|  |
| --- |
| C:\Users\lbarros.DEI\AppData\Local\Microsoft\Windows\INetCache\Content.Word\EE-C.PNG |
| Bruno Miguel Vasconcelos da Silva, a88289  Diogo Miguel Cunha Fernandes, a88262  Duarte Miguel Novo Rodrigues, a88259  Francisco Lopes Salgado, a88231  João Pedro Dias Miranda, a88237  José Tomás Lima de Abreu, a88218  **DWR-19**  ***Digital Waiter Robot*** | |
| Projeto integrador  Laboratórios e Práticas Integradas  Trabalho realizado sob a orientação do  **Professor Luís Barros** | |
| 23 junho 2021 | |

**Índice**

[Lista de Figuras 5](#_Toc75031087)

[Lista de Tabelas 7](#_Toc75031088)

[Acrónimos e Siglas 9](#_Toc75031089)

[Capítulo 1 Introdução 11](#_Toc75031090)

[1.1 Introdução 11](#_Toc75031091)

[1.2 Enquadramento 11](#_Toc75031092)

[1.3 Especificações Previstas 12](#_Toc75031093)

[1.3.1 Especificações funcionais 13](#_Toc75031094)

[1.3.2 Especificações técnicas 13](#_Toc75031095)

[1.4 Planeamento 14](#_Toc75031096)

[Capítulo 2 Arquitetura e Módulos Utilizados 15](#_Toc75031097)

[2.1 Introdução 15](#_Toc75031098)

[2.2 Sensores 16](#_Toc75031099)

[2.2.1 *Array* de Sensores de reflexão 17](#_Toc75031100)

[2.2.2 Sensor de obstáculos 17](#_Toc75031101)

[2.3 Driver 18](#_Toc75031102)

[2.4 Módulos 18](#_Toc75031103)

[2.4.1 *Radio Frequency Identification* (RFID) 19](#_Toc75031104)

[2.4.2 Bluetooth 20](#_Toc75031105)

[*2.5* Microcontrolador e *Shield* 21](#_Toc75031106)

[2.6 Circuito de Alimentação 22](#_Toc75031107)

[Capítulo 3 Implementação em Software 25](#_Toc75031108)

[3.1 Introdução 25](#_Toc75031109)

[3.2 Periféricos 26](#_Toc75031110)

[3.2.1 *Direct Memory Acess* (DMA) 27](#_Toc75031111)

[3.2.2 *Analog to Digital Converter* (ADC) 28](#_Toc75031112)

[*3.2.3* *TIMER* 29](#_Toc75031113)

[3.2.4 *Serial Peripheral Interface* (SPI) 31](#_Toc75031114)

[3.2.5 *Universal Synchronous/Asynchronous Receiver/Transmitter* (USART) 32](#_Toc75031115)

[3.3 Descrição de *Software* e Módulos Criados 33](#_Toc75031116)

[3.3.1 MódulosCriados 33](#_Toc75031117)

[3.3.2 Circuito de Controlo 40](#_Toc75031118)

[3.3.2.1 O que é um Controlador? 40](#_Toc75031119)

[3.3.2.2 Análise do sistema de controlo 41](#_Toc75031120)

[3.3.2.3 Controlador implementado 43](#_Toc75031121)

[3.4 Simulações 46](#_Toc75031122)

[Capítulo 4 Lista de Componentes 47](#_Toc75031123)

[Capítulo 5 Circuito Mecânico Implementado 53](#_Toc75031124)

[Capítulo 6 Resultados Experimentais 57](#_Toc75031125)

[Capítulo 7 Análise do Produto 59](#_Toc75031126)

[7.1 Introdução 59](#_Toc75031127)

[7.2 Fiabilidade 59](#_Toc75031128)

[7.3 Segurança 60](#_Toc75031129)

[7.4 Certificação 61](#_Toc75031130)

[Capítulo 8 Conclusões 65](#_Toc75031131)

[8.1 Conclusão 65](#_Toc75031132)

[8.2 Sugestões de Trabalho Futuro 66](#_Toc75031133)

[8.2.1 Possíveis Opções Alternativas para o Desenho dos Circuitos 67](#_Toc75031134)

[8.2.2 Evolução do Atual Desenho para um Sistema Baseado em Microcomputador 68](#_Toc75031135)

[Referências 69](#_Toc75031136)

Lista de Figuras

[Figura 1.1 - Diagrama de Gantt do planeamento inicial. 14](#_Toc75031137)

[Figura 1.2 - Diagrama de Gantt do planeamento revisto. 14](#_Toc75031138)

[Figura 2.1 - Diagrama geral das principais interações no sistema. 15](#_Toc75031139)

[Figura 2.2 - *Array* de sensores QTR-8A. 17](#_Toc75031140)

[Figura 2.3 - Módulo Driver L298N. 18](#_Toc75031141)

[Figura 2.4 - Módulo RFID MFR522 19](#_Toc75031142)

[Figura 2.5 - Módulo *Bluetooth* HC-05 20](#_Toc75031143)

[Figura 2.6 - Diagrama da comunicação Bluetooth 21](#_Toc75031144)

[Figura 2.7 - Shield (a) Esquemático; (b) Implementação. 22](#_Toc75031145)

[Figura 2.8 - Esquema de ligação das baterias e BMS. 22](#_Toc75031146)

[Figura 3.1 - Diagrama da máquina de estados. 25](#_Toc75031147)

[Figura 3.2 - Exemplo de transferência do DMA. 27](#_Toc75031148)

[Figura 3.3 - Configurações possíveis do DMA. 28](#_Toc75031149)

[Figura 3.4 - conversor por aproximação sucessiva. 28](#_Toc75031150)

[Figura 3.5 - Modo de funcionamento independente (a) Single-channel, single conversion mode; (b) Multichannel, single conversion mode; (c) Single-channel, continuous conversion mode; (d) Multichannel, continuous conversion mode. 29](#_Toc75031151)

[Figura 3.6 – Comparação das características do *timer.* 30](#_Toc75031152)

[Figura 3.7 - Esquema de ligação entre Master e Slave. 32](#_Toc75031153)

[Figura 3.8 - Esquema de ligação entre 2 UARTs. 32](#_Toc75031154)

[Figura 3.9 - Estrutura de um pacote de dados enviado por UART 33](#_Toc75031155)

[Figura 3.10 - Divisão do software criado nos vários módulos. 34](#_Toc75031156)

[Figura 3.11 - Estrutura que define um motor. 35](#_Toc75031157)

[Figura 3.12 - Enumerado que representa os sensores do QTR utilizados. 35](#_Toc75031158)

[Figura 3.13 - Estrutura que agrupa as variáveis de cálculo do algoritmo PID. 36](#_Toc75031159)

[Figura 3.14 - Duração dos vários timeouts, em segundos, e definição das flags respetivas. 36](#_Toc75031160)

[Figura 3.15 - Enumerado com os possíveis estados de movimento. 37](#_Toc75031161)

[Figura 3.16 - Módulo RFID: a) Estrutura que define um cartão RFID; b) Estado do leitor RFID. 37](#_Toc75031162)

