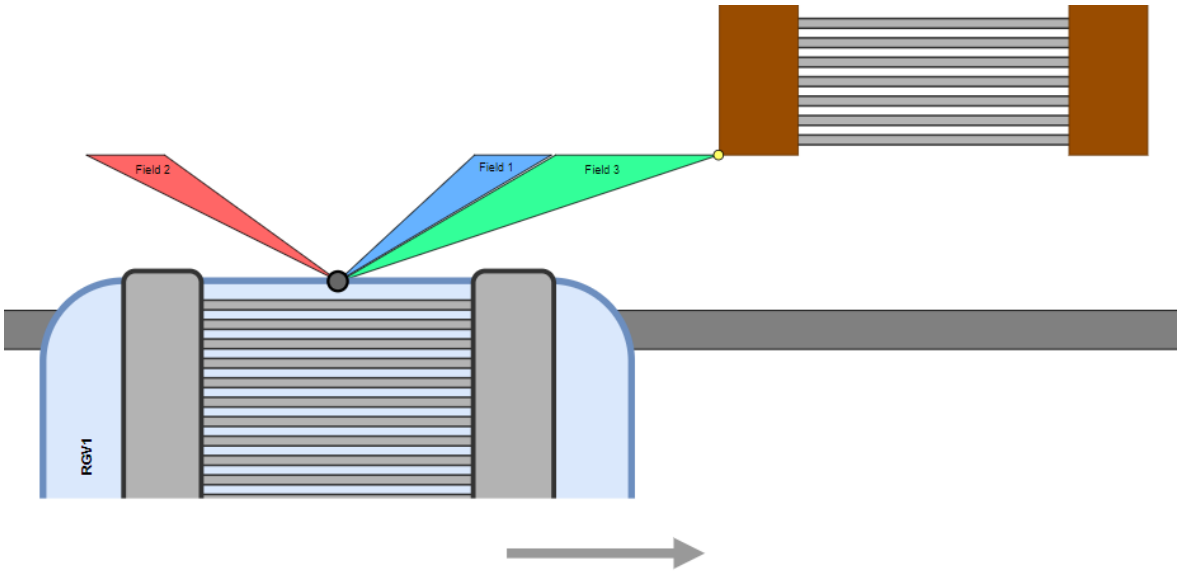


Figura 115 – Deteção da primeira perna do transportador pelo Field3

No programa, isto resume-se à leitura da entrada IN_Field3. Quando a primeira perna é detetada é ativada a variável First_Leg_Slow. Esta é a variável que controla a diminuição da velocidade do RGV em metade.

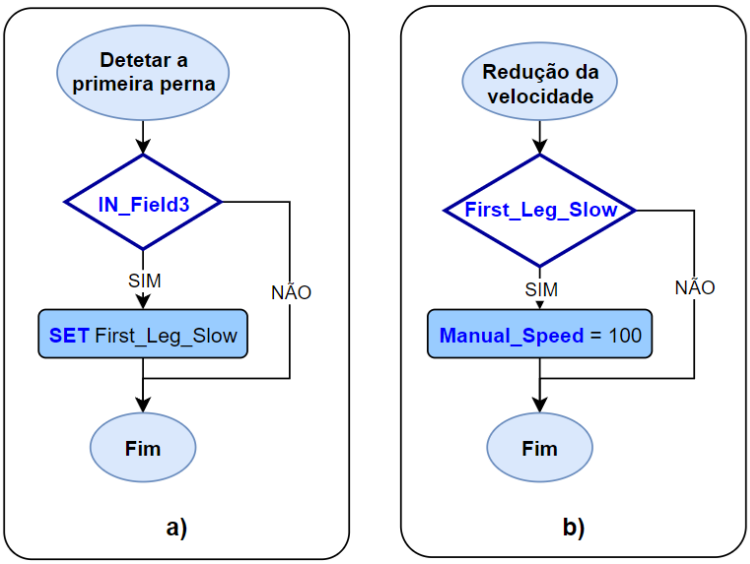


Figura 116 – a) Deteção da primeira perna do transportador
b) Ativação da variável First_Leg_Slow

De seguida, quando o segundo campo, Field2, detetar a primeira perna do transportador, como mostra a Figura 117, o veículo para durante dois segundos. No final desta paragem, o sensor altera para a segunda configuração.

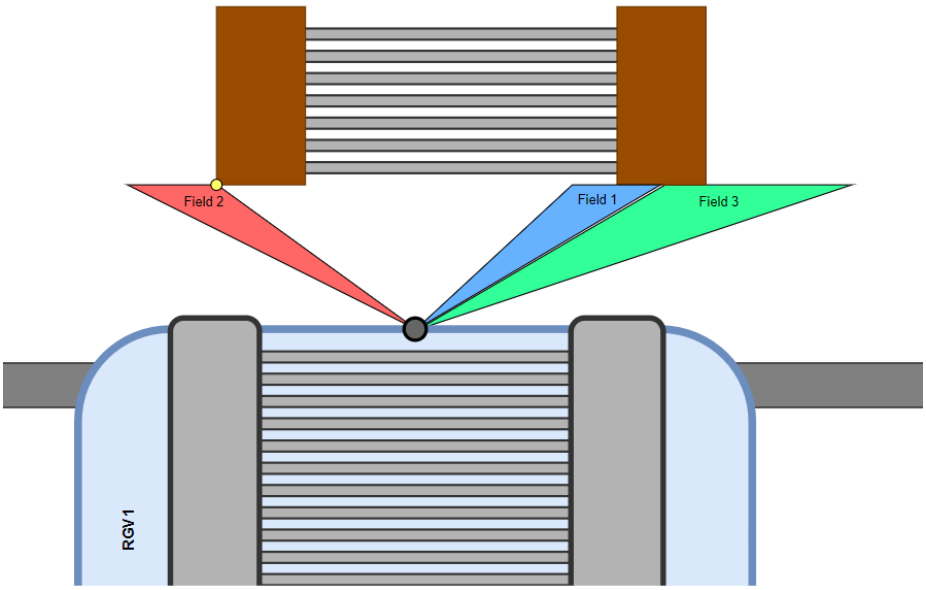


Figura 117 – Deteção da primeira perna do transportador pelo Field2

Na rotina, este processo consiste na verificação do estado da variável First_Leg_Slow e do segundo campo, visível na Figura 118. Se ambas estiverem ativas o RGV para durante dois segundos, isto é, utiliza-se um temporizador de dois segundos no qual é atribuída uma

velocidade 0. Após a conclusão do temporizador, é reiniciada a variável First_Leg_Slow, o RGV desloca-se a uma velocidade de 20 mm/s, e é ativada a saída OUT_Config. Esta saída está ligada à entrada In1 do sensor, que permite selecionar o número da sua configuração. Ao ativar a saída, a configuração passa a ser a segunda definida.

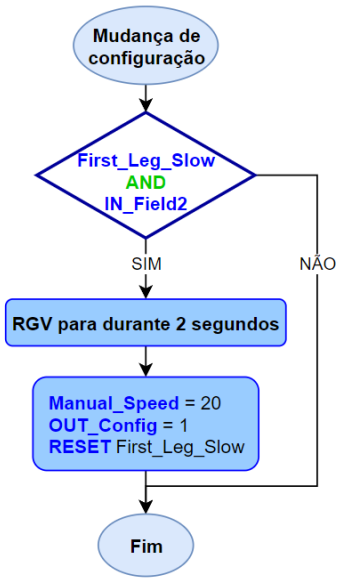


Figura 118 – Mudança de configuração do sensor

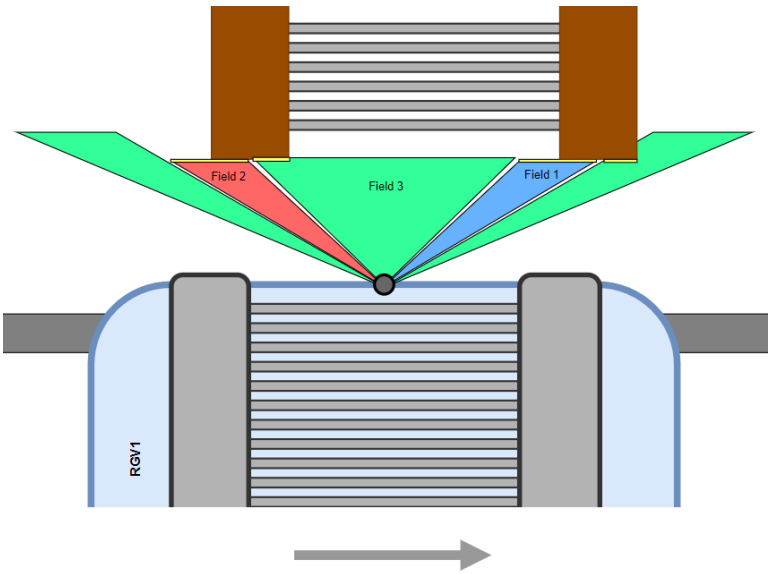


Figura 119 – Segunda configuração do sensor

A Figura 119 apresenta o esquema da segunda configuração do sensor após a mudança. Quando ocorre a mudança, os três campos passam a estar em deteção. Com o veículo a

deslocar-se a uma velocidade muito reduzida (20 mm/s), verifica-se o alinhamento quando o Field3 deixa de detetar, registando assim a posição.

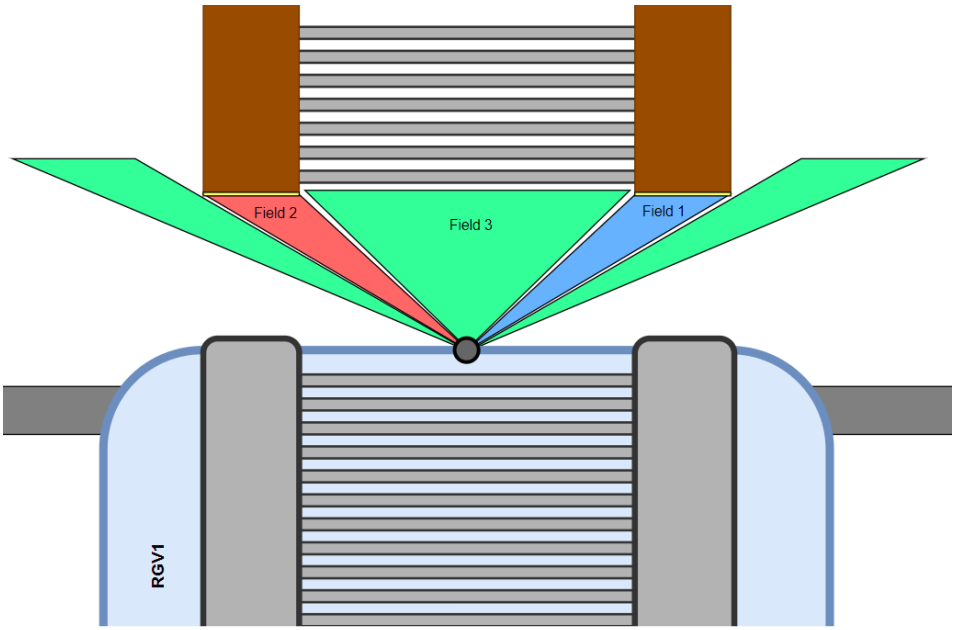


Figura 120 – Deteção da posição de alinhamento

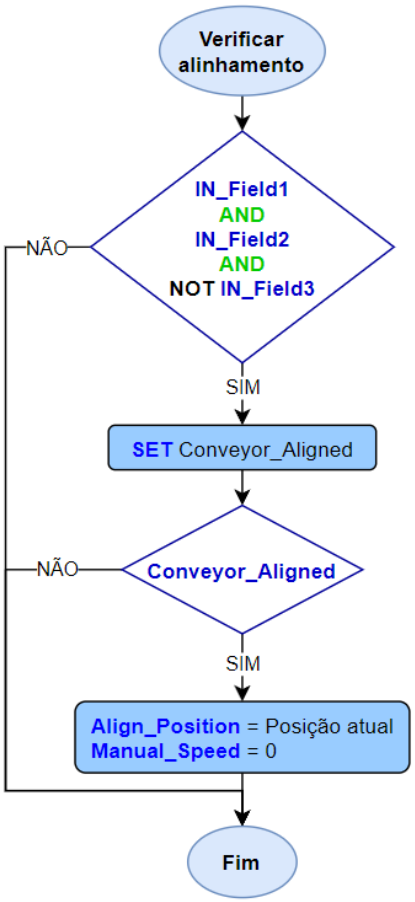


Figura 121 – Verificação do alinhamento e registo da posição

Na rotina, como se pode ver pela Figura 121, após ser verificado o alinhamento através da confirmação da condição referida acima, é ativada a variável Conveyor_Aligned. Se esta variável estiver ativa, o RGV para e é armazenada a posição atualmente lida pelo *encoder*.

6. TESTES E RESULTADOS

Este capítulo relata todos os testes efetuados, ao mesmo tempo que são analisados os resultados de cada um. Os testes têm como objetivo a verificação da melhor configuração possível dos sensores, assim como a análise do bom funcionamento das rotinas, de maneira a que seja possível corrigir eventuais problemas.

Devido à incapacidade de testar as rotinas criadas num circuito real com vários RGVs, os testes foram realizados com um RGV, numa secção de carril com cerca de 5 metros. Esta zona de testes encontra-se esquematizada na Figura 122.

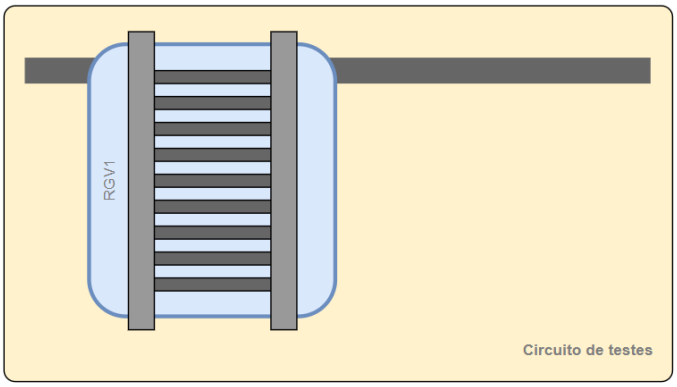


Figura 122 – Zona de testes

Uma vez que o troço de carril era bastante pequeno, implementaram-se alguns mecanismos de segurança no programa, como a definição de limites de posições e a operação com velocidades reduzidas. Além destes mecanismos, teve-se em conta a realização dos testes através de velocidades manuais, ou seja, o veículo muda a velocidade automaticamente, mas apenas se o botão do controlo remoto estiver a ser premido. Se este botão não estiver a ser premido o veículo permanece parado.

