

## **Automatização de um Braço Robótico Combinado com o Controlo de um Tapete Rolante**

**SANTO ANTÓNIO FRANCISCO OLÍMPIO DE FREITAS**

(Licenciado em Engenharia de Automação, Controlo e  
Instrumentação)

Trabalho Final de Mestrado para obtenção do grau de Mestre  
em Engenharia Eletrotécnica - Ramo de Automação e Eletrónica Industrial

Orientador: Prof. Doutor Armando José Leitão Cordeiro

Júri:

Presidente: Prof. Doutor Luís Manuel dos Santos Redondo  
Vogais: Prof. Doutora Maria Graça Vieira Brito Almeida  
Prof. Doutor Armando José Leitão Cordeiro

abril de 2022



“Alguns homens veem as coisas como são, e dizem ‘Porquê’. Eu sonho com as coisas que nunca foram e digo “Por que não?” (George Bernard Shaw)



# Agradecimentos

Agradeço a Deus em primeiro lugar, por me ter dado sabedoria e saúde para conseguir alcançar esta conquista.

Agradeço à minha mãe pelo apoio moral que me tem dado durante estes anos todos longe de casa.

A minha noiva Jufânia Canga, minha companheira de jornada. O meu muito obrigado por levantar-me sempre que eu precisava e estender-me a mão sempre que eu pedia.

Agradeço também ao meu orientador, Professor Doutor Armando Cordeiro, que não deixou faltar nada para que este trabalho fosse possível e pelo seu apoio durante a sua realização.



# **Resumo**

Nos dias de hoje, o universo da indústria exige cada vez mais organização para que os processos de produção funcionem de forma precisa e otimizada. Para tal é necessário que exista vários tipos de equipamentos interligados entre si, entre os quais dois deles são descritos neste trabalho: braço robótico e tapete rolante ou geralmente chamado de esteira transportadora. Estes tipos de dispositivos são projetados para deslocar materiais ou objetos a uma distância pré-definida, resultando num maior dinamismo dos processos industriais. A sua versatilidade em diversos aspetos e inúmeras aplicações permite que os mesmos sejam requisitados em diversos segmentos da indústria, investigação e desenvolvimento, como é o caso da indústria automóvel, indústria alimentar, farmacêuticas, extração mineira e até em algumas áreas da construção civil, entre outros.

Com base neste princípio, desenvolveu-se este trabalho de Dissertação, com fins didáticos para que os alunos de licenciatura ou visitantes do ISEL pudessem utilizar. A solução consiste no controlo de um sistema (braço robótico apoiado por um tapete rolante) com ligação por supervisão. Este sistema é constituído pela esteira transportadora, um motor DC e um pequeno manipulador robótico com seis graus de liberdade. O controlo é feito a partir de dois microcontroladores do tipo Arduino UNO, ligados a um controlador industrial ILC 131 ETH da Phoenix Contact para a sua supervisão.

**Palavras-Chave:** Driver de motor DC, Arduino, Controlador Industrial, Tinkerkit Braccio, Protocolo RS232, Conversão RS232/TTL.



# ***Abstract***

These days, the world of industry demands more and more organization so that production and manufacturing processes work accurately and optimally. For this it is necessary that there are several types of equipment interconnected, among which two of them are described in this work: robotic arm and conveyor belt or generally called conveyor belt. These types of devices are designed to move materials or objects at a pre-defined distance, resulting in greater dynamism of industrial processes. Their versatility in several aspects and countless applications allows them to be requested in different segments of industry and research and development, such as the automobile industry, food industry, pharmaceuticals, mining and even in some areas of civil construction, among others.

Based on this principle, this dissertation work was developed, for didactic purposes for undergraduate students or visitors to ISEL, which consists of the control of a system (robotic arm supported by a treadmill) connected by supervision. This system consists of a conveyor belt, a DC motor and a small robotic manipulator with six degrees of freedom. Control is performed using two Arduino UNO type microcontrollers, connected to an industrial controller ILC 131 ETH from Phoenix Contact for supervision.

**Keywords:** DC motor driver, Arduino, PLC ILC 131 Eth, Tinkerkit Braccio, RS232 protocol, RS232 / TTL conversion.



# Índice

Agradecimentos .....	v
Resumo.....	vii
Abstract .....	ix
Índice.....	xi
Lista de Figuras.....	xiii
Lista de Tabelas .....	xvi
Lista de Siglas e Acrónimos.....	xviii
<b>1 Introdução.....</b>	<b>3</b>
<b>1.1 Objetivos .....</b>	<b>4</b>
<b>1.2 Estrutura .....</b>	<b>4</b>
<b>2 Estado da Arte .....</b>	<b>7</b>
<b>2.1 Evolução da Automação e Robótica .....</b>	<b>7</b>
<b>2.2 Robótica Industrial .....</b>	<b>8</b>
<b>2.2.1 Objetivo da Robótica Industrial .....</b>	<b>8</b>
<b>2.2.2 Classificação dos Robôs.....</b>	<b>9</b>
<b>2.2.3 Anatomia dos Robôs.....</b>	<b>10</b>
<b>2.3 Breve Historia dos Tapetes Rolantes .....</b>	<b>12</b>
<b>2.3.1 Constituição dos tapetes rolantes.....</b>	<b>15</b>
<b>2.3.2 Aplicações de Sistemas de Tapete Rolante.....</b>	<b>15</b>
<b>2.4 Motores DC .....</b>	<b>17</b>
<b>2.4.1 Constituição, Princípio de Funcionamento .....</b>	<b>17</b>
<b>2.4.2 Controlo de Motores DC.....</b>	<b>19</b>
<b>2.4.3 Controlo da Velocidade do Motor DC por PWM.....</b>	<b>21</b>
<b>2.5 Microcontrolador Arduino.....</b>	<b>23</b>
<b>2.5.1 Origem do Arduino .....</b>	<b>24</b>
<b>2.5.2 Tipos de Arduino .....</b>	<b>25</b>
<b>2.6 Comunicação RS232 .....</b>	<b>25</b>
<b>3 Solução desenvolvida.....</b>	<b>31</b>
<b>3.1 Descrição do Robô Manipulador usado.....</b>	<b>31</b>
<b>3.2 Descrição do Tapete Rolante Utilizado .....</b>	<b>37</b>
<b>3.2.1 Características do Motor DC usado.....</b>	<b>39</b>
<b>3.2.2 Aplicação do PWM no Circuito Controlado .....</b>	<b>41</b>
<b>3.2.3 Controlador de Velocidade e Sentido de rotação.....</b>	<b>43</b>
<b>3.3 PLC ILC 131 ETH Phoenix Contact.....</b>	<b>48</b>
<b>3.4. Caraterísticas do Arduino UNO usado .....</b>	<b>52</b>

<b>3.5</b>	<b>Comunicação RS232 entre o PLC ILC 131 ETH e o Arduíno UNO.....</b>	<b>54</b>
3.5.1	<i>Configuração física RS232 no PLC ILC 131 ETH .....</i>	55
3.5.2	<i>Bloco RS232_INIT.....</i>	57
3.5.3	<i>Bloco RS232_RECEIVE &amp; RS232_SEND .....</i>	58
3.5.4	<i>Configuração RS232 no Arduíno .....</i>	59
3.5.5	<i>Conversor RS232 para TTL.....</i>	59
3.5.6	<i>Ambiente de desenvolvimento WebVisit.....</i>	61
3.5.7	<i>Aspeto final da aplicação desenvolvida.....</i>	64
<b>4</b>	<b>Resultados experimentais .....</b>	<b>71</b>
4.1	<b>Ponto 1 (Home).....</b>	<b>71</b>
4.2	<b>Ponto 2 (Pick).....</b>	<b>74</b>
4.3	<b>Ponto 3 (Fechar a garra).....</b>	<b>76</b>
4.4	<b>Ponto 4 (Place).....</b>	<b>79</b>
4.5	<b>Ponto 5 (Abrir a garra).....</b>	<b>81</b>
<b>5</b>	<b>Conclusões .....</b>	<b>87</b>
5.1	<b>Perspetivas de trabalho futuro .....</b>	<b>87</b>
<b>6</b>	<b>Referências .....</b>	<b>88</b>
<b>Anexo A</b>	<b>.....</b>	<b>2</b>

## ***Lista de Figuras***

Figura 1 - Anatomia dos <i>robôs</i> .....	10
Figura 2 - Configurações base dos <i>robôs</i> .....	11
Figura 3 - Graus de Liberdades de um <i>Robô</i> .....	12
Figura 4 - Esteira Transportadora Industrial .....	14
Figura 5 - Constituição Tapete Rolante .....	<b>Erro! Marcador não definido.</b>
Figura 6 - Sistema de Embaladora OMRON [4].....	16
Figura 7 - Partes Constituintes de um Motor DC [5] .....	18
Figura 8 - Tipos de máquina de corrente contínua [6] .....	18
Figura 9 - Ilustração do princípio de funcionamento de um motor DC .....	19
Figura 10 - Conversor CC/CC Quatro Quadrantes [6] .....	20
Figura 11 - Circuito de Controlo Linear.....	21
Figura 12 - Sinal PWM .....	22
Figura 13 - Aplicação do Sinal PWM .....	23
Figura 14 - Sinais PWM com Frequências Idênticas e Diferentes <i>Duty Cycles</i> ...23	
Figura 15 - Microcontrolador Arduíno UNO .....	24
Figura 16 – Esquema de Comunicação Série .....	26
Figura 17 - Ligações Previstas na norma RS-232-C.....	26
Figura 18 - Sequência Genérica de bits na transmissão série assíncrona de uma palavra.....	28
Figura 19 - Robô Manipulador com Seis Graus de Liberdade [9] .....	32
Figura 20 - Ilustração robô Braccio.....	33
Figura 21 - Arduíno Braccio Shield.....	35
Figura 22 - Esquema de Ligação Servodrive.....	35
Figura 23 - Braccio Teach 1 .....	37
Figura 24 - Tapete Rolante.....	38
Figura 25 – Motor e correia do Tapete .....	38
Figura 26 - Esquema de polias acopladas por correia [11].....	39
Figura 27 - Motor DC 28PA51G da Botnroll [12].....	39
Figura 28 - Terminais de ligação do motor DC com interface de encoder .....	40
Figura 29 - PWM Aplicado ao <i>Chopper</i> de 4 Quadrante.....	42
Figura 30 – Aplicação Real do Sinal PWM na Ponte H .....	42
Figura 31 - ponte H L298N .....	43
Figura 32 - Exemplo de Aplicação Ponte H .....	44
Figura 33 - Chip ST L298N.....	45
Figura 34 - Diagrama de Blocos do Chip L298N [13].....	45
Figura 35 - Esquema de Ligação Com Motores de 5V .....	47

Figura 36 - Esquema de Ligação Com Motores de 12V .....	47
Figura 37 - PLC ILC 131 ETH Phoenix Contact.....	48
Figura 38 - Detalhes de configuração do PLC ILC 131 ETH.....	49
Figura 39 - Imagem dos módulos de expansão adicionados .....	49
Figura 40 - Visão geral PC WORX .....	50
Figura 41 - Raíz do projeto.....	51
Figura 42 - Barra de ferramentas .....	51
Figura 43 - Funções PC WORX .....	52
Figura 44 - Janelas de mensagem PC WORX .....	52
Figura 45 - Descrição Arduino UNO.....	53
Figura 46 - Ambiente de Programação Arduino .....	54
Figura 47 - Trama RS232 enviada pelo PLC.....	55
Figura 48 - Ligação RS232 PLC ILC 131 ETH .....	56
Figura 49 - Interface RS232 PLC ILC 131 ETH.....	57
Figura 50 - RS232 INIT PC WORX .....	57
Figura 51 - Estrutura de dados do bloco RS23_INIT .....	58
Figura 52 - Bloco RS232_RECEIVE & RS232_SEND.....	58
Figura 53 - Conversor RS232/TTL .....	59
Figura 54 - Esquema de ligação Conversor RS232/TTL .....	60
Figura 55 - Inicialização série Arduino.....	61
Figura 56 - Webvisit Express.....	62
Figura 57 - WebVisit PDD <i>Flags</i> [14].....	62
Figura 58 - WebVisit (Importar mwt) [14].....	63
Figura 59 - WebVisit (Mapeamento de Variáveis) .....	63
Figura 60 - Menu Principal .....	64
Figura 61 - Menu Robot Teach.....	65
Figura 62 - Menu Lista de Pontos - Parte I.....	65
Figura 63 -Menu Lista de Pontos - Parte II.....	66
Figura 64 - Menu AUTO .....	67
Figura 65 - Ponto 1(Home) HMI .....	71
Figura 66 - Trama RS232 PLC -> Arduino Ponto 1 .....	72
Figura 67 - Formato da Trama PLC Ponto 1.....	72
Figura 68 - Ponto 1 (Home) Real.....	73
Figura 69 - Ponto 2(Pick) HMI .....	74
Figura 70 - Trama RS232 PLC -> Arduino Ponto 2 .....	74
Figura 71 - Formato da Trama PLC Ponto 2.....	75
Figura 72 - Ponto 2 (Pick) Real .....	75
Figura 73 - Ponto 3(Fechar a garra) HMI.....	76

Figura 74 - Trama RS232 PLC -> Arduíno Ponto 3 .....	77
Figura 75 - Formato da Trama PLC Ponto 3.....	77
Figura 76 - Ponto 3 (Fechar a garra) Real.....	78
Figura 77 - Ponto 4(Place) HMI.....	79
Figura 78 - Trama RS232 PLC -> Arduíno Ponto 4 .....	79
Figura 79 - Formato da Trama PLC Ponto 4.....	80
Figura 80 - Ponto 4 (Place) Real .....	80
Figura 81 - Ponto 5(Abrir a garra) HMI .....	81
Figura 82 - Trama RS232 PLC -> Arduíno Ponto 5 .....	82
Figura 83 - Formato da Trama PLC Ponto 5.....	82
Figura 84 - Ponto 5 (Abrir a garra) Real .....	83

## ***Lista de Tabelas***

Tabela 1 - Estados Possíveis da ponte H.....	20
Tabela 2 - Correspondência Entre o sinal elétrico e valor lógico .....	27
Tabela 3 - Correspondência Entre as Fichas e os Circuitos de Ligação Série....	27
Tabela 4 - Esquema de Ligação Shield Braccio .....	35
Tabela 5 - Funções de Controlo do Motor .....	46
Tabela 6 – Descrição dos Pinos de Alimentação Arduino UNO.....	53
Tabela 7 - Trama RS232 entre PLC e o Arduino.....	54
Tabela 8 - <i>Pinout</i> RS232 PLC ILC 131 ETH.....	56
Tabela 9 - Sinais PWM Robô Ponto 1(Home) .....	73
Tabela 10 - Sinais PWM Robô Ponto 2(Pick) .....	76
Tabela 11 - Sinais PWM Robô Ponto 3(Fechar a garra).....	78
Tabela 12 - Sinais PWM Robô Ponto 4(Place) .....	81
Tabela 13 - Sinais PWM Robô Ponto 5(Abrir a garra) .....	83



## ***Lista de Siglas e Acrónimos***

PLC	<i>Programable Logic Controller</i>
$i$	Relação de Transmissão
$\emptyset_B$	Diâmetro da Polia B
$\emptyset_A$	Diâmetro da Polia A
<i>FIFO</i>	First in First out
<i>CC</i>	Corrente Contínua
<i>DC</i>	<i>Direct Current</i>
T	Binário (N.m)
KT	Constante de Binário
I	Corrente no Rotor
RPM	Rotações Por Minuto
PWM	<i>Pulse Width Modulation</i>
TBJs	<i>Transistores Bipolares de Junção</i>
Ton	<i>Tempo Ligado</i>
Toff	<i>Tempo Desligado</i>
PPR	<i>Pulsos Por Rotação</i>
ICSP	<i>In-Circuit Serial Programming</i>
SPI	<i>Serial Peripheral Interface</i>
MISO	<i>Master In Slave Out</i>
SCK	<i>Serial Clock</i>
SS	<i>Slave Select</i>
Gdl	<i>Grau de Liberdade</i>
Gdm	<i>Grau de Movimento</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
RS232	<i>Recommended Standards 232</i>
DTE	<i>Estações de Dados Terminais</i>
DCE	<i>Estações de comunicações de dados</i>
LED	<i>Light Emitter Diode</i>

# ***Capítulo 1***

## ***Introdução***

**Resumo:** O presente capítulo apresenta uma introdução ao tema da dissertação assim como o enquadramento e motivação científica para a elaboração da mesma, prosseguindo com os objetivos a concretizar e uma breve descrição da organização do trabalho.



# 1 Introdução

As novas tecnologias desempenham um papel fundamental no desenvolvimento da humanidade e das sociedades em geral. Elas são responsáveis por inúmeros contributos para a melhoria da qualidade da vida, além de serem um dos mais fortes aspetos impulsionadores da maioria das economias mundiais.

Os sistemas automatizados ocupam um lugar de destaque nos objetivos de competitividade e produtividade de muitas indústrias, tendo estes sistemas progredido através dos avanços tecnológicos registados ao longo dos últimos anos, tendo-se atingido atualmente um grau de complexidade e sofisticação nunca antes visto. Em muitos casos, estes sistemas são compostos por subsistemas com origens de diferentes fabricantes, com diferentes soluções e funcionalidades, sendo exigido a estes que desempenhem funções de controlo ou de segurança com elevada fiabilidade. A evolução dos sistemas automatizados tem ocorrido dentro de todos os níveis dos processos industriais, desde níveis mais baixos como sensores e atuadores com funções programáveis, até um nível mais alto com controladores com elevada capacidade de diagnóstico de falhas a sistemas avançados de supervisão.

Esta evolução nos sistemas automatizados é normalmente fruto de um processo contínuo com pequenos avanços num sistema globalizado em constante mudança e como tal, é necessário dotá-los com capacidades de integração em novos sistemas e tecnologias. Isto é fundamental, uma vez que, existe a incapacidade de implementar novos sistemas automatizados a cada evolução tecnológica, principalmente devido aos custos que estes envolvem.

Esta dissertação de mestrado é então focada na área de automação industrial e robótica tentando de alguma forma replicar conceitos e aplicações semelhantes às encontradas no mundo real. Para tal adquiriu-se um braço robótico e também um tapete rolante para apoio às atividades do braço robótico. Dado que os equipamentos industriais apresentam custos de aquisição elevadíssimos, adquiriram-se equipamentos didáticos a preços muito mais reduzidos, mas que permitem em grande parte demonstrar os conceitos e os conhecimentos adquiridos no Mestrado em Engenharia Eletrotécnica do Instituto Superior de Engenharia do ISEL.

## 1.1 Objetivos

Esta dissertação foi desenvolvida no âmbito do Mestrado em Engenharia Eletrotécnica, ramo de Automação e Eletrónica industrial do ISEL. Teve como objetivo implementar uma aplicação de controlo para um braço robótico manipulador com seis graus de liberdade apoiado por um tapete rolante, solução esta com ligação por supervisão, para fins didáticos e para tal seguiu-se os seguintes pontos:

- Desenvolver modos de controlo do manipulador e do tapete com microcontrolador Arduíno, tais como ensinamento de pontos, gravação de pontos, ordem de marcha para ambos, mudança no sentido de rotação do tapete e variação na velocidade de rotação;
- Efetuar a comunicação do microcontrolador Arduíno com o PLC ILC 131 ETH através do protocolo RS232, para o PLC enviar comandos de movimentos do robô e o tapete rolante através do Arduíno;
- Dimensionar um tapete rolante nas vertentes mecânicas, acionamento eletromecânico e controlo;
- Acionar e controlar um motor DC em malha aberta para aplicar no tapete rolante;
- Desenvolver um sistema a supervisão usando WebVisit da Phoenix Contact.

## 1.2 Estrutura

A dissertação encontra-se dividida da seguinte forma:

- No capítulo 1 faz-se uma pequena introdução e apresentam-se os principais objetivos do trabalho desenvolvido;
- No capítulo 2 descrevem-se os conceitos, tipos, formas de controlo e aplicações dos tapetes rolantes e manipuladores industriais existentes atualmente;
- No capítulo 3 descrevem-se de forma detalhada os equipamentos utilizados para a realização deste trabalho bem como esquemas elétricos, *hardware* e *software*;
- No capítulo 4 apresentam-se os resultados experimentais;
- No capítulo 5 são discutidas as conclusões da dissertação e sugeridos desenvolvimentos futuros.

## ***Capítulo 2***

### ***Estado da arte***

**Resumo:** O presente capítulo apresenta os conceitos sobre os tapetes rolantes, tipos existentes, forma de controlo e aplicação. De igual forma é apresentado também os conceitos gerais dos manipuladores industriais e uma visão geral sobre o protocolo de comunicação RS232.



## 2 Estado da Arte

O estado da arte é uma das partes mais importantes de qualquer trabalho técnico ou científico que se pretende desenvolver, isto porque se trata de uma pesquisa, mais ou menos exaustiva, sobre outros trabalhos relevantes já realizados em áreas semelhantes de modo a evidenciar o que já foi descoberto ou desenvolvido sobre o tema em questão, evitando assim que se dedique muito tempo com investigações e descobertas desnecessárias. Mas, para além disso, auxilia também na melhoria e desenvolvimento de novos conceitos e soluções.

### 2.1 Evolução da Automação e Robótica

Nas últimas décadas tem-se assistido a um crescente interesse pelas áreas da automação industrial e da robótica, motivado, nomeadamente, por preocupações relacionadas com o aumento da produtividade, redução de horários e segurança no trabalho. Esse interesse tem levado diversas entidades públicas e privadas, tais como universidades, agências governamentais e empresas, a efetuar investigação, desenvolvimento e aplicações nessas áreas. O início da era da automação industrial remonta ao século XVIII, numa altura em que as máquinas, apesar de forma muito rudimentar, começavam a fazer parte do processo produtivo das indústrias. [1]

Muitos anos mais tarde, o desenvolvimento das técnicas de produção veio criar novas necessidades que são somente possíveis de satisfazer com o uso das máquinas programáveis e flexíveis, dando origem aos primeiros robôs industriais nos anos 50 do século passado.

Foi no ano de 1956 que, com base na experiência então existente no campo dos manipuladores comandados e das máquinas ou ainda das ferramentas de comando numérico, que os inventores George Devol e Joseph F. Engelberger, construíram o primeiro robô industrial. A partir dessa data a robótica tem vindo a afirmar-se como uma ciência autónoma, de carácter multidisciplinar, penetrando em áreas tradicionalmente ligadas à engenharia eletrotécnica e mecânica, entre outras, revelando uma importância crescente em áreas tão distintas como a exploração espacial, a exploração subaquática, a medicina ou em muitas áreas da indústria. É na indústria que tem sido investido o maior esforço, sendo a indústria automóvel um grande exemplo disso, especialmente nos robôs de pintura e de soldadura que fazem hoje parte integrante da sua “força laboral”. [1]

## 2.2 Robótica Industrial

A Robótica é uma ciência multidisciplinar que conjuga áreas tão vastas como a matemática, cinemática, dinâmica, lógica, mecânica, eletrônica, informática, etc. Embora tenha origem na ficção científica do século passado, só na segunda metade do nosso século é que através da concretização de vários projetos, a robótica se estabeleceu como uma tecnologia suportada em manipuladores mecânicos altamente automatizados, a Robótica Industrial. [2]

A definição 'oficial' de um *Robô Industrial* é fornecida pela Associação das Indústrias de Robótica (*Robotics Industries Association -RIA*): "*Robô industrial é um manipulador reprogramável e multifuncional projetado para mover materiais, peças, ferramentas, ou equipamentos específicos com movimentos variados e programados para a execução de tarefas diversificadas.*"

A Automação Industrial pode ser dividida em três grandes classes:

- Automação fixa -para o fabrico de um grande volume de produtos com ciclos de vida alargados. [2]
- Automação programável para o fabrico de pequenos volumes de vários produtos com ciclos de vida limitados. [2]
- Automação flexível, situa-se entre os dois tipos acima descritos. Foi levada à prática nas últimas duas décadas com sistemas que são conhecidos por: Sistemas de Produção Flexível (*FMS -Flexible Manufacturing Systems*) e Sistemas Integrados de Gestão e Produção (*CIM -Computer Integrated Manufacturing*). [2]

### 2.2.1 Objetivo da Robótica Industrial

A produção industrial dos nossos dias está sujeita cada vez mais a regras de mercado que encurtam o ciclo de vida dos produtos e obrigam a aumentar a sua variedade. A automação fixa tenderá a ser substituída pela automação flexível e, portanto, a Robótica Industrial substituirá a Maquinaria de Automação fixa, as Máquinas Ferramentas de Controlo Numérico e os Manipuladores Sequenciais, num número crescente de aplicações industriais. Os objetivos da Robótica Industrial podem ser sintetizados nos seguintes pontos fundamentais:

1. Aumentar a produtividade, conseguido pelo aumento da velocidade de trabalho dos robôs, diminuindo o tempo utilizado pelos manipuladores, aumenta-se o rendimento global da linha de produção; [2]
2. Potenciar a flexibilidade e a adaptação a séries de produção curtas; [2]
3. Aumentar o rendimento de outras máquinas e ferramentas integradas no sistema de

- produção em conjunto com os robôs; [2]
4. Conseguir amortizações de investimento rápidas, como consequência da substituição da mão-de-obra, melhor utilização e duração de ferramentas, menores desperdícios de matérias-primas, redução de avarias, aumento de utilização diária e anual dos equipamentos de produção e redução da manutenção; [2]
  5. Melhoria da qualidade dos produtos fabricados, devido à precisão de movimentos dos robôs e a possibilidade de instrumentalizar o controlo de qualidade; [2]
  6. Diminuir os stocks de matérias-primas e de produtos acabados bem como dos seus prazos de entrega; [2]
  7. Realizar trabalhos em condições de perigo e insalubridade, temperaturas elevadas, contaminação, gases tóxicos e materiais inflamáveis, explosivos ou radioativos. [2]

### **2.2.2 Classificação dos Robôs**

Os robôs podem ser classificados em cinco tipos:

- Manipuladores-manuais - quando controlados diretamente por um operador; de sequência fixa, quando se repete a uma cadência fixa o conjunto de movimentos previamente preparado e; de sequência variável, quando são alteráveis alguns ciclos da sequência de movimentos; [2]
- *Robôs* de repetição ou aprendizagem - manipuladores que se limitam a repetir uma sequência de movimentos executada previamente por um operador, diretamente sobre o manipulador ou fazendo uso de um dispositivo auxiliar: pistola de programação com pulsadores ou teclas, *joystics* ou modelo reduzido do manipulador; [2]
- *Robôs* controlados por computador - manipuladores ou outros sistemas mecânicos multifuncionais controlados por computador, cuja sequência de movimentos não necessita de ser preparada através de movimentação dos elementos da máquina, mas sim controlada pelo computador por utilização de uma linguagem de programação específica, composta por instruções adaptadas ao robô, sendo a sequência de movimentos obtida pela execução de um programa previamente elaborado; [2]
- *Robôs* inteligentes - *robôs* controlados por computadores, mas que interagem com o ambiente que os rodeia através de sensores podendo tomar decisões (auto-programáveis) em tempo real. Sobre eles incide atualmente o maior esforço de

investigação, por aplicação de sistemas de voz sintética, visão artificial e inteligência artificial. No tempo, a sua aplicação torna-se técnica e economicamente viável para graus crescentes de inteligência; [2]