[Figura 3.17 - Estado do módulo Bluetooth. 37](#_Toc75031163)

[Figura 3.18 - Definição de um callback de um comando e da estrutura que define um comando. 38](#_Toc75031164)

[Figura 3.19 - Ações possíveis a realizar num checkpoint, e, definição de um checkpoint. 38](#_Toc75031165)

[Figura 3.20 - Definição de uma janela deslizante. 38](#_Toc75031166)

[Figura 3.21- Definição da estrutura debounce. 39](#_Toc75031167)

[Figura 3.22 - Estados da máquina de estados. 39](#_Toc75031168)

[Figura 3.23 - Definição da máquina de estados (a) Array de funções de estado; (b) Execução da máquina de estados. 39](#_Toc75031169)

[Figura 3.24 - Ações de controlo (a) ação proporcional; (b) ação integral; (c) ação derivativa 41](#_Toc75031170)

[Figura 3.25 - Diagrama de blocos do sistema de controlo 43](#_Toc75031171)

[Figura 3.26 - Serviço de Rotina à Interrupção do algoritmo PID 46](#_Toc75031172)

[Figura 5.1 - Desenho (à esquerda) e imagem real (à direita) do DWR (a) vista superior; (b) vista inferior; (c) vista lateral direita; (d) vista dianteira; (e) vista traseira. 55](#_Toc75031173)

[Figura 5.2 - Modelo 3D do DWR 55](#_Toc75031174)

[Figura 7.1 - Marcação CE. 62](#_Toc75031175)

[Figura 7.2 - Símbolos de perigo: (a) perigoso para o ambiente; (b) corrosivo; (c) comburente; (d) inflamável; (e) explosivo; (f) tóxico; (g) vários perigos; (h) *Eletrostatic Sensitive Device - ESD*. 63](#_Toc75031176)

[Figura 7.3 - Símbolo WEEE. 63](#_Toc75031177)

Lista de Tabelas

[Tabela 2.1 - Sensores e sua utilização. 17](#_Toc75031178)

[Tabela 2.2 - Tabelas de verdade do *driver* L298N (a) Controlo do Motor A; (b) Controlo do Motor B. 18](#_Toc75031179)

[Tabela 2.3 - Gamas de frequência e alcance etiquetas RFID 19](#_Toc75031180)

[Tabela 2.4 - *Pinout* módulo Bluetooth HC-05 20](#_Toc75031181)

[Tabela 3.1 - Mapeamento dos *timers*. 30](#_Toc75031182)

[Tabela 3.2 - Linhas lógicas para a transferência de dados do protocolo SPI. 31](#_Toc75031183)

[Tabela 4.1 - Lista de componentes. 47](#_Toc75031184)

[Tabela 6.1 - *Duty cycle* do sinal PWM de saída em função da tensão de entrada. 57](#_Toc75031185)

[Tabela 7.1 - Tempo de vida dos componentes com maior probabilidade de falha, usados no AWR. 60](#_Toc75031186)

[Tabela 8.1 - Número de horas despendidas por elemento 66](#_Toc75031187)

Acrónimos e Siglas

|  |  |
| --- | --- |
| **Acrónimo/Sigla** | **Significado** |
| DWR | *Digital Waiter Robot*  Robô Empregado de Mesa Digital |
| LED | *Light Emitting Diode*  Díodo Emissor de Luz |
| PWM | *Pulse Width Modulation*  Modulação de Largura de Pulso |
| BMS | *Battery Management System*  Sistema de Manutenção das Baterias |
| FW | *Forward*  Para a Frente |
| BW | *Backwards*  Para Trás |
| FWR | *Forward Right*  Motor Direito para a Frente |
| FWL | *Forward Left*  Motor Esquerdo para a Frente |
|  |  |
| BWR | *Backwards Right*  Motor Direito para Trás |
| BWL | *Backwards Left*  Motor Esquerdo para Trás |
| ESD | *Electrostatic-Sensitive Device*  Equipamento Sensível à Eletricidade Estática |
| PCB | *Printed Circuit Board*  Placa de Circuito Impresso |
| IC | *Integrated Circuit*  Circuito Integrado |
| AmpOp | AmplificadorOperacional |
| WEEE | *Waste Electrical and Electronic Equipment*  Resíduos de Equipamentos Elétricos e Eletrónicos |
| MTBF | *Mean Time Between Failures*  Período Médio Entre Falhas |
|  | Frequência de Oscilação |

# Introdução

## Introdução

Perante o atual panorama pandémico da Covid-19 [1] pretende-se, com a realização do Projeto Integrador da Unidade Curricular de Laboratórios e Práticas Integradas II (LPI II) do curso Mestrado Integrado em Engenharia Eletrónica Industrial e Computadores, a implementação de uma ideia com o objetivo de diminuir os contactos interpessoais que possam surgir no momento da entrega de bens a pessoas hospitalizadas. Esta situação pandémica é uma oportunidade para acelerar a transformação da área saúde.

Tendo em consideração que os hospitais tiveram um aumento considerável do número de internamentos [2], pretende desenvolver-se um produto que permita a entrega e recolha de bens essenciais de forma segura. De forma a facilitar a sua desinfeção e o seu manuseamento, o robô deverá ter superfícies lisas e uma interface simples.

A maioria das ideias nesta área tem como foco principal a saúde pública da população em geral, tais como, robôs de desinfeção [3], robôs que repõem o *stock* em hospitais [4] ou que medem a temperatura corporal através de câmaras [5]. Prevê-se que o *Digital Waiter Robot* (DWR) possa ser aplicado em contexto hospitalar. Na China construiu-se um robô (*little peanut*) [6] com a mesma finalidade, que foi utilizado num hotel para entregar comida porta a porta a hóspedes com suspeita de infeção.

A versatilidade do sistema permitirá que o DWR possa auxiliar na distribuição de bens noutros contextos dependendo das funcionalidades requeridas, como, por exemplo, na indústria hoteleira ou em ambiente doméstico.

## Enquadramento

O DWR é um robô seguidor de linha focado na automatização de um hospital. O DWR estará parado num local apropriado à espera do envio de um pedido e de uma rota por parte da unidade de controlo. Assim que esta conclua a comunicação com o robô, um funcionário responsável deverá colocar no seu suporte os pedidos respetivos. O robô fará chegar cada pedido a cada paciente, parando apenas nos quartos solicitados. Depois de atender a todos os pedidos, o DWR voltará ao local de onde partiu.

Considere-se o exemplo de o DWR ser responsável pela distribuição de bens alimentares. O robô deve ser colocado próximo dos locais de confeção dos alimentos. Assim que a unidade de controlo envie a rota a percorrer e as refeições a distribuir, um dos responsáveis pela secção alimentar deverá colocar estes pedidos na base do robô, dar ordem de início de marcha, e o robô tratará de tudo o resto. O DWR pode assumir outras funções, como por exemplo, na distribuição de medicamentos. A unidade de controlo poderia ser uma base de dados de gestão autónoma que tendo os dados dos quartos ocupados e dos pacientes aí hospitalizados gerisse as suas necessidades, como, por exemplo, qual a medicação prescrita, bem como o horário a que deve ser tomada. Neste projeto, implementou-se uma versão mais simples de forma a ser possível validar o conceito do robô. A unidade de controlo é uma aplicação onde, apenas, se escolhe o quarto onde o pedido deve ser entregue, e as rotas para chegar ao destino estão desde já pré-estabelecidas.