6.1. REGISTO DE POSIÇÕES DAS CURVAS E AGULHAGENS

Os testes realizados para a rotina do registo de posições das curvas e agulhagens têm dois objetivos: encontrar a melhor configuração possível do leitor RFID e verificar, através de simulações, o funcionamento da rotina.

6.1.1. CONFIGURAÇÃO DO LEITOR RFID

A leitura das *tags* contém várias variáveis que influenciam as posições obtidas. Existem múltiplas variáveis externas que não se conseguem controlar, como o ruído e as vibrações em redor do leitor, e outras variáveis, como o alcance de deteção do leitor e a velocidade do veículo. Para conseguir uma boa configuração do leitor é necessário realizar alguns testes com alcances de deteção e velocidades diferentes.

A velocidade do RGV é alterada através do programa. Já o alcance de deteção, como já foi mencionado, pode ser controlado através da potência de transmissão do leitor. Quanto maior for a potência de transmissão maior será o alcance de deteção. Para verificar a melhor potência de transmissão a utilizar, desenvolveu-se uma rotina de teste.

O leitor RFID é instalado debaixo do motor de translação, a uma altura de 20 cm, perto da roda da frente que está apoiada no carril, enquanto a *tag* é colocada no chão. Esta instalação encontra-se na Figura 123. O RGV desloca-se de uma posição inicial até a uma posição final, estando sempre com o leitor em modo de leitura contínua. Enquanto o

veículo se encontra em movimento, o leitor regista a posição onde o leitor começa a detetar a *tag*, e a posição onde este deixa de detetar.

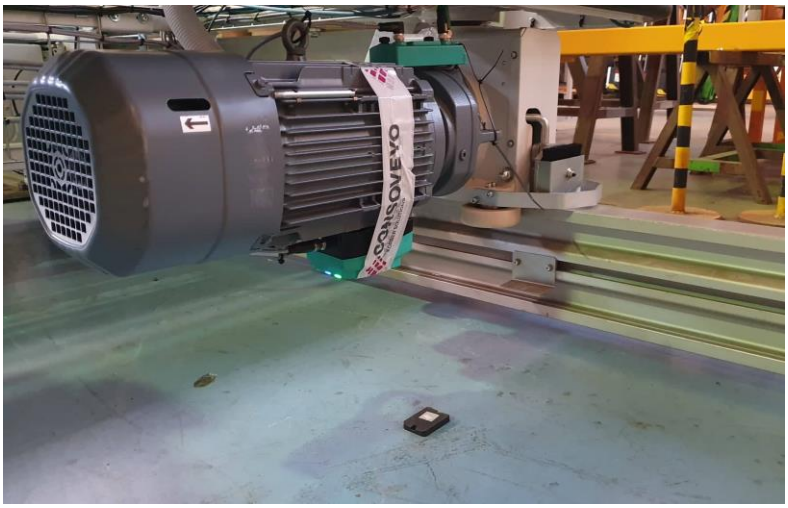


Figura 123 – Instalação do leitor RFID no RGV

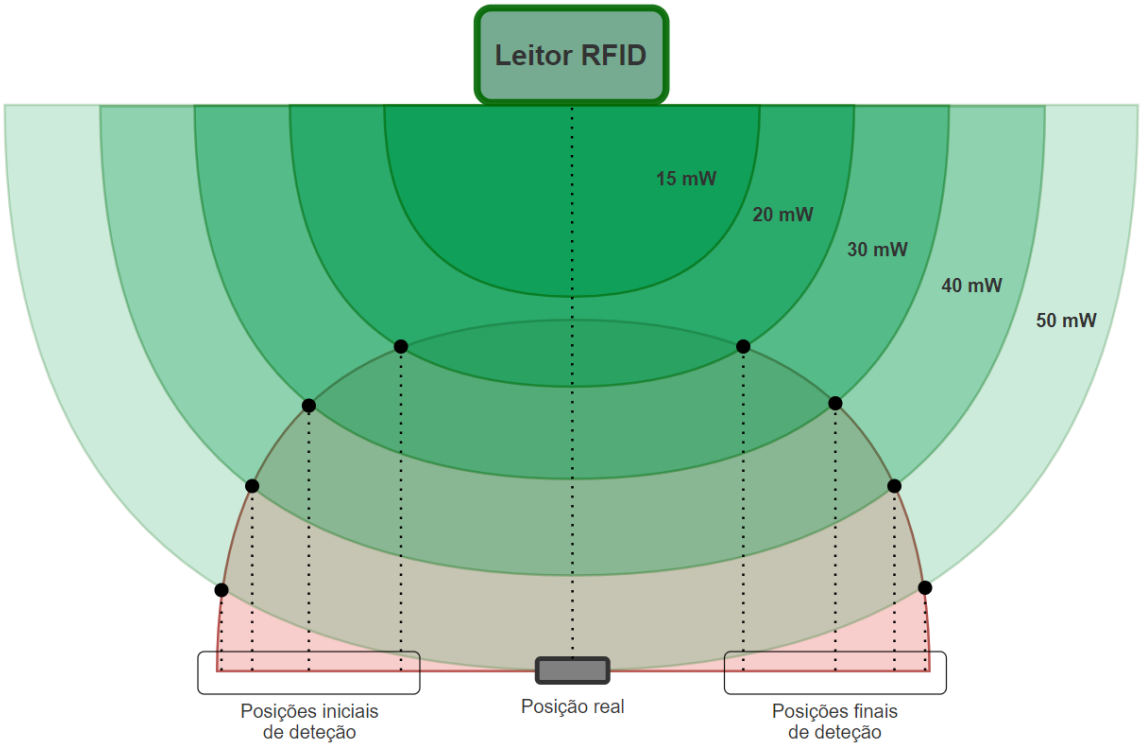


Figura 124 – Esquema do funcionamento dos campos de deteção RFID

O esquema da Figura 124 consolida a análise do funcionamento da potência de transmissão e das posições que se pretende registar. Quanto maior for a potência de transmissão do leitor maior será o campo de deteção, isto leva ao registo de maiores desvios da posição real. A posição real registada é a posição ,obtida através do *encoder*, quando o leitor RFID encontra-se alinhado com a *tag*. Neste caso, a posição real é 110463.

Para validar esta hipótese realizaram-se quatro ensaios, cada um com uma potência de transmissão diferente. Em cada ensaio foram realizadas quatro tentativas para três velocidades (100, 200 e 300 mm/s).

As tabelas abaixo referem-se ao resultados obtidos em cada ensaio.

Tabela 9 – Valores obtidos para 50 mW

PT = 50 mW	Posição Real = 110463

Velocidade (mm/s)	Tentativa	Posição Inicial de Deteção (mm)	Desvio Inicial (mm)	Desvio Médio Inicial (mm)	Posição Final de Deteção (mm)	Desvio Final (mm)	Desvio Médio Final (mm)
100	1	110339	124	121	110575	-112	-113
	2	110340	123		110578	-115	
	3	110348	115		110575	-112	
	4	110341	122		110576	-113	
200	1	110353	110	105,25	110594	-131	-122,25
	2	110375	88		110513	-50	
	3	110357	106		110618	-155	
	4	110346	117		110616	-153	
300	1	110345	118	121,75	110664	-201	-177
	2	110337	126		110640	-177	
	3	110335	128		110625	-162	
	4	110348	115		110631	-168	

Tabela 10 – Valores obtidos para 40 mW

PT = 40 mW	Posição Real = 110463

Velocidade (mm/s)	Tentativa	Posição Inicial de Deteção (mm)	Desvio Inicial (mm)	Desvio Médio Inicial (mm)	Posição Final de Deteção (mm)	Desvio Final (mm)	Desvio Médio Final (mm)
100	1	110369	94	96,75	110566	-103	-102,5
	2	110366	97		110565	-102	
	3	110370	93		110569	-106	
	4	110360	103		110562	-99	
200	1	110367	96	91,75	110601	-138	-138,75
	2	110349	114		110605	-142	
	3	110380	83		110612	-149	
	4	110389	74		110589	-126	
300	1	110381	82	95,25	110629	-166	-163
	2	110386	77		110631	-168	
	3	110357	106		110647	-184	
	4	110347	116		110597	-134	

Tabela 11 – Valores obtidos para 30 mW

PT = 30 mW	Posição Real = 110463

Velocidade (mm/s)	Tentativa	Posição Inicial de Deteção (mm)	Desvio Inicial (mm)	Desvio Médio Inicial (mm)	Posição Final de Deteção (mm)	Desvio Final (mm)	Desvio Médio Final (mm)
100	1	110391	72	79,25	110542	-79	-85,75
	2	110374	89		110558	-95	
	3	110381	82		110547	-84	
	4	110389	74		110548	-85	
200	1	110390	73	64	110574	-111	-117,25
	2	110398	65		110585	-122	
	3	110406	57		110577	-114	
	4	110402	61		110585	-122	
300	1	110415	48	51,5	110600	-137	-151,25
	2	110399	64		110629	-166	
	3	110420	43		110605	-142	
	4	110412	51		110623	-160	

Tabela 12 – Valores obtidos para 20 mW

PT = 20 mW	Posição Real = 110463

Velocidade (mm/s)	Tentativa	Posição Inicial de Deteção (mm)	Desvio Inicial (mm)	Desvio Médio Inicial (mm)	Posição Final de Deteção (mm)	Desvio Final (mm)	Desvio Médio Final (mm)
100	1	110413	50	47	110521	-58	-49
	2	110420	43		110524	-61	
	3	110425	38		110497	-34	
	4	110406	57		110506	-43	
200	1	110405	58	23,25	110559	-96	-75
	2	110465	-2		110545	-82	
	3	110449	14		110528	-65	
	4	110440	23		110520	-57	
300	1	110442	21	19,5	110544	-81	-78,75
	2	110430	33		110559	-96	
	3	110464	-1		110544	-81	
	4	110438	25		110520	-57	

Através da análise dos dados obtidos, é possível retirar algumas conclusões.

A primeira conclusão é a influência da velocidade na leitura. Quanto maior for a velocidade menor será o desvio do início da deteção relativamente à posição real. Isto acontece, porque o leitor contém um tempo de resposta de deteção. Se o veículo se deslocar a grandes velocidades a *tag* será detetada mais tarde. Isto pode-se observar na Tabela 12, onde em alguns casos, a posição inicial de deteção ultrapassa a posição real da *tag*. Como o leitor RFID apresenta um atraso no tempo de reposta e o RGV desloca-se a grandes velocidades, a posição final de deteção sofre um aumento no desvio da posição real.

A segunda conclusão a retirar é a influência do alcance de deteção do leitor. Quanto menor for o alcance, mais preciso será o leitor na aquisição das posições, isto é, menor será o desvio da posição real. Os valores médios do desvio ao utilizar uma potência de transmissão de 20 mW são menores do que para uma potência de transmissão de 50 mW. Ao utilizar uma potência de transmissão de 15 mW verificou-se que o leitor RFID já não detetava a *tag,* ou seja, possuía um alcance de leitura insuficiente.

Conclui-se assim que, para esta instalação do leitor RFID, a melhor configuração seria a que tivesse uma potência de transmissão de 20 mW e uma velocidade do veículo de 100 mm/s, uma vez que esta configuração é a que apresenta resultados mais precisos e mais constantes.

6.1.2. SIMULAÇÃO DA ROTINA IMPLEMENTADA

Dada a indisponibilidade de testar o programa num circuito real, o correto funcionamento da rotina desenvolvida é verificado com recurso a simulações. Neste caso, são utilizadas duas *tags*. As *tags* possuem dois códigos diferentes, um para a posição de entrada e outro para a posição de saída.

Neste teste, o leitor RFID permanece parado e com o comando de leitura contínua ativado, sendo que as *tags* são detetadas manualmente, isto é, a *tag* é que vai ao encontro do leitor. As configurações das curvas e das agulhagens estão de acordo com o circuito analisado anteriormente, sendo que os segmentos se encontram nos Anexos A e B.

Simulação do registo das curvas

Para realizar a simulação do registo das curvas, inicialmente é necessário escrever os códigos "CI" e "CO" em ambas as *tags*. Após esta escrita, o leitor é colocado no modo de leitura contínua.