- *Micro-robôs- robôs* destinados a fins educacionais, de diversão e de investigação, comercializados a preços acessíveis apresentam uma estrutura e funcionamento similares aos *robôs* industriais. [2]

### 2.2.3 Anatomia dos *Robôs*

A anatomia do *robô* define a configuração física do corpo, braço e punho. A maioria dos robôs industriais são montados numa base fixada ao piso, o corpo está ligado à base e o braço ao corpo, na extremidade do braço está o punho, ligado ao punho está um manipulador (órgão terminal). Os movimentos relativos entre estes componentes são assegurados por juntas que podem ter movimentos rotativos ou deslizantes. [2]

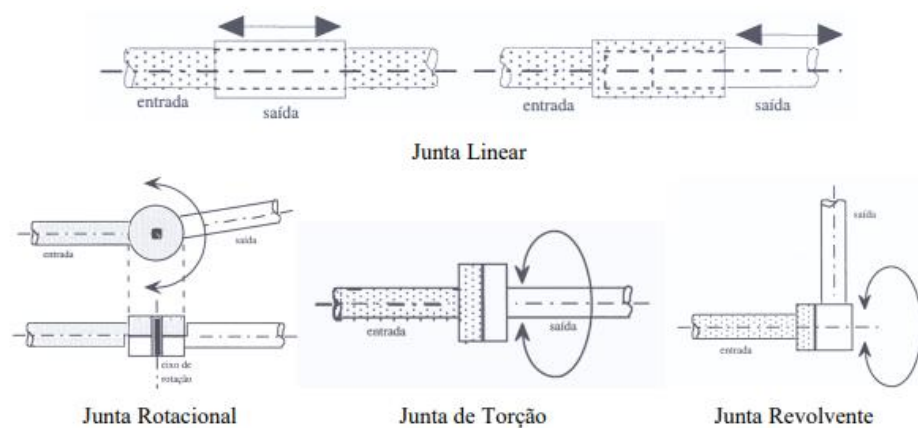


Figura 1 - Anatomia dos *robôs* [2]

Os robôs industriais apresentam-se numa grande variedade de formas, configurações e tamanhos, mas podem na maioria das vezes ser classificados num dos quatro tipos de configurações base, como se pode observar na Figura 2. [2]

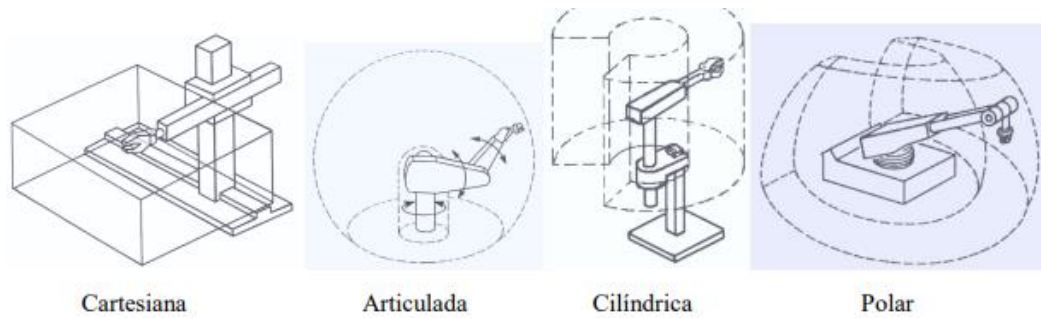


Figura 2 - Configurações base dos *robôs* [2]

O volume de trabalho é o espaço dentro do qual o *robô* se pode movimentar com a extremidade do punho. A convenção de usar a extremidade do punho para definir o volume de trabalho procura evitar a confusão que se estabeleceria na sua definição com órgãos terminais de tamanhos diferentes que poderiam ser ligados ao punho. Mais se justifica a sua não inclusão por serem elementos opcionais ao *robô* base. [2]

Os quatro tipos de configurações base são então:

- Configuração em coordenadas cartesianas, robô XYZ ou *robô* retilíneo - Os seus movimentos assentam em três juntas lineares. Deslocando-se as três réguas umas em relação às outras o robô define um volume de trabalho com a forma de um paralelepípedo. Quando estes *robôs* apresentam grandes dimensões são ainda designados por robôs de pórtico: [2]
- Configuração angular ou robô articulado - Semelhante a um braço humano, é composto por dois elos retilíneos correspondentes ao antebraço e ao braço humanos montados num pedestal vertical. Estes elos estão ligados por juntas rotativas correspondentes ao ombro e ao cotovelo. Um punho está ligado à extremidade do antebraço com as respetivas juntas: [2]
- Configuração cilíndrica - É constituído por uma coluna vertical e um suporte que se pode mover para cima e para baixo ao longo da coluna. O braço ligado ao suporte pode ser movido radialmente em relação à coluna, através de uma junta linear. O volume de trabalho alcançado é cilíndrico; [2]
- Configuração polar ou de coordenadas esféricas - Usa um braço telescópico (elo linear) que se pode levantar ou baixar através de uma junta rotativa, montada numa base também rotativa. O volume de trabalho alcançado é, pois, esférico. Os movimentos dos robôs são normalmente divididos em duas categorias:
  - Movimentos do braço e do corpo

- Movimentos do punho/mão.

Conforme se pode constatar na descrição da anatomia dos robôs aos movimentos do braço e do corpo estão associadas três juntas a que correspondem três GRAUS DE LIBERDADE. Aos movimentos do punho/mão podem estar associadas duas ou três juntas. A Figura 3 ilustra os graus de liberdade correspondentes a uma configuração típica de um punho/mão com três juntas. Sendo:

Eixo de "Roll" (Rotacional) -Rotação do punho em torno do eixo do braço do robô:

Eixo de "Pitch" (Inclinação) -Orientação vertical ou inclinação do punho, se o eixo rotacional estiver na posição central corresponde à rotação do punho para cima e para baixo:

Eixo de "Yaw" (Orientação Horizontal) -Se o eixo rotacional estiver na sua posição central corresponde à rotação do punho para a esquerda e para a direita. [2]

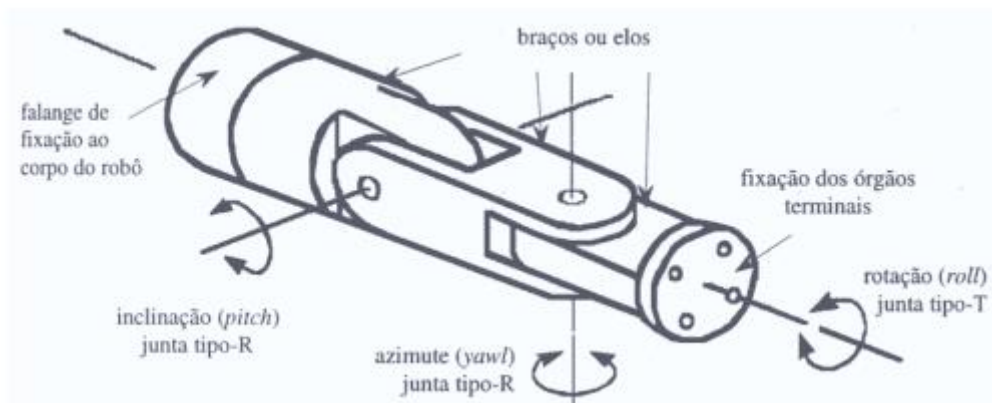


Figura 3 - Graus de Liberdades de um Robô [2]

### 2.3 Breve Historia dos Tapetes Rolantes

Nas indústrias, a movimentação e a agilidade são essenciais para que os processos de finalização de produtos sejam concluídos em grande escala. Contudo, nem sempre foi assim. Nos períodos que antecederam a revolução industrial, o trabalho manual de transporte e produção era o mais comum, com o emprego esporádico de máquinas que realizavam atividades simples. Logo, os trabalhadores dividiam-se em diversas etapas para tentar fazer com que o processo de produção fosse o mais direto possível e esses serviços incluíam serviços de montagem, limpeza e os de transporte de matéria-prima. [3]

A esteira industrial foi uma das ferramentas responsáveis por facilitar o transporte de materiais. Como já dito, os trabalhadores empenhavam-se em diversas tarefas a fim de realizar em menos tempo possível os processos de fabricação. [3]

Com o estouro da Revolução Industrial no fim do século XVIII, a indústria passou por grandes mudanças em sua estrutura devido à entrada de novas tecnologias que proporcionavam aquela agilidade e capacidade de transporte tão desejada. Todavia, o processo de produção sofreu um salto gigantesco quando os trabalhadores começaram a ser substituídos pelas máquinas. Uma das tecnologias, propriamente dita, foi o desenvolvimento da esteira transportadora. Essas esteiras tiveram lugar em quase todos os setores da indústria. [3]

No ano de 1892 as primeiras esteiras para transporte de minérios começaram a existir, e a partir deste momento ganharam grande importância na extração mineira, garantindo que os minérios fossem levados de um lado para o outro sem a necessidade de esforço bruto, o que também implicava num aumento de mão de obra para aumentar a extração. [3]

A presença das esteiras transportadoras revolucionou, de algum modo todo o sector da extração mineira e nos anos seguintes foram surgindo mais tecnologias e novos materiais para serem usados como forma de acionamento das esteiras, nomeadamente as correias de borracha. [3]

Contudo, com o início da Segunda Guerra Mundial, muitos países viram-se com deficiência de borracha e algodão, que eram materiais importantes para a fabricação de correias de transmissão, todavia, estudos de manipulação química foram responsáveis pela criação de novas fibras sintética que viriam a ser usadas por fabricantes de correias em substituição de alguns componentes de borracha e algodão. [3]

A partir de 1900, o mercado também deu espaço para a criação da esteira elétrica transportadora em aço. Com o surgimento do aço inoxidável, este tipo de esteira ganhou destaque no transporte de alimentos, como a cana-de-açúcar e outros produtos alimentares. Anos mais tarde as esteiras transportadoras foram sendo inseridas ao plano de linha de montagem da indústria automóvel que, até então, eram somente uma ideia no papel, permitindo assim que carros se deslocassem por um sistema de rolamento ou roletes enquanto eram montados. Graças aos novos tipos de materiais empregados, as esteiras deixaram de ser somente uma peça para o funcionamento de indústrias e mineração, como também foram inseridas em outros âmbitos, como é o que acontece com as esteiras de elevação de carga e descarga em muitos serviços. [3]

Uma esteira transportadora industrial ou simplesmente tapete rolante é um tipo de equipamento que oferece grandes benefícios às empresas que necessitam de processos automatizados dentro da sua estrutura. Diversos segmentos utilizam esteira transportadora

industrial para melhorar a sua produtividade, uma vez que é um equipamento que permite o transporte de materiais com elevada rapidez. Ela também auxilia na economia de custos para a empresa, já que sua utilização pode resultar na diminuição da mão de obra. [3]

A esteira transportadora industrial (Figura 4) é essencial em sistemas de linha de montagem, por exemplo. Ela pode transportar partes de produtos, ou o produto finalizado em si. Além de auxiliar no transporte, ela também auxilia na separação de diversos tipos de material, representando um importante papel em processos de reciclagem de peças e outros materiais recicláveis dentro do ambiente industrial. [3]



Figura 4 - Esteira Transportadora Industrial [3]

O funcionamento de uma esteira transportadora industrial é basicamente o mesmo de uma esteira comum, como por exemplo as encontradas em caixas de supermercados e esteiras de transporte de pessoas que geralmente podem ser encontradas nos centros comerciais ou em grandes empresas. Contam com no mínimo duas polias, que ao entrarem em movimento são responsáveis por acionar a esteira em si. Na esteira são depositados os materiais e produtos que se deseja transportar. [3]

### **2.3.1 Constituição dos tapetes rolantes**

No passado usavam-se como tapete rolantes várias polias envoltas por uma liga de material diferente. Essas polias eram acionadas de tal forma que a liga que as cobrisse se movimentasse e tudo que estivesse em sua superfície também. [3]

Atualmente, o padrão de funcionamento não é assim tão diferente do passado, com a diferença que agora se utilizam materiais mais práticos, flexíveis e duradouros para que as esteiras aguentem mais peso e calor. [3]

Os mancais de rolamentos são parte das peças empregadas na produção de esteiras. Eles servem como apoio e eixo. São compostos de ferro, ou outro material mais resistente, e revestidos por resinas para que não sobreaqueçam e derretam devido a fricção. Eles são constituídos por dois anéis e dentro deles é colocada uma peça rolante. [3]

Os roletes de carga trata-se também de uma peça fundamental para o funcionamento das esteiras. Eles garantem sustentabilidade e velocidade, quando combinados com mancais de rolamento. Os materiais utilizados na sua produção implicam que a esteira será capaz de aguentar uma quantidade maior de peso. [3]

A combinação dessas duas peças, com a adição da liga de fibra, permite estruturar uma esteira simples. Outros fundamentos, como a parte elétrica, também fazem parte de algumas estruturas de esteiras mais modernas. [3]

### **2.3.2 Aplicações de Sistemas de Tapete Rolante**

Existem atualmente diversas aplicações e soluções muito engenhosas usando sistemas de tapetes rolantes combinados com as mais diversas soluções e tecnologias. Por exemplo, a empresa de Automação Industrial OMRON desenvolveu um sistema de embalagem/equipamento de carregamento robótico orientado por sistema de visão (Figura 5). [4]



Figura 5 - Sistema de Embaladora OMRON [4]

A título de exemplo, descreve-se o funcionamento desta solução seguidamente.

➤ Descrição da máquina

A máquina é composta por um *robô* montado numa estrutura que se sobrepõe a dois tapetes rolantes que funcionam em paralelo. Um dos tapetes transporta os produtos e o outro as embalagens. O *robô* recolhe os produtos de forma aleatória e coloca-os nas embalagens em movimento. A localização imediata dos produtos em movimento é determinada pelo sistema de visão, através das imagens adquiridas pela câmara estacionária. Por seu lado, o sensor de registo acompanha o posicionamento das embalagens. Este controlo dos dois tapetes permite que o *robô* recolha e transfira produtos de um tapete para o outro e proceda ao seu embalamento. [4]

➤ Sistema de controlo robótico

Os sistemas robóticos podem ser utilizados de forma eficiente como módulos de alimentação principais ou como soluções de carregamento secundárias ou de fim de linha. Com a integração das bibliotecas de objetos de *software* robótico nos controladores, os *robôs* são agora de implementação e manutenção fáceis, o que representa um verdadeiro estímulo à sua utilização. [4]

➤ Grupos de funções principais da máquina

**Tapete rolante de produtos:** um tapete rolante controlado por sistema servo transporta os produtos em intervalos aleatórios. [4]

**Deteção e registo do produto:** uma câmara e um codificador registam a posição dos produtos e guardam os valores dinamicamente, numa memória intermédia de dados *FIFO*, antes do seu envio para o *robô*. [4]

**Tapete rolante de embalagens:** são fornecidas duas correias laterais controladas pelo sistema servo, com engates de indexação para as partes frontal e traseira das embalagens. [4]

**Deteção e registo das embalagens:** um codificador e uma célula de registo registam as posições vazias na embalagem e guardam os valores dinamicamente, numa memória intermédia de dados *FIFO*, antes do seu envio para o *robô*. [4]

**Sistema de transferência robotizado:** o *robô* Delta-3 recolhe os produtos em movimento e coloca-os em recipientes em movimento. [4]

## 2.4 Motores DC

Um motor DC ou simplesmente motor de corrente contínua, é uma máquina capaz de converter energia elétrica em energia mecânica, sendo por este motivo usado em várias aplicações de tração das partes moveis de robôs, manipuladores industriais, tapetes rolantes e em diversos dispositivos mecatrónicos. Justifica-se a sua abordagem nesta fase uma vez que foi o tipo de motor usado no acionamento do tapete rolante.

### 2.4.1 Constituição, Princípio de Funcionamento

Os motores DC são constituídos por duas partes essenciais, uma parte móvel chamada de Rotor ou Armadura e uma parte fixa a que se dá o nome de Estator ou Campo.

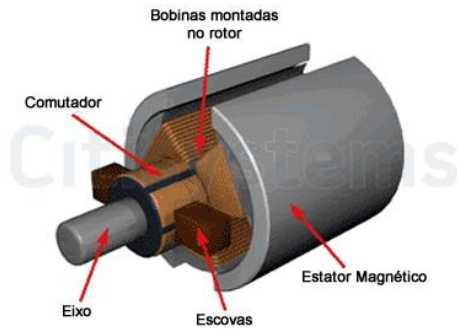


Figura 6 - Partes Constituintes de um Motor DC [5]

O Rotor é a parte girante, montado sobre o eixo da máquina, construído de um material ferromagnético envolto em um enrolamento chamado de enrolamento de armadura, este enrolamento liga ao anel comutador. [6]

O Estator como indica o nome é a parte estática da máquina, montada em volta do rotor, de forma que o mesmo possa girar internamente, também constituído de material ferromagnético envolto em um enrolamento chamado de enrolamento de campo, que tem a função de produzir um campo magnético fixo para interagir com o campo da armadura. Dependendo do tipo de motor DC este campo magnético pode ser produzido por ímãs permanentes. [6]

Pode-se então dizer, que existem vários tipos de motores CC de acordo ao tipo de excitação, como se pode observar na figura abaixo. [6]

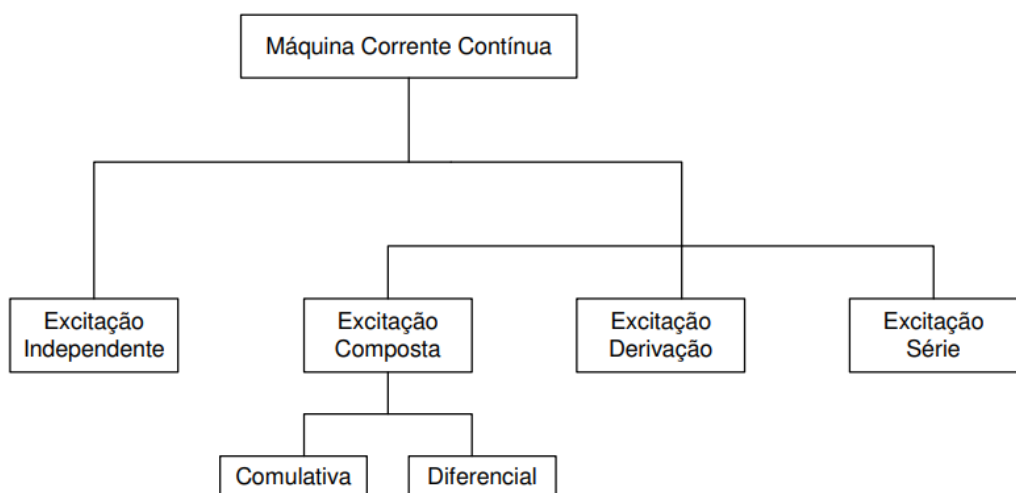


Figura 7 - Tipos de máquina de corrente contínua [6]

Para se compreender o princípio base de funcionamento de um motor DC de forma

simplificada deve-se observar a Figura 8.

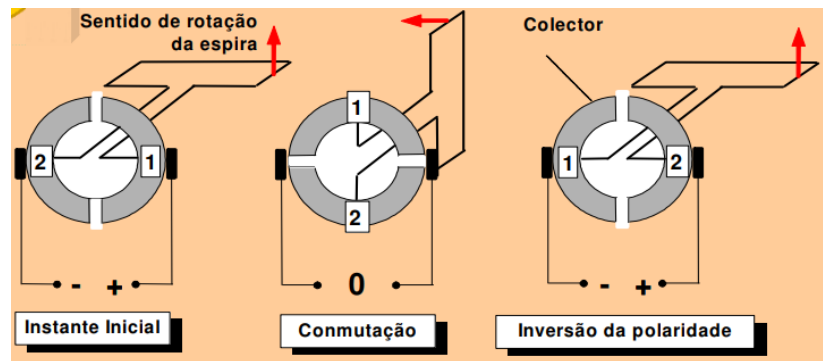


Figura 8 - Ilustração do princípio de funcionamento de um motor DC [6]

De uma forma resumida, se invertermos os terminais de alimentação do rotor, o sentido da corrente na bobina também inverterá e o motor irá girar em sentido contrário. Ou seja, o sentido de rotação do motor depende completamente do sentido da corrente imposta por sua vez pela tensão aplicada.

### 2.4.2 Controlo de Motores DC

Uma vez que alterando a tensão aplicada aos terminais do rotor através das escovas varia-se a velocidade e alternando o sentido da corrente altera-se a direção, pode-se dizer que são duas informações fundamentais para o desenvolvimento de um controlador de velocidade e direção.

A solução adotada baseia-se no princípio de funcionamento de um conversor DC/DC de quatro quadrantes (Figura 9) ou *chopper* de 4 quadrantes, que na qual com o funcionamento dos respetivos quadrantes permite o controlo dos parâmetros do motor, tais como regulação da velocidade, arranque, travagem regenerativa e inversão no sentido de rotação. Ou muitas vezes chamados de ponte H. O termo Ponte H é usado devido a forma da letra H que o circuito de potência possui. [6]

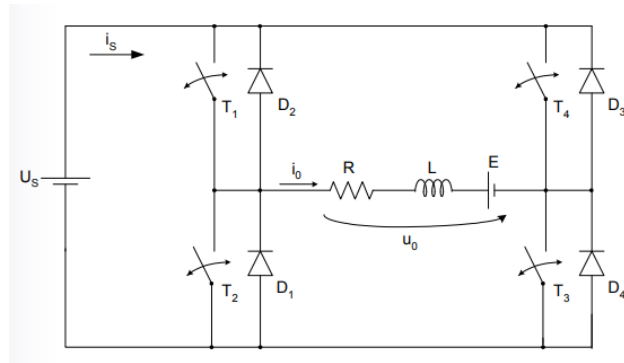


Figura 9 - Conversor CC/CC Quatro Quadrantes [6]

Este conversor, como se pode observar é constituído por quatro comutadores estáticos de potência (semicondutores de potência), cada um com dois estados possíveis (Ligado ou Desligado). Em termos de funcionamento estes comutadores devem apenas estar desligados, ou caso estejam ligados devem estar em funcionamento apenas aos pares em diferentes braços da ponte, caso contrário haveria um curto-circuito e os semicondutores se iriam danificar, ou seja, no momento em que estiverem dois ligados os outros dois devem estar desligados. [6]

Com isto pode-se formar sete estados possíveis para este circuito, como se pode observar na Tabela 1.

Tabela 1 - Estados Possíveis da ponte H

Nº Estado	Semicondutores Estáticos de Potência				Estado do Motor / Conversor
	Comutador T2	Comutador T3	Comutador T1	Comutador T4	
1	Desligado	Desligado	Desligado	Desligado	Parado
2	Desligado	Ligado	Ligado	Desligado	Roda Sentido Horário
3	Ligado	Desligado	Desligado	Ligado	Roda Sentido Anti- Horário
4	Ligado	Ligado	Desligado	Desligado	Travagem Regenerativa Sentido Horário
5	Desligado	Desligado	Ligado	Ligado	Travagem Regenerativa

					Sentido Anti-Horário
6	Ligado	Desligado	Ligado	Desligado	Curto-Circuito
7	Desligado	Ligado	Desligado	Ligado	Curto-Circuito

### 2.4.3 Controlo da Velocidade do Motor DC por PWM

Nos pontos anteriores referiu-se os tipos de travagens do motor com o circuito de potência e como se mudava o sentido de rotação, não tendo sido mencionado técnicas de variação da velocidade do motor. Existem duas tecnologias básicas que servem para se fazer o controlo da velocidade do motor, que são o controlo linear e o controlo por PWM (modulação por largura de impulso).

O controlo linear segue um princípio básico, que consiste em variar linearmente a corrente ou tensão aplicada numa carga. Uma forma de representação deste circuito pode ser observada na Figura 10.

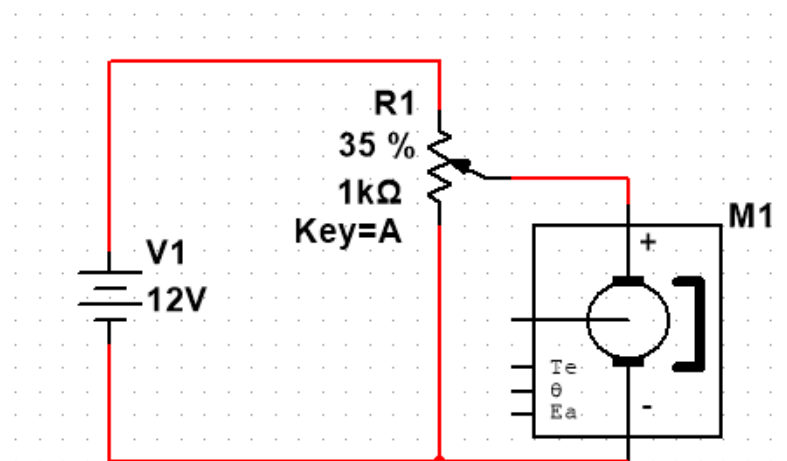


Figura 10 - Circuito de Controlo Linear

Este tipo de controlo possui duas grandes desvantagens pelas quais não foi utilizado. Uma delas dá-se pelo facto de não se conseguir obter um controlo preciso em baixas rotações devido a inercia do motor, a velocidade não acompanha linearmente a tensão de alimentação, possuindo um arranque irregular e de difícil controlo.

O controlo por PWM é muito mais preciso, pois permite um controlo de baixa velocidade com precisão. Este método consiste em ligar e desligar o motor a uma frequência fixa através de semicondutores de potência (o mesmo usado pela ponte H), fazendo com que seja aplicada aos terminais do motor uma tensão média proporcional à relação entre o intervalo de tempo em que o semicondutor está fechado ( $T_{on}$ ), e o período de comutação ( $T$ ). Esta relação é chamada por *Duty Cycle* ( $D$ ). Geralmente este tipo de controlo é normalmente utilizado com recurso a componentes eletrónicos, tais como os TBJS, MOSFETs e IGBTs.