## Especificações Previstas

O DWR seguirá uma linha preta previamente colocada no piso do hospital, que define os locais acessíveis pelo robô. Como os hospitais possuem vários quartos, em vários corredores, o robô terá de ser capaz de percorrer um percurso com várias intercessões de corredores. Assim, o sistema a ser implementado será constituído por dois subsistemas: o robô DWR e a unidade de controlo. Apesar de neste projeto ser usado apenas um robô, a unidade de controlo poderá, eventualmente, controlar vários.

Como a alimentação do robô será a baterias, eventualmente, terá de ser ligado à rede elétrica para ser carregado. Assim, existirá uma estação de carregamento que estará presente num local denominado por base.

O tipo de desenvolvimento deste produto pode ser classificado como “ofensivo”. “O objetivo é colocar no mercado um produto com funcionalidades e características inovadoras ou com preço significativamente mais baixo do que produtos com funcionalidades e características equivalentes, de forma a obter para o produto quota de mercado ou aumento da quota de mercado em relação a produtos antecessores” [7].

### Especificações funcionais

A unidade de controlo é responsável por uma aplicação de interface de gestão de pedidos.

Cada quarto e cruzamento no percurso deverá ter um identificador único e o DWR deve ser capaz de os detetar para que possa seguir a rota previamente estabelecida parando apenas nos quartos previstos. Após o paciente efetuar a recolha dos bens a si destinados, poderá acionar o robô de forma que este reinicie o seguimento da linha. Assim que não existam mais pedidos pendentes, o DWR retorna à base.

Para evitar que o DWR colida, este deve ter um sistema de deteção de obstáculos, que, ao detetar um objeto no seu percurso, faz com que o este pare. Se ao fim de um determinado intervalo de tempo, previamente estabelecido, a via se mantiver obstruída, deve ser emitido um sinal sonoro para alertar as pessoas ao seu redor. Se o problema for resolvido, o DWR continua o seu trajeto. Caso contrário, entrará num estado de erro e envia uma notificação para a aplicação de interface, alertando um funcionário responsável do sucedido

### Especificações técnicas

Para implementação do sistema de controlo do DWR será usado um microcontrolador STM32F767ZI-NUCLEO [8], e o IDE STM32CubeIDE [9], que integra as ferramentas necessárias para a configuração de todos dos periféricos.

Para cumprir o objetivo de seguir de linha, usar-se-á um *array* de oito sensores de reflexão com saídas analógicas. Ao contrário dos sensores digitais que apresentam apenas dois níveis nas suas saídas, alto ou baixo, este tipo de sensores possuem uma maior sensibilidade, permitindo que o sistema de seguidor de linha apresente menos oscilações.

A deteção e identificação de intercessões de corredores será feita por meio da tecnologia RFID [11]. O sistema de deteção de obstáculos será composto por um módulo de sensores de distância infravermelhos adequado para calcular com precisão a distância a objetos que possam aparecer na frente do DWR. O sistema de alerta sonoro fará uso de um *buzzer* ativo. A comunicação entre o robô e a unidade de controlo será implementada recorrendo a tecnologia *Bluetooth.*

## Planeamento

Na Figura 1.1, mostra-se o diagrama de Gantt do planeamento inicial elaborado na etapa 0 deste projeto. Face às dificuldades encontradas, o planeamento inicial revelou-se demasiado ambicioso. Assim, este teve de ser encurtado, eliminando-se o módulo de controlo remoto. A Figura 1.2 apresenta o diagrama de Gantt com as alterações efetuadas.



Figura 1.1 - Diagrama de Gantt do planeamento inicial.

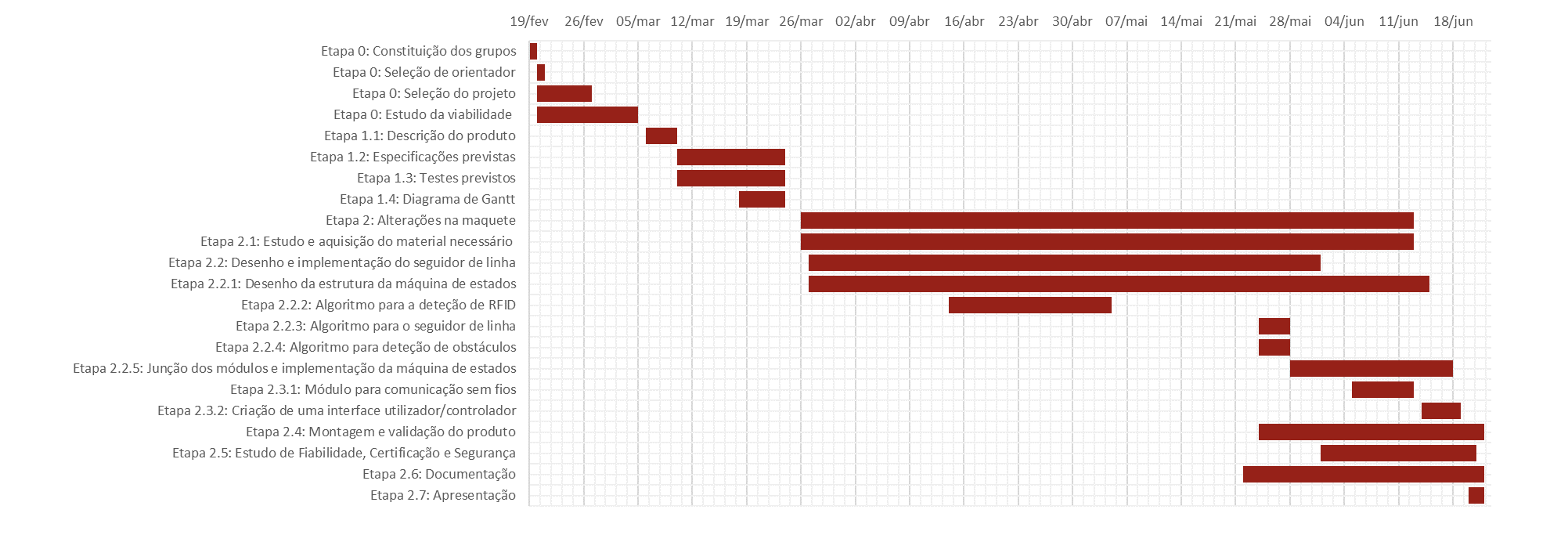


Figura 1.2 - Diagrama de Gantt do planeamento revisto.

# Arquitetura e Módulos Utilizados

## Introdução

O principal objetivo do DWR é auxiliar na distribuição de bens a várias pessoas, sendo controlado por um responsável. Na Figura 2.1, é apresentado o diagrama geral das principais interações do DWR. Considere‑se o operador como o responsável pelo DWR e o utilizador como a pessoa à qual se destinam os bens.