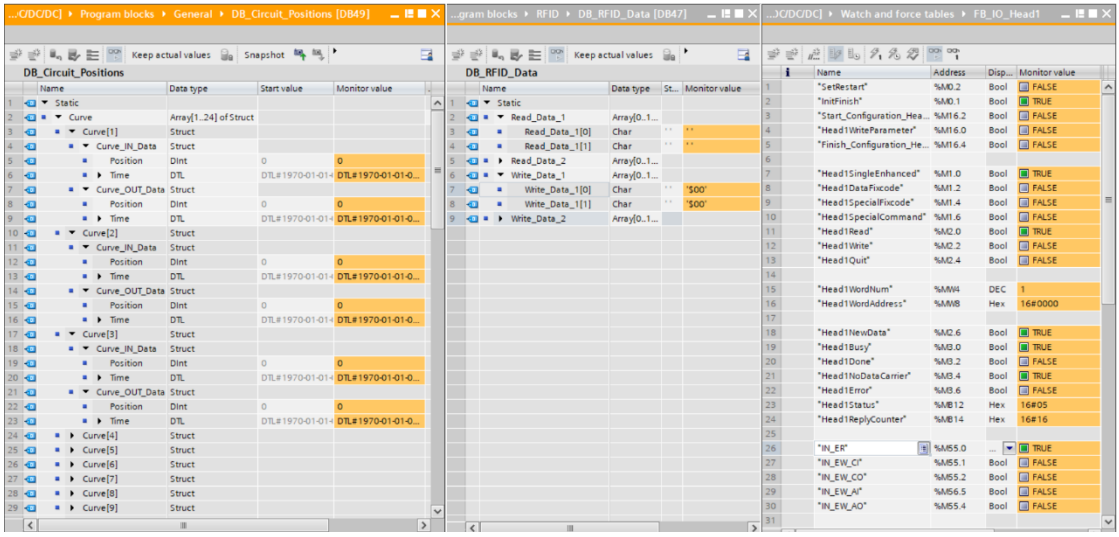
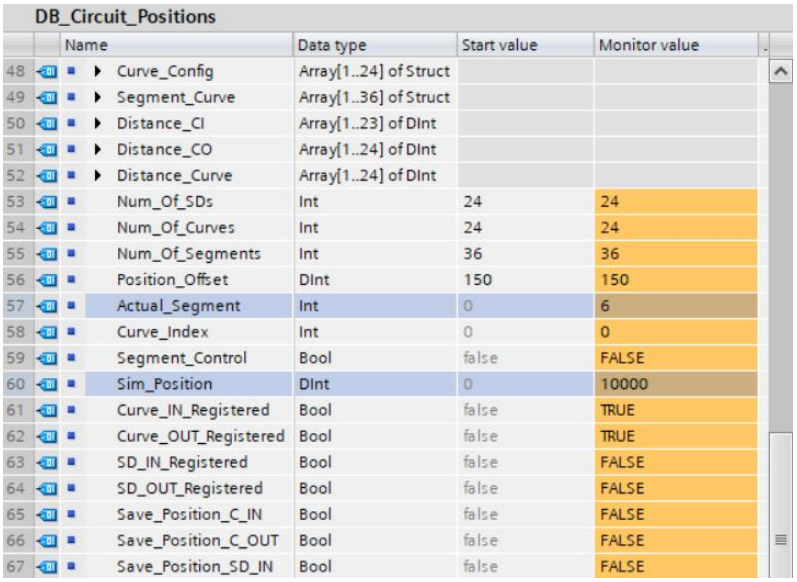


Figura 125 – DB_Circuit_Positions inicialmente vazia

Na Figura 125 é possível verificar as três janelas que são analisadas. A primeira, DB_Circuit_Positions, consiste na base de dados que possui as configurações das curvas e armazena as posições detetadas. A segunda, DB_RFID_Data, abrange os códigos detetados pelo leitor RFID. Já a terceira janela, baseia-se na tabela de visualização dos parâmetros do leitor.



DB_Circuit_Positions

	Name	Data type	Start value	Monitor value
48	Curve_Config	Array[1..24] of Struct		
49	Segment_Curve	Array[1..36] of Struct		
50	Distance_CI	Array[1..23] of DInt		
51	Distance_CO	Array[1..24] of DInt		
52	Distance_Curve	Array[1..24] of DInt		
53	Num_Of_SDs	Int	24	24
54	Num_Of_Curves	Int	24	24
55	Num_Of_Segments	Int	36	36
56	Position_Offset	DInt	150	150
57	Actual_Segment	Int	0	6
58	Curve_Index	Int	0	0
59	Segment_Control	Bool	false	FALSE
60	Sim_Position	DInt	0	10000
61	Curve_IN_Registered	Bool	false	TRUE
62	Curve_OUT_Registered	Bool	false	TRUE
63	SD_IN_Registered	Bool	false	FALSE
64	SD_OUT_Registered	Bool	false	FALSE
65	Save_Position_C_IN	Bool	false	FALSE
66	Save_Position_C_OUT	Bool	false	FALSE
67	Save_Position_SD_IN	Bool	false	FALSE

Figura 126 – Alteração do segmento e posição atual para registo das curvas

Visto que o RGV não se encontra num circuito, é necessário alterar manualmente os valores do segmento e da posição, como mostra a Figura 126.

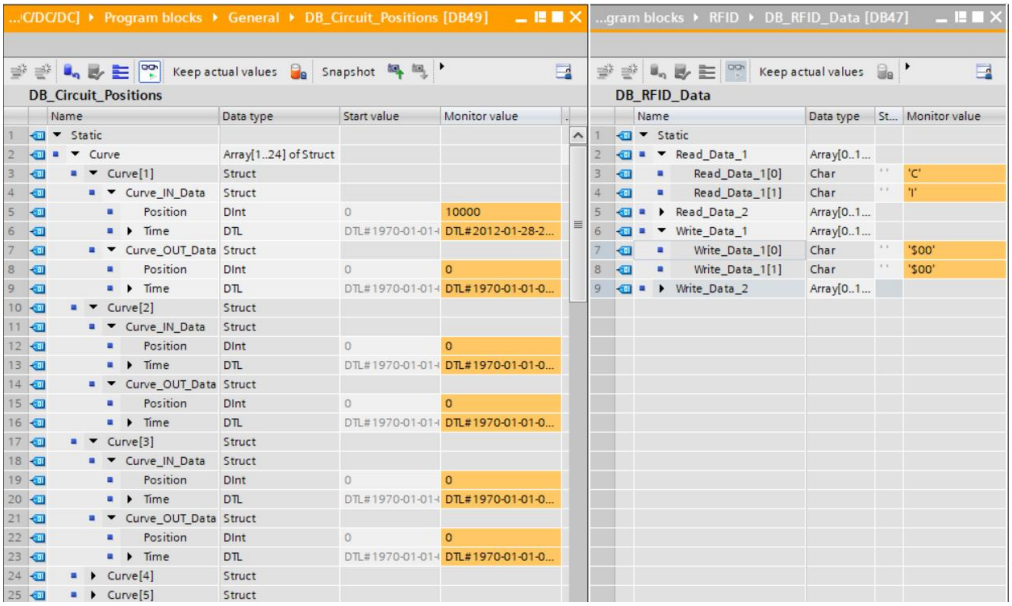


Figura 127 – Deteção da posição de entrada da curva 1

Após ser detetado o código “CI”, a posição 1000, que é colocada manualmente, é armazenada na primeira curva, que está presente no segmento 6.

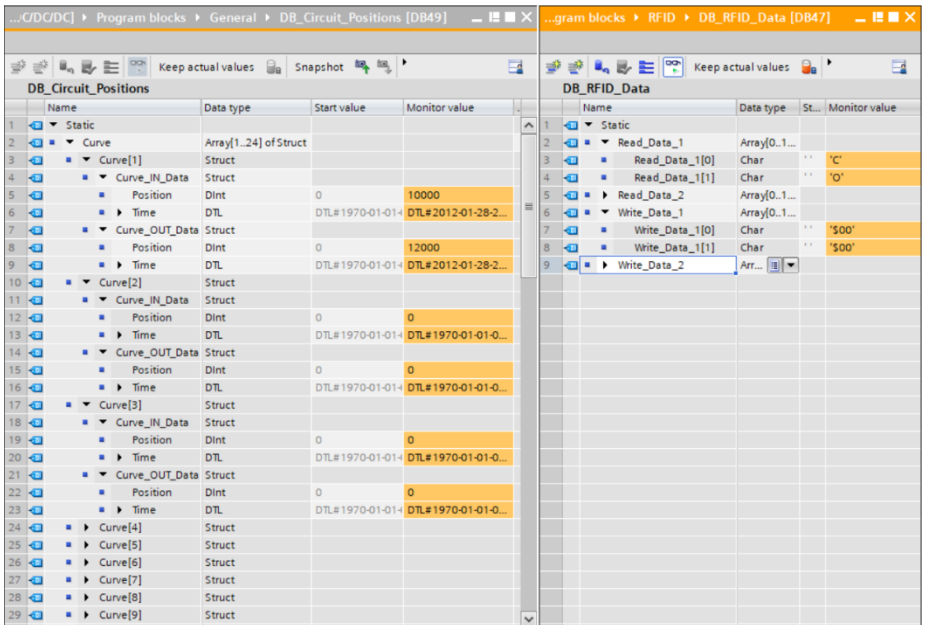


Figura 128 – Deteção da posição de saída da curva 1

Após alterar o valor da posição novamente, deteta-se o código “CO” e regista-se a posição de saída da curva correspondente, como está presente na Figura 128. O mesmo acontece na Figura 129, verificando-se o registo de posições de curvas consecutivas no mesmo segmento.

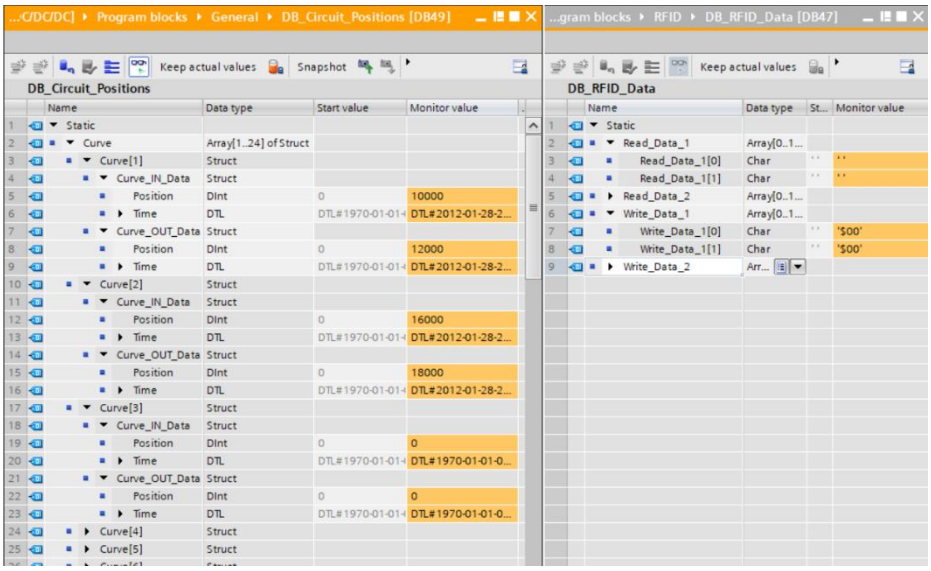


Figura 129 – Deteção das posições de entrada e saída da curva 2

Alterando-se o valor do segmento atual para 8, a próxima curva a ser detetada será a curva 3. Isto verifica-se na Figura 130.

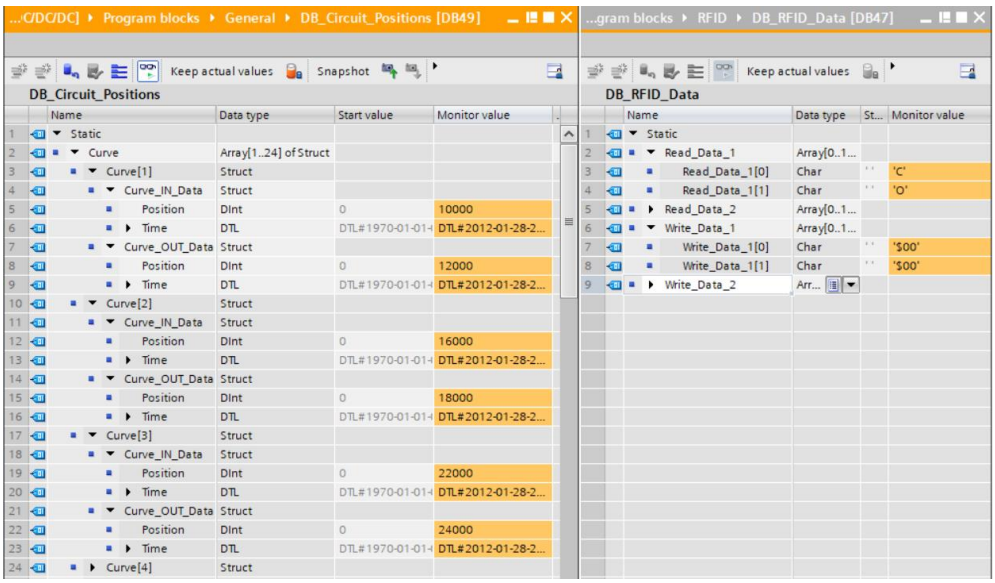


Figura 130 – Deteção das posições de entrada e saída da curva 3

Com isto, verifica-se que o funcionamento lógico da rotina, ou seja, a correspondência dos segmentos das curvas e o registo de posições disponíveis, encontra-se a funcionar corretamente.

Além das posições das curvas, é possível ainda verificar as suas distâncias, armazenadas no vetor Distance_Curve.

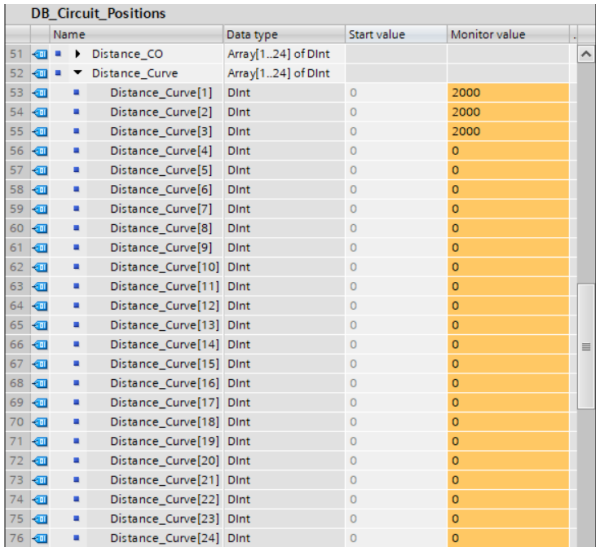
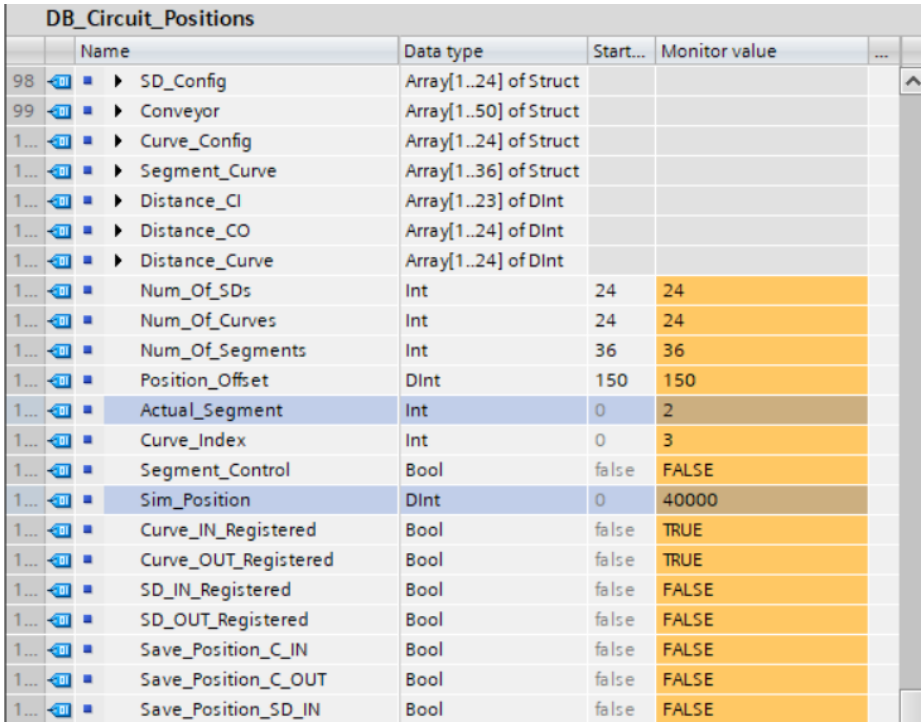


Figura 131 – Distância das curvas detetadas

Simulação do registo das agulhagens

Na simulação do registo das posições das agulhagens, tal como nas curvas, é necessário inicialmente escrever os códigos nas *tags* correspondentes. Os códigos que se pretende detetar são “AI” e “AO”. O vetor que armazena os valores das posições das agulhagens, Switch, encontra-se vazio no começo.