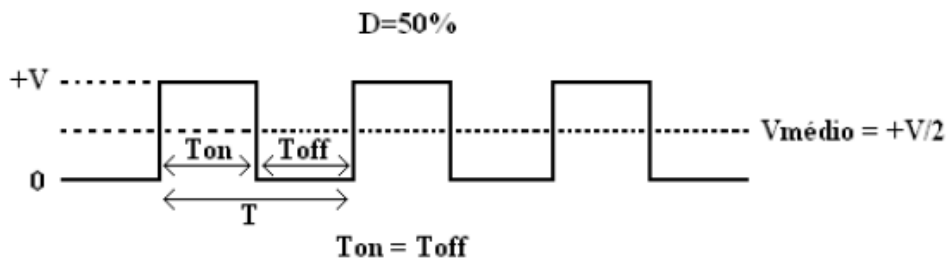


Figura 11 - Sinal PWM

No exemplo da Figura 11 está representado um sinal PWM com um *Duty Cycle* de 50% e pode ser calculado a partir das equações (4) e (5).

$$D = \frac{T_{on}}{T} * 100 \quad (4)$$

$$V_m = V * D \quad (5)$$

Um pequeno exemplo de aplicação pode ser observado na Figura 12, onde o sinal PWM abre e fecha o comutador a uma frequência fixa, resolvendo o problema do controlo a baixas rotações, uma vez que a tensão dos pulsos aplicada ao motor é a tensão nominal, sendo a tensão média proporcional ao *duty cycle*.

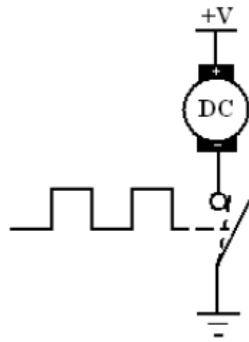


Figura 12 - Aplicação do Sinal PWM

Em mais uma forma de exemplo pode-se observar a Figura 13 com três sinais PWM diferentes, ambos com a mesma frequência, mas com *duty-cycle* diferentes.

A forma de onda A, indica que quando aplicada ao motor faz com que o mesmo rode com uma velocidade muito reduzida, porque o tempo em que o semicondutor fica desligado é muito maior do que o tempo em que está ligado, sendo a tensão média aplicada ao motor muito baixa.

A forma de onda B, indica que o tempo que o semicondutor está ligado com uma tensão nominal é exatamente o mesmo tempo em que está desligado, fazendo com que o valor médio da tensão na carga seja aproximadamente a metade da tensão de alimentação.

A forma de onda C corresponde ao estado em que o motor roda aproximadamente a velocidade nominal (146 RPM no caso do motor escolhido).

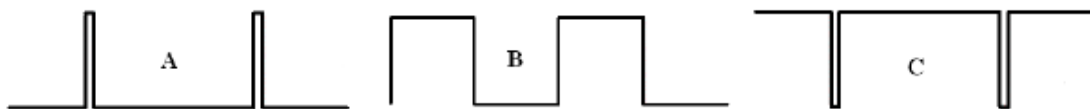


Figura 13 - Sinais PWM com Frequências Idênticas e Diferentes *Duty Cycles*

## 2.5 Microcontrolador Arduíno

Este microcontrolador é o centro do controlo do sistema do tapete rolante. É o responsável por gerar os sinais PWM, controlar o sentido de rotação do motor e gerir as entradas e saídas. Neste trabalho usa-se dois Arduíno, um para o controlo do robô e outro para o controlo do tapete. O Arduíno usado é o UNO (Figura 14).



Figura 14 - Microcontrolador Arduino UNO

### 2.5.1 Origem do Arduino

O Projeto do Arduino nasceu no *Interaction Design Institute* na cidade de Ivrea, comuna italiana da região do Piemonte, província de Turim, em 2005. [7]

O professor do Instituto, *Massimo Banzi* procurava uma forma de possibilitar o uso de tecnologia nos projetos dos estudantes de *design*, com baixo custo e facilidade de aprendizado e utilização. Com a colaboração de *David Cuartielles*, um investigador visitante da Universidade de *Malmö*, na Suécia e *Gianluca Martino*, um engenheiro italiano, foi produzida, no Instituto, um protótipo inicial de 200 placas, que foram rapidamente vendidas. [7]

A nova placa foi chamada Arduino em referência a um bar local *Bar di Re Arduino*, frequentado por membros do corpo docente e alunos do Instituto. O nome do bar homenageia um rei europeu do início do século XI. A popularidade desta placa cresceu rapidamente quando o grande público percebeu a sua fácil utilização e o seu baixo custo. [7]

As primeiras placas da plataforma Arduino tiveram o seu início com a comunicação série e componentes discretos e eram vendidas desmontadas em *kits* ou apenas a placa de circuito impresso. Essas primeiras versões ainda não apresentavam o nome Arduino, porém hoje em dia são conhecidas como *Arduino Serial* e *Arduino Serial v2*. [7]

## 2.5.2 Tipos de Arduíno

O Arduíno é uma plataforma de *hardware* e *software open source*<sup>1</sup>, programável, que permite interagir com o mundo exterior a partir da aquisição de sinais de sensores e interagir controlando circuitos de iluminação, motores e outros atuadores. [7]

O projeto Arduíno é disponibilizado em diversas versões para atender a diversos tipos de aplicações. Existe atualmente no mercado os modelos de arduino *UNO*, *Mini*, *Leonardo*, *Micro*, *Due*, *Nano*, *Mega*, *Yun*, *Gemma* e *LilyPad*, sendo o UNO o usado neste trabalho de dissertação. [7]

## 2.6 Comunicação RS232

A comunicação RS232 define, numa perspectiva muito alargada, as características físicas e funcionais a satisfazer nas interfaces de comunicação digital série ponto-a-ponto prevê o caso de comunicação entre duas estações de dados terminais (DTE) com equipamentos de comunicação (DCE) intercalados para adaptação ao canal de condução (p. ex. modems ligados a linhas telefónicas), esquematizado na Figura 15. Contudo, em muitos casos apenas estão presentes os equipamentos terminais (Figura 16). [8]

Trata-se, pois, de um sistema de comunicação série apenas entre dois participantes, também chamado ponto-a-ponto. Esta “norma” define, numa perspectiva muito alargada, as características físicas e funcionais a satisfazer nas interfaces de comunicação digital série ponto-a-ponto. [8]

---

<sup>1</sup> Código Aberto, ou *open source*, é um modelo de desenvolvimento que disponibiliza todas as informações a todos os usuários.

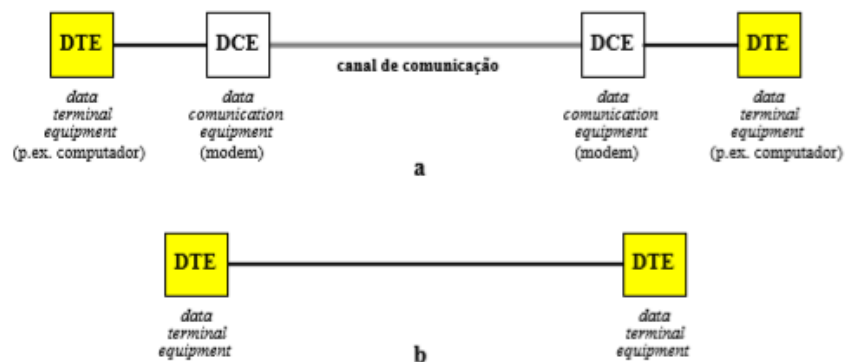


Figura 15 – Esquema de Comunicação Série [8]

A Figura 15-a) representa o conjunto de equipamentos de um sistema de comunicação série previstos pela norma RS-232-C enquanto que a Figura 15-b) representa um caso que intervêm apenas equipamentos terminais(DTE). [8]

A transmissão é feita através de condutores, por sinais eléctricos binários em tensão. O número total de condutores de ligação (circuitos na terminologia da norma), previstos no caso mais geral, é superior a vinte. Esse vasto conjunto de circuitos de comando e de sinalização de estado destina-se a cobrir uma gama muito alargada de situações incluindo os detalhes de estabelecimento de comunicações telefónicas. [8]

Na maioria dos casos, porém, apenas é usada um pequeno subconjunto desses circuitos, que se resume aos indicados na Figura 16. Existem três tipos de circuitos, a saber:

- Circuitos de dados (p. ex. TXD e RXD);
- Circuitos de controlo (para comando e aviso de estado – p. ex. RTS, CTS, DSR, DTR);
- Circuitos de cadência para a transmissão síncrona. [8]

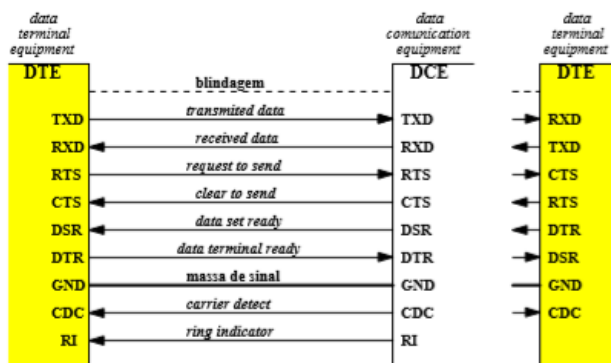


Figura 16 - Ligações Previstas na norma RS-232-C [8]

A norma estipula correspondência entre o nível dos sinais eletricos e o valor logico de acordo a Tabela 2.

Tabela 2 - Correspondência Entre o sinal elétrico e valor lógico

Valor Lógico	Nível de Tensão	
	Sinais de dados TXD e RXD	Sinais de controlo e de cadência
0	+3 a +15V	-3 a -15V
1	-3 a -15V	+3 a +15V

Os dados são enviados pela estação DTE através de TXD. Os dados que são recebidos nesta estação chegam através de RXD. Os dados só podem ser transmitidos se as ligações de controlo RTS, CTS, DSR e DTR estiverem todas no estado lógico 1. Em repouso TXD deve permanecer ao nível lógico '1'. [8]

DTR é passado a '1' para solicitar a ligação do equipamento de comunicação ao canal de comunicação. DSR é posto a '1' pelo equipamento de comunicação para indicar que está ligado ao canal de comunicação. [8]

RTS é posto a '1' pela estação terminal para preparar o equipamento de comunicação para o modo 'transmissão'. Em resposta, CTS é posto a '1' pelo equipamento de comunicação quando estiver preparado para iniciar a transmissão e estiver a assinalar também CDC a '1' (sinal de linha). [8]

Os dispositivos previstos para o estabelecimento da ligação física são as fichas de 9 e de 25 pinos, cuja correspondência com os circuitos de interligação é a indicada na Tabela 3. [8]

Tabela 3 - Correspondência Entre as Fichas e os Circuitos de Ligação Série

Designação	Nº do pino na ficha	
	Ficha D 25	Ficha D 9
Blindagem	1	-
TXD	2	3
RXD	3	2
RTS	4	7
CTS	5	8
DSR	6	6
GND	7	5
DCD	8	1
DTR	20	4

A Figura 17 apresenta a descrição genérica dos sinais transmitidos em modo assíncrono, por cada palavra, em comunicação série.

A transmissão da palavra começa sempre com um start bit (com valor '0'); depois seguem-se os bits de dados (em número a definir, p.ex. 7 ou 8); pode existir ou não um bit de paridade para controlo de erros de transmissão; existe sempre pelo menos um stop bit (com valor '1'). [8]

Atente-se na correspondência "simétrica" entre os níveis lógicos das linhas de dados (RXD e TXD) e os níveis de tensão e ainda no facto de o estado de repouso das linhas de dados ser definido ao nível lógico '1' (i.e. tensão  $-V_s$ ). [8]

A mesma figura mostra ainda exemplos de sinais correspondentes a um mesmo carácter com diferentes parametrizações de transmissão. [8]

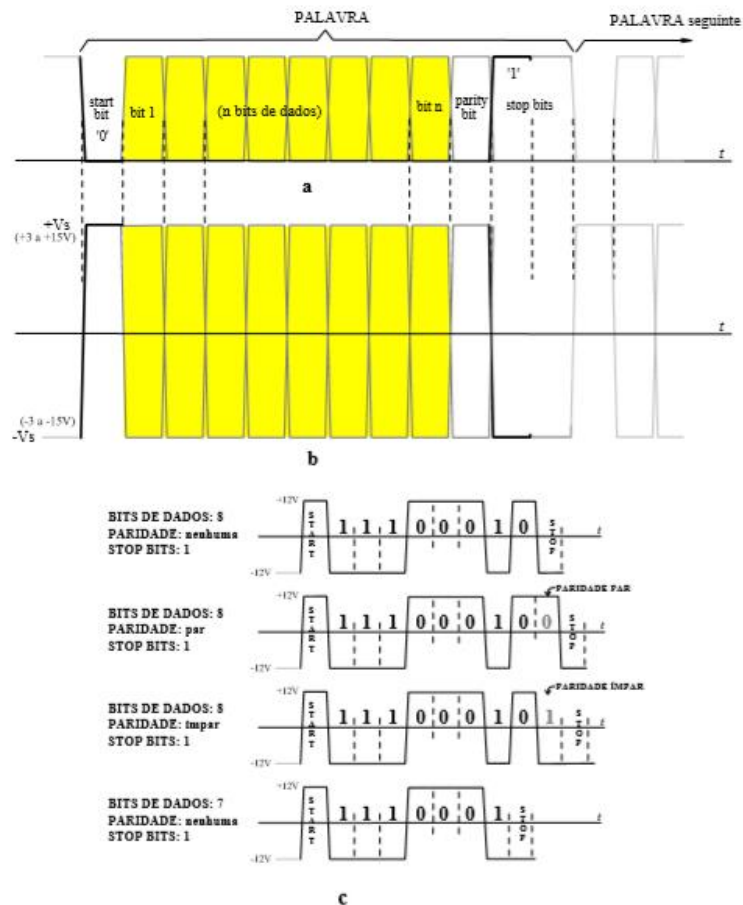


Figura 17 - Sequência Genérica de bits na transmissão série assíncrona de uma palavra [8]

## ***Capítulo 3***

### ***Solução Desenvolvida***

**Resumo:** O presente capítulo apresenta com detalhes a solução de controlo e supervisão implementada para a automatização de um Braço Robótico Combinado com um Tapete Rolante.



### 3 Solução desenvolvida

Nesta secção apresenta-se a solução desenvolvida de braço robótico manipulador apoiado por um tapete rolante. Apresenta-se a descrição de todo o hardware desenvolvido bem como a programação de todos os elementos que conduziram ao resultado final de acordo com os objetivos propostos.

Este trabalho está dividido em quatro grandes partes. A primeira parte diz respeito a um Arduíno que é responsável pelo controlo do tapete rolante, a sua velocidade e respetivos sentidos de direção, bem como as ordens de arranque e paragem. A segunda parte diz respeito a outro Arduíno que é responsável pelo controlo dos movimentos do robô manipulador de 6 eixos usado. A terceira parte que diz respeito a aplicação da supervisão, que na qual permite o utilizador interagir com o sistema, sendo capaz de mover junta a junta do robô em modo manual, gravar até no máximo 6 pontos para o robô, parar, arrancar e mudar o sentido de rotação do tapete. A quarta e última parte, sendo a chave do trabalho, diz respeito a integração ou comunicação entre o PLC e Arduíno, pelo que o sistema geral é controlado pelo PLC.

Em termos de material utilizado, a solução proposta é basicamente constituída pelo seguinte material:

- 1x Robô Manipulador;
- 1x Tapete Rolante;
- 1x Motor DC;
- 1x Driver de Controlo do Motor DC;
- 1x PLC ILC 131 Phoenix Contact;
- 2x Arduíno UNO;
- 1x Módulo conversor RS232/TTL;
- 1x Divisor de tensão;
- 1x LCD.

Apresenta-se nas secções seguintes mais detalhes sobre todos os elementos utilizados.

#### 3.1 Descrição do Robô Manipulador usado

A escolha do robô manipulador dependeu do volume de trabalho, porque o tapete rolante é pequeno e o espaço de trabalho é reduzido. Selecionou-se um robô com seis graus de

liberdade (Figura 18) tendo como elemento terminal uma garra.



Figura 18 - Robô Manipulador com Seis Graus de Liberdade [9]

Trata-se de um braço robótico totalmente operacional, controlado por uma placa Arduino. É fornecido em formato de *kit* para que seja possível montá-lo de acordo com as necessidades de cada utilizador. Pode também ser usado de várias maneiras e se adapta com bastante facilidade. [9]

A Figura 19 apresenta uma imagem real do robô, montado e em fase de testes para conhecer a sua posição no espaço em relação aos comandos angulares das suas juntas.

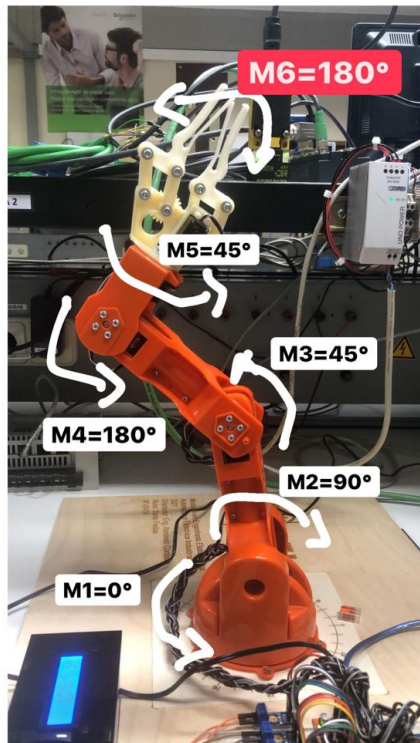


Figura 19 - Ilustração robô Braccio

**Descrevem-se seguidamente as características deste robô:**

- Distância máxima de operação: 80cm
- Altura máxima: 52cm
- Largura da base: 14cm
- Largura do elemento terminal: 90mm
- Comprimento do cabo: 40cm
- Capacidade máxima de carga: 150g
- Peso máximo na base de configuração: 400g
- Servomotores: 2x SR 311, 4x SR 431
- Peso total: 0.792Kg
- Tensão de trabalho: 5VDC
- Consumo: 20W
- Corrente máxima: 1,1 A nas juntas M1 a M4 e 750mA nas juntas M5 e M6

Quanto aos servomotores SR 311 presente na garra e no punho e o SR 431 presente nas restantes juntas, vale a pena realçar que possuem um motor eletromecânico, que apresenta

movimento proporcional a um comando, como dispositivos de controlo em cadeia fechada, ou seja: recebem um sinal de controlo; que verifica a posição atual para controlar o seu movimento indo para a posição desejada com velocidade monitorada externamente sob feedback de um dispositivo denominado taquímetro. Está também disponível um sensor de efeito Hall, um *encoder*, um *resolver*, ou um *tachsin*, dependendo do modelo do servomotor e aplicação em causa. Em comparação com os motores DC que giram indefinidamente, o eixo dos servo motores possuem a liberdade de apenas cerca de 180º graus (360º em alguns modelos) mas são bastante precisos quanto à sua posição. [10]

#### **Característica do servo SR 311:**

- Sinal de controlo: PWM analógico
- Binário: 4.8v(0,30 N.m) e 6.0v(0,38 N.m)
- Peso: 27g
- Dimensões: 31,3x16,5x28,6 mm
- Velocidade: 4.8v (1.14 s/60º) e 6.0v(0.12 s/60º)
- Faixa de rotação: 180

#### **Característica do servo SR 431:**

- Sinal de controlo: PWM analógico
- Binário: 4.8v(1,12 N.m) e 6.0v(1,4 N.m)
- Peso: 62g
- Dimensões: 42,0x20,5x39,5 mm
- Velocidade: 4.8v (0.20 s/60º) e 6.0v(0.18 s/60º)
- Faixa de rotação: 180º

Em termos de controlo, este robô vem com um *driver* de potência, uma vez que a fonte do Arduíno não é suficiente para alimentar todos os *servodrives*, tal como mostra a Figura 20.

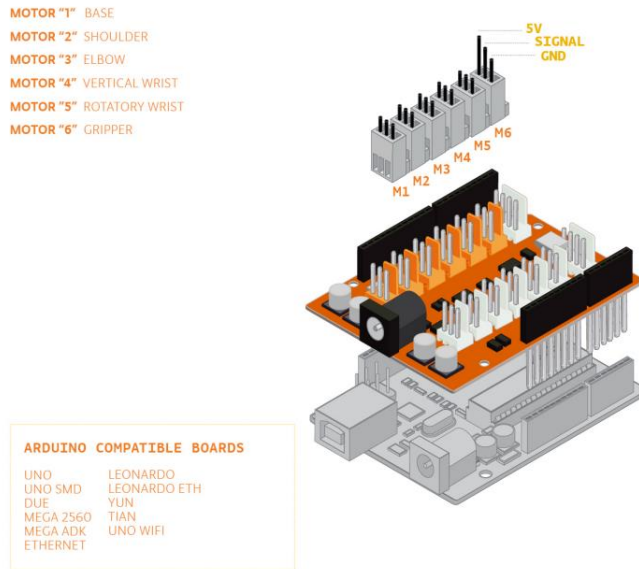


Figura 20 - Arduíno Braccio Shield

A Figura 21 ilustra o esquema de ligação de cada servo, onde o sinal de PWM será enviado pelo Arduíno.

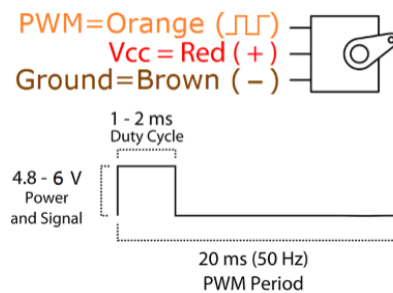


Figura 21 - Esquema de Ligação Servodrive

De forma resumida, os conectores M1 a M6 correspondem ao controlo individual de cada servo de I0 a I5 correspondem a entradas analógicas que podem ser usadas para controlar o robô a partir de inputs externos. A Tabela 4 representa essa nomenclatura.

Tabela 4 - Esquema de Ligação Shield Braccio

Nome do Conector na <i>Shield</i>	Pino do Arduíno
M1	11

Nome do Conector na <i>Shield</i>	Pino do Arduino
M2	10
M3	9
M4	6
M5	5
M6	3
I0	A0 (também conhecido como 14)
I1	A1 (15)
I2	A2 (16)
I3	A3 (17)
I4	A4 (18)
I5	A5 (19)
TWI	SCL, SDA
Serial	RX0, TX0

Abaixo partilham-se algumas imagens durante o processo de adaptação e instrução dos pontos de referência do robô.

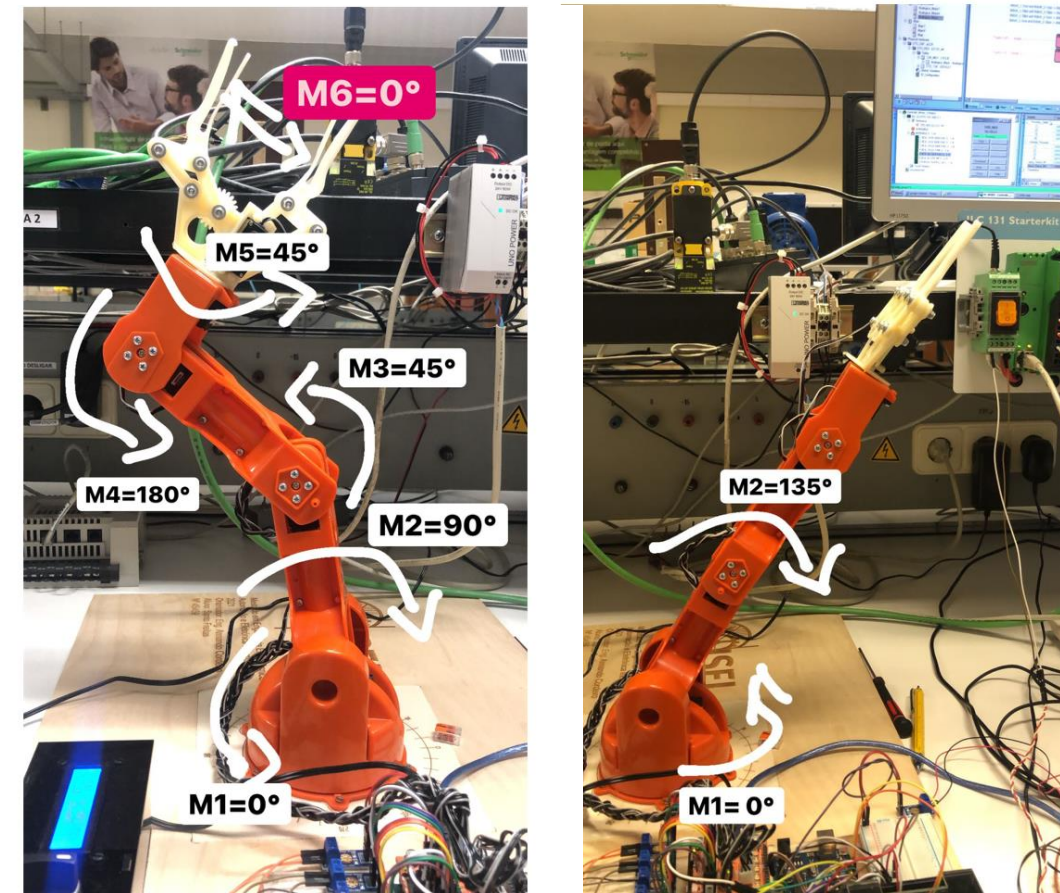


Figura 22 - Braccio Teach 1

### 3.2 Descrição do Tapete Rolante Utilizado

O tapete usado neste trabalho é constituído por duas polias, uma correia, uma esteira composta por tubos plásticos com de rolamentos internos, como se pode observar nas figuras abaixo.

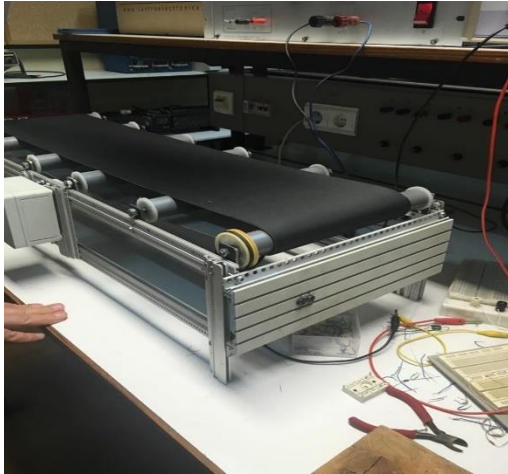


Figura 23 - Tapete Rolante

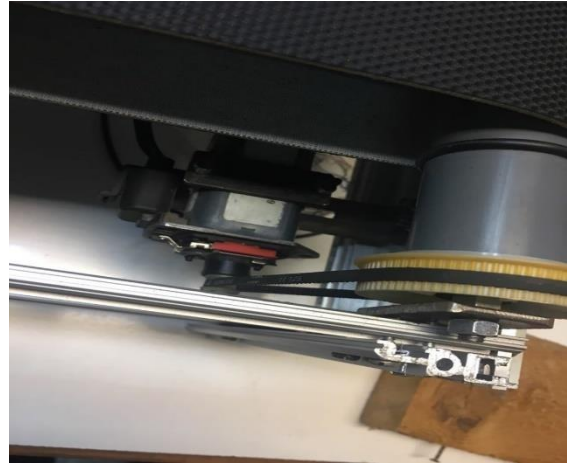


Figura 24 – Motor e correia do Tapete

Os diâmetros e o número de dentes das polias têm influência no dimensionamento eletromecânico do tapete, isto porque o objetivo é conseguir garantir baixas rotações na polia acoplada aos tubos de plástico com rolamentos. Este princípio de funcionamento é chamado de relação de transmissão, que consiste na relação entre as velocidades rotacionais de duas ou mais engrenagens interligadas. Neste caso existem duas engrenagens, sendo a primeira que está a receber a força de rotação diretamente do motor (ou seja, a velocidade de rotação é conhecida) e a segunda engrenagem (velocidade que depende da relação de transmissão) que está interligada aos tubos de rolamentos.

