

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
CAMPUS SANTA MÔNICA

Ruchele Pavanelli de Souza

Integração de uma impressora 3D com função contínua na
produção e logística utilizando a etiquetagem RFID

UBERLÂNDIA
2022

RUCHELE PAVANELLI DE SOUZA

Integração de uma impressora 3D com função contínua na produção e logística utilizando a etiquetagem RFID

Trabalho de Conclusão de Curso da Engenharia de Controle e Automação da Universidade Federal de Uberlândia - UFU - Câmpus Santa Mônica, como requisito para a obtenção do título de Graduação em Engenharia de Controle e Automação

Universidade Federal de Uberlândia – UFU

Faculdade de Engenharia Elétrica

Orientador: Prof. Renato S. Carrijo

UBERLÂNDIA

2022

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

RUCHELE PAVANELLI DE SOUZA

Esta Monografia foi julgada adequada para a obtenção do título de Bacharel em Engenharia de Controle e Automação, sendo aprovada em sua forma final pela banca examinadora:

Orientador(a): Prof. Dr. Renato S. Carrijo
Universidade Federal de Uberlândia - UFU

Prof. Dr. Fábio Vincenzi Romualdo da Silva
Universidade Federal de Uberlândia - UFU

Prof. Dr. Josué Silva de Moraes
Universidade Federal de Uberlândia - UFU

Uberlândia, 22 de março de 2022

*Dedico este trabalho aos meus pais,
José Maria e Marilda, e especialmente
ao meu avô Ivo (In memoriam),
grandes colaboradores e incentivadores
ao longo da minha jornada.*

Agradecimentos

Agradeço a Deus, primeiramente, que me deu força para concluir esta etapa de minha vida.

Gostaria de agradecer a minha família e amigos. Especialmente meus pais que sempre me apoiaram com tudo o que eu precisava durante a minha vida e a minha irmã por me ouvirem em momentos difíceis. Aos meus tios, especialmente minha tia Mônica, a qual serviu-me de exemplo para o ingresso neste curso. Gostaria também de agradecer ao meu avô (In Memoriam) por ter sempre me apoiado e mostrado orgulho ao acompanhar meu progresso.

A todos os amigos que, direta ou indiretamente, participaram da minha formação, meu muito eterno agradecimento.

Agradeço ao INSA Strasbourg, instituição que me acolheu por 2 anos na França, durante meu intercâmbio, e que me proporcionou todo o suporte necessário para que eu finalizasse o curso de maneira efetiva mesmo estando fora do meu país de origem.

À instituição de ensino – FEELT, juntamente com a UFU, que me proporcionaram a oportunidade de possuir um ensino superior e a expansão de meus horizontes.

Agradeço a todos os professores por me proporcionarem o conhecimento, não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender. A palavra mestre nunca fará justiça aos professores dedicados aos quais sem nominar terão os meus eternos agradecimentos.

Ao meu orientador Renato, pelas suas correções e incentivos.

*“Por vezes sentimos que aquilo
que fazemos não é senão uma gota no oceano.
Mas sem ela o oceano seria menor.”,
(Madre Teresa de Calcutá)*

Resumo

Com o aumento da automatização de processos na indústria, a utilização de diferentes tecnologias e recursos ocorreu de forma espontânea. Dentre as diversas áreas de implementação de novas tecnologias, logística e triagem são duas das zonas mais afetadas por possíveis erros humanos, necessitando-se assim de uma mudança nos processos utilizados atualmente de forma a torná-los menos manuais. Este tipo de automação é proposto pela 4ª Revolução Industrial, chamada Indústria 4.0 e permite que todo o sistema funcione corretamente sem a força de trabalho humana e por consequência com menor susceptibilidade a erros. Neste contexto, este projeto propôs uma experiência de laboratório usando um controlador lógico programável (CLP), sensores, etiquetas RFID, braços robóticos e um veículo automaticamente guiado a fim de melhorar a qualidade do processo de seleção e triagem de peças. Usando um CLP, foi possível controlar a retirada e a escrita/leitura de cada peça feita pela impressora 3D, estabelecer a rota do veículo até a área de triagem e fiabilizar a triagem de acordo com a informação contida em cada peça impressa. O objetivo principal foi reduzir o grau de intervenção da mão-de-obra humana nas diferentes etapas de produção. Este método proporcionou uma queda nos erros de triagem e também permitiu que todo o processo ocorra de forma autônoma. Ao interligar os diferentes elementos do sistema, criou-se uma rede fechada e totalmente independente capaz de garantir a passagem de todas as informações essenciais desde a impressão até a triagem.

Palavras-chave: automação, indústria 4.0, CLP, sensores, robôs, triagem, veículo autônomo.

Abstract

With the increasing automation of processes in industry, the use of different technologies and resources has occurred spontaneously. Among the various areas of implementation of new technologies, logistics and sorting are two of the areas most affected by possible human errors, thus requiring a change in the processes currently used in order to make them less manual. This type of automation is proposed by the 4th Industrial Revolution, called Industry 4.0, and allows the whole system to work correctly without human labor and therefore with less susceptibility to errors. In this context, this project proposed a laboratory experiment using a programmable logic controller (PLC), sensors, RFID tags, robotic arms, and an automatically guided vehicle in order to improve the quality of the parts selection and sorting process. Using a PLC, it was possible to control the picking and writing/reading of each part made by the 3D printer, establish the route of the vehicle to the sorting area, and make the sorting reliable according to the information contained in each printed part. The main objective was to reduce the degree of human labor intervention in the different production steps. This method provided a drop in sorting errors and also allowed the whole process to take place autonomously. By interconnecting the different elements of the system, a closed and totally independent network was created that ensures the passage of all essential information from printing to sorting.

Keywords: automation, 4.0 industry, PLC, RFID sensors, robots, AGV.

Lista de ilustrações

Figura 1 – Adaptada de (Volpato 2017) Representação das principais etapas do processo de AM	7
Figura 2 – Adaptada de (Volpato 2017) - Diferentes área de Aplicação da AM, sendo a produção de peças funcionais e encaixes/montagens as duas áreas com aproximadamente metade da utilização do processo	8
Figura 3 – Adaptada de (LUGLI) - Sistema tradicional de um CLP, onde é possível observar que o sistema central é responsável por receber todas as conexões. Conseqüentemente, têm-se um elevado número de cabos e uma complexidade na resolução de possíveis problemas	9
Figura 4 – Adaptada de (LUGLI) - Sistema atual com rede industrial onde é possível observar um menor número de cabos diretamente ligados à unidade central além de organizar e segmentar o fluxo de dados	10
Figura 5 – Adaptada de (Miller 2009) - Encapsulamento do UDP	13
Figura 6 – Adaptada de (Miller 2009) - Cabeçalho do UDP onde é possível observar dois campos de 16 bits que são responsáveis por identificar a origem e o destino da informação, além de informações como tamanho, o dado em questão e o checksum	14
Figura 7 – Adaptada de (Silva 2016) - Etapas de um ciclo SCAN, onde recebe-se os sinais através dos periféricos de entrada, administra-se as funções executando a lógica de programação e posteriormente gera-se um sinal para ser enviado aos periféricos de saída	16
Figura 8 – Adaptada de (Franchi e Camargo) - Exemplo de CLP compacto, onde o usuário tem acesso apenas ao conectores de entrada e saída	17
Figura 9 – Adaptada de (Franchi e Camargo) - Arquitetura de CLP modular, onde é possível observar a presença de módulos variados responsáveis por executarem diferentes funções	18
Figura 10 – Adaptada de (Franchi e Camargo) - Etapas de um ciclo SCAN, onde lê-se as entradas, executa-se a lógica de programação definida pelo usuário e finalmente atualiza-se as saídas	19
Figura 11 – Adaptada de (Franchi e Camargo) - Fluxo de execução de um PLC	20
Figura 12 – Adaptada de (Silva 2016) - Exemplos de entradas e saídas de um PLC	21
Figura 13 – Adaptada de (Silva 2016) - Tabela com os diferentes tipos de representação de variáveis	21
Figura 14 – Adaptada de (ROGGIA L 2016) - Exemplo de utilização da linguagem Ladder	22

Figura 15 – Adaptada de (ROGGIA L 2016) - Elementos principais do diagrama de contatos	23
Figura 16 – Adaptada de (ROGGIA L 2016) - Funções avançadas do diagrama de contatos	24
Figura 17 – Funcionamento da tecnologia RFID	24
Figura 18 – Adaptada de (Siemens 2012) - PLC S7 1200 utilizado no projeto	30
Figura 19 – Adaptada de (Siemens 2009) Interface do software TIA Portal	31
Figura 20 – Adaptada de (IFM - AL1302) - Layout do AL1302 Mestre	32
Figura 21 – Adaptada de (IFM - O5D100) - Layout do sensor fotoelétrico O5D100 .	33
Figura 22 – Adaptada de (IFM - OGD592) Layout do sensor fotoelétrico OGD592 .	33
Figura 23 – Adaptada de (Shenzhen Yuejiang Technology Co 2017) - Dobot Magician	34
Figura 24 – Adaptada de (Shenzhen Yuejiang Technology Co 2017) - Grau de rotação do Dobot	35
Figura 25 – Adaptada de (SDP Mini Manual) - AGV SDP Mini	36
Figura 26 – Adaptada de (Siemens 2020) - MDS 400 - Tag em formato de cartão .	37
Figura 27 – Adaptada de (Siemens 2020) - Informações técnicas sobre a Tag	37
Figura 28 – Adaptada de (IFM - DTI513) - Layout do DTI513	37
Figura 29 – Adaptada de (IFM - DTI513) - Informações técnicas sobre o sensor . .	38
Figura 30 – Adaptada de (Tiertime - X5) - Tiertime X5	39
Figura 31 – Modo de Funcionamento “Impressão contínua”	39
Figura 32 – Modo de Funcionamento “Impressão contínua”	40
Figura 33 – Impressora, esteira, robô 1 e veículo autônomo	40
Figura 34 – Diagrama de funcionamento completo do projeto	41
Figura 35 – Rede de comunicação entre o mestre, os sensores e o CLP	42
Figura 36 – Adaptada de (Shenzhen Yuejiang Technology Co 2017) - Interface do software utilizado para obter as posições do Dobot: DobotStudio	43
Figura 37 – Adaptada de (Shenzhen Yuejiang Technology Co 2017) - Função Teaching e Playback utilizada para obter as coordenadas exatas de cada posição do robô	44
Figura 38 – Servidor responsável pela comunicação com a impressora	45
Figura 39 – Página demo.html com as funções reconhecidas pela impressora	46
Figura 40 – Rede entre o PLC e o Mestre	49
Figura 41 – Configuração de portas do Mestre AL1302 contendo os 4 sensores utilizados no projeto: Port 1 - 1°sensor de presença peça, Port 2 - 1°sensor RFID responsável por escrever na etiqueta, Port 3 - 2°sensor de presença peça e Port 4 - 2°sensor RFID responsável por ler a etiqueta	49

Figura 42 – Função de detecção de peça para o sensor 1 onde IN1 corresponde ao valor de luminosidade lido pelo 1º sensor em tempo real e IN2 é o valor de referência usado para fazer a comparação. Se o valor em IN1 é menor ou igual à IN2, uma variável de controle é acionada para promover o funcionamento de outros processos	50
Figura 43 – Função de detecção de peça para o sensor 2 - Função de detecção de peça para o sensor 2 onde IN1 corresponde ao valor de luminosidade lido pelo 2º sensor em tempo real e IN2 é o valor de referência usado para fazer a comparação. Se o valor em IN1 é menor ou igual à IN2, uma variável de controle é acionada para promover o funcionamento de outros processos	50
Figura 44 – Função para inicializar a comunicação com o Dobot	51
Figura 45 – Rede 9: Se sensor 1 ativado (placa preta sob o mesmo), a variável de controle para ativar a esteira é ativada, a variável para ativar o movimento do robô é ativada e a variável de controle para parar a esteira é desativada	51
Figura 46 – Detecção de uma peça pelo sensor 1	52
Figura 47 – Rede 10: se a variável de controle responsável por validar a demanda de SetHoçe do robô estiver ativada, um pedido de SetHome será enviado ao robô e a mesma variável de controle é passada a 0	52
Figura 48 – Posição SetHome do robô (à espera da peça)	53
Figura 49 – Rede 12: Se sensor 2 ativado (placa preta sob o mesmo), a variável de controle para parar a esteira é ativada e variável de controle para acionar a esteira é desativada	53
Figura 50 – Detecção de uma peça pelo sensor 2	54
Figura 51 – Rede 13: Se a variável de controle de parada da esteira estiver ativada, um comando é enviado ao robô para parar o motor responsável pelo comando da esteira e a simultaneamente a variável de controle para acionar os movimentos do robô é acionada	54
Figura 52 – Começo da sequência de movimentos depois do SetHome	55
Figura 53 – Rede 14: Se a variável de controle para acionar os movimentos do robô estiver ativada, primeiramente o programa aguarda 1s e posteriormente as posições desejada em x,y e x são enviadas para as variáveis do robô através do protocolo UDP e finalmente, ativa-se uma variável intermediária de controle para poder ativar as próximas mudanças de posição e desativa-se a variável de controle responsável por acionar o primeiro movimento do robô	55

Figura 54 – Rede 21 - A cada movimento realizado (utilizando as variáveis auxiliares), o comando SetPTPCmd é acionado. Tal comando é necessário para ativar o movimento do Dobot e escolher seu modo de funcionamento.	56
Figura 55 – Detecção da peça pelo sensor que encerra o funcionamento da esteira, Dobot recupera a placa + peça e começa a sequência de movimentos até depositar a peça sobre o AGV.	57
Figura 56 – Bloco-função DTIxxx para estabelecer a comunicação com o AL1302	58
Figura 57 – Adaptada de (IFM 2018) - Parâmetros de entrada do bloco - Parte 1	59
Figura 58 – Adaptada de (IFM 2018) - Parâmetros de entrada do bloco - Parte 2	59
Figura 59 – Exemplo de descoberta do parâmetro HW ID	60
Figura 60 – Adaptada de (IFM 2018) - Parâmetros de saída do bloco	61
Figura 61 – Bloco adaptado segundo as necessidades do projeto	61
Figura 62 – Adaptada de (IFM 2018) - Armazenamento da informação contida no UID na memória da TAG	62
Figura 63 – Simulação e visualização da tabela de variáveis se modo de operação = 1, é possível observar que o sensor está em modo de execução, o modo de funcionamento atual é o de número 1, uma etiqueta foi detectada no raio do sensor, é possível saber o tamanho dos dados contidos na etiqueta e o valor armazenado na posição 4 do vetor de dados da etiqueta é 1653	62
Figura 64 – Adaptada de (IFM 2018) - Mapa de informações do modo de escrita automática	63
Figura 65 – Simulação e visualização da tabela de variáveis se modo de operação = 2, é possível observar que o sensor está em modo de execução, o modo de funcionamento atual é o de número 2, uma etiqueta foi detectada no raio do sensor e a leitura foi terminada pois a variável Done está ativada	64
Figura 66 – Adaptada de (IFM 2018) - Mapa de informações do modo de escrita automática	65
Figura 67 – Atribuição do modo de operação 3 ao sensor	65
Figura 68 – Simulação e visualização da tabela de variáveis se modo de operação = 3	66
Figura 69 – Interface SLATEC RobotStudio	67
Figura 70 – Painel com comandos para controlar a impressora	69
Figura 71 – Digital Twin - Visualização completa do projeto	69
Figura 72 – Exemplo de barra utilizada no projeto	70
Figura 73 – Mesa do projeto no CAD	70
Figura 74 – AGV no CAD	71
Figura 75 – Sensores IFM no CAD	71
Figura 76 – Trilho Linear no CAD	71
Figura 77 – Impressora 3D no CAD	72

Lista de tabelas

Tabela 1 – Adaptada de (Frequency Table) - Tabela de frequências e suas aplicações	25
Tabela 2 – Adaptada de (Shenzhen Yuejiang Technology Co 2017) - Tabela de características técnicas do Dobot	34
Tabela 3 – Adaptada de (Shenzhen Yuejiang Technology Co 2017) - Tabela de características físicas do Dobot	34
Tabela 4 – Adaptada de (Tiertime - X5) - Informações técnicas sobre a impressora	38

Lista de abreviaturas e siglas

AGV - Autonomous guide vehicle

API - Application Programming Interface

AM - Addictive Manufacturing

CAD - Computer-aided Design

CLP - Controlador lógico programável

E/S - Entrada/Saida

GUID - Globally Unique Identifier

IP - Internet Protocol

IPP - Internet Printing Protocol

LD - Ladder Diagram

MMS - Manufacturing Messaging Specifications

RFID - Radio Frequency Identification System

SFC - Sequential Function Chart

ST - Structured Text

TCP - Transmission Control Protocol

TIE - Tabela Imagem das Entradas

TIS - Tabela Imagem das Saidas

UDP - User Datagram Protocol

UHF - Ultra High Frequency

Sumário

	Resumo	vii
	Lista de Figuras	ix
	Lista de Tabelas	xiii
	Lista de Abreviações	xv
1	INTRODUÇÃO	1
1.1	Justificativa	1
1.2	Objetivo	2
1.3	Organização	3
2	REFERENCIAL TEÓRICO	4
2.1	Impressão 3D	4
2.1.1	Tecnologias de Impressão 3D	5
2.1.1.1	Fabricação por Filamento Fundido (FFF)	5
2.1.1.2	Estereolitografia (SLA)	5
2.1.1.3	Sinterização Seletiva a Laser (SLS)	6
2.2	Automação Industrial	8
2.3	Sistemas	8
2.4	Protocolos industriais	8
2.4.1	Vantagens das redes industriais	9
2.4.2	Tecnologia Fieldbus	10
2.4.3	Profibus	10
2.4.3.1	Arquitetura de rede	11
2.4.3.2	PROFIBUS-DP	11
2.4.3.3	PROFIBUS-PA	12
2.4.3.4	PROFINET	12
2.5	Protocolo UDP	13
2.5.1	Cabeçalho	13
2.6	Tecnologia IO-Link	14
2.6.1	Funcionamento	15
2.6.2	Vantagens	15
2.7	Controlador Lógico Programável (CLP)	16
2.7.1	Arquitetura	16
2.7.2	Tipos de CLP	16

2.7.2.1	Compactos	17
2.7.2.2	Modulares	17
2.7.3	Modos de operação de um CLP	18
2.7.3.1	Modo Programação	18
2.7.3.2	Modo Execução	18
2.7.4	Tipos de Entradas e Saídas	21
2.7.5	Tipos de variáveis	21
2.8	Linguagem LADDER	22
2.9	Tecnologia RFID	24
2.9.1	Tag ou etiqueta	25
3	MATERIAS E MÉTODOS	27
3.1	Especificações e limitações	27
3.1.1	Etiquetagem RFID e transporte	27
3.1.1.1	Especificações	27
3.1.1.2	Principais desafios	28
3.1.2	Veículo automaticamente guiado (AGV)	28
3.1.2.1	Especificações	28
3.1.2.2	Principais dificuldades	29
3.1.3	Classificação dos itens	29
3.1.3.1	Especificações	29
3.1.3.2	Principais dificuldades	29
3.2	Materiais utilizados	30
3.2.1	CLP Siemens 1200 1214C DC/DC	30
3.3	AL1302 Mestre	31
3.4	O5D100 - Sensor de distância fotoelétrico	32
3.5	OGD592 - Sensor de distância fotoelétrico	33
3.6	DOBOT	33
3.6.1	Características Técnicas	34
3.6.2	Características Físicas	34
3.6.3	Área de Trabalho Dobot	35
3.7	SDP Mini RPlidar Experimental 2WD Robot Platform	35
3.8	Etiquetas RFID	36
3.8.1	MDS D400 - Tag	36
3.8.2	DTI513 - Leitura e escrita RFID	37
3.8.3	Tiertime X5	38
3.9	Metodologia	39
3.9.1	Sistema original	39
3.10	Proposição final do projeto	40
3.11	Comunicação TAG - CLP	41

3.11.1	Definição das posições e das ações do robô	42
3.11.2	Lógica de envio das posições ao robô	44
3.12	Aplicativo para impressão das peças	44
4	RESULTADOS E PROGRAMAÇÃO	48
4.1	Recuperação e deslocamento da placa	48
4.2	Configuração e programação Tia Portal	48
4.3	Sensores de distância	49
4.4	Comunicação Dobot	50
4.5	Etiquetas RFID	57
4.5.1	Leitura do UID da TAG	61
4.5.2	Leitura dos dados da TAG de forma automática	63
4.5.3	Escrita dos dados da TAG de forma automática	64
4.6	AGV - Veículo Autônomo	66
4.7	Software para envio de impressões	67
4.7.1	Resultados	68
4.8	Digital Twin	69
4.8.1	Visão Geral do Projeto	69
4.8.2	Itens desenhados separadamente	70
4.8.3	Conexão com NX e Tia Portal	72
4.8.3.0.1	OPC UA	72
4.8.3.0.2	Sinais	72
4.8.3.0.3	Equações de Runtime	73
4.8.3.0.4	Operações	73
5	DESAFIOS ENCONTRADOS E MELHORIAS	74
5.0.1	Melhorias gerais do código TIA Portal	74
5.0.2	Dobot	74
5.0.3	IFM	74
5.0.4	NX Digital Twin	75
5.0.5	AGV	75
6	CONCLUSÃO	76
	REFERÊNCIAS	78
	ANEXOS	81
	ANEXO A – TUTORIAL UDP	82

ANEXO B – TUTORIAL LR DEVICE	100
ANEXO C – TUTORIAL RFID	104
ANEXO D – TUTORIAL MODIFICAÇÕES UDP	108
ANEXO E – TUTORIAL DOBOT - WIRESHARK	112
ANEXO F – TUTORIAL BIBLIOTECA IFM	123

1 Introdução

O conceito "Indústria 4.0" (Sung 2018) está cada vez mais presente nas fábricas. O conceito de Indústria 4.0 é o nome dado à tendência atual de automação e intercâmbio de dados nas tecnologias de fabricação. Tal conceito inclui sistemas ciberfísicos, a Internet das coisas, a computação em nuvem e a computação cognitiva. A Indústria 4.0 é comumente chamada de "Quarta Revolução Industrial", que tem sucesso e melhora diretamente a indústria 3.0 e sua automação de máquinas.

Através do uso da Internet das Coisas (Li et al. 2020) e de sistemas ciberfísicos, redes virtuais (Sun 2020) utilizadas para controlar objetos físicos, a "fábrica inteligente" (Malik et al. 2020) é caracterizada pela comunicação contínua e instantânea entre as diversas ferramentas e estações de trabalho integradas nas cadeias de produção e fornecimento. O uso de sensores de comunicação proporciona à ferramenta de produção uma capacidade de autodiagnóstico e assim permite seu controle remoto, bem como sua melhor integração no sistema de produção global.

Este projeto tem como intuito utilizar o sistema de etiquetagem RFID (Duan e Cao 2020) para triagem de peças que são produzidas por uma impressora 3D de modo contínuo. O transporte e posteriormente a triagem das mesmas é feito por dois braços robóticos e por um AGV. Através deste projeto, pode-se mostrar como é possível construir um sistema de produção totalmente independente e automatizado.

1.1 Justificativa

A Indústria 4.0 está revolucionando a forma como as empresas fabricam, melhoram e distribuem seus produtos. Em suma, o objetivo é aumentar a qualidade do produto e paralelamente diminuir os custos de produção. Estima-se que nos próximas décadas, 80% dos empregos vão deixar de existir por causa deste novo conceito que se faz cada vez mais presente. No entanto, outros tipos de mão-de-obra irão consequentemente surgir como forma de atender tais mudanças. Tal tipo de mão-de-obra deve ser especializada e técnica. O objetivo é que o indivíduo seja capaz de estabelecer uma relação de cooperação entre a tecnologia de informação e os processos de produção.

Quando o termo "Indústria 4.0" foi introduzido, em 2012, o Brasil ocupava uma posição relativamente importante no ranking de países que utilizavam a manufatura: o 5 lugar, sendo que Estados Unidos, China e Alemanha ocupavam os 3 primeiros lugares. Porém, em 2016, o Brasil já ocupava o 29 lugar. Tal queda dá-se as limitações para adquirir equipamentos com recursos avançados tendo como mão-de-obra pessoas não especializadas

para utilizá-los.

Dentro do contexto da necessidade da introdução da tecnologia nos ambientes de trabalho, este projeto foi pensado para promover a autonomia de um processo de produção que é utilizado em diversas empresas, a impressão 3D.

A parte inovadora deste projeto é a junção de diferentes tecnologias num mesmo ambiente sendo controladas por uma mesma central, capaz de gerenciar e dar autonomia por completo para a execução da tarefa. Ou seja, os braços robóticos, os sensores e o AGV estarão conectados através de um CLP. Tal trabalho é de considerável relevância uma vez que este diminui a necessidade de mão de obra humana nas etapas de fabricação e transporte de um produto, diminuindo assim a possibilidade de erros causados por falha humana. Tais erros são responsáveis por gerarem a maior parte (80%) das falhas nas linhas de produção atualmente. Portanto, o objetivo é conseguir diminuir a porcentagem de erros e problemas dentro de uma linha de produção.

A proposição deste tema cria uma área de exploração e desenvolvimento não apenas com relação ao aumento dos componentes interconectadas, mas também com a aplicação deste tipo de tecnologia em diferentes ramos e setores da indústria atual. Portanto, tal trabalho tem como pontos positivos:

- Diminuição de falhas humanas;
- Disseminação dos conceitos para diferentes áreas da indústria;
- Interconexão de objetos através de um único componente;
- Desenvolvimento de procedimentos para que diferentes componentes se interconectem e trabalhem simultaneamente sem interferências;
- Desenvolvimento de mão de obra especializada e qualificada para atender a necessidade do manuseio de novas tecnologias com processos automatizados.

1.2 Objetivo

Portanto, o objetivo deste projeto é reunir todas as diferentes etapas do processo de produção e triagem de peças utilizando apenas um núcleo de processamento: o CLP. Desta forma, nenhum trabalho manual será requisitado e pode-se garantir uma total autonomia na linha de produção. Além disso, a centralização dos componentes em um único centro é vantajosa pois, em caso de problemas ou necessidade de alteração de alguma ação específica, a mudança será feita em apenas um lugar, não necessitando de ajustes complementares em outras centrais ou componentes específicos.

Como objetivos secundários, por um lado, a ideia é reduzir a utilização de conexões físicas

entre os diferentes equipamentos. Logo, protocolos de comunicação sem fio tais como UDP e TCP são considerados no estudo a fim de serem implementados nos robôs e no AGV. Por outro lado, busca-se qualificar a mão-de-obra de forma a torná-la especializada e técnica para o bom manuseio de novas tecnologias.

1.3 Organização

O presente trabalho está organizado da seguinte forma:

Capítulo 2: Referencial teórico contendo as principais definições e os principais conceitos presentes no projeto

Capítulo 3: A metodologia utilizada assim como os materiais escolhidos

Capítulo 4: Resultados obtidos

Capítulo 5: Desafios encontrados e melhorias para o futuro

Capítulo 6: Conclusão

2 Referencial Teórico

2.1 Impressão 3D

A impressora 3D é um dispositivo projetado para imprimir objetos com vários materiais como metal, polímero, cera, amido ou argila. O processo é chamado manufatura aditiva e consiste em adicionar material líquido ou em pó e empilhar camadas. A AM (ou manufatura aditiva) pode ser definida como um processo de fabricação por meio da adição sucessiva de material na forma de camadas, com informações obtidas diretamente de uma representação geométrica computacional 3D do componente. Normalmente, tal modelo geométrico é originado de um sistema CAD (computer-aided design). Tal processo permite a fabricação de componentes físicos a partir de diferentes tipos de materiais (Volpato 2017). Tal processo está em alta crescente no cenário atual, porém, como toda nova tecnologia contém vantagens e inconvenientes. Como vantagem, tem-se:

- Maior liberdade no design: criação de geometrias mais complexas e detalhadas as quais não podem ser produzidas por outros métodos de fabricação;
- Possibilidade de personalização das peças: cada item pode ser personalizado de acordo com as necessidades do projeto e sem afetar os custos de fabricação;
- Grande variedade de materiais: Os materiais mais comuns atualmente são os diferentes tipos de plásticos. O uso de metais também pode ser encontrado em algumas aplicações na indústria. Em suma, as peças podem ter diferentes características, resultando em produtos finais com propriedades únicas e adequadas a aplicações específicas;
- Custos iniciais baixos: enquanto que na fabricação tradicional é necessário um molde exclusivo os quais são caros, a impressão 3D não precisa de ferramentas especializadas;
- Prototipagem de baixo custo e retorno rápido.

Como inconveniente, tem-se:

- Alto investimento: para cobrir o investimento em impressoras 3D e seus insumos, é necessário fabricar-se em grandes quantidades;
- Custos de manutenção: a impressão 3D exige conhecimento técnico para um melhor resultado além de possuir uma manutenção que deve ser contínua;
- Uso de softwares específicos.

2.1.1 Tecnologias de Impressão 3D

2.1.1.1 Fabricação por Filamento Fundido (FFF)

Tal processo caracteriza-se pela deposição de materiais em camadas para construção do objeto modelado. Como característica, têm-se uma baixa complexidade e baixo preço da matéria prima, sendo acessível. As máquinas que utilizam a tecnologia são as mais comuns, utilizando polímeros variados (ABS, PLA, FLEX, PETG) com a possibilidade de misturar materiais (metais, cerâmicos, madeira, etc) (Tecnologias de Impressão). Como vantagem, têm-se:

- Baixa complexidade para operação e manutenção;
- Máquinas e materias primas mais acessíveis;
- Versatilidade.

Como desvantagem, têm-se:

- As camadas construídas ficam aparentes;
- Limitações em impressões coloridas.

2.1.1.2 Estereolitografia (SLA)

Usa como materia-prima uma resina líquida foto sensível. Por meio da aplicação de um laser UV, a resina se acumula na base. Após a cura dessa resina, a base da impressora levanta uma altura compatível com a espessura da camada e o laser percorre o caminho para construção da próxima camada, dando sequência a leitura do objeto fatiado (Tecnologias de Impressão). Como vantagem, têm-se:

- Alta resolução e precisão na construção dos objetos, ideal para objetos pequenos e superdetalhados;
- Máquinas mais compactas;
- Estrutura de funcionamento complexa porém com operação e manutenção simples.

Como desvantagem, têm-se:

- Materia-prima e máquinas com custos altos;
- Material com risco químico elevado.

2.1.1.3 Sinterização Seletiva a Laser (SLS)

Trabalha com matéria-prima em pó. A aglomeração e endurecimento do material também é feito por meio da aplicação de um laser UV, no entanto o processo é bastante diferente da SLA. Por meio de um rolo de material é despejado na mesa de impressão uma fina camada de pó, em seguida um laser percorre a área de impressão desenhando a camada resultante do fatiamento do objeto 3D, pela parte superior da máquina. A base da impressora abaixa uma altura compatível com a espessura da camada, o rolo despeja mais uma camada de pó e o laser constrói a próxima camada do objeto, dando sequência a fabricação (Tecnologias de Impressão). Como vantagem, têm-se:

- Peças possuem maior resistência mecânica e alta pressão;
- Não há necessidade de material de suporte;
- Fidelidade em impressões coloridas.

Como desvantagem, têm-se:

- Objeto final poroso logo necessidade de acabamento;
- Matéria-prima, operação e manutenção complexas.

Logo, ao comparar as diferentes tecnologias de impressão e suas características, é possível afirmar que pelo custo-benefício, a FFF é a mais utilizada e mais adequada por ser versátil e ter baixa complexidade para operação.

O processo inicia-se com o modelo 3D da peça sendo fracionado com o objetivo de obter-se "curvas de nível" 2D as quais definirão onde o material será ou não adicionado. Desta forma, a peça física é gerada por meio do empilhamento sequencial das camadas. As etapas do processo são:

- Modelagem da peça a fim de gerar-se um modelo geométrico 3D ;
- Obtenção do modelo geométrico em um formato específico para AM, no caso do projeto, o tipo STL. Como o formato de arquivo de conversão de dados entre o modelo CAD e o equipamento de prototipagem rápida, STL é o padrão industrial de fato no campo da prototipagem rápida. No arquivo STL, cada faceta triangular é representada pelos três vértices do triângulo e uma unidade vetorial normal. O vetor normal se estende do interno para o externo. Os três vértices e o vetor normal da faceta satisfazem a regra da mão direita (Pan, Chen e Chen 2014) ;
- O planejamento do processo para a fabricação por camada ;

- A fabricação da peça no equipamento de AM ;
- O pós-processamento que varia bastante de acordo com a tecnologia utilizada.

Figura 1 – Adaptada de (Volpato 2017) Representação das principais etapas do processo de AM

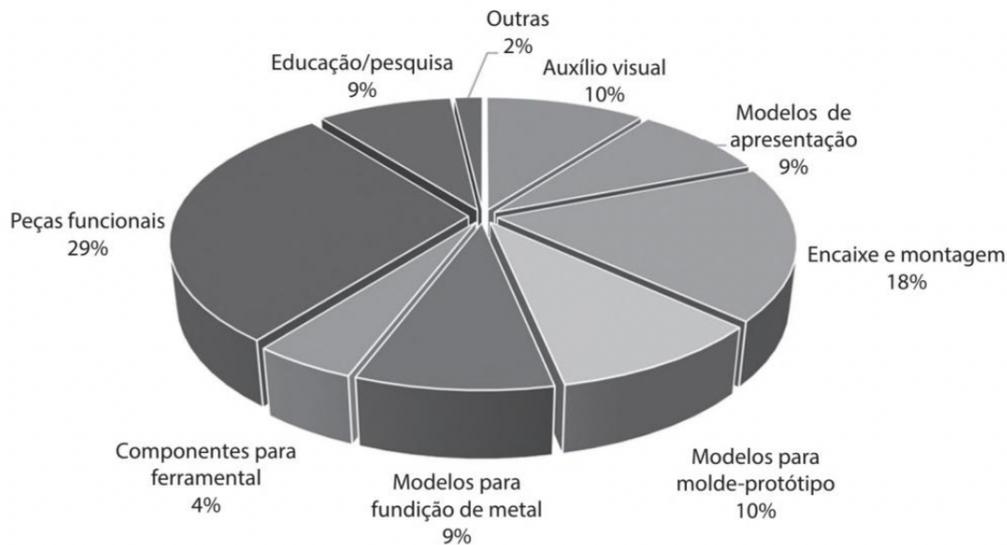


A facilidade de automatização de forma a minimizar a intervenção do operado é uma característica importante para sua utilização.

Como principais vantagens dessas tecnologias (Volpato 2017), é possível citar:

- Liberdade geométrica independentemente da complexidade da peça ;
- Pouco desperdício de material e utilização eficiente de energia ;
- Não é necessário dispositivos de fixação ;
- Não é necessária a troca de ferramentas durante a fabricação do componente ;
- A fabricação da peça é realizada em um único equipamento do início ao fim ;
- Com relação à trajetória de ferramentas, não é preciso cálculos complexos. Isto é possível pois ao trabalhar-se com camadas planas, o planejamento do processo é simplificado, uma vez que os cálculos se restringem à obtenção de trajetórias no plano 2D;

Figura 2 – Adaptada de (Volpato 2017) - Diferentes área de Aplicação da AM, sendo a produção de peças funcionais e encaixes/montagens as duas áreas com aproximadamente metade da utilização do processo



2.2 Automação Industrial

2.3 Sistemas

Um sistema de controle pode ser dividido em duas partes interdependentes:

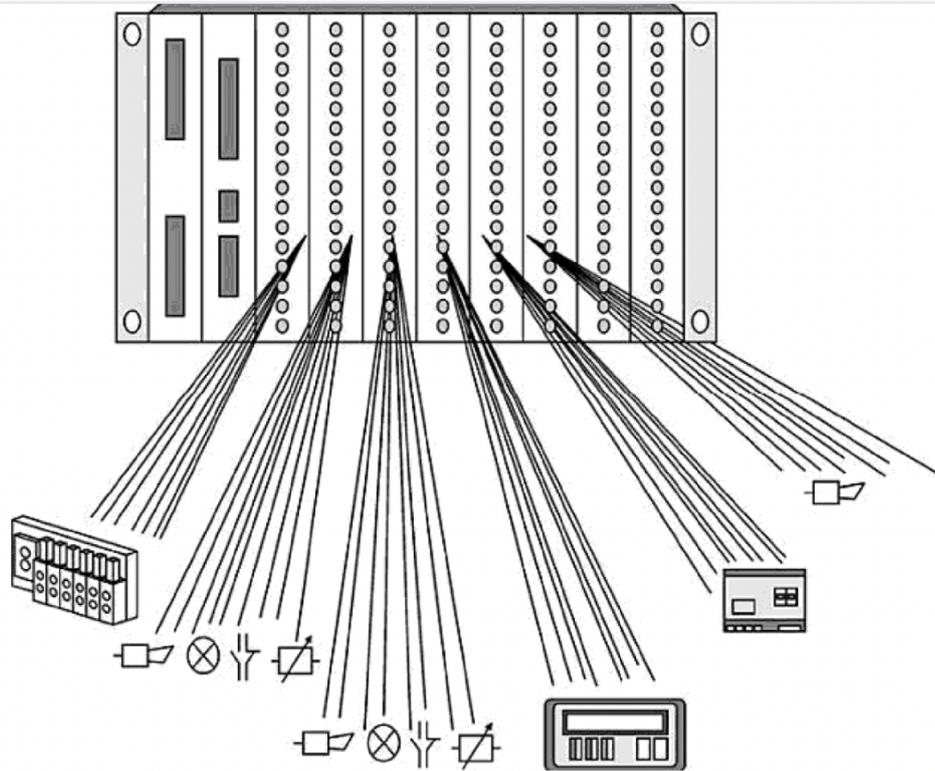
1. Sistema Controlado : executa a operação física
2. Equipamento de Controle : recebe as informações do operador/do processo e emite ordens ao Sistema Controlado

2.4 Protocolos industriais

As redes industriais se tornam cada vez mais presentes no meios industriais. A rápida evolução da eletrônica e da engenharia de software e a miniaturização de componentes e peças são os principais fatores para o desenvolvimento dos sistemas de automação distribuídos com redes de campos industriais. (Rosario 2009) A utilização de tais ferramentas e métodos torna possível desenvolver sistemas de automação compostos com sensores, atuadores, controladores interligados entre si por uma rede a fim de cooperar para a realização de tarefas. Tais equipamentos de rede geram mais confiabilidade e reduzem o custo em comparação aos sistemas centralizados. (Rosario 2009)

2.4.1 Vantagens das redes industriais

Figura 3 – Adaptada de (LUGLI) - Sistema tradicional de um CLP, onde é possível observar que o sistema central é responsável por receber todas as conexões. Consequentemente, têm-se um elevado número de cabos e uma complexidade na resolução de possíveis problemas



Na imagem acima, é possível observar o sistema citado anteriormente, tendo um CLP como o componente responsável por receber todas as conexões. O problema deste tipo de sistema é seu alto custo de implementação uma vez que o número de cabos é elevado e é extremamente difícil encontrar problemas relativos ao sistema.

norma IEC 61158, e agora é CPF 3 que inclui todas as variações de cabeamento do PROFIBUS e PROFINET.

O sistema PROFIBUS é considerado multimestre por permitir a operação conjunta dos equipamentos. Os dispositivos denominados mestres determinam a comunicação de dados em um barramento. Tal comunicação é estabelecida enquanto o dispositivo mestre possui direito de acesso ao barramento (ou token). O token é um mecanismo de arbitragem que deve ser implementado para evitar possíveis colisões no barramento quando mais de uma estação deseja transmitir uma mensagem. (STEMMER 2001)

Os mestres são chamados de estações ativas no barramento. Já os dispositivos escravos são dispositivos de periferia como, válvulas, módulos de E/S, transmissores etc. Esses periféricos não possuem direito de acesso ao barramento, e somente enviam ou reconhecem alguma informação do mestre quando for solicitado.

2.4.3.1 Arquitetura de rede

A arquitetura da rede Profibus é baseada em protocolo de rede que segue o modelo ISO/OSI. No Profibus DP são utilizadas as camadas 1 e 2 e também a Interface do Usuário. Já no Profibus PA e Profinet, além dessas, a camada 7 também é utilizada. Essa arquitetura simplificada garante uma transmissão de dados eficiente e rápida.

- **Camada 1:** inclui o meio físico onde a mensagem é transportada. É nesta camada que ocorre o transporte dos dados por um conjunto serial de bits entre dois equipamentos terminais. A camada física não interpreta dados, sendo responsável somente pela passagem dos dados para a camada de enlace. (TANEMBAUM A. S. 2002)
- **Camada 2:** representa a camada de enlace. É nessa camada que são formados os telegramas de mensagem. Aqui é feito o controle de quando e por qual caminho a mensagem irá trafegar, com o objetivo de evitar colisões entre dois ou mais equipamentos que querem transmitir ao mesmo tempo.
- **Camada 3:** A camada 7 é quem faz a interface entre a máquina e o usuário. Acima da camada 7 está a funcionalidade “real” do instrumento tal como medição, atuação, controle ou a interface de operação de um configurador. (BERGE 2002)

2.4.3.2 PROFIBUS-DP

Originalmente, o PROFIBUS foi definido como PROFIBUS-DP e possuía uma Camada de Aplicação que seguia um padrão já obsoleto chamado MMS (Manufacturing Messaging Specification). Mais tarde, a Organização PROFIBUS abandonou os requisitos para a Camada de Aplicação e desenvolveu uma API diretamente para a Camada de Enlace de Dados para eficiência. O PROFIBUS-DP é uma das redes de automação industrial mais

utilizadas. Como muitos outros protocolos PLC, o PROFIBUS-DP utiliza transferências de registradores. O PROFIBUS-DP é definido como um protocolo de barramento de tokens, mas com exceção de uma situação em que o CP é redundante, não há tokens passados; o protocolo real usado é. senhor de escravos. O mecanismo de passagem de token é usado apenas para transferir o domínio do barramento entre os dois processadores de um CP redundante, pois o barramento permite apenas um mestre por vez. Os dispositivos escravos na rede PROFIBUS são geralmente RTUS ou acopladores para outros barramentos como PROFIBUS-PA ou AS-i. A implementação normal do PROFIBUS usa EIA/TIA 485 em fiação STP (Par trançado blindado) full-duplex. A velocidade de transferência de dados varia com o comprimento da linha. Para comprimento de linha inferior a 3 m, a taxa de transferência é normalmente especificada para 10 Mbps. Para comprimentos de linha mais longos, a taxa de transferência é normalmente de 9600 bps. Há uma variedade de opções de cabeamento de fibra óptica oferecidas por alguns fornecedores para funcionar em velocidades mais altas.