Para verificar o funcionamento da lógica do registo das agulhagens, são analisadas as agulhagens SD3 e SD4, presentes na Figura 90. Os valores dos segmentos das entradas e das saídas destas agulhagens estão presentes no Anexo B.



DB_Circuit_Positions

	Name	Data type	Start...	Monitor value
98	SD_Config	Array[1..24] of Struct		
99	Conveyor	Array[1..50] of Struct		
1...	Curve_Config	Array[1..24] of Struct		
1...	Segment_Curve	Array[1..36] of Struct		
1...	Distance_CI	Array[1..23] of DInt		
1...	Distance_CO	Array[1..24] of DInt		
1...	Distance_Curve	Array[1..24] of DInt		
1...	Num_Of_SDs	Int	24	24
1...	Num_Of_Curves	Int	24	24
1...	Num_Of_Segments	Int	36	36
1...	Position_Offset	DInt	150	150
1...	Actual_Segment	Int	0	2
1...	Curve_Index	Int	0	3
1...	Segment_Control	Bool	false	FALSE
1...	Sim_Position	DInt	0	40000
1...	Curve_IN_Registered	Bool	false	TRUE
1...	Curve_OUT_Registered	Bool	false	TRUE
1...	SD_IN_Registered	Bool	false	FALSE
1...	SD_OUT_Registered	Bool	false	FALSE
1...	Save_Position_C_IN	Bool	false	FALSE
1...	Save_Position_C_OUT	Bool	false	FALSE
1...	Save_Position_SD_IN	Bool	false	FALSE

Figura 132 – Alteração do segmento e posição atual para registo das agulhagens

Como já foi mencionado, as agulhagens, ao contrário das curvas, possuem múltiplas entradas e saídas. Para registar-se a posição da agulhagem correspondente é necessário encontrar uma correspondência no valor do segmento. Ao alterar o valor do segmento atual para 2, e ao passar a *tag* com o código “CI”, é armazenada a posição 40000 na primeira entrada da agulhagem SD3. A Figura X retrata o resultado pretendido.

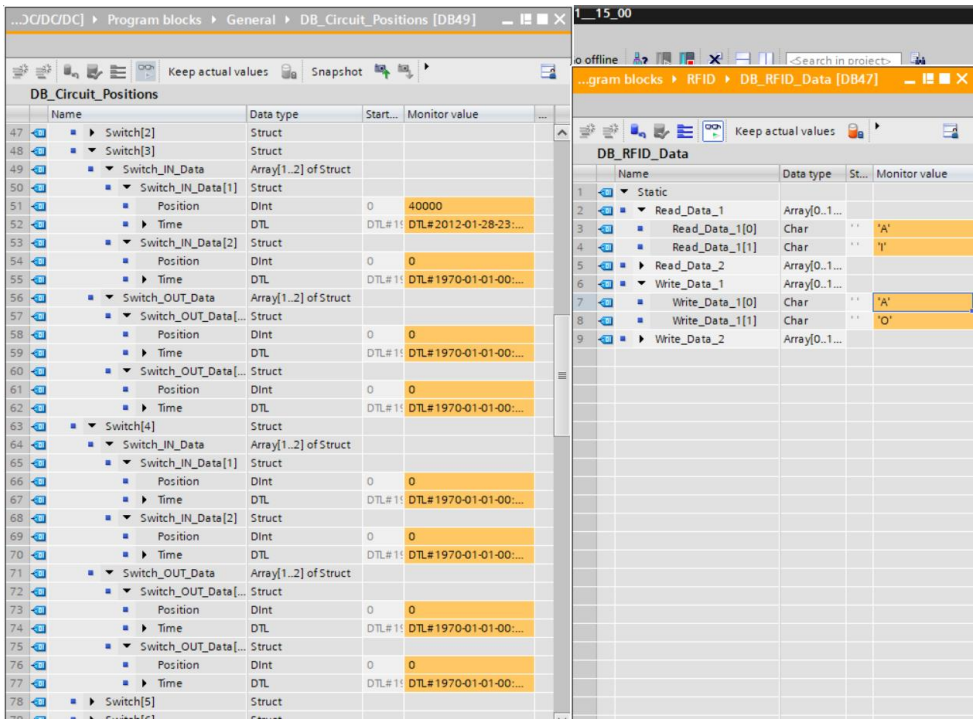


Figura 133 – Deteção da posição da primeira entrada da agulhagem SD3

A agulhagem SD3 é do tipo 2, isto é, possui duas entradas e uma saída. A sua saída está presente no segmento 3. Alterando-se o valor do segmento atual para 3 e ao passar uma *tag* com o código "AO", é registada a posição 42000 na saída desta agulhagem, como está presente na Figura 134.

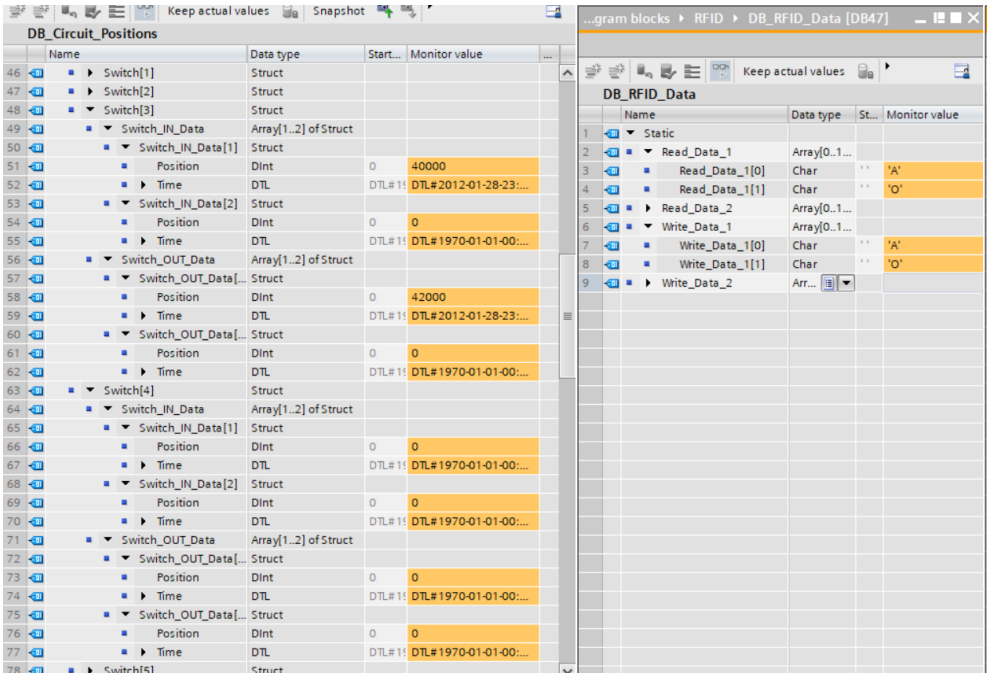


Figura 134 – Deteção da posição da saída da agulhagem SD3

A agulhagem SD4 é do tipo 1, ou seja, tem uma entrada e duas saídas. A entrada encontra-se no segmento 21. Ao alterar o valor do segmento atual para 21 e se o leitor detetar o código “AI”, é registada uma posição simulada de 120000.

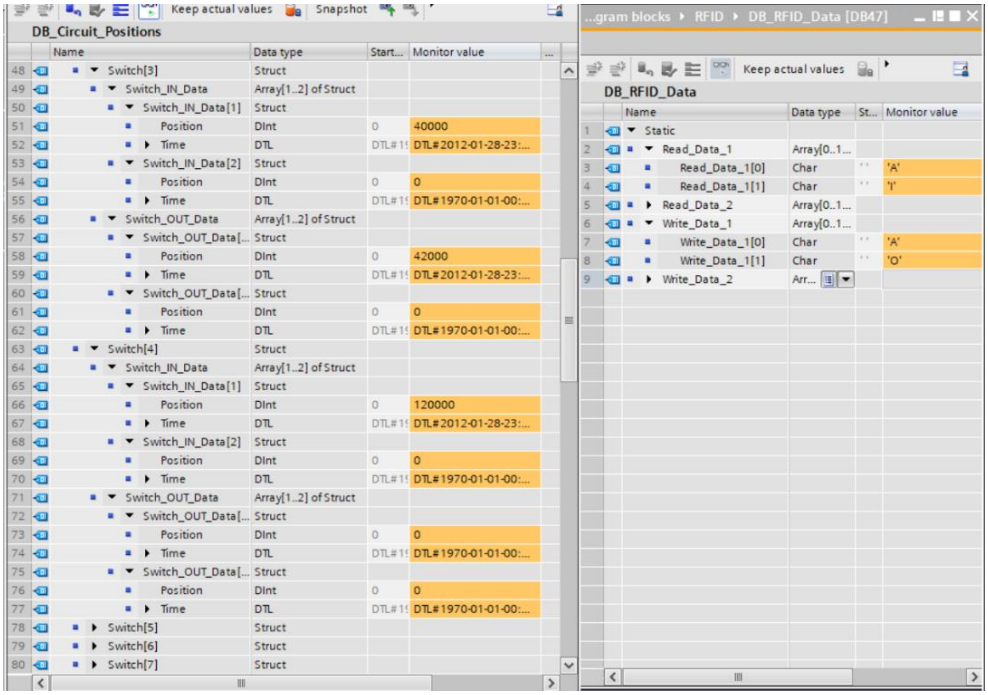
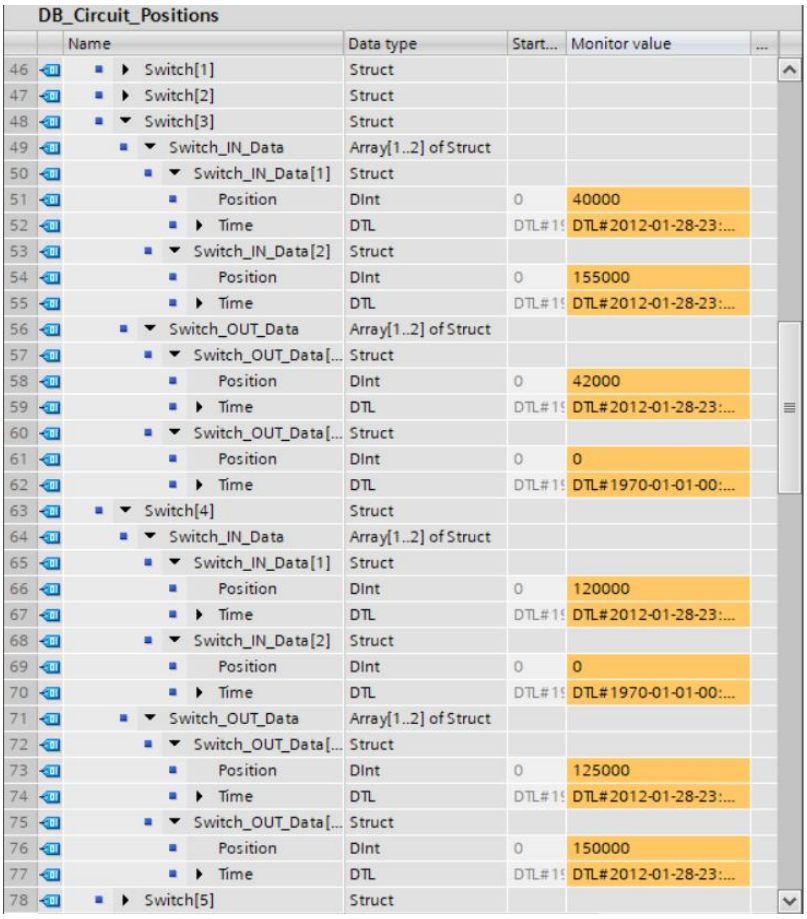


Figura 135 – Deteção da posição da entrada da agulhagem SD4

Após o leitor registar a posição da primeira saída da agulhagem SD4, para registar a segunda saída é necessário o RGV dar outra volta ao circuito. O veículo ao passar novamente pela agulhagem SD3 não deteta as posições já registadas, entrando em ação o mecanismo de bloqueio.

Desta vez a agulhagem SD4 é ativada para a posição de curva, permitindo o registo da saída desta agulhagem e a segunda entrada da agulhagem SD3. Após a deteção destas posições, é possível observar na Figura 136, que a correspondência entre segmentos ocorre corretamente, armazenando os valores das posições nas agulhagens indicadas pelo Anexo B.