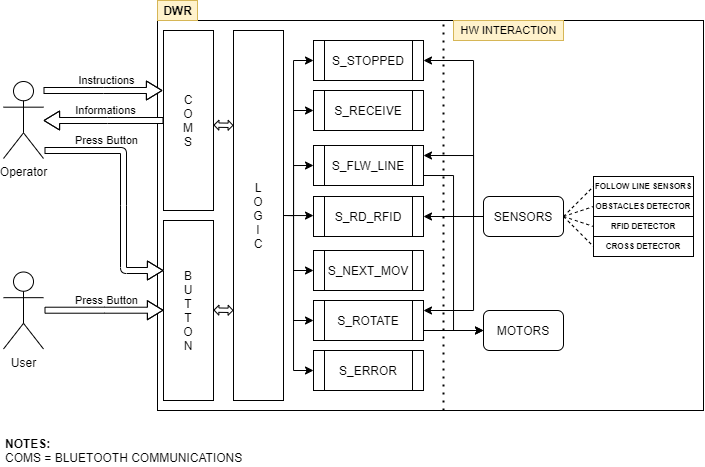


Figura 2.1 - Diagrama geral das principais interações no sistema.

O operador deve enviar instruções para o robô, via *Bluetooth,* (VER INTERAÇÃO COM O OPERADOR – APLICAÇÃO ???) selecionando a rota que pretende que este efetue. Após isto, o operador pode iniciar o movimento do DWR, pressionando o botão de pressão presente na lateral deste. Este botão também é usado pelo utilizador, que, depois de levantar os bens a si relativos, pode premi-lo, permitindo ao robô continuar o seu percurso. Além disso, o DWR envia informações para o operador, de modo que este possa monitorizar o seu estado de funcionamento.

O DWR é estruturado em diferentes blocos. Os blocos de comunicação e botão permitem a interação entre o robô e o operador / utilizador. Estes blocos relacionam-se com a lógica desenvolvida para controlar os estados do robô, tendo isso sido feito através de uma máquina de estados. Os diferentes estados de funcionamento do robô são, tal como apresentado na Figura 2.1, S\_STOPPED, S\_RECEIVE, S\_FLW\_LINE, S\_RD\_RFID, S\_NEXT\_MOV, S\_ROTATE e S\_ERROR. No estado S\_STOPPED, o DWR está parado à espera de algum estímulo. O estado S\_RECEIVE dedica‑se à escolha de novas rotas, através da comunicação entre o operador e o robô. O estado S\_FLW\_LINE implementa o controlo do seguidor de linha. O estado S\_RD\_RFID é responsável pela leitura de um cartão RFID de identificação unívoca para cada quarto e cruzamento. O estado S\_NEXT\_MOV é um estado de decisão, responsável por fazer transitar o robô para um estado que esteja de acordo com o percurso a realizar. O estado S\_ROTATE executa o controlo da mudança de direção do robô. O estado S\_ERROR é o estado para o qual o robô transita aquando da ocorrência de um erro que comprometa o normal funcionamento do sistema, informando o operador do sucedido.

A camada de interação com o *hardware* é composta por sensores e atuadores. As saídas dos sensores de obstáculos, de linha, de RFID e de cruz (sensor de paragem) são utilizadas nos estados S\_STOPPED, S\_FLW\_LINE, S\_RD\_RFID e S\_ROTATE. Os atuadores, os motores, são controlados nos estados S\_FLW\_LINE e S\_ROTATE.

## Sensores

Um sensor é um dispositivo que responde a um estímulo do ambiente, físico ou químico, produzindo um sinal que pode ser transformado noutra grandeza física para fins de medição.

Para o DWR, foram usados sensores infravermelhos, que são muito utilizados em aplicações que envolvem leitura e deteção de proximidade. Estes usam a luz infravermelha, que é uma radiação eletromagnética, de baixa frequência, estando abaixo do espetro eletromagnético da luz visível ao olho humano. Este tipo de sensores fazem a sua leitura através da utilização de um emissor, normalmente, laser ou LED, e um recetor fotoelétrico, que contém um elemento optoelétrico, como, por exemplo um fotodíodo ou um fototransístor. Este recetor fotoelétrico deteta a luz vinda do emissor e converte a intensidade da luz recebida num sinal elétrico em tensão. Perante as características acima enumeradas, os sensores infravermelhos podem ser usados para implementar sensores de obstáculos e sensores leitores de linha.

### *Array* de Sensores de reflexão

Com o objetivo de o DWR seguir a linha escolheu-se o *array* de sensores QTR-8A, apresentado na Figura 2.2. Este pode ser alimentado com 3,3 V ou 5 V e tem oito sensores analógicos que apresentam na sua saída um valor de tensão compreendido entre 0 V e o valor de alimentação. Sobre uma superfície branca, os sensores medem uma tensão de, aproximadamente, 0 V. Já quando se encontram sobre uma superfície preta, os sensores medem uma tensão de, sensivelmente, 3,3 V. Para as restantes cores, os sensores apresentam tensões entre estas duas gamas. No contexto deste projeto, é indispensável a escolha de um *array* que produza na saída uma tensão máxima de 3,3 V, uma vez que este é o limite que os *analog to digital converters* (ADCs) da STM32F767ZI-NUCLEO.

Uma imagem com texto

Descrição gerada automaticamente

Figura 2.2 - *Array* de sensores QTR-8A.

A Tabela 2.1 apresenta a finalidade dos sensores do QTR-8A nos circuitos do DWR.

Tabela 2.1 - Sensores e sua utilização.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sensor 1** | **Sensor 2** | **Sensor 3** | **Sensor 4** | **Sensor 5** | **Sensor 6** | **Sensor 7** | **Sensor 8** |
| Deteção cruzamento /quarto | - | Seguidor de linha | Deteção linha preta | Deteção linha preta | Seguidor de linha | - | Deteção cruzamento /quarto |

### Sensor de obstáculos

Face à aplicação do DWR, torna-se necessário implementar um sistema de deteção de obstáculos, de forma a evitar a colisão com eventuais objetos que impeçam a sua passagem. Para isso, utilizou-se o sensor de distância infravermelho [ref] da fabricante SHARP. Este sensor permite medir distâncias a objetos entre 10 e 80 cm, tendo uma saída do tipo analógico. Assim, para esta aplicação, este sensor será usado como sensor de proximidade. Este pode ser alimentado com tensões entre 4,5 V e 5,5 V, tendo sido usada uma tensão de alimentação de 5 V.

## Driver

Para controlo dos motores utilizou-se o módulo *driver* apresentado na Figura 2.3. Este *driver* possui seis entradas, três para cada motor: os sinais de *ENA* e *ENB* permitem ativar ou desativar os motores, controlando as suas velocidades de rotação. Por sua vez, os sinais de *IN1*, *IN2*, *IN3* e *IN4* definem o modo de rotação de cada motor. Na Tabela 2.2 encontra-se exemplificado o modo de operação do motor em função da combinação lógica dos sinais de entrada.