Esta relação de transmissão pode ser determinada usando a seguinte equação (1):

$$i = \frac{\Phi_A}{\Phi_B} \quad (1)$$

A título de exemplo, pode considerar-se a Figura 25, que consiste na determinação da relação de transmissão de duas engrenagens ligadas por meio de uma correia. Considerando que a polia B seja a que esteja acoplada ao motor e a Polia A, a que esteja conectada ao eixo de rotação, neste caso, de forma simples, pode-se efetuar a contagem do número de dentes da polia B, 20 dentes por exemplo, o mesmo processo para a polia A, 30 dentes por exemplo. Agora é simples, basta dividir o número de dentes da polia A pela B, ou seja:

$$i = \frac{n_A}{n_B} = \frac{30}{20} = 1.5$$

Este resultado quer dizer, que para a polia A girar uma vez é necessário que a polia B gire uma vez e meia. Fazendo assim que efetivamente haja uma redução de velocidade dos tubos de rolamentos.

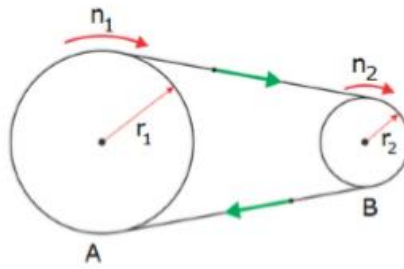


Figura 25 - Esquema de polias acopladas por correia [11]

### 3.2.1 Características do Motor DC usado

A Figura 26 representa uma vista em perspectiva do motor DC usado neste projeto. Este motor DC é personalizado e de alta qualidade e é destacado por ser silencioso e possuir um elevado binário. Este motor disponibiliza também um encoder ou codificador ótico. [12]



Figura 26 - Motor DC 28PA51G da Botnroll [12]

Este motor possui uma caixa de velocidades e um encoder incorporado com uma resolução de 13 pulsos por rotação (PPR). A Figura 27 representa o esquema de ligações elétricas. [12]

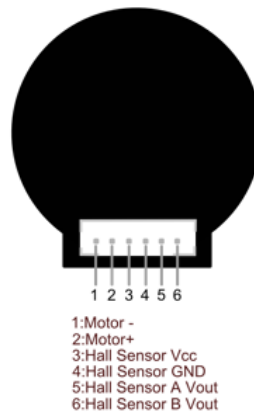


Figura 27 - Terminais de ligação do motor DC com interface de encoder [12]

O encoder poderia ter sido usado para se fazer o controlo de velocidade do motor em malha fechada, mas por limitação de números de saídas e entradas binárias do microcontrolador decidiu-se fazer o controlo do motor em malha aberta, assunto que é descrito com detalhes nos pontos a seguir.

De um modo geral, os motores possuem as seguintes características principais:

- Corrente nominal de 0,23 [A];
- Tensão nominal de 12 [VDC];
- Velocidade de 146 [rpm];
- Binário de 0,98 [N.m].

A corrente nominal corresponde à corrente na qual o motor DC opera em condições nominais. Esta corrente é dependente da carga aplicada no eixo do rotor. Para a escolha do motor considerou-se este aspeto, porque era imperativo garantir que o motor rodasse num regime em que a carga não exigisse mais corrente do que a máxima recomendada pelo fabricante. O motor escolhido tem uma corrente em vazio de 0.23A e uma corrente em plena carga de 3.6A.

A tensão nominal, corresponde a tensão na qual o motor DC pode operar permanentemente sem se danificar, podendo operar em baixa tensão quando se pretende reduzir a velocidade de rotação ou o binário disponível pelo motor.

Estas variáveis são importantes serem conhecidas para que se possa dimensionar

eficientemente os circuitos de potência e controle, como será visto no ponto a seguir.

Ainda sobre as características do motor DC, o binário é a força que o motor pode exercer em relação a uma distância do eixo.

$$T = K_T \times I_a \quad (2)$$

O motor escolhido tem um binário máximo de 0.98 N.m.

Por fim, a velocidade, sendo medida em rotações por minuto (RPM), depende da corrente, da tensão, e da carga aplicada. Qualquer um destes fatores influenciam diretamente na velocidade do motor. A expressão de cálculo pode ser observada abaixo.

$$N = \frac{U - (R_a * I_a)}{K_T} \quad (3)$$

O motor escolhido possui uma caixa de velocidades com uma relação de transmissão de 51:1 com 8000 RPM sem carga antes da caixa de velocidade e com 146 RPM depois da caixa de velocidades. Ou seja, a base de trabalho em relação às velocidades são os 146 RPM que o motor fornece.

### 3.2.2 Aplicação do PWM no Circuito Controlado

Até ao momento já se sabe como variar a velocidade do motor e neste ponto se vai descrever como funciona no circuito conversor da ponte H. Considerando os semicondutores T1, T2, T3 e T4 (

Figura 28).

Nesta situação o semicondutor T1 é fechado e o sinal PWM é aplicado ao semicondutor T4 e variando assim a velocidade no sentido inverso.

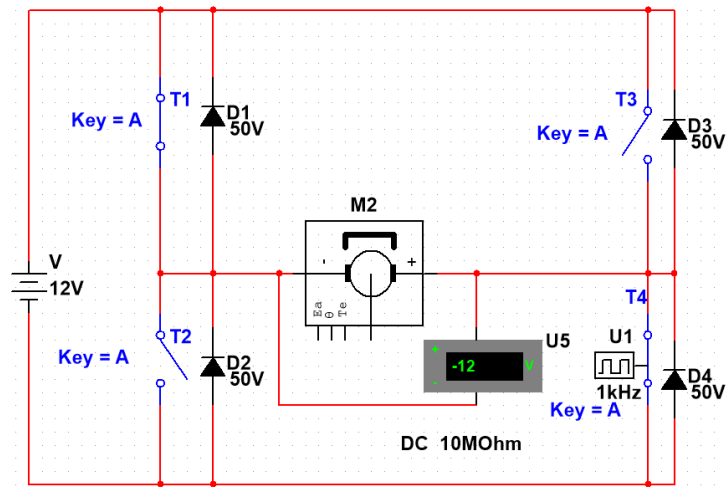


Figura 28 - PWM Aplicado ao *Chopper* de 4 Quadrante

Para variar a velocidade no sentido direto, fecha-se o semicondutor T2 em vez do T1 e o sinal PWM aplica-se no semicondutor T3 em vez do T4.

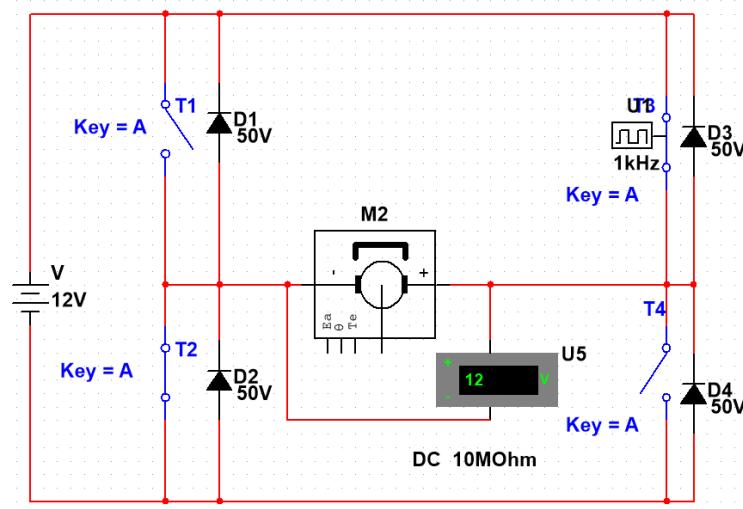


Figura 29 – Aplicação Real do Sinal PWM na Ponte H

A grande desvantagem dos controlos de sinais por PWM deve-se ao facto de possuir frequência de comutação geralmente muito elevada dependendo do tipo de aplicação.

### 3.2.3 Controlador de Velocidade e Sentido de rotação

Para o desenvolvimento do controlador de velocidade e direção a partir do princípio de funcionamento da ponte H surgiram duas grandes questões:

- Existe no mercado pontes H prontas para implementar no projeto?
- Se for pretendido desenvolver uma, quais são as melhores opções de semicondutores de potência?

Após algumas consultas em plataformas de vendas de componentes eletrônicos constatou-se que sim, existem muitos circuitos integrados que possuem a função de ponte H, em particular no ramo automóvel. Mas para a aplicação neste projeto, escolheu-se a opção do *driver* L298N.

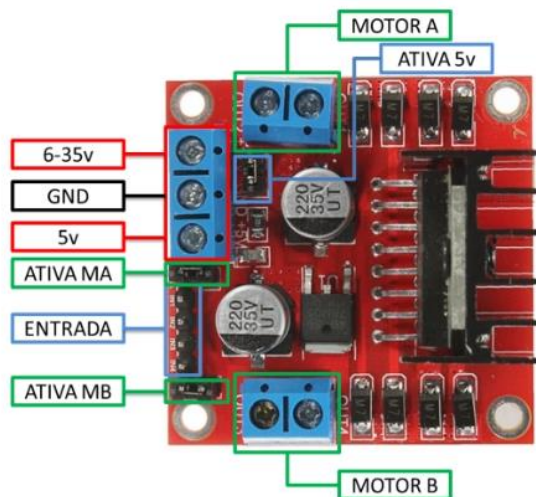


Figura 30 - ponte H L298N

Este *driver* permite o controlo de até dois motores DC ou um motor de passo. É projetado para controlar cargas indutivas como relés, solenoides, motores DC e motores de passo, permitindo o controlo não só do sentido de rotação do motor, como também da sua velocidade através de sinais PWM. Uma das grandes vantagens deste *driver* é a resposta eficiente aos sinais de PWM. Um exemplo de aplicação pode ser observado na Figura 31.

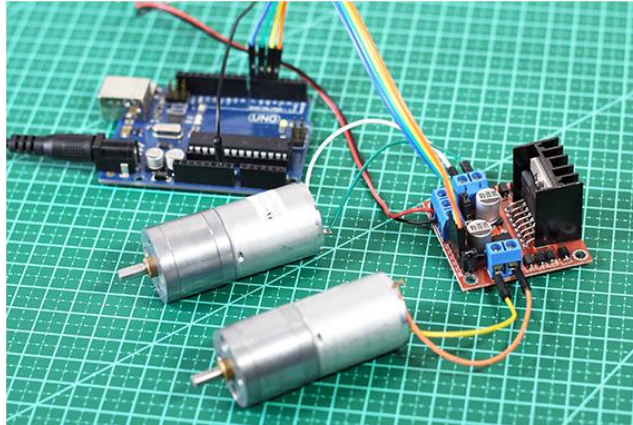


Figura 31 - Exemplo de Aplicação Ponte H

Este *driver* de controlo possui as seguintes características técnicas:

- Tensão de operação de 4-35V DC;
- Chip ST L298N;
- Controlo de 2 Motores DC ou 1 Motor de Passo;
- Corrente de operação máxima de 2A por canal ou 4A máximos;
- Tensão lógica de 5V;
- Corrente lógica entre 0-36mA;
- Limites de temperatura -20 a 135°C;
- Potência máxima de 25W;
- Dimensões de 43x43x27mm;
- Peso 30g

O Chip ST L298N (Figura 32) é um integrado de ponte dupla completa, projetado para acionar cargas indutivas e aceitar sinais de controlo com lógica TTL padrão. Duas entradas de ativação são fornecidas para ativar ou desativar o dispositivo independentemente dos sinais de entrada.

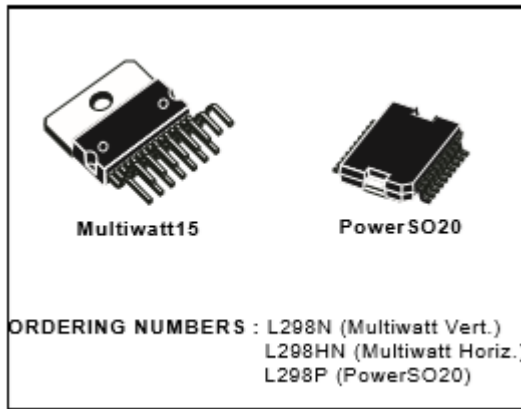


Figura 32 - Chip ST L298N

Os emissores dos transístores inferiores (Figura 33) de cada ponte são conectados juntos e o terminal externo correspondente pode ser usado para a conexão de um sensor resistivo externo. Uma entrada de alimentação adicional é fornecida para que a lógica funcione a uma tensão mais baixa.

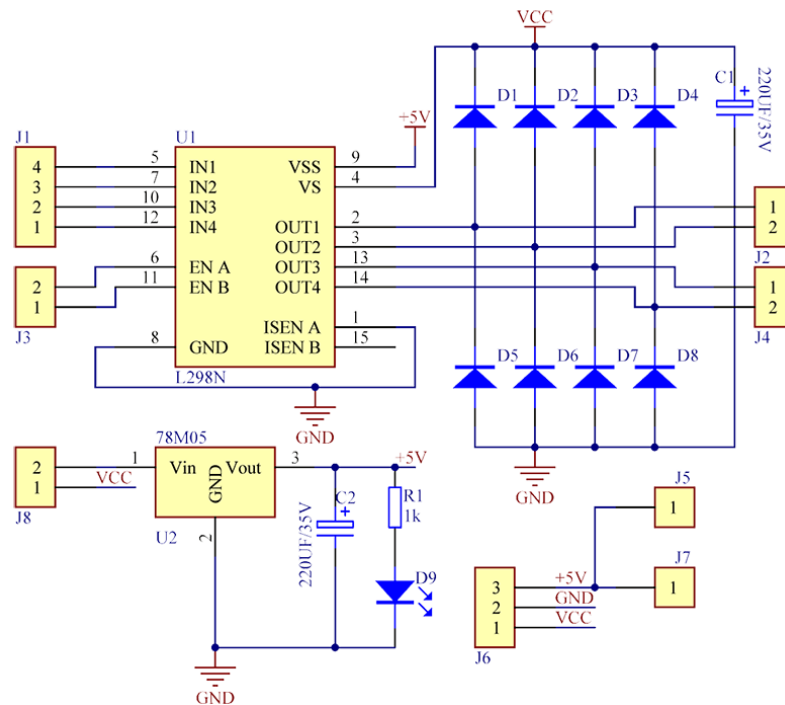


Figura 33 - Diagrama de Blocos do Chip L298N [13]

Em termos de funcionamento da ponte H L298N temos as seguintes configurações:

- (Motor A) e (Motor B): Referem-se aos conectores para as ligações de dois motores DC ou um motor de passo;
- (Ativa MA) e (Ativa MB): São os pinos responsáveis pelo controlo dos sinais PWM dos motores A e B respetivamente. Caso os conectores estiverem com *jumpers* significa que não haverá controlo de velocidade, pois os pinos estarão interligados aos 5V. Estes pinos podem ser usados em conjunto com os pinos PWM do Microcontrolador;
- (Ativa 5V) e (5V), esta opção é em relação ao regulador de tensão que este driver possui. Ou seja, quando a ponte estiver a operar entre 6-35V DC o regulador de tensão disponibiliza uma saída em tensão regulada de +5V no seu pino 5V que para uma utilização externa poder alimentar um circuito;
- (6-35V) e (GND): Pinos que servem para a alimentação do driver enquanto este tiver a controlar o motor DC;
- (Entrada): Este barramento é composto pelos pinos IN1, IN2, IN3 e IN4. Estes pinos são as entradas da ponte que recebem sinais PWM do microcontrolador e que fazem os motores A e B rodarem e alterar os sentidos. IN1 e IN2 correspondem ao motor A e IN3 e IN4 ao motor B.

Relativamente à saída do *driver* para o controlo do motor foi escolhida a opção referente ao motor A, tenho neste caso os sinais de entrada IN1 e IN2. Os comandos são dados de acordo a Tabela 5.

Tabela 5 - Funções de Controlo do Motor

<b>Funções do Motor</b>	<b>IN1</b>	<b>IN2</b>
Rodar em Sentido Horário	5V	GND
Rodar em Sentido Anti-Horário	GND	5V
Paragem Opção A	GND	GND
Paragem Opção B	5V	5V

Existem duas formas possíveis para fazer as ligações elétricas do motor ao *driver*. Uma das formas é se o motor a ser usado funcionar com uma alimentação de 5V DC como se pode observar na Figura 34.

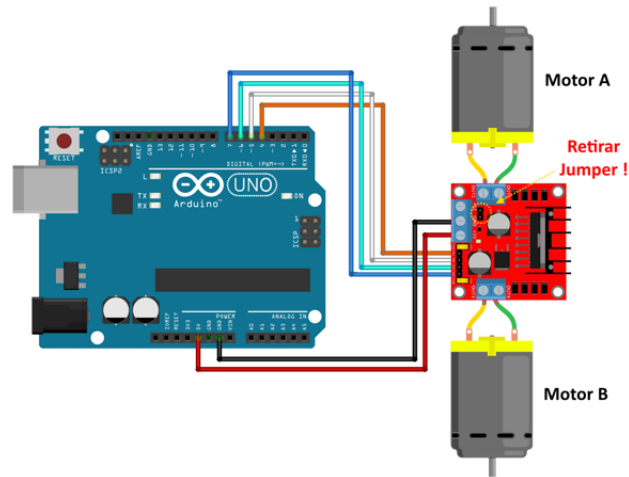


Figura 34 - Esquema de Ligação Com Motores de 5V

Nesta situação o conector no pino “Ativa 5V” deverá ser desconectado para permitir que a alimentação do motor venha direto do próprio canal de alimentação do microcontrolador.

Em alternativa, se o motor for de tensão superior a 5V como neste caso prático (12V), o *driver* deverá ser alimentado a partir de uma fonte externa e mantendo o conector conectado no pino Ativa 5V como se pode observar na Figura 35.

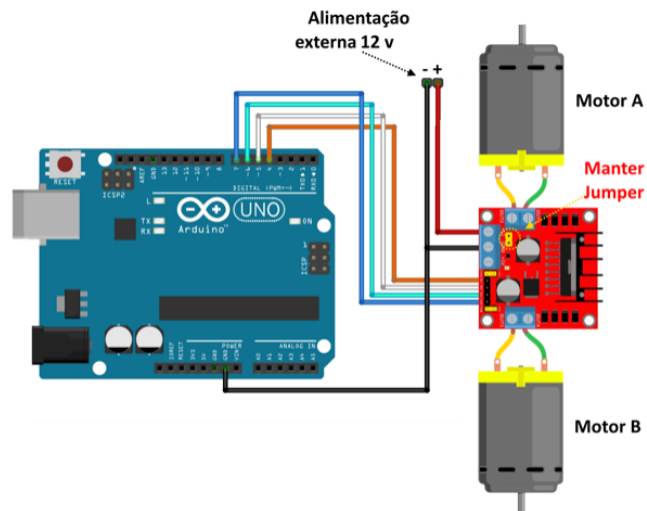


Figura 35 - Esquema de Ligação Com Motores de 12V

### 3.3 PLC ILC 131 ETH Phoenix Contact

O sistema do tapete rolante funciona em modo manual e automático, ambos controlados pelo PLC ILC 131 existente no Laboratório de Automação e Robótica. A Figura 36 representa o sistema de controlo físico existente atualmente no laboratório.

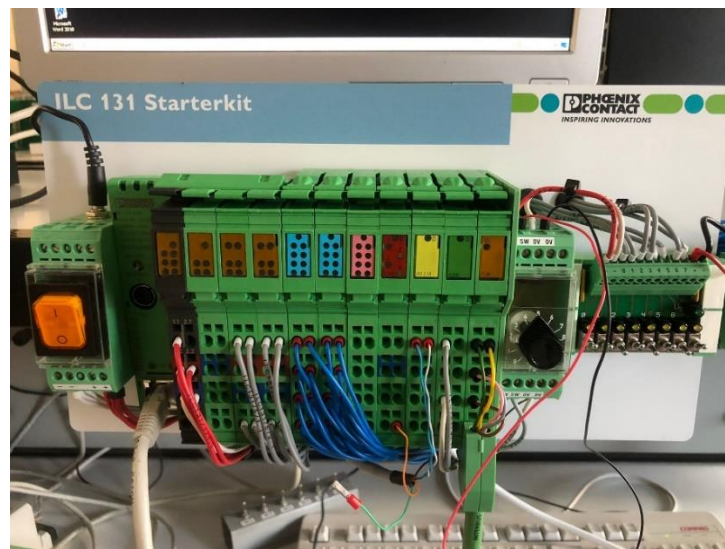


Figura 36 - PLC ILC 131 ETH Phoenix Contact

Este equipamento é programado e parametrizado pelo software do próprio fabricante, PC Worx baseado na norma IEC 61131-3. Este controlador pertence a família dos controladores *inline* e pode comunicar através dos protocolos PROFINET e Modbus/TCP baseados em Ethernet. Em forma de configuração básica possui oito pontos de entrada digital e quatro pontos de saída digital, podendo estes serem expandidos ou reduzidos para atender as necessidades de novas aplicações através de novos módulos de expansão de entrada e saída tanto analógicos quanto binários e instalados ao lado do PLC, já que este equipamento é modular. Possui também uma porta Ethernet e uma porta serial para configuração, parametrização e comunicação com o sistema de supervisão.

Um cabo de rede Ethernet deve ser conectado entre a porta RJ45 do PLC e a porta RJ45 do PC de comando e dever-se-á a seguir garantir, que tanto o PC quanto o PLC devam estar na mesma gama de IPs.

Device Details	
Projeto_SantoFreitas \Project\	
Project name	Value
Project name	Projeto_SantoFreitas
Creator	7WRNZF
Computer name at project creation	PXCSW-N0082
MULTIPROG version at project creation	5.51.8.0
PC WORX version at project creation	PC WORX 6.30.1503
Creation date	2015-09-22T08:46:08+01:00
Last editor	BRIDGEPOINT-SF
Computer name at last project backup	DESKTOP-OLDH1DK
MULTIPROG version at last project backup	5.51.689.0
PC WORX version at last project backup	PC WORX 6.30.2972
Date of last project backup	2021-11-11T20:42:59-00:00
Domain Postfix	
Template for DNS name creation	
First IP Address	192.168.0.2
Last IP-Address	192.168.0.254
Subnetmask	255.255.255.0
Default Gateway	
Use DHCP	No
Subnet Check for Multi MAC Devices	On
Certificate information	
Organization	PHOENIX CONTACT GmbH & Co. KG
Organizational Unit	
Locality	Blomberg
State or Province	Nordrhein-Westfalen
Country	DE

Figura 37 - Detalhes de configuração do PLC ILC 131 ETH

Por exemplo, considerando as configurações da Figura 37, implica, que para que seja possível comunicarmos com o mesmo, o computador de comando deverá estar na mesma máscara de rede e com um endereço de IP na mesma gama, por exemplo 192.168.0.7.

Uma vez estabelecida a comunicação entre o computador e o PLC, o passo a seguir é efetuar um *scan* ou inserir manualmente a referência de cada carta conectada ao mesmo:

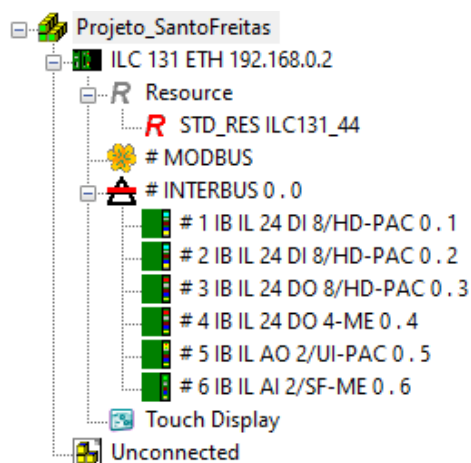


Figura 38 - Imagem dos módulos de expansão adicionados

O passo final é basicamente familiarizar-se com o ambiente de programação e os seus respetivos recursos. A Figura 39 representa a visão geral do ambiente de programação do *software* PC WORX da Phoenix Contact.

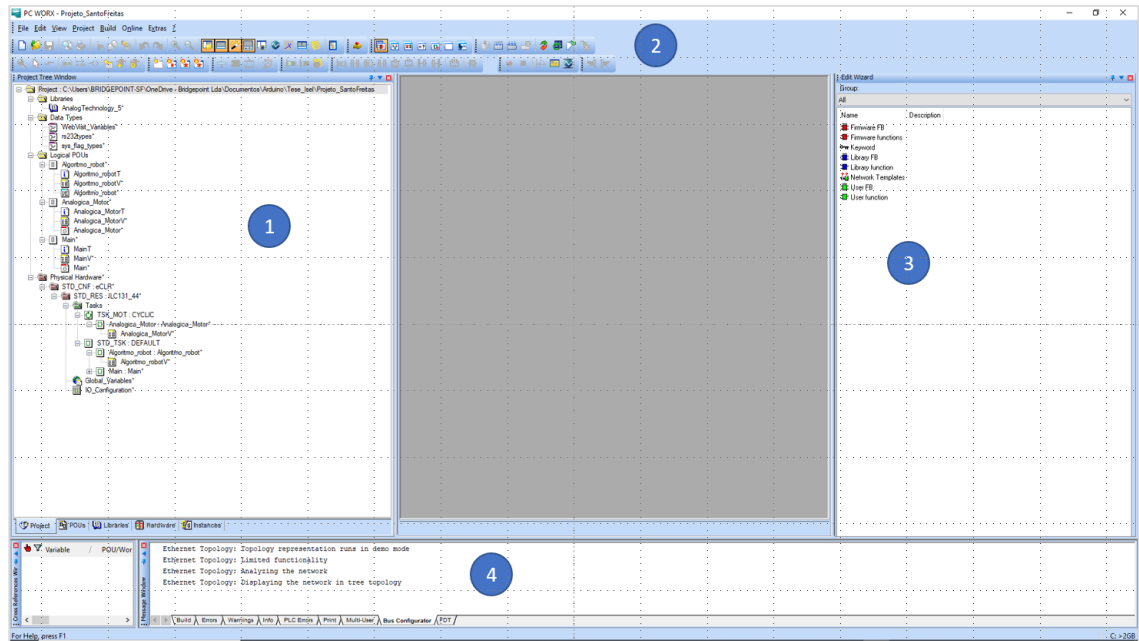


Figura 39 - Visão geral PC WORX

Como se pode observar na Figura 39, o ponto 1 representa a árvore do projeto. É aqui onde é possível desenvolver a programação, criar funções, chamar programas, criar tipo de dados, criar variáveis locais e globais e adicionar bibliotecas específicas. Já na Figura 40 pode-se observar com melhor qualidade este ponto.