DB_Circuit_Positions

	Name	Data type	Start...	Monitor value
46	Switch[1]	Struct		
47	Switch[2]	Struct		
48	Switch[3]	Struct		
49	Switch_IN_Data	Array[1..2] of Struct		
50	Switch_IN_Data[1]	Struct		
51	Position	DInt	0	40000
52	Time	DTL	DTL#1	DTL#2012-01-28-23:...
53	Switch_IN_Data[2]	Struct		
54	Position	DInt	0	155000
55	Time	DTL	DTL#1	DTL#2012-01-28-23:...
56	Switch_OUT_Data	Array[1..2] of Struct		
57	Switch_OUT_Data[...	Struct		
58	Position	DInt	0	42000
59	Time	DTL	DTL#1	DTL#2012-01-28-23:...
60	Switch_OUT_Data[...	Struct		
61	Position	DInt	0	0
62	Time	DTL	DTL#1	DTL#1970-01-01-00:...
63	Switch[4]	Struct		
64	Switch_IN_Data	Array[1..2] of Struct		
65	Switch_IN_Data[1]	Struct		
66	Position	DInt	0	120000
67	Time	DTL	DTL#1	DTL#2012-01-28-23:...
68	Switch_IN_Data[2]	Struct		
69	Position	DInt	0	0
70	Time	DTL	DTL#1	DTL#1970-01-01-00:...
71	Switch_OUT_Data	Array[1..2] of Struct		
72	Switch_OUT_Data[...	Struct		
73	Position	DInt	0	125000
74	Time	DTL	DTL#1	DTL#2012-01-28-23:...
75	Switch_OUT_Data[...	Struct		
76	Position	DInt	0	150000
77	Time	DTL	DTL#1	DTL#2012-01-28-23:...
78	Switch[5]	Struct		

Figura 136 – Deteção das posições das agulhagens SD3 e SD4

6.2. DETEÇÃO DE RGVS

Foram realizados alguns testes da deteção de RGVs, com o objetivo de verificar o funcionamento do envio de comandos do *Master Controller* e a deteção de distâncias máximas entre RGVs.

6.2.1. SIMULAÇÃO DO ENVIO DE COMANDOS DO *MASTER CONTROLLER*

Uma vez que não é possível testar o programa com dois RGVs num circuito real, o teste da verificação dos comandos do *Master* é realizado através de simulações. As simulações são realizadas através do *software* S7-PLCSIM V16, da Siemens.

Para começar o processo é necessário inicializar as variáveis, como na Figura 137. Esta fase consiste na atribuição do número do RGV, da lista das posições com os pontos da curva e dos números de pontos que se pretende enviar. Após inicializar as variáveis, pressiona-se o botão Start, isto é, coloca-se a TRUE.

DB_MC_DATA_CONFIG

	Name	Data type	Start value	Monitor value
1	Static			
2	RGV_Num	Struct		
3	RGV_Front	Int	1	1
4	RGV_Back	Int	2	2
5	Curve_Positions	Array[1..10] of DInt		
6	Curve_Positions[1]	DInt	0	10000
7	Curve_Positions[2]	DInt	0	10200
8	Curve_Positions[3]	DInt	0	10400
9	Curve_Positions[4]	DInt	0	10600
10	Curve_Positions[5]	DInt	0	10800
11	Curve_Positions[6]	DInt	0	11000
12	Curve_Positions[7]	DInt	0	11200
13	Curve_Positions[8]	DInt	0	11400
14	Curve_Positions[9]	DInt	0	11600
15	Curve_Positions[10]	DInt	0	12000
16	Min	Int	1	1
17	Max	Int	1	2
18	Index_Position	Int	0	1
19	Actual_Macro_Step	Int	0	1
20	Previous_Macro_Step	Int	0	1
21	Number_of_Cycles	Int	0	0
22	Start	Bool	false	TRUE
23	End	Bool	false	FALSE

Figura 137 – Inicialização das variáveis

Depois da inicialização das variáveis, o processo entra no *Macro Step* 1. Nesta etapa, o RGV da frente está no *step* 62, isto é, recebeu o comando RESET. Podem-se verificar os *steps* através do DB_RGV_COM_STEP, Figura 138.

DB_RGV_COM_STEP

	Name	Data type	Offset	Start value	Monitor value
1	Static				
2	RGV	Array[1..64] of Int	0.0		
3	RGV[1]	Int	0.0	5	62
4	RGV[2]	Int	2.0	5	5

Figura 138 – *Step* do primeiro RGV em início de *reset*

Quando o RGV recebe o comando para dar *reset* ele avisa o *Master* que está a reiniciar. Este processo simula-se através da ativação do parâmetro Reset_On_Exec na base de dados de receção do *Master*.

DB_RGV_RECV

	Name	Data type	Offset	Start value	Monitor value
1	Preset_To_Exec	Bool	29.6	false	FALSE
1	Insert_To_Exec	Bool	29.7	false	FALSE
1	Movement_On_Exec	Bool	30.0	false	FALSE
1	B30_b1	Bool	30.1	false	FALSE
1	Reset_On_Exec	Bool	30.2	false	TRUE
1	Stop_On_Exec	Bool	30.3	false	FALSE
1	B30_b4	Bool	30.4	false	FALSE
1	B30_b5	Bool	30.5	false	FALSE
1	Preset_On_Exec	Bool	30.6	false	FALSE
1	Insert_On_Exec	Bool	30.7	false	FALSE
1	Movement_Finished	Bool	31.0	false	FALSE
1	B31_b1	Bool	31.1	false	FALSE
1	Reset_Finished	Bool	31.2	false	FALSE
1	Stop_Finished	Bool	31.3	false	FALSE

Figura 139 – Simulação do RGV em *reset*

A ativação deste parâmetro leva a que RGV mude para o *step* 64, Figura 140.

DB_RGV_COM_STEP

	Name	Data type	Offset	Start value	Monitor value
1	Static				
2	RGV	Array[1..64] of Int	0.0		
3	RGV[1]	Int	0.0	5	64
4	RGV[2]	Int	2.0	5	5

Figura 140 – *Step* do primeiro RGV em processo de *reset*

O comando RESET está terminado após o *Master* receber a confirmação do RGV, através do parâmetro Reset_Finished, como mostra a Figura 141. Com isto o *step* do veículo fica com o valor *default* 5.

DB_RGV_RECV

	Name	Data type	Offset	Start value	Monitor value
1	Preset_To_Exec	Bool	29.6	false	FALSE
1	Insert_To_Exec	Bool	29.7	false	FALSE
1	Movement_On_Exec	Bool	30.0	false	FALSE
1	B30_b1	Bool	30.1	false	FALSE
1	Reset_On_Exec	Bool	30.2	false	FALSE
1	Stop_On_Exec	Bool	30.3	false	FALSE
1	B30_b4	Bool	30.4	false	FALSE
1	B30_b5	Bool	30.5	false	FALSE
1	Preset_On_Exec	Bool	30.6	false	FALSE
1	Insert_On_Exec	Bool	30.7	false	FALSE
1	Movement_Finished	Bool	31.0	false	FALSE
1	B31_b1	Bool	31.1	false	FALSE
1	Reset_Finished	Bool	31.2	false	TRUE
1	Stop_Finished	Bool	31.3	false	FALSE

Figura 141 – Simulação do RGV com *reset* concluído

DB_RGV_COM_STEP

	Name	Data type	Offset	Start value	Monitor value
1	Static				
2	RGV	Array[1..64] of Int	0.0		
3	RGV[1]	Int	0.0	5	5
4	RGV[2]	Int	2.0	5	5

Figura 142 – *Step* do primeiro RGV com *reset* concluído

Para o *Master* poder enviar o comando POSITIONING ao RGV da frente, precisa primeiro de confirmar as condições desse RGV. Na Figura 143 verificam-se, a destacado, as condições necessárias. Se as condições se verificarem o comando é enviado juntamente com a primeira posição (presente na Figura 144), e o RGV entra no *step* 22.

DB_RGV_RECV

	Name	Data type	Offset	Start value	Monitor value
76	B10_b6	Bool	10.6	false	FALSE
77	B10_b7	Bool	10.7	false	FALSE
78	Automatic	Bool	11.0	false	TRUE
79	Manual	Bool	11.1	false	FALSE
80	B11_b2	Bool	11.2	false	FALSE
81	Route_Change_Ena...	Bool	11.3	false	FALSE
82	Reseting	Bool	11.4	false	FALSE
83	Ready	Bool	11.5	false	TRUE
84	Stopped	Bool	11.6	false	FALSE
85	Emergency	Bool	11.7	false	FALSE
86	Pos_Act_X	DInt	12.0	0	0
87	Moving_Front	Bool	16.0	false	FALSE
88	Moving_Rear	Bool	16.1	false	FALSE
89	Positioned_X	Bool	16.2	false	FALSE
90	Actual_Position_Valid	Bool	16.3	false	TRUE
91	Cnv01_LU_Received	Bool	16.4	false	FALSE

Figura 143 – Verificação das condições do primeiro RGV para POSITIONING

DB_RGV_SEND

	Name	Data type	Offset	Start value	Monitor value
1	Static				
2	RGV	Array[1..64] of Struct	0.0		
3	RGV[1]	Struct	0.0		
4	Pos_X_Final	DInt	0.0	0	10000

Figura 144 – Envio da primeira posição

DB_RGV_COM_STEP

	Name	Data type	Offset	Start value	Monitor value
1	Static				
2	RGV	Array[1..64] of Int	0.0		
3	RGV[1]	Int	0.0	5	22
4	RGV[2]	Int	2.0	5	5

Figura 145 – *Step* do primeiro RGV em início de posicionamento

Tal como no comando RESET, após receber o comando POSITIONING, o RGV envia uma mensagem a indicar que está em movimento, Figura 146. Após receber a mensagem, o *Master* coloca o RGV no *step* 24.

DB_RGV_RECV

	Name	Data type	Offset	Start value	Monitor value
1...	Preset_To_Exec	Bool	29.6	false	FALSE
1...	Insert_To_Exec	Bool	29.7	false	FALSE
1...	Movement_On_Exec	Bool	30.0	false	TRUE
1...	B30_b1	Bool	30.1	false	FALSE
1...	Reset_On_Exec	Bool	30.2	false	FALSE
1...	Stop_On_Exec	Bool	30.3	false	FALSE
1...	B30_b4	Bool	30.4	false	FALSE
1...	B30_b5	Bool	30.5	false	FALSE
1...	Preset_On_Exec	Bool	30.6	false	FALSE
1...	Insert_On_Exec	Bool	30.7	false	FALSE
1...	Movement_Finished	Bool	31.0	false	FALSE
1...	B31_b1	Bool	31.1	false	FALSE
1...	Reset_Finished	Bool	31.2	false	FALSE
1...	Stop_Finished	Bool	31.3	false	FALSE

Figura 146 – Simulação do RGV em posicionamento

DB_RGV_COM_STEP

	Name	Data type	Offset	Start value	Monitor value
1	Static				
2	RGV	Array[1..64] of Int	0.0		
3	RGV[1]	Int	0.0	5	24
4	RGV[2]	Int	2.0	5	5

Figura 147 – *Step* do primeiro RGV em processo de posicionamento

Com a receção da confirmação da chegada do RGV à primeira posição, presente na Figura 148, o *Master* dá como terminado o envio do comando, colocando o *step* do veículo a 5.

DB_RGV_RECV

	Name	Data type	Offset	Start value	Monitor value
1...	Preset_To_Exec	Bool	29.6	false	FALSE
1...	Insert_To_Exec	Bool	29.7	false	FALSE
1...	Movement_On_Exec	Bool	30.0	false	FALSE
1...	B30_b1	Bool	30.1	false	FALSE
1...	Reset_On_Exec	Bool	30.2	false	FALSE
1...	Stop_On_Exec	Bool	30.3	false	FALSE
1...	B30_b4	Bool	30.4	false	FALSE
1...	B30_b5	Bool	30.5	false	FALSE
1...	Preset_On_Exec	Bool	30.6	false	FALSE
1...	Insert_On_Exec	Bool	30.7	false	FALSE
1...	Movement_Finished	Bool	31.0	false	TRUE
1...	B31_b1	Bool	31.1	false	FALSE
1...	Reset_Finished	Bool	31.2	false	FALSE
1...	Stop_Finished	Bool	31.3	false	FALSE

Figura 148 – Simulação do RGV com posicionamento concluído

Estes dois comandos repetem-se para o segundo RGV. Concluídos ambos os comandos no RGV de trás, incrementa-se o número de ciclos e enviam-se os comando RESET e POSITIONING ao veículo da frente, ficando desta forma o processo em ciclo.

DB_MC_DATA_CONFIG

	Name	Data type	Start value	Monitor value
1	Static			
2	RGV_Num	Struct		
3	RGV_Front	Int	1	1
4	RGV_Back	Int	2	2
5	Curve_Positions	Array[1..10] of DInt		
6	Curve_Positions[1]	DInt	0	10000
7	Curve_Positions[2]	DInt	0	10200
8	Curve_Positions[3]	DInt	0	10400
9	Curve_Positions[4]	DInt	0	10600
10	Curve_Positions[5]	DInt	0	10800
11	Curve_Positions[6]	DInt	0	11000
12	Curve_Positions[7]	DInt	0	11200
13	Curve_Positions[8]	DInt	0	11400
14	Curve_Positions[9]	DInt	0	11600
15	Curve_Positions[10]	DInt	0	11800
16	Min	Int	1	1
17	Max	Int	1	2
18	Index_Position	Int	0	2
19	Actual_Macro_Step	Int	0	1
20	Previous_Macro_Step	Int	0	1
21	Number_of_Cycles	Int	0	1
22	Start	Bool	false	FALSE
23	End	Bool	false	FALSE

Figura 149 – Aumento do índice de posição e número de ciclos

Através destas simulações, verifica-se que a função desenvolvida na troca de comandos entre o *Master Controller* e o RGV encontra-se a funcionar corretamente.

6.2.2. TESTE DE DETEÇÃO DE RGVS

Como não havia de hipótese de utilizar dois RGVs, simulou-se a traseira do RGV da frente através de dois postes interligados por uma fita. Para verificar a precisão e a repetibilidade do sistema foram realizados dois ensaios com várias tentativas.