Uma imagem com texto, eletrónica

Descrição gerada automaticamente

Figura 2.3 - Módulo Driver L298N.

Tabela 2.2 - Tabelas de verdade do *driver* L298N (a) Controlo do Motor A; (b) Controlo do Motor B.

|  |  |
| --- | --- |
| **(a)** | **(b)** |
| |  |  |  |  | | --- | --- | --- | --- | | ENA | IN1 | IN2 | Motor A | | 0 V | N/A | N/A | OFF | | 5 V | 0 V | 0 V | Trava | | 5 V | 0 V | 5 V | Sentido anti-horário | | 5 V | 5 V | 0 V | Sentido horário | | 5 V | 5 V | 5 V | Trava | | |  |  |  |  | | --- | --- | --- | --- | | ENB | IN3 | IN4 | Motor B | | 0 V | N/A | N/A | OFF | | 5 V | 0 V | 0 V | Trava | | 5 V | 0 V | 5 V | Sentido anti-horário | | 5 V | 5 V | 0 V | Sentido horário | | 5 V | 5 V | 5 V | Trava | |

## Módulos

Ao longo do percurso, o DWR tem de tomar várias decisões de modo a seguir a rota proposta fazendo as paragens necessárias para entrega de bens. De modo a tornar este propósito possível, o DWR tem de ser capaz de identificar e distinguir os cruzamentos e/ou quartos existentes ao longo do percurso. Decidiu-se então atribuir a cada um destes uma etiqueta RFID com códigos de identificação únicos. De modo a ser possível comunicar com o robô à distância, decidiu-se implementar comunicação por *Bluetooth*.

### *Radio Frequency Identification* (RFID)

A identificação por radio frequência (RFID) é uma tecnologia de leitura sem contacto que usa ondas eletromagnéticas para ler o código de identificação de uma etiqueta RFID. Como cada etiqueta possui um código único, estas podem ser usadas para associar um ID único a um objeto. Existem dois tipos de etiquetas RFID: as passivas e as ativas. Etiquetas passivas usam a energia fornecida pelas ondas eletromagnéticas para induzir uma corrente na antena, de modo a transmitir os dados da mesma. As etiquetas ativas possuem uma fonte de alimentação própria, como uma bateria, para alimentar os circuitos necessários para a transmissão.

Outras características que influenciam a escolha das etiquetas e do módulo de leitura, são a frequência de comunicação, o alcance e o preço. Atualmente as etiquetas disponíveis no mercado operam em três gamas de frequência e têm diferentes alcances, como apresentado na Figura 3.1. Sabendo que, por norma, os módulos RFID que operam a frequências mais altas têm preços mais elevados e que não existem muitos leitores RFID do tipo *Low-Frequency* disponíveis no mercado, decidiu‑se usar um modulo RFID do tipo *High‑Frequency*. Além do referido, as etiquetas não precisam de uma fonte de alimentação para o seu funcionamento e também porque o alcance de deteção destas se encontra na gama pretendida para o DWR. Escolheu-se, então, o módulo MFRC522[ref], Figura 2.4, uma vez que oferece as características pretendidas.

Tabela 2.3 - Gamas de frequência e alcance etiquetas RFID

|  |  |  |  |
| --- | --- | --- | --- |
|  | Gama de Frequência | Alcance | Tipo de Alimentação |
| Low-Frequency | Entre 30 KHz e 300 KHz | Até 10 cm | Passiva |
| High-Frequency | 13,56 MHz | Até 30 cm | Passiva/Ativa |
| Ultra High-Frequency | Entre 300 MHz e 3 GHz | Entre 20 m e 100 m | Ativa |

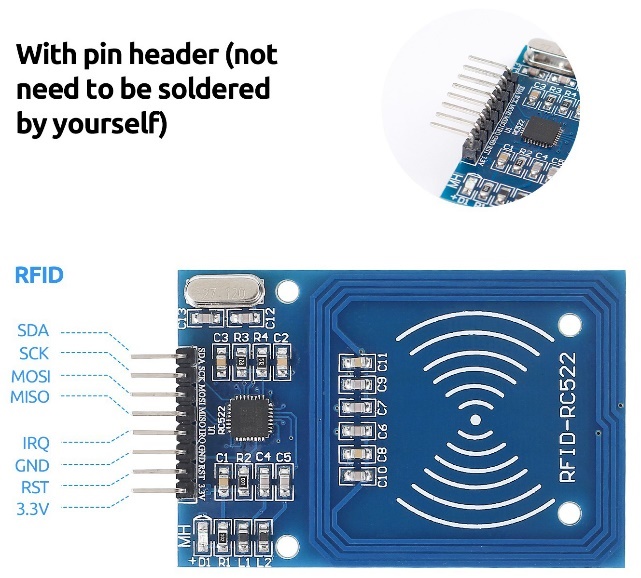


Figura 2.4 - Módulo RFID MFR522

### Bluetooth

Optou-se pela tecnologia Bluetooth por ser uma tecnologia fiável e de fácil implementação. Para a implementação desta funcionalidade, usou-se o módulo *Bluetooth* HC-05 [ref], que se apresenta na Figura 2.5. Este módulo possui seis pinos com as especificações descritas na Tabela 2.4



Figura 2.5 - Módulo *Bluetooth* HC-05

Tabela 2.4 - *Pinout* módulo Bluetooth HC-05

|  |  |
| --- | --- |
| Receive (Rx) | Receção de dados por comunicação série |
| Transmite (Tx) | Transmissão de dados por comunicação série |
| State | Indica que um dispositivo foi ligado ao módulo e está ligado a um LED do mesmo |
| Enable (EN) | É usado para alterar o modo de funcionamento do módulo para o modo de *AT Command*. Neste modo o HC-05 pode receber um conjunto de comandos, que são enviados através de comunicação série para modificar alguns parâmetros do módulo, como por exemplo o *baud rate* |
| +5 V e GND | Alimentação do módulo |

Como se pode observar no diagrama da Figura 2.6, o utilizador interage com a camada de aplicação e os dados por si inseridos são enviados para o módulo HC-05 através da camada protocolar *Bluetooth*. A comunicação entre o módulo HC-05 e o microcontrolador é feita por comunicação série RS232 com *baud rate* de 9600 Bits/s.

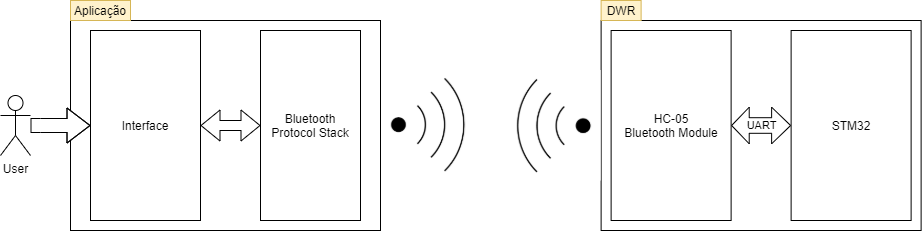


Figura 2.6 - Diagrama da comunicação Bluetooth

## Microcontrolador e *Shield*

O microcontrolador STM32F767ZI [ref] oferece diversos periféricos e pinos de General Purpose Input/Output (GPIO) que vão ser usados para diferentes finalidades. Em muitos casos, as entradas e saídas não estão organizadas de forma conveniente e existe uma enorme quantidade de ligações cruzadas na parte superior da STM. Além de não ser visualmente apelativo, esta desorganização pode levar problemas de ruídos e maus contactos, entre outros. Usando esta metodologia, sempre que se necessitasse de conectar novos módulos havia uma grande dificuldade em o fazer. Estes problemas, levaram à necessidade da criação de uma shield, uma placa de circuito impresso (PCB), com o propósito de organizar e melhorar as conexões necessárias, recorrendo-se para o efeito ao software KiCad.