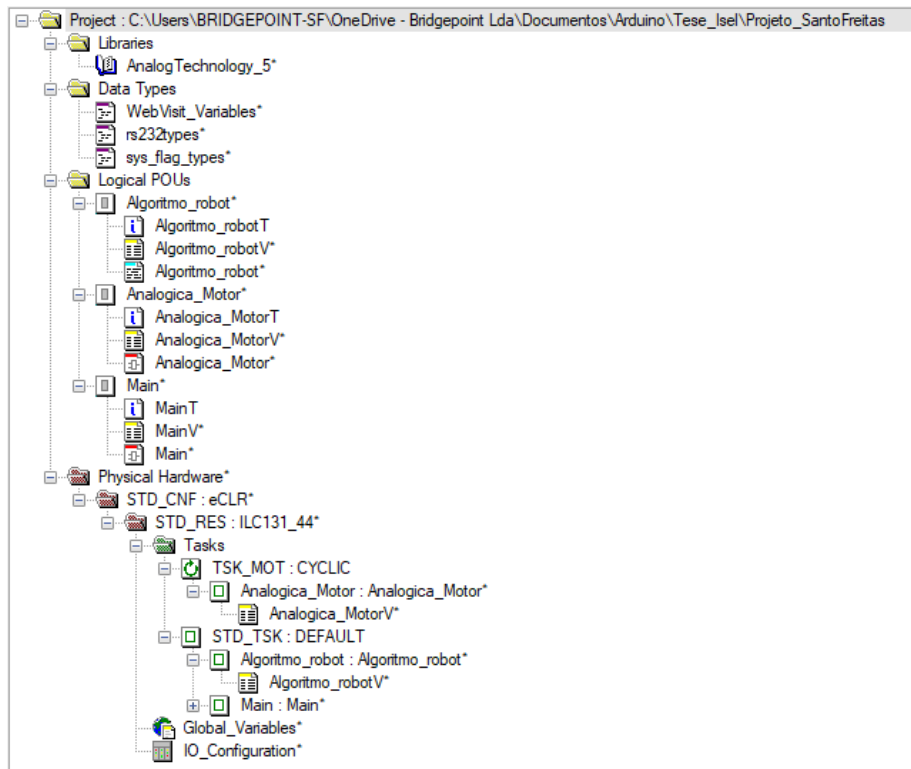


Figura 40 - Raíz do projeto

O ponto 2 representa a barra de ferramentas com muitas opções uteis e com atalhos. Realço o atalho de *download* do programa e o de *debug* do programa.



Figura 41 - Barra de ferramentas

O ponto 3 é um dos mais utilizados, uma vez que é lá onde se acedem as funções e operações. A Figura 42 ilustra resumidamente o que se pode encontrar neste ponto.

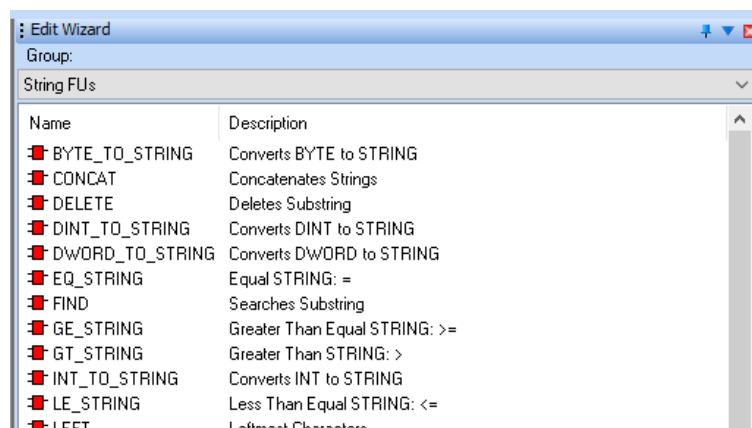


Figura 42 - Funções PC WORX

Por fim o ponto 4, que apresenta basicamente os resultados da compilação do programa, tais como avisos e erros e informações do próprio PLC.

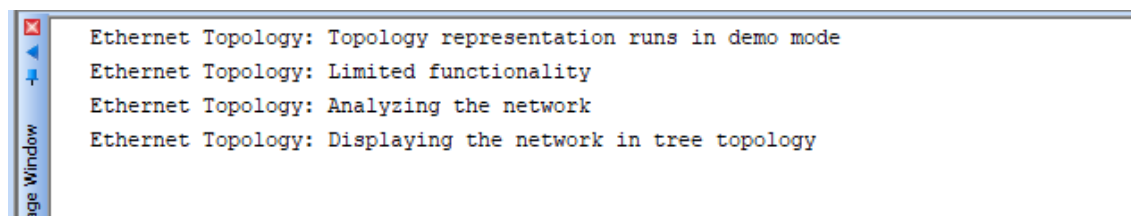


Figura 43 - Janelas de mensagem PC WORX

Para concluir esta breve apresentação do *software* de programação PC WORX, realço aqui dois recursos importantes, que este PLC possui e que serão melhor detalhados ao longo deste trabalho. Trata-se da possibilidade de comunicação por RS232 com dispositivos externos, que no caso desta dissertação foi o Arduíno e a possibilidade de usar o sistema de supervisão graças ao servidor web que o mesmo possui e programado pelo *software Webvisit*. Ambos recursos foram a chave para que este trabalho de dissertação fosse concluído com sucesso.

### 3.4. Características do Arduíno UNO usado

O UNO foi escolhido por ser compatível com várias aplicações e plataformas e apresentada um custo razoável. É baseado no microcontrolador ATmega328 da ATMEL, possui catorze pinos de entradas/saídas digital dentre os quais seis têm a opção de PWM, seis entradas analógicas, uma conexão USB, uma entrada de alimentação, uma conexão ICSP, um botão de *reset* e dois pinos que servem de interface para a comunicação série. Estas características podem ser observadas na Figura 44.

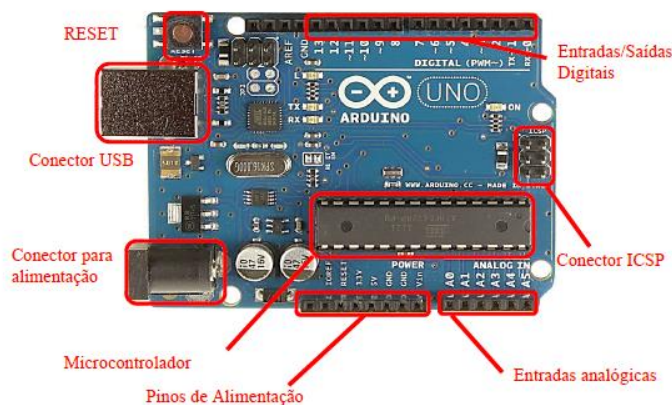


Figura 44 - Descrição Arduino UNO [7]

A placa pode ser alimentada a partir da conexão USB ou com uma fonte de alimentação externa ou ainda por uma bateria, porque a alimentação é selecionada pelo Arduino automaticamente. A tensão recomendada é de 5VDC.

A Tabela 6 representa com mais detalhes as funções dos pinos de alimentação do Arduino.

Tabela 6 – Descrição dos Pinos de Alimentação Arduino UNO

VIN	Entrada de alimentação da placa quando se está a utilizar uma fonte de alimentação externa.
GND	Pinos terra, tensão 0 V de referência.
5V	Fornecer tensão de 5 V para alimentação de sensores, placas auxiliares e circuitos externos. A corrente máxima suportada pelo microcontrolador é de 200 mA.
3V3	Fornecer tensão de 3,3 V para alimentação de sensores, placas auxiliares e circuitos externos. A corrente máxima suportada é de 50mA.
RESET	Pino conectado ao RESET do microcontrolador. Pode ser utilizado para um <i>reset</i> externo do microcontrolador. Quando este pino comuta para o valor LOW (0V) o microcontrolador é reiniciado.

Em relação aos quatorze pinos digitais da placa, estes podem ser usados como entrada ou como saída digital e apresentam dois níveis lógicos de tensão (Low - Nível baixo 0V) e (High – Nível alto 5V) com uma corrente de até 40mA. Existem algumas particularidades em alguns dos pinos tais como:

- Serial: Pinos 0 (RX) e 1 (TX). São utilizados para garantir a comunicação série, ou seja, o pino RX recebe os dados da porta série e o TX envia;
- Interrupções Externas: Os pinos 2 e 3 podem ser configurados para ativar a rotina de

serviço a interrupção externa;

Para se fazer a programação do Arduíno usa-se uma linguagem própria proveniente do C++. O ambiente de desenvolvimento é uma plataforma composta por um editor de texto (Figura 45) para escrever o programa.

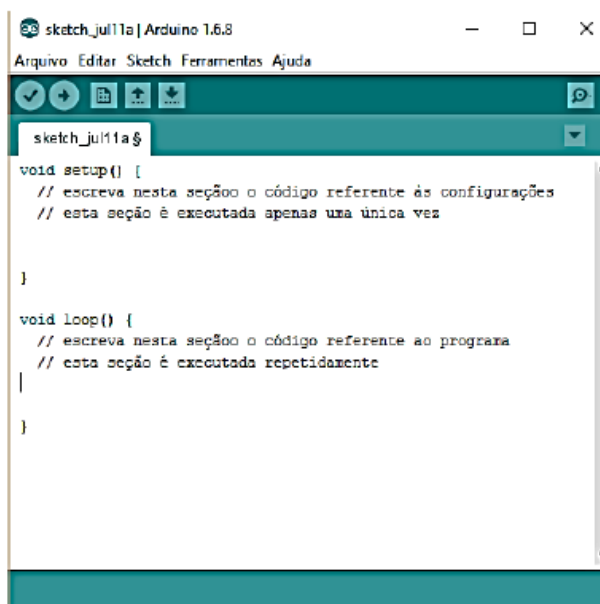


Figura 45 - Ambiente de Programação Arduíno

### 3.5 Comunicação RS232 entre o PLC ILC 131 ETH e o Arduíno UNO

A mensagem enviada pelo PLC contém 27 caracteres separados por vírgula e a descrição encontra-se na Tabela 7.

Tabela 7 - Trama RS232 entre PLC e o Arduíno

Trama enviada pelo PLC ao Arduíno	
Start, sentidoTapete, M1, M2, M3, M4, M5, M6	
Exemplo: 0,2,090,090,090,090,090	
Start=0	Indica que o tapete e robô devam parar
Start =1	Indica que o tapete e o robô têm a liberdade de iniciar os seus movimentos, dependendo da ordem do sentido de rotação do tapete
SentidoTapete=0	Indica que o tapete deve parar e (o robô entrar em modo teach caso Start seja=1)
SentidoTapete=1	Indica que o tapete deve rodar em sentido

	direto
SentidoTapete=2	Indica que o tapete deve rodar em sentido inverso
M1 a M6 =xxx	Indica valores de ângulos entre 000-180º para cada eixo do robô, desde o eixo 1 ao eixo 6

Na Figura 46 é possível observar a lógica implementada no PLC para o envio da mensagem, sendo que as variáveis "Run\_Sistema", "Ordem\_Tapete", "Junta\_M1 a Junta M6" são escritas diretamente pelo utilizador a partir da aplicação do sistema de supervisão.

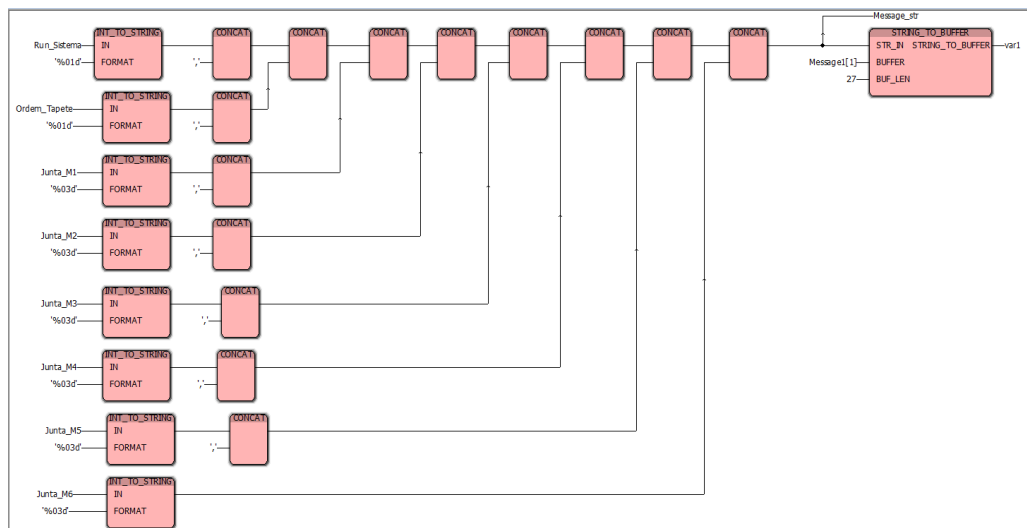


Figura 46 - Trama RS232 enviada pelo PLC

Do lado do Arduino que controla o robô (onde está a comunicação RS232) desenvolveu-se um algoritmo que lê frequentemente a informação no buffer da porta série e guarda a informação recebida numa variável. Após isto o algoritmo separa em variáveis distintas a informação lida tendo como base a vírgula que separa os valores esperados. (Algoritmo em anexo).

### 3.5.1 Configuração física RS232 no PLC ILC 131 ETH

Como já mencionado em pontos anteriores, este PLC possui uma interface direta de comunicação RS232 como se pode observar na Figura 47. Para isto é necessário que de um lado do cabo, esteja conectada uma ficha do tipo PS/2 e do outro lado uma ficha DB9 fêmea. Como se pode observar, os terminais 1,2 e 4 da ficha PS/2 e do lado da DB9 os terminais 2,3 e 5 respetivamente.

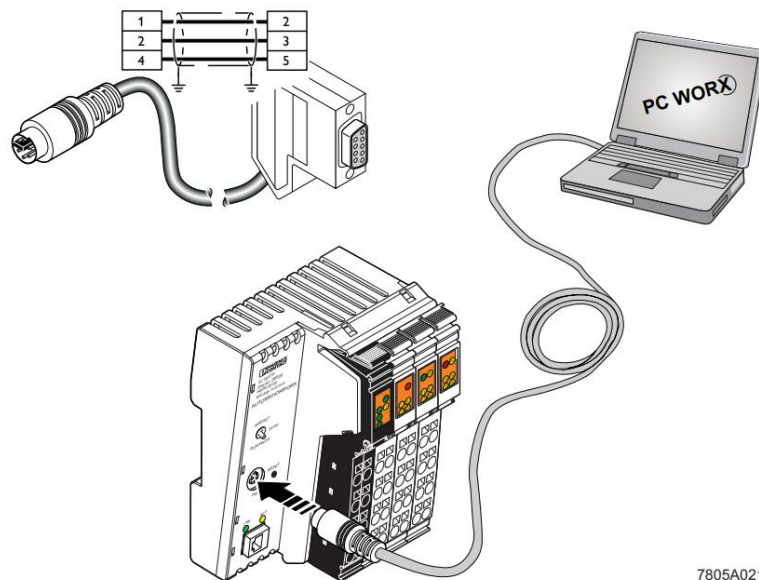


Figura 47 - Ligação RS232 PLC ILC 131 ETH

Nesta configuração é considerada apenas o circuito de dados (TxD,RxD,GND) a 3 fios. A mencionar na Tabela 8.

Tabela 8 - Pinout RS232 PLC ILC 131 ETH

Terminal da ficha PS/2	Designação	Terminal da ficha DB9	Designação
1	TxD	2	RxD
2	RxD	3	TxD
4	GND	5	GND

Uma vez estabelecida corretamente a ligação elétrica para a comunicação, o passo a seguir é efetuar a configuração no *software*. O PC WORX possui blocos diretos de inicialização, envio e recepção de dados RS232 como se pode observar na Figura 48 retirada do manual do fabricante.

### 2.12.3 Serial PRG interface - function blocks

This interface can be used for the following:

- Assignment of the IP address or working with Diag+
- Communication with I/O devices via function blocks

The following function blocks are available in the PC Worx/PC Worx Express software:

Table 2-2 Function block overview

Function block	Short description
RS232_INIT	Parameterization of the serial interface  You can use this function block to specify the following parameters of the serial interface: <ul style="list-style-type: none"> <li>- Protocol: Transparent</li> <li>- Baud rate: 1200, 2400, 4800, 9600, 19200, 38400, 57600 or 115200</li> <li>- Data width: 8 data bits, even parity</li> <li>- Number of stop bits: 1</li> <li>- Hardware flow control: None</li> </ul>
RS232_RECEIVE	Reading the internal receive memory of the serial interface
RS232_SEND	Data transmission to the internal transmit memory of the serial interface

Once the RS232\_INIT function block has been activated, the interface is parameterized accordingly so that it is only possible to communicate with the connected I/O devices.

Deactivating the RS232\_INIT function block or performing another cold restart or warm start enables the IP address to be assigned and the controller to be accessed with Diag+.

For additional information about the function blocks, please refer to the online help for the PC Worx/PC Worx Express software.

#### I/O devices that can be connected

Various I/O devices (e.g., modem, printer, barcode scanner) can be connected to the controller via the serial PRG interface. The connected devices are addressed from the application program using function blocks.

Figura 48 - Interface RS232 PLC ILC 131 ETH

### 3.5.2 Bloco RS232\_INIT

Este bloco é responsável por definir os parâmetros de comunicação, tais como a taxa de transmissão, o fluxo de controlo, o protocolo, o número de dados de transmissão e o sinal de paragem. Na Figura 49 pode-se observar o respetivo bloco.

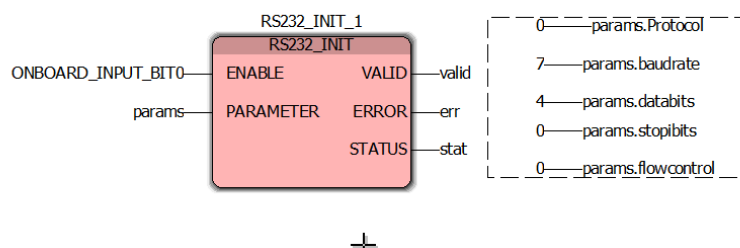


Figura 49 - RS232 INIT PC WORX

Este bloco é habilitado apenas se o utilizador colocar em estado ligado o interruptor associado ao endereço ONBOARD\_INPUT\_BIT0. A variável "params" é uma estrutura de dados

do tipo T\_RS232, com a seguinte configuração:

```

1 Type
2 T_RS232 : STRUCT
3     protocol      :INT; (* 0: transparent,
4                       data transmission without protocol *)
5     baudrate      :INT; (* 3: 1200kBit,
6                       5: 2400kBit,
7                       6: 4800kBit,
8                       7: 9600kBit,
9                       8: 19200kBit,
10                      9: 38400kBit,
11                     10: 57600kBit,
12                     11: 115200kBit *)
13     databits     :INT; (* 4: 8 data bits, even parity *)
14     stopbits     :INT; (* 1: 1 stop bit *)
15     flowcontrol  :INT; (* 0: not supported *)
16     error_pattern :INT; (* not supported *)
17     first_delimiter :INT; (* not supported *)
18     second_delimiter :INT; (* not supported *)
19     XON_pattern  :INT; (* not supported *)
20     XOFF_pattern :INT; (* not supported *)
21     END_STRUCT;
22 END_TYPE

```

Figura 50 - Estrutura de dados do bloco RS232\_INIT

### 3.5.3 Bloco RS232\_RECEIVE & RS232\_SEND

A Figura 51 representa os blocos de envio e recepção de mensagem. No caso deste trabalho de dissertação apenas são enviadas mensagem para o Arduino a cada 200ms. O formato da mensagem enviada contém 27 caracteres que como foi explicado nas secções anteriores.

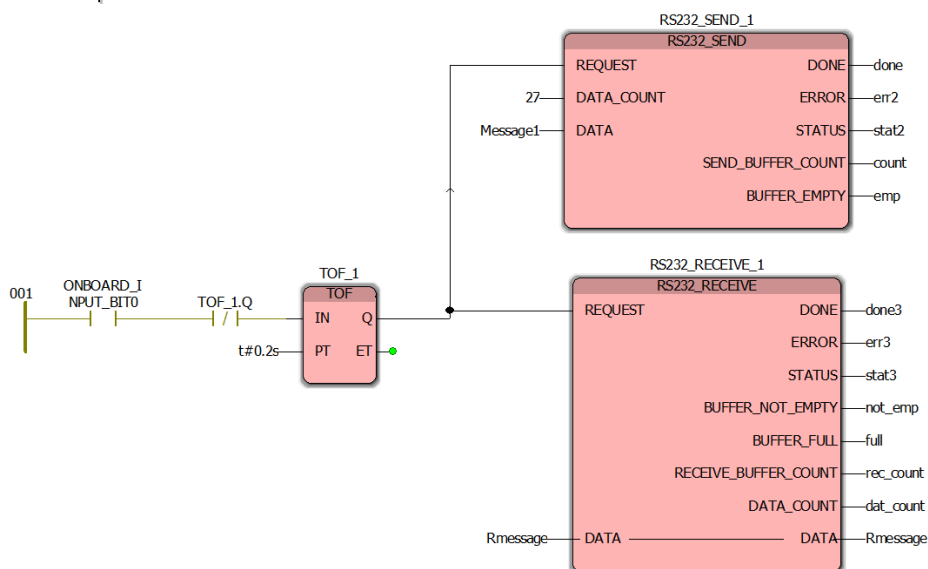


Figura 51 - Bloco RS232\_RECEIVE & RS232\_SEND

### 3.5.4 Configuração RS232 no Arduíno

A Comunicação Série no Arduíno é um recurso utilizado para fazer a comunicação ponto a ponto entre o Arduíno e um computador, um módulo, outro dispositivo com estes recursos ou até mesmo outro Arduíno.

A comunicação série nos pinos TX / RX usa níveis lógicos de tensão TTL (5V ou 3,3V dependendo da placa). Portanto não se deve conectar estes pinos diretamente a uma interface série RS232, uma vez que elas operam com níveis de tensão que podem chegar a +/- 15 V e podem danificar a placa Arduíno.

Sabemos que o PLC ILC 131 ETH mantém a comunicação RS232 com computadores, modem, leitores de código de barras e etc. Mas para que seja possível a comunicação com microcontroladores é necessário que se faça uma adaptação de sinal. Esta adaptação de sinal baseia-se num circuito capaz de converter sinais lógicos de tensão RS232 para sinais com níveis lógicos de tensão de 5V ou 3.3V. Este conversor é conhecido como conversor RS232/TTL.

### 3.5.5 Conversor RS232 para TTL

O módulo conversor RS232 MAX3232 para DB9 é usado para comunicar microcontroladores de nível TTL a um PC ou dispositivo industrial, que neste caso é um PLC, por meio da porta série de nível RS232-C padrão. O módulo usa um conector de pino de 4 vias (GND / RxD / TxD / VCC) para fazer a *interface* com o Arduíno e outro conector DB9 fêmea para conectar ao dispositivo RS232, que neste caso é o PLC. A Figura 52 representa o conversor usado no projeto.



Figura 52 - Conversor RS232/TTL

Na Figura 53 apresento um esquema genérico de ligação elétrica entre o PLC ao conversor RS232/TTL e entre o conversor RS232/TTL ao Arduino. Nota: Antes de descarregar uma nova programação no Arduino, os pinos TxD e RxD dos mesmos devem estar livres de conexão externa, caso contrário há um risco enorme de danificar a placa durante este processo.

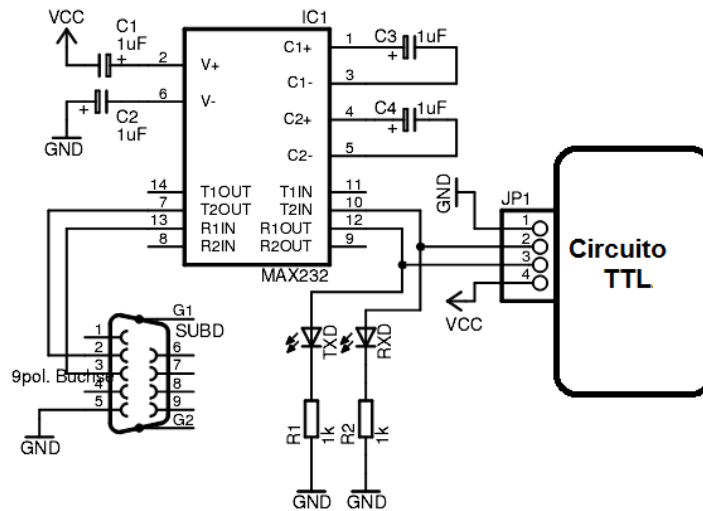


Figura 53 - Esquema de ligação Conversor RS232/TTL

Para concluir este capítulo, considerando que as ligações elétricas estão bem feitas, basta apenas iniciar a aplicação do Arduino e no Setup definir a taxa de transmissão, que neste caso é de 9600Kb/s. Segue o exemplo na Figura 54.

```
Braccio_SF_Tese | Arduino 1.8.16 (Windows Store 1.8.51.0)
Ficheiro Editar Rascunho Ferramentas Ajuda

Braccio_SF_Tese
#include <Braccio.h>
#include <Servo.h>
#include <String.h>

Servo base; // Estrutura para controlo da Junta M1
Servo shoulder; // Estrutura para controlo da Junta M2
Servo elbow ; // Estrutura para controlo da Junta M3
Servo wrist_ver; // Estrutura para controlo da Junta M4
Servo wrist_rot; // Estrutura para controlo da Junta M5
Servo gripper; // Estrutura para controlo da Junta M6

String PLC_CMD; //Variavel para receber a trama RS232 vinda do PLC ILC 131 ETH da Phoenix Contact
int Tapete_1 = 4; // Sinal elétrico enviado ao arduino que faz o controlo do tapete rolante
int Tapete_2 = 7; // Sinal elétrico enviado ao arduino que faz o controlo do tapete rolante
int start=0,rotacao=0,M1=90,M2=90,M3=90,M4=90,M5=90,M6=90,M3_security=0; // Variáveis inteiras para auxiliar no controlo do sistema
String getValue(String data, char separator, int index); // Protótipo da função que efetua split a trama recebida via RS232

void setup() {

  pinMode(Tapete_1,OUTPUT); // Define pino 4 como entrada
  pinMode(Tapete_2,OUTPUT); // Define pino 7 como entrada
  Braccio.begin(); // Função que manda o robot para uma posição de setup inicial -> Obrigatório para habilitar os 6 eixos do robô
  Serial.begin(9600); // Inicializa a comunicação serial
  digitalWrite(Tapete_1,HIGH); // Ativa para indicar que a comunicação entre o PCL e o arduino ainda não foi estabelecida
  digitalWrite(Tapete_2,HIGH); // Ativa para indicar que a comunicação entre o PCL e o arduino ainda não foi estabelecida
}
```

Figura 54 - Inicialização série Arduino

### 3.5.6 Ambiente de desenvolvimento WebVisit

A aplicação foi desenvolvida no *software* Webvisit da Phoenix Contact. É uma ferramenta de engenharia usada para criar visualizações *web* para todos os controladores com servidor web integrado, como é o caso do PLC ILC 131 ETH. Foi usada a versão *express*, que limita o desenvolvimento da aplicação em até no máximo 10 páginas e 60 variáveis partilhadas. A Figura 55 representa de uma forma genérica o ambiente de desenvolvimento desta aplicação.