No primeiro ensaio, a fita é colocada linearmente em frente ao RGV, como indicam as Figura 150 e 151. Desta forma é possível verificar o comportamento do veículo e do programa numa situação em que se desloca numa zona reta.

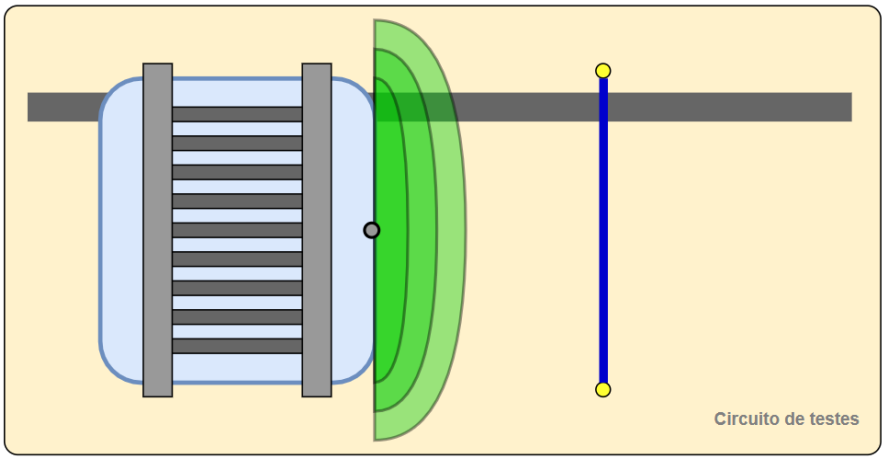


Figura 150 – Esquema do teste de deteção frontal da fita

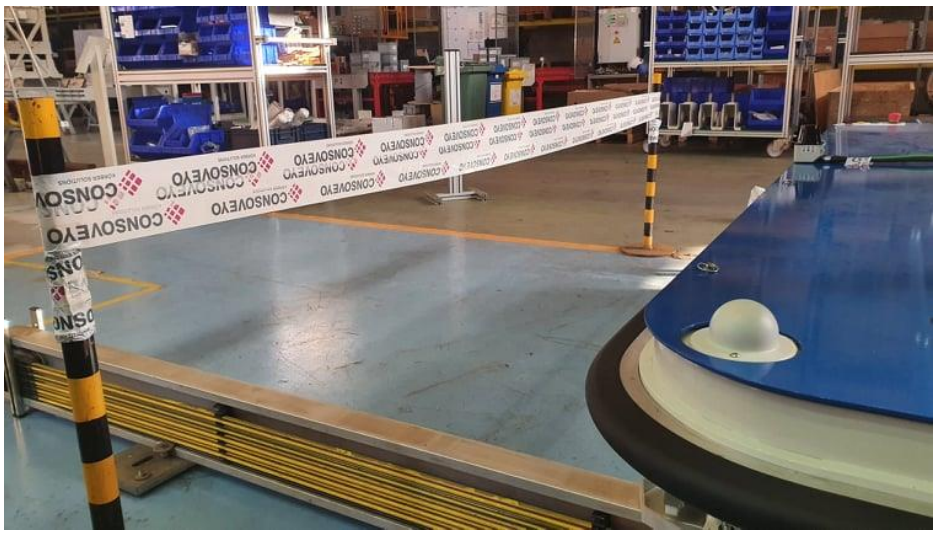


Figura 151 – Fita colocada numa posição linear à frente do RGV

Foram realizadas dez tentativas de deteção da fita. Na Tabela 13 estão representados os valores da posição registada no momento em que a distância do RGV chegasse aos 0,2 m.

Tabela 13 – Posições registadas no teste frontal

Tentativa	Posição registada (mm)
1	110957
2	110956
3	110959
4	110956
5	110956
6	110957
7	110958
8	110957
9	110954
10	110955

Estes valores mostram-se bastante repetitivos, uma vez que a diferença entre a máxima e a mínima posição detetada é 5 mm. Como neste sistema as velocidades dos RGVs são reduzidas e a distância ao RGV da frente é de 20 cm, um valor de 5 mm de diferença mostra-se aceitável.

O segundo ensaio serve para verificar o comportamento da deteção dos RGVs nas curvas. Para isso é colocada uma fita, na diagonal, em frente ao RGV, como está representado nas Figuras 152 e 153. Deste modo pretende-se que o sensor detete a fita em zonas laterais.

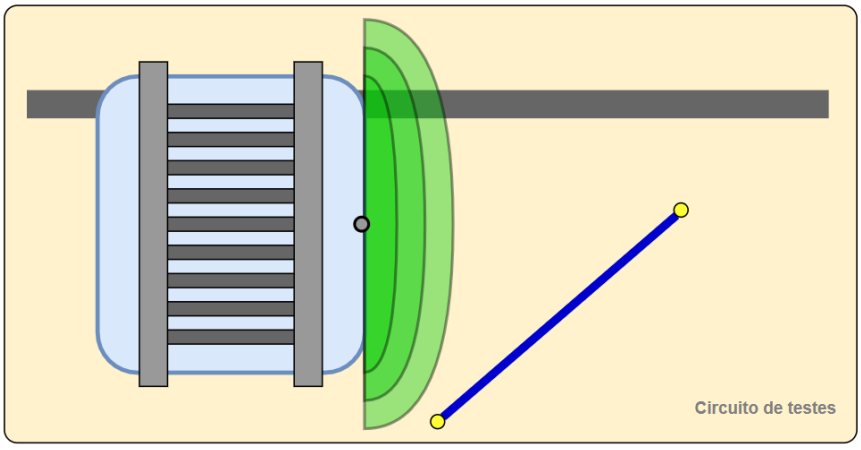


Figura 152 – Esquema do teste de deteção diagonal da fita

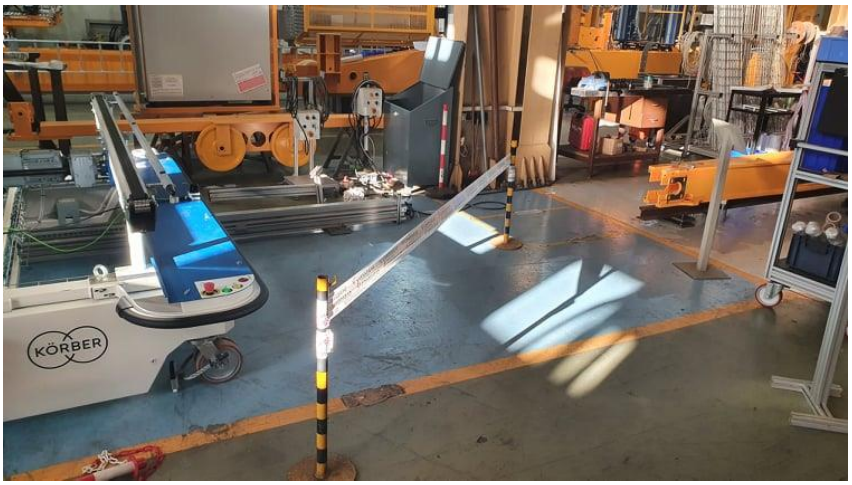


Figura 153 – Fita colocada numa posição diagonal à frente do RGV

A Tabela 14 indica os resultados das dez tentativas realizadas. Cada posição indica o momento em que a distância entre os RGVs era 0,2 m.

Tabela 14 – Posições registadas no teste diagonal

Tentativa	Posição registada (mm)
1	110611
2	110608
3	110603
4	110615
5	110604
6	110599
7	110608
8	110612
9	110611
10	110604

A diferença entre a máxima e a mínima posição detetada é de cerca de 16 mm. Este valor excede a diferença do ensaio anterior, mostrando maiores irregularidades na deteção das curvas. No entanto, um valor de 16 mm acaba por ser uma variação curta relativamente aos 20 cm de distância entre RGVs. Estes valores acabam por se mostrar aceitáveis para a implementação deste sistema.

Os momentos de deteção da fita a 20 cm podem ser observados através do PC, pelo *software* SOPAS, como estão representados nas Figuras 154 e 155.

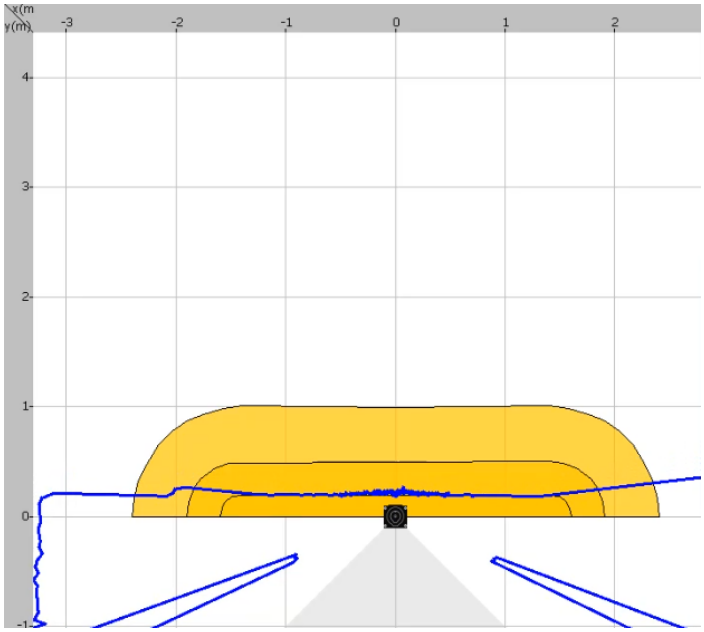


Figura 154 – Verificação do momento de deteção frontal

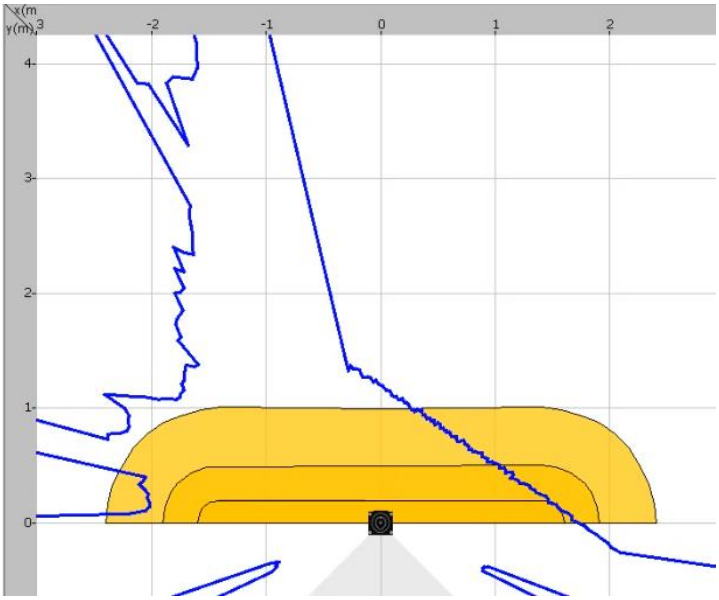


Figura 155 – Verificação do momento de deteção diagonal

6.3. ALINHAMENTO DOS TRANSPORTADORES

Os testes realizados nesta etapa têm como objetivo a verificação do processo de alinhamento desenvolvido. Pretende-se, assim, verificar o desvio da posição real de alinhamento.

Como no circuito de testes não existe transportadores, estes tiveram de ser simulados através de dois suportes. Estes suportes possuem as mesmas dimensões das pernas dos transportadores e foram colocados junto ao carril. A Figura 156 representa um esquema aproximado do circuito utilizado para o teste de alinhamento.

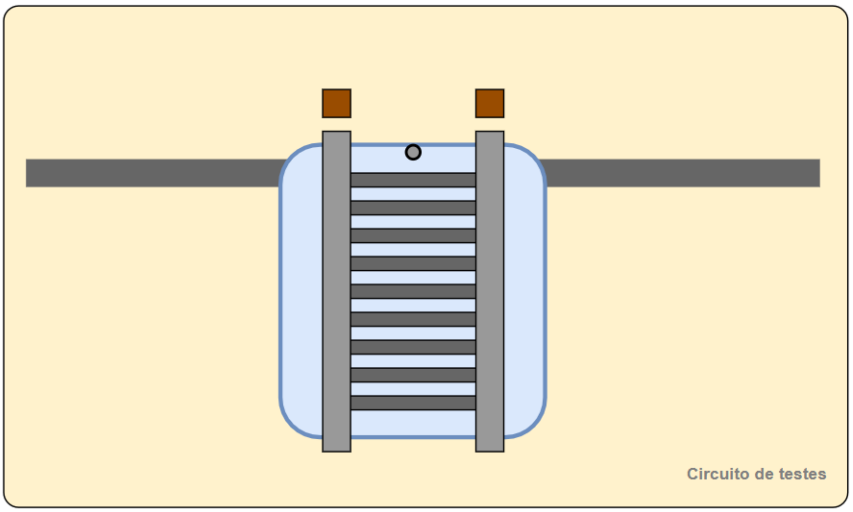


Figura 156 – Esquema do circuito de testes de alinhamento

Figura 157 – Instalação do sensor no RGV

Como é possível observar nas Figuras 156 e 157, o sensor é instalado na lateral do RGV, de forma a poder identificar os suportes. A distância entre os suportes é a mesma distância entre o transportador do RGV.

Os testes consistem na realização do processo de alinhamento, através da rotina desenvolvida, com a utilização de dois conjuntos diferentes de configurações. Com isto, pretende-se verificar qual dos conjuntos apresenta o menor desvio da posição detetada relativamente à posição real.

Inicialmente, é necessário retirar o valor da posição real de alinhamento. Para isso, alinharam-se os transportadores manualmente, adquirindo a posição que se encontra no *encoder*. Nestes testes, a posição real obtida é 110810.

Para cada conjunto de configurações foram realizadas cinco tentativas de registo da posição de alinhamento. Em ambos os conjuntos, a primeira configuração é igual, alterando-se apenas a segunda configuração. Na Figura 158 encontra-se representada a primeira configuração, vista a partir do *software* SOPAS.