O layout da shield criada está apresentado na Figura 2.7 (a). Foi projetada com o propósito único de servir os interesses do DWR. Deste modo, não se projetou uma shield genérica para o microcontrolador em questão, mas sim, uma tendo em conta os periféricos, sensores e módulos utilizados, bem como a sua disposição no circuito mecânico, evitando, ao máximo, que os fios atravessassem o microcontrolador. Além disto, integrou-se nesta shield, uma fonte que permite a alimentação dos componentes, e o módulo Bluetooth compactando todas as ligações sem necessidade de fios externos ocupando o menor espaço possível. A Figura 2.7 (b) apresenta a implementação da shield criada.

|  |  |
| --- | --- |
|  | Uma imagem com texto, eletrónica  Descrição gerada automaticamente |
| (a) | (b) |

Figura 2.7 - Shield (a) Esquemático; (b) Implementação.

## Circuito de Alimentação

Para alimentar os circuitos são necessárias tensões de 12 V, 5 V e 3,3 V. Para obter o primeiro valor de tensão, foram utilizadas três baterias recarregáveis LI-ION de 3,7 V [REF] em série. De modo a aumentar a autonomia do DWR, foram colocadas mais três baterias, cada uma em paralelo com as três já existentes. Para proteção das baterias foi usado um dispositivo de BMS [REF]. Este dispositivo controla a descarga das baterias não deixando que a sua tensão desça abaixo de um limite de segurança. Como as baterias necessitam de ser carregadas, adicionou-se ao circuito os terminais de entrada de modo a ser possível ligar um carregador [REF] em paralelo com a carga. Para proteção de todos os componentes, foi usado um fusível entre os terminais do carregador e a carga [REF]. O esquema de ligação é apresentado na Figura 2.8.

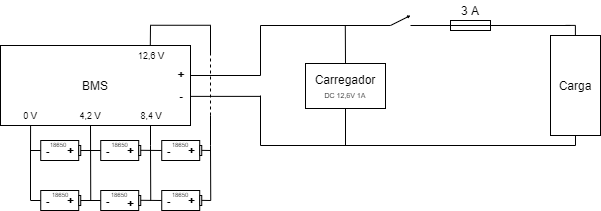


Figura 2.8 - Esquema de ligação das baterias e BMS.

Cada motor usado no DWR consome uma corrente de 580 mA [REF], e como este possui dois motores, a corrente total do par será 1160 mA. Quanto aos sensores usados, o QTR-8A consome, sensivelmente, 100 mA, o RFID 26 mA, o sensor de distância 30 mA e módulo bluetooth é 40 mA. O microcontrolador STM32F767ZI tem um consumo máximo de 258 mA. Somando as correntes consumidas por estes componentes, e assumindo que os restantes componentes têm consumos desprezáveis comparativamente com os restantes, conclui-se que consumo total do DWR será de, aproximadamente, 1614 mA. Visto que cada célula possui 2200 mAh e considerando que esta tem um rendimento de 80 %, então estão disponíveis 1760 mAh. Como cada célula está colocada em paralelo com uma outra, no total, estão disponíveis 3520 mAh para todos os circuitos. Assim, estimou-se uma autonomia de 2 horas e 10 minutos. Caso se pretenda aumentar este valor, poder-se-á adicionar mais células em paralelo com as existentes.

De forma a obter tensões de 3.3V e 5 V, necessárias para alimentação de sensores e módulos do DWR, usou-se uma fonte de alimentação para *breadboard* V2 – 5 V / 3.3 V DC [REF]. Esta fonte é alimentada via USB. Para o efeito usou-se um circuito abaixador para 5V, com quatro portas USB, que tem como entrada os 12V provenientes da BMS [REF]. Uma vez que adquiro este módulo, optou-se por alimentar a STM de igual modo. Atendendo que o *step-down* debita, no máximo 8 A, é suficiente para alimentar todos os componentes que necessitam de 5 V via USB.

# Implementação em Software

## Introdução

O DWR tem de executar ações distintas durante toda a sua atividade. De forma a ser possível executar a ação correta em todos os momentos, é necessário um sistema que faça a gestão do estado atual do robô e das entradas que possam despoletar a alteração deste mesmo estado. Na Figura 3.1, está representado o diagrama que representa a máquina de estados implementada para este robô. Esta é composta por quatro estados principais: S\_STOPPED, S\_RECEIVE, MOVEMENT e S\_ERROR. O estado de MOVEMENT pode ser subdividido em quatro estados secundários: S\_FLW\_LINE, S\_RD\_RFID, S\_NEXT\_MOV e S\_ROTATE.



Figura 3.1 - Diagrama da máquina de estados.

Inicialmente, o DWR encontra-se no estado S\_STOPPED. Quando o operador iniciar a transmissão de uma nova rota e não houver nenhuma rota em curso, o robô transita para o estado S\_RECEIVE. Após receber a rota selecionada pelo operador, o robô transita do estado S\_RECEIVE para o estado S\_STOPPED, onde espera que lhe seja dada a permissão para iniciar o seu movimento, através da pressão do botão presente na sua lateral. Assim que este botão seja pressionado, o sistema evolui para o estado de S\_FLW\_LINE. Neste estado o robô segue a linha até que aconteça algum dos seguintes imprevistos: presença de um obstáculo no percurso do robô, deteção de uma cruz de paragem (quarto ou cruzamento) ou ocorrência de falha no controlador do seguidor de linha. Caso o percurso esteja obstruído por um obstáculo, o DWR para, evitando a colisão com o mesmo, voltando para estado S\_STOPPED. Se for detetada uma linha horizontal por ambos os sensores das extremidades do array de sensores (cruz de paragem), significa que o robô está na presença de um quarto ou cruzamento, sendo necessário efetuar a leitura do cartão RFID a este associado, no estado S\_RD\_RFID. Caso o cartão RFID seja lido com sucesso, o sistema evolui para o estado S\_NEXT\_MOV, onde determina qual o próximo estado do DWR, de acordo com a rota selecionada. No estado S\_NEXT\_MOV, o robô transita para o estado S\_FLW\_LINE se o RFID detetado for relativo a um quarto que não esteja marcado como local de paragem na rota ou a um cruzamento em que não seja necessário efetuar a mudança de direção. Pelo contrário, caso seja necessário efetuar paragem no quarto atual, dá-se uma transição para o estado S\_STOPPED. No caso do RFID detetado ser relativo a um cruzamento e for necessário mudar de direção, o sistema transita para o estado S\_ROTATE. Uma vez neste estado, o DWR roda na direção indicada pela rota, até estar orientado na direção pretendida, voltando ao estado S\_FLW\_LINE.