2.4.3.3 PROFIBUS-PA

O PROFIBUS-PA foi projetado para aplicações de controle de processos, e utiliza uma Camada Física idêntica ao Foundation Fieldbus H1, 31,25 kbps, codificado Manchester em fio de cobre de par trançado blindado, e suporta instrumentos de campo intrinsecamente seguros. Diferentemente de H1, o Data Link e todas as camadas superiores são definidas pela especificação PROFIBUS. Os instrumentos PROFIBUS-PA podem ser configurados para realizar cálculos de condicionamento de sinal, como suavização, linearização, conversão de unidades de engenharia e verificação de limite de alarme. Não há padrões para essas funções; cada fornecedor define seu próprio conjunto de funções. Todas as transações estão estritamente sob controle do host e, portanto, não pode haver controle de loop de feedback no dispositivo de campo.

2.4.3.4 PROFINET

Uma das aplicações do PROFINET é substituir o PROFIBUS por uma rede baseada em Ethernet. Todas as interfaces oferecidas para PROFIBUS também são oferecidas para PROFINET. A vantagem óbvia é a maior taxa de dados e extensão da Ethernet com cabeamento de fibra óptica e, ultimamente, com transmissão sem fio Wi-Fi. Existem versões específicas do PROFINET para aplicações de segurança e para aplicações síncronas de alta velocidade em controle de máquina onde o PROFINET-RT pode ser usado, ou para aplicações de altíssima velocidade em corte de metal onde uma variação de hardware especial do chip PROFINET é usado para suportar PROFINET-IRT, tempo real isócrono. (Budampati e Kolavennu 2015)

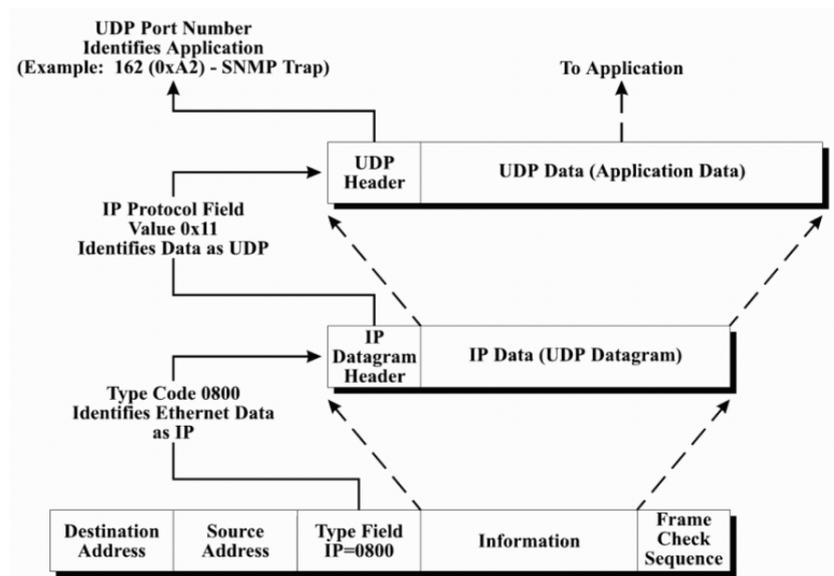
2.5 Protocolo UDP

O User Datagram Protocol (UDP) foi projetado para fornecer aos processos de aplicativos a capacidade de transferir dados para outros aplicativos em máquinas remotas e com uma sobrecarga mínima. O UDP, conforme descrito pela RFC 768, assume que o Internet Protocol (IP) está em uso como o protocolo de rede subjacente e usa o endereço de protocolo 0x11, (17). O UDP não consegue interagir diretamente com os protocolos de acesso das tecnologias de rede subjacentes, e também não oferece confiabilidade adicional: não são enviados avisos de recebimento, os dados não são ordenados no recebimento e, exceto os fornecidos pelo ICMP, não há mecanismos de controle de fluxo. Em vez disso, o UDP é melhor descrito como um protocolo de melhores esforços sem conexão que é orientado a transações e depende do processo de aplicação para fornecer qualquer confiabilidade em relação àquela fornecida pelo IP. Como tal, ao usar o UDP, a entrega não pode ser garantida, a duplicação de datagramas pode ocorrer e os datagramas podem até ser descartados se chegarem a uma taxa mais rápida do que o host de destino pode processá-los. No entanto, apesar do que pode parecer à primeira vista desvantagens significativas, mais especificamente aquelas que exigem sobrecargas de protocolo extremamente baixas, o UDP é um protocolo fundamental sobre o qual muitas funções de uma internet são construídas. (Miller 2009)

2.5.1 Cabeçalho

Como é comum na maioria dos protocolos, o UDP utiliza um cabeçalho de comprimento fixo, além de uma área de dados de comprimento variável. O datagrama UDP, como é chamado, viaja dentro de um datagrama IP que, como no TCP, é posteriormente encapsulado no quadro físico apropriado.

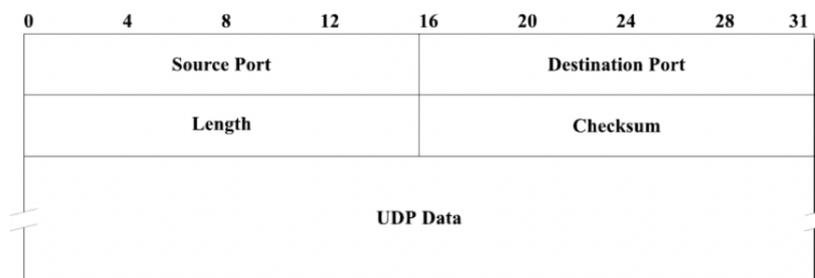
Figura 5 – Adaptada de (Miller 2009) - Encapsulamento do UDP



Como é possível observar, embora os dados reais do aplicativo possam estar enterrados sob três cabeçalhos separados, há informações em cada nível que nos informam quais protocolos estão em uso e, portanto, como esses dados devem ser tratados. Nesse caso, contamos com o campo Type do quadro Ethernet físico para identificar a camada de rede em uso como IP e, em seguida, o campo Protocol do cabeçalho IP Datagram para identificar o transporte como sendo UDP. O número da porta identifica o próprio processo de inscrição.

Como dissemos, o UDP coloca apenas uma pequena sobrecarga na rede, com o próprio cabeçalho UDP sendo pequeno em comparação com outros protocolos com apenas 64 bits (8 octetos) de comprimento. Por causa disso, o cabeçalho fornece apenas informações rudimentares para o protocolo da camada superior (Aplicativo) e, de fato, fornece pouco mais do que simples informações de endereçamento. Assim, embora a natureza compacta do UDP o torne ideal para certas aplicações, ele não pode ser usado onde um transporte de dados confiável é a principal preocupação. Para essas aplicações, o TCP deve ser usado, pois o UDP não fornece a numeração de sequência e o sistema de reconhecimento necessário para entrega de dados garantida. (Miller 2009)

Figura 6 – Adaptada de (Miller 2009) - Cabeçalho do UDP onde é possível observar dois campos de 16 bits que são responsáveis por identificar a origem e o destino da informação, além de informações como tamanho, o dado em questão e o checksum



Para o cabeçalho UDP, os dois campos de porta de dezesseis bits (porta de origem e porta de destino) identificam os processos de aplicativos nos hosts de origem e destino, respectivamente. O campo Source Port é, efetivamente, opcional. Onde usado, porém, é a porta para a qual todas as respostas devem ser enviadas. No entanto, os aplicativos de servidor podem optar por não usar esse campo ou, alternativamente, podem usar um número de porta zero.

2.6 Tecnologia IO-Link

Antes do surgimento da tecnologia IO-Link, os controladores podiam se comunicar apenas com os fieldbuses, mas não no nível de sensores e atuadores. Neste contexto, o IO-Link foi a primeira interface de comunicação ponto a ponto a

conseguir fazer com que sensores e atuadores se comunicassem, permitindo, ainda, uma parametrização de dispositivos de forma central, transmitindo diagnósticos a partir do equipamento até o controlador, e a troca de dados de forma digital e com alta tecnologia de sinal (IO-LINK).

2.6.1 Funcionamento

Cada dispositivo IO-Link possui um arquivo chamado IODD (IO device description ou descrição de arquivo de entrada e saída) que contém informações sobre o equipamento e seus recursos IO-Link disponíveis. Uma vez que este dispositivo é interligado a um cartão Mestre IO-Link estas informações e recursos ficam disponíveis para o sistema de controle, sendo possível acessá-las através do controlador.

Primeiramente, forma-se uma rede ponto a ponto, entre um componente IO-Link (sensor ou atuador) e um cartão Mestre IO-Link (ponto do I/O remoto), por onde a informação irá fluir através da utilização deste protocolo.

O Mestre IO-Link por sua vez transmite os dados dos dispositivos de campo ao controlador através de uma rede fieldbus (ethernet, devicenet, profibus, profinet, entre outras).

Sendo assim, as informações dos dispositivos IO-Link ficam transparentes para o controlador possibilitando uma grande gama de programações possíveis, fazendo com que o sensoriamento da máquina seja muito mais integrado ao sistema de controle. (Vantagens IO-LINK)

2.6.2 Vantagens

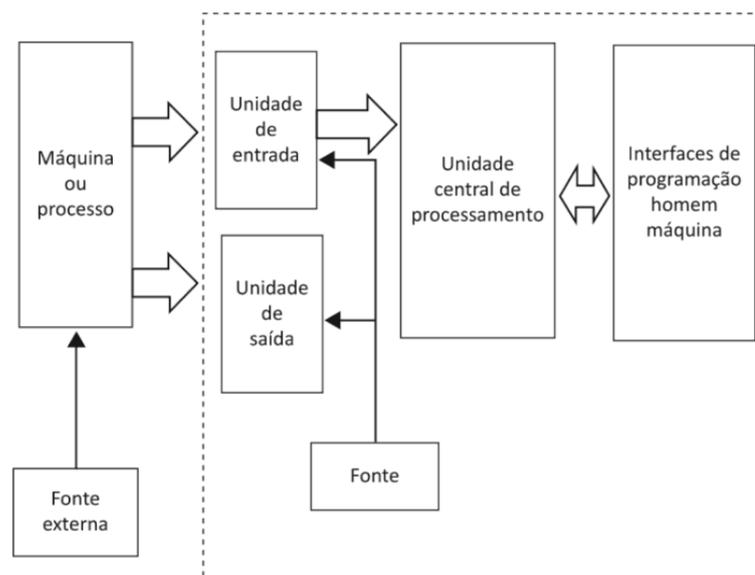
- Como os dispositivos IO-Link possuem comunicação com o controlador da máquina é possível salvar a parametrização de um sensor no próprio controlador. Isso faz com que fique salvo no sistema de controle os dados e parâmetros da aplicação.
- Dispositivos IO-Link são capazes de fornecer informações de diagnóstico, como temperatura interna, número de ciclos, horas de funcionamento, intensidade luminosa, entre outros. Desta maneira, há um diagnóstico preventivo muito mais eficaz.
- Os sensores IO-Link são capazes de operar em diversos modos de detecção (modo oposto, retro-reflexivo, difuso, supressão de fundo).
- Os sensores com IO-Link funcionam com apenas três fios o que faz com que sua ligação elétrica seja muito simples. (Vantagens IO-LINK)

2.7 Controlador Lógico Programável (CLP)

Um controlador lógico programável pode ser definido como um computador industrial capaz de armazenar informações para a realização do controle de um processo.

2.7.1 Arquitetura

Figura 7 – Adaptada de (Silva 2016) - Etapas de um ciclo SCAN, onde recebe-se os sinais através dos periféricos de entrada, administra-se as funções executando a lógica de programação e posteriormente gera-se um sinal para ser enviado aos periféricos de saída



- **Unidade de entrada:** recebe sinais elétricos da máquina ou processo e tem compatibilidade com o circuito eletrônico do CLP.
- **Unidade de saída:** recebe os sinais processados pelo CLP e gera um sinal elétrico para a utilização da máquina ou do processo.
- **Unidade de processamento:** administra todas as funções, é responsável por receber os sinais da unidade de entrada, executa a lógica de programação do usuário e atualiza o resultado na unidade de saída.
- **Fonte de alimentação:** adapta a energia elétrica para que o CLP possa funcionar de forma correta.

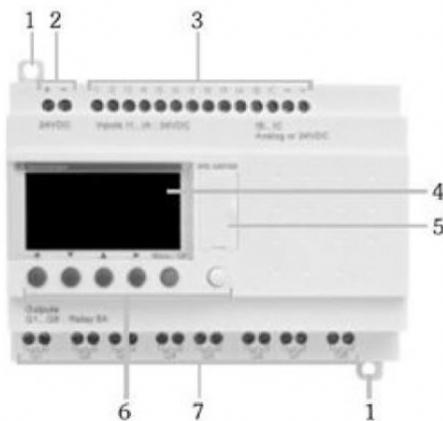
2.7.2 Tipos de CLP

Existem 2 tipos principais de CLP: compactos e modulares.

2.7.2.1 Compactos

Possuem incorporados em uma única unidade a fonte de alimentação, a CPU e os módulos de E/S. Desta forma, o usuário tem acesso apenas aos conectores do sistema E/S. Tal estrutura é empregadas em CLPs de pequeno porte.

Figura 8 – Adaptada de (Franchi e Camargo) - Exemplo de CLP compacto, onde o usuário tem acesso apenas ao conectores de entrada e saída

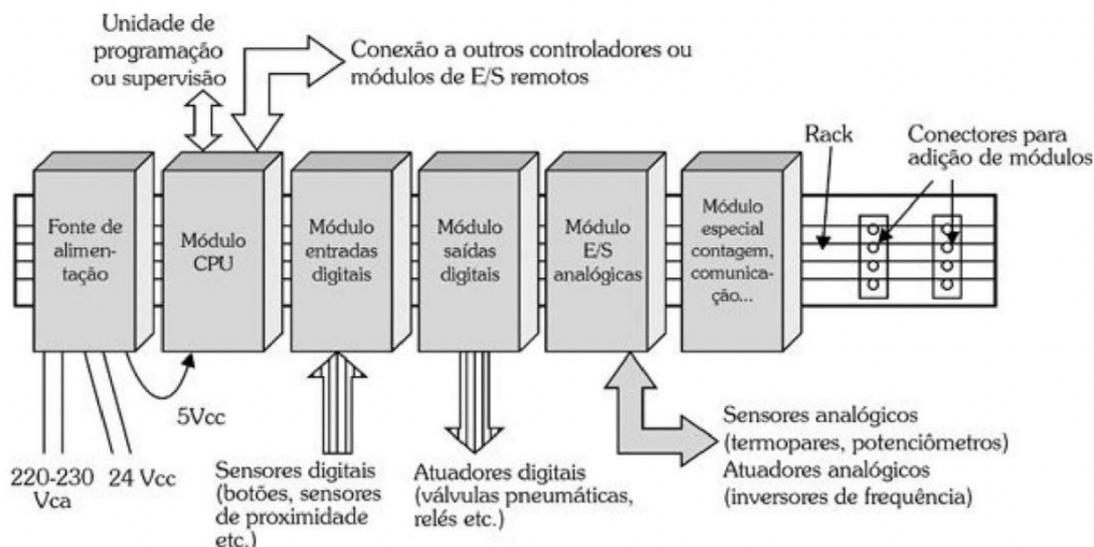


1. Dois terminais de fixação.
2. Dois terminais para fonte de alimentação.
3. Terminais para conexão das entradas.
4. Display LCD com 4 linhas e 18 caracteres.
5. Slot para cartão de memória ou conexão com PC ou interface de comunicação com modem.
6. Seis botões para programação e entrada de parâmetros.
7. Terminais para conexão das saídas.

2.7.2.2 Modulares

Tais PLCs são compostos por uma estrutura modular, sendo que cada módulo é responsável por executar uma determinada função. O sistema de entrada/saída é decomposto em módulos de acordo com suas características. Eles são colocados em posições predefinidas (denominadas *racks*), formando uma configuração de médio e grande porte. Esse tipo de CLP tem uma gama variada de modelos, partindo dos denominados MicroCLPs que suportam uma pequena quantidade de E/S e alcançando os CLPs de grande porte os quais são capazes de tratar até milhares de pontos E/S.

Figura 9 – Adaptada de (Franchi e Camargo) - Arquitetura de CLP modular, onde é possível observar a presença de módulos variados responsáveis por executarem diferentes funções



2.7.3 Modos de operação de um CLP

O PLC, de maneira geral, pode funcionar em modo programação ou em modo execução.

2.7.3.1 Modo Programação

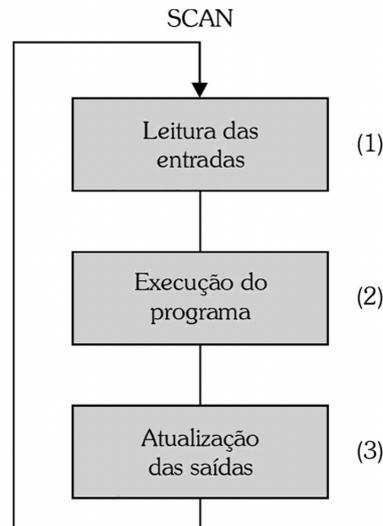
Neste modo, não há a execução de nenhum programa. Logo, o PLC fica aguardando para ser configurado ou receber novos programas.

2.7.3.2 Modo Execução

Tal modo permite a execução do código anteriormente carregado. O funcionamento do CLP é baseado num sistema multiprocessado em que há uma estrutura de software que realiza continuamente ciclos de leitura chamados de scan (Franchi e Camargo). O scan é composto de 3 processos básicos:

1. Leitura dos dados através dos dispositivos via interface de entrada
2. Execução do programa de controle armazenado na memória
3. Escrita ou atualização dos dispositivos de saída via interface de saída

Figura 10 – Adaptada de (Franchi e Camargo) - Etapas de um ciclo SCAN, onde lê-se as entradas, executa-se a lógica de programação definida pelo usuário e finalmente atualiza-se as saídas



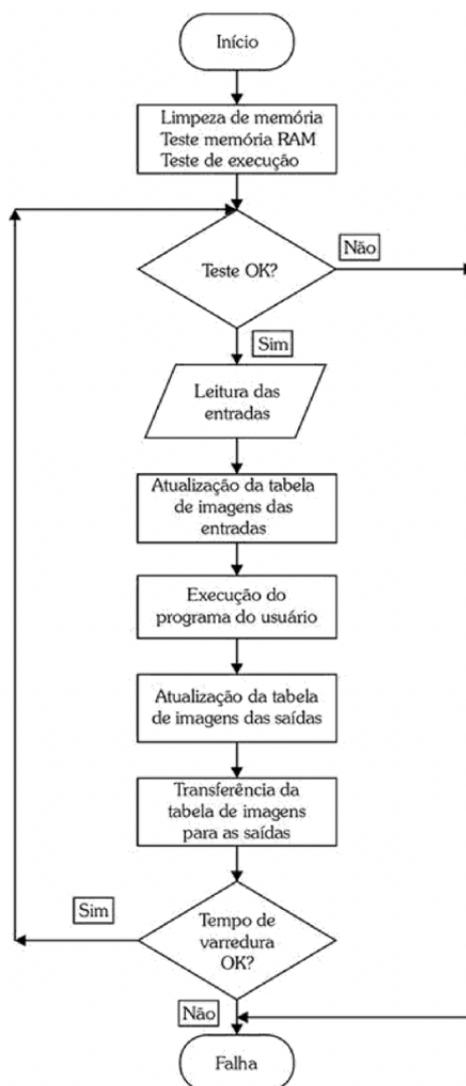
Assim que o PLC é energizado, primeiramente uma limpeza da memória é realizada, logo após um teste de memória RAM é feito e finalmente um teste de executabilidade do programa encerra a rotina de inicialização do aparelho.

A próxima etapa realizada pela CPU é a leitura em laço fechado dos pontos de entrada. Neste processo, a CPU endereça o sistema de E/S, coleta os estados atuais dos dispositivos que estão conectados e armazena tais informações em forma de bits (1 ou 0). O armazenamento é feito em uma região da memória denominado Tabela Imagem das Entradas (TIE). Ao executar a lógica programada, a TIE é utilizada para obter os estados dos dispositivos. Os resultados das lógicas programas que atuam em determinadas saídas são armazenadas em uma área da memória denominado Tabela Imagem das Saídas (TIS).

Na etapa de atualização das saídas, ocorre-se uma varredura na tabela TIS e uma atualização das saídas externas através do endereçamento do sistema E/S para atualizar o estado dos dispositivos de saída. Também é realizada a atualização de outros operandos tais como resultados aritméticos, contagens, temporizadores, entre outros. Com o fim da atualização da tabela imagem, a transferência dos valores da tabela imagem das saídas para os cartões de saída é feita encerrando-se assim o ciclo de varredura.

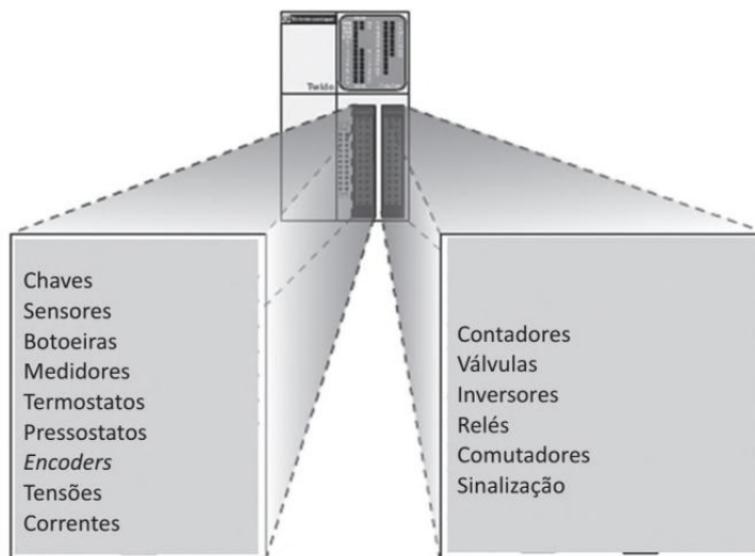
Com relação à verificação de erros, um tempo de processamento é definido sendo que fica a cargo do circuito Watch Dog Timer supervisioná-lo. Logo, se o tempo máximo for ultrapassado, a execução do programa é interrompida.

Figura 11 – Adaptada de (Franchi e Camargo) - Fluxo de execução de um PLC



2.7.4 Tipos de Entradas e Saídas

Figura 12 – Adaptada de (Silva 2016) - Exemplos de entradas e saídas de um PLC



2.7.5 Tipos de variáveis

As variáveis fornecem um meio de identificação para objetos de dado cujo conteúdo pode mudar.

Figura 13 – Adaptada de (Silva 2016) - Tabela com os diferentes tipos de representação de variáveis

Sinal inicial	Identificação de memória	Tamanho do dado		Descrição
%	M (Acesso à memória)	X	(1 bit)	Acesso às variáveis booleanas.
	I (Entrada física)	W	(16 bits)	Acesso às variáveis com 16 bits de tamanho: INT, UINT e WORD.
	Q (Saída física)	D	(32 bits)	Acesso às variáveis com 32 bits de tamanho: DINT, UDINT, DWORD.
		T	(32 bits)	TIME, DATE, TOD e DATE_AND_TIME.
		R	(32 bits)	Acesso às variáveis com 32 bits de tamanho: REAL.
		A	-----	O conteúdo dessa região é definido pelo usuário conforme a necessidade do projeto.

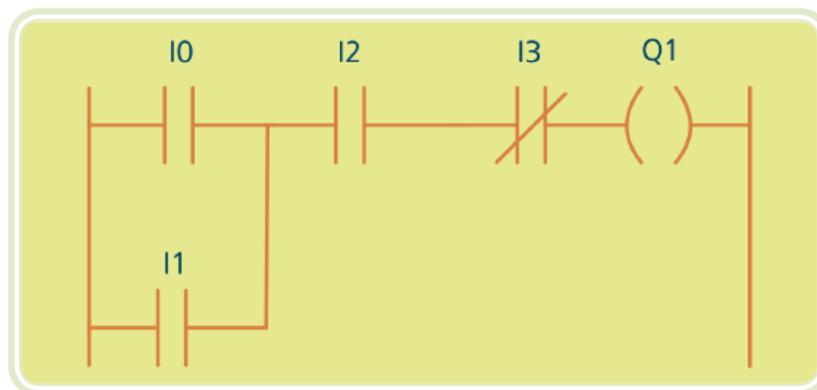
Para serem acessadas, as variáveis podem utilizar nomes simbólicos e ser de diferentes tipos de dados. É possível que elas tenham alocação dinâmica ou associada a posições físicas da memória.

2.8 Linguagem LADDER

A fim de padronizar uma semântica e uma sintaxe de um conjunto de linguagens de programação para um CLP, a norma IEC61131-3 foi criada. As mais comuns são: texto estruturado (Structured Text, ST), diagrama de blocos funcionais (Function Block Diagram, FBD), diagrama Ladder (Ladder Diagram, LD) e gráfico sequencial de função (Sequential Function Chart, SFC). A definição dos elementos das linguagens são definidos na programação do controlador (Silva 2016).

No caso do projeto, a linguagem escolhida foi o diagrama de blocos (ou linguagem Ladder). Esta linguagem é a mais utilizada em CLPs e assemelha-se a um diagrama elétrico.

Figura 14 – Adaptada de (ROGGIA L 2016) - Exemplo de utilização da linguagem Ladder



No exemplo apresentado, para que a saída Q1 seja acionada, pelo menos uma das entradas I0 e I1 deve estar acionada e, além disso, a entrada I2 deve estar acionada e a entrada I3 deve estar desacionada.

Os diagramas de contatos são uma forma de programação de CLPs por meio de símbolos gráficos, representando contatos e bobinas. Os diagramas são compostos estruturalmente de duas linhas verticais e de linhas horizontais (escada), sob as quais são colocadas as instruções a serem executadas.

O programa deve ser construído partindo do pressuposto de que as instruções devem ser “energizadas” a partir de um “caminho de corrente” entre as duas barras, sendo que o fluxo simulado de “corrente elétrica” em uma lógica flui no sentido da barra da esquerda para a barra da direita. A barra da direita pode ser omitida da representação.

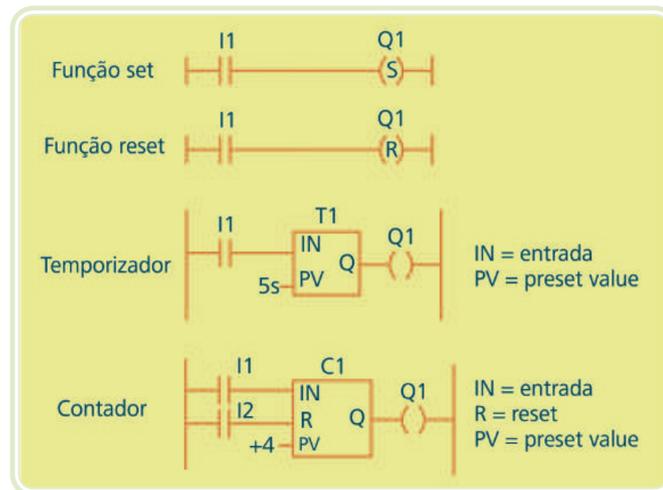
Figura 15 – Adaptada de (ROGGIA L 2016) - Elementos principais do diagrama de contatos

Principais elementos dos diagramas de contatos		
Tipo	Símbolo	Elemento elétrico
Contato aberto		
Contato fechado		
Saída (bobina)		

Existem outros tipos de funções mais avançadas:

- **Função set:** tem por finalidade acionar uma bobina e mantê-la acionada mesmo após cessar o estímulo.
- **Função reset:** tem por finalidade desacionar uma bobina previamente acionada pela função set.
- **Temporizador:** tem por finalidade acionar ou desligar uma memória ou uma saída de acordo com um tempo programado. No temporizador com retardo na energização, por exemplo, uma saída será ligada após decorrido um determinado tempo a partir do acionamento do temporizador. No temporizador com retardo na desenergização, uma saída será desligada após decorrido um determinado tempo a partir do acionamento do temporizador. Essas funções são utilizadas, por exemplo, em chaves de partida de motores de indução, como a partida estrela-triângulo.
- **Contador:** tem por finalidade ativar uma memória ou uma saída após uma determinada contagem de eventos. Um contador crescente pode, por exemplo, acionar uma saída após um botão ou um sensor ter sido acionado um determinado número de vezes previamente programado.

Figura 16 – Adaptada de (ROGGIA L 2016) - Funções avançadas do diagrama de contatos

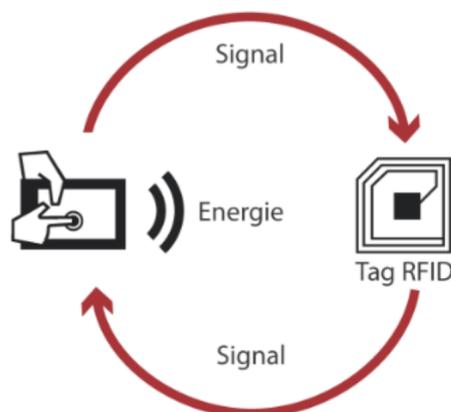


2.9 Tecnologia RFID

RFID (Radio Frequency Identification) é uma tecnologia que nos permite salvar e recuperar dados remotamente nas chamadas tags ou tags RFID ou mesmo tags de rádio. Altamente utilizada na indústria de segurança, a tecnologia RFID tem muitas aplicações.

A tecnologia RFID se baseia em transferências de energia por eletromagnetismo. Para implementá-la, é necessário ter marcadores (tags, tags ou chips RFID que também são chamados de transponders) e um leitor RFID. O leitor RFID emite radiofrequências que ativam os chips RFID que passam nas proximidades. Para isso, ele lhes envia a energia necessária a uma curta distância.

Figura 17 – Funcionamento da tecnologia RFID



As etiquetas RFID contêm um circuito integrado e uma antena, a a qual recebe a energia sem fio emitida pelo leitor para alimentar seu circuito integrado para transmitir

dados para o leitor RFID (também chamado de interrogador). O leitor então converte as ondas de rádio para uma forma de dados mais utilizável.

A RFID apresenta várias vantagens se comparada a outros tipos de sistemas similares. Por exemplo, ela utiliza comunicação sem fios, não necessitando assim de contato físico nem de campo visual direto. Além disso, permite a leitura e escrita de dados a grandes velocidades e um grande armazenamento de informação, sendo superior a tecnologias similares nestes pontos. Outro ponto positivo é o fato de as tags, além de serem baratas (ficando ainda mais quando compradas em grandes quantidades), poderem ser reutilizadas e exigirem uma manutenção mínima. Também, o sistema pode ser utilizado em ambientes severos, com condições adversas, tais como, locais húmidos, molhados, sujos, poeirentos, corrosivos, sujeitos a altas ou baixas temperaturas, sujeitos a vibrações ou choques, entre outros (Neiva et al. 2012).

Tabela 1 – Adaptada de (Frequency Table) - Tabela de frequências e suas aplicações

Tipo de frequência	Banda	Alcance	Velocidade dos dados	Regulações ISO/IEC 180000	Aplicações
LF	120 - 150 kHz	10 cm	Baixa	Parte 2	Identificação de animais Coleta de dados de fábricas
HF	13.56 MHz	10 cm - 1 m	De baixa à moderada	Parte 3	Controle de acesso Rastreamento de bagagem e livros
UHF	433 MHz	1 - 100 m	Moderada	Parte 7	Aplicações de defesa com tags ativas
UHF	865 - 868 MHz	1 - 12 m	De moderada à alta	Parte 6	Utilizado por ferrovias
SHF	2450 - 5800 MHz	1 - 2 m	Alta	Parte 4	Bluetooth
Faixa de banda larga	3.1 - 10 GHz	>200 m	Alta	Não definido	-

2.9.1 Tag ou etiqueta

Tag é um transponder (abreviatura de transmitter-responder) que, segundo a definição, é um dispositivo de comunicação eletrônico de automação com o objetivo de receber, amplificar e retransmitir um sinal em uma frequência diferente ou transmitir de

uma fonte uma mensagem pré-determinada em resposta à outra pré-definida proveniente de outra fonte.

Existem três tipos de etiquetas ou tags:

- **Passivas:** não possuem alimentação interna nem transmissor incorporado. Por isso, o circuito integrado e as comunicações são feitas utilizando a energia gerada pelo sinal de radiofrequência, sendo este a única fonte de alimentação da etiqueta em questão. Consequentemente, tal tipo possui um alcance reduzido (de mm a poucos cm). Possuem um identificador único denominado GUID (da designação inglesa Global Unique IDentification), também podem possuir uma EEPROM para armazenar dados. Um exemplo de etiqueta passiva são os cartões de acesso à portas com fechaduras eletrônicas. Como vantagens, tais tags são mais baratas, mais flexíveis e mais duradouras. Porém possuem um alcance extremamente reduzido.
- **Semi-passivas:** são tags acopladas a baterias, as quais permitem alimentar o circuito integrado. No entanto, não possuem um transmissor integrado e, assim como as passivas, necessitam da energia do sinal de radiofrequência proveniente do leitor para efetuar as comunicações de volta para o leitor.
- **Ativas:** possuem alimentação própria e um transmissor integrado permitindo, por isso, maiores níveis de potência do que as passivas e semi-passivas e, consequentemente, um maior alcance (na ordem das dezenas de metros). Podem estar ligadas a uma infraestrutura que forneça energia ou possuir uma bateria. No caso da utilização de baterias, a vida útil destas varia conforme a quantidade de energia armazenada e o número de operações de leitura/escrita efetuados, podendo chegar aos 10 anos. (Neiva et al. 2012).

3 Materias e Métodos

O Instituto Nacional de Ciências Aplicadas de Estrasburgo (INSA Strasbourg) comprou recentemente uma nova impressora 3D com uma função de impressão contínua. Tal modo permite que a impressora produza cada modelo 3D em uma placa para cada objeto impresso. A função "impressão contínua" funciona da seguinte forma: uma vez que um objeto é impresso em sua própria chapa, o conjunto é então empurrado pela chapa seguinte, para se preparar para a próxima etapa de impressão.

Após a etapa de impressão, a placa com a peça será transportada da caixa da impressora para o braço de manipulação robótico através de uma esteira transportadora. Quando a placa com o objeto impresso chegar ao final da linha, um sensor o detectará e, através do programa de software no CLP principal, o passo anterior mandará um comando para a esteira parar e para o braço do robô (Dobot Magician neste caso) agarrar a placa com o objeto sobre ela. A mesa do projeto contém uma impressora 3D modelo Tiertime X5, um transportador, dois braços robóticos chamados "Dobot" e um transportador de objetos. A impressora 3D é capaz de imprimir continuamente e tem o objeto solicitado por um terceiro. Em seguida, o objeto impresso é levado na esteira e durante o transporte neste, o objeto passa por um processo de etiquetagem RFID. Uma vez no lugar certo, o objeto é levado pelo primeiro Dobot que o coloca no AGV. Já no AGV, a peça é levada para a região de triagem de peças, onde primeiramente é lida por um sensor RFID e, de acordo com a informação obtida na leitura, é levada para uma das quatro estações diferentes disponíveis através do segundo Dobot disponível.

3.1 Especificações e limitações

Considerando a complexidade do projeto, algumas especificações foram impostas para que os objetivos e o planejamento do projeto pudesse ser melhor estudado e estruturado. Tais especificações foram propostas pelos professores responsáveis pelo acompanhamento do projeto e devem ser minimamente seguidas.

3.1.1 Etiquetagem RFID e transporte

3.1.1.1 Especificações

- O Dobot tem de ser coordenado com a esteira transportadora usando sensores de distância IR;

- As bandejas têm de ser recolhidas e transferidas no carrinho AGV pelo Dobot usando Wi-Fi;
- Os itens ou bandejas têm de ser etiquetados usando RFID: no caso de etiquetar os itens, um sistema mecânico automatizado tem de ser construído para esta finalidade. As etiquetas ou serão círculos de plástico duro ou impressas em papel. A máquina teria de ser desenvolvida levando em conta vários fatores. Pela questão do tempo curto para este projeto, optou-se por etiquetar todas as bandejas de antemão;
- As informações sobre as peças devem ser escritas em etiquetas RFID. As informações serão sobre gravadas nas etiquetas para cada ciclo;
- Os Dobots têm de ser configurados usando seu módulo Wi-Fi;
- Os Dobots têm de ser conectados com o CLP Siemens;
- O Dobot / o CLP tem de ser coordenado com o AGV.

3.1.1.2 Principais desafios

- Programação Dobot;
- Precisão Dobot;
- Procedimento de escrita RFID;
- Comunicação Wi-Fi com Dobot;
- Programação PLC;
- Comunicação entre PLC e sensor RFID.

3.1.2 Veículo automaticamente guiado (AGV)

3.1.2.1 Especificações

- O robô deve ser autônomo;
- O robô é controlado por meio de Wi-Fi.
- Comunicação Wi-Fi com o CLP Siemens tem que ser alcançada: os dados enviados ao robô pelo aplicativo do fabricante têm que ser analisados para coordenar o AGV com os Dobots. Isto requer a análise de dados complexos enviados por Wi-Fi e pode levar muito tempo e não é a prioridade. A opção preferida a partir de agora é usar a Interface do AGV e automatizar os controles do computador. O robô inicialmente não seria coordenado com o CLP, mas esperaria uma quantidade de tempo pré-definida

em cada local onde tem de estar. Este recurso já foi parcialmente testado e não deve apresentar problemas de implementação.

- Para ir mais longe usando esta solução, precisamos encontrar uma maneira de enviar dados para o computador e integrá-los totalmente ao ciclo.

3.1.2.2 Principais dificuldades

- Comunicação com o robô usando o CLP;
- Comunicação com o computador usando o PLC e automatizar o computador em consequência;
- Coordenação da AGV com o resto dos elementos.

3.1.3 Classificação dos itens

3.1.3.1 Especificações

- As etiquetas RFID têm que ser lidas e as informações enviadas ao CLP a fim de controlar o Dobot;
- O Dobot tem que ser configurado usando seu módulo Wi-Fi;
- O Dobot tem que ser ligado ao CLP Siemens;
- O Dobot / o CLP tem que ser coordenado com o AGV;
- O Dobot tem que ser capaz de agarrar peças específicas ou se apenas uma peça for transportada de cada vez, ele poderá inclinar a bandeja. Esta é a opção preferida;
- O Dobot tem que se mover lateralmente usando um trilho.

3.1.3.2 Principais dificuldades

- Programação Dobot;
- Dobot e precisão do trilho;
- Procedimento de leitura RFID do item;
- A comunicação Wi-Fi com Dobot aparentemente não está funcionando plenamente e é propensa a bugs;
- Programação PLC;
- Comunicação PLC e leitor RFID;
- Coordenação precisa do trilho e do CLP/Dobot.

3.2 Materiais utilizados

Para o desenvolvimento deste projeto, foram utilizados alguns materiais específicos que serão apresentados a seguir assim como suas características.

3.2.1 CLP Siemens 1200 1214C DC/DC

O CLP pode ser descrito como um dispositivo eletrônico digital com a possibilidade de processar funções que utilizam lógica binária, executar comandos a fim de controlar e monitorar sistemas ou equipamentos (Rosario 2009).

As entradas e saídas do sistema são conectados aos instrumentos de campo de forma física, interfaceando a CPU e o meio externo. Desta forma, estas entradas e saídas ligadas a ele são atualizadas de forma automática.

Uma das maiores vantagens do CLP é a facilidade na alteração de sua lógica, bastando apenas reprogramar as instruções de sua memória.

Como vantagem, o CLP é perfeitamente adaptável ao ambiente industrial, não interferindo em seu funcionamento as vibrações, variações de temperatura ou alimentação e nem ruídos elétricos.

O CLP executa ciclicamente uma sequência de instruções, sendo as seguintes fases as mais importantes: a leitura das variáveis de entrada, a execução do programa de aplicação e a atualização das variáveis de saída (Pupo 2002). Na primeira etapa, os sensores presentes no campo realizam a transformação de grandezas físicas em pulsos elétricos. Estes são enviados ao CLP e armazenados em sua memória até que haja alguma mudança na variável. Inicia-se então a segunda etapa, onde o CLP executa o programa, seguindo suas instruções. Por fim tem-se o último passo, onde ao finalizar um ciclo de varredura do programa, o CLP atua na saída e atualiza as variáveis, iniciando um novo ciclo.