## User interface

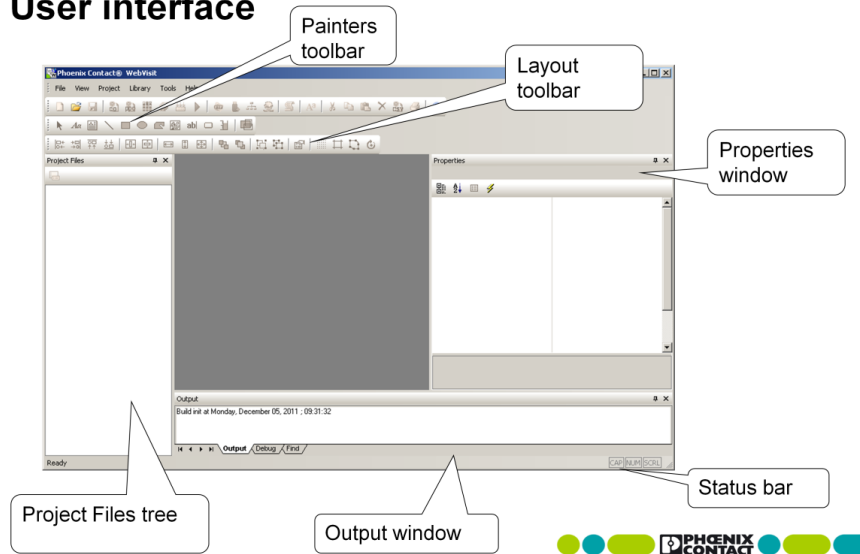


Figura 55 - Webvisit Express

Para que as variáveis estejam acessíveis para consumo ou produção pela aplicação da supervisão, as mesmas devem antes de tudo serem selecionadas no PC Worx através da caixa de texto PDD, como se pode observar na Figura 56.

Name	Typ	Verwendung	Beschreibung	Adresse	Anfangswert	Remanent	PDD	OPC
<b>WebVisitVariablen</b>								
Mixer	ST_Mixer	VAR_GLOBAL	Als Strukturvariable. Wird in der POE Main über die Variable ON...			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ON_OFF_Button_Mixer	BOOL	VAR_GLOBAL	Schaltet den Schaltzustand der Strukturvariable aus der Visu			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Ventilatorordre	INT	VAR_GLOBAL	Simuliert den drehenden Ventilator			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Gasengine	BOOL	VAR_GLOBAL	Wird über Debug simuliert			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Gasthroughput	INT	VAR_GLOBAL	Wird über Debug simuliert. Hier z.B. von 0 bis 30 m³			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Heating	BOOL	VAR_GLOBAL	Schaltet die Heizung aus der Visu			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ProgramInfo	AR_0_2	VAR_GLOBAL	Für das Makro Dropdownlist			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Ventilator	BOOL	VAR_GLOBAL	Schaltet den Ventilator aus der Visu			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sawtooth	DINT	VAR_GLOBAL	Ein Sägezahn von 0 bis 30000			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Fill_Level	WORD	VAR_GLOBAL	Das Polt als externe Füllvelelsimulation	%MW2		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>System Variables</b>								
PLCMODE_ON	BOOL	VAR_GLOBAL	PLC status ON	%MX1.0.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
PLCMODE_RUN	BOOL	VAR_GLOBAL	PLC status RUN	%MX1.0.1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
PLCMODE_STOP	BOOL	VAR_GLOBAL	PLC status STOP	%MX1.0.2		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
PLCMODE_HALT	BOOL	VAR_GLOBAL	PLC status HALT	%MX1.0.3		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
PLCDEBUG_BPSET	BOOL	VAR_GLOBAL	Breakpoint set	%MX1.1.4		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLCDEBUG_FORCE	BOOL	VAR_GLOBAL	Variable(s) forced	%MX1.2.0		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLCDEBUG_POWERFL...	BOOL	VAR_GLOBAL	Powerflow ON	%MX1.2.3		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLC_TICKS_PER_SEC	INT	VAR_GLOBAL	System ticks per second	%MW1.44		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLC_SYS_TICK_CNT	DINT	VAR_GLOBAL	Number of system ticks	%MD1.52		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Variables with the PDD flag are provided in WebVisit

Figura 56 - WebVisit PDD Flags [14]

Após o passo anterior estar concluído deve-se descarregar a programação no PLC e manter localmente uma versão atualizada do projeto para que possa ser possível importar no software WebVisit, como se pode observar na Figura 57.

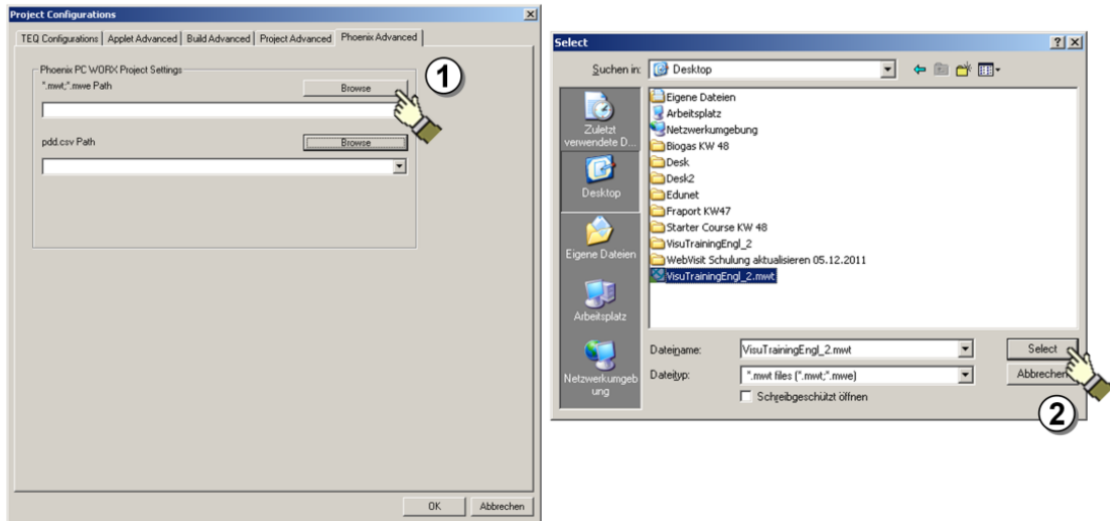


Figura 57 - WebVisit (Importar mwt) [14]

Com o projeto importado no WebVisit é possível inserir botões, textos, tabelas e imagens e de seguida mapeá-las com as variáveis que anteriormente foram partilhadas. Abaixo uma pequena ilustração do mapeamento de uma variável.

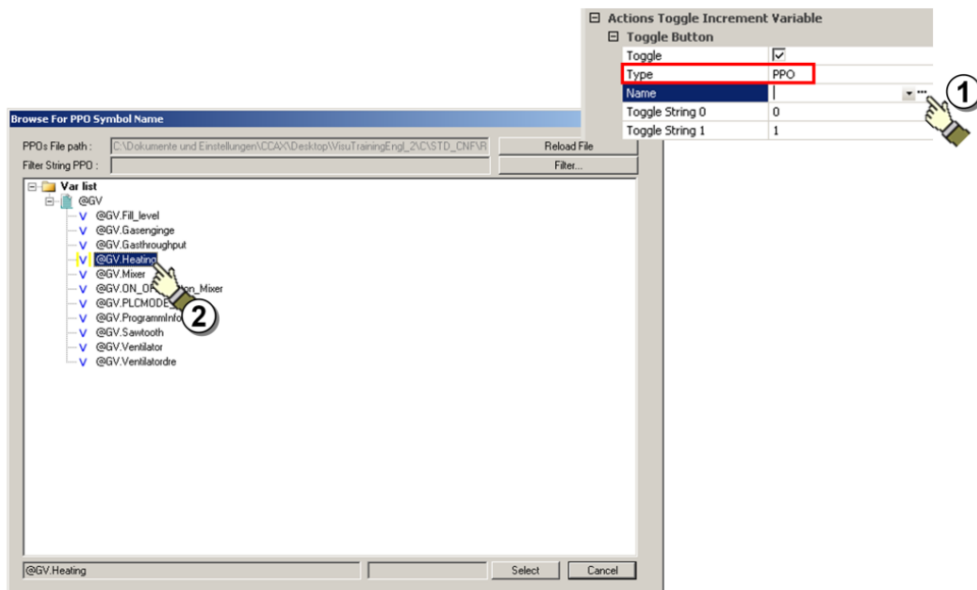


Figura 58 - WebVisit (Mapeamento de Variáveis)

### 3.5.7 Aspetto final da aplicação desenvolvida

Para aceder à *interface* final é necessário que o computador que esteja conectado com o PLC possua um navegador *web*, após isto basta abrir o navegador e no motor de busca principal inserir o endereço IP do PLC. A Figura 59 representa o menu principal.

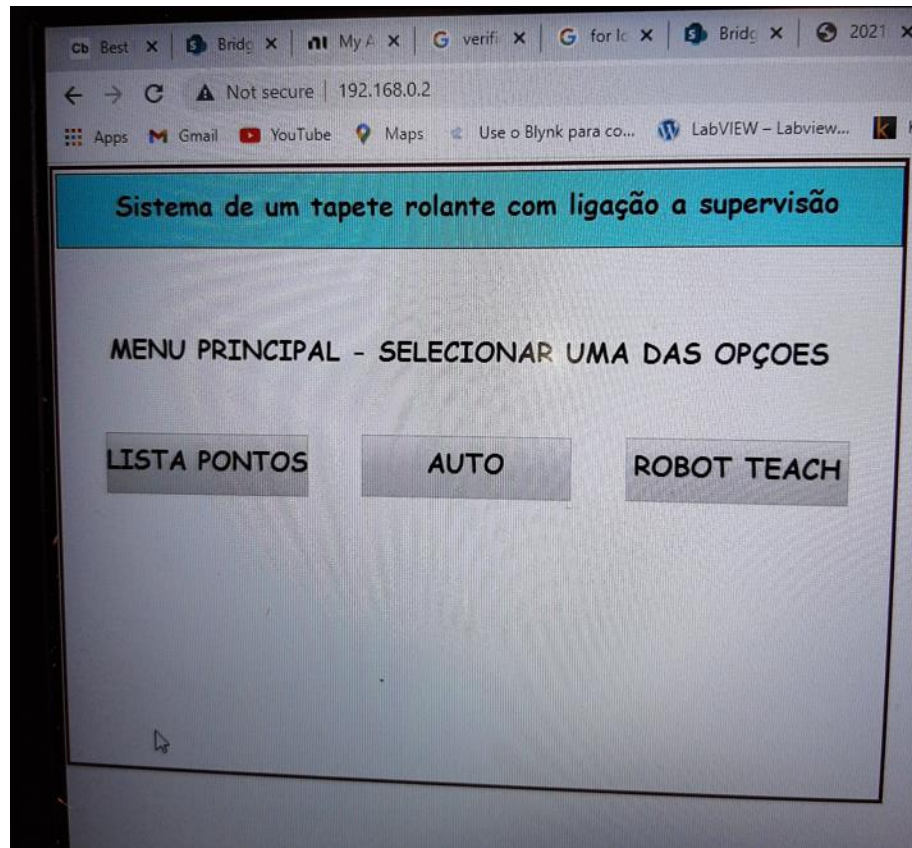


Figura 59 - Menu Principal

Neste menu é possível escolher abrir a página da lista de pontos, do modo automático ou a página de ensinamento de pontos do robô.

Se a opção selecionada for a página “Robot Teach” aparecerá o que está representado na Figura 60, onde o utilizador terá a capacidade de manipular individualmente as juntas do robô. A utilização desta aplicação pressupõe-se que o utilizador já tenha lido a informação que se encontra na Tabela 7.

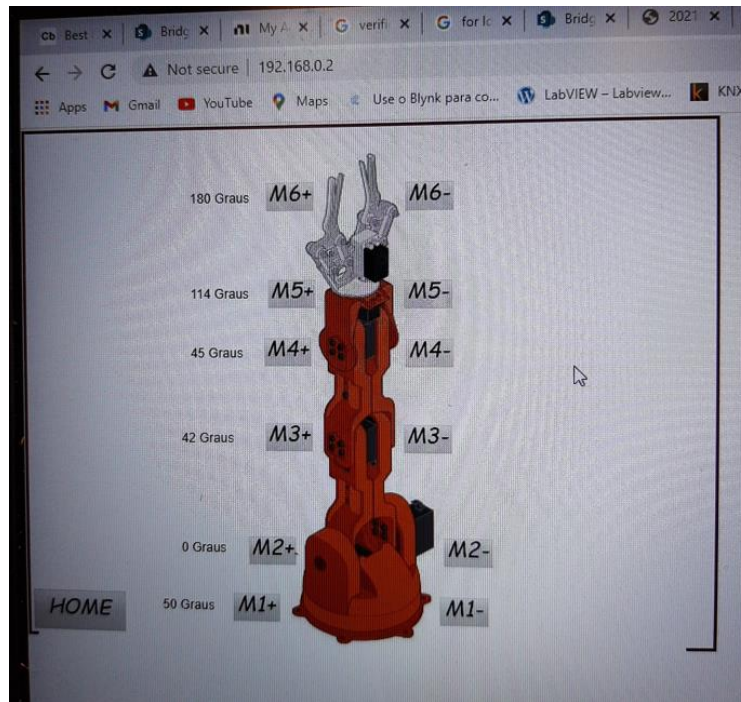


Figura 60 - Menu Robot Teach

Se o botão “Home” for pressionado, esta página será fechada automaticamente e retornará a página do menu principal onde o utilizador poderá ainda escolher entre as duas outras páginas restantes. No caso da página “Lista de pontos”, aparecerá o que está representado na Figura 61.

Sistema de um tapete rolante com ligação a supervisão

**TABELA DE PONTOS - ROBOT**

	M1	M2	M3	M4	M5	M6
P#1	50°	0°	96°	90°	114°	90°
P#2	50°	0°	42°	90°	114°	90°
P#3	50°	0°	42°	45°	114°	180°
P#4	50°	0°	96°	180°	114°	0°

HOME

Página de 1 de 2

NEXT->

Figura 61 - Menu Lista de Pontos - Parte I

Aqui o utilizador pode manualmente inserir os pontos desejados, estando apenas limitado em no máximo 6 pontos. Se clicar no botão “Home” retornará a página do menu principal e se clicar em “Next” abrirá a página com a continuação da lista de pontos, como se pode observar na Figura 62.

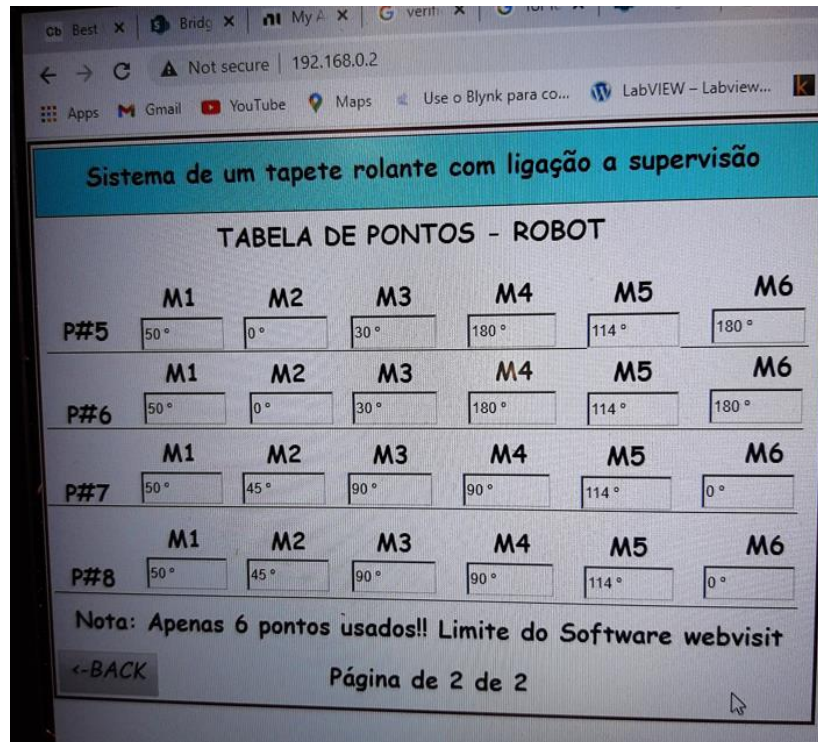


Figura 62 -Menu Lista de Pontos - Parte II

As coordenadas das juntas que forem inseridas no ponto P#6 serão automaticamente refletidas nos restantes pontos 7 e 8.

Para terminar, existe a última página onde utilizador pode efetuar a ordem de arranque/paragem do sistema, uma vez que os pontos do robô já estejam definidos. A representa a página “AUTO”.

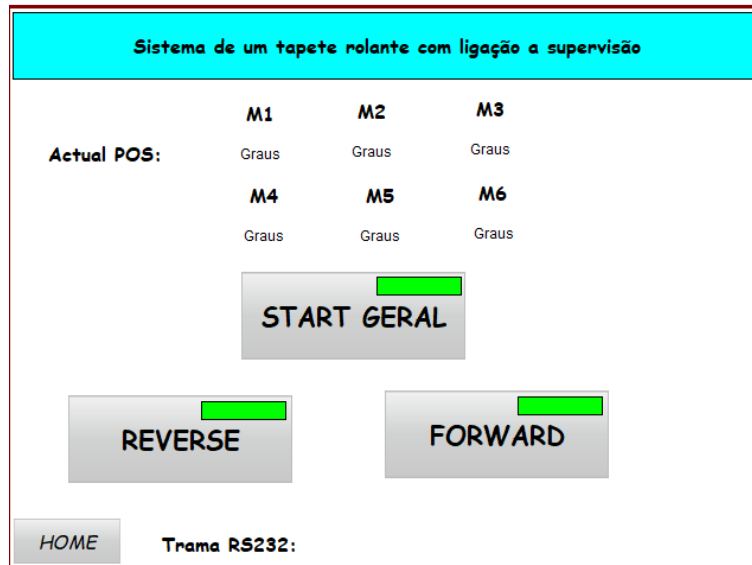


Figura 63 - Menu AUTO



## ***Capítulo 4***

### ***Resultados experimentais***

**Resumo:** O presente capítulo apresenta os dados recolhidos na comunicação RS232 entre o PLC e o Arduino, através da porta COM do Arduino, bem como as formas de onda de tensão PWM aplicada ao circuito de controlo do robô.



## 4 Resultados experimentais

Os resultados experimentais deste trabalho consistem na apresentação da informação que é enviada pelo PLC ao Arduíno, dos sinais de comando para enviar o robô a um ponto pré-definido pelo utilizador. No caso dos sinais de comando, está informação foi recolhida com recurso a um osciloscópio digital que existe no laboratório de automação e robótica do ISEL. Quanto a trama enviada pelo PLC, foi usada a própria porta COM do Arduíno para extrair e apresentar esta informação. As secções seguintes contêm para cada ponto definido para o robô as seguintes imagens:

- Página da aplicação da supervisão, indicando para que coordenadas de junta está o robô a ser enviado
- Porta série do Arduíno com a informação da comunicação RS232, formatada de acordo a Tabela 7 - Trama RS232 entre PLC e o
- Formato dos dados no PLC provenientes da aplicação de supervisão
- Imagem real da localização do robô
- Forma de onda PWM de cada junta do robô

### 4.1 Ponto 1 (Home)

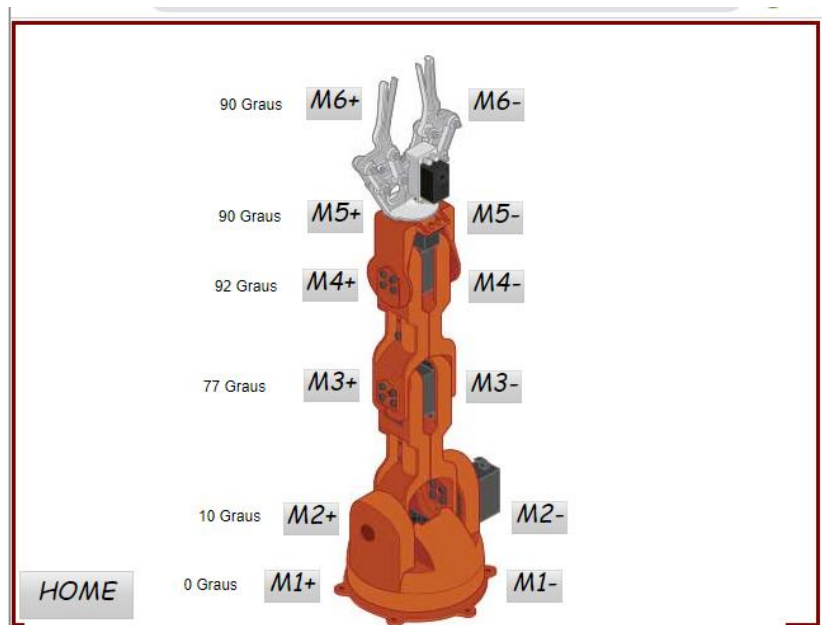


Figura 64 - Ponto 1(Home) HMI

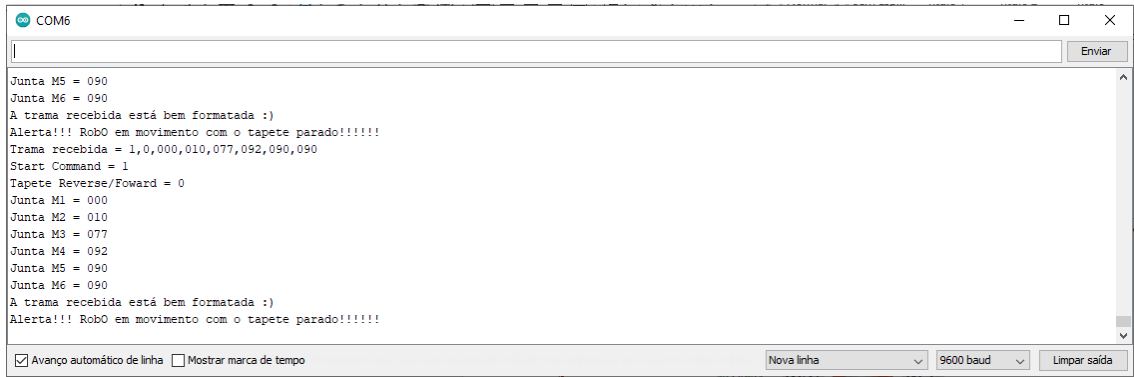


Figura 65 - Trama RS232 PLC -> Arduino Ponto 1

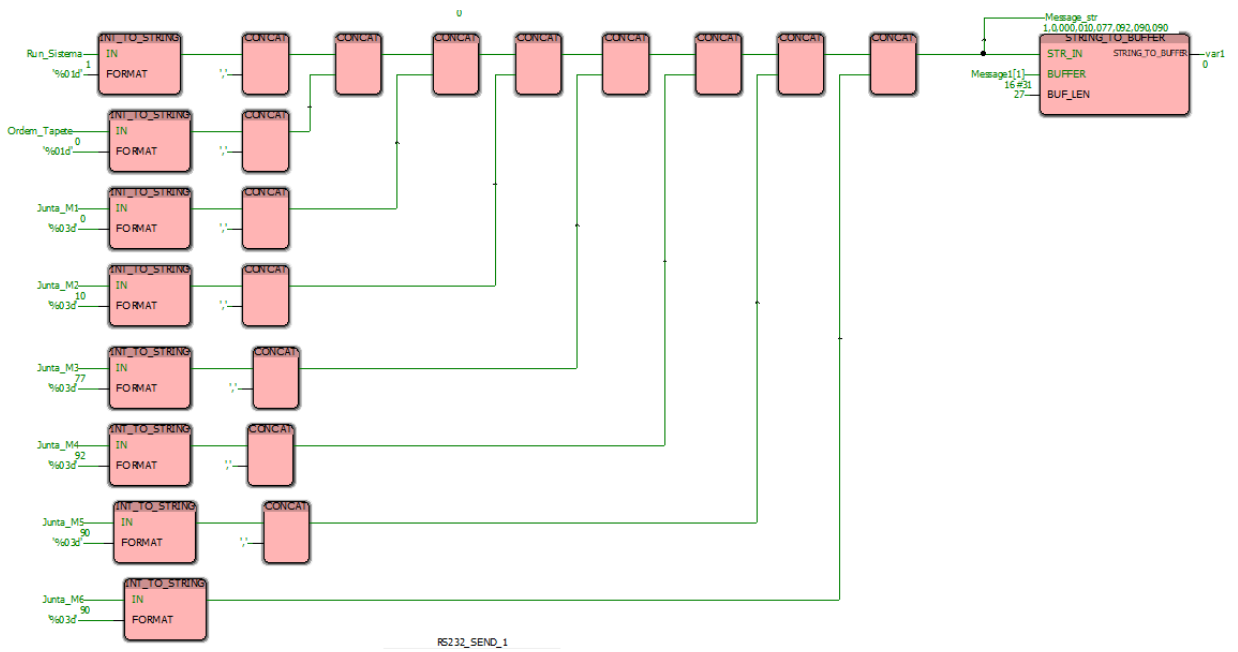


Figura 66 - Formato da Trama PLC Ponto 1

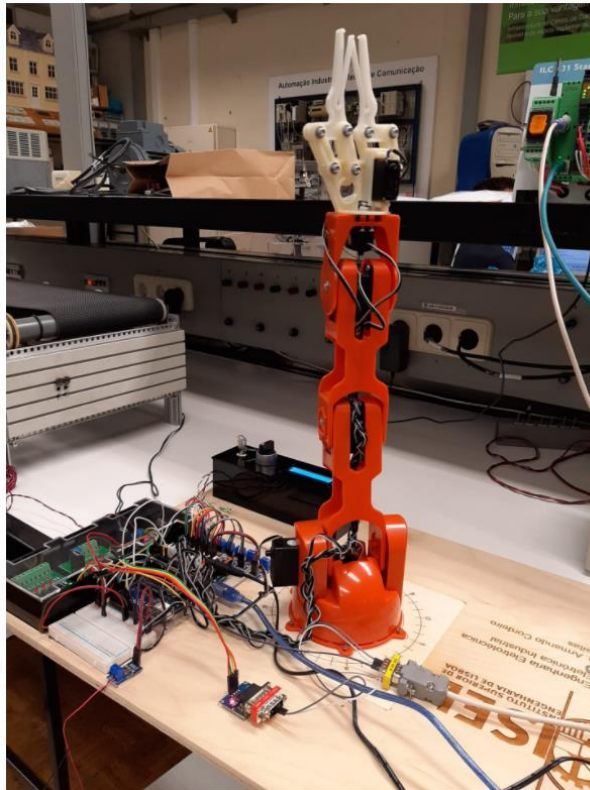
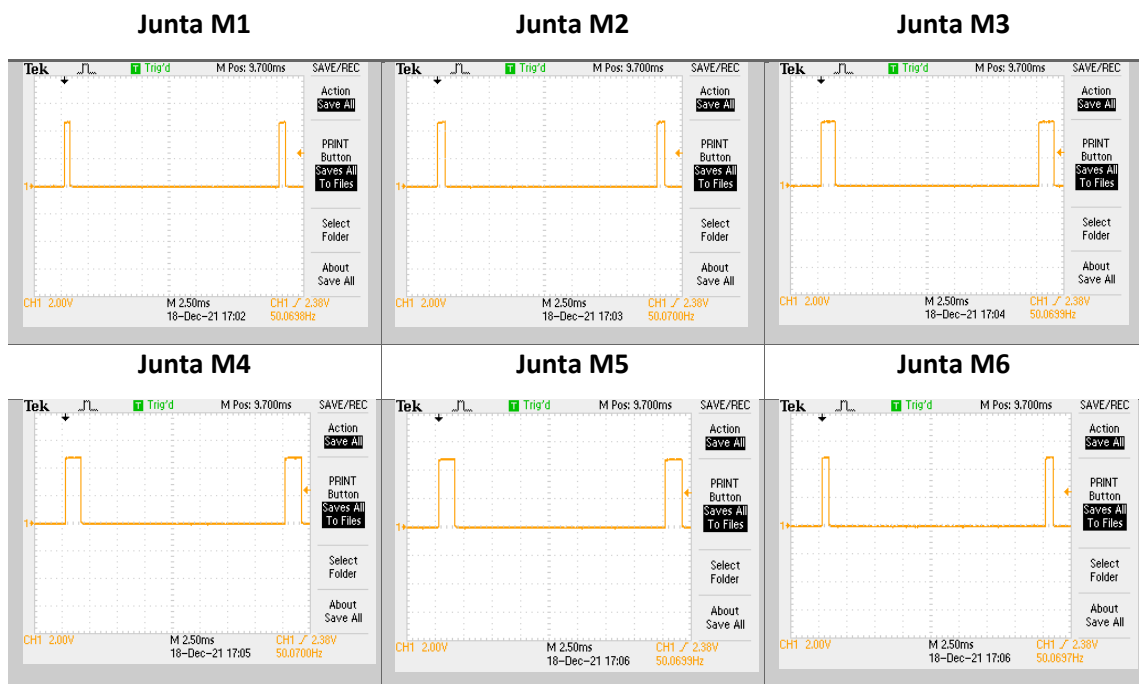