Todas estas ações têm timeouts associados, que são ativos quando a ação demora mais tempo a ser realizada do que o esperado, permitindo ter controlo sobre o robô em casos imprevisíveis, transitando para o estado S\_ERROR, por exemplo, quando o DWR se encontrar parado à espera de que o percurso seja desobstruído por um longo período de tempo, quando ocorre um erro (cartão errado ou demasiado tempo) na leitura de um cartão RFID ou quando ocorre um erro na mudança de direção do robô. Além disso, o robô transita para o estado S\_ERROR quando o robô sai totalmente da linha durante o seu percurso. Uma vez neste estado, o DWR precisa da intervenção de um responsável para que possa voltar ao seu estado de funcionamento normal.

## Periféricos

De modo a cumprir com os objetivos proposto neste projeto, é necessário o uso de diferentes periféricos presentes no microcontrolador. Para gerir bases de tempo e gerar sinais PWM utilizam-se *timers.* Para a leitura do arrayde sensores de linha e do sensor de obstáculo faz-se uso do ADC em conjunto com o DMA. O módulo de RFID utiliza o protocolo SPI e a comunicação entre o microcontrolador e o módulo *Bluetooth* realiza-se através da UART.

### *Direct Memory Acess* (DMA)

Uma unidade de Acesso à Memória Direta (DMA) é um elemento lógico digital que pode ser usado em conjunto com o microprocessador para executar as operações de transferência de memória. Deste modo reduz-se, significativamente, a carga da unidade central de processamento (CPU). O DMA permite realizar transferências entre memórias, entre o periférico e a memória e entre a memória e o periférico. Os dispositivos compartilham o barramento de memória e os barramentos de periféricos com o CPU, tal como mostrado na Figura 3.2. No diagrama, o dispositivo DMA lê o valor de um periférico a partir do barramento do periférico e grava na memória através do barramento de memória. Na STM o princípio é o mesmo, mas com os vários barramentos de periféricos.

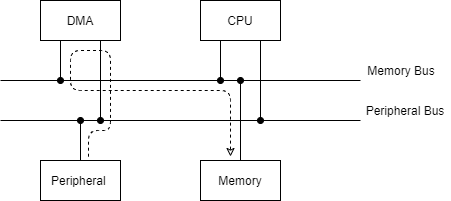


Figura 3.2 - Exemplo de transferência do DMA.

Na STM32F767ZI existem dois controladores DMA (DMA1 e DMA2) que possuem dezasseis streams (canais de acesso à memória) no total, oito para cada. Existem quatro níveis de prioridade para o pedido da stream. Caso esta tenha igual nível em várias streams*,* a streamcom o nível mais baixo sobrepõe-se à do nível mais alto, por exemplo, a stream0 tem prioridade em relação à stream1.

O DMA pode ser configurado em dois modos de transferência de dados: modo direto, em que a transferência de dados é feita de forma imediata, ou modo first *in* first *out* (FIFO)*,* em que os dados são armazenados temporariamente antes de serem transmitidos para a memória. Em relação aos tamanhos dos dados, estes podem variar entre um byte, halfword(2 *bytes* consecutivos) ou word (4 *bytes* consecutivos). Outras potencialidades do DMA surgem, no facto, de os apontadores se auto incrementarem, possibilitando a escrita de várias posições de memória consecutivas e a sua reprogramação ser automática (modo circular). De forma a inicializar o DMA entre um periférico específico e uma memória, é necessário configurar o canal apropriadamente e definir todas as configurações necessárias de acordo com a Figura 3.3.

Uma imagem com mesa

Descrição gerada automaticamente

Figura 3.3 - Configurações possíveis do DMA.

### *Analog to Digital Converter* (ADC)

Um Analog *to Digital Converter* (ADC) é um circuito eletrónico que converte uma entrada analógica num valor digital. Este valor representa o nível de tensão num código binário e permite aos microcontroladores comunicar com o mundo real com base nos valores amostrados. A STM32F767ZI possui ADCs de aproximação sucessiva (composto por um comparador e um *Digital to* Analog *Converter* (DAC) interno para aproximar, sucessivamente, o valor de saída do ADC ao valor de entrada, Figura 3.4), com até dezanove canais multiplexados, permitindo a leitura dezasseis fontes externas, duas fontes internas e o canal . É possível configurar os canais com uma resolução de 12-bit, 10-bit, 8-bit ou 6‑bit, alinhamento de dados à direita ou à esquerda, e o tempo de amostragem. O ADC apenas realiza leituras corretamente para valores de entrada compreendidos entre 0 V e 3,3 V.

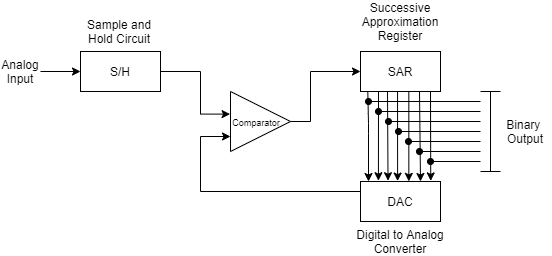


Figura 3.4 - conversor por aproximação sucessiva.

Cada ADC pode ser configurado, no modo independente, como Single*-*channelou Multichannel. No primeiro, apenas é lido um canal, ao passo que, no segundo, são lidos vários canais sucessivamente. Além disso, pode ser configurado em *single* conversionmodee continuousconversionmode. No primeiro, o ADC realiza uma única conversão. No segundo, o ADC inicializa uma nova conversão logo que a conversão em curso termine. As possíveis combinações destes modos de operação estão apresentadas na Figura 3.5. É possível definir a flagde final de conversão para fim de cada conversão ou para o fim de todas as conversões. O ADC possui um único registo de dados, por isso, quando se lê múltiplas entradas analógicas, é essencial que os dados sejam lidos entre duas amostras. Isto pode ser realizado através de polling*,* interrupção ou DMA. Interrupções e DMA podem ser disparados no final de cada conversão. Em dispositivos com 2 ADCs ou mais, como é o caso de versão da STM32 em uso, é possível configurar os ADCs em modo duplo/triplo e o DMA para guardar os dados. Este periférico permite mais modos de funcionamento que não serão abordados, uma vez que o seu grau de complexidade não serve o propósito do trabalho desenvolvido.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| (a) | (b) | (c) | (d) |

Figura 3.5 - Modo de funcionamento independente (a) Single-channel, single conversion mode; (b) Multichannel, single conversion mode; (c) Single-channel, continuous conversion mode; (d) Multichannel, continuous conversion mode.