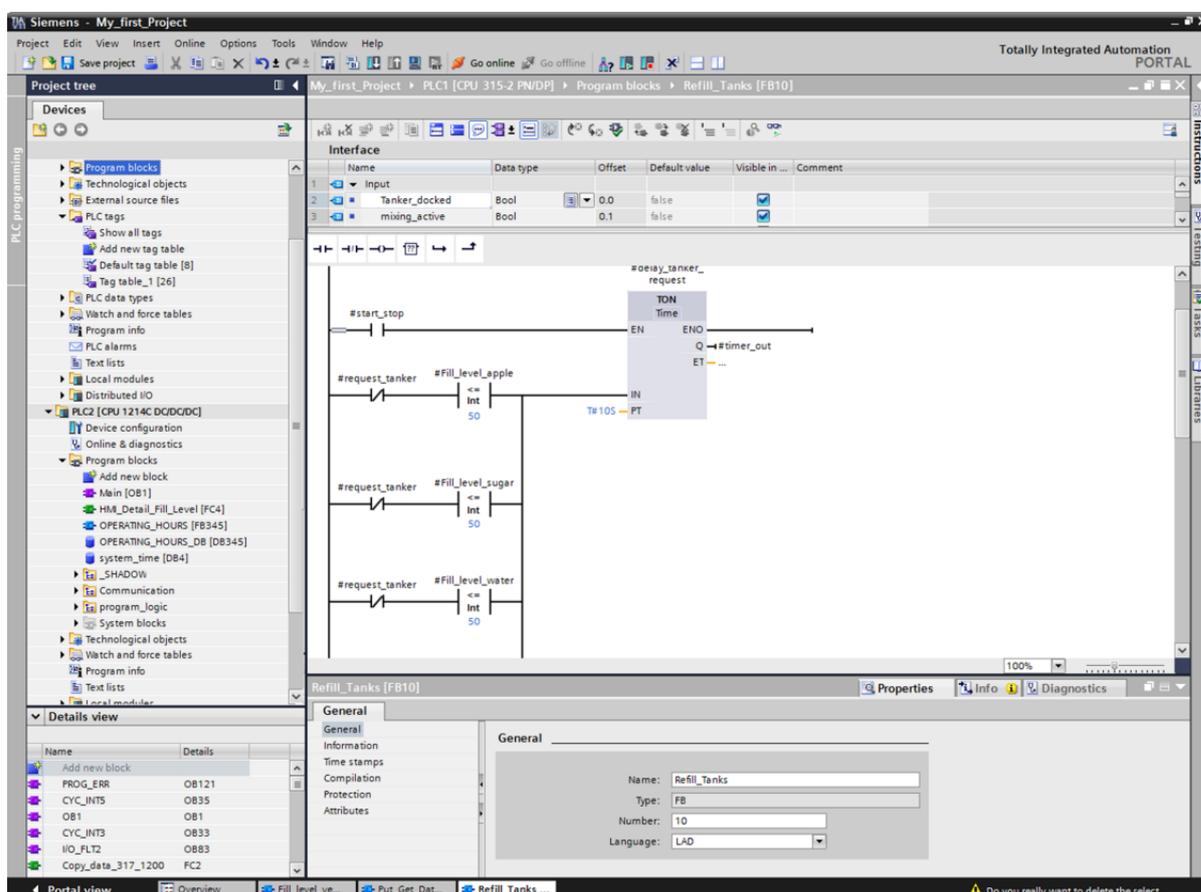
Figura 18 – Adaptada de (Siemens 2012) - PLC S7 1200 utilizado no projeto



No caso deste projeto, foi utilizado um CLP da marca Siemens o qual tem interfaces Ethernet/Profinet, bem como entradas/saídas digitais e analógicas. Ele pode ser usado

para funções tecnológicas como contagem, medição PID, controle de movimento e registro. Tal componente foi utilizado para transmitir o programa de automação através do software TIA Portal para os vários sensores.

Figura 19 – Adaptade de (Siemens 2009) Interface do software TIA Portal



3.3 AL1302 Mestre

Este dispositivo, utilizando a tecnologia IO-Link, serve como um gateway entre os dispositivos IO-Link e a rede PROFINET. Este produto contém:

- 2 portas para acesso à interface PROFINET;
- 1 porta ethernet para aplicações de internet de coisas. A interface permite acesso separado das redes de computadores aos parâmetros, processos e dados de monitoramento do IO-Link Mestre e dispositivos IO-Link conectados. Protocolos diferentes (por exemplo, JSON TCP / IP) são suportados;
- Gateway para a transmissão de dados de processo e parâmetros entre os dispositivos IO-Link conectados e o controlador PROFINET de nível superior;
- 8 entradas digitais adicionais (tipo 2 de acordo com a EN 61131-2).

Figura 20 – Adaptada de (IFM - AL1302) - Layout do AL1302 Mestre



O PROFINET (Process Field Network) é um protocolo aberto de comunicação industrial, baseado no Fast Ethernet, que manteve do padrão Ethernet original a forma de endereçamento, o formato, o tamanho do Frame e o mecanismo de detecção de erros (Oliveira 2016).

Além disso, o PROFINET se baseia nos protocolos TCP, UDP e IP para configuração, troca de dados e diagnóstico de rede, sendo seu principal objetivo a criação de um ambiente de rede industrial integrado, robusto e seguro (Oliveira 2016).

3.4 O5D100 - Sensor de distância fotoelétrico

Os sensores fotoelétricos são sensores bastante estáveis e não apresentam degradação significativa ao longo do seu período de uso. Além disso, devido ao seu tempo de resposta bastante curto, são apropriados para aplicações que necessitem observar variações de radiação quase que instantaneamente (Victoria 2008).

Os sensores de medição de distância por IR existentes baseiam-se em um de dois métodos, ou por triangulação, que é o caso do sensor utilizado nesta investigação, ou pela medição da intensidade da energia refletida pelo obstáculo (Mota 2018).

No caso do segundo princípio mede-se a intensidade da energia que depois de emitida pelo LED infravermelho é refletida pelo objeto.

Em suma, este sensor de distância envia um raio do tipo laser para medir a distância entre ele e o obstáculo a sua frente. Os dados digitais são exibidos no sensor, mas também podem ser recuperados através do software Tia Portal.

Figura 21 – Adaptada de (IFM - O5D100) - Layout do sensor fotoelétrico O5D100



3.5 OGD592 - Sensor de distância fotoelétrico

Seu funcionamento é idêntico ao descrito na seção anterior. O fator que os difere é o intervalo de medição: sendo o primeiro capaz de detectar objetos até três metros e este até trinta centímetros.

Tal sensor foi escolhido por ser mais rentável economicamente e por não haver necessidade de operá-lo em detecções que fossem além de dez centímetros.

Figura 22 – Adaptada de (IFM - OGD592) Layout do sensor fotoelétrico OGD592



3.6 DOBOT

Dobot Magician é um braço robótico de mesa leve e multifuncional disponível comercialmente, tendo um corpo preto e branco com qualidade consistente nas pontas dos dedos. Além disso, é um braço de Robô All-in-One para a educação. Ele pode ser controlado por PC, telefone, gesto ou voz, pois pode se comunicar por USB, Wifi e Bluetooth (Islam et al.).

O Dobot é um robô 3 graus de liberdade que possui três motores de passo para

acionar suas articulações (base, ombro e cotovelo). O mesmo possui 13 portas de extensão, 1 programável e 2MB de armazenamento de comandos offline (Zhu, Gu e Yang).

Figura 23 – Adaptada de (Shenzhen Yuejiang Technology Co 2017) - Dobot Magician



3.6.1 Características Técnicas

Tabela 2 – Adaptada de (Shenzhen Yuejiang Technology Co 2017) - Tabela de características técnicas do Dobot

Número de eixos	4
Carga útil	500g
Alcance máximo	320mm
Repetibilidade	0,2mm
Communication	USB / WiFi * / Bluetooth
Alimentação	100 V - 240 V, 50/60 Hz
Fornecimento de energia	12 V / 7 A DC
Consumo	60 W max
Temperatura de operação	-10°C à 60°C

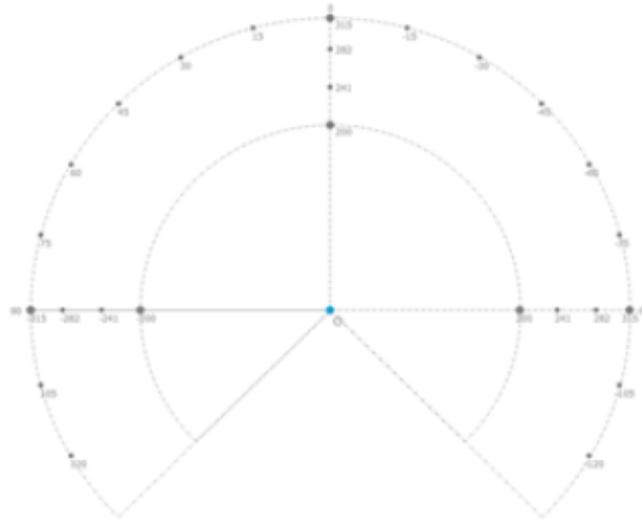
3.6.2 Características Físicas

Tabela 3 – Adaptada de (Shenzhen Yuejiang Technology Co 2017) - Tabela de características físicas do Dobot

Peso Net	3,4 kg
Peso bruto (versão padrão)	7,2 kg
Peso bruto (versão Educação)	8 kg
Dimensão de base (superfície)	158 mm x 158 mm
Materiais	Liga de alumínio 6061, plástico ABS
Controlador	Controle integrado Dobot
Fixação do robô	Mesa

3.6.3 Área de Trabalho Dobot

Figura 24 – Adaptada de (Shenzhen Yuejiang Technology Co 2017) - Grau de rotação do Dobot



O braço tem uma área de trabalho que vai desde :

- 0 a 315 mm ao longo do eixo X em relação à base;
- -135° a 135° de rotação ao redor do eixo Z (J1).

3.7 SDP Mini RPLidar Experimental 2WD Robot Platform

Um AGV típico consiste de: estrutura, baterias, sistema elétrico, unidade de acionamento, direção, unidade de parada de precisão, controlador de bordo, unidade de comunicação, sistema de segurança, e plataforma de trabalho. Os sistemas AGV são utilizados principalmente para a distribuição de materiais em ambientes de depósito e movimentação de material de e para as áreas de produção e áreas de armazenamento em instalações fabris (Sezen 2011).

A Plataforma SDP Mini RPLidar Experimental 2WD é uma plataforma de desenvolvimento de pequeno porte projetada pela Slamtec com a qual os usuários podem ter um entendimento geral sobre a solução Slamware e examinar seu desempenho básico. Neste projeto, chamamos este componente de AGV ou veículo guiado de forma autônoma. O SDP Mini é uma estrutura compacta adequada para a avaliação preliminar da solução Slamware ou para o desenvolvimento de robôs com necessidades básicas de localização e função de navegação.

Ele é formado por componentes-chave incluindo Slamware Core, RPLidar A2 e placa Breakout 3.0, e requer apenas 8 baterias AA. Ele será usado para levar a peça para a área de triagem.

Figura 25 – Adaptada de (SDP Mini Manual) - AGV SDP Mini



3.8 Etiquetas RFID

Hoje em dia, a rastreabilidade ocupa um papel importante no setor. Ela fornece rastreamento completo e permite coletar informações sobre o produto, tais como:

- Nome do Produto;
- Data/Horário de confecção;
- Informações com relação ao fabricante.

Para este fim, o cliente requer uma etiqueta RFID que contenha as diversas informações relativas à peça impressa anexada a cada placa do produto.

Antes de explicar a abordagem utilizada para responder ao pedido do cliente, é necessário explicar a tecnologia RFID e seu princípio de funcionamento.

3.8.1 MDS D400 - Tag

Considerando a aplicação do projeto, o tipo passivo foi o escolhido. O formato escolhido para a tag foi baseado na distância de leitura/escrita e também na geometria do objeto em relação à chapa que servirá de base para a impressão da peça. Havia outra opção disponível no laboratório: uma etiqueta redonda. Testes foram feitos e a distância de leitura/escrita obtida foi de cerca de 2 cm enquanto que para a tag em forma de cartão foi de 7 cm. O segundo problema encontrado foi que as etiquetas redondas foram fixadas a

uma base quadrada de 4cmx4cm, impossibilitando a colocação desta etiqueta redonda na placa base da impressora 3D.

Figura 26 – Adaptada de (Siemens 2020) - MDS 400 - Tag em formato de cartão



Figura 27 – Adaptada de (Siemens 2020) - Informações técnicas sobre a Tag

radio frequencies	
operating frequency / rated value	13.56 MHz
range / maximum	650 mm; range is reader dependent: observe http://support.automation.siemens.com/WW/view/en/67384964
protocol / with radio transmission	ISO 15693
transfer rate / with radio transmission / maximum	26.5 kbit/s
product feature / multitag-capable	Yes
electrical data	
product component / backup battery	No
memory	
type of memory	FRAM
storage capacity / of the user memory	2000 byte
type of memory organization	UID (fixed code) 8 bytes, user memory 2000 bytes, configuration memory 40 bytes
number of read cycles / at ambient temperature < 40 °C / maximum	1000000000000

3.8.2 DTI513 - Leitura e escrita RFID

A unidade de leitura/gravação RFID é usada para ler e escrever diferentes informações nas etiquetas. A parametrização e o intercâmbio de dados são feitos através da interface integrada IO-Link.

Figura 28 – Adaptada de (IFM - DTI513) - Layout do DTI513



Figura 29 – Adaptada de (IFM - DTI513) - Informações técnicas sobre o sensor

Power supply [V]	19,2...28,8 DC
Consumption [mA]	< 50
Protection Class	III
Frequency [MHz]	13,56
RFID Standard	ISO 15693
Distance of TAG [mm]	read : ≤ 60
Communication interface	IO-Link

3.8.3 Tiertime X5

Esta impressora 3D é um modelo profissional capaz de impressão contínua. É um modelo projetado para a produção de protótipos de impressões em 3D na cadeia e onde é possível lançar até 12 placas de construção completas de forma independente.

Tabela 4 – Adaptada de (Tiertime - X5) - Informações técnicas sobre a impressora

Dimensões	850 x 625 x 520 mm
Peso	52 kg
Tecnologia de impressão MEM	Modelização por extrusão em fusão
Diâmetro da base	0.2mm, 0.4mm, 0.6mm, 0.5mm
Velocidade de impressão	200 mm/seg
Volume de impressão	180 x 230 x 200 mm
Resolução	50 - 400 microns
Diâmetro do filamento	1,75 mm
Software	UP Studio
Extensões suportadas	.stl,.obj,.up3,.tsk,.3mf,.ply,.off,.3ds
Conectividade	WI-FI, Ethernet, USB
Temperatura máxima da extrudora	299 °C
Temperatura máxima da cama aquecida	100 °C
Temperatura ambiente	20 - 30 °C
Temperatura de estocagem	0 - 35°C
Alimentação	220 W

Figura 30 – Adaptada de (Tiertime - X5) - Tiertime X5

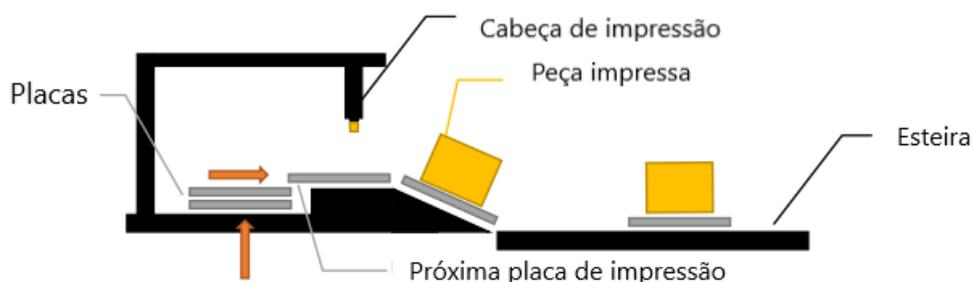


3.9 Metodologia

3.9.1 Sistema original

A impressora 3D em seu modo normal de utilização funciona utilizando o mecanismo de superposição de placas de impressão como pode ser visto na figura 31. Após a finalização da impressão, a segunda placa avança e conseqüentemente empurra a peça impressa localizada sob a primeira placa. Desta forma, tal placa é projetada para fora do campo de impressão do equipamento e a porta de saída é acionada. Finalmente, a peça chega até uma esteira onde será transportada até a próxima etapa do projeto.

Figura 31 – Modo de Funcionamento “Impressão contínua”



A partir do momento que a placa é ejetada para a esteira, a mesma é ativada através de um primeiro sensor óptico e só será desacionada no momento em que a placa

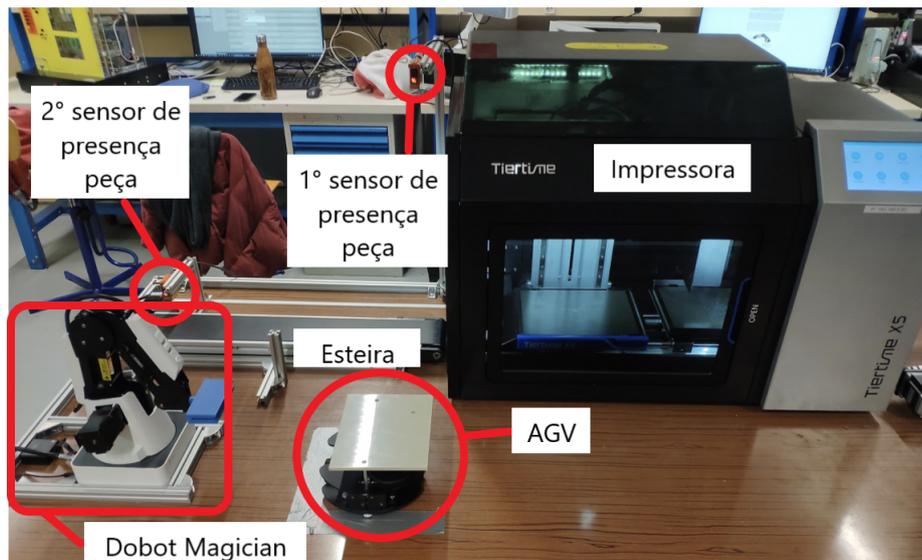
seja detectada por um sensor de proximidade responsável por determinar a posição final da placa e por acionar o robô responsável pelo transporte da peça em questão.

Figura 32 – Modo de Funcionamento “Impressão contínua”



Na figura abaixo, é possível visualizar o sistema impressora, esteira, robô 1 e veículo autônomo.

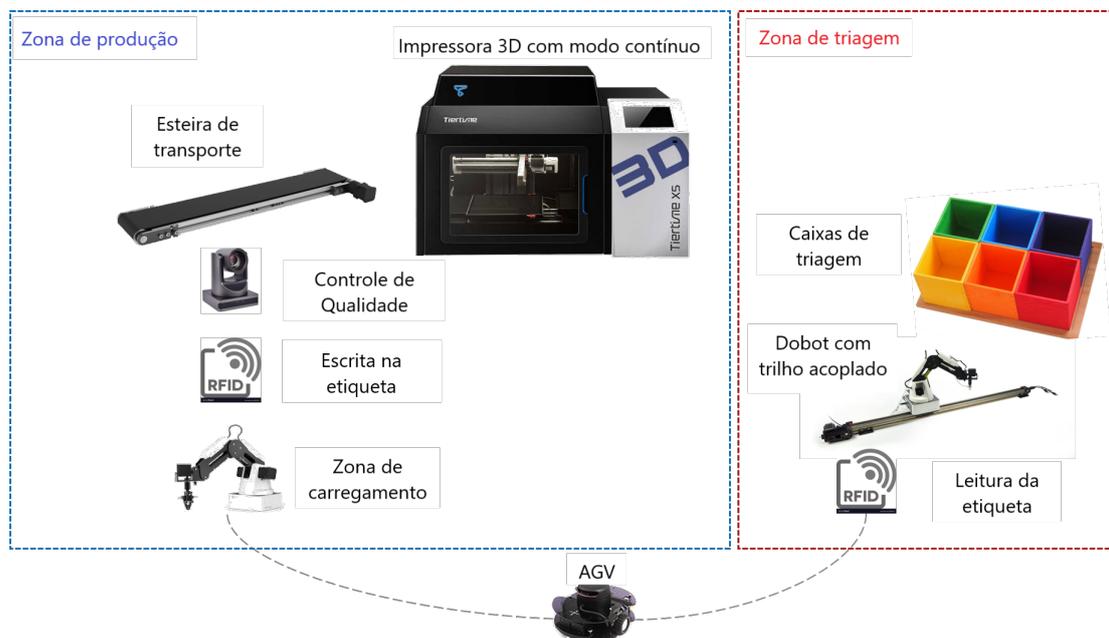
Figura 33 – Impressora, esteira, robô 1 e veículo autônomo



3.10 Proposição final do projeto

O conjunto final do projeto consiste na junção de diferentes sensores, robôs e um veículo autônomo de forma que estes possam trocar informações entre si sem que haja qualquer interação humana durante o processo. O diagrama completo de funcionamento do projeto pode ser melhor explicado através da Figura 34.

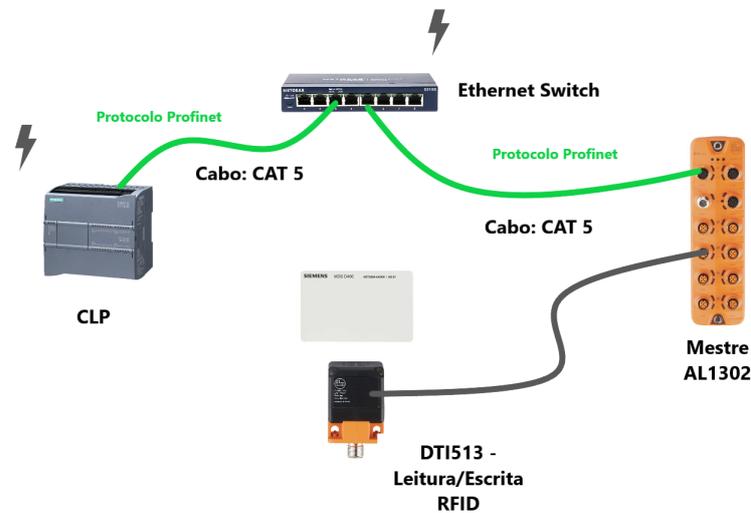
Figura 34 – Diagrama de funcionamento completo do projeto



3.11 Comunicação TAG - CLP

Existem diferentes métodos de implementação, mas foi escolhida a IFM para implementar um novo protocolo de comunicação: IO-LINK. O IO-Link é um padrão de rede de comunicação industrial de curta distância, bidirecional, digital, ponto a ponto, com fio (ou sem fio) (IEC 61131-9) utilizado para conectar sensores e atuadores digitais a um tipo de barramento de campo industrial ou tipo de Ethernet industrial. Seu objetivo é fornecer uma plataforma que possibilite o desenvolvimento e o uso de sensores e atuadores que produzam e consumam conjuntos de dados que, por sua vez, podem ser usados para otimizar os processos e operações industriais automatizados. Um sistema IO-Link consiste em um IO-Link master e um ou mais dispositivos IO-Link, ou seja, sensores ou atuadores. O IO-Link master fornece a interface para o controlador de nível superior (CLP) e controla a comunicação com os dispositivos IO-Link conectados. Um IO-Link mestre pode ter uma ou mais portas IO-Link às quais somente um dispositivo pode ser conectado de cada vez. Este também pode ser um "hub" que permite a conexão de sensores e atuadores de comutação clássicos.

Figura 35 – Rede de comunicação entre o mestre, os sensores e o CLP



3.11.1 Definição das posições e das ações do robô

As etapas a serem feitas pelo Dobot são:

1. Uma peça sai da impressora 3D (sobre uma placa) e é colocada início da esteira devido ao mecanismo interno da impressora.
2. A esteira começa a funcionar e leva a peça até o ponto de detecção de parada desta.
3. Quando a placa com a peça está em frente ao sensor de detecção no ponto de retirada da peça, a esteira para.
4. O braço robótico do Dobot é responsável por retirar a placa da esteira afim de colocá-la na estação de transferência do processo (AGV).
5. O robô deposita a peça em um suporte localizado sob o AGV.

Primeiramente, o robô teve que ser inicializado e programado. Para fazer esta tarefa, foi necessário realizar uma comunicação com o mesmo através de um link serial usando o software Realterm o qual interpreta seu protocolo de comunicação. Posteriormente, foi necessário programar as sequências de posições que o Dobot deveria realizar para completar sua tarefa. Foi preciso utilizar o software **Dobotstudio**, a fim de realizar a familiarização com o robô e estudar os movimentos possíveis. No Dobotstudio, vários métodos de programação são disponibilizados (playback, blockly, script...). O interesse de utilizar tal software foi devido principalmente à necessidade de recuperação das coordenadas de cada posição necessária para recuperar a placa na qual está a peça impressa e movê-la para o AGV.

Figura 36 – Adaptada de (Shenzhen Yuejiang Technology Co 2017) - Interface do software utilizado para obter as posições do Dobot: DobotStudio



Em suma, quatro funções são enviadas ao Dobot:

1. Ativar a esteira.
2. Parar a esteira.
3. Recuperar a placa.
4. Depositar a placa.

Por exemplo, a instrução responsável por fazer o robô recuperar a placa (coordenadas: X:-10.6324 Y: 219.0075 Z: -85.3188 R: 92.7794):

$$AA AA 13 54 03 01 4F 1E 2A C1 CB 01 5B 43 3A A3 AA C2 00 00 00 00 00 9D \quad (3.1)$$

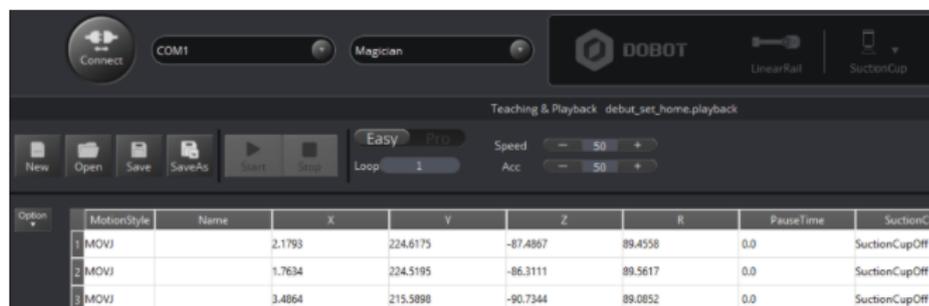
Esse código hexadecimal foi enviado do Realterm ao módulo Wifi conectado ao Dobot. Um outro exemplo: ativar a esteira (diretamente conectada ao Dobot):

$$AA AA 08 87 03 00 01 43 37 00 00 FB \quad (3.2)$$

Dessa forma, é possível obter todos os códigos necessários para cada uma das tarefas.

Graças à função Teaching & Playback, foi possível também obter todos os códigos e testar todas as posições necessárias à realização da tarefa.

Figura 37 – Adaptada de (Shenzhen Yuejiang Technology Co 2017) - Função Teaching e Playback utilizada para obter as coordenadas exatas de cada posição do robô



3.11.2 Lógica de envio das posições ao robô

A metodologia utilizada para enviar os diferentes frames ao robô de acordo com a posição desejada foi a adição de atrasos (delays) entre as diferentes posições garantindo assim que cada frame seja enviado e recebido pelo módulo Wifi acoplado ao Dobot sem que nenhuma informação seja perdida. A sequência de movimentos começa com um retorno à posição inicial e tem duas vantagens principais: de um lado, as coordenadas enviadas após o comando “home” são referenciadas à partir da posição “casa” previamente definida. Por outro lado, tal função permite ao usuário de escolher a posição inicial desejada através do software do equipamento. Para o bom funcionamento do Dobot, é preciso estabelecer a conexão OPC UA e a programação do protocolo UDP utilizando o software Tia Portal a fim de ter o controle do Dobot e da esteira através do CLP.

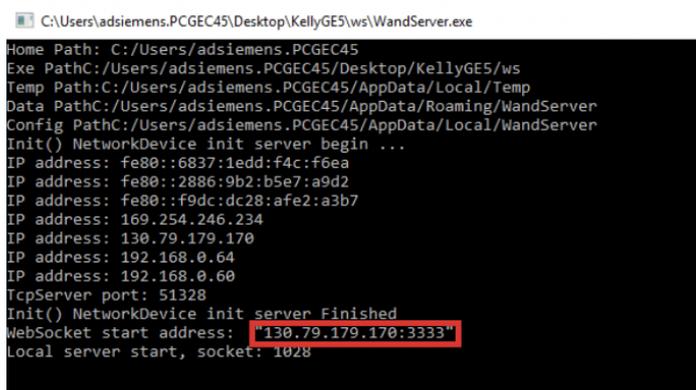
3.12 Aplicativo para impressão das peças

A metodologia para o desenvolvimento da aplicação da impressora seguiu os seguintes passos: Primeiramente, foram estudados os principais protocolos de comunicação atualmente utilizados para a impressão 3D, IPP e Windows 3D, como descrito na introdução. O Internet Printing Protocol (IPP) é um protocolo de impressão baseado em HTTP que, em sua última versão, se descreve como um padrão que permite que computadores pessoais e dispositivos móveis encontrem e imprimam em rede e impressoras USB sem o uso de software específico do fornecedor. A Microsoft oferece suporte nativo para impressão 3D em seus sistemas operacionais e fornece um driver genérico que gera um código G padrão. O código G é uma linguagem que os humanos usam para dizer a uma máquina como fazer algo. Com a impressão 3D, o código G contém comandos para mover peças através da impressora. O arquivo salvo será convertido em código G, a linguagem que a impressora entende e usa para criar uma impressão 3D. Durante a pesquisa, foi descoberto que muitas características da impressão 3D moderna dependem de um formato de arquivo usado para codificar modelos do objeto a ser impresso. No caso do projeto, a impressora X5 3D usa

o formato .tsk. Embora o manual da impressora declare que aceita outros formatos de impressão, como .up3, .ups, .tsk, .stl, .obj, .3mf, .ply, .off, .3ds. O software da impressora transforma todos estes formatos em .tsk antes da impressão.

Este formato é um tipo de arquivo de modelo fatiado criado especificamente para impressoras Tiertime 3D. Pode ser usado como fonte de entrada de dados enviados a impressoras Tiertime 3D compatíveis para imprimir sem passar pelo processo de corte. O arquivo .TSK consiste em três conjuntos de informações, o perfil de impressão, o perfil da impressora e os dados do modelo. Estes representam os três diferentes aspectos do processo de impressão. O perfil de impressão é um conjunto de parâmetros relacionados com as configurações de impressão, tais como: tamanho do bico, tipo de filamento, número do modelo da impressora, etc. Se a impressora alvo for compatível com o perfil da impressora definido no arquivo .TSK, este arquivo pode ser impresso na impressora alvo. Os dados do modelo são os dados do modelo fatiado gerado pelo cortador usando os parâmetros definidos no perfil de impressão. (Introducing The Tiertime Task) Descobriu-se, portanto, que o fabricante tem seu próprio API, seu próprio formato e que eles não estão disponíveis ao público. Enquanto isso, contatou-se a equipe de apoio da Third Party e eles forneceram um kit de ferramentas que ajudou com o desenvolvimento do trabalho. No kit de ferramentas, havia uma documentação pdf para explicar a API, um arquivo chamado wandserver demo.html4 para demonstração e teste e um wandserver.exe4 e seus arquivos auxiliares, o programa em execução converte um PC em um servidor de impressão capaz de controlar várias impressoras através da rede local. As seguintes figuras mostram a página wandserver.exe e demo.html que foram disponibilizadas no kit de ferramentas pela equipe de suporte tiertime.

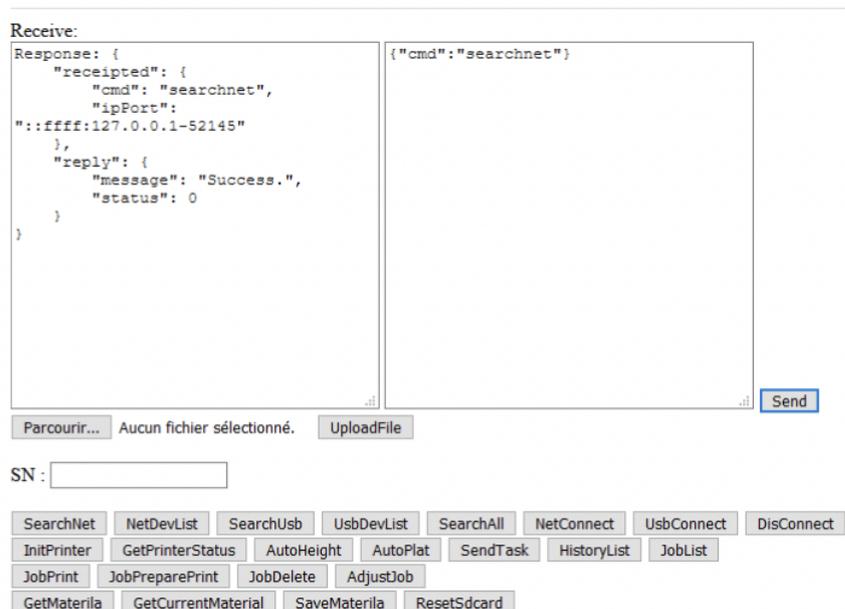
Figura 38 – Servidor responsável pela comunicação com a impressora



```
C:\Users\adsiemens.PCGEC45\Desktop\KellyGE5\WandServer.exe
Home Path: C:/Users/adsiemens.PCGEC45
Exe Path:C:/Users/adsiemens.PCGEC45/Desktop/KellyGE5/ws
Temp Path:C:/Users/adsiemens.PCGEC45/AppData/Local/Temp
Data PathC:/Users/adsiemens.PCGEC45/AppData/Roaming/WandServer
Config PathC:/Users/adsiemens.PCGEC45/AppData/Local/WandServer
Init() NetworkDevice init server begin ...
IP address: fe80::6837:1edd:f4c:f6ea
IP address: fe80::2886:9b2:b5e7:a9d2
IP address: fe80::f9dc:dc28:afe2:a3b7
IP address: 169.254.246.234
IP address: 130.79.179.170
IP address: 192.168.0.64
IP address: 192.168.0.60
TcpServer port: 51328
Init() NetworkDevice init server Finished
Websocket start address: "130.79.179.170:3333"
Local server start, socket: 1028
```

A tela preta acima é a tela enquanto o programa está em execução. A porta destacada em vermelho pode receber comandos de api para controlar a impressora conectada.

Figura 39 – Página demo.html com as funções reconhecidas pela impressora



Para o desenvolvimento da aplicação web, foi escolhido trabalhar com a ferramenta Spring Boot, que é uma estrutura para facilitar o desenvolvimento de aplicações baseadas em Spring e Javascript, oferecendo ferramentas para obter uma aplicação empacotada em frasco, totalmente autônoma. A comunicação entre a aplicação web e a impressora será feita através de um servidor que, através do protocolo de rede WebSocket, envia as instruções da página html. Os WebSockets permitem uma conexão permanente entre o navegador e o servidor. Isto torna cada pedido mais rápido e leve. Como bônus, o servidor pode enviar solicitações ao navegador para notificá-lo sobre novos conteúdos e também enviar notificações (que o conteúdo mudou, uma mensagem chegou) para o navegador no momento em que o evento ocorre. Estas instruções são enviadas usando os protocolos de transporte UDP quando o servidor está tentando se comunicar com a impressora e depois que a conexão é estabelecida, ele muda para TCP. O papel do UDP é permitir a transmissão de dados (na forma de datagramas) de forma muito simples entre duas entidades, cada uma definida por um endereço IP e um número de porta. Não é necessária nenhuma comunicação prévia para estabelecer a conexão e ela usa um modo de transmissão sem conexão. Ao contrário do UDP, o TCP (Protocolo de Controle de Transmissão) é orientado para a conexão. Quando a máquina A (o computador com o servidor) envia dados para a máquina B (a impressora), a máquina B é notificada da chegada dos dados, e acusa o recebimento dos dados com um aviso de recebimento. É aqui que entra a verificação CRC dos dados. Isto é baseado em uma equação matemática, que permite verificar a integridade dos dados transmitidos. Assim, se os dados recebidos forem corrompidos, o protocolo TCP permite que os destinatários peçam ao remetente para enviar os dados corrompidos de volta. A comunicação entre o servidor e a página html é feita pela Socket.

Os soquetes permitem a comunicação entre dois processos diferentes na mesma máquina ou em máquinas diferentes. É uma maneira de se comunicar com outros computadores usando descritores de arquivo padrão Unix. No Unix, toda ação de entrada/saída é feita escrevendo ou lendo um descritor de arquivo. Um descritor de arquivo é um número que identifica de forma única um arquivo aberto no sistema operacional de um computador. Ele descreve um recurso de dados e como ele pode ser acessado. Em resumo, o servidor web é responsável por receber solicitações da página html programada em javascript e enviá-las para a impressora no idioma apropriado que ela seja capaz de interpretar.

Entretanto, algumas das características desenvolvidas neste kit de ferramentas causaram problemas ou não funcionaram adequadamente, como a função de download. Assim, a aplicação, além de integrar, ajustar e concatenar as funções fornecidas pelo fabricante, teve que implementar uma nova função de upload de arquivo. E também a partir das respostas do servidor para lidar com erros, por exemplo, uma impressão não pode ser enviada se a impressora não estiver conectada à tomada, conectada ou inicializada.

4 Resultados e programação

4.1 Recuperação e deslocamento da placa

Uma vez que o segundo sensor envia ao Dobot o comando de parar a esteira, seu braço robótico pode, por sua vez, recuperar a placa contendo a peça. Logo, é preciso implementar os movimentos do Dobot através do Tia Portal.

Neste ponto, os sensores não são capazes de interagir com o Dobot para condicionar as ações. Além disso, os testes de deslocamento múltiplo do Dobot tais como rotação e deslocamento em certas coordenadas necessitam o sequenciamento de seus movimentos. De fato, o movimento da placa não é composto de apenas um deslocamento. Cada sequência N só será válida e ativa quando o controle da sequência N-1 for concluído. Utilizando esta lógica, o braço robótico é capaz de mover a placa com uma taxa de falha extremamente baixa. Além disso, devido à lentidão na recepção/transmissão do módulo Wifi do Dobot devido a muitas trocas na rede interna, foi prudente a inserção de atrasos entre cada sequência a fim de espaçar as instruções enviadas ao Dobot.

Tendo estabelecida a comunicação entre o CLP Siemens e o Dobot através do módulo Wifi e inicializando os parâmetros corretos como mencionado anteriormente, foi possível enviar os frames para o Dobot utilizando funções pré-definidas disponíveis no Tia Portal.

4.2 Configuração e programação Tia Portal

Primeiramente, foi preciso aprender a utilizar o software Tia Portal uma vez que não houve treinamento prévio para este tipo de ferramenta. Graças a diversas pesquisas e o auxílio de pessoas com conhecimento na área, foi possível encontrar algumas respostas sobre como configurar/programar os sensores.

O trabalho foi dividido em quatro partes uma vez que quatro sensores foram utilizados: dois para detecção de distâncias e dois para leitura e escrita da etiqueta (tag).

Todos os sensores estão interligados à partir de um mestre AL1302 usando a tecnologia IO-link. Usando o Tia Portal, a conexão de todos os elementos da rede foi estabelecida.

Figura 40 – Rede entre o PLC e o Mestre



É possível identificar o PLC Siemens à esquerda e o Mestre à direita. Ambos são conectados através de uma conexão Profinet. Também é possível encontrar o endereço IP previamente configurado a fim de se adequar aos dispositivos reais do sistema. O mestre é utilizado para estabelecer a comunicação com os sensores e o PLC é capaz de se comunicar com o Dobot através do módulo Wifi. O sistema é composto, como explicado anteriormente, de dois sensores de distância e dois sensores RFID. Tais elementos foram implementados no software da seguinte forma:

Figura 41 – Configuração de portas do Mestre AL1302 contendo os 4 sensores utilizados no projeto: Port 1 - 1º sensor de presença peça, Port 2 - 1º sensor RFID responsável por escrever na etiqueta, Port 3 - 2º sensor de presença peça e Port 4 - 2º sensor RFID responsável por ler a etiqueta

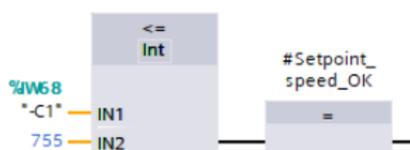
Module	Châssis	Empla..	Adresse I	Adresse...	Type	N° d'article
al1302imp3d	0	0			AL1302	AL1302
X1	0	0 X1			AL1302	
8 Ports_1	0	1			8 Ports	AL1302
IO-Link Master	0	1 1			IO-Link Master	
IO-Link In 16 Byte + PQI	0	1 Port 1	68...84		IO-Link In 16 Byte ...	
IO-Link In/Out 32/32 Byt...	0	1 Port 2	102...134	64...95	IO-Link In/Out 32/3...	
IO-Link In 16 Byte + PQI_1	0	1 Port 3	85...101		IO-Link In 16 Byte ...	
IO-Link In/Out 32/32 Byt...	0	1 Port 4	135...167	96...127	IO-Link In/Out 32/3...	
Disabled_1	0	1 Port 5			Disabled	
Disabled_2	0	1 Port 6			Disabled	
Disabled_3	0	1 Port 7			Disabled	
Disabled_4	0	1 Port 8			Disabled	

Estão disponíveis para a utilização 8 portas no AL1302. As portas 1 e 3 correspondem aos sensores de distância enquanto as portas 2 e 4 são ocupadas pelos sensores de leitura e escrita RFID. É possível ver os diferentes endereços de entrada e de saída correspondentes. O restante das portas foi desativado já que as mesmas não foram utilizadas para o projeto.