Figura 67 - Ponto 1 (Home) Real

Tabela 9 - Sinais PWM Robô Ponto 1(Home)





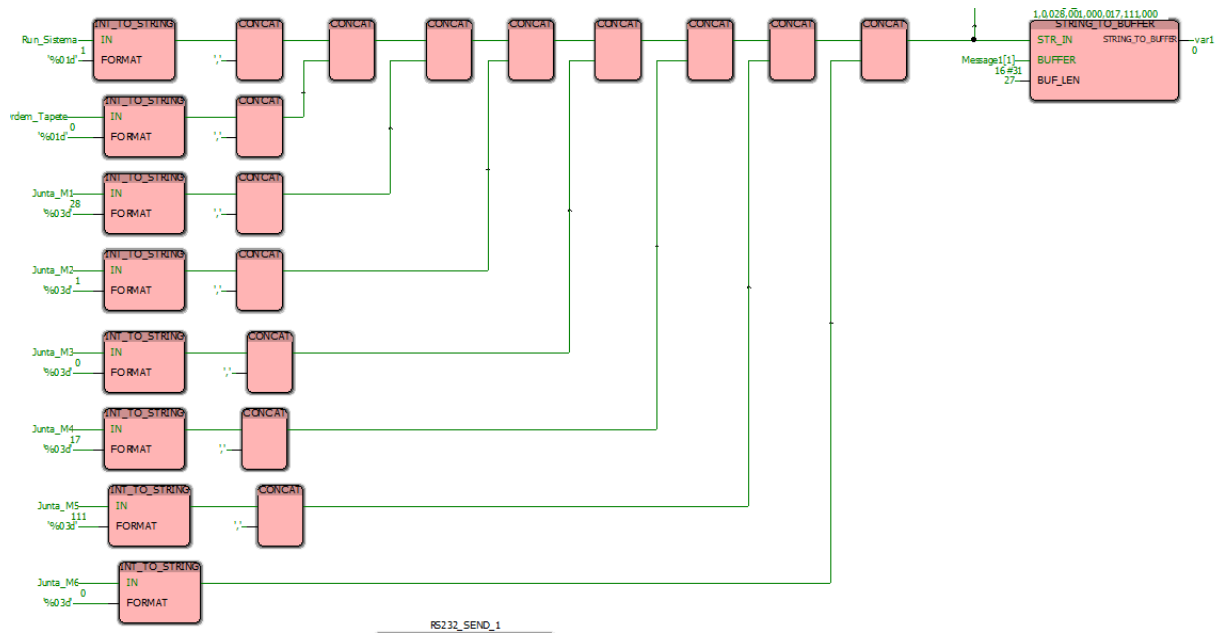


Figura 70 - Formato da Trama PLC Ponto 2

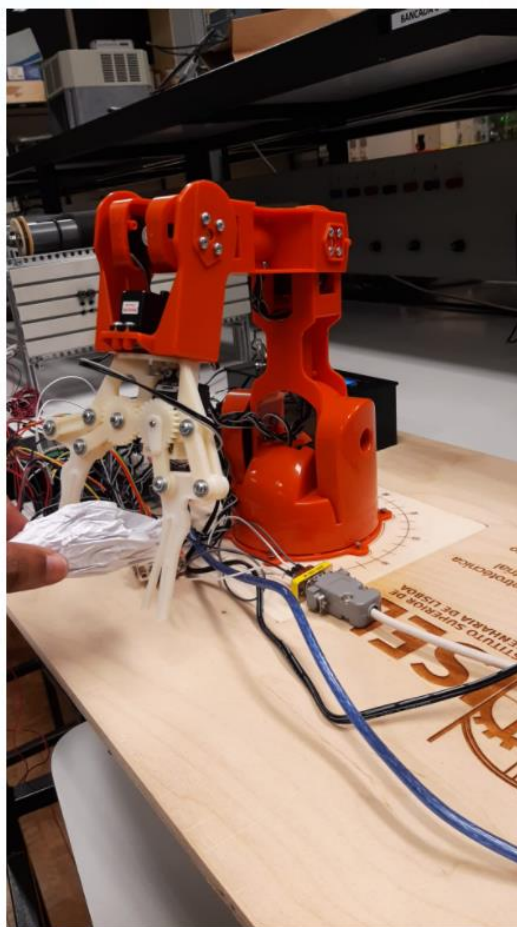
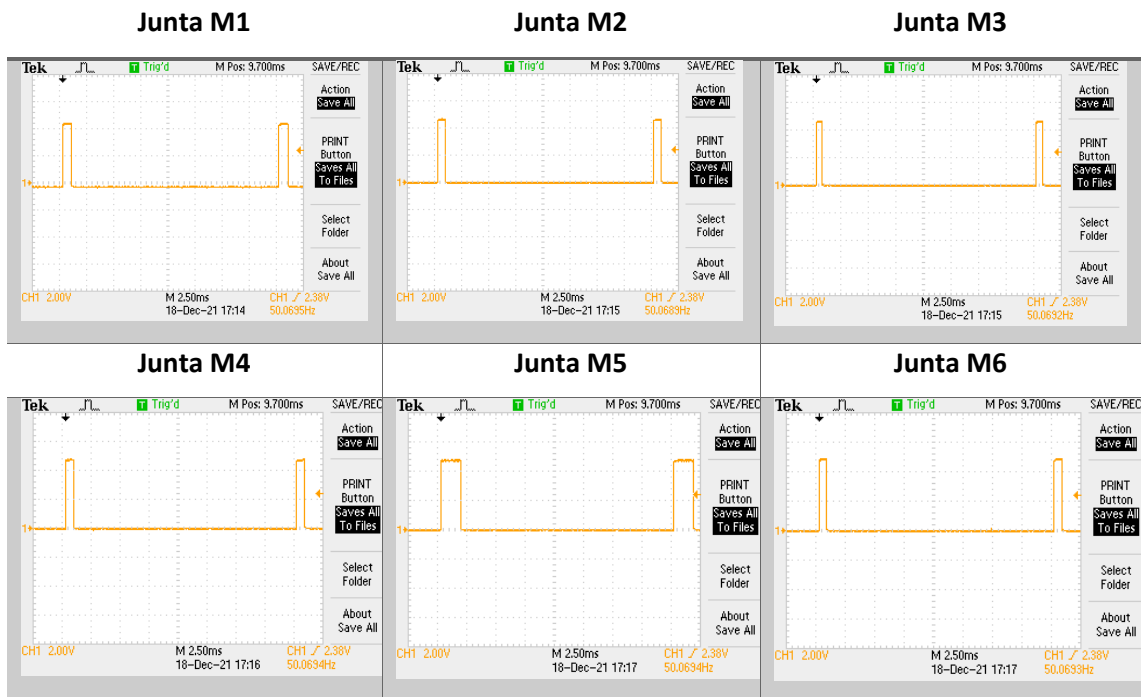


Figura 71 - Ponto 2 (Pick) Real

Tabela 10 - Sinais PWM Robô Ponto 2(Pick)



### 4.3 Ponto 3 (Fechar a garra)

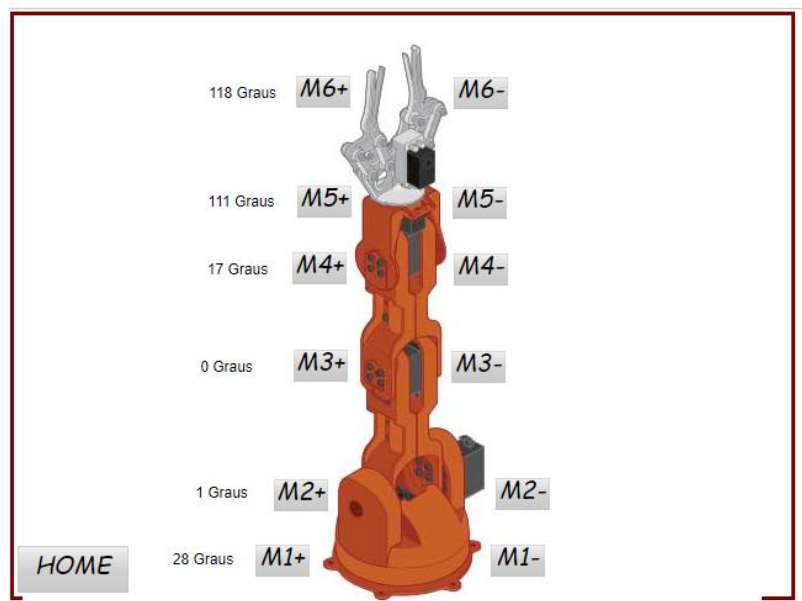


Figura 72 - Ponto 3(Fechar a garra) HMI

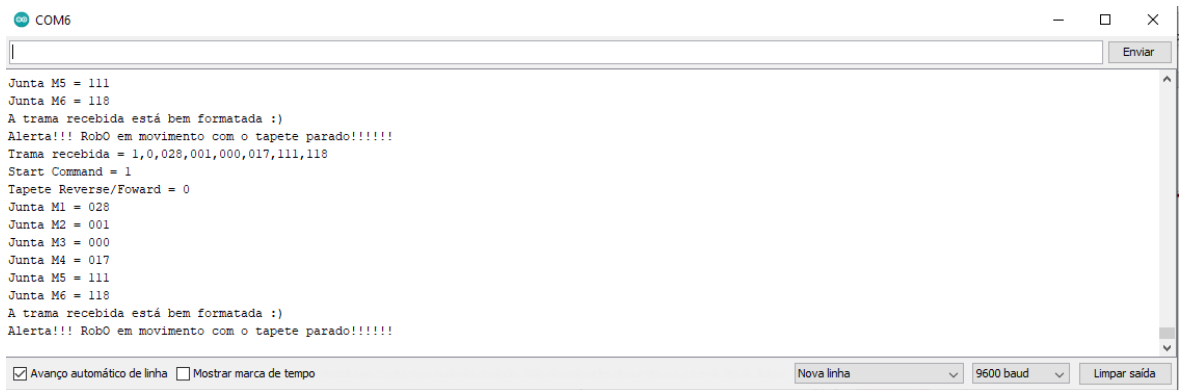


Figura 73 - Trama RS232 PLC -> Arduíno Ponto 3

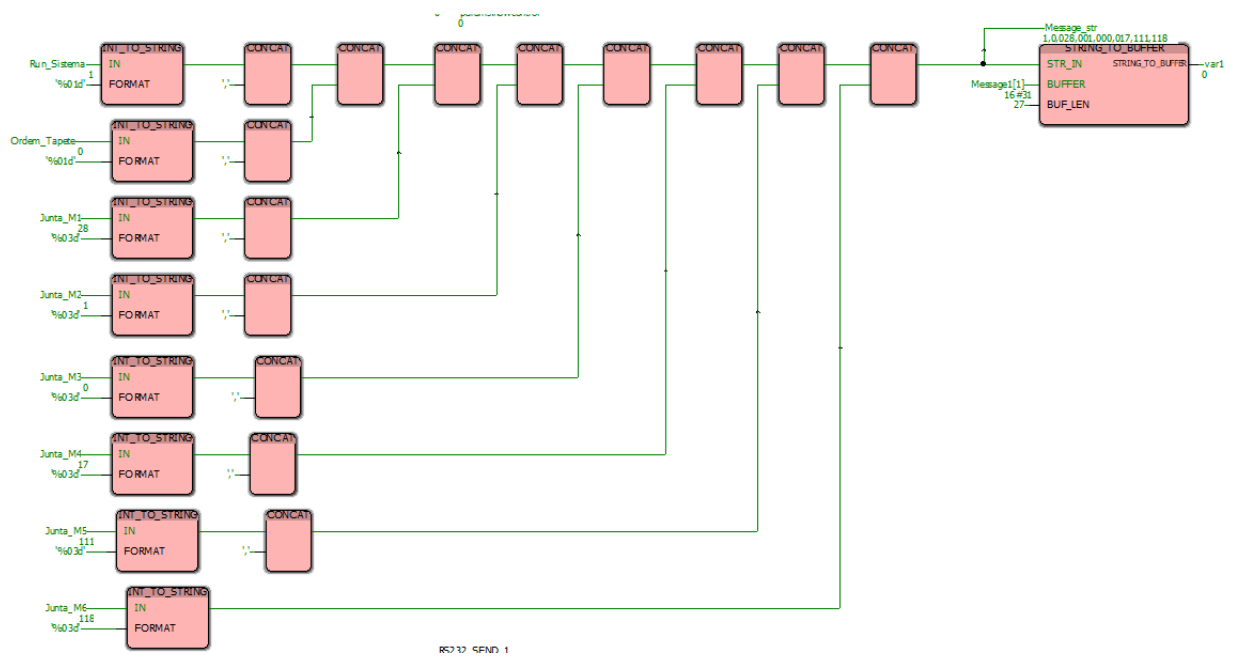


Figura 74 - Formato da Trama PLC Ponto 3

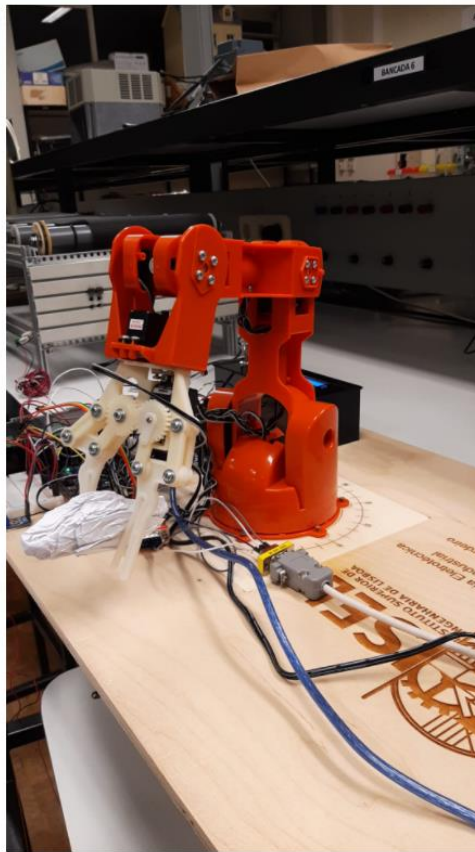
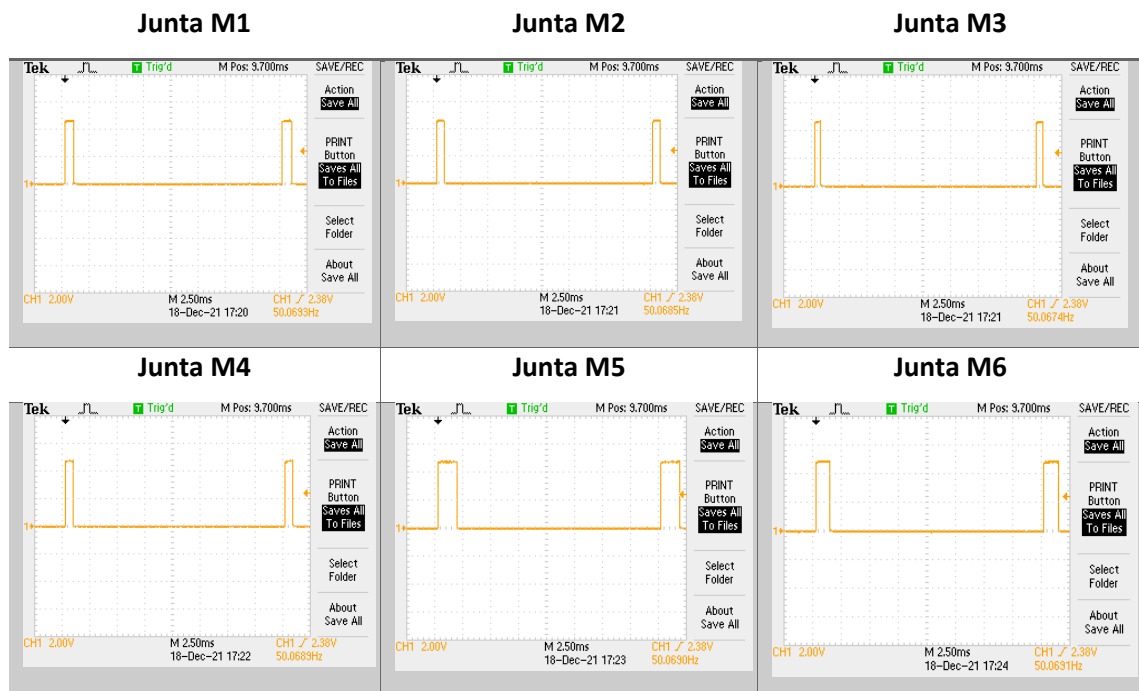


Figura 75 - Ponto 3 (Fechar a garra) Real

Tabela 11 - Sinais PWM Robô Ponto 3(Fechar a garra)



#### 4.4 Ponto 4 (Place)

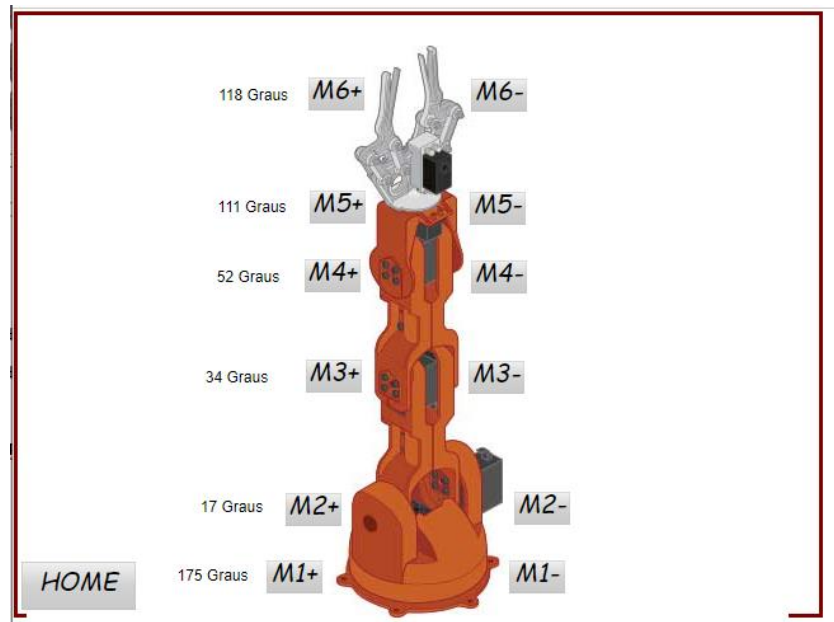


Figura 76 - Ponto 4(Place) HMI

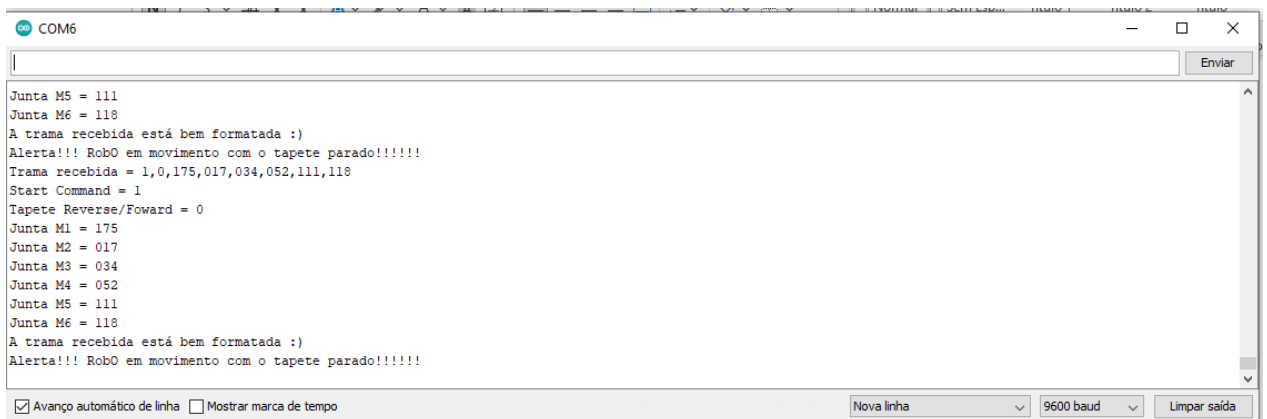


Figura 77 - Trama RS232 PLC -> Arduíno Ponto 4

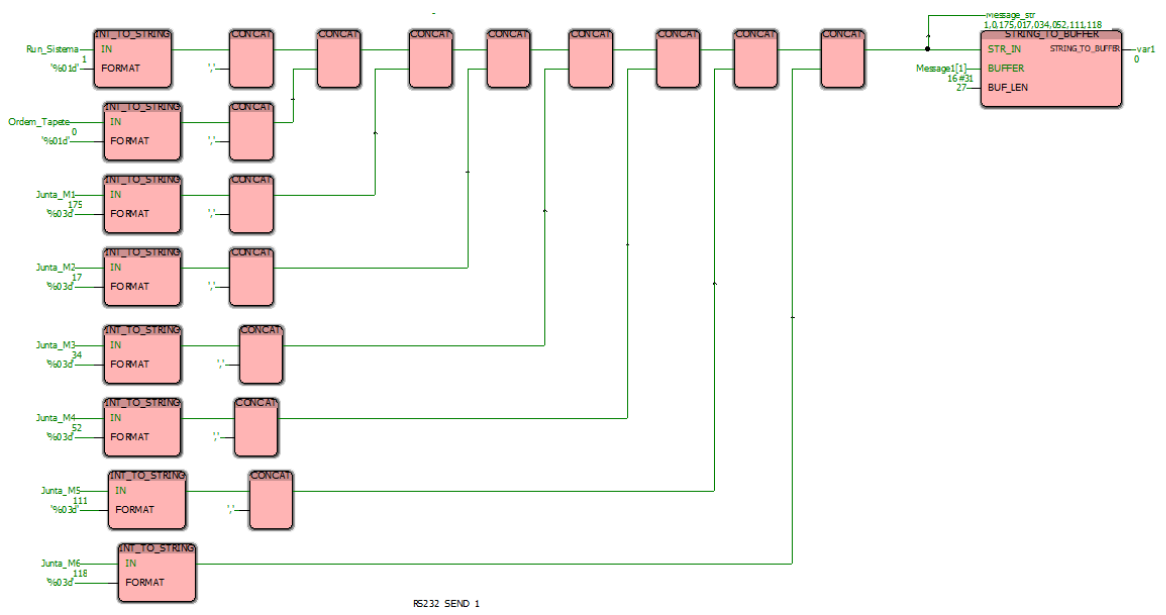
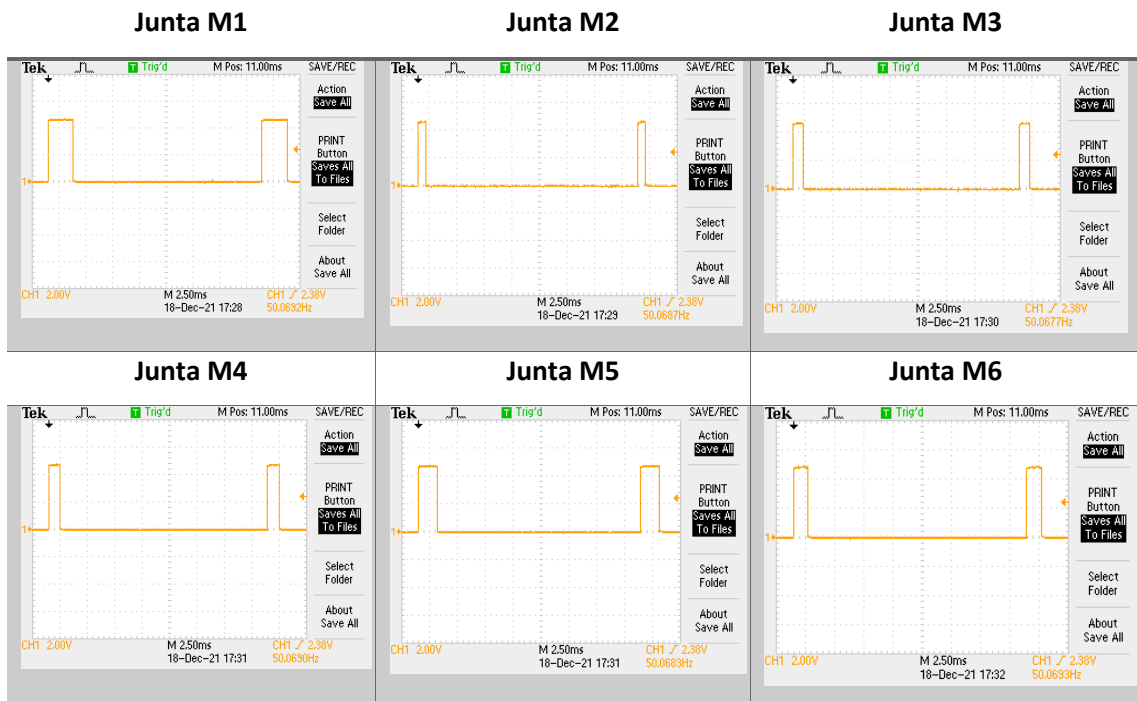