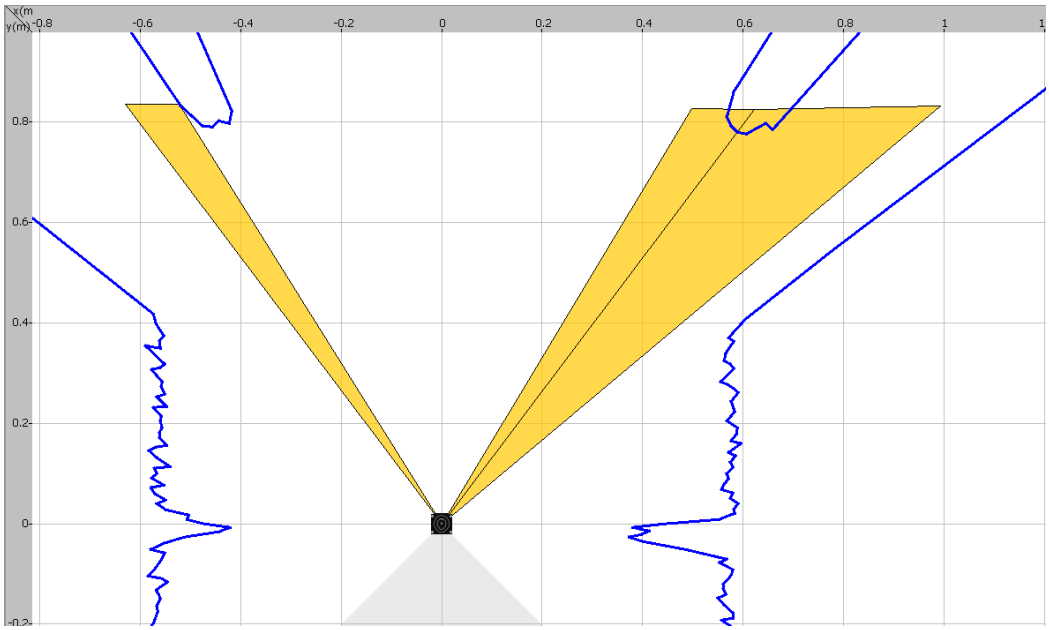


Figura 158 – Primeira configuração do sensor

No primeiro conjunto, a segunda configuração tem o aspeto da Figura 159.

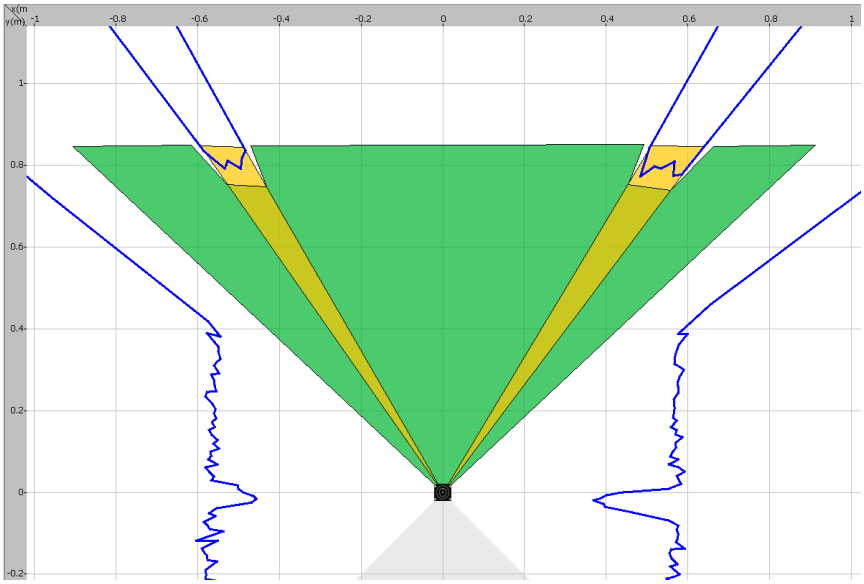


Figura 159 – Segunda configuração do primeiro conjunto

No primeiro conjunto de configurações, após correr várias vezes a rotina de alinhamento desenvolvida, obtiveram-se os valores presentes na Tabela 15.

Tabela 15 – Valores obtidos no primeiro conjunto

Tentativa	Posição registada (mm)	Desvio da posição real (mm)	Desvio médio (mm)
1	110802	8	8,4
2	110800	10	
3	110803	7	
4	110802	8	
5	110801	9	

Os valores registados de desvio são bastante significativos. Um desalinhamento com cerca de 4 mm pode implicar a má troca de cargas ou até mesmo a queda das mesmas. Com a utilização desta segunda configuração, o desvio médio das cinco tentativas é 8,4 mm. Este valor resulta do espaço existente entre o Field3 e as pernas do transportador.

Tendo este valor em conta, desenvolveu-se outra configuração. A nova configuração criada, presente na Figura 160, faz parte do segundo conjunto de configurações.

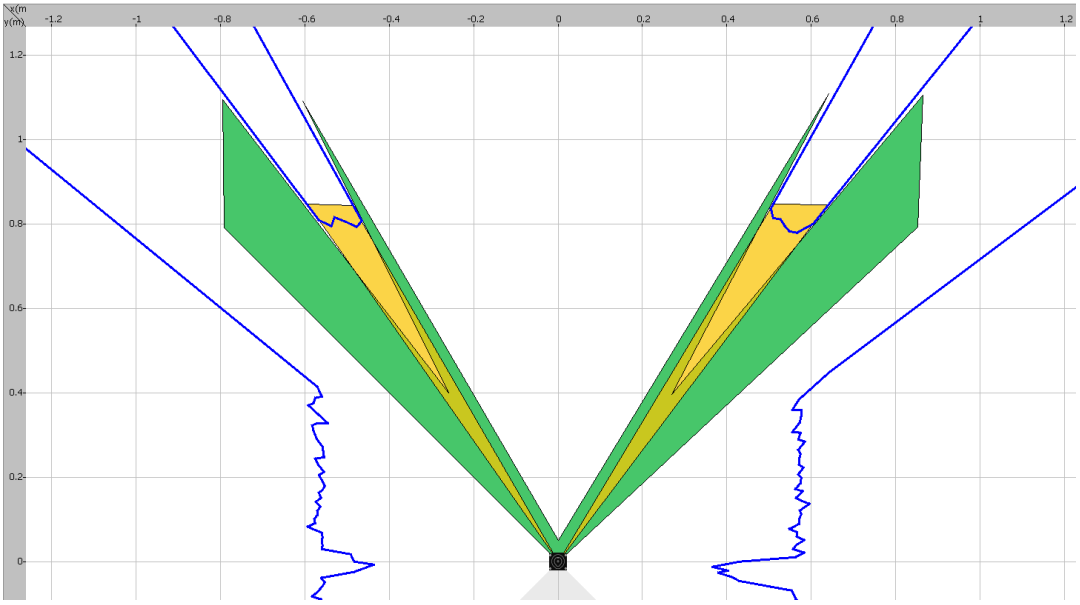


Figura 160 – Segunda configuração do segundo conjunto

Uma vez que o sensor possui uma resolução angular de 1°, para diminuir a distância entre o Field3 e as pernas do transportador é necessário colocar os pontos do Field3 mais distantes. Deste modo é possível obter valores precisos, como se pode observar na Tabela 16.

Tabela 16 – Valores obtidos no segundo conjunto

Tentativa	Posição registada (mm)	Desvio da posição real (mm)	Desvio médio (mm)
1	110809	1	0,8
2	110808	2	
3	110810	0	
4	110810	0	
5	110809	1	

A utilização desta configuração permite obter valores de alinhamento bastante precisos. O desvio médio das cinco medições é 0,8 mm, sendo que em nenhuma das tentativas o desvio ultrapassou os 2 mm. Estes valores mostram-se aceitáveis, podendo declarar-se esta hipótese como ideal para o alinhamento dos transportadores.

Na Figura 161 é possível observar o momento do registo da posição de alinhamento, visto pelos *softwares* do PLC e do sensor. Pode-se verificar que apenas as duas entradas IN_Field1 e IN_Field2 estão ativas e que a posição registada apresenta um desvio de 0 mm relativamente à posição real de alinhamento.

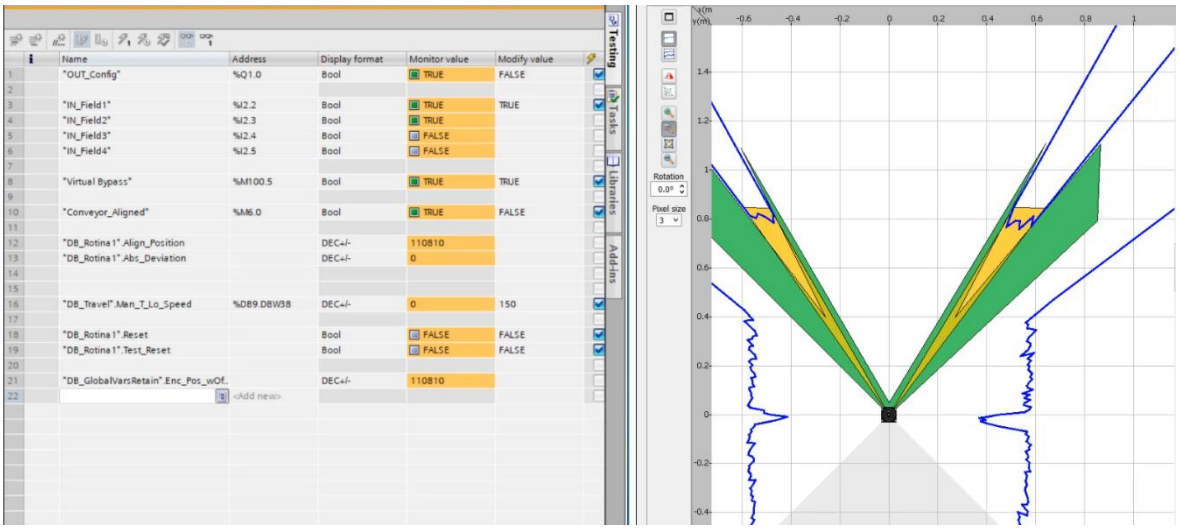


Figura 161 – Verificação do funcionamento da rotina de alinhamento

7. CONCLUSÃO

Neste capítulo são abordadas as principais conclusões a retirar do desenvolvimento deste projeto, e sugestões de possíveis melhorias.

Relativamente ao registo dos limites das curvas e das agulhagens, verificou-se que a utilização de leitores e das *tags* RFID funciona corretamente, desde a deteção dos códigos até ao registo das posições. Uma das maiores dificuldades deste desafio foi o desenvolvimento da lógica do registo das posições, visto que as curvas e as agulhagens possuem comportamentos diferentes. Este problema resolveu-se com a implementação da correspondência de segmentos. Os testes indicaram valores aceitáveis para este sistema, dado que para a deteção destes pontos não são necessárias precisões na casa do mm. A velocidade e o alcance do campo de deteção influenciam os resultados obtidos, sendo que a melhor hipótese a implementar neste sistema seria a última.

Uma possível melhoria seria a instalação do leitor RFID no RGV, através de um suporte. Nos testes realizados o leitor foi colocado debaixo do motor de translação. As vibrações e os ruídos magnéticos do motor são fatores de influência nos resultados.

Na deteção de RGVs, a utilização de um sensor LiDAR apresentou-se como uma mais-valia para o sistema, uma vez que mostrou-se ser uma solução fiável. Os testes realizados exibiram valores bastante precisos e repetitivos, principalmente no caso da deteção frontal de RGVs. O aumento da altura da traseira dos RGVs seria uma melhoria a ter em conta, visto que, em certos casos, o leitor teria de estar um pouco inclinado. Com a traseira do RGV mais alta seria possível a instalação do sensor LiDAR de forma plana.

Finalmente, no caso do alinhamento dos transportadores, verificaram-se valores muito satisfatórios, onde por vezes o desvio era 0 mm. O reaproveitamento do sensor LiDAR mostrou-se uma solução económica. O desenvolvimento da rotina de alinhamento para vários transportadores consecutivos poderá ser uma possível melhoria.

De modo geral, conclui-se que todos os objetivos inicialmente indicados foram concluídos com sucesso. Este projeto ofereceu um grande aproveitamento pois foi possível adquirir experiência profissional, rodeado de excelentes profissionais, assim como foram adquiridos novos conhecimentos, novas técnicas e novos métodos de resolução de problemas.

Referências Documentais

[1] V. Lindström, M. Winroth, J. Stahre, and J. Frohm, “Levels of automation in manufacturing,” 2008.

[2] M. P. Groover, *Fundamentals of Modern Manufacturing: Materials, Processes, and Systems*, vol. 7. 1996.

[3] A. Seksak Elsaid, W. A. Mohamed, and S. Ghazy Ramadan, “Speed Control of Induction Motor Using PLC and SCADA System,” *Journal of Engineering Research and Applications*, vol. 6, no. 1, pp. 98–104, 2016.

[4] B. Kehal, “A look into Automation & its Different Types,” *AIAutomation*, Sep. 2019. https://medium.com/aiautomation/a-look-into-automation-its-different-types-f4266049f54d

[5] “Tipos de automação industrial: qual o ideal para sua empresa?,” *Fersiltec*, Mar. 2019. https://fersiltec.com.br/blog/tipos-de-automacao-industrial-ideal-empresa/

[6] A. Hanspal, “Be Paranoid, Android: Here Come The Humans,” *Forbes*. https://www.forbes.com/sites/amarhanspal/2021/02/17/be-paranoid-android-here-come-the-humans/

[7] H. J. Wilson and P. R. Daugherty, “Collaborative Intelligence: Humans and AI Are Joining Forces,” *Harvard Business Review*, Jul. 2018. https://hbr.org/2018/07/collaborative-intelligence-humans-and-ai-are-joining-forces

[8] Zebra, “Warehousing Vision Study,” 2019.