4.3 Sensores de distância

Os endereços dos sensores são úteis para que seja possível implementar e utilizar os dados que tais endereços retornam ao Tia Portal. A seguinte função foi criada para utilizar os dados retornados pelos dois sensores:

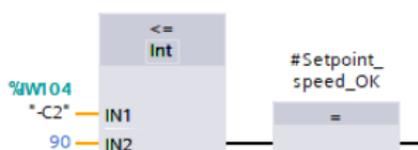
Figura 42 – Função de detecção de peça para o sensor 1 onde IN1 corresponde ao valor de luminosidade lido pelo 1º sensor em tempo real e IN2 é o valor de referência usado para fazer a comparação. Se o valor em IN1 é menor ou igual à IN2, uma variável de controle é acionada para promover o funcionamento de outros processos



Acima tem-se a função usando o sensor 1 (ODS100), o qual será responsável por detectar a saída da placa da impressora e por induzir o transporte da placa com a esteira. Tal sensor tem como condição verdade valores que estejam abaixo de 755. Tal parâmetro foi introduzido experimentalmente durante os vários testes realizados com os instrumentos do projeto.

Durante seu funcionamento, a impressora emite uma placa que será detectada pelo sensor 1 (usando o endereço %IW68). Como saída da função, um número do tipo booleano é recuperado sendo verdadeiro (True) quando uma placa passa pelo sensor 1 ou Falso (False) se não há peça no início da esteira. Quanto ao sensor 2 (OGD592), ele é usado para parar a esteira uma vez que a placa tenha percorrido toda a extensão necessária e assim acionar a fase “movimento Dobot” para pegar e mover a placa com a peça.

Figura 43 – Função de detecção de peça para o sensor 2 - Função de detecção de peça para o sensor 2 onde IN1 corresponde ao valor de luminosidade lido pelo 2º sensor em tempo real e IN2 é o valor de referência usado para fazer a comparação. Se o valor em IN1 é menor ou igual à IN2, uma variável de controle é acionada para promover o funcionamento de outros processos



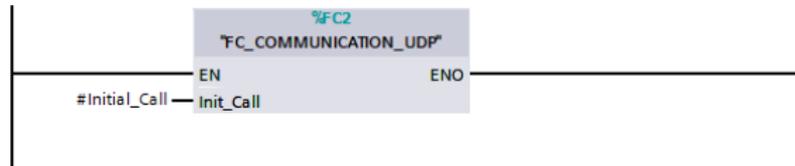
Desta vez, temos o valor de 90, o qual é inferior ao valor limite definido para a função do sensor 1. Isto se deve ao fato de que o sensor 2 foi colocado geograficamente muito mais próximo ao sistema do que o sensor 1.

4.4 Comunicação Dobot

Após a implementação das funções dos sensores, o sistema do Dobot deve ser inicializado. A função abaixo é responsável por inicializar o Dobot para posteriormente

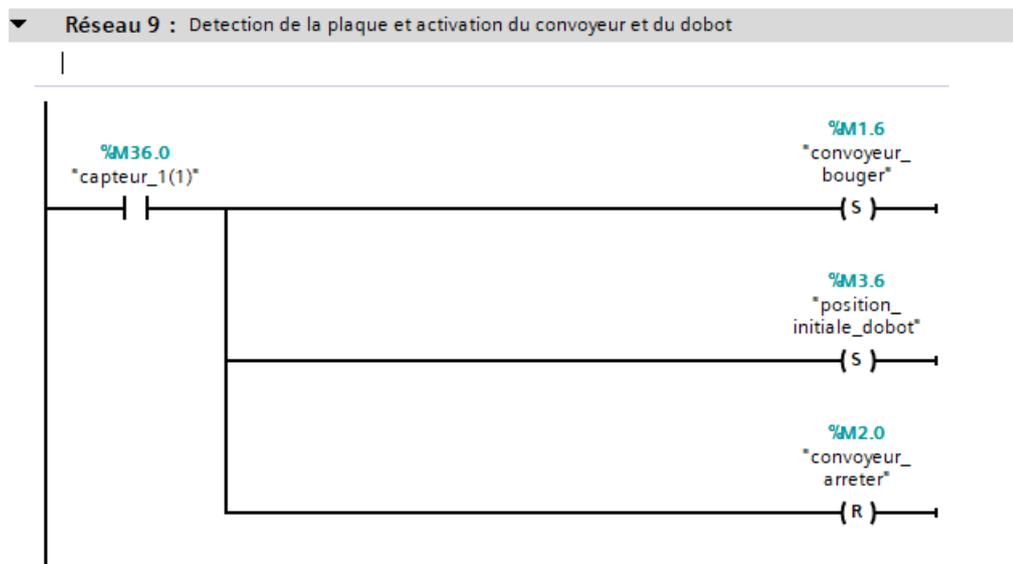
enviar-lhe os frames posteriormente.

Figura 44 – Função para inicializar a comunicação com o Dobot



Tal função permite o estabelecimento da comunicação entre o CLP Siemens e o Dobot usando um módulo Wifi. Assim, será possível enviar os frames contendo as informações de cada posição para o Dobot utilizando funções pré-definidas no Tia Portal. Uma vez feita a inicialização com os parâmetros corretos (endereço IP do Dobot, endereço IP do CLP, número da porta do Dobot, número da porta do CLP, velocidade e sentido da esteira e frames correspondentes), é possível proceder à realização do código para avançar a esteira.

Figura 45 – Rede 9: Se sensor 1 ativado (placa preta sob o mesmo), a variável de controle para ativar a esteira é ativada, a variável para ativar o movimento do robô é ativada e a variável de controle para parar a esteira é desativada



Através desta rede, é possível ativar três variáveis intermediárias: a primeira ativará a esteira transportadora, a segunda pedirá um SetHome ao Dobot e a terceira fará a esteira parar. Cada uma delas será utilizada em diferentes partes do código, sua ativação/desativação sendo apenas feita quando o sensor detecta a presença de uma peça, mudando a variável para verdadeiro.

Figura 46 – Detecção de uma peça pelo sensor 1

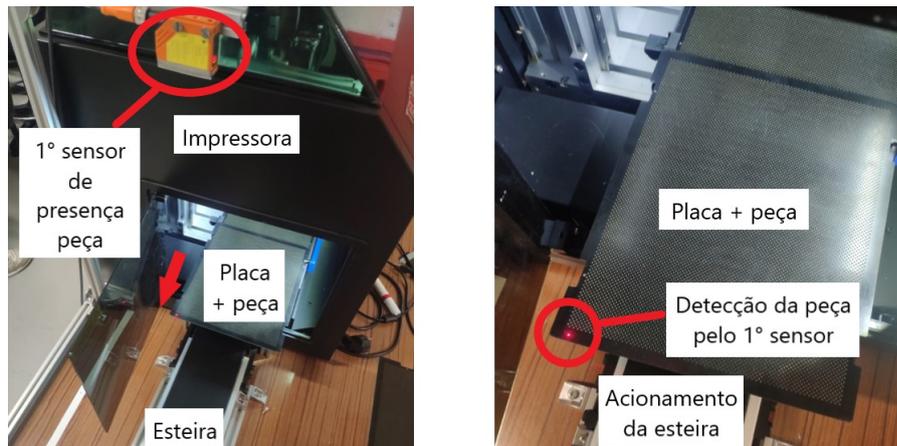
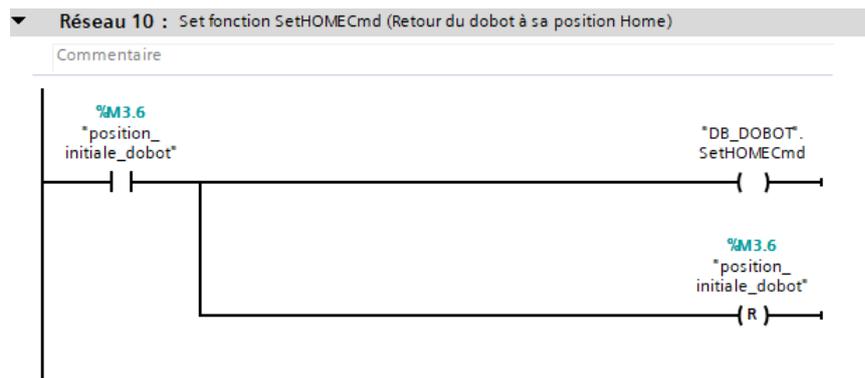
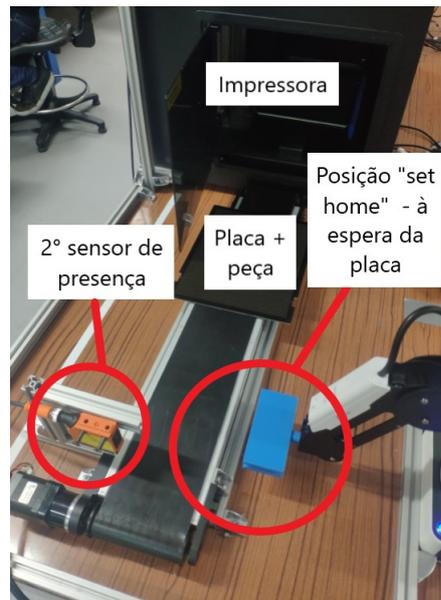


Figura 47 – Rede 10: se a variável de controle responsável por validar a demanda de SetHoçe do robô estiver ativada, um pedido de SetHome será enviado ao robô e a mesma variável de controle é passada a 0



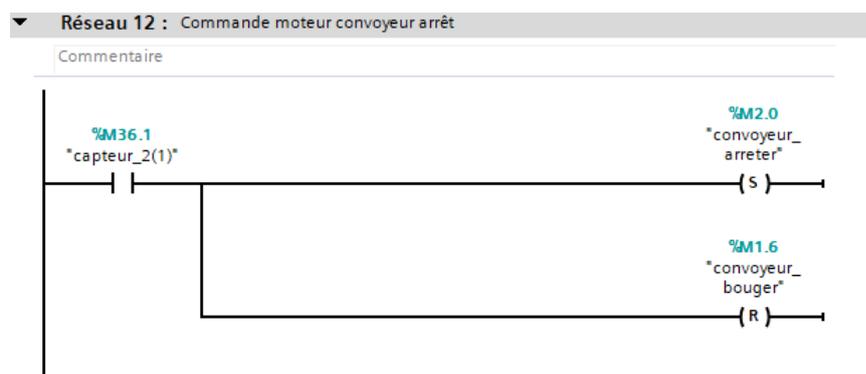
Nesta rede, um Sethome é enviado ao Dobot usando a variável “position initiale dobot” que foi acionada anteriormente de forma que, quando a peça for detectada pelo sensor 2, o Dobot esteja na posição inicial desejada (aguardando pela placa).

Figura 48 – Posição SetHome do robô (à espera da peça)



Acima, através da variável intermediária que foi acionada com a detecção da peça pelo sensor 1, é possível enviar o comando de ativação do sistema transportador usando a variável *convoyeurbouger* (em português: *movimentar esteira*)

Figura 49 – Rede 12: Se sensor 2 ativado (placa preta sob o mesmo), a variável de controle para parar a esteira é ativada e variável de controle para acionar a esteira é desativada



Nesta rede, é possível desativar a operação da esteira transportadora desde que o sensor 2 detecte a chegada da placa no final da extensão desta. Com tal condição estabelecida, a variável de parada a esteira é passada à verdadeira (True) e, conseqüentemente, a variável de parada da esteira é definida como falsa (False).

Para evitar possíveis problemas com relação às variáveis de controle dos equipamentos, foi decidido criar 2 variáveis (ativação e desativação). Desta forma, a cada comando, força-se a variável em questão a 0 ou a 1 e deixa o programa mais bem detalhado.

Figura 50 – Detecção de uma peça pelo sensor 2

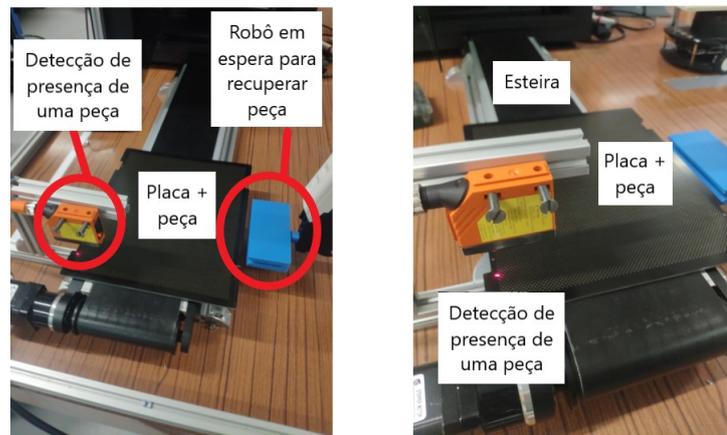
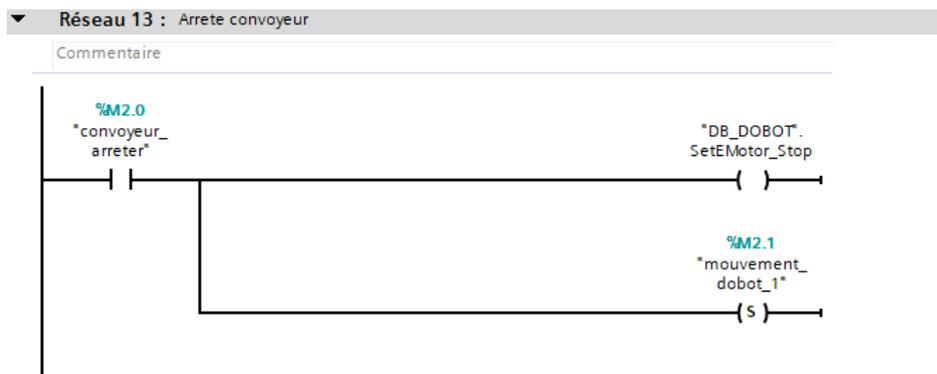


Figura 51 – Rede 13: Se a variável de controle de parada da esteira estiver ativada, um comando é enviado ao robô para parar o motor responsável pelo comando da esteira e a simultaneamente a variável de controle para acionar os movimentos do robô é acionada

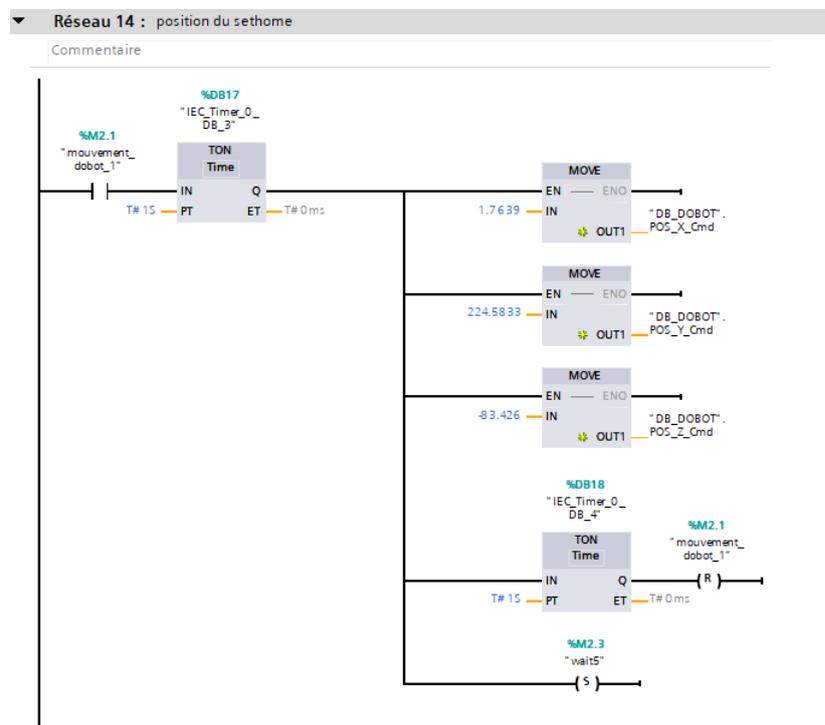


Nesta etapa do código, se a variável intermediária relacionada à parada da esteira transportadora, duas ações são realizadas em paralelo: primeiro, um frame é enviado ao Dobot a fim de parar a esteira. Em seguida, uma variável também intermediária é passada à verdadeira (True), a qual será responsável por ativar o movimento do Dobot para que ele leve a peça para o veículo autônomo.

Figura 52 – Começo da sequência de movimentos depois do SetHome



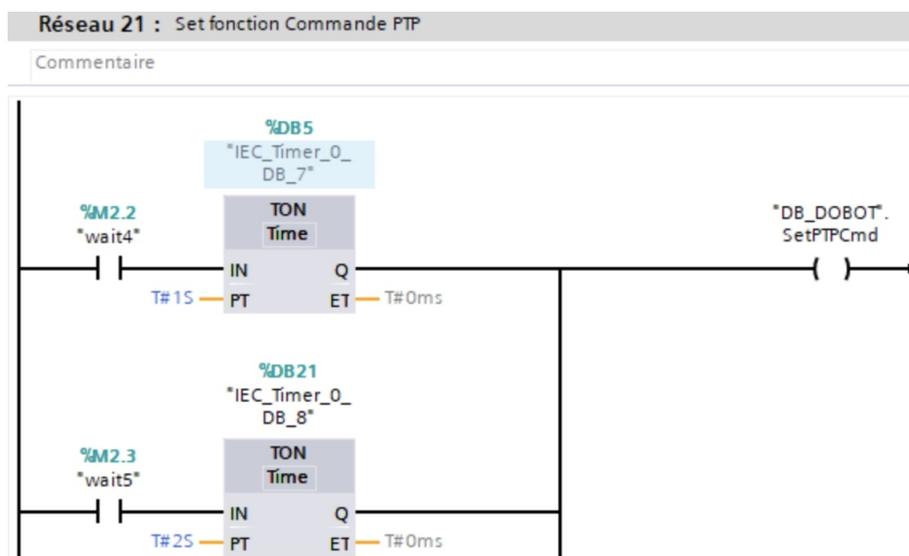
Figura 53 – Rede 14: Se a variável de controle para acionar os movimentos do robô estiver ativada, primeiramente o programa aguarda 1s e posteriormente as posições desejada em x,y e z são enviadas para as variáveis do robô através do protocolo UDP e finalmente, ativa-se uma variável intermediária de controle para poder ativar as próximas mudanças de posição e desativa-se a variável de controle responsável por acionar o primeiro movimento do robô



Implementado no Tia Portal, o movimento do Dobot é realizado utilizando coordenadas no espaço cartesiano (x,y,z). Tendo realizado o SetHome do robô, estas coordenadas correspondem àquelas utilizadas anteriormente nos testes com o DobotStudio. Basta repetir as diferentes sequências para realizar a tarefa. Entre cada ação, são usadas memórias de “espera”. Estes comandos garantem que uma ação só seja ativada quando a anterior for

desativada, com um atraso. Desta forma, a sequência pode ser verificada para garantir que ela esteja funcionando sem interrupções.

Figura 54 – Rede 21 - A cada movimento realizado (utilizando as variáveis auxiliares), o comando SetPTPCmd é acionado. Tal comando é necessário para ativar o movimento do Dobot e escolher seu modo de funcionamento.



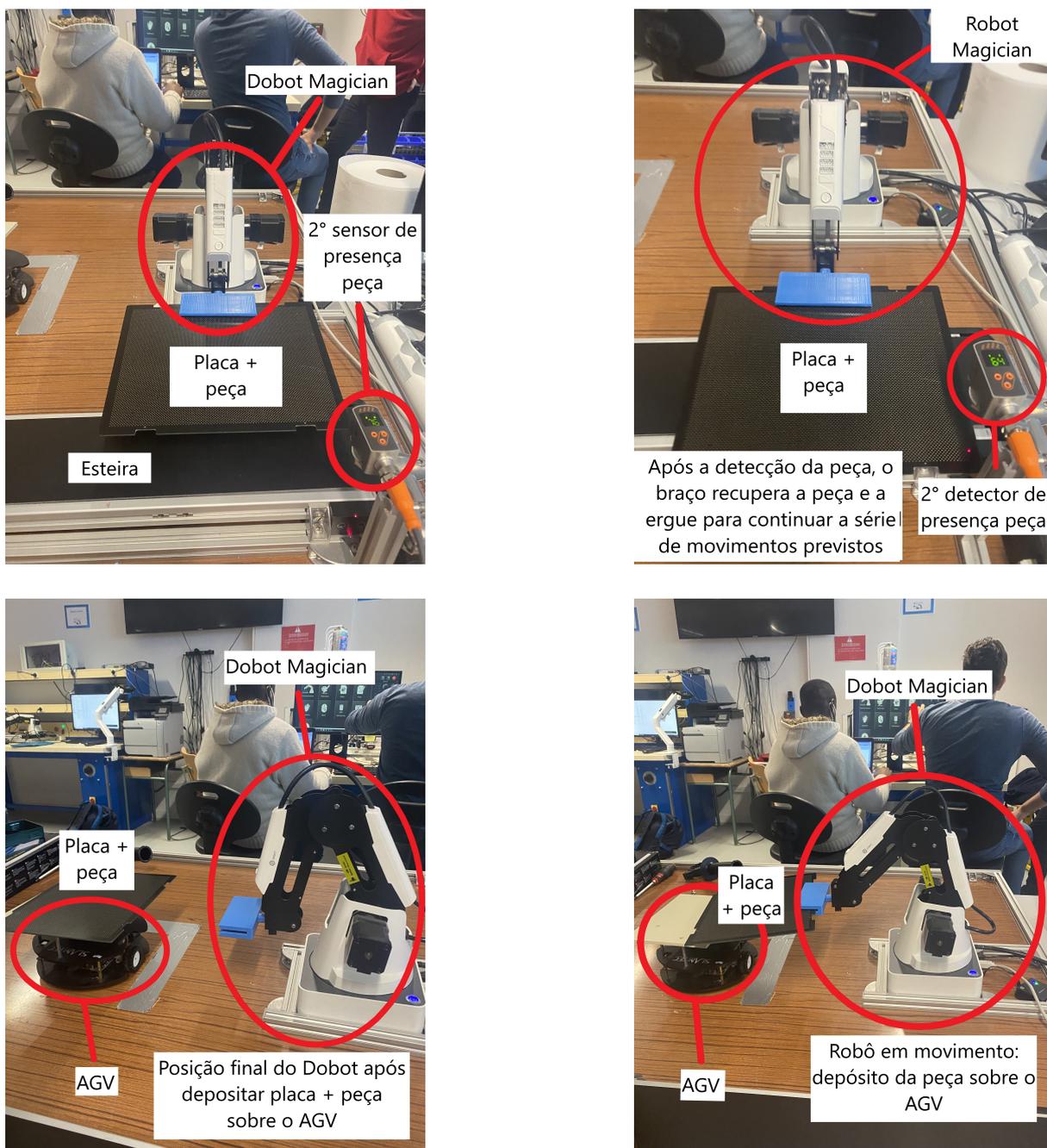


Figura 55 – Detecção da peça pelo sensor que encerra o funcionamento da esteira, Dobot recupera a placa + peça e começa a sequência de movimentos até depositar a peça sobre o AGV.

4.5 Etiquetas RFID

Na página de ajuda da IFM, é possível fazer o download da biblioteca das cabeças de leitura e escrita a fim de utilizar suas funções pré-definidas e adaptá-las ao projeto.

Tal biblioteca é capaz de disponibilizar o bloco de funções DTIxxx. Tal bloco contém um código para facilitar a troca de dados com uma ID-TAG (etiqueta) através do

dispositivo DTI513 no programa de usuário de um Siemens S7-1200 e S7-1500. O bloco de funções foi criado em linguagem SCL através do Tia Portal e para utilizá-lo é necessário arrastá-lo da pasta da biblioteca até a pasta do programa a ser executado.

Figura 56 – Bloco-função DTIxxx para estabelecer a comunicação com o AL1302



Existem 5 modos de operação para o bloco em questão:

- 1 : Leitura do UID da TAG (número de identificação único)
- 2 : Leitura automática dos dados
- 3 : Escrita automática dos dados
- 4 : Leitura dos dados
- 5 : Escrita dos dados

Para realizar os testes e a aplicação do projeto foram utilizados os modos 1,2,4 e 5.

Descrição dos parâmetros de entrada do bloco:

Figura 57 – Adaptada de (IFM 2018) - Parâmetros de entrada do bloco - Parte 1

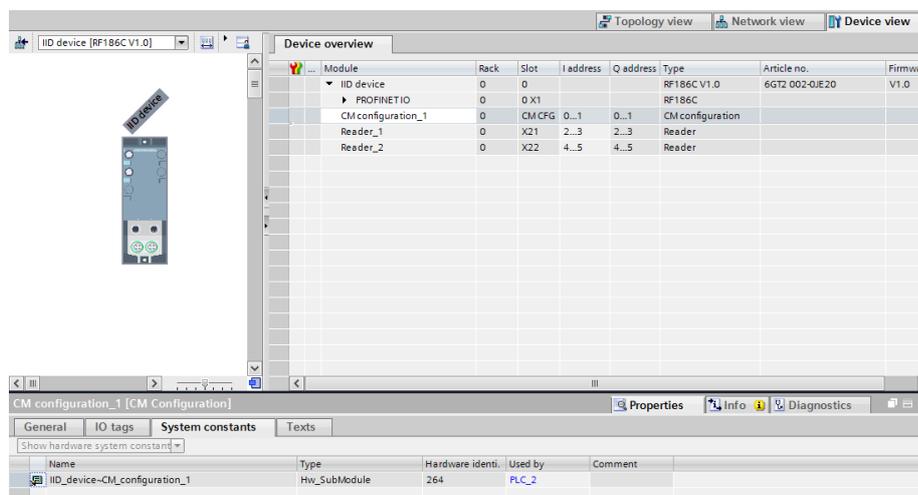
Name	Data type	Description
<i>HW_ID</i>	HW_IO	Hardware Identifier of the DTxxxx device (see chapter 3.2)
<i>DTL_InAddr</i>	UINT	input start address of 32 bytes data from the device (only if parameter <i>HW_ID</i> = 0, see chapter 3.3)
<i>DTL_OutAddr</i>	UINT	output start address of 32 bytes data to the device (only if parameter <i>HW_ID</i> = 0, see chapter 3.3)
<i>Execute</i>	BOOL	activate the operating mode of the DTxxxx function block (see chapter 3.4)
<i>Force</i>	BOOL	force asynchronous read/write again (see chapter 3.5)
<i>Mode</i>	UINT	operating mode of the DTxxxx function block (see chapter 3.4) 1 = read the UID of the ID-TAG automatically 2 = read an area of the USER memory of the ID-TAG automatically 3 = write data to an area of the USER memory of the ID-TAG automatically 4 = read an area of the USER memory of the ID-TAG one time 5 = write data to an area of the USER memory of the ID-TAG one time 15 = switch off the antenna field
<i>TagMemAddr</i>	UINT	relevant only if <i>Mode</i> = 4 or 5: start address in the USER memory of the ID-TAG to read or write data (see chapter 3.6)

Figura 58 – Adaptada de (IFM 2018) - Parâmetros de entrada do bloco - Parte 2

<i>Length</i>	UINT	relevant only if <i>Mode</i> = 4 or 5: number of Bytes to read from or write to the USER memory of the ID-TAG (see chapter 3.6)
<i>WrData</i>	VARIANT	send buffer (data area that contains the data to send) only data type ARRAY OF BYTE is valid (see chapter 3.7)
<i>RdData</i>	VARIANT	receive buffer (data area where to store the read data) only data type ARRAY OF BYTE is valid (see chapter 3.7)

- *HW ID*: Identificador do hardware (dispositivo) do componente DTxxxx. Tal número é responsável por precisar qual equipamento fará os comandos designados. Para determinar esse número, é preciso entrar nas configurações do dispositivo:

Figura 59 – Exemplo de descoberta do parâmetro HW ID



- *DTI InAddr*: Início do endereço de entrada do dispositivo.
- *DTI OutAddr*: Início do endereço de saída do dispositivo.
- *Execute*: Ativa o modo de operação do bloco DTIxxx.
- *Force*: Força de modo assíncrono a leitura/escrita novamente.
- *Mode*: Modo de operação do bloco DTIxxx.
 1. : leitura do UID da etiqueta automaticamente.
 2. : leitura de uma área da memória (USER) da etiqueta automaticamente.
 3. : escrita de dados em uma área da memória (USER) automaticamente.
 4. : leitura de uma área da memória (USER) uma vez.
 5. : escrita de dados em uma área da memória (USER) uma vez.
 6. : desligamento do campo da antena.
- *TagMemAddr*: parâmetro relevante apenas se Mode = 4 ou 5. É o endereço de início na memória USER da etiqueta para leitura ou escrita.
- *Length*: parâmetro relevante apenas se Mode = 4 ou 5. Número de bytes para leitura ou escrita na memória USER da etiqueta.
- *WrData*: envio do buffer.
- *RdData*: recebimento do buffer.

Descrição dos parâmetros de saída do bloco:

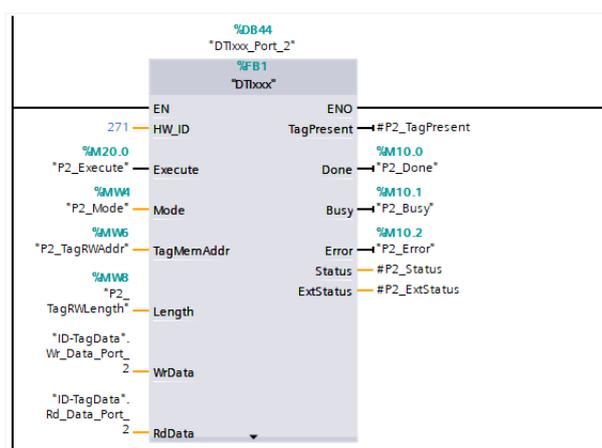
Figura 60 – Adaptada de (IFM 2018) - Parâmetros de saída do bloco

Name	Data type	Description
TagPresent	BOOL	ID-TAG detected in the antenna field (see chapter 3.8)
Done	BOOL	completed without error (see chapter 3.9)
Busy	BOOL	operating mode in process (see chapter 3.9)
Error	BOOL	operating mode abort with error (see chapter 3.9)
Status	WORD	status information or error code (see chapter 3.10)
ExtStatus	WORD	extended status- or error information (see chapter 3.10)

- *TagPresent*: Variável que indica a presença (1) ou ausência (0) de uma ID-TAG (etiqueta) no campo de alcance da antena.
- *Done*: Variável que indica se o processo de leitura e/ou escrita foi finalizado sem erros.
- *Busy*: Variável que indica se o modo de operação escolhido está em progresso.
- *Error*: Variável que indica se o modo de operação escolhido foi abortado devido a um erro.
- *ExitStatus*: Variável que mostra informações estendidas do status ou do erro em questão.

Ao adaptar o bloco segundo as necessidades do projeto, foi possível obter:

Figura 61 – Bloco adaptado segundo as necessidades do projeto



4.5.1 Leitura do UID da TAG

No modo de operação "Reading UID" (Leitura do UID), é lido o UID da TAG (número de identificação único da TAG). Em seguida, o UID está disponível na entrada de dados do processo. Se não houver uma TAG na área de leitura da antena, os 8 bytes do

UID contêm o valor "0x00". Assim que uma TAG é detectada pelo dispositivo, o UID é transferido. Se uma nova TAG for detectada na área de leitura da antena durante o tempo de retenção de dados, o UID da nova TAG é transferido. A tabela abaixo descreve onde os dados do UID são armazenados na memória da TAG:

Figura 62 – Adaptada de (IFM 2018) - Armazenamento da informação contida no UID na memória da TAG

Byte	Process data output	Process data input
0	command value = 0x00	command value = 0x00
1	status	status
2	ignored	UID 0
3	ignored	UID 1
4	ignored	UID 2
5	ignored	UID 3
6	ignored	UID 4
7	ignored	UID 5
8	ignored	UID 6
9	ignored	UID 7
10	ignored	0x00
11	ignored	0x00
12	ignored	0x00
13	ignored	0x00
14	ignored	0x00

Figura 63 – Simulação e visualização da tabela de variáveis se modo de operação = 1, é possível observar que o sensor está em modo de execução, o modo de funcionamento atual é o de número 1, uma etiqueta foi detectada no raio do sensor, é possível saber o tamanho dos dados contidos na etiqueta e o valor armazenado na posição 4 do vetor de dados da etiqueta é 1653

i	Nom	Adresse	Format d'affichage	Valeur visualisation
1	"DTxxx_Port_2".Execute		BOOL	<input checked="" type="checkbox"/> TRUE
2	"DTxxx_Port_2".iMode		DEC	1
3	"DTxxx_Port_2".TagPresent		BOOL	<input checked="" type="checkbox"/> TRUE
4	"DTxxx_Port_2".TagMemAddr		DEC	0
5	"DTxxx_Port_2".Length		DEC	32
6	"DTxxx_Port_2".Error		BOOL	<input type="checkbox"/> FALSE
7	"DTxxx_Port_2".Busy		BOOL	<input type="checkbox"/> FALSE
8	"ID-TagData".Rd_Data_Port_2[4]		Hexa	16#53
9	"ID-TagData".Wr_Data_Port_2[0]		DEC	5
10	"ID-TagData".Wr_Data_Port_2[4]		DEC	0
11	"DTxxx_Port_2".Status		Hexa	16#0000
12		<Ajouter>		

Para simular a operação real do bloco, foi utilizada uma tabela de observação que nos permite observar diretamente o estado em tempo real de nossas variáveis. Ao verificar a linha de número 8 (segundo quadrado vermelho), o valor contido na variável faz referência à uma parte do UID (neste caso, 16#53). A realização da mesma operação de leitura em outras células do array nos dará mais informações sobre o UID.

Podemos extrair como informação que:

- O modo de operação escolhido é o de nº 1
- A TAG está presente no campo de leitura da antena do sensor
- Não há erros na leitura / o sensor não está ocupado
- O byte da posição 4 da matriz da TAG tem como valor 16#53

4.5.2 Leitura dos dados da TAG de forma automática

No modo de operação "Dados de leitura automática" os bytes de 0 a 28 representam os dados na área de memória da TAG de identificação. A área de memória é definida pelos parâmetros "TagMemAdress" e "Length". Os dados na imagem do processo são atualizados assim que uma TAG de identificação entra na faixa de detecção.

Figura 64 – Adaptada de (IFM 2018) - Mapa de informações do modo de escrita automática

Byte	Process data output	Process data input
0	command value = 0x01	command value = 0x01
1	status	status
2	ignored	Data 0
3	ignored	data 1
4	ignored	data 2
5	ignored	data 3
6	ignored	data 4
7	ignored	data 5
8	ignored	data 6
9	ignored	data 7
10	ignored	data 8
11	ignored	data 9
12	ignored	data 10
13	ignored	data 11
14	ignored	data 12

Para simular a operação real do bloco usamos novamente uma tabela de observação de variáveis:

Figura 65 – Simulação e visualização da tabela de variáveis se modo de operação = 2, é possível observar que o sensor está em modo de execução, o modo de funcionamento atual é o de número 2, uma etiqueta foi detectada no raio do sensor e a leitura foi terminada pois a variável Done está ativada

//							
*P2_Execute	%M20.0	BOOL	TRUE	TRUE	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
*P2_Mode	%MW4	DEC	2	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
*P2_TagRWAddr	%MW6	DEC	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
*P2_TagRWLength	%MW18	DEC	29	29	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
//							
*DTxxx_Port_2'.TagPresent		BOOL	TRUE		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
*DTxxx_Port_2'.Done		BOOL	TRUE		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
*DTxxx_Port_2'.Busy		BOOL	FALSE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
*DTxxx_Port_2'.Error		BOOL	FALSE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
*DTxxx_Port_2'.Status		Hexa	16#0000		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
*DTxxx_Port_2'.ExtStatus		Hexa	16#7202		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
//							
*ID-TagData'.Rd_Data_Port_2[0]		Hexa	16#05		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
*ID-TagData'.Rd_Data_Port_2[1]		Hexa	16#00		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Podemos extrair como informação que:

- O modo de operação escolhido é o de nº 2
- A TAG está presente no campo de leitura da antena do sensor
- A leitura foi terminada (variável Done)
- O byte da posição 0 da matriz da TAG tem como valor 16#05 (tal valor foi previamente estipulado e escrito na TAG)

Tal modo será utilizado pelo segundo sensor RFID localizado ao lado da seção de triagem da maquete. A ideia é que tal sensor seja capaz de ler qual o número da peça de acordo com a ordem na qual a mesma foi fabricada e enviar tal informação ao segundo robô para que este leve a peça até a respectiva caixa seguindo a ordem de produção (de 1 a 4).

4.5.3 Escrita dos dados da TAG de forma automática

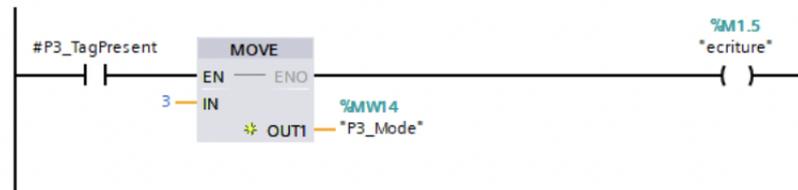
No modo operacional "Escrita automática de dados", os dados a serem escritos são definidos pela imagem de saída dos dados do processo. Os dados são definidos através dos parâmetros TagMemAdress" e "Length". Os dados (incluindo endereço e comprimento) são escritos para a TAG assim que ela entra na faixa de detecção Um máximo de 29 bytes pode ser definido na imagem de saída de dados do processo (byte 0 a 28) Para áreas de memória com um comprimento de dados < 29 bytes, os dados restantes são ignorados e não são escritos na etiqueta Se a escrita for bem sucedida, os dados escritos são espelhados na imagem de entrada de dados do processo. Se a TAG sair do campo de detecção, os dados são mantidos na imagem de processo de acordo com o tempo de retenção de dados. Se o tempo de retenção de dados for excedido e não houver TAG no campo de detecção, os dados são preenchidos com o valor 0x00. Se a escrita não for bem sucedida, o valor do erro é exibido na imagem do processo.

Figura 66 – Adaptada de (IFM 2018) - Mapa de informações do modo de escrita automática

Byte	Process data output	Process data input
0	command value = 0x02	command value = 0x02
1	status	status
2	data 0	data 0
3	data 1	data 1
4	data 2	data 2
5	data 3	data 3
6	data 4	data 4
7	data 5	data 5
8	data 6	data 6
9	data 7	data 7
10	data 8	data 8
11	data 9	data 9
12	data 10	data 10
13	data 11	data 11
14	data 12	data 12

Para poder ativar um modo, é preciso atribuir o número do modo à variável modo do sensor em questão. Para este processo, utilizamos o bloco a seguir:

Figura 67 – Atribuição do modo de operação 3 ao sensor



Para simular a operação real do bloco usamos novamente uma tabela de observação de variáveis:

Figura 68 – Simulação e visualização da tabela de variáveis se modo de operação = 3

	Nom	Adresse	Format d'affichage	Valeur visualisatio
1	//			
2	*P3_Execute	%M12.0	BOOL	TRUE
3	*P3_Mode	%MW14	DEC	3
4	*P3_TagRWAddr	%MW16	DEC	0
5	*P3_TagRWLength	%MW18	DEC	29
6	//			
7	*DTxxx_Port_3.TagPresent		BOOL	TRUE
8	*DTxxx_Port_3.Done		BOOL	FALSE
9	*DTxxx_Port_3.Busy		BOOL	TRUE
10	*DTxxx_Port_3.Error		BOOL	FALSE
11	*DTxxx_Port_3.Status		Hexa	16#7302
12	*DTxxx_Port_3.ExtStatus		Hexa	16#0000
13	//			
14	*ID-TagData".Rd_Data_Port_3[0]		Hexa	16#00
15	*ID-TagData".Rd_Data_Port_3[1]		Hexa	16#00
16	*ID-TagData".Rd_Data_Port_3[2]		Hexa	16#00
17	*ID-TagData".Rd_Data_Port_3[3]		Hexa	16#00
18	*ID-TagData".Rd_Data_Port_3[4]		Hexa	16#00
19	*ID-TagData".Rd_Data_Port_3[5]		Hexa	16#00
20	*ID-TagData".Rd_Data_Port_3[6]		Hexa	16#00
21	*ID-TagData".Rd_Data_Port_3[7]		Hexa	16#00
22	*ID-TagData".Rd_Data_Port_3[8]		Hexa	16#00
23	*ID-TagData".Rd_Data_Port_3[9]		Hexa	16#00
24	*ID-TagData".Rd_Data_Port_3[10]		Hexa	16#00
25	*ID-TagData".Rd_Data_Port_3[11]		Hexa	16#00
26	*ID-TagData".Rd_Data_Port_3[12]		Hexa	16#00
27	//			
28	*ID-TagData".Wr_Data_Port_3[0]		Hexa	16#B7

Podemos extrair como informação que:

- O modo de operação escolhido é o de nº 3
- A TAG está presente no campo de leitura da antena do sensor
- A escrita está sendo feita (variável Done/Busy)
- O valor a ser escrito está na variável do sensor *"ID-TagData".Wr_Data_Port_3[0]* (16#B7) e será escrito na variável *ID-TagData[0]*

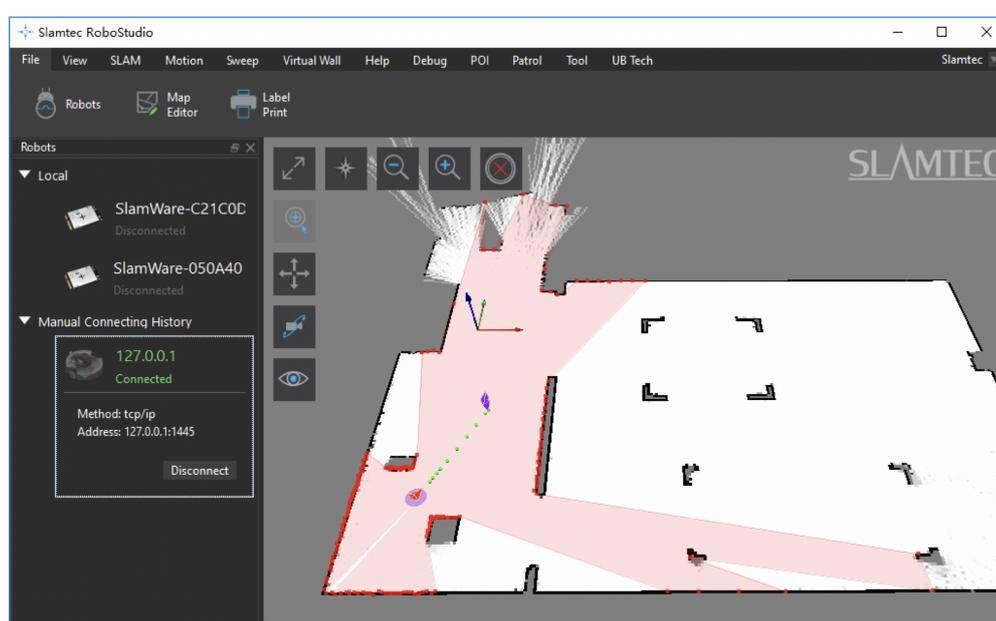
Tal modo será utilizado pelo primeiro sensor RFID localizado ao lado do veículo autônomo. A ideia é que tal sensor seja capaz de escrever qual o número da peça de acordo com a ordem na qual a mesma foi fabricada. Para este processo, a cada vez que houvesse uma TAG presente (ou seja *"DTxxx_Port".TagPresent = TRUE*), um contador seria incrementado de 1 até 4. Da mesma forma, escrever-se-ia o valor atual do contador na TAG a fim de identificar seu número na ordem de produção. Para escrever tal valor, o processo é similar ao utilizado anteriormente para atribuir o número do modo à variável modo do sensor (utilizando o bloco função *Move*).