Figura 78 - Formato da Trama PLC Ponto 4



Figura 79 - Ponto 4 (Place) Real

Tabela 12 - Sinais PWM Robô Ponto 4(Place)



#### 4.5 Ponto 5 (Abrir a garra)

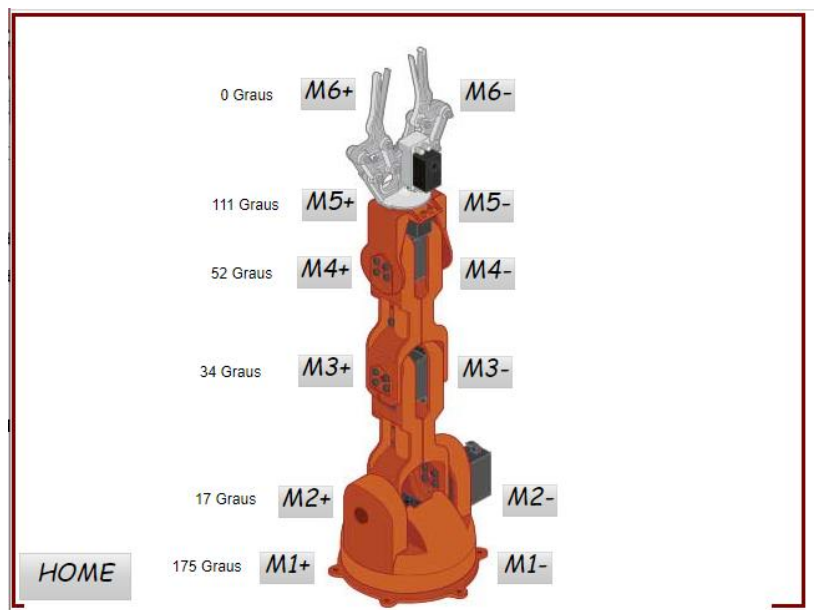


Figura 80 - Ponto 5(Abrir a garra) HMI



Figura 81 - Trama RS232 PLC -> Arduíno Ponto 5

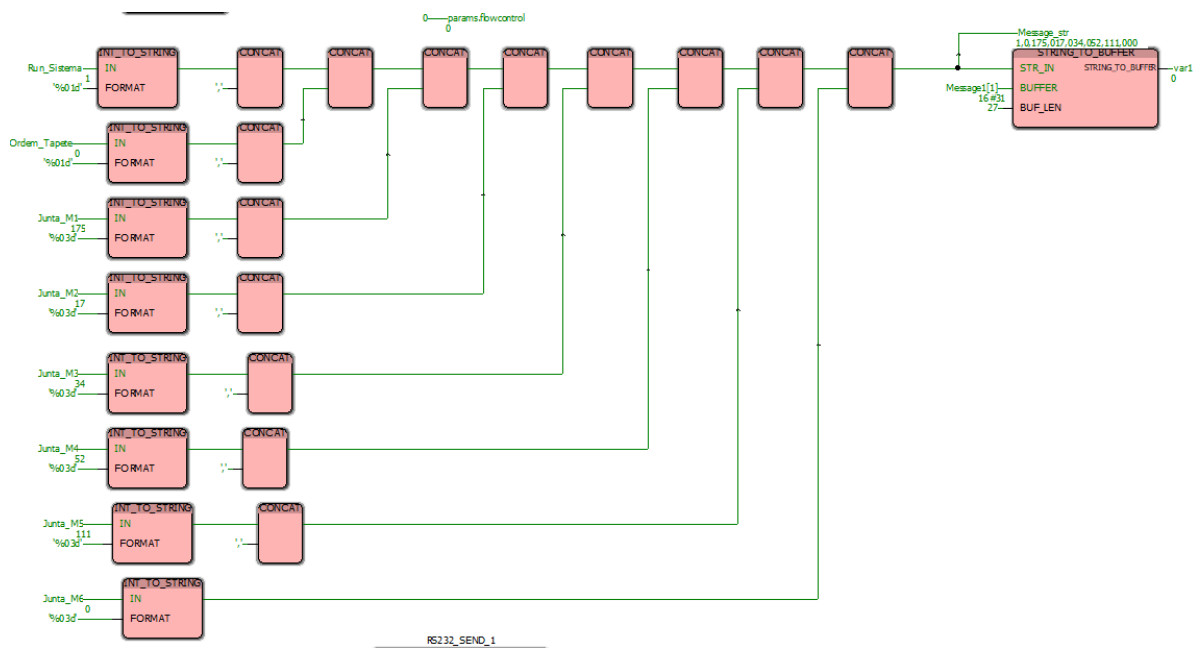


Figura 82 - Formato da Trama PLC Ponto 5

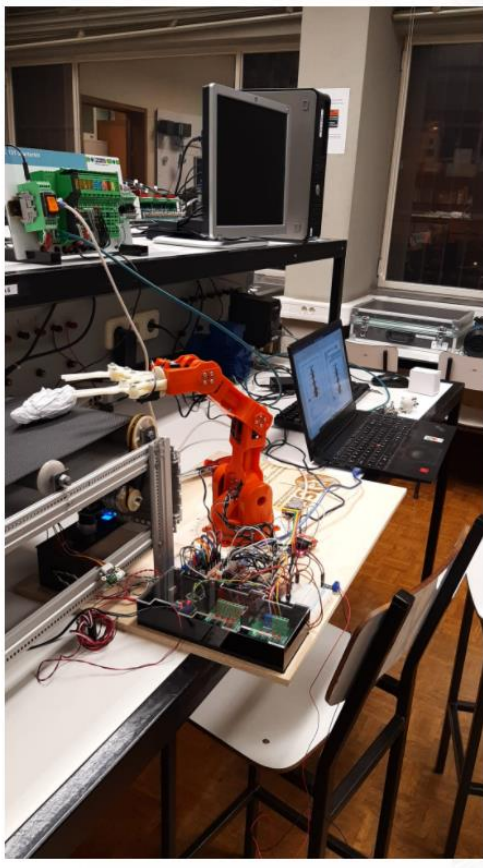
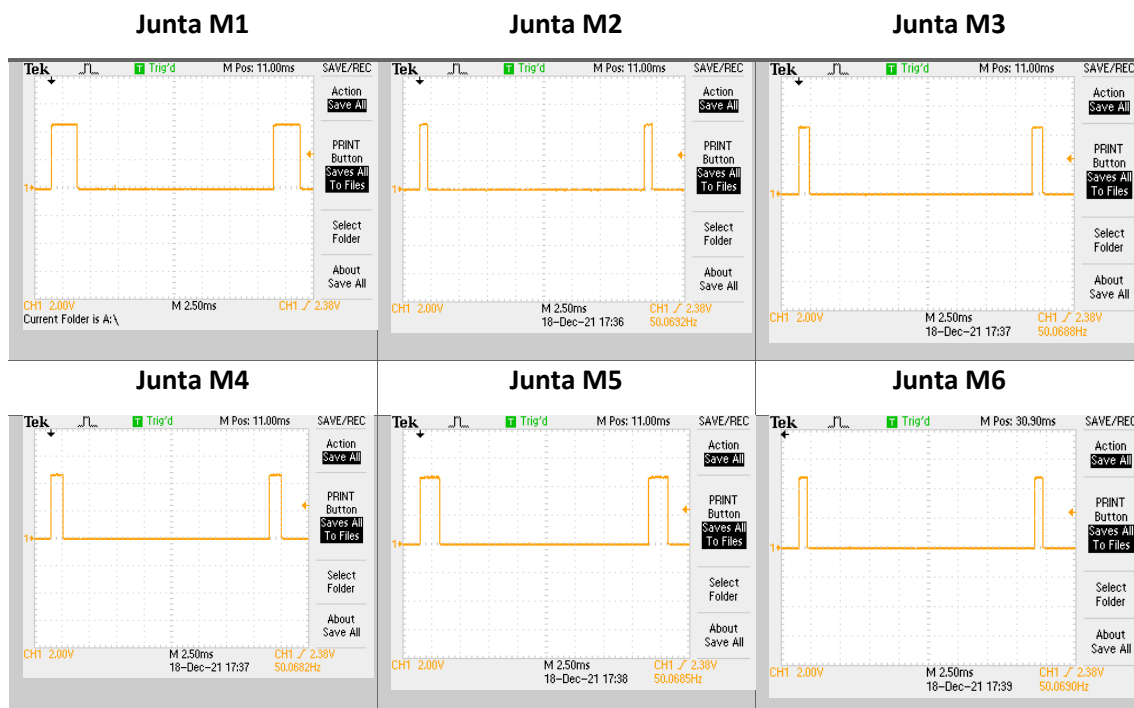


Figura 83 - Ponto 5 (Abrir a garra) Real

Tabela 13 - Sinais PWM Robô Ponto 5(Abrir a garra)





## ***Capítulo 5***

### ***Conclusões e perspectivas de trabalho futuro***

**Resumo:** O presente capítulo apresenta de forma resumida as vantagens da realização deste trabalho, os pontos fracos e as propostas de melhorias futuras.



## 5 Conclusões

Com a realização deste trabalho foi possível pôr em prática os conhecimentos adquiridos nas unidades curriculares de Automação, Redes e Supervisão, Conversores Eletrónicos, Eletrónica de Potência e Robótica. Foi possível também ter contacto direto com a comunicação RS232 e perceber os detalhes das ligações, bem como a programação do microcontrolador Arduino e o controlador industrial ILC 131 ETH da Phoenix Contact.

Com a realização deste trabalho, o utilizador passa a ser capaz de ensinar 8 pontos ao robô, como por exemplo um para a posição de *home*, um para a posição de *pick*, um para a aproximação de *pick*, um para pegar a peça, um para a posição de *place*, um para a aproximação de *place* e por fim um para largar a peça. O utilizador também é capaz de controlar a velocidade e sentido de rotação do tapete rolante, tudo isto a partir do sistema de supervisão desenvolvido.

A versão *express* do *software* WebVisit permite apenas que sejam partilhadas no máximo 60 variáveis e isto limita a possibilidade de criação de mais pontos para o robô.

### 5.1 Perspetivas de trabalho futuro

Como este trabalho foi desenvolvido para fins didáticos para os alunos de Licenciatura do ISEL, existem muitos pontos que podem ser melhorados pelos mesmos. No caso do tapete rolante por exemplo, a estrutura mecânica e a tela podem ser substituídas por outras que tenham um melhor aspeto visual.

O Arduino responsável pelo controlo do tapete rolante pode ser eliminado e o controlo pode passar diretamente para o PLC e assim pode-se adicionar sensores magnéticos ou sensores de fibra ótica para detetar a presença de peça no início e no fim do tapete. Pode-se também adicionar um pequeno sistema de visão que calcula a posição de *pick* ou de *place* da peça e enviar as coordenadas de juntas para o robô.

Caso não haja o objetivo de melhoria deste trabalho, ainda assim pode servir como proposta de projeto para os alunos da Licenciatura, onde eles não tenham acesso aos *softwares* e desenvolvam por eles mesmo com base apenas no *hardware*.

## 6 Referências

- [1] A. M. Lopes, Modelação Cinemática e Dinâmica de Manipuladores de Estrutura em Série, Porto, 2001/2002.
- [2] F. N. J. G. Almeida, Sistemas Robóticos, Instituto Superior de Engenharia de Lisboa, 2008.
- [3] B. Solutions, “<http://www.bpsolutions.com.br/aprenda-funcionamento-esteira-industrial/>,” 1994. [Online]. [Acedido em Julho 2019].
- [4] OMRON, “<https://industrial.omron.pt/pt/solutions/packaging/packaging-machine-automation-solutions/robotic-infeed-module/>,” [Online]. [Acedido em Julho 2019].
- [5] Setembro 2019. [Online]. Available: <https://www.citisystems.com.br/motor-cc/>.
- [6] P. E. D. Foito, Eletrónica de Potência - Conversores\_CC\_CC, IPS\_Est Setúbal, 2017.
- [7] L. H. M. d. Castro, O Uso do Arduino e do Processing no Ensino de Física, Rio de Janeiro, 2016.
- [8] J. Palma, Introdução às Redes de Campo de Automação - ISEL, Chelas,Lisboa, 2004.
- [9] “[https://pt.rs-online.com/web/p/shields-para-arduino/1113738/?cm\\_mmc=PT-PLA-DS3A-\\_google\\_-\\_CSS\\_PT\\_PT\\_Raspberry\\_Pi\\_%26\\_Arduino\\_y\\_M%C3%B3dulos\\_de\\_Desarrollo\\_Whoop\\_-\\_%28PT:Whoop!%29+Shields+para+Arduino\\_-\\_1113738&matchtype=&aud-772940708159:pla-306242761339&gclid=...](https://pt.rs-online.com/web/p/shields-para-arduino/1113738/?cm_mmc=PT-PLA-DS3A-_google_-_CSS_PT_PT_Raspberry_Pi_%26_Arduino_y_M%C3%B3dulos_de_Desarrollo_Whoop_-_%28PT:Whoop!%29+Shields+para+Arduino_-_1113738&matchtype=&aud-772940708159:pla-306242761339&gclid=...),” Novembro 2021. [Online].
- [10] “<https://pt.wikipedia.org/wiki/Servomotor>,” Setembro 2019. [Online].
- [11] P. P. S. C. Lino, “Polias, Correias e Transmissão de Potência,” 2013.
- [12] “Botnroll,” Setembro 2019. [Online]. Available: [https://www.botnroll.com/pt/motores-dc/430-motor-12v-com-encoder-146rpm.html?search\\_query=motor&results=239](https://www.botnroll.com/pt/motores-dc/430-motor-12v-com-encoder-146rpm.html?search_query=motor&results=239).
- [13] “Wiki L298N,” [Online]. Available: [http://wiki.sunfounder.cc/index.php?title=Motor\\_Driver\\_Module-L298N](http://wiki.sunfounder.cc/index.php?title=Motor_Driver_Module-L298N). [Acedido em Dezembro 2021].
- [14] Phoenix. Contact, WebVisit 6.01.02, 2012.
- [15] M. Coorporation, “Página do Office,” [Online]. Available: [www.office.com](http://www.office.com). [Acedido em 04 04 2017].
- [17] P. M. B. Moreira, Sistema de Supervisão e Controlo de Estruturas, Porto, Fevereiro de 2011.



# Anexo A

```

//sketch created by Santo Freitas para o control do tapete
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
int modo_manual = 8;
int marchaTapete_Left=2;
int marchaTapete_Right=4;
int Controlo_robot_1=9; // Sinal do PLC para fazer toogle ao controlo dos eixos do robô
int Controlo_robot_2=10; // Sinal do PLC para fazer toogle ao controlo dos eixos do robô

/*
Controlo de velocidade de motor DC com PWM
Autores: Santo Freitas
*/
void arranque_motor_sentido_direto(int velocidade_direta);
void arranque_motor_sentido_inverso(int velocidade_inversa);
void paragem_motor();
#define IN1_PWM 3 //pino do Arduino que terá a ligação para o driver de motor (ponte H) L298N)
#define IN2_PWM 5 //pino do Arduino que terá a ligação para o driver de motor (ponte H) L298N)
float valorR1      = 30000.0; //VALOR DO RESISTOR 1 DO DIVISOR DE TENSÃO
float valorR2      = 7500.0; // VALOR DO RESISTOR 2 DO DIVISOR DE TENSÃO

```

```

//#define TEMPO_RAMPA 2000
const int Regulador_Velocidade = A0;
float tensaoEntrada      = 0.0; //VARIÁVEL PARA ARMAZENAR O VALOR DE TENSÃO DE ENTRADA DO SENSOR
float tensaoMedida      = 0.0; //VARIÁVEL PARA ARMAZENAR O VALOR DA TENSÃO MEDIDA PELO SENSOR
int leituraSensor       = 0; //VARIÁVEL PARA ARMAZENAR A LEITURA DO PINO ANALÓGICO
int VELOCIDADE = 0;
//int RAMPA_SUBIDA=0 ; // definir a rampa de aceleração

void setup()
{
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Inicializando...");
  lcd.setCursor(0,1);
  delay(1000);
  lcd.print("Definindo I/O...");
  delay(1000);
  lcd.clear();
  pinMode (modo_manual,INPUT_PULLUP);

```

```

pinMode (marchaTapete_Left,INPUT_PULLUP);
pinMode (marchaTapete_Right,INPUT_PULLUP);
pinMode (Controlo_robot_1,INPUT_PULLUP);
pinMode (Controlo_robot_2,INPUT_PULLUP);
pinMode(IN1_PWM, OUTPUT);//configura como saída pino terá a ligação para o driver de motor (ponte H) L298N)
pinMode(IN2_PWM, OUTPUT);//configura como saída pino terá a ligação para o driver de motor (ponte H) L298N)
pinMode(Regulador_Velocidade, INPUT);
Serial.begin(9600);
Serial.println("Abrir Comunicação");

}

void loop()
{
bool Read_robot_1      = digitalRead(Controlo_robot_1);
bool Read_robot_2      = digitalRead(Controlo_robot_2);
leituraSensor          = analogRead(Regulador_Velocidade ); //FAZ A LEITURA DO PINO ANALÓGICO E ARMAZENA NA VARIÁVEL O VALOR LIDO
tensaoEntrada          = (leituraSensor * 5.0) / 1024.0; //VARIÁVEL RECEBE O RESULTADO DO CÁLCULO
tensaoMedida           = tensaoEntrada / (valorR2/(valorR1+valorR2)); //VARIÁVEL RECEBE O VALOR DE TENSÃO DC MEDIDA PELO SENSOR
VELOCIDADE             = map( tensaoMedida,0,10,0,255);
lcd.setCursor(0,0);

```

```
lcd.print("Veloc. Tapete");
```

```
lcd.setCursor(0,1);
```

```
lcd.print(VELOCIDADE);
```

```
lcd.setCursor(3,1);
```

```
lcd.print("%");
```

```
delay(1000);
```

```
lcd.clear();
```

```
if ((Read_robot_1== HIGH and Read_robot_2== HIGH)or (Read_robot_1== LOW and Read_robot_2== LOW)) {
```

```
  paragem_motor();
```

```
  lcd.setCursor(0,0);
```

```
  lcd.print("Rs232 nOK or Stop");
```

```
  lcd.setCursor(0,1);
```

```
  lcd.print("Tapete OFF");
```

```
  delay(500);
```

```
  lcd.clear();
```

```
  Serial.println("Entrou em ambos highs or Lows : Tapete parado! Ou RS232 não está OK"); //IMPRIME O TEXTO NA SERIAL
```

```
}
```

```
if (Read_robot_1== LOW and Read_robot_2 == HIGH){
```

```
  arranque_motor_sentido_direto(VELOCIDADE);
```

```
  lcd.setCursor(0,0);
```

```
lcd.print("Tapete Foward");
lcd.setCursor(0,1);
lcd.print("Good Work :)");
delay(500);
lcd.clear();
Serial.println("Entrou em um low e um High : Tapete em modo foward"); //IMPRIME O TEXTO NA SERIAL
}
if (Read_robot_1== HIGH and Read_robot_2 == LOW){
  arranque_motor_sentido_inverso(VELOCIDADE);
  lcd.setCursor(0,0);
  lcd.print("Tapete Reverse");
  lcd.setCursor(0,1);
  lcd.print("Good Work :)");
  delay(500);
  lcd.clear();
  Serial.println("Entrou em um low e um High : Tapete em modo reverse"); //IMPRIME O TEXTO NA SERIAL
}
}
```

```
void arranque_motor_sentido_direto(int velocidade_direta)
{
    analogWrite(IN1_PWM, 0);
    analogWrite(IN2_PWM, velocidade_direta);
}
void arranque_motor_sentido_inverso(int velocidade_inversa)
{
    analogWrite(IN1_PWM, velocidade_inversa);
    analogWrite(IN2_PWM, 0);
}
void paragem_motor()
{
    analogWrite(IN1_PWM, 0);
    analogWrite(IN2_PWM, 0);
}
```

```
//Algoritmo para o controlo do robô por RS232
#include <Braccio.h>
#include <Servo.h>
#include <String.h>

Servo base; // Estrutura para controlo da Junta M1
Servo shoulder; // Estrutura para controlo da Junta M2
Servo elbow ; // Estrutura para controlo da Junta M3
Servo wrist_ver; // Estrutura para controlo da Junta M4
Servo wrist_rot; // Estrutura para controlo da Junta M5
Servo gripper; // Estrutura para controlo da Junta M6
String PLC_CMD; //Variavel para receber a trama RS232 vinda do PLC ILC 131 ETH da Phoenix Contact
int Tapete_1 = 4; // Sinal elétrico enviado ao arduino que faz o controlo do tapete rolante
int Tapete_2 = 7; // Sinal elétrico enviado ao arduino que faz o controlo do tapete rolante
int start=0,rotacao=0,M1=90,M2=90,M3=90,M4=90,M5=90,M6=90,M3_security=0; // Variáveis inteiras para auxiliar no controlo do sistema
String getValue(String data, char separator, int index); // Protótipo da função que efetua split a trama recebida via RS232
```

```

void setup() {
  pinMode(Tapete_1,OUTPUT); // Define pino 4 como entrada
  pinMode(Tapete_2,OUTPUT); // Define pino 7 como entrada
  Braccio.begin(); // Função que manda o robô para uma posição de setup inicial -> Obrigatório para habilitar os 6 eixos do robô
  Serial.begin(9600); // Inicializa a comunicação serial
  digitalWrite( Tapete_1,HIGH); // Ativa para indicar que a comunicação entre o PCL e o arduino ainda não foi estabelecida
  digitalWrite(Tapete_2,HIGH); // Ativa para indicar que a comunicação entre o PCL e o arduino ainda não foi estabelecida
}

```

```

void loop() {

```

```

  //Serial.println("A Aguardar comunicação RS232 estabelecida entre o arduino e o PLC ");

```

```

  while (Serial.available())

```

```

  {

```

```

    // Aguarda que seja recebido algum caracter via porta série

```

```

    //Formato que se está a espera de receber ://

```

```

    /*      start,sentidoTapete,M1,M2,M3,M4,M5,M6      */

```

```

    /*          0,0,090,090,090,090,090          */

```

```
/* start = 0 : Indica parar o sistema (Tapete+Robô)
 * start = 1 : Arrancar o sistema
 * sentido = 0 : Parar o motor do tapete
 * sentido = 1 : Rodar o motor do tapete no sentido direto
 * sentido = 2 : Rodar o motor do tapete no sentido inverso
 * M1 a M6 = Valores de coordenada do robô
```

PLC da Phoenix Contact ILC 131 ETH - RS232

```
*/
```

```
PLC_CMD = Serial.readString(); // Recebe e armazena a trama RS232 na variavel PLC_CMD
Serial.print("Trama recebida = ");
Serial.println(PLC_CMD);
String CMD1 = getValue(PLC_CMD,',',0); // Armazena os caracteres
start = CMD1.toInt();
Serial.print("Start Command = ");
Serial.println(CMD1);
String CMD2 = getValue(PLC_CMD,',',1);
rotacao = CMD2.toInt();
Serial.print("Tapete Reverse/Foward = ");
```

```
Serial.println(CMD2);
String CMD3 = getValue(PLC_CMD,',',2);
M1 = CMD3.toInt();
Serial.print("Junta M1 = ");
Serial.println(CMD3);
String CMD4 = getValue(PLC_CMD,',',3);
M2 = CMD4.toInt();
Serial.print("Junta M2 = ");
Serial.println(CMD4);
String CMD5 = getValue(PLC_CMD,',',4);
M3 = CMD5.toInt();
Serial.print("Junta M3 = ");
Serial.println(CMD5);
String CMD6 = getValue(PLC_CMD,',',5);
M4 = CMD6.toInt();
Serial.print("Junta M4 = ");
Serial.println(CMD6);
String CMD7 = getValue(PLC_CMD,',',6);
M5 = CMD7.toInt();
Serial.print("Junta M5 = ");
Serial.println(CMD7);
```

```
String CMD8 = getValue(PLC_CMD,',',7);
```

```
M6 = CMD8.toInt();
```

```
Serial.print("Junta M6 = ");
```

```
Serial.println(CMD8);
```

```
if ( start>=0 && start<2 && rotacao>=0 && rotacao<3 && M1>=0 && M1<=180 && M2>=0 && M2<=180 && M3>=0 && M3<=180 && M4>=0  
&& M4<=180 && M5>=0 && M5<=180 && M6>=0 && M6<=180){
```

```
Serial.println("A trama recebida está bem formatada :) ");
```

```
if(start==1){
```

```
Braccio.ServoMovement(20, M1, M2, M3, M4, M5, M6);
```

```
if ( rotacao==0){
```

```
digitalWrite( Tapete_1,LOW);
```

```
digitalWrite(Tapete_2,LOW);
```

```
Serial.println("Alerta!!! RobO em movimento com o tapete parado!!!!!! ");
```

```
    } else if ( rotacao==1 ){
```

```
        digitalWrite( Tapete_1,HIGH);
```

```
        digitalWrite(Tapete_2,LOW);
```

```
        Serial.println(" RobO em movimento com o tapete em sentido direto :) ");
```

```
    } else {
```

```
        digitalWrite( Tapete_1,LOW);
```

```
        digitalWrite(Tapete_2,HIGH);
```

```
Serial.println(" RobO em movimento com o tapete em sentido inverso :) ");  
}
```

```
}else{
```

```
  Braccio.ServoMovement(20, 60, 10, 90, 90, 90, 90); // Função de movimento do robô
```

```
  digitalWrite( Tapete_1,LOW);
```

```
  digitalWrite(Tapete_2,LOW);
```

```
  Serial.println(" Sistema Parado! Aguarda Start do PLC ");
```

```
}
```

```
}else Serial.println("A trama recebida não está bem bem formatada ou recebeu valores não  
permitidos :( ");
```

```
}
```

```
}
```

```
String getValue(String data, char separator, int index)
```

```
{
```

```
    int found = 0;
```

```
    int strIndex[] = {0, -1};
```

```
    int maxIndex = data.length()-1;
```

```
    for(int i=0; i<=maxIndex && found<=index; i++){
```

```
        if(data.charAt(i)==separator || i==maxIndex){
```

```
            found++;
```

```
            strIndex[0] = strIndex[1]+1;
```

```
            strIndex[1] = (i == maxIndex) ? i+1 : i;
```

```
        }
```

```
    }
```

```
    return found>index ? data.substring(strIndex[0], strIndex[1]) : "";
```

```
}
```