[9] POLLY, “Top 5 Industries That Will Be Transformed By Robotics and Automation,” *Robotics & Automation News*. Jun. 2020. [Online]. Available:

https://roboticsandautomationnews.com/2020/06/17/top-5-industries-that-will-be-transformed-by-robotics-and-automation/33221/

[10] K. Matthews, “5 Industries that Robotics Have Disrupted Drastically,” *Robotics Business Review*, Sep. 2019. https://www.roboticsbusinessreview.com/news/5-industries-that-robotics-have-disrupted-drastically/

[11] P. Baker and Z. Halim, “An exploration of warehouse automation implementations: Cost, service and flexibility issues,” *Supply Chain Management*, vol. 12, no. 2, pp. 129–138, 2007, doi: 10.1108/13598540710737316.

[12] F.-L. Chang, Z.-X. Liu, X. Zheng, and D.-D. Liu, “Research on Order Picking Optimization Problem of Automated Warehouse,” 2007.

[13] “Automated Storage and Retrieval Systems (AS/RS),” *Inc.com*, 2020. https://www.inc.com/encyclopedia/automated-storage-and-retrieval-systems-as-rs.html

[14] “Sistema de recuperação e armazenamento automatizado,” *Honeywell*. [Online]. Available: https://sps.honeywell.com/us/en/products/automation/solutions-by-technology/asrs-systems

[15] “WMS & WCS & WES,” *SCS*. http://www.scs1.co.kr/?page_id=500&lang=en

[16] “Warehouse Control System | Warehouse Control Software,” *Outsource Equipment*. https://www.osequip.com/services/warehouse-control-systems/

[17] “Veículos automaticamente guiados,” *Körber Supply Chain*. https://www.koerber-supplychain.com/pt/solucoes/equipamentos-para-movimentacao-de-materiais/veiculos-automaticamente-guiados/

[18] “Veículos guiados por trilho,” *Körber Supply Chain*. https://www.koerber-supplychain.com/pt/solucoes/equipamentos-para-movimentacao-de-materiais/veiculos-guiados-por-trilho/

[19] V. D. Hunt, A. Puglia, and M. Puglia, *RFID: A Guide To Radio Frequency Identification*. 2006.

[20] Young and James, *LiDAR For Dummies, Autodesk and DLT Solutions Special Edition*. 2011.

[21] B. Schwarz, "LiDAR: Mapping the world in 3D," *Nature Photonics*, vol. 4, no. 7, pp. 429–430, Jul. 2010, doi: 10.1038/nphoton.2010.148.

[22] Oliveira e Sá, "Controladores Lógicos Programáveis (PLC)."

[23] E. Lindsjö, "WCS," *Consafe Logistics*, 2021. https://www.consafelogistics.com/solutions/wcs/

[24] "Stacker Cranes," *Consoveyo*. https://www.consoveyo.com/en/products/stacker-cranes.html

[25] "Supply Chain," *Körber AG*. https://www.koerber.com/en/supply-chain

[26] "O2D222 - Sensor de reconhecimento de objectos," *IFM*. https://www.ifm.com/pt/pt/product/O2D222

[27] "O3D302 - Sensor 3D," *IFM*. https://www.ifm.com/pt/pt/product/O3D302

[28] SICK, "Line guidance and grid localization GLS6 SICK," 2021. [Online]. Available: https://www.sick.com/ag/en/localization-and-positioning-solutions/line-guidance-and-grid-localization/gls6/c/g507063

[29] SICK, "OLS10-BP112311 OLS LINE GUIDANCE AND GRID LOCALIZATION," 2021. [Online]. Available: www.sick.com/OLS

[30] SICK, "MLSE-0300A2NP0 MLS LINE GUIDANCE AND GRID LOCALIZATION," 2021. [Online]. Available: www.sick.com/MLS

[31] "DTI421 - cabeça de leitura e gravação RFID HF," *IFM*. https://www.ifm.com/pt/pt/product/DTI421

[32] SICK, "RFU620-10100 RFU62x," 2021. [Online]. Available: www.sick.com/RFU62x

[33] Pepperl+Fuchs, “RFID Read/Write Heads,” *Pepperl+Fuchs*. Oct. 2021. [Online]. Available: https://www.pepperl-fuchs.com/global/en/classid_7478.htm

[34] “O1D100 - Sensor óptico de distância,” *IFM*. [Online]. Available: https://www.ifm.com/pt/pt/product/O1D100

[35] SICK, “SG2-AAA00011IA0000,” 2021. [Online]. Available: www.sick.com/scanGrid2

[36] SICK, “TIM351-2134001,” 2021. [Online]. Available: www.sick.com/TiM3xx

[37] Pepperl+Fuchs, “3-D LiDAR sensor OMD10M-R2300-B23-V1V1D-4S,” *Pepperl+Fuchs*. Oct. 2021. [Online]. Available: https://www.pepperl-fuchs.com/global/en/classid_53.htm

Anexo A. Configuração das curvas

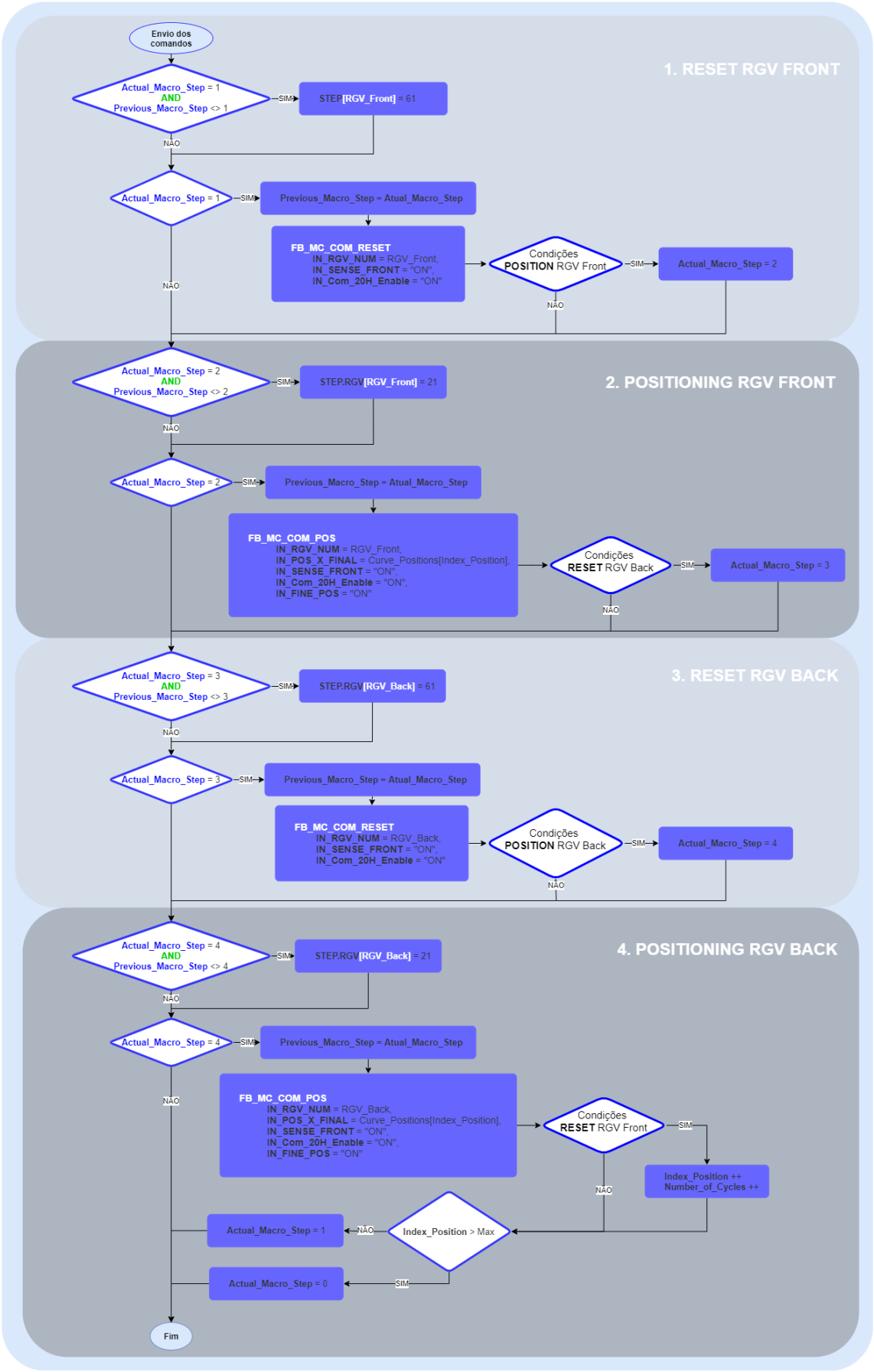
Curva	Segmento
1	6
2	6
3	8
4	8
5	9
6	9
7	11
8	12
9	12
10	13
11	15
12	15
13	16
14	16
15	16
16	16
17	17
18	19
19	20
20	20
21	1
22	1
23	25
24	25

Segmento	Número de curvas
1	**2**
2	0
3	0
4	0
5	0
6	**2**
7	0
8	**2**
9	**2**
10	0
11	**1**
12	**2**
13	**1**
14	0
15	**2**
16	**4**
17	**1**
18	0
19	**1**
20	**2**
21	0
22	0
23	0
24	0
25	**2**
26	0
27	0
28	0
29	0
30	0
31	0
32	0
33	0
34	0
35	0
36	0
Total	24

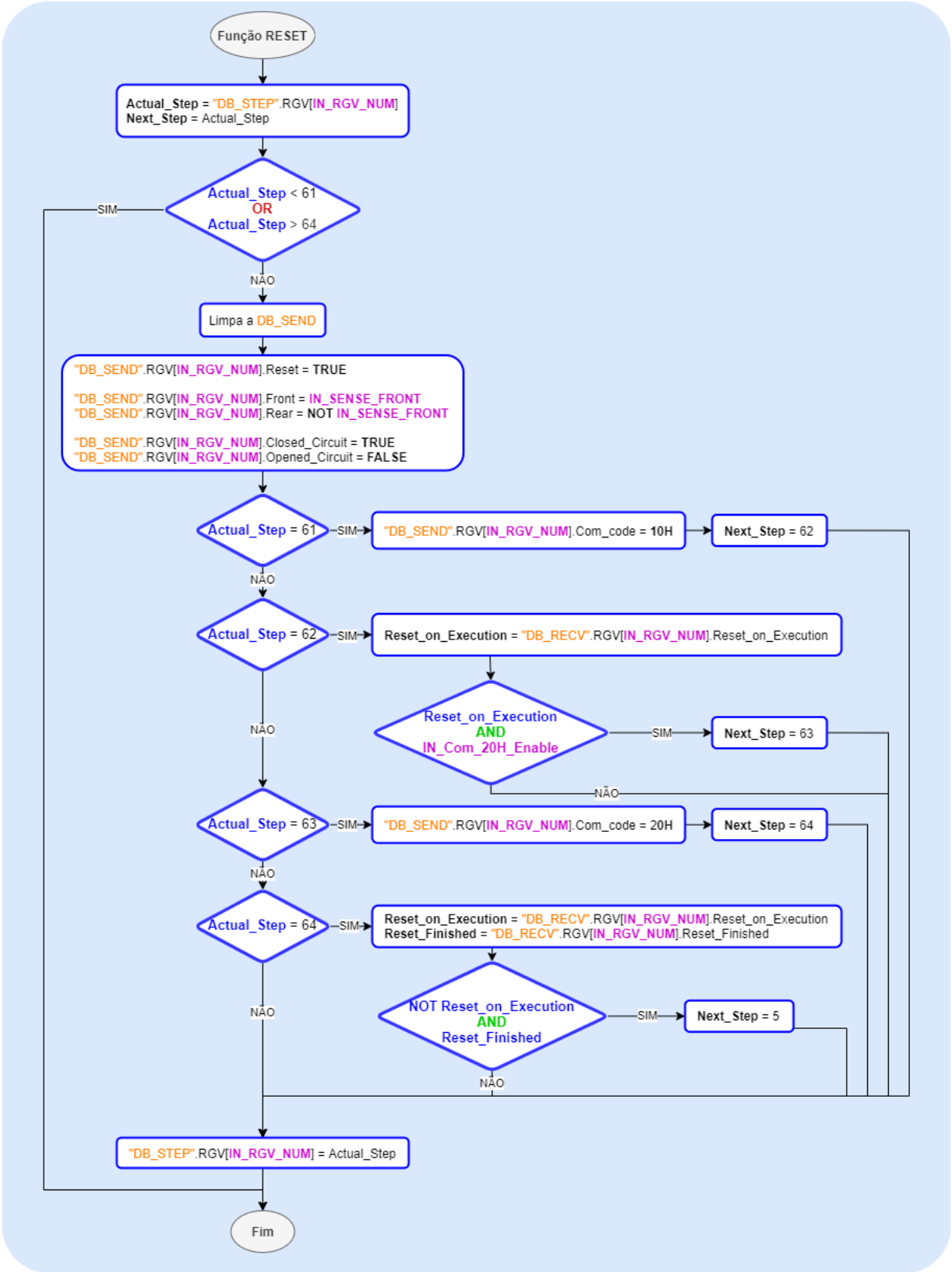
Anexo B. Configuração das agulhagens

Agulhagem	IN_1	IN_2	OUT_1	OUT_2
1	**1**	-	**2**	**23**
2	**22**	**23**	**1**	-
3	**2**	**24**	**3**	-
4	**21**	-	**22**	**24**
5	**3**	**25**	**4**	-
6	**5**	-	**6**	**26**
7	**4**	-	**5**	**25**
8	**20**	**26**	**21**	-
9	**6**	**27**	**7**	-
10	**19**	-	**20**	**27**
11	**18**	-	**19**	**28**
12	**17**	-	**18**	**29**
13	**7**	-	**8**	**30**
14	**30**	**28**	**31**	-
15	**31**	**29**	**32**	-
16	**16**	**32**	**17**	-
17	**8**	**33**	**9**	-
18	**15**	-	**16**	**33**
19	**9**	-	**10**	**34**
20	**14**	**34**	**15**	-
21	**10**	**35**	**11**	-
22	**13**	-	**14**	**35**
23	**12**	-	**13**	**36**
24	**11**	**36**	**12**	-

Anexo C. Fluxograma do envio de comandos



Anexo D. Fluxograma da função FB_MC_COM_RESET



Anexo E. Fluxograma da função FB_MC_COM_POS