4.6 AGV - Veículo Autônomo

O AGV tem que ser conduzido utilizando um software dedicado: SLAMTEC RobotStudio. Este software nos permite definir Pontos de Interesse (POIs) usando um

plugin e podemos então automatizar o processo de ir e vir entre esses pontos. O robô irá calcular um caminho, evitando obstáculos para chegar ao seu destino. O mesmo é capaz de carregar as placas entre os dois cabeçotes de marcação RFID. No entanto, ele não pode ser acionado usando um CLP por enquanto, pois os pacotes enviados como dados de posição (foi utilizada a ferramenta Wireshark para realizar a análise de tais informações) não são facilmente legíveis e são enviadas em grandes quantidades, tornando a filtragem difícil. Mesmo que os pacotes fossem identificados, não é garantido que seríamos capazes de emular a comunicação uma vez que não existe um padrão de envio como os protocolos já conhecidos.

Figura 69 – Interface SLAMTEC RobotStudio



4.7 Software para envio de impressões

Como descrito nas especificações técnicas, a impressora tem seu próprio software chamado UP Studio a partir do qual é possível criar o objeto 3D ou baixar um arquivo externo. Só é possível começar a imprimir baixando seu software ou diretamente de um pen drive USB acoplado à impressora. Para tornar a impressora parte de uma cadeia de produção totalmente otimizada e automatizada, o objetivo é desenvolver uma própria aplicação web na qual qualquer pessoa conectada à rede interna da sala possa acessar a impressora 3D através de seu endereço IP. Primeiramente, a impressora foi conectada via Wifi à rede interna do laboratório e sua folha de dados e software foram estudados a fim de se familiarizar com a impressora. Decidiu-se desenvolver uma aplicação web devido a algumas vantagens como: a capacidade de alcançar todos os dispositivos, ou seja, a aplicação é programada para rodar em qualquer sistema operacional, seja iOS, Android ou Windows Phone. Uma aplicação web é capaz de ser executada no navegador

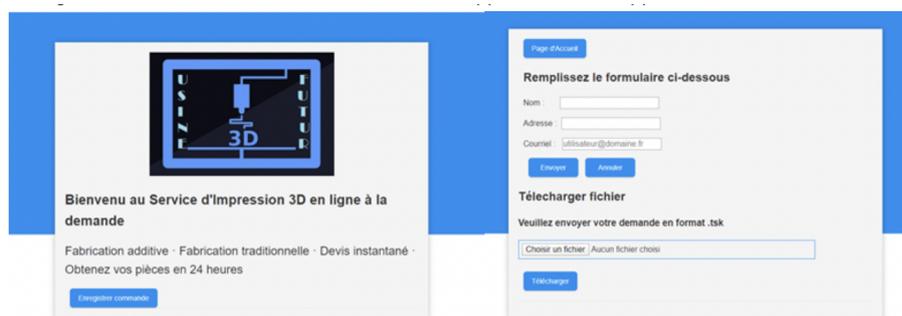
web do dispositivo através de uma simples URL. Além do fato de não precisar ser baixado e instalado de aplicativos como Google Play ou Apple's App Store. Isto se traduz em economia de dinheiro, pois o link direto através de uma aplicação web é gratuito. Também uma API web não precisa ser atualizada como fazem as aplicações atuais.

4.7.1 Resultados

Como as funções fornecidas estavam em Javascript, foi decidido desenvolver a aplicação web usando a ferramenta Spring Boot e continuar nesta linguagem de programação. O trabalho realizado, portanto, consistiu principalmente na coleta e automatização das funções fornecidas pelo fabricante e na ligação entre elas, a fim de facilitar a conexão do usuário e o envio do arquivo para impressão. Também foi criada uma plataforma de download, que armazena os arquivos baixados em uma pasta específica no computador do servidor através do botão "submit"(enviar). O botão de impressão permite escolher o caminho e o nome do arquivo, enviá-lo para a impressora com a função "sendTask"; obtém o ID de impressão e o envia para a função "jobPreparePrint" que realiza o processo de calibração, pré-aquece e inicia o processo de impressão. O resultado final é uma página web estruturada da seguinte forma. Uma página inicial, uma página para o cliente preencher o formulário e enviar o pedido e uma página para o administrador se conectar à impressora e enviar os pedidos para impressão. Na interface do cliente, ele preenche um formulário com seus dados para que mais tarde possa ser implementado na etiqueta RFID para o processo de rastreamento de pedidos. Isto é seguido por uma sugestão de melhoria para projetos futuros. Na página do administrador, ele recupera os arquivos enviados pelo cliente e os envia para impressão. Entretanto, antes de fazer isso, é necessário verificar se a impressora está na rede, conectá-la e rubricá-la. Tudo isso pode ser feito simplesmente clicando no botão "Iniciar". Se ocorrer um problema, ou seja, se a impressora estiver desligada ou se ela e o computador não estiverem conectados à rede, aparecerá uma mensagem na parte superior da tela indicando. Caso contrário, uma mensagem OK aparecerá. A mesma coisa vai acontecer quando você começar a imprimir. Se houver um problema, seja uma porta aberta, se não houver material suficiente para imprimir, ou se a impressão for enviada no formato errado, aparecerá uma mensagem de erro. Se tudo estiver correto, aparecerá uma mensagem indicando o número de identificação do pedido e, posteriormente, que a impressão foi iniciada. o site é acessado a partir do número do endereço IP do computador que contém o servidor.

Os números abaixo mostram o resultado final da aplicação desenvolvida:

Figura 70 – Painel com comandos para controlar a impressora

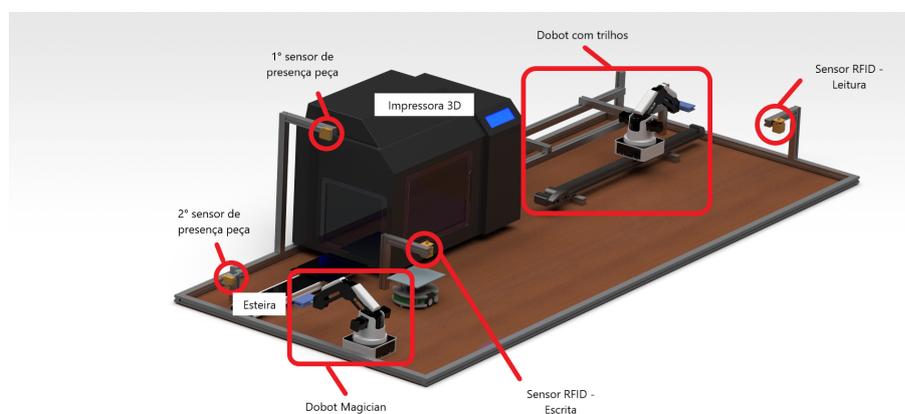


4.8 Digital Twin

Este projeto foi projetado para ter um "gêmeo digital", isto significa que se tem uma versão digital em tempo real a qual representa os objetos físicos.

4.8.1 Visão Geral do Projeto

Figura 71 – Digital Twin - Visualização completa do projeto



4.8.2 Itens desenhados separadamente

Figura 72 – Exemplo de barra utilizada no projeto



As hastes foram redimensionadas e usadas para várias partes do CAD, como também foram usadas para fixar vários itens na versão física do modelo. Elas também delimitam as bordas do modelo. Finalmente são usadas para projetar caixas com a finalidade de receber os itens impressos em 3D classificados.

Figura 73 – Mesa do projeto no CAD



Figura 74 – AGV no CAD

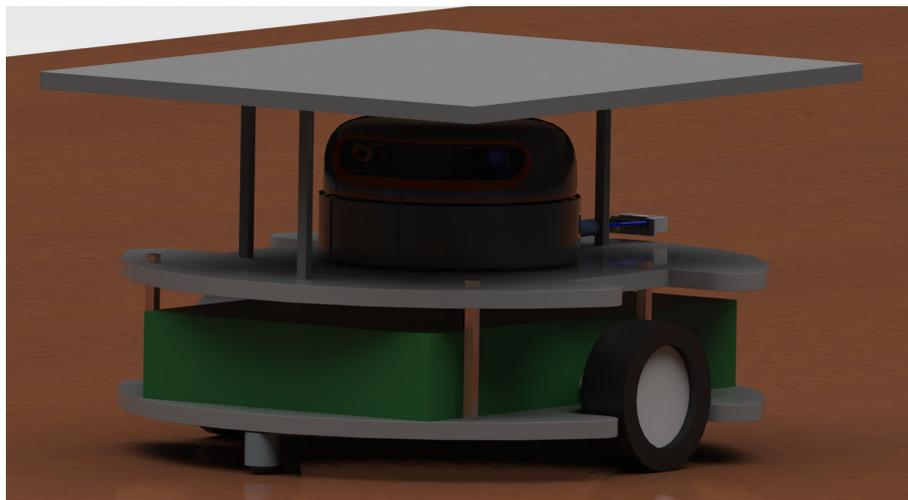


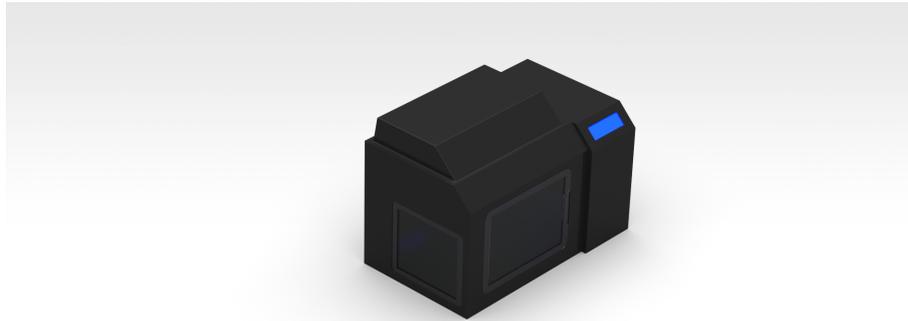
Figura 75 – Sensores IFM no CAD



Figura 76 – Trilho Linear no CAD



Figura 77 – Impressora 3D no CAD



4.8.3 Conexão com NX e Tia Portal

4.8.3.0.1 OPC UA

Para que o NX receba dados, os dados são armazenados em um servidor usando OPC UA. Os dados podem ser acessados por qualquer máquina logada na mesma rede. Pode-se enviar valores de sensores, bem como posições Dobot para tornando o Digital Twin para a versão física um servo. Logo, tem sido possível de recuperar dados dos sensores de distância e parte dos movimentos do Dobot. A versão digital 3D é capaz de copiar os primeiros 20 segundos da versão física do projeto.

Um código separado baseado no usado no modelo físico foi escolhida por algumas razões:

- Perturbações caso fosse necessário trabalhar no OPC UA enquanto outra pessoa já estava trabalhando no modelo
- O OPC UA é designado para CLPs com versões acima de 1500. No entanto, a versão de CLP usada para construir o código foi 1200.
- Testar o Digital Twin significava testar no modelo em tempo real, e tal ação não poderia ser feita sem utilizar o tempo útil para realizar melhorias no modelo real. Também, isso poderia significar cometer erros no código do TIA Portal ao criar variáveis e sinais que retardassem ainda mais o processo.

4.8.3.0.2 Sinais

Vários sinais são importados para o modelo NX :

- Valores dos sensores de distância : estes permitem controlar a esteira transportadora, assim como o Dobot

- Leitores RFID de presença de tags : estes permitem o controle do AGV e a comutação para o Dobot montado sobre trilhos
- Valores de ângulo conjunto para duas posições do Dobot: estes permitem o controle do Dobot para as duas primeiras posições. Como o Dobot é controlado usando coordenadas cartesianas no programa TIA Portal, não é fácil obter valores conjuntos e isso tem sido feito obtendo os valores conjuntos de acordo com o DobotStudio e enviando-os simultaneamente.

4.8.3.0.3 Equações de Runtime

Usando as equações de Runtime do NX, é possível criar macros e condições ao invés das operações usuais. É possível copiar um valor de sinal (vindo do software TIA Portal) para um valor no Digital Twin, como o movimento da esteira transportadora ou a posição Dobot. O uso do Runtime (tempo de execução) coloca vários problemas: o controle não é sequencial e às vezes a sobreposição ou condições não utilizadas podem resultar em movimentos erráticos.

4.8.3.0.4 Operações

As operações são usadas para mover a esteira transportadora. No entanto, quando a placa está entre os dois sensores de distância, é impossível notar a diferença entre quando ela não está em nenhum lugar fazendo com que o tempo de funcionamento não consiga iniciar e parar a esteira na hora certa. Logo, o Digital Twin é capaz de copiar os primeiros 20 segundos da versão física usando dados em tempo real. A questão principal então reside na posição Dobot, pois os dados são enviados usando coordenadas X,Y,Z no Portal TIA e a versão NX é movida usando ângulos das juntas. É também impossível buscar a posição atual do Dobot no TIA Portal. O protocolo implementado funciona em malha aberta. Outra questão residia na forma como os dois braços principais do Dobot se movem juntos, a ligação mecânica é complexa e portanto não tem sido possível de mover as juntas de forma completamente livre. Finalmente, o Dobot não tem uma liberdade de rotação de 360 graus, e o controle de posição tem que usar uma direção para a rotação. Como teste, foram utilizados limites de ângulo e tal ação não impediu o Dobot CAD de ir aonde o real não pode.

5 Desafios encontrados e melhorias

Esta seção foi escrita principalmente como uma orientação para esclarecer as principais melhorias a serem feitas :

5.0.1 Melhorias gerais do código TIA Portal

- OPC UA só está disponível em CLPs da série 1500, o código está atualmente em um CLP 1200 e não permite o controle síncrono no NX MCD. O NX twin usa um código 1500 , semelhante à versão principal.
- O Dobot é acionado atualmente usando posições X,Y,Z no Portal TIA, e tem que ser acionado usando os valores conjuntos no NX MCD. O código do Portal TIA deve ter um conversor, semelhante ao que o Dobot Studio tem que enviar o valor das juntas no OPC UA. A solução mais fácil é enviar os valores das juntas para cada vez que uma coordenada X,Y,Z é enviada na TIA usando o conversor integrado do DobotStudio.
- O código UDP também funciona em circuito aberto e aparentemente não está buscando a posição atual do Dobot e seria muito interessante tentar usá-lo como escravo ou pelo menos buscar posições para alimentar o NX MCD.
- Ao mudar de um Dobot para outro no TIA (trocando IPs quando uma etiqueta RFID é lida no final da área de classificação do modelo), o nome do Dobot pode ser levado em consideração.

5.0.2 Dobot

- A posição do Dobot Linear Rail não é monitorada por nenhum sensor, e é muito difícil de controlar em loop totalmente aberto, pois o código UDP TIA Portal permite apenas um controle binário (stop and go). Um sensor de distância deve ser instalado.
- O Dobot às vezes vai muito longe em uma de suas articulações e precisa ser movido manualmente, isto pode ser removido pelo sensor modyfing e opções de base no DobotStudio.

5.0.3 IFM

- As etiquetas RFID são volumosas e não cabem sob uma das placas, é possível usar etiquetas RFID de papel. Elas precisam ser lidas a uma distância suficiente,

pois a escolha das placas em vez dos outros tipos de sensores foi impulsionada principalmente por esta característica.

5.0.4 NX Digital Twin

- O modelo 3D do Dobot montado sobre trilhos poderia incluir a corrente de cabo, é provável que seja difícil de animar mesmo que haja uma opção de elo de corrente no software.
- Tente buscar ângulos de junção automaticamente da TIA ao invés de X,Y,Z para escravizar o Dobot
- Se o Dobot for devidamente escravizado, a AGV pode ser movida usando as variáveis de presença das etiquetas dos leitores RFID e a animação não deve ser um problema.
- O movimento do eixo Dobot J1 é problemático, pois não conseguimos definir limites de ângulo e a rotação tem que ser definida como sentido horário, sentido anti-horário ou o caminho mais curto. Isto às vezes resulta no Dobot CAD para seguir o caminho errado para chegar a um destino.
- Melhorar o modelo da impressora 3D para permitir que as placas venham de dentro
- O modelo AGV poderia ter rodas de funcionamento adequado, mas não é um problema fácil no NX e um mecanismo real tem que ser construído. Ele também poderia se parecer mais com a versão física.

5.0.5 AGV

- O AGV deve ser acionado de forma otimizada usando o PLC, a WireShark nos permite monitorar os pacotes e deve ser possível enviar os pacotes necessários para escravizar o robô para os sensores RFID. Estes pacotes parecem difíceis de pegar, pois o número de pacotes enviados é alto e alguns dos pacotes não são legíveis.

6 Conclusão

Através deste projeto, foi possível entender a importância do processo de automação e a inter-relação com relação a uma linha de produção. Ao estabelecer uma comunicação entre diferentes sensores, atuadores e interfaces, garante-se um funcionamento sem interferência humana e portanto com menor probabilidade de erros.

Foi possível colocar em prática os conhecimentos na área de automação com relação à programação das etapas de detecção e deslocamento da peça através do software TiaPortal utilizando a linguagem ladder. Além disso, foi possível escrever uma documentação técnica responsável por orientar e detalhar o passo a passo da instalação dos sensores e do estabelecimento da comunicação com o PLC.

Com o estabelecimento do protocolo UDP para a comunicação com os Dobots, foi possível compreender o funcionamento do envio e da recepção de frames de acordo com o movimento desejado e o tratamento feito pelo PLC de forma a gerenciar e traduzir tais demandas em forma de comandos de saída.

De forma geral, foi possível ver todo o projeto de forma detalhada tanto na parte mecânica na parte de programação, seja para uma aplicação ou no software TiaPortal. Como resultado, tem-se uma linha de produção automatizada. Este projeto me permitiu adquirir habilidades no campo da Indústria 4.0, particularmente no campo da automação.

Por outro lado, tal projeto permitiu o crescimento em autonomia, em capacidade, em curiosidade e especialmente em adaptabilidade. Por muitas vezes aquilo que foi planejado ou estudado não foi bem-sucedido ou não pôde ser implementado devido a limitações. Portanto, sempre foi necessário procurar diferentes soluções para o mesmo problema.

Este projeto representa uma abertura para o mundo profissional uma vez que este simula uma pequena etapa da produção industrial em uma fábrica.

Com relação aos benefícios do projeto, a redução da intervenção humana dentre os processos de fabricação e triagem de cada peça fideliza as etapas e diminui a porcentagem de erros humanos. Além disto, formou-se um sistema completamente independente e fechado que é capaz de estabelecer a comunicação entre diferentes máquinas e equipamentos que funcionam de maneiras distintas. O estabelecimento da comunicação UDP entre robô e CLP foi importante uma vez que um bloco função foi criado na interface TIA Portal sendo este possível de ser utilizado em outras aplicações e em outros campos visto que ele segue um padrão e que foi desenvolvido pensando na possibilidade de ser replicado.

O desenvolvimento dos sensores de leitura e escrita foi crucial uma vez que a implementação de ambas as ações foi bem-sucedida. Anteriormente, só era possível realizar

a leitura e em modo automático. Agora, com o controle de todas as funções, é possível escolher quando e qual tipo de ação cada um dos sensores fará.

A descoberta da não-padronização no envio dos pacotes do AGV à rede conectada permitiu que o Instituto se adiantasse na aquisição de um novo componente que seja compatível com as necessidades do projeto.

Referências

- BERGE, J. *Fieldbuses for Process Control: Engineering, Operation and Maintenance*. [S.l.]: ISA – The Instrumentation, Systems and Automation Society, 2002. Citado na página 11.
- BRANQUINHO, M. et al. *Segurança de Automação Industrial e SCADA*. Elsevier Editora Ltda., 2014. ISBN 9788535277876. Disponível em: <<https://books.google.fr/books?id=FVkaBQAAQBAJ>>. Citado na página 10.
- BUDAMPATI, R.; KOLAVENNU, S. *Industrial Wireless Sensor Networks: Monitoring, Control and Automation*. Elsevier Science, 2015. (Woodhead Publishing Series in Electronic and Optical Materials). ISBN 9781782422372. Disponível em: <<https://books.google.fr/books?id=pEadBAAAQBAJ>>. Citado na página 12.
- DEY, C.; SEN, S. *Industrial Automation Technologies*. CRC Press, 2020. ISBN 9781000068764. Disponível em: <https://books.google.fr/books?id=yH_oDwAAQBAJ>. Citado na página 10.
- DUAN, K.-K.; CAO, S.-Y. Emerging RFID technology in structural engineering – a review. *Structures*, Elsevier BV, v. 28, p. 2404–2414, dez. 2020. Disponível em: <<https://doi.org/10.1016/j.istruc.2020.10.036>>. Citado na página 1.
- FRANCHI, C.; CAMARGO, V. de. *Controladores Lógicos Programáveis: Sistemas Discretos*. Saraiva Educação S.A. ISBN 9788536525167. Disponível em: <https://books.google.fr/books?id=_LZiDwAAQBAJ>. Citado 5 vezes nas páginas ix, 17, 18, 19 e 20.
- FREQUENCY Table. <https://www.researchgate.net/figure/APPLICATION-FOR-DIFFERENT-TYPES-OF-FREQUENCY-RANGE_tbl1_275037870>. Citado 2 vezes nas páginas xiii e 25.
- IFM. *Notice d'utilisation - Tête de lecture/écriture RFID avec IO-Link*. [S.l.], 2018. Citado 6 vezes nas páginas xii, 59, 61, 62, 63 e 65.
- IFM - AL1302. <<https://www.ifm.com/fr/fr/product/AL1302?tab=documents>>. Acessado em: 2022-02-12. Citado 2 vezes nas páginas x e 32.
- IFM - DTI513. <https://www.ifm.com/fr/fr/search?query=dti513&searchSessionId=1646322564244.h31vsxk#!/product?_type=product>. Citado 3 vezes nas páginas x, 37 e 38.
- IFM - O5D100. <<https://www.ifm.com/fr/fr/product/O5D100>>. Citado 2 vezes nas páginas x e 33.
- IFM - OGD592. <<https://www.ifm.com/fr/fr/product/OGD592>>. Citado 2 vezes nas páginas x e 33.
- INTRODUCING The Tiertime Task. <[https://www.tiertime.com/introducing-the-Tiertime-Task/.](https://www.tiertime.com/introducing-the-Tiertime-Task/)> [Online; acessado em 19-11-2020]. Citado na página 45.

- IO-LINK. <<https://www.profibus.org.br/noticia/io-link-entenda-melhor-sobre-a-tecnologia>>. Acessado em: 2022-02-12. Citado na página 15.
- ISLAM, M. R. et al. Cartesian trajectory based control of dobot robot. Citado na página 33.
- LI, X. et al. Value creation during fourth industrial revolution: Use of intellectual capital by most innovative companies of the world. *Technological Forecasting and Social Change*, Elsevier BV, p. 120479, nov. 2020. Disponível em: <<https://doi.org/10.1016/j.techfore.2020.120479>>. Citado na página 1.
- LUGLI, A. *Redes Industriais para Automação Industrial: AS-I, Profibus e Profinet*. Saraiva Educação S.A. ISBN 9788536529134. Disponível em: <<https://books.google.fr/books?id=0YuwDwAAQBAJ>>. Citado 3 vezes nas páginas ix, 9 e 10.
- MALIK, P. K. et al. Industrial internet of things and its applications in industry 4.0: State of the art. *Computer Communications*, Elsevier BV, nov. 2020. Disponível em: <<https://doi.org/10.1016/j.comcom.2020.11.016>>. Citado na página 1.
- MILLER, P. *TCP/IP: The Ultimate Protocol Guide*. Brown Walker Press, 2009. (TCP/IP: The Ultimate Protocol Guide : Complete 2 Volume Set, vol. 1). ISBN 9781599424910. Disponível em: <<https://books.google.fr/books?id=Ts4SKa6qLLYC>>. Citado 3 vezes nas páginas ix, 13 e 14.
- MOTA, R. F. d. S. *Sensor de distância por infravermelhos para a caracterização do espaço de trabalho*. Tese (Doutorado), 2018. Citado na página 32.
- NEIVA, J. P. B. et al. localização e orientação "indoor" com recurso à tecnologia rfid. 2012. Citado 2 vezes nas páginas 25 e 26.
- OLIVEIRA, V. d. S. L. *PROTOCOLO DE COMUNICAC AO PROFINET PARA REDES DE AUTOMAC AO*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2016. Citado na página 32.
- PAN, X.; CHEN, K.; CHEN, D. Development of rapid prototyping slicing software based on stl model. In: *Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. [S.l.: s.n.], 2014. p. 191–195. Citado na página 6.
- PUPO, M. S. *Interface homem-máquina para supervisão de um CLP em controle de processos através da WWW*. Tese (Doutorado) — Universidade de São Paulo, 2002. Citado na página 30.
- ROGGIA L, F. R. C. *Automação Industrial*. [S.l.]: Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria, Rede e-Tec Brasil, 2016. Citado 5 vezes nas páginas ix, x, 22, 23 e 24.
- ROSARIO, J. M. *Automação Industrial*. [S.l.]: Editora Baraúna, 2009. Citado 2 vezes nas páginas 8 e 30.
- SDP Mini Manual. <<https://wiki.slamtec.com/display/SD/KBSW180119+SLAMWARE+SDP+Mini+User+Manual>>. Citado 2 vezes nas páginas x e 36.

- SEZEN, B. Modeling automated guided vehicle systems in material handling. *Doğuş Üniversitesi Dergisi*, v. 4, n. 2, p. 207–216, 2011. Citado na página 35.
- SHENZHEN YUEJIANG TECHNOLOGY CO. *Dobot Magician User Manual*. [S.l.], 2017. Citado 6 vezes nas páginas x, xiii, 34, 35, 43 e 44.
- SIEMENS. *SIMATIC - Tia Portal - Step 7 Basic V10.5*. [S.l.], 2009. Citado 2 vezes nas páginas x e 31.
- SIEMENS. *S7-1200 Programmable Controller - System Manual*. [S.l.], 2012. Citado 2 vezes nas páginas x e 30.
- SIEMENS. *MDS 400 - Datasheet*. [S.l.], 2020. Citado 2 vezes nas páginas x e 37.
- SILVA, E. da. *Introdução às linguagens de programação para CLP*. [S.l.]: Editora Blucher, 2016. ISBN 9788521210528. Citado 4 vezes nas páginas ix, 16, 21 e 22.
- STEMMER, M. *Sistemas Distribuídos e Redes de Computadores para Controle e Automação Industrial*. [S.l.]: Universidade Federal de Santa Catarina, 2001. Citado na página 11.
- SUN, P. Security and privacy protection in cloud computing: Discussions and challenges. *Journal of Network and Computer Applications*, Elsevier BV, v. 160, p. 102642, jun. 2020. Disponível em: <<https://doi.org/10.1016/j.jnca.2020.102642>>. Citado na página 1.
- SUNG, T. K. Industry 4.0: A korea perspective. *Technological Forecasting and Social Change*, Elsevier BV, v. 132, p. 40–45, jul. 2018. Disponível em: <<https://doi.org/10.1016/j.techfore.2017.11.005>>. Citado na página 1.
- TANEMBAUM A. S., S. M. V. *Distributed Systems – Principles and Paradigms*. [S.l.]: Prentice Hall, 2002. Citado na página 11.
- TECNOLOGIAS de Impressão. <<https://www.inovahouse3d.com.br/post/blog-tecnologias-de-impressao-3d>>. Citado 2 vezes nas páginas 5 e 6.
- TIERTIME - X5. <https://www.tiertime.com/wp-content/uploads/2019/12/X5_User_manual_2019-3-21.pdf>. Citado 4 vezes nas páginas x, xiii, 38 e 39.
- VANTAGENS IO-LINK. <<https://www.eletronor.com.br/blog/io-link-a-porta-de-entrada-para-a-industria-conectada/>>. Acessado em: 2022-02-12. Citado na página 15.
- VICTORIA, I. M. B. *Medição da Radiação Solar Global e Difusa Utilizando Piranômetros com Sensores Fotoelétricos*. Dissertação (Mestrado) — Universidade Federal de Pelotas, 2008. Citado na página 32.
- VOLPATO, N. *Manufatura aditiva: tecnologias e aplicações da impressão 3D*. Editora Blucher, 2017. ISBN 9788521211518. Disponível em: <<https://books.google.fr/books?id=ni9dDwAAQBAJ>>. Citado 4 vezes nas páginas ix, 4, 7 e 8.
- ZHU, H.; GU, H.; YANG, Y. Identify obstacles and grab items by robot arms. Citado na página 34.

Anexos

ANEXO A – Tutorial UDP

Programmation UDP avec API Siemens

Dans ce tutoriel

- 1 Matériel nécessaire et branchements
- 1 Travail préparatoire: mise en place du réseau Ethernet
- 3 Etape 1: Création de la configuration Hardware du projet
- 5 Etape 2: Chargement dans l'API de la configuration matériel
- 7-17 Etape 3: Création du programme utilisateur

ATTENTION: Il faut impérativement avoir l'API Siemens et le partenaire UDP sur le même sous réseau Ethernet

Introduction

Ce tutoriel a pour but d'aider à la programmation d'une communication UDP sur un automate Siemens (S7-1200 et S7-1500).

Matériel nécessaire et branchements

Matériel nécessaire :

- PC (ad-SIEMENS de la salle Usine du futur) avec TIA Portal V14 (Step7)
- le partenaire UDP distant (dans notre cas un robot de marque Dobot)
- un réseau Ethernet commun entre les deux systèmes (WIFI par exemple)
- API Siemens (S7-1200 ou S7-1500)

Travail préparatoire : mise en place du réseau Ethernet

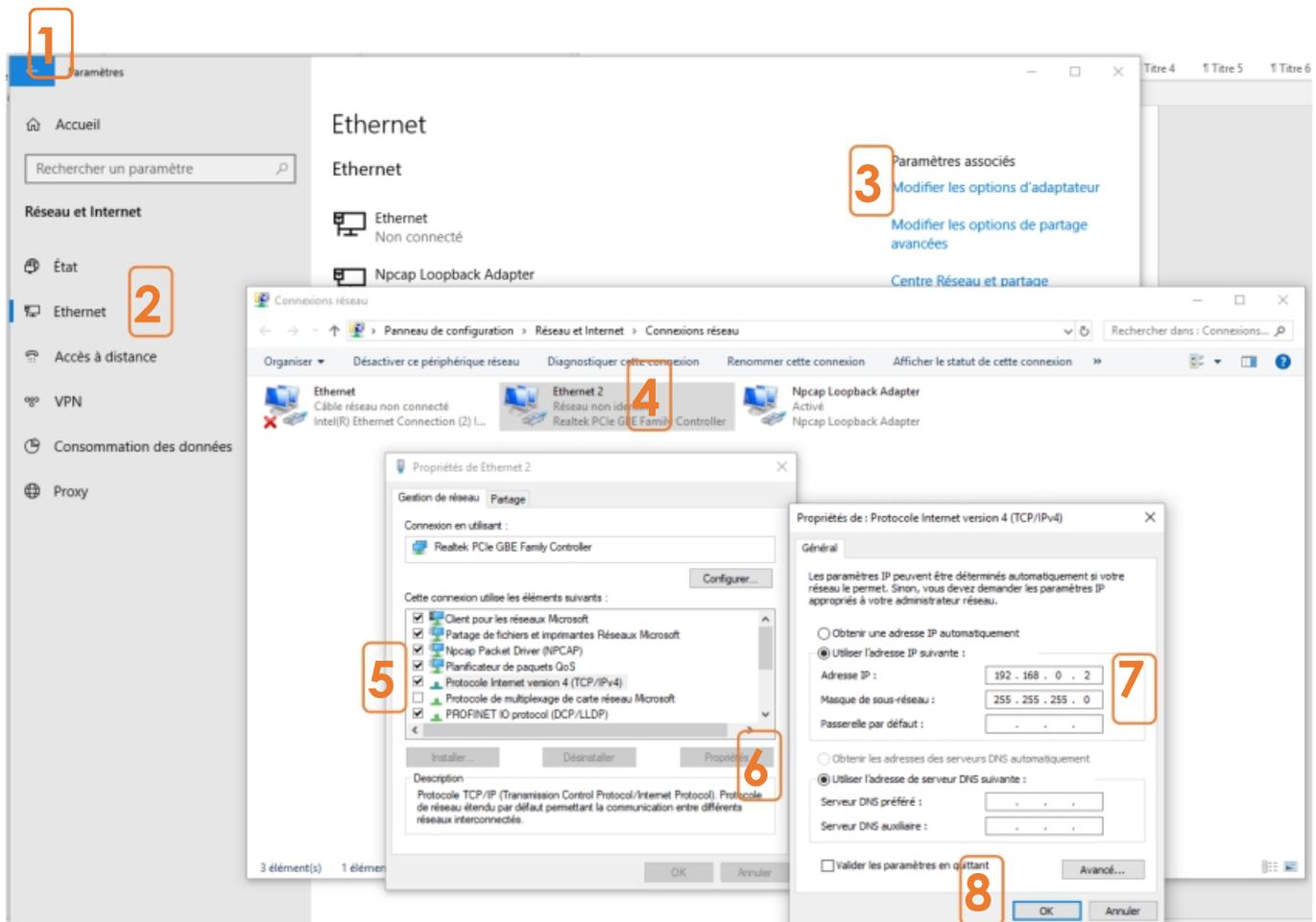
Il faut que l'API et le partenaire UDP soient présents physiquement sur le même réseau Ethernet pour pouvoir établir une communication UDP.

Dans notre exemple, l'API possède l'adresse IP 192.168.04 et le partenaire UDP à l'adresse IP 192.168.0.3 (voir Tuto « Connexion WIFI »).

Pour communiquer avec l'API depuis le logiciel d'ingénierie TIA Portal, veuillez vérifier que votre PC est relié à l'API (soit directement avec câbles croisés soit via un switch commun) et que l'adresse du port IP du PC soit configurée dans le même sous réseau.

Pour modifier l'adresse IP de la carte Ethernet du PC :

Propriétés > Réseau et Internet (1) > Ethernet (2) > Modifiez les options d'adaptateur (3) > Clic droit sur la carte de communication utilisée (4) > Propriétés > Protocole Internet version 4 (TCP/IPv4) (5) > Propriétés (6) > Utiliser l'adresse IP suivante (7) > RENSEIGNER AVEC UNE ADRESSE LIBRE DU MEME RESEAU (dans notre cas : 192.168.0.2) > OK (8)



Création du programme UDP

La version de TIA utilisée pour la création de ce tuto est V14.

En cas d'utilisation d'une version plus récente, des modifications peuvent être nécessaires.

Lorsque le PC, l'API Siemens et le partenaire UDP sont sur le même réseau Ethernet, lancez le logiciel TIA Portal :



Puis faites : Projet > Nouveau (indiquer éventuellement le chemin de sauvegarde du projet).

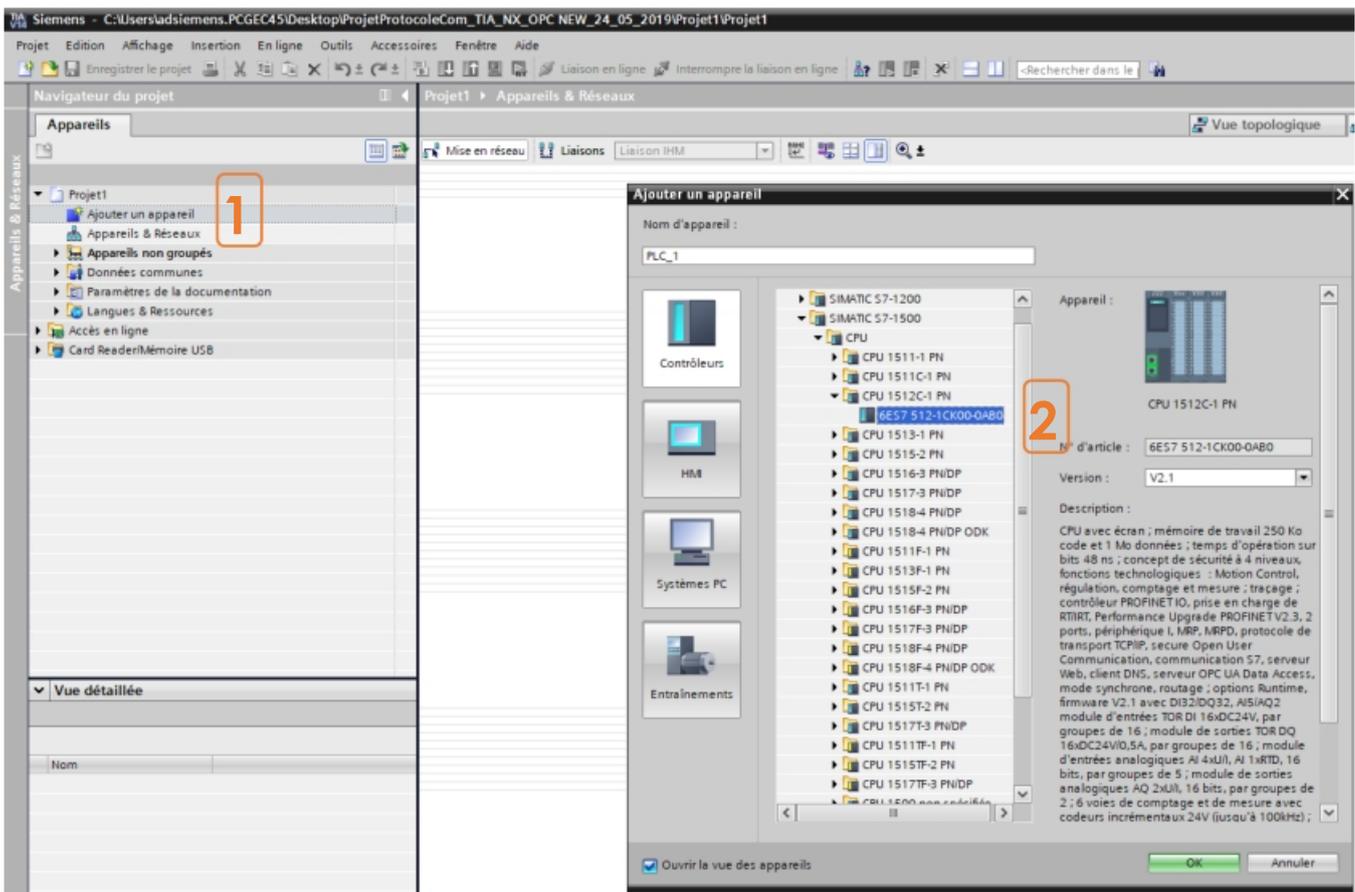
Maintenant que le projet TIA Portal est créé, on peut commencer à créer la configuration hardware du système, puis la programmation de l'API à proprement parlé.

➤ Etape 1 : Création de la configuration Hardware du projet

La première étape consiste à renseigner dans le projet tous les systèmes hardwares physiquement présents.

Pour faire cela : Cliquez sur "Ajouter un appareil" (1) puis cherchez la référence (ou la description) de l'API utilisée (2).

Dans notre cas, il s'agit d'un S7-1512 (réf 6ES7512-1CK00-0AB0)

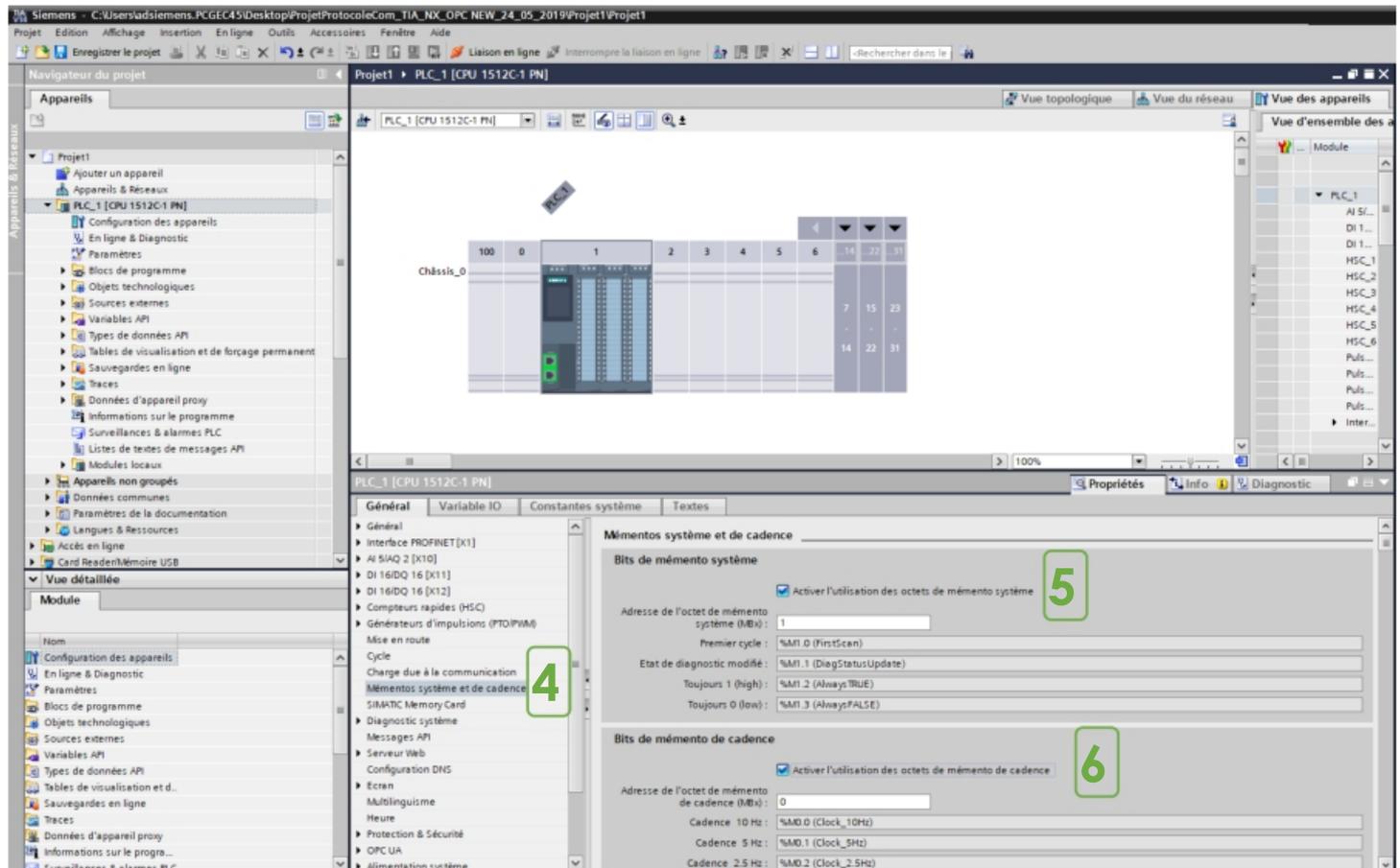
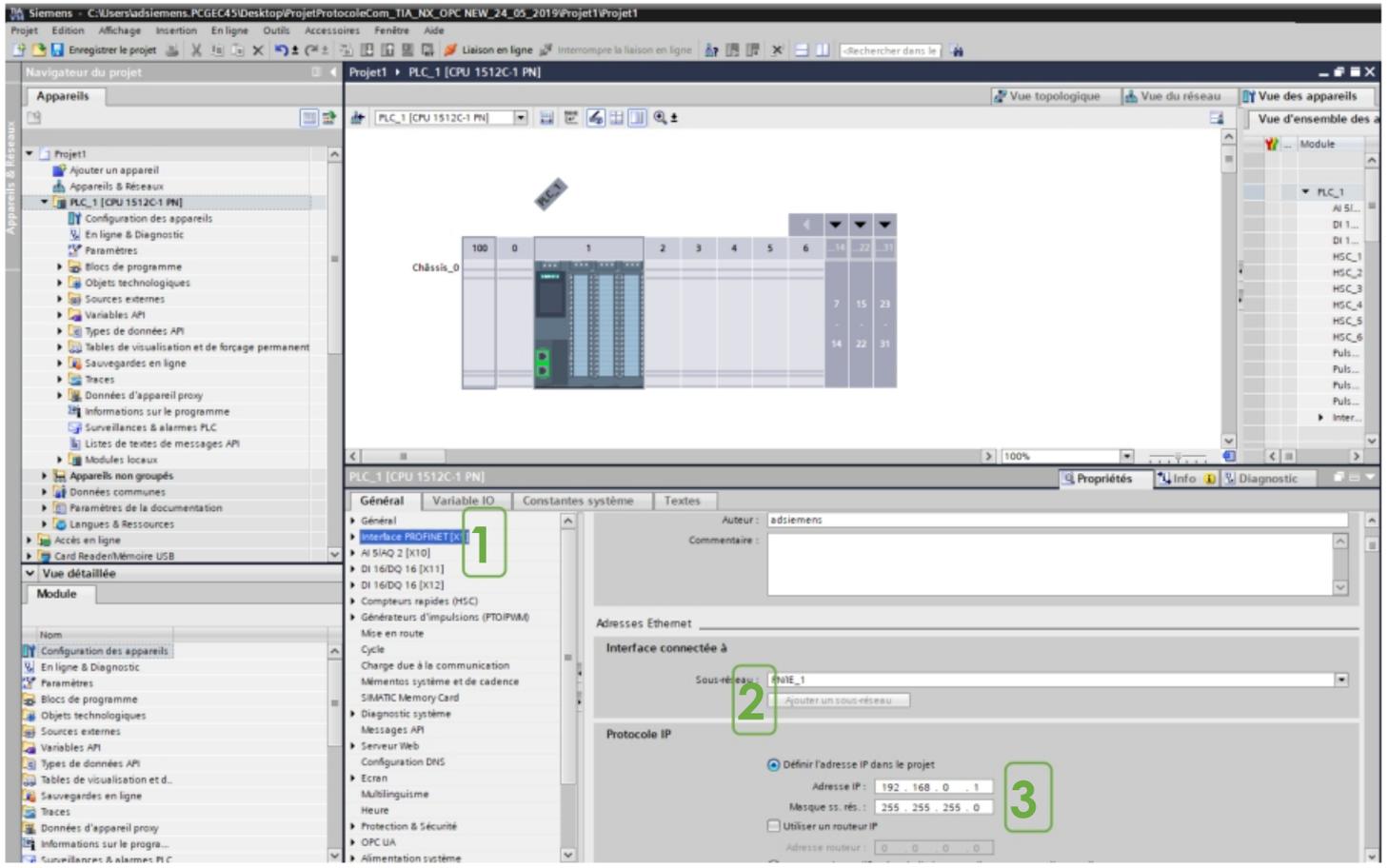


Ensuite il faut modifier les propriétés de l'API nouvellement ajouté dans la configuration HW :

Sélectionnez l'API dans la vue des appareils puis sous la fenêtre « Propriétés » :

- Cliquez sur « Interface PROFINET (X1) » (1) puis sur « Ajouter un sous-réseau » (2) puis renseignez l'adresse IP (3).
- Cliquez sur « Mémentos système et de cadence » (4) puis « Valider l'utilisation des octets de memento système » (5) et « Valider l'utilisation des octets de memento de cadence » (6).

Attention : Veuillez renseigner une adresse IP unique et dans le même sous réseau que le PC et le partenaire distant



Concernant la prise en compte du partenaire distant dans la configuration Hardware du projet, il y a 3 cas possibles :

- Le partenaire distant est un matériel Siemens. Dans ce cas, pour l'ajouter dans la configuration hardware du projet il suffit de le chercher dans le catalogue matériel et de l'ajouter.
- Le partenaire distant n'est pas un matériel Siemens mais il possède un fichier de description matériel (GSD). Il faut au préalable ajouter le fichier GSD au projet avant de pouvoir le chercher dans le catalogue matériel
- Le partenaire distant n'est pas un matériel Siemens et ne possède pas de fichier de description GSD. Il est impossible de l'ajouter dans la configuration hardware du projet. La prise en compte matériel du partenaire distant se fera par programmation de l'API.

Dans notre cas, avec le robot Dobot, nous nous trouvons dans le 3^{ème} cas : ce n'est pas un matériel Siemens et nous n'avons pas son fichier de description GSD.

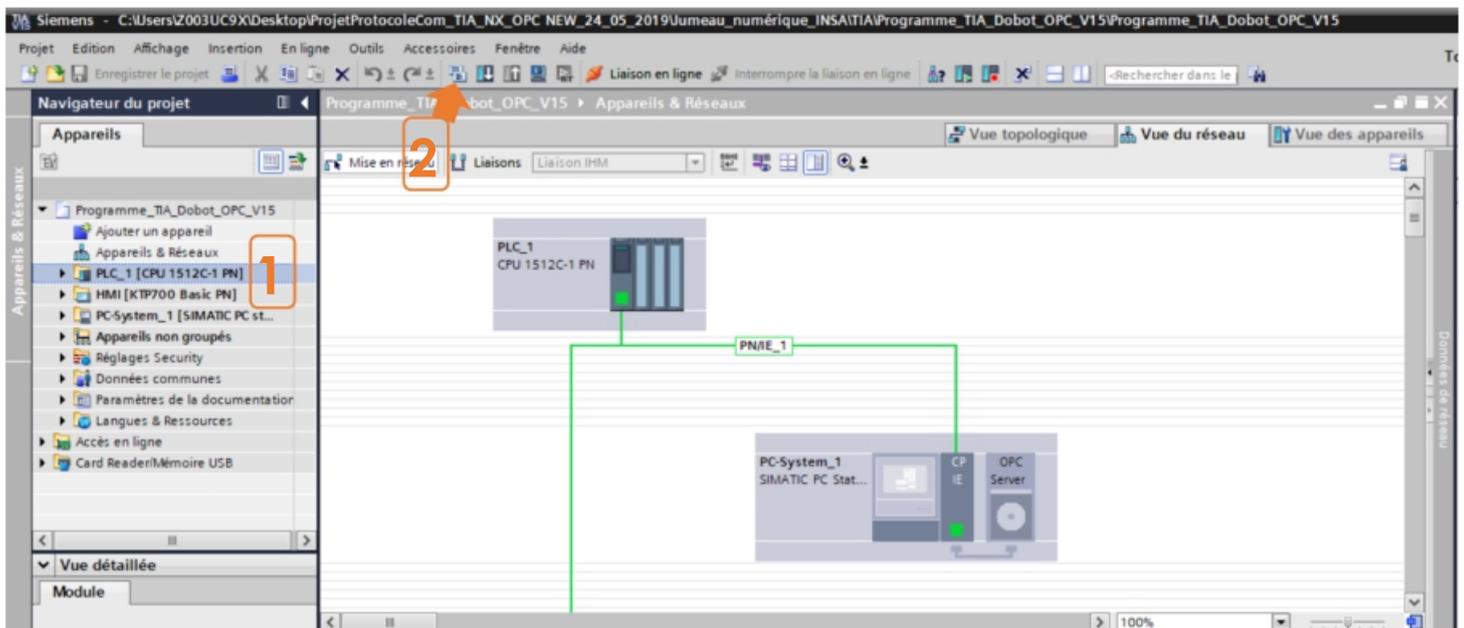
Attention : Ce tuto concerne la programmation d'une communication UDP entre un API Siemens et un partenaire non pris en compte dans la configuration matériel (car ne possédant pas de fichiers GSD - General Station Description file)

➤ Etape 2 : Chargement dans l'API de la configuration matériel

Lorsque la configuration du hardware du projet est terminée, il faut la charger dans l'API.

Au préalable, vous devez compiler le projet : cliquez sur le nom de l'API créé dans le navigateur de projet (1) puis sur « Compiler » (2).

Vérifiez qu'il n'y pas d'erreur à l'issue de la compilation dans la fenêtre « Info/compiler »



Quand la compilation est terminée, il faut charger dans l'API la configuration.

Toujours en ayant la sélection sur le nom de l'API dans le navigateur du projet, cliquez sur « Charger la station » (1). Dans la fenêtre qui apparaît configurez la liaison (2) :

- Type d'interface : renseignez le nom du réseau IE (dans notre cas c'est PN/IE)
- Interface PG/PC : renseignez le nom de la carte réseau utilisée (celle où se trouve le port Ethernet)
- Liaison avec interface/sous réseau et 1^{ère} passerelle : renseignez les informations de l'API (en général X130, Port1)

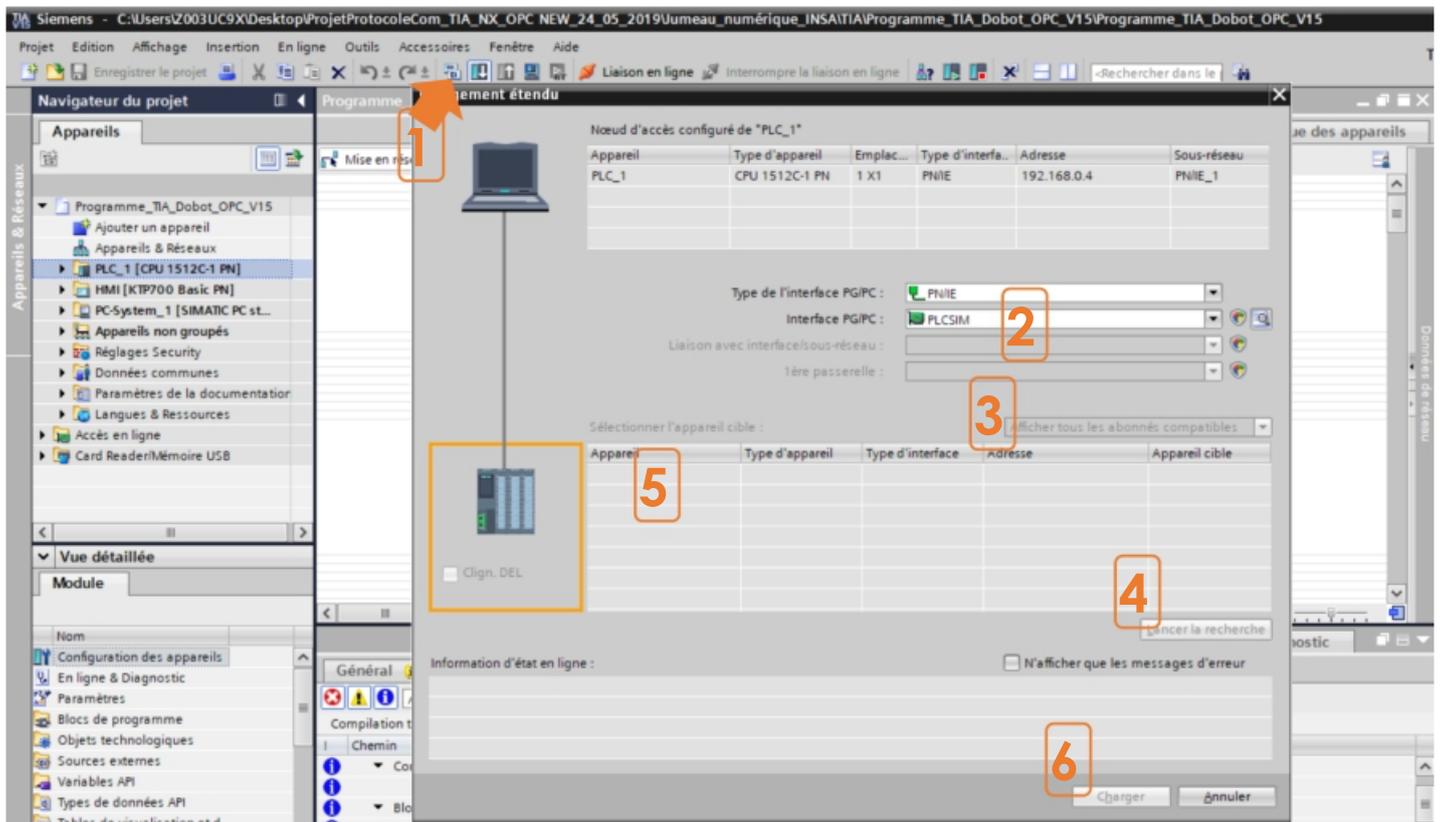
Sélectionnez « Afficher tous les abonnés accessibles » (3), puis cliquez sur « Lancer la recherche » (4).

TIA cherche à communiquer avec l'API physique, suivant les paramètres précédemment renseignés.

Si TIA trouve l'API, il l'affiche dans « Sélectionner l'appareil cible » (5).

Cliquez sur l'API trouvé (5), puis sur « Charger » (6).

A l'issue du chargement, vérifiez dans la fenêtre d'info l'absence d'erreur.



➤ Etape 3 : Création du programme utilisateur

Maintenant que la configuration hardware est créée et chargée dans l'API, on peut commencer à écrire le programme utilisateur, qui établira la communication UDP entre l'API et le partenaire.

Pour la suite du tuto, celui-ci sera moins détaillé qu'il ne l'a été jusqu'à présent car la création d'un programme utilisateur d'un API Siemens avec TIA Portal nécessite un nombre très important de manipulations et d'étapes, qui ne peuvent pas être détaillés dans un seul tuto.

En cas de difficulté, vous pouvez vous référer au « SIMATIC STEP 7 Basic/Professional V15.1 and SIMATIC WinCC V15.1 » à cette adresse internet :

<https://support.industry.siemens.com/cs/document/109755202/simatic-step-7-basic-professional-v15-1-and-simatic-wincc-v15-1?dti=0&lc=en-WW>.

Structure du programme TIA

La structure de base du programme se compose ainsi : nous y retrouvons l'OB1 (OB principal et cyclique) (1) qui appelle 3 fonctions différentes.

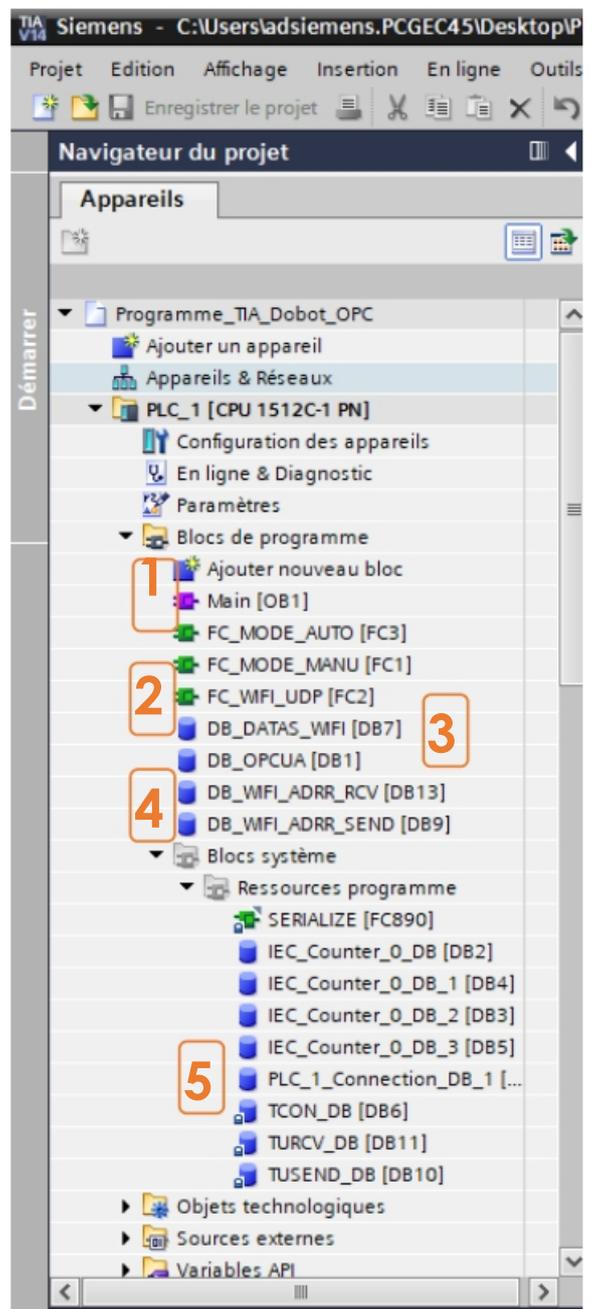
Seul la fonction « FC_WIFI_UDP » (2) nous intéresse puisque c'est cette fonction qui contient le code nécessaire aux échanges UDP.

En plus de la fonction FC_WIFI_UDP, nous avons utilisé une zone mémoire : le DB « DB_DATS_WIFI » (3). C'est dans ce DB que sont stockées toutes les variables et les constantes nécessaires.

En plus de ce DB, 3 autres DB (4) (5) sont nécessaires :

- DB_WIFI_ADRR_RCV : DB de paramètres contenant l'adresse et le port du partenaire UDP pour la réception des données
- DB_WIFI_ADRR_SEND : DB de paramètres contenant l'adresse du partenaire et le port du partenaire UDP pour l'envoi des données
- PLC_1_Connection_DB_1 : DB de paramètre pour la mise en place de la connexion UDP entre l'API et le partenaire

Attention : Tous les programmes utilisateurs sont différents et leurs bons fonctionnements dépendent de nombreux paramètres. Dans ce tuto, le programme présenté fonctionne correctement pour notre configuration, mais cela ne veut pas dire qu'il fonctionnera forcément pour des configurations différentes



FC « FC_WIFI_UDP »

Créez un FC dans le programme utilisateur. Ce FC contiendra le code nécessaire à l'établissement de la liaison UDP.

Lorsque le FC est créé, par glissé/déposé (depuis la fenêtre navigateur du projet), déposez le dans un réseau de l'OB1 afin qu'il soit appelé à chaque cycle API.

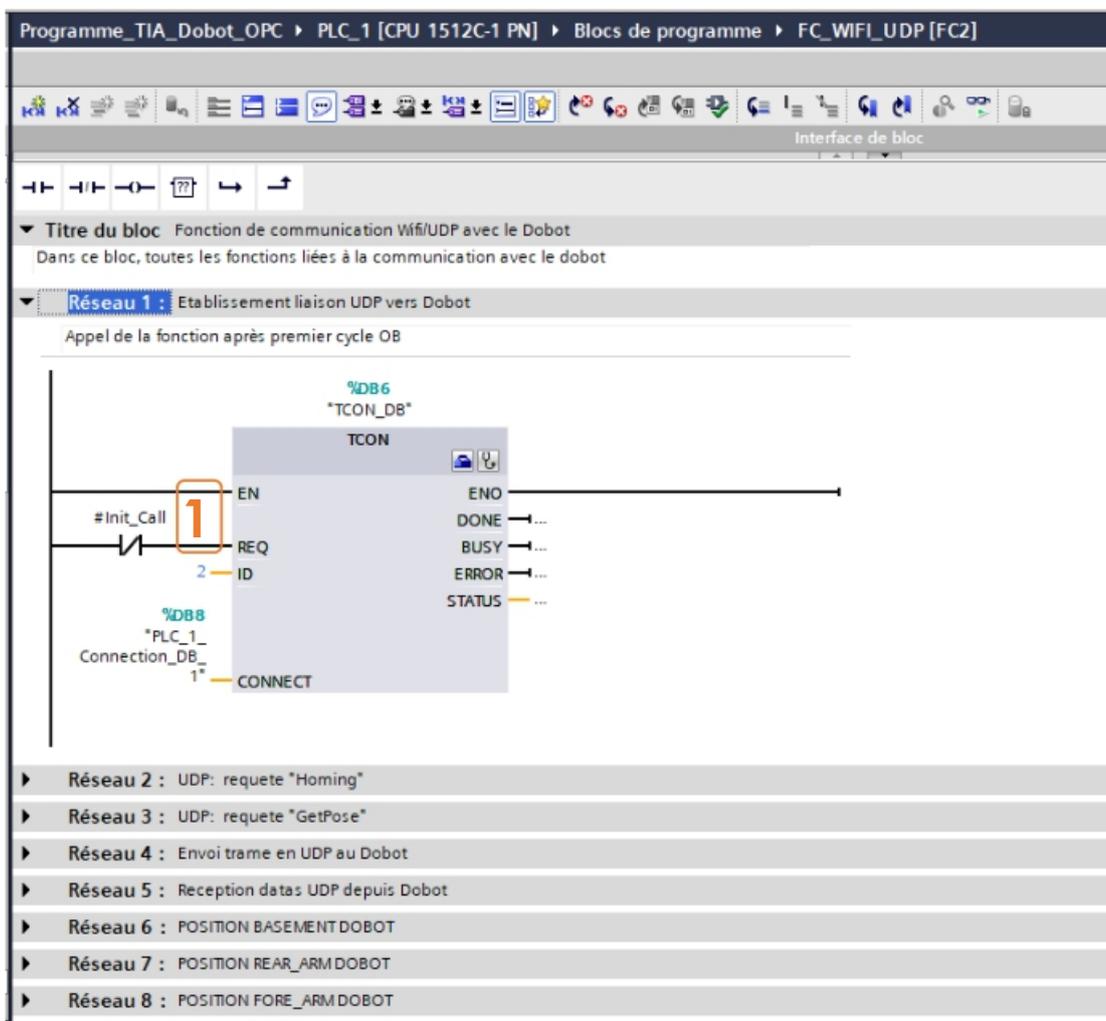
Instruction TCON

Allez dans le FC de communication, puis cherchez la première instruction importante pour la communication UDP : « TCON ».

Cette instruction permet de créer la liaison de communication UDP entre l'API et le partenaire. Elle est nécessaire puisque nous n'avons pas le partenaire UDP dans la configuration hardware. Ainsi cette instruction, couplée à des paramètres, permet de se passer de la configuration hardware du partenaire.

L'instruction TCON se trouve sous : *Instructions/Communication/Communication Open User/Autres*

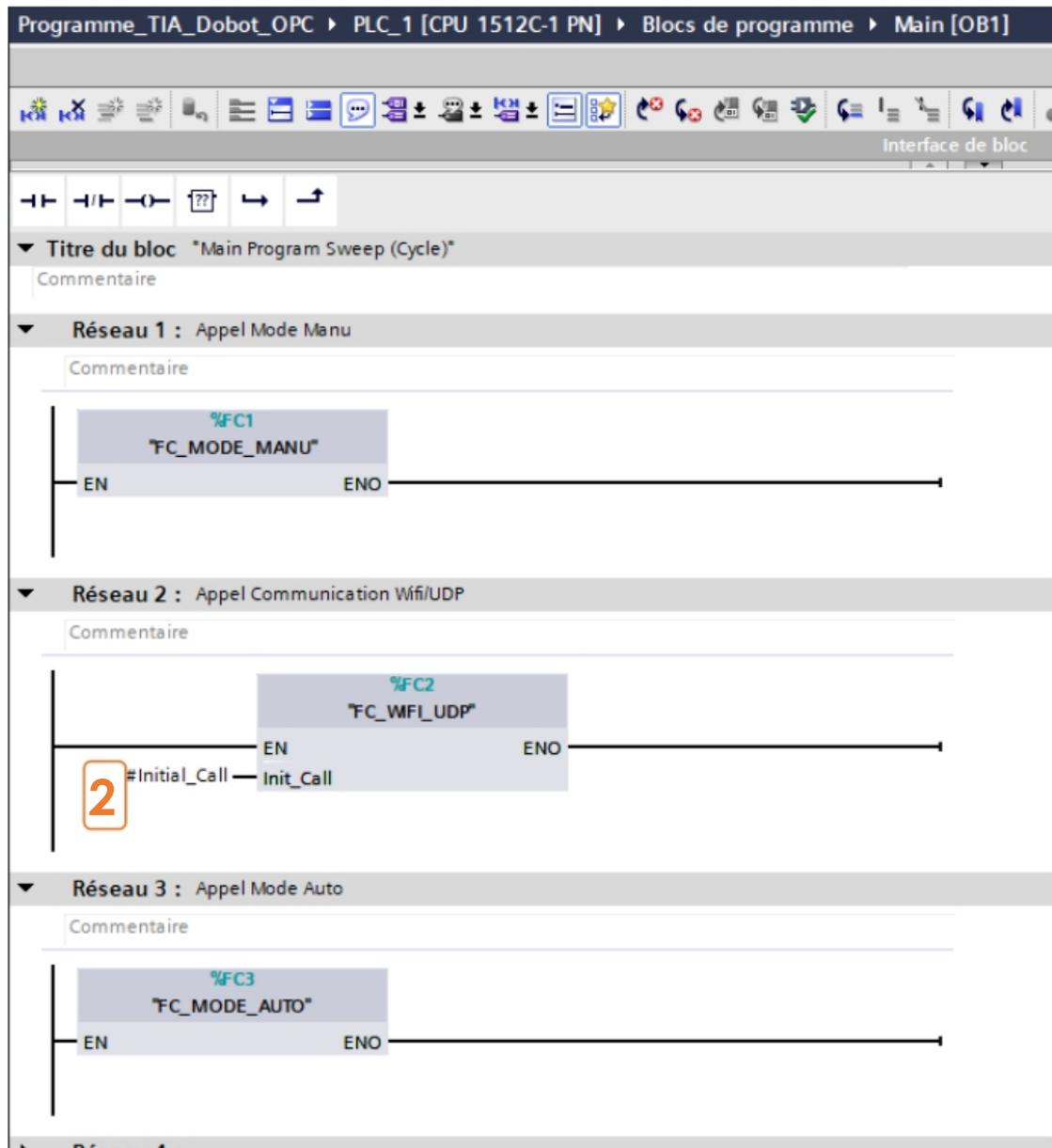
Déposez l'instruction TCON dans un réseau du FC par glissé/déposé. Validez la demande de création du DB par TIA.



Dans les données du FC, créez un nouvel Input « Init_Call ». Ce booléen passe à 1 après un premier un premier cycle API. Il est utilisé pour démarrer l'instruction TCON sur front montant après un premier cycle API.

Connecter Init_Call à l'entrée REQ de TCON (1) via un contact à fermeture.

Puis allez dans l'OB1 et connectez la nouvelle entrée du FC « Init_Call » à l'Input « Initial_Call » présente nativement dans les paramètres de l'OB1 (2)

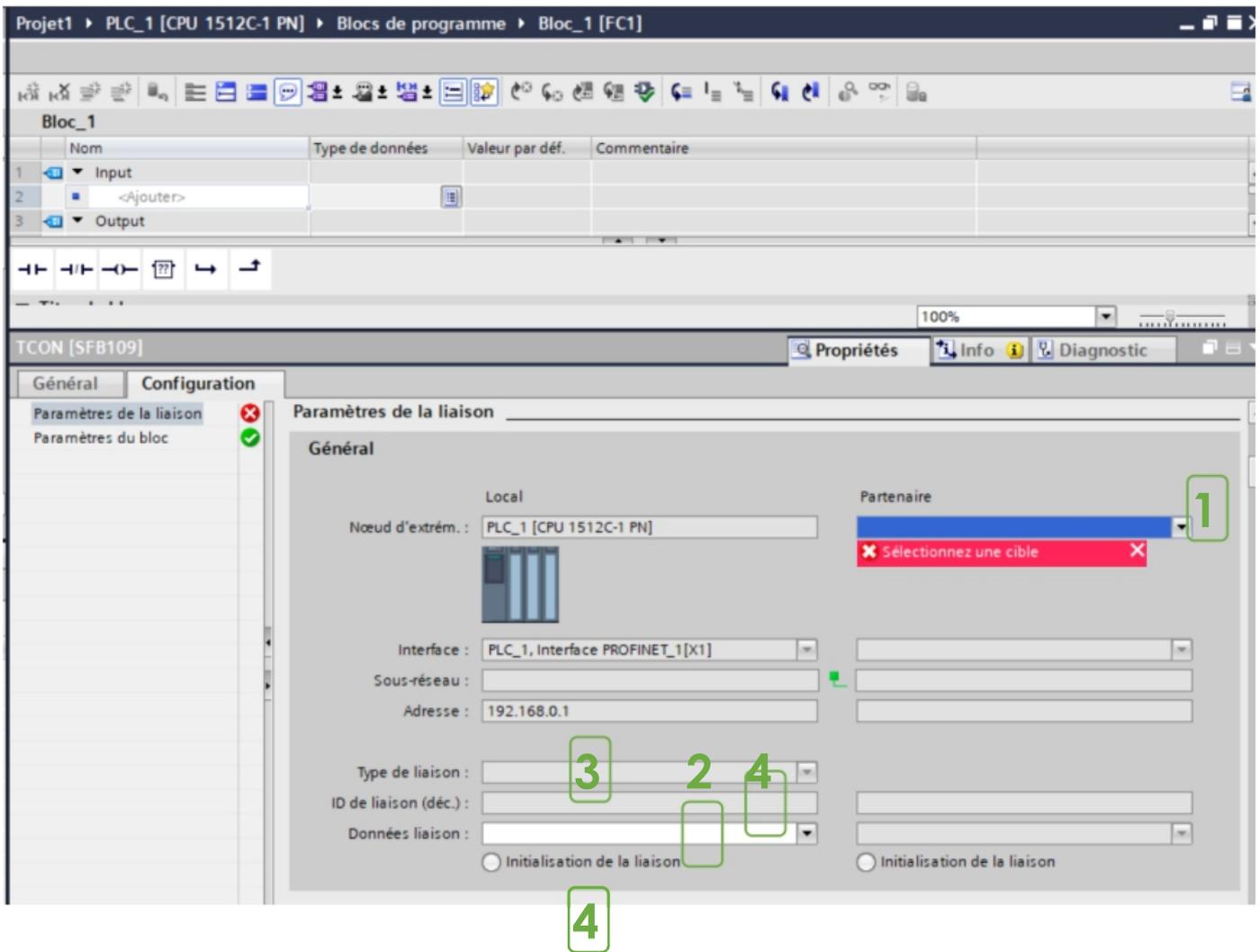


Retournez dans le FC content l'instruction TCON, puis cliquer sur le bloc TCON et allez dans « Propriétés », « Configuration » puis dans « Paramètres du bloc ».

Dans cette fenêtre, sous partenaire, sélectionnez « Non spécifié (1) ». Puis cliquez sur « Nouvelle liaison » dans « Données liaison » (2) et choisir « UDP » dans « Type de liaison » (3).

Donnez un ID de liaison (arbitraire) (4), dans notre cas il s'agit de l'ID 2.

Enfin, renseignez le numéro du port local (5). Dans notre cas, pour communiquer avec un Dobot il est indispensable de renseigner le numéro de port : 12345.



Allez dans le DB créé pour la liaison UDP nouvellement créée (dans notre cas il s'agit du DB « PLC_1_Connexion_1 »), puis renseignez l'adresse IP du partenaire UDP (1).

Attention l'adresse est à renseigner en notation hexadécimale (4 bytes).

Programme_TIA_Dobot_OPC > PLC_1 [CPU 1512C-1 PN] > Blocs de programme > Blocs système > Ressources programme > PLC_1_Connexion_DB_1 [DB8]

Conserv. les valeurs actuelles Instantané Copier les instantanés dans les valeurs de départ

PLC_1_Connexion_DB_1

Nom	Type de données	Valeur de départ	Rémanence	Accessible ...	Écritu...	Visible de...	Valeur de ..	Surveillan...	Commentaire
Static									
Interfaceld	HW_ANY	64		✓	✓	✓			HW-identifier of IE-interface submodule
ID	CONN_OUC	2		✓	✓	✓			connection reference / identifier
ConnectionType	Byte	19		✓	✓	✓			type of connection: 11=TCP/IP, 19=UDP (17=TCP
ActiveEstablished	Bool	false		✓	✓	✓			active/passive connection establishment
RemoteAddress	IP_V4			✓	✓	✓			remote IP address (IPv4)
ADDR	Array[1..4] of Byte			✓	✓	✓			IPv4 address
ADDR[1]	Byte	16#C0		✓	✓	✓			IPv4 address dobot (hexa)
ADDR[2]	Byte	16#A8		✓	✓	✓			IPv4 address dobot (hexa)
ADDR[3]	Byte	16#00		✓	✓	✓			IPv4 address dobot (hexa)
ADDR[4]	Byte	16#03		✓	✓	✓			IPv4 address dobot (hexa) (03)
RemotePort	UInt	0		✓	✓	✓			remote UDP/TCP port number
LocalPort	UInt	12345		✓	✓	✓			local UDP/TCP port number

Attention 1 : Le numéro de port local peut être arbitraire. Néanmoins, lors de l'établissement d'une liaison UDP il se peut que le partenaire cherche à communiquer uniquement avec certains ports. Cela dépend de la façon dont est programmé le partenaire.

Attention 2 : L'adresse IP à renseigner pour le partenaire UDP dans les propriétés de l'instruction TCON doit être fait en notation hexadécimale.

Instruction TUSEND

L'instruction TUSEND est l'instruction utilisée pour envoyer des requêtes avec un encapsulage UDP au partenaire distant.

Elle se trouve sous : *Instructions/Communication/Communication Open User/Autres*

Par glissé/déposé, ajoutez l'instruction TUSEND dans le FC de communication.

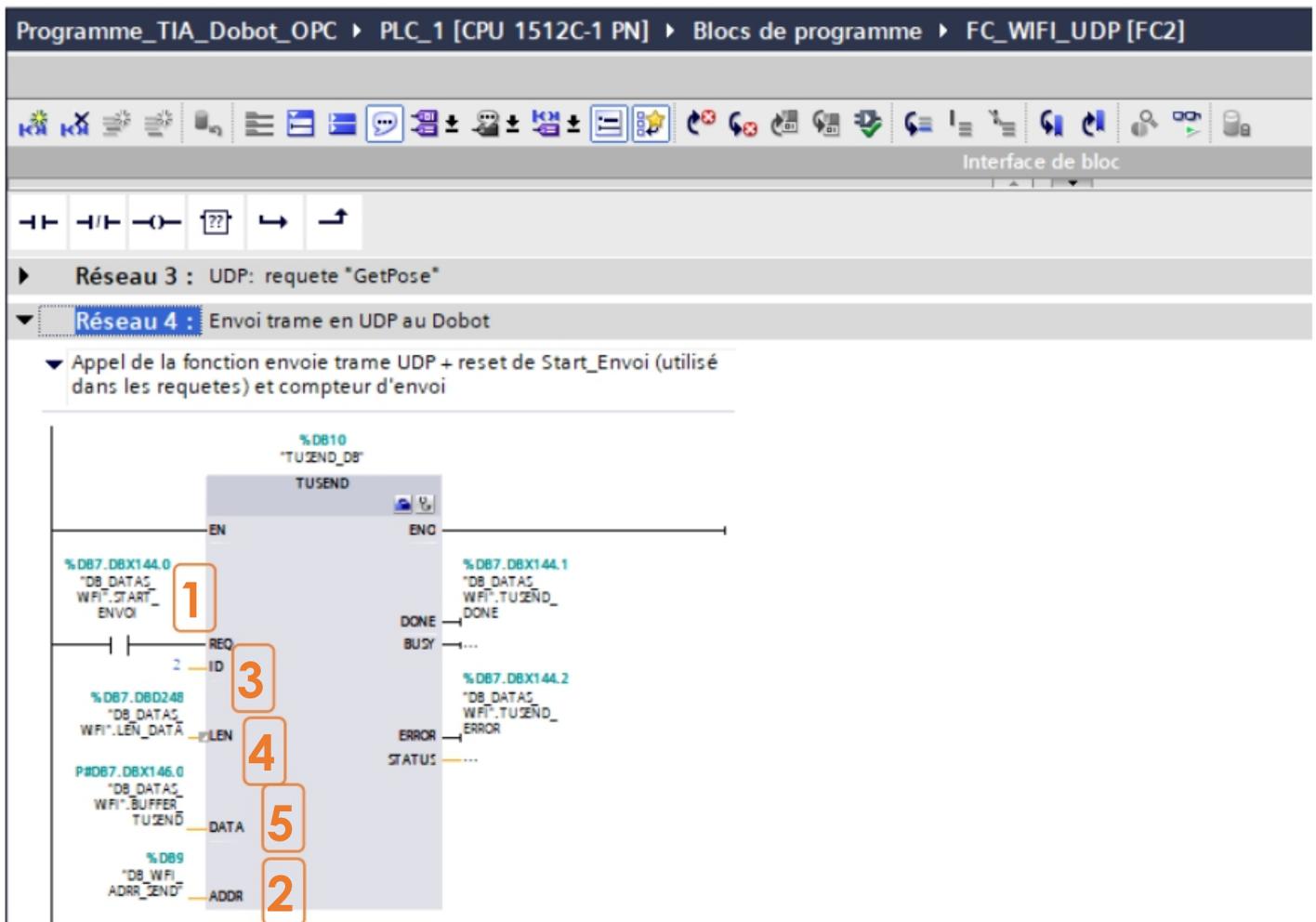
Ensuite depuis le navigateur du projet, créez un nouveau DB. Ce DB contiendra l'adresse IP et le port du partenaire. Il sera lié à l'instruction TUSEND. Dans notre cas, ce DB se nomme « DB_WIFI_ADRR_SEND ».

Comme pour TCON, l'adresse IP doit être renseignée en hexadécimale ainsi que le port du partenaire. Dans notre cas, le numéro du port distant est 8899 car nous communiquons avec un Dobot.

Nom	Type de données	Valeur de départ	Rémanence	Accessible	Écritu...	Visible da...	Valeur de ...	Surveillan...	Commentaire
Static									
REM_IP_ADDR	Array[1..4] of U...								remote station address
REM_IP_ADDR[1]	USint	16#C0							Adresse IP en hexa du dobot / 192.
REM_IP_ADDR[2]	USint	16#A8							Adresse IP en hexa du dobot / 168.
REM_IP_ADDR[3]	USint	16#00							Adresse IP en hexa du dobot / 0.
REM_IP_ADDR[4]	USint	16#03							Adresse IP en hexa du dobot / 3 (03)
REM_PORT_NR	USint	16#22C3							Numéro de port en hexa du dobot / 8899
RESERVED	Word	16#0							unused; has to be 0

Attention 1 : Le numéro du port distant est rarement arbitraire. En général, lors de l'établissement de la liaison UDP, le partenaire indique le numéro de port qu'il ouvre pour la communication UDP. Cette étape dépend de la programmation du partenaire.

Attention 2 : L'adresse IP à renseigner pour le partenaire UDP dans les propriétés de l'instruction TCON ainsi que le numéro de port distant doivent être fait en notation hexadécimale.



Retournez dans le FC de communication sur l'instruction TUSEND.

Sur l'entrée REQ, connectez un bit qui passe à 1 lorsqu'une requête UDP doit être envoyée et qui reprends la valeur 0 lorsque l'envoi de la requête est réussi (information disponible via la sortie « DONE ») (1).

Connectez à l'entrée ADDR le DB nouvellement créé qui contient l'adresse et le numéro du port distant (2). Dans notre cas il s'agit du DB « DB_WFI_ADDR_SEND ».

Sur l'entrée ID, renseignez le numéro identique au numéro de liaison crée avec l'instruction TCON (3). Dans notre cas, il s'agit du numéro 2.

Avec l'entrée LEN, connectez une variable qui contient la longueur (en octets) de la donnée UDP à envoyer (4).

La donnée UDP à envoyer doit être liée à l'entrée DATA. Idéalement, utilisez un tableau de byte (Array of bytes) pour contenir les données à envoyer (5).

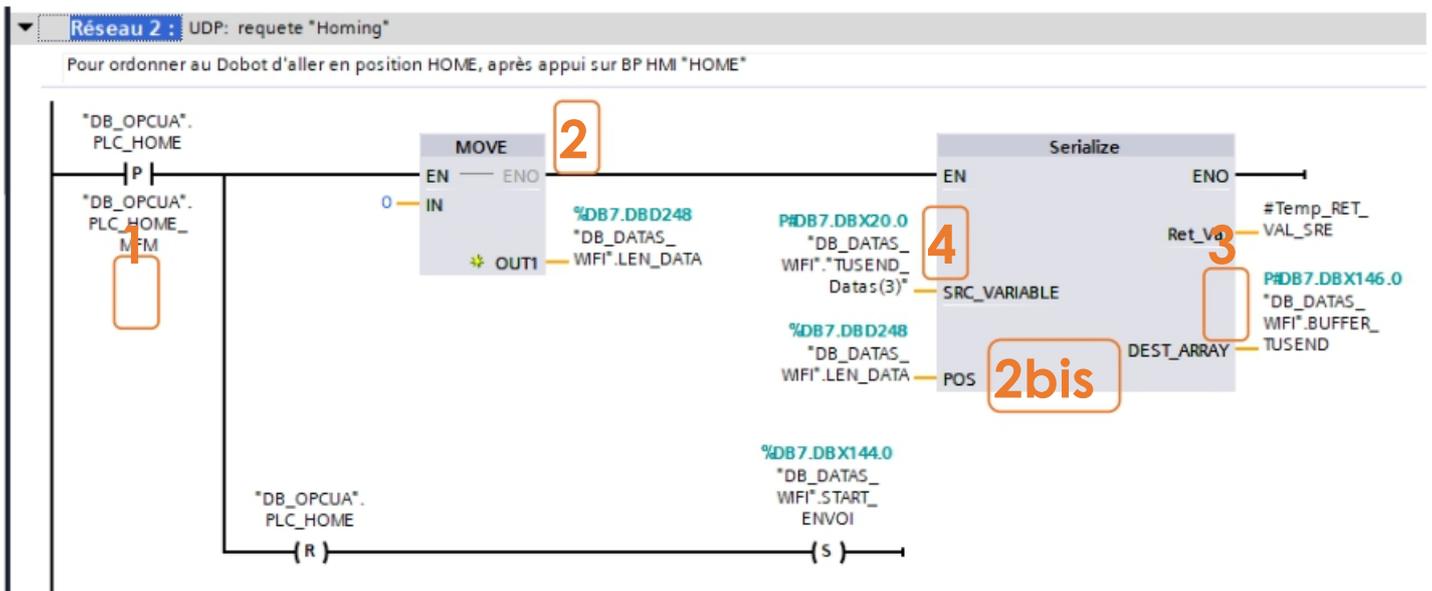
Attention : Il est important de ne valider qu'une seule fois, via l'utilisation d'un front montant d'un bit, la validation de l'envoi de la requête (sur l'entrée REQ). Si ce contrôle n'est pas fait, il y'a un risque que le réseau soit saturé par le nombre de requêtes envoyés consécutivement par l'API.

Création de la donnée à envoyer en UDP

Créez un nouveau réseau avec lequel on met en forme les données à envoyer avec l'instruction TUSEND.

Dans notre cas, nous envoyons au Dobot une requête « Homing » qui demande au Dobot de retourner à sa position initiale.

Après l'utilisation d'un front montant d'un bit de validation (1), chargez la valeur 0 dans une variable qui contiendra la longueur de la donnée (utiliser une instruction MOVE) (2). Ajoutez l'instruction SERIALIZE qui chargera dans le tableau d'octets (utilisé par le TUSEND) (3) la structure de la donnée à envoyer (4). Connectez la variable contenant la longueur de la donnée à envoyer (préalablement chargé à 0) à l'entrée POS de l'instruction SERIALIZE (2bis).



Créez une structure qui contient les données à envoyer. Cette structure peut être contenue dans un DB unique ou dans un DB global.

La structure de la donnée à envoyer est fonction du programme du partenaire distant. Il faut avoir accès au programme du partenaire pour savoir quelle est la structure attendue. Si on n'a pas accès au programme, cette information peut être contenue dans les documentations du partenaire.

Dans notre cas, la structure pour l'envoi de la requête « Homing » est constituée comme ci-dessous (issue de la doc Dobot) :

Table 33 The command packet of GetHOMEParams

Header	Len	Payload			Checksum	
		ID	Ctrl			Params
			rw	isQueued		
0xAA 0xAA	2+0	30	0	0	Empty	Payload checksum

Attention : Sans l'information de la structure attendue par le partenaire, il est impossible de créer une communication UDP.

Instruction TURCV

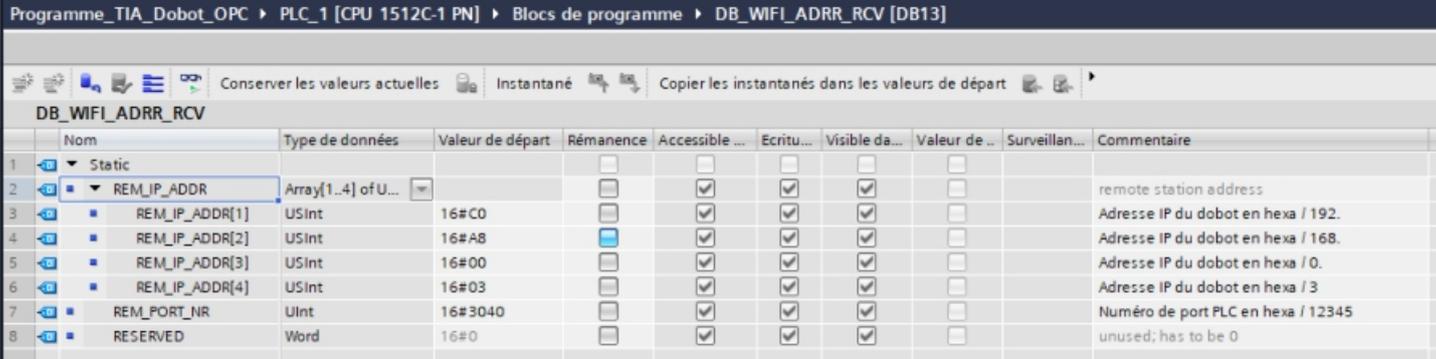
L'instruction TURCV est utilisée pour la réception des trames UDP envoyées par le partenaire à destination de l'API.

Elle se trouve sous : *Instructions/Communication/Communication Open User/Autres*

Par glissé/déposé, ajoutez l'instruction TURCV dans le FC de communication.

Ensuite depuis le navigateur du projet, créer un nouveau DB. Ce DB contiendra l'adresse IP du partenaire et le port ouvert de l'API. Il sera lié à l'instruction TURCV. Dans notre cas, ce DB se nomme « DB_WIFI_ADRR_RCV ».

L'adresse IP doit être renseignée en hexadécimal ainsi que le numéro du port de l'API. Dans notre cas, le numéro du port local est 12345, c'est le même port que celui renseigné dans TCON.



Nom	Type de données	Valeur de départ	Rémanence	Accessible ...	Écritu...	Visible da...	Valeur de ..	Surveillan...	Commentaire
1	Static		<input type="checkbox"/>						
2	REM_IP_ADDR	Array[1..4] of U...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			remote station address
3	REM_IP_ADDR[1]	USInt	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Adresse IP du dobot en hexa / 192.
4	REM_IP_ADDR[2]	USInt	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				Adresse IP du dobot en hexa / 168.
5	REM_IP_ADDR[3]	USInt	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Adresse IP du dobot en hexa / 0.
6	REM_IP_ADDR[4]	USInt	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Adresse IP du dobot en hexa / 3
7	REM_PORT_NR	UInt	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Numéro de port PLC en hexa / 12345
8	RESERVED	Word	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			unused; has to be 0

Attention 1 : Le numéro du port local doit impérativement être identique au numéro de port déjà renseigné dans TCON

Attention 2 : L'adresse IP à renseigner pour le partenaire UDP ainsi que le numéro de port local doivent être fait en notation hexadécimal.

Retournez dans le FC de communication sur l'instruction TURCV nouvellement ajoutée.

Comme pour TCON, connectez le bit « Init_Call » à l'entrée EN_R afin de valider l'instruction de réception des trames UDP après une premier cycle API (1).

Ensuite renseignez TURCV comme suit :

- Sur l'entrée ID, mettre le même numéro d'ID de liaison identique à celui utilisé par TCON et TUSEND. Dans notre cas il s'agit de l'ID 2 (2).
- Sur l'entrée LEN : renseignez avec la valeur 0. Le paramètre 0 indique que TURCV adapte automatiquement la longueur de la zone de réception des données automatiquement en fonction des trames reçues (3).
- Sur l'entrée ADRR, connectez le DB créé précédemment contenant l'adresse IP du partenaire et le port local (4). Dans notre cas il s'agit du DB « DB_WIFI_ADRR_RCV ».
- Sur l'entrée DATA, connectez un tableau d'octets (Array of bytes) qui servira de zone mémoire pour la réception des données UDP (5).

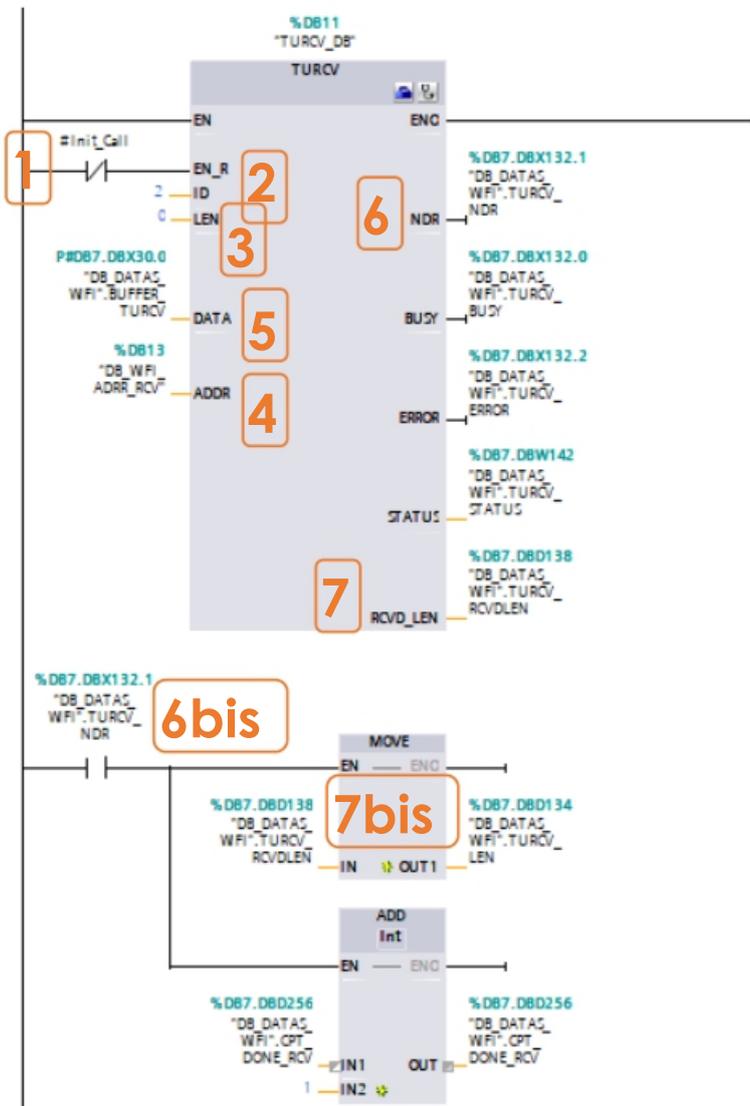
Pour préparer la mise en forme des données reçues, créez un bit qui passe à 1 lorsque la fonction de réception est terminée (6). Puis connectez à la sortie RCVD_LEN une variable qui contiendra la longueur de la trame UDP reçue (7). Lorsque le bit de confirmation de trame reçue passe à 1 (6bis), chargez dans une nouvelle variable la longueur de la trame effectivement reçue (7bis).

Programme_TIA_Dobot OPC ▶ PLC_1 [CPU 1512C-1 PN] ▶ Blocs de programme ▶ FC_WIFI_UDP [FC2]



- ▶ Réseau 3 : UDP: requete "GetPose"
- ▶ Réseau 4 : Envoi trame en UDP au Dobot
- ▼ Réseau 5 : Reception datas UDP depuis Dobot

▶ Appel de la fonction après premier cycle OB et reception en dur de l...



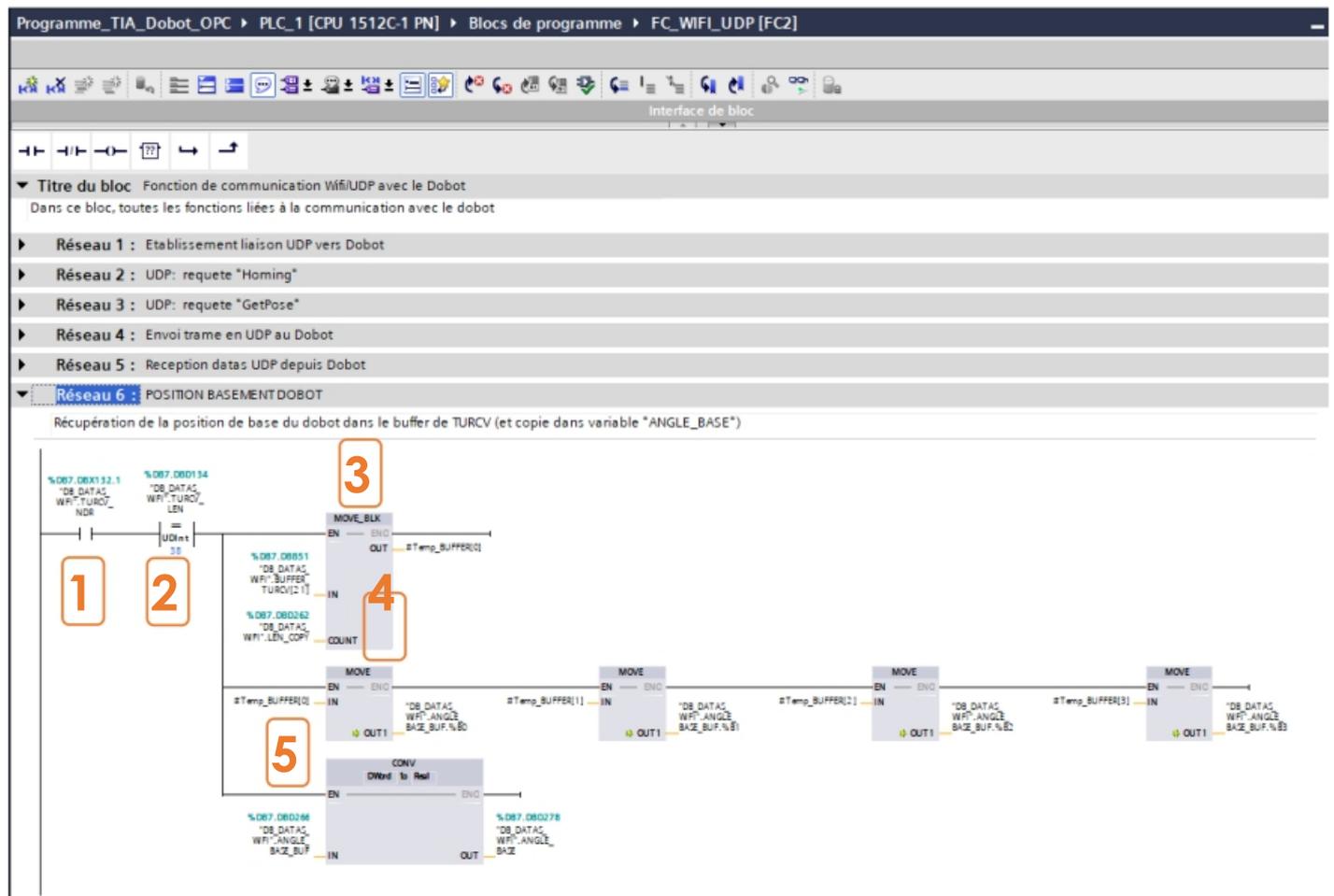
Mise en forme de la frame UDP reçue

Pour mettre en forme les données reçues du partenaire distant, il existe deux méthodes :

- Reconnaissance de l'ID de donnée : un ID unique est envoyé dans la structure des données envoyer et permet ainsi la reconnaissance de la frame reçue.
- Reconnaissance de la longueur de donnée : la longueur des frames reçues est connue à l'avance et permet de reconnaître quelle frame est reçue.

Dans notre cas, nous avons choisis de reconnaître les frames reçues par la distinction de la longueur des données reçues.

Attention : Utiliser la méthode de la distinction de la longueur des données reçues ne garantit pas à 100% la bonne reconnaissance des frames reçues. Seule la distinction par la reconnaissance de l'ID de données permet une identification à 100%



Dans le FC de communication, créez un nouveau réseau.

Dans ce réseau, après validation de la bonne réception d'une trame par l'instruction TURCV (1), ajoutez un comparateur sur la longueur de la trame reçue (2) afin de déterminer le type de trame reçue. Dans notre cas, nous faisons la comparaison avec le chiffre 38, puisque nous attendons la trame de positions du Dobot, qui fait une longueur de 38 octets.

Créez dans les paramètres du FC un tableau d'octets temporaires (Array of bytes) qui servira de zone mémoire tampon pour la mise en forme de la trame reçue. Via la fonction MOVE_BLK, charger dans la mémoire tampon la zone de réception utilisée par l'instruction TURCV (3). La longueur des données qui doivent être copiées est stockée dans une constante. Dans notre cas, il s'agit de la constante « LEN_COPY » qui contient la valeur 4 (4).

Créez une structure conforme à la donnée attendue. Enfin, chargez dans la structure, adaptée aux données attendues, les bytes contenues dans la zone mémoire tampon (5).

Une donnée UDP n'a pas de format pré-établi et chaque matériel communiquant organise les données envoyées arbitrairement. Cela dépend du programme du partenaire distant. Cette information peut être trouvée soit dans le programme du partenaire distant soit dans sa documentation.

Dans notre cas, la structure est adaptée à la réception de la trame qui contient les positions du Dobot.

La structure pour la réception de la requête « GetPose » (Positions du Dobot) est constituée comme ci-dessous (issue de la doc Dobot) :

Table 22 The returned command packet of GetPose

Header	Len	Payload				Checksum
		ID	Ctrl		Params	
			rw	isQueued		
0xAA 0xAA	2+32	10	0	0	Pose (See Program 1)	Payload checksum

Program 1 Pose definition

```
typedef struct tagPose {
float x;           //Robotic arm coordinate system x
float y;           //Robotic arm .
float z;           //Robotic arm coordinate system z
float r;           //Robotic arm coordinate system r
float jointAngle[4]; //Robotic arm 4 axis(The basement, rear arm, forearm,EndEffector) angles
} Pose;
```

Attention : Sans l'information de la structure des données reçues, il est impossible d'établir une communication UDP

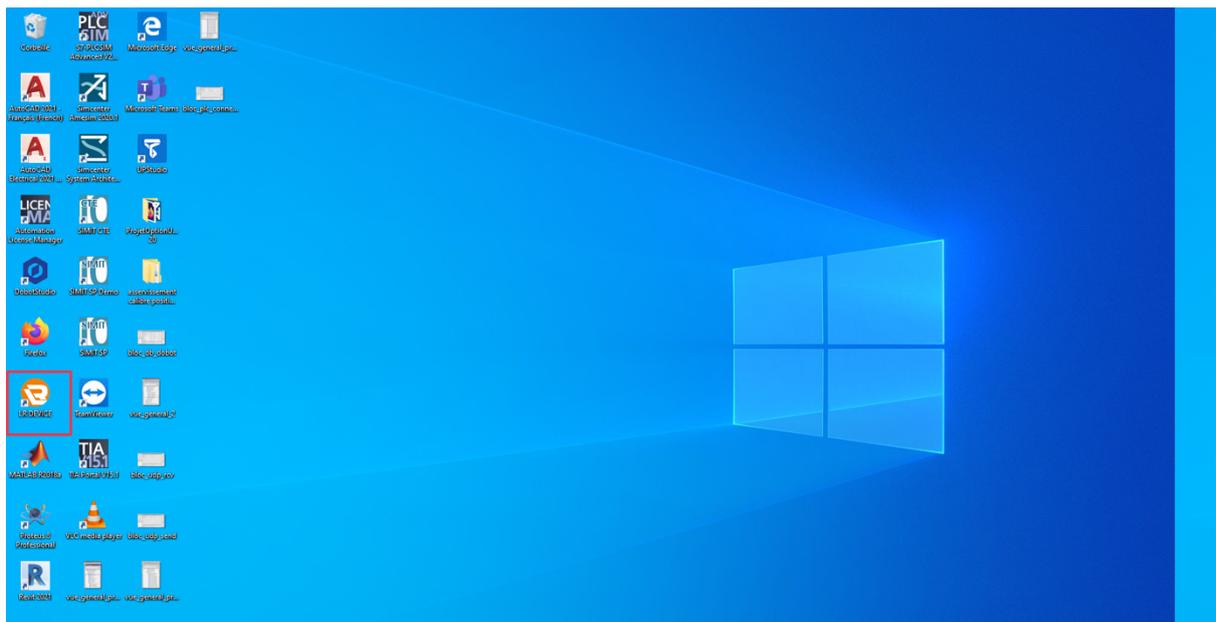
ANEXO B – Tutorial LR Device

Tuto – LR Device et AL-1302

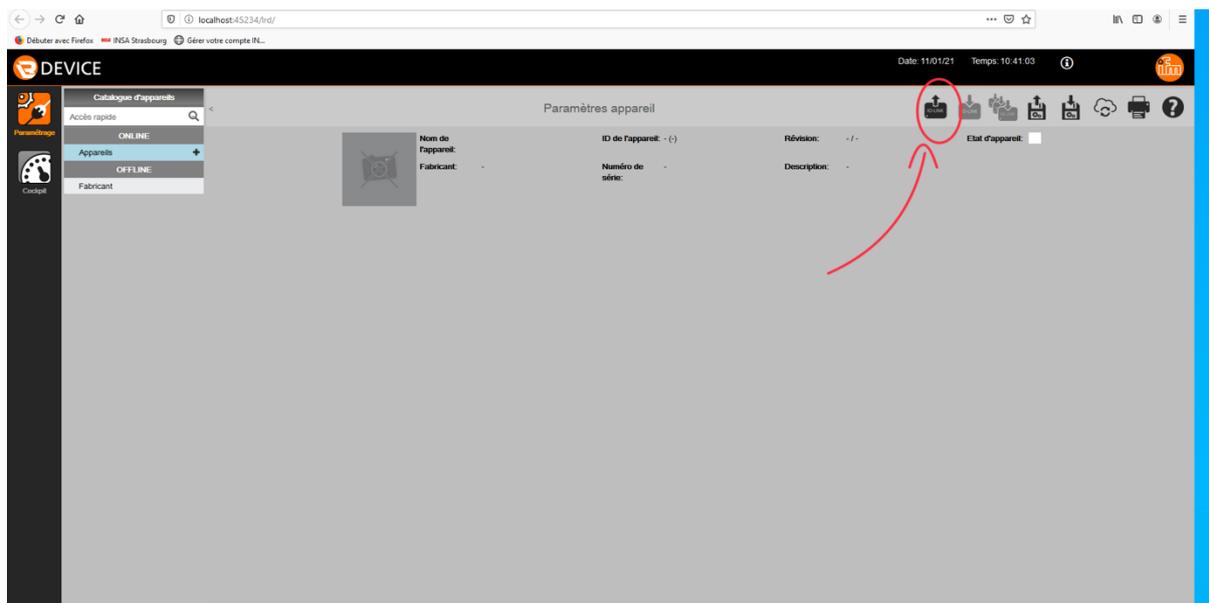
Pour commencer, on doit câbler le dispositif AL-1302 de la manière suivante :



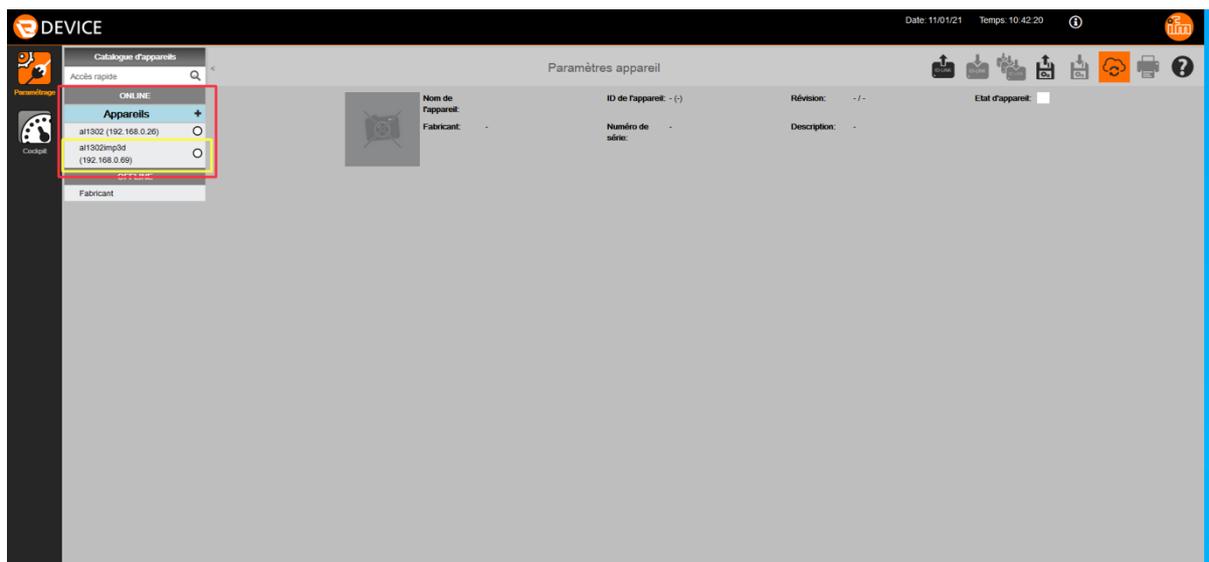
Le câble vert est là réseau *Profinet* et les autres câbles sont les capteurs IFM qui seront utilisables dans le projet.



On doit ouvrir l'application **LR device**.



Cette page doit s'afficher. Pour commencer la détection d'un dispositif, on sur **Lire de l'appareil**.



Après quelques instants, si l'appareil est bien branché, une liste de dispositifs disponibles apparaîtra à gauche. Alors, il faut cliquer sur l'option correspondant. Dans notre cas, c'est le deuxième.

The screenshot displays the 'Paramètres appareil' (Device Parameters) configuration page. The interface includes a sidebar with a device catalog, a top navigation bar with the 'Ecrire' (Write) button highlighted in yellow, and a main configuration area. The configuration area is divided into two sections: 'Paramètre' (Parameter) and 'Valeur' (Value). The 'Paramètre' section lists various device settings, and the 'Valeur' section shows the current values for these settings. A red box highlights the 'Application Tag', 'IP address', and 'Subnet mask' fields. The 'Application Tag' is set to 'AL1302', the 'IP address' is '192.168.0.69', and the 'Subnet mask' is '255.255.255.0'. The 'Ecrire' button is located in the top right corner of the configuration area.

Paramètre	Valeur	Unité	Min	Max	Description
Port 1	Access rights	Fieldbus + IoT			Defines the access rights for the IO-Link Master
Port 2	DHCP	Static IP			IoT IP address setting with DHCP or static IP
Port 3	IP address				IoT IP address of IO-Link Master
Port 4	Subnet mask				IoT subnet mask of IO-Link Master
Port 5	Default gateway IP address				IoT default gateway IP address
Port 6	MAC address				IoT MAC address of IO-Link Master
Port 7	IP address LR Agent or SMARTOBSERVER				Target IP of LR Agent or SMARTOBSERVER for sending process data
Port 8	Port LR Agent or SMARTOBSERVER				Target port of LR Agent or SMARTOBSERVER for sending process data
Info	Interval LR Agent or SMARTOBSERVER				Type in the sending interval to LR Agent or SMARTOBSERVER for process data
Firmware	Application Tag				Name for IO-Link Master in LR SMARTOBSERVER structure
	IP address				Fieldbus IP address of IO-Link Master
	Subnet mask				Fieldbus subnet mask of IO-Link Master
	Default gateway IP address				Fieldbus default gateway IP address
					Fieldbus name. Allowed characters: 'a-z' (lower case letters), '0-9' (digits), '-' (minus), '.' (point, separator)

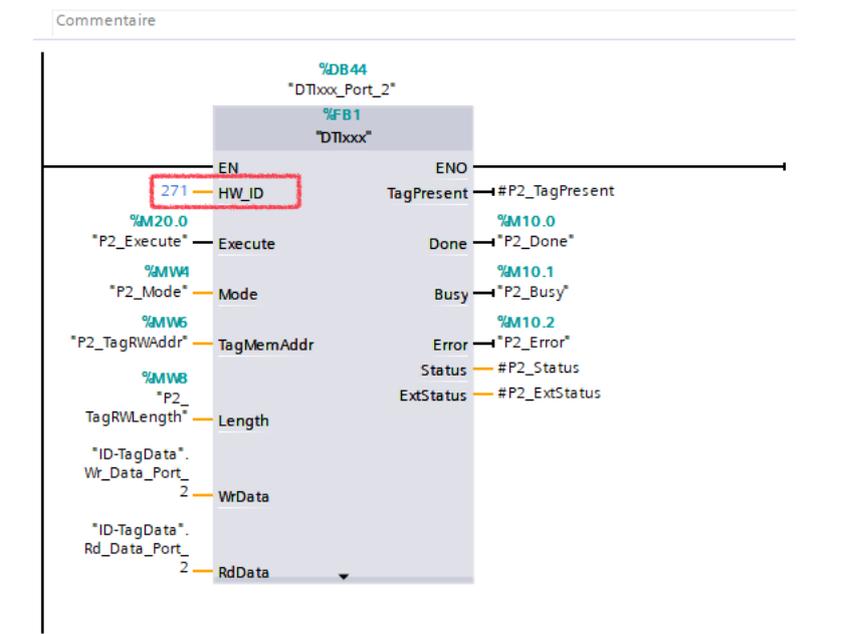
Une page de configuration de paramètres sera disponible. C'est possible de visualiser les capteurs qui sont liés au réseau et aussi éditer ses paramètres. Une autre option est le choix de l'adresse IP de l'appareil selon le besoin. Après tous les changements nécessaires, il faut sauvegarder les informations. Alors, on va en haut à droite (jaune) et on clique sur **écrire de l'appareil**.

ANEXO C – Tutorial RFID

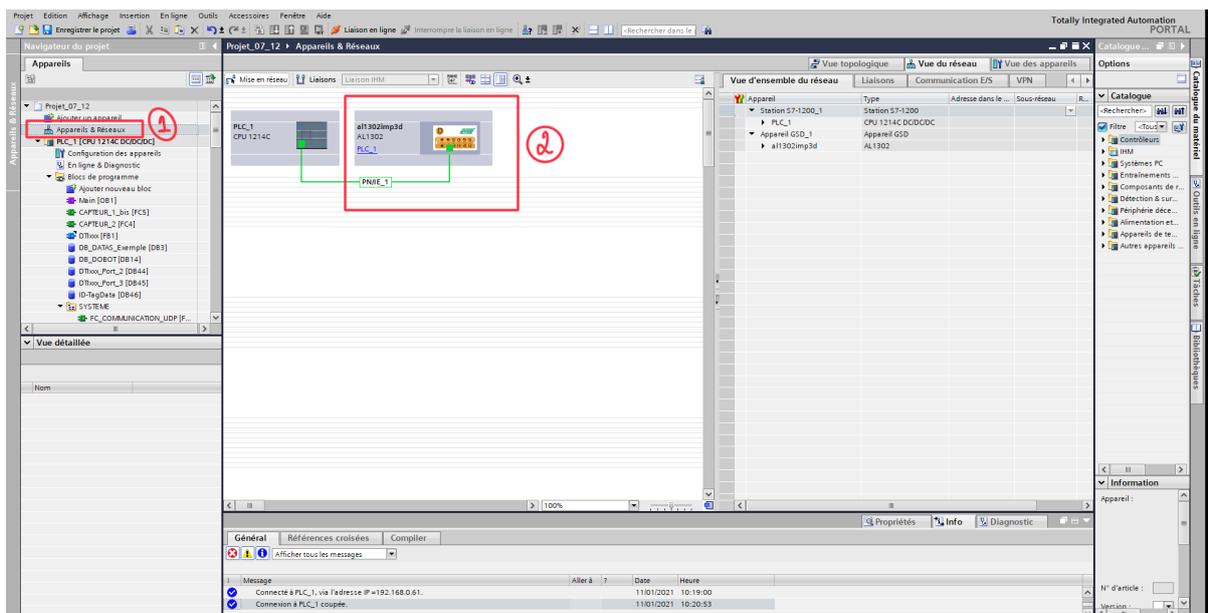
Tuto – RFID

Configuration du bloc système

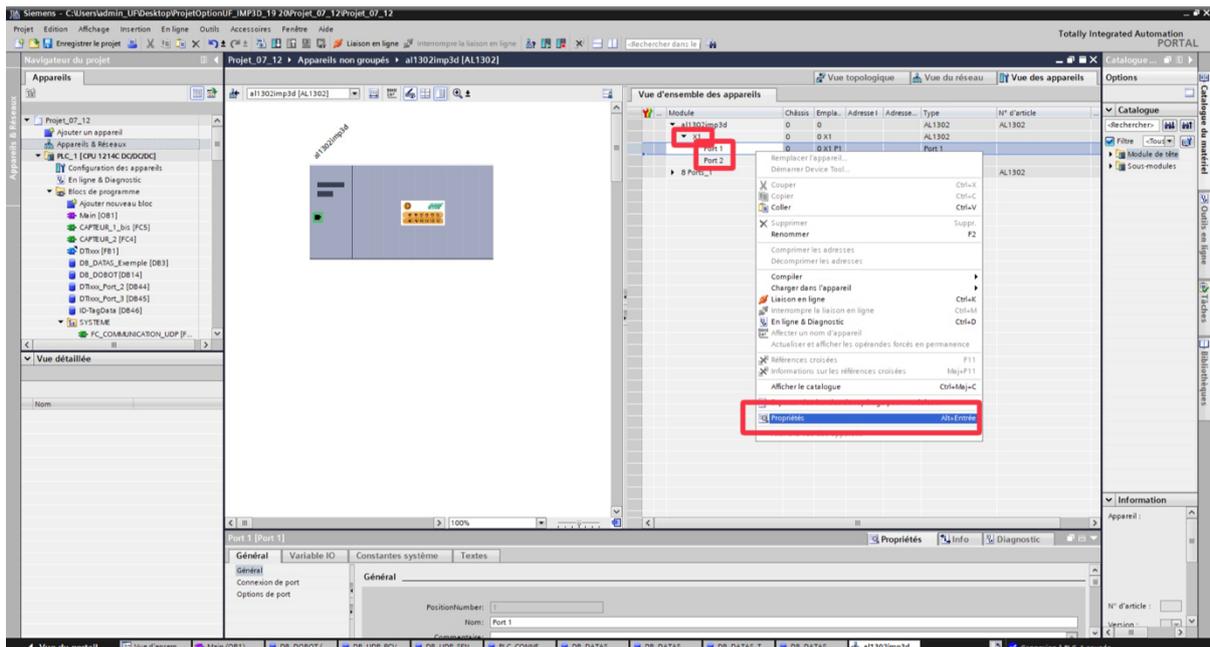
La fonction qui permet le fonctionnement des tags est déjà prêt et disponible sur : <https://www.ifm.com/fr/fr/product/DTI513?tab=documents>. Après sa colocation sur le projet, un bloc est créé avec plusieurs entrées et sorties.



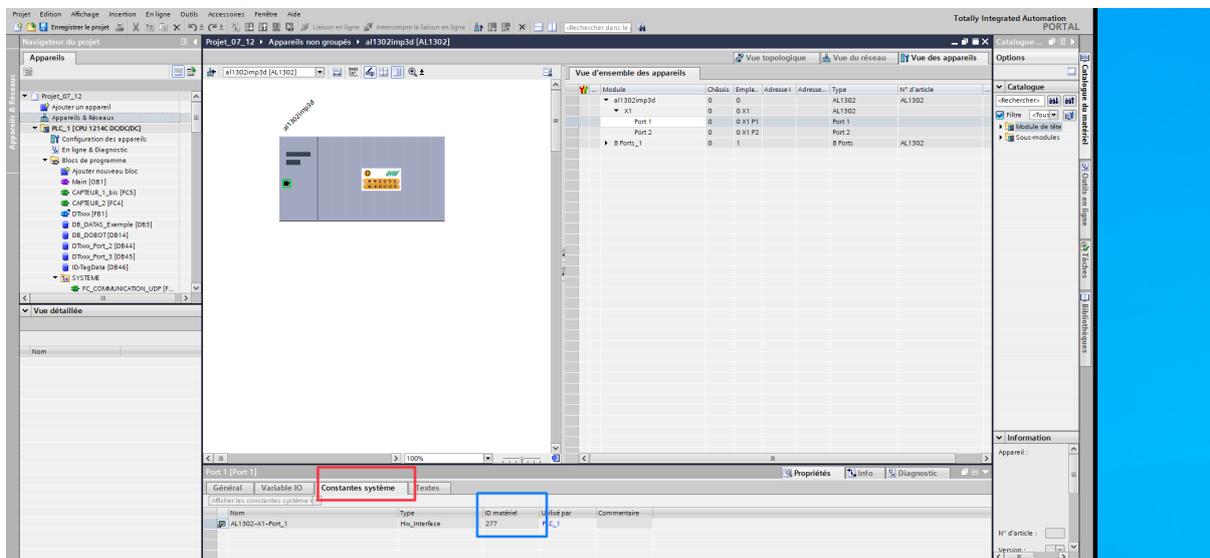
Il y a un paramètre qui a besoin d'être modifié : il s'agit de l'identification hardware du dispositif.



Il faut d'abord aller dans l'onglet **Appareils & Réseau** (1). Ensuite, on clique sur l'appareil **al302imp3d** (2). Une vue d'ensemble du réseau s'ouvrira à droite.



Dans la vue d'ensemble, on clique sur **X1**. Deux configurations de ports vont ouvrir (les deux capteurs qui ont été branchés). Puis, on clique sur une de ces deux avec le bouton à droite de la souris et pour finir on choisit **Propriétés**.



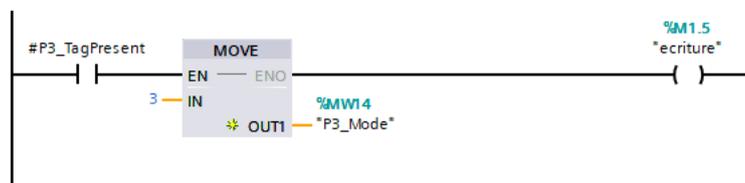
Une petite fenêtre va apparaître en bas. Il faut choisir l'onglet **Constantes système**. Ensuite, on verra une valeur décrite comme **ID matériel**. C'est cette valeur qui doit être mise comme entrée dans le bloc de la fonction du RFID (**HW_ID**).

Lecture et écriture utilisant TIA Portal

Il y a différents modes de fonctionnement du appareil comme c'est montré ci-dessous :

Valeur de commande	Mode de fonctionnement
0x00	Lecture de l'UID
0x01	Lecture automatique des données
0x02	Ecriture automatique des données
0x03	Lecture des données
0x04	Ecriture des données

Dans notre cas, on va utiliser la lecture automatique de données et l'écriture automatique de données. Pour pouvoir activer les modes dans chaque capteur, il faut créer un réseau sur TIA Portal.

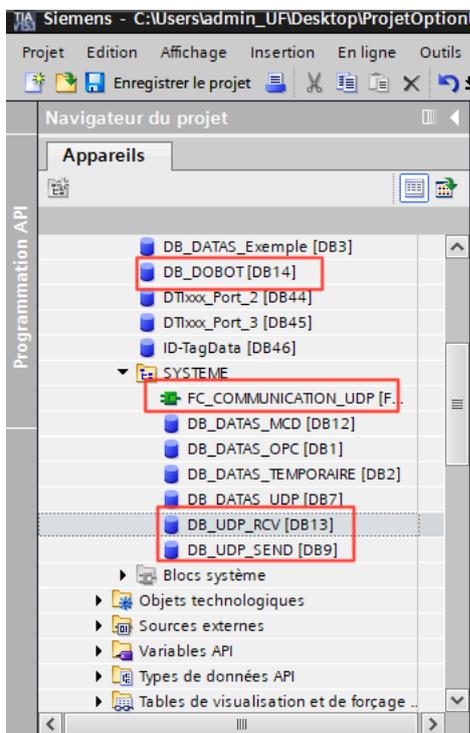


Par exemple, dans ce cas là, on veut habiliter l'écriture. Alors, si on détecte une tag dans le capteur, on copie une valeur à une variable (P3_Mode) et puis on active la variable écriture. Cette variable qui reçoit la valeur est responsable pour indiquer quel type de mode le capteur fonctionne. On va utiliser 2 pour la lecture et 3 pour l'écriture (comme vu sur la datasheet). Ces variables accompagnent le programme qui est disponible pour habiliter le fonctionnement des capteurs.

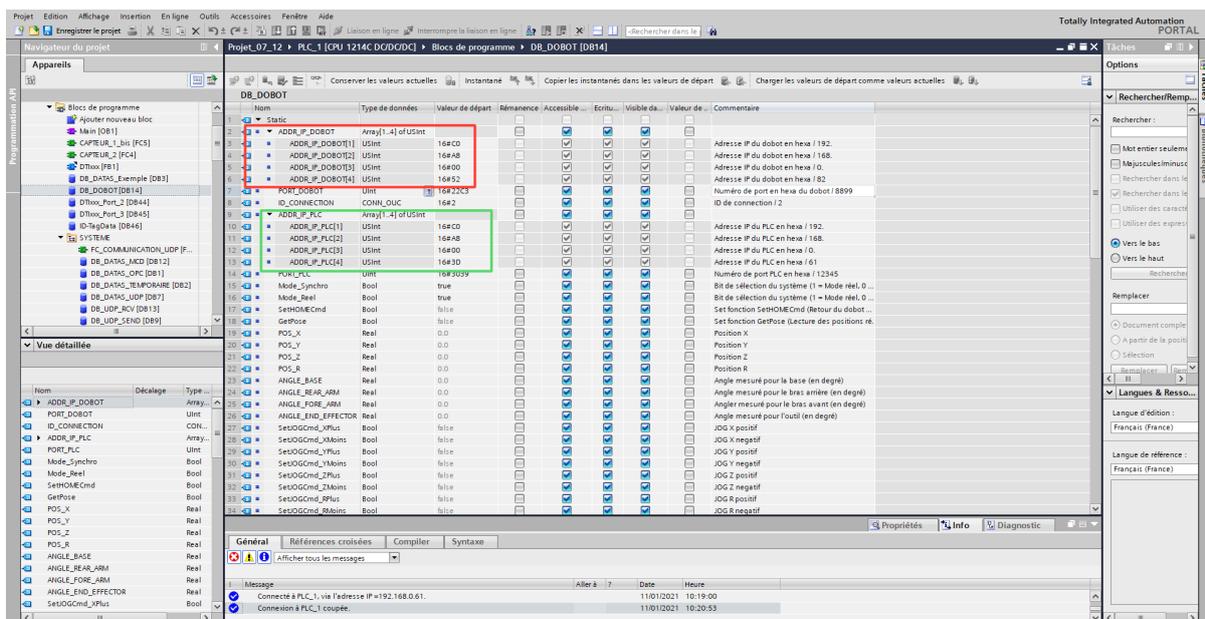
ANEXO D – Tutorial Modificações UDP

Tuto – Modifications UDP

Après l'ouverture du projet, des modifications nécessaires seront faites en 4 blocs du système : **DB_DOBOT**, **DB_UDP_RCV**, **DB_UDP_SEND** et **PLC_COMMUNICATION** (dedans le Blocs système).



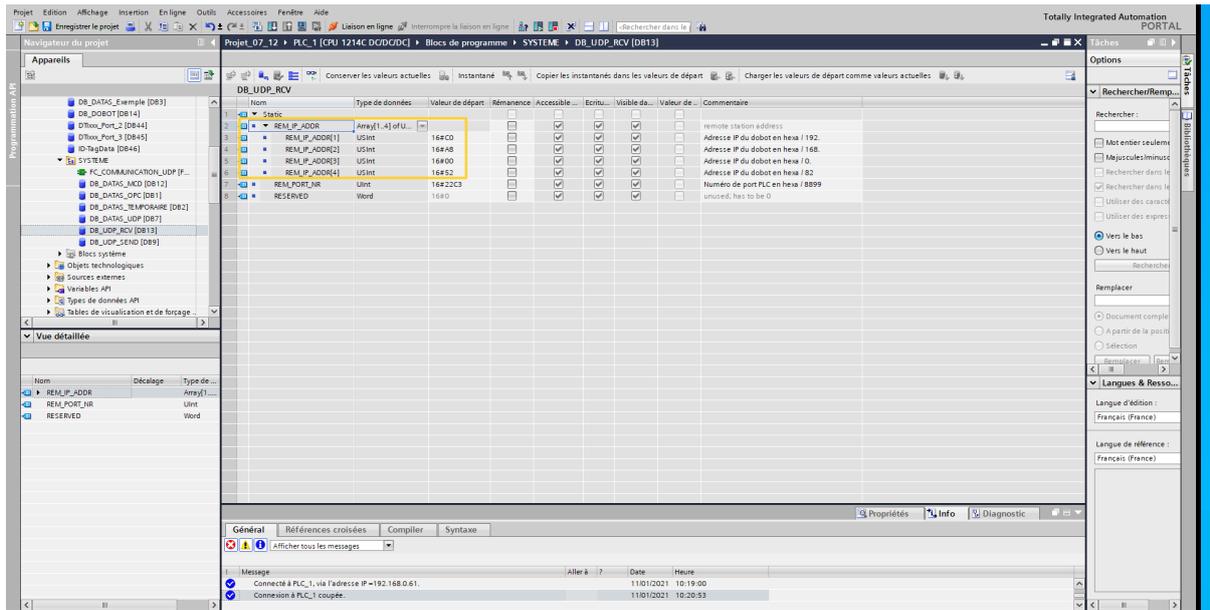
DB_DOBOT :



Deux modifications ont besoin d'être faites : l'adresse IP du DOBOT et l'adresse IP du PLC. Il faut que les deux soient convertis en hexadécimal.

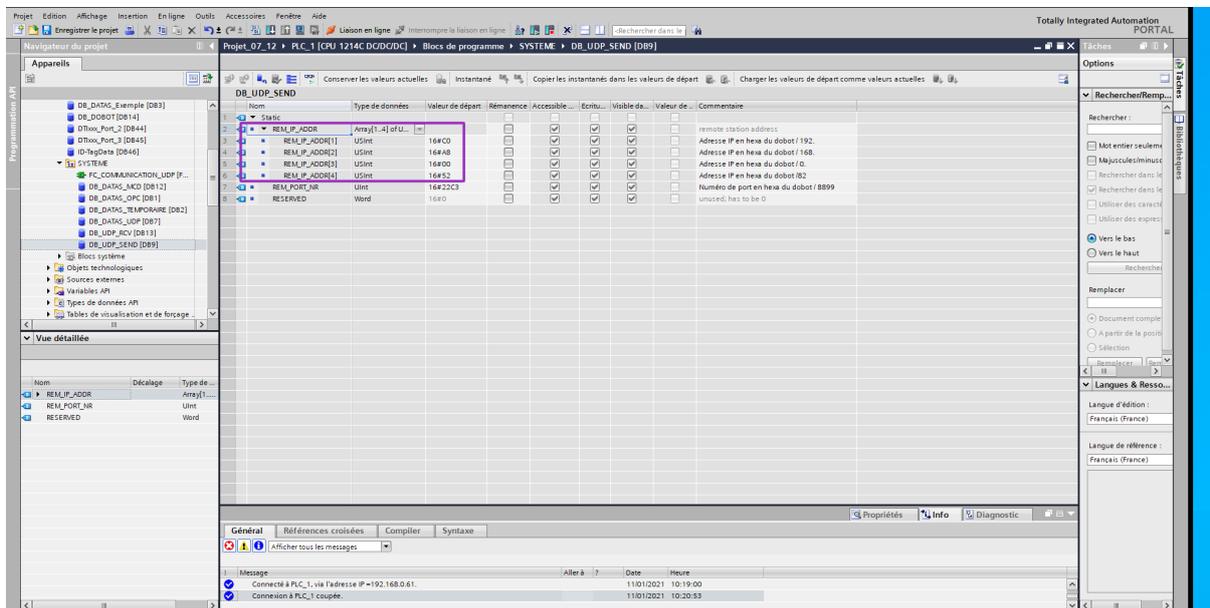
DB_UDP_RCV :

PAVANELLI – RODRIGUES – ROGER – VALENTIN
 Projet UF - 2020/2021



Dans le bloc de réception de données, il faut ajouter l'adresse IP du DOBOT en hexadécimal.

DB_UDP_SEND :



Dans le bloc d'envoi de données, il faut ajouter l'adresse IP du DOBOT en hexadécimal.

PLC_CONNECTION_UDP :

The screenshot displays the configuration of the PLC_CONNECTION_UDP block in the SIMATIC Manager. The main table lists the parameters and their values:

Nom	Type de données	Valeur de départ	Rémanence	Accessible	Entré	Viable de	Valeur de	Commentaire
Static								
Interfaceid	HW_ADDR	64						Identificateur of E-Interface submodule
ID	CONN_DUC	2						connection reference / identifier
ConnectionType	Byte	19						type of connection: 11=TCP/IP, 19=UDP (17=TCP)
ActiveEstablished	Bool	false						active/passive connection establishment
RemoteAddress	Array(1..4) of Byte							remote IP address (IPv4)
ADDR[1]	Byte	166C						IPv4 address
ADDR[2]	Byte	0D						IPv4 address (hexa)
ADDR[3]	Byte	6A						IPv4 address (hexa)
ADDR[4]	Byte	82						IPv4 address (hexa) (82)
RemotePort	UInt	0						remote UDP/TCP port number
LocalPort	UInt	12345						local UDP/TCP port number

The status bar at the bottom shows a message: "Connecté à PLC_1, via l'adresse IP = 192.168.0.61." and "Connexion à PLC_1 coupée." with a timestamp of 11/01/2021 10:20:53.

Dans le bloc d'envoi de configuration du PLC, il faut ajouter l'adresse IP du DOBOT en hexadécimal.

ANEXO E – Tutorial DOBOT - Wireshark

DIAKHATE
Mame Diouma

Tutoriels

Situations:

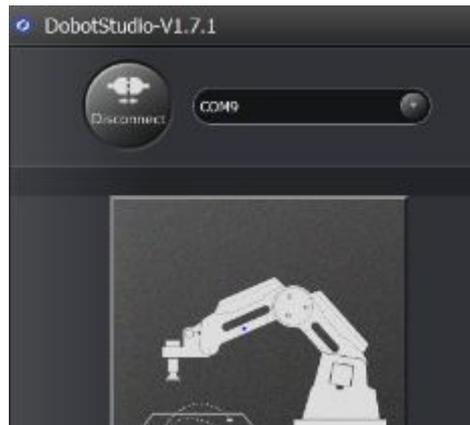
- Nom du WIFI : edimax
- Mot de passe: 12345678
- Adresse du routeur WIFI :1 92.168.0.1

Configurer un module wifi du Dobot

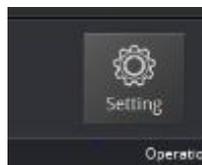


WiFi module

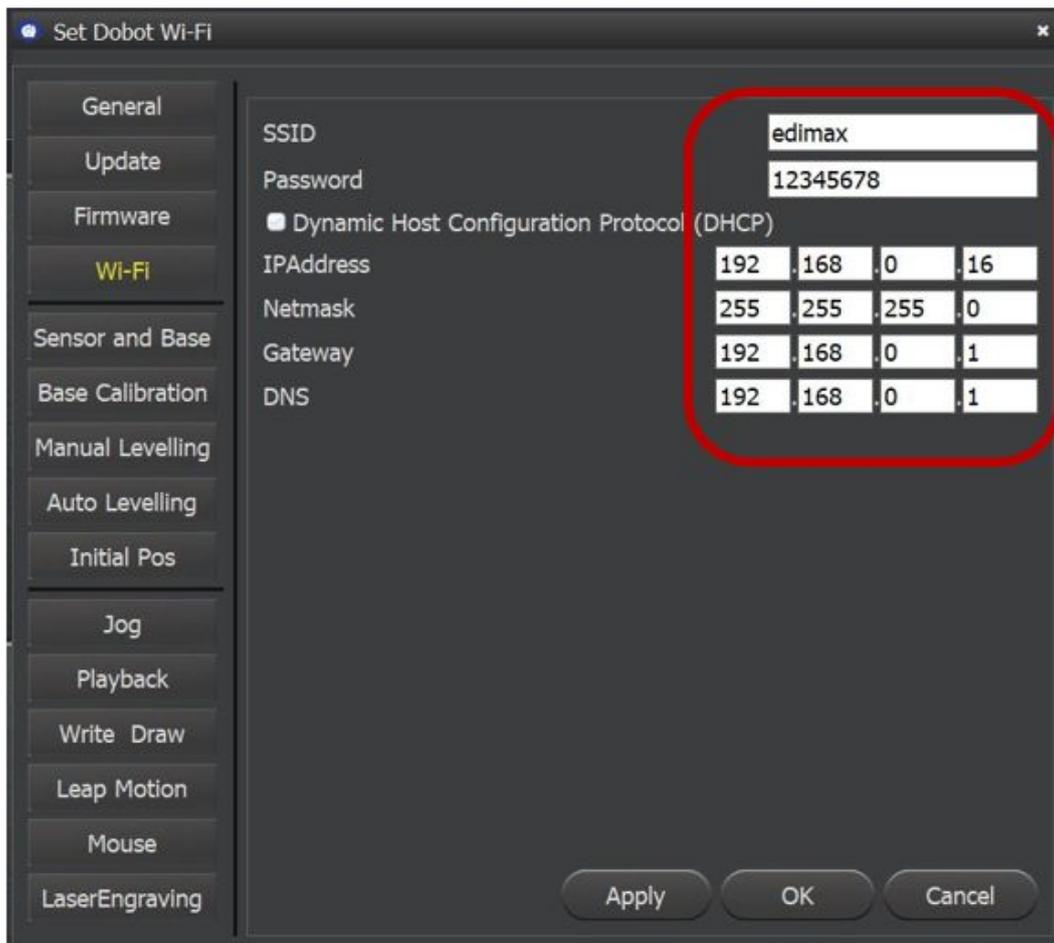
1. Brancher le câble USB sur le dobot.
2. Ouvrir DobotStudio
3. Connecter le COM9



4. Aller sur Settings



5. Cliquer sur Wifi tr emplier les champs encadrés en rouge

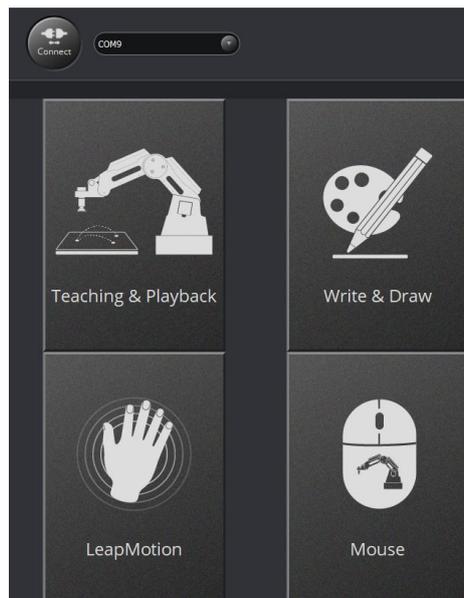


- Remplir le nom Wifi dans la zone de saisie "SSID";
- Remplir le mot de passe Wifi dans la zone de saisie "Mot de passe";
- Remplir "IP Address", "Netmask", "Gateway", "DNS". L'IP Address correspond à l'adresse qu'on veut donner au robot. Dans la Gateway, mettez l'adresse du routeur WIFI de la salle.

Cliquer sur OK pour terminer la configuration. Attendez quelques secondes, et le voyant vert du module sans fil WiFi du Dobot sera allumé, ce qui signifie que le robot est bien connecté.

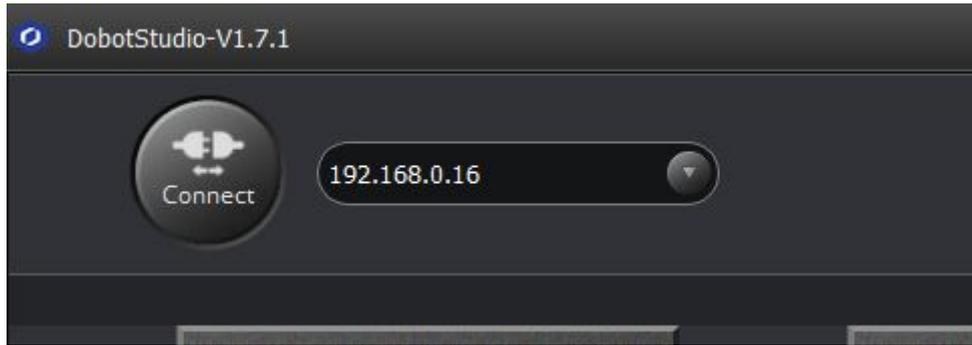


6. Déconnecter le COM9



7. Débrancher le câble USB du robot

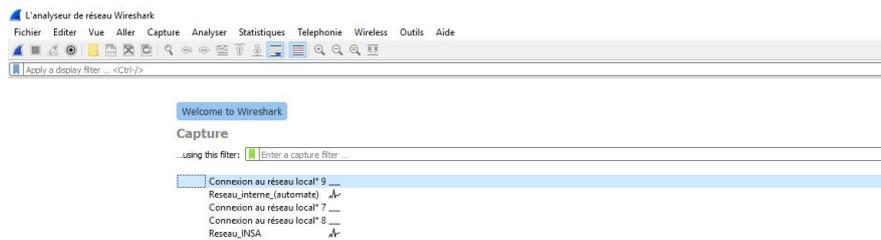
8. Voir si l'adresse donnée s'affiche correctement :



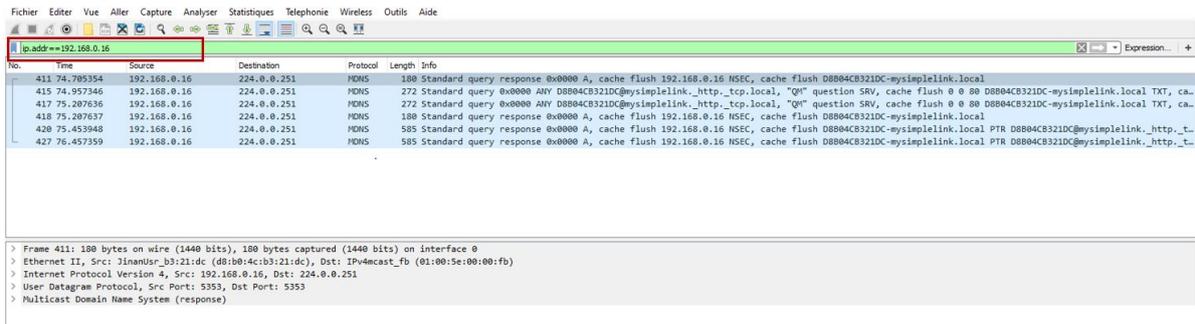
Sélectionnez l'adresse, cliquez sur Connecter, puis vous pouvez contrôler le Dobot par le biais du WiFi. Pour les utilisations futures, il suffit d'insérer le module WiFi et de connecter le Dobot par l'adresse.

Filtrer une adresse IP sur Wireshark

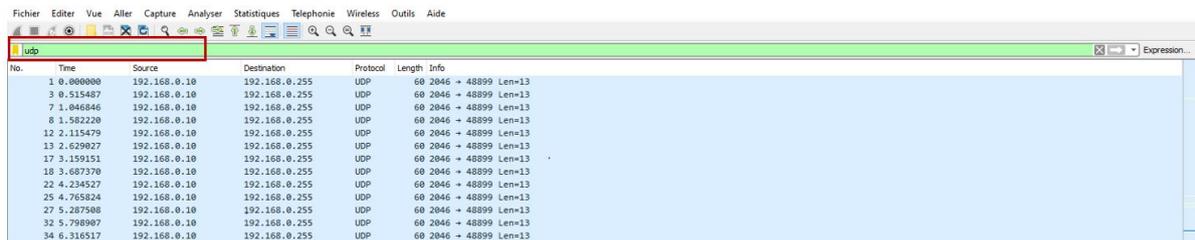
1. Ouvrir Wireshark



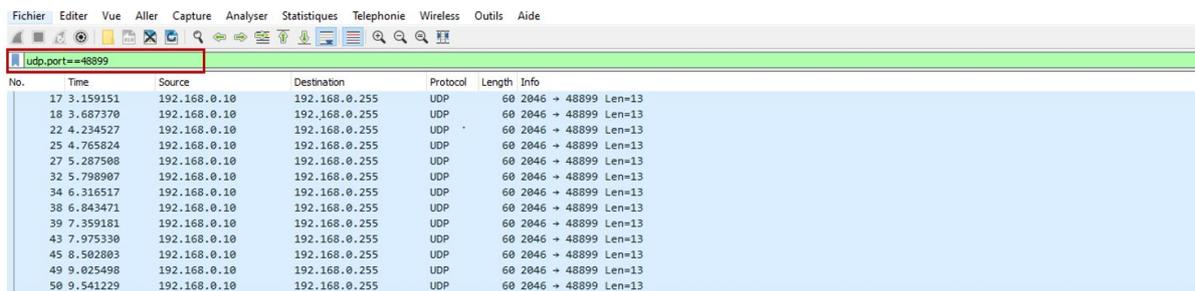
2. Sélectionner le réseau " Réseau_interne_(automate) "
3. Pour filtrer une adresse sur Wireshark, utiliser la commande ***ip.addr = "..."*** comme suit:



4. Pour filtrer les communications utilisant le protocole UDP, écrire **udp** sur la zone de filtrage.



5. Pour filtrer un port de communication utilisant le protocole UDP par exemple, écrire la commande **udp.port== "..."** comme suit:



Visualiser les trames envoyées entre le Dobot et l'ordinateur via Wireshark

1. Aller sur Wireshark
2. Choisir le réseau "Reseau_interne_(automate)"
3. Filtrer le port de communication 8899 utilisant le protocole UDP.
4. Envoyer une instruction au robot n'utilisant pas DobotStudio.

Exemple: GetPose qui permet de connaître la position actuelle du Dobot.



Cette commande correspond à la trame suivante: AA A0 20 A0 0F 6

5. Chercher la communication faite entre le robot (192.168.0.16) et l'ordinateur (192.168.0.20).

The screenshot shows the Wireshark interface for a network capture named "Reseau_interne_(automate)". The filter bar is set to "ludp.port==8899". The packet list pane shows a series of UDP packets between 192.168.0.16 and 192.168.0.20. Packet 233 is selected, and its details pane shows the following information:

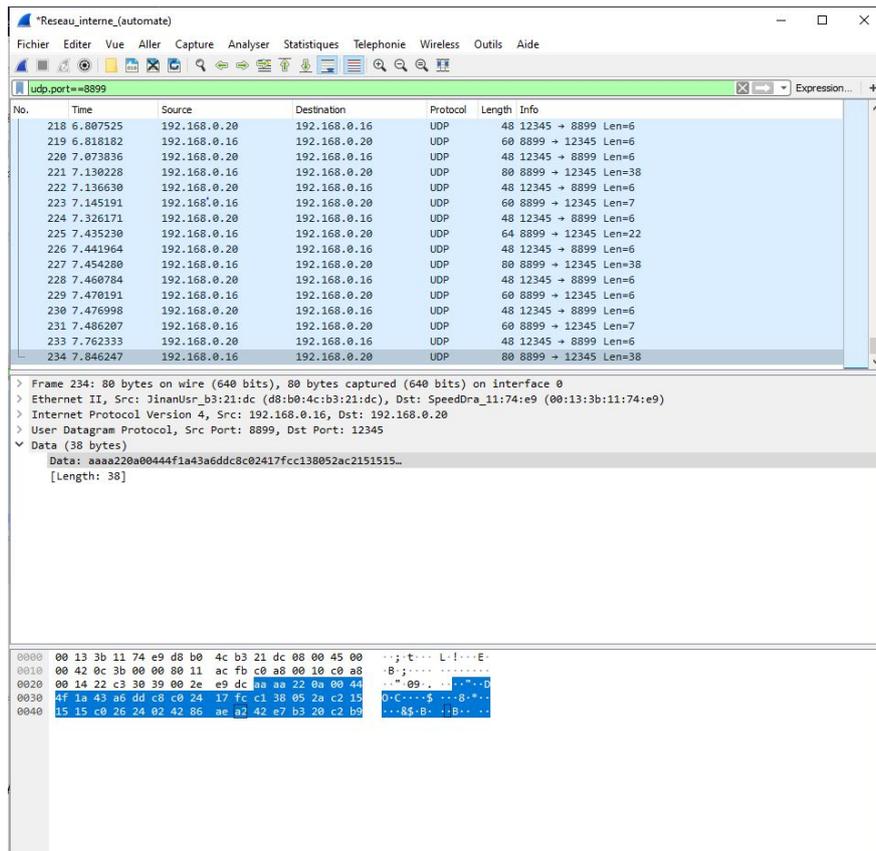
- Frame 233: 48 bytes on wire (384 bits), 48 bytes captured (384 bits) on interface 0
- Ethernet II, Src: SpeedDra_11:74:e9 (00:13:3b:11:74:e9), Dst: JinanUsr_b3:21:dc (d8:b0:4c:b3:21:dc)
- Internet Protocol Version 4, Src: 192.168.0.20, Dst: 192.168.0.16
- User Datagram Protocol, Src Port: 12345, Dst Port: 8899
- Data (6 bytes)

The raw data pane at the bottom shows the hex and ASCII representation of the captured data:

```
0000 d8 b0 4c b3 21 dc 00 13 3b 11 74 e9 08 00 45 00 ..L.!...;t...E-
0010 00 22 59 a0 00 00 80 11 00 00 c0 a8 00 14 c0 a8 .."Y.....
0020 00 10 30 39 22 c3 00 0e 81 94 aa aa 02 0a 00 fe ..09"....
```

La dernière communication fait transmettre l'instruction AAA A0 20 A0 0F 6.

et le robot répond ceci:



En bleu on a la position actuelle dud obot.

Visualiser les communications faites entre le Dobot et l'automate via WireShark

1. Aller sur WireShark
2. Envoyer une instruction aud obotv iau nI HMo ud irectements urT IAp ortal.
3. Choisir le réseau "Reseau_interne_(automate)"
4. Filtrer les communications utilisant le protocoleA RP,é crire **arp** sur laz oned e filtrage.

No.	Time	Source	Destination	Protocol	Length	Info
313	227.588637	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
314	228.586196	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
315	229.586201	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
316	230.595631	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
317	231.594040	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
319	232.594168	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
320	233.541620	SpeedDra_11:74:e9	Broadcast	ARP	42	Who has 192.168.0.250? Tell 192.168.0.20
321	233.603572	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
324	234.602199	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
325	235.602137	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
326	236.612894	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
327	237.610210	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
328	238.610085	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
329	239.620965	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
331	240.618173	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
332	241.618265	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
334	242.628980	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62

Définition Protocole ARP:

L'**Address Resolution Protocol** (ARP, protocole de résolution d'adresse) est un protocole utilisé pour traduire une adresse de protocole couche réseau (typiquement une adresse IPv4) en une adresse de protocole de couche liaison (typiquement une adresse MAC).

Fonctionnement du protocole ARP:

Un ordinateur connecté à un réseau informatique souhaite mettre en contact une destination d'un autre ordinateur dont il connaît l'adresse IP placée dans le même sous-réseau. Dans ce cas, cet ordinateur va lancer une émission en tentant de effectuer une requête ARP en broadcast au niveau 2. Cette requête est de type « Quelle est l'adresse MAC correspondant à l'adresse IP adresse IP? Répondez à mon Adresse IP ».

Puisqu'il s'agit d'un broadcast, tous les ordinateurs du segment vont recevoir la requête. En observant son contenu, ils pourront déterminer quelle est l'adresse IP sur laquelle porte la recherche. La machine qui possède cette adresse IP sera la seule à répondre en envoyant à la machine émettrice une réponse ARP du type « Je suis l'adresse IP, mon adresse MAC est adresse MAC ». Pour émettre cette réponse au bon ordinateur, il crée une entrée dans son cache ARP à partir des données contenues dans la requête ARP qu'il vient de recevoir.

La machine à l'origine de la requête ARP reçoit la réponse, mais elle ne peut donc envoyer à l'ordinateur concerné le message qu'elle avait tenté.

Il suffit donc d'un broadcast ou d'un unicast pour créer une entrée dans le cache ARP de deux ordinateurs.

5. Chercher la communication faisant appel à l'adresse du Dobot (ici l'adresse 192.168.0.16), à celle de l'automate (192.168.0.21) ou celle de l'IHM (192.168.0.22).

*Reseau_interne_(automate)

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

arp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
2	0.999979	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
3	2.007005	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
4	3.007078	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
5	4.004279	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
6	5.014718	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
7	5.034550	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.16? Tell 192.168.0.21
8	6.012054	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
9	7.012304	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
11	8.022977	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
12	9.020081	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
13	10.020157	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
14	11.027197	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
15	12.024155	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
16	13.024359	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
17	14.034873	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62
18	15.032545	SiemensI_16:4b:d3	Broadcast	ARP	60	Who has 192.168.0.61? Tell 192.168.0.62

6. Faire clic droit sur la ligne, choisir *filtered conversation* et sélectionner ethernet.

On obtient une liste de toutes les communications faites pendant la session sans avoir elles effectuées directement avec l'automate à travers l'IHM.

*Reseau_interne_(automate)

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

eth.addr eq ac:64:17:65:7f:c7 and eth.addr eq ff:ff:ff:ff:ff:ff

No.	Time	Source	Destination	Protocol	Length	Info
7	5.034550	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.16? Tell 192.168.0.21
27	20.093647	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
102	80.097164	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
114	86.302782	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.16? Tell 192.168.0.21
197	140.093421	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
280	200.092875	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
355	260.093148	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
448	320.092947	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
523	380.093227	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
606	440.092812	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
688	500.093339	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
766	560.092967	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
840	620.092913	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
936	680.093341	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21

7. En cliquant sur la communication entre 192.168.0.16 et l'automate (192.168.0.21), on a l'état suivant:

Reseau_interne_automate

Fichier Editer Vue Aller Capture Analyser Statistiques Telephone Wireless Outils Aide

eth.addr eq ac:64:17:65:7f:c7 and eth.addr eq ff:ff:ff:ff:ff:ff

No.	Time	Source	Destination	Protocol	Length	Info
7	5.834558	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.16? Tell 192.168.0.21
27	20.093847	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
102	80.097164	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
114	96.380782	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.16? Tell 192.168.0.21
197	148.093421	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
288	208.092875	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
355	268.093148	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
448	328.092947	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
523	388.093227	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
606	448.092812	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
688	508.093339	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
765	568.092967	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
848	628.092913	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
936	688.093341	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
1014	748.092873	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
1107	808.093548	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21
1181	868.093478	Siemens_65:7f:c7	Broadcast	ARP	60	Who has 192.168.0.22? Tell 192.168.0.21

> Frame 7: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0

> Ethernet II, Src: Siemens_65:7f:c7 (ac:64:17:65:7f:c7), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> Destination: Broadcast (ff:ff:ff:ff:ff:ff)

> Source: Siemens_65:7f:c7 (ac:64:17:65:7f:c7)

Type: ARP (0x0806)

Padding: 00000000000000000000000000000000

> Address Resolution Protocol (request)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: request (1)

Sender MAC address: Siemens_65:7f:c7 (ac:64:17:65:7f:c7)

Sender IP address: 192.168.0.21

Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)

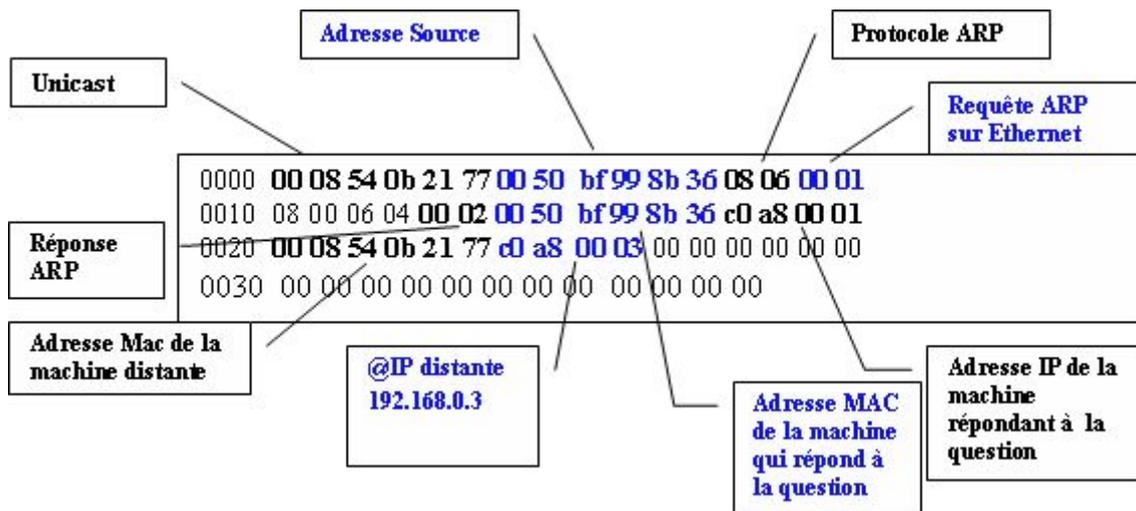
Target IP address: 192.168.0.16

```

0000 ff ff ff ff ff ac 64 17 65 7f c7 08 06 00 01 .....d .....
0010 00 00 00 00 00 00 01 ac 64 17 65 7f c7 08 00 15 .....d .....
0020 00 00 00 00 00 c0 a8 00 10 00 00 00 00 00 .....
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

Avec:



ANEXO F – Tutorial Biblioteca IFM

Tutoriel importation bibliothèque AL1302 ifm sur Tia Portal

- Aller sur le site ifm et taper dans la barre de recherche : GSD al1302. Il y a normalement qu'un seul résultat dans téléchargement :

Résultat de recherche

Votre recherche sur gsd al1302 a donné 1 résultat.



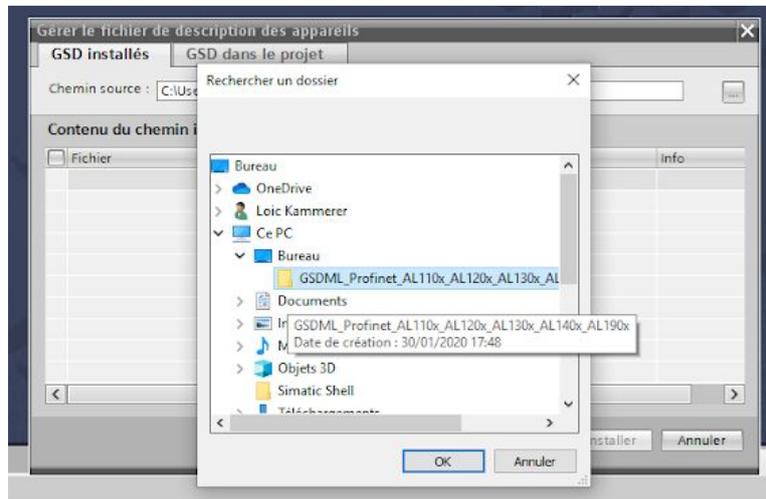
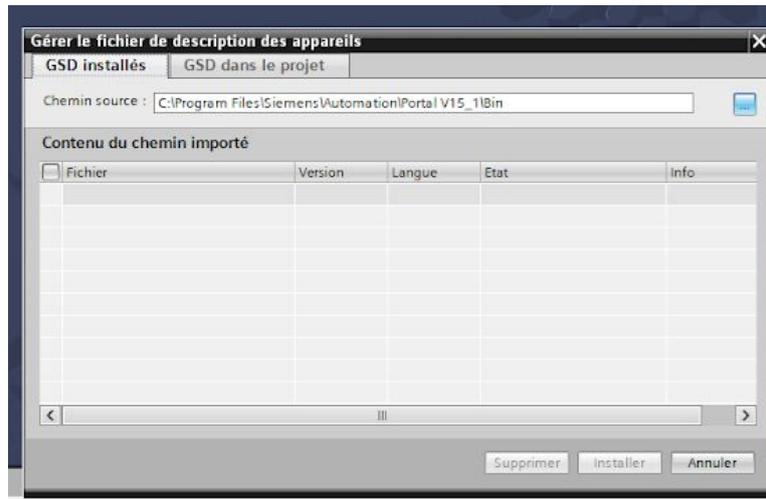
- Se connecter à son compte ifm (facile à créer) et télécharger le fichier suivant puis le décompresser :



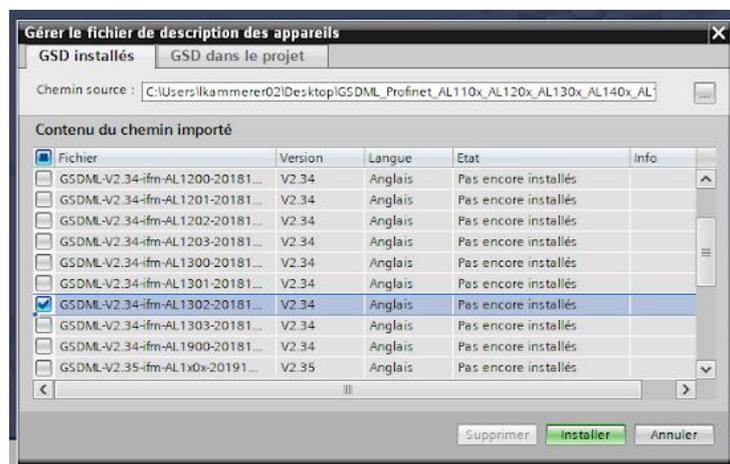
- Aller ensuite sur Tia Portal l'onglet "Outils" et cliquer sur "Gérer le fichier de description des appareils" :



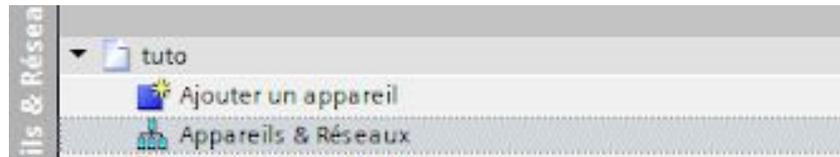
- Parcourir les fichiers et trouver celui téléchargé et décompressé précédemment :



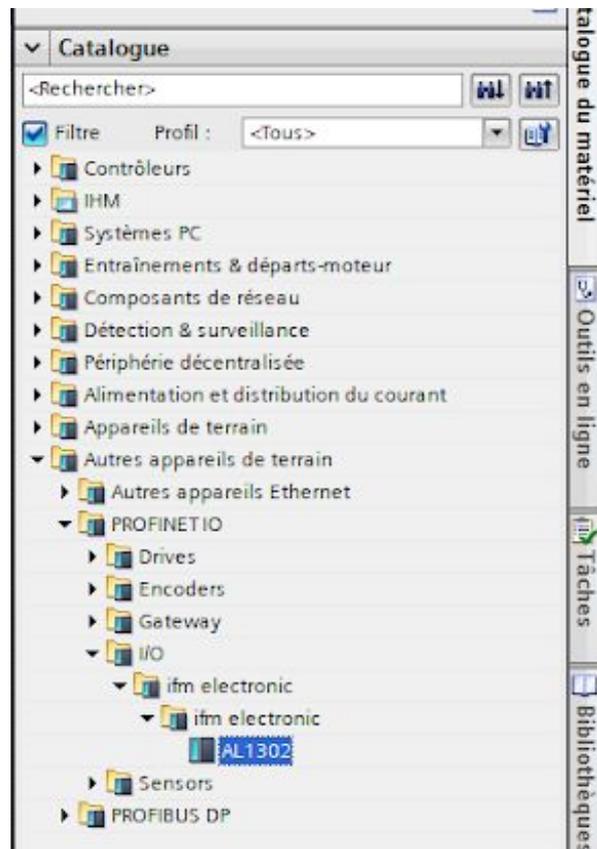
- Vous avez maintenant tous les fichiers correspondant aux maîtres ALxxx, choisissez celui qui vous intéresse et installez le :



- Une fois installé, il faut le mettre dans le réseau. Allez dans “Appareils & Réseaux”



- Pour trouver l'appareil ifm, allez maintenant dans le catalogue à droite et cliquer sur “Autres appareils de terrain” -> “PROFINET IO” -> “I/O” -> “ifm electronic” -> “ifm electronic” et choisissez votre appareil :



- Glissez alors l'appareil dans la fenêtre “Appareils & Réseaux” pour pouvoir l'utiliser :