Neste projeto, foram utilizados dois ADCs com resolução de 12-bit e alinhamento dos dados à direita. O ADC2 foi usado para a leitura do sensor de obstáculos e o ADC3 para a leitura dos sensores utilizados do QTR-8A. Assim, foi usado um canal do ADC2 (*Single-*channel) e seis canais do ADC3 (Multichannel). Como se pretende realizar várias conversões, foi usado o modo de conversão contínua, com flagde final de conversão para fim de cada conversão, em ambos os ADCs. Para a transferência dos dados deste periférico para a memória utilizou-se DMA.

### *TIMER*

Um *timer* é um dispositivo de *hardware* capaz de medir uma base de tempo. Pode ser usado, por exemplo, para ativar eventos a frequências ou atrasos conhecidos, gerar sinais a várias frequências, sinais de saída modulados por largura de pulso (PWM) e medir pulsos de entrada.

A STM32F767ZI tem dois *timers* básicos (TIM6 e TIM7), dez *timers* do tipo *general-*purpose(TIM2 até ao TIM5 e TIM9 até ao TIM14) e dois *timers* do tipo advanced*-*control(TIM1 e TIM8). Os *timers* básicos são os mais simples. Podem ser usados para gerar bases de tempo, circuitos de sincronização para despoletar *digital to* analogconverters(DAC) e gerar interrupções/DMA. Os *General* purposepodem ser configurados como outputcompare*,* one*-pulse* mode *output,* inputcapturee PWMgeneration output *compare,* one*-pulse* mode *output,* input capturee PWMgeneration.Estas informações podem ser observadas Figura 3.6. Na Tabela 3.1, pode‑se ver os barramentos a que cada timer pertence, bem como a frequência máxima do barramento.

Uma imagem com mesa

Descrição gerada automaticamente

Figura 3.6 – Comparação das características do *timer.*

Tabela 3.1 - Mapeamento dos *timers*.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Timer | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| APB1(108 MHz) |  | x | x | x | x | x | x |  |  |  |  | x | x | x |
| APB2(216 MHz) | x |  |  |  |  |  |  | x | x | x | x |  |  |  |

Para configurar o *timer* com a frequência desejada é necessário configurar os valores de prescaler de preload. O prescaler divide a frequência de oscilação do barramento, ao passo que o preload define o número de contagens do *timer*. Por exemplo, se o barramento tiver uma frequência de 108 MHz e o prescaler um valor de 108 ­‑ 1 (a contagem do registo começa em zero), a frequência do timer será de 108 MHz / 108 = 1 MHz, o que equivale a uma base de tempo de 1 us. Se o valor de preload for 10 ‑ 1, o timer terá uma frequência de 1 MHz / 10= 100 KHz, ou seja, um período de 1 / 100 KHz =10 us. Tendo em conta estes parâmetros e o valor da frequência do timer em questão, é possível calcular a frequência do *timer* usando a equação (3.1).

|  |  |  |
| --- | --- | --- |
|  |  | (3.1) |

Neste projeto, forem usados quatro *timers* para aplicações distintas. Para controlo dos timeoutsque podem ocorrer durante o tempo de operação do DWR, do período de amostragem do algoritmo PID, sensor de obstáculos e do debounce apenas a necessário um *timer* que despolete uma interrupção ao fim de um determinado período de tempo, ou seja, um *timer* básico. Para o período de amostragem do algoritmo PID e sensor de obstáculos utilizou-se o mesmo *timer*, TIM6, com uma base de tempo 10 ms. Para o debouncedo botão o TIM7 com uma base de tempo de 50 ms. Como o microcontrolador apenas possui dois *timers* básicos, utilizou-se um *timer* genérico, TIM3, para gerir os timeouts com uma base de tempo de 1 segundo. Para gerar os sinais de PWM à saída do controlador PID é necessário um *timer* no modo PWMGeneration. Sendo assim, selecionou-se um *timer* genérico, TIM4,e utilizaram-se dois canais (channel3e channel4).

### *Serial Peripheral Interface* (SPI)

A *Serial* Peripheral *Interface* (SPI) é um protocolo de comunicação de curto alcance criado para a troca de dados entre microcontroladores ou entre microcontroladores e sensores. O SPI é síncrono e necessita que o transmissor e o recetor estejam sincronizados através de uma linha de clock partilhada. Este suporta transferências de dados em três modos: full-duplex (transferência de dados bidirecional simultânea), half-duplex (transferência de dados bidirecional não simultânea) ou simplex (transferência de dados unidirecional).

Os dispositivos conectados por SPI encontram-se numa relação master-slave*.* O *master* é, tipicamente, um microcontrolador que envia instruções ao slave, que, na maioria dos casos, se trata de um sensor, um *chip* de memória ou um display. De modo a proceder à comunicação full‑duplex entre um dispositivo *master* e um dispositivo slave*,* as 4 linhas lógicas para a transferência de dados do protocolo SPI, Tabela 3.2, do *master* e do slave devem ser conectados segundo a configuração apresentada na Figura 3.7.

Tabela 3.2 - Linhas lógicas para a transferência de dados do protocolo SPI.

|  |  |
| --- | --- |
| Serial Clock (SCLK) | Saída proveniente do master para sincronizar as transferências de dados nas linhas MISO e MOSI. |
| Master Out Slave In (MOSI) | Saída de dados do master com destino ao slave. |
| Master In Slave Out (MISO) | Saída de dados do slave com destino ao master. |
| Slave Select, (SS) | Linha que permite selecionar o dispositivo com o qual se pretende comunicar. |

Uma imagem com texto

Descrição gerada automaticamente

Figura 3.7 - Esquema de ligação entre Master e Slave.

Neste projeto, utilizou-se o protocolo SPI para comunicação com o chip MFRC522 [ref]. Optou‑se por utilizar a SPI3, dado que os pinos mapeados para esta interface não entram em conflito com pinos usados por outros periféricos. Foi configurada para funcionar com dados no formato Motorola e tamanho de 8-bit, em que o MSB é enviado primeiro. Configurou-se, ainda, o clock da interface para a transferência de dados ocorrer a 3,375 Mbits/s (sabendo que o clock do barramento APB1 é 54 Mhz, o prescalerterá valor 16), a polaridade do clock (CPOL) a lowe a sua fase (CPHA) a zero. Deste modo, executa-se a captura dos dados provenientes do MFRC522 durante a transição ascendente.

### *Universal Synchronous/Asynchronous Receiver/Transmitter* (USART)

A universal asynchronous receiver transmitter (UART) é uma comunicação série largamente usada para comunicar entre dois dispositivos, como sistemas embebidos, microcontroladores e computadores. Na comunicação UART, a comunicação direta entre dois dispositivos faz-se através de duas linhas de dados, a de transmissão (Tx) e a de recessão (Rx), como representado na Figura 3.8.

Uma imagem com texto, relógio

Descrição gerada automaticamente

Figura 3.8 - Esquema de ligação entre 2 UARTs.

Este periférico transmite dados de forma assíncrona, ou seja, não necessita de um sinal de clock para se manter sincronizado. Em vez disso, o emissor acrescenta *bits* ao pacote a ser transmitido, sinalizando o início e o fim dos dados a ser transferidos, tal como mostrado na Figura 3.9.



Figura 3.9 - Estrutura de um pacote de dados enviado por UART

Quanto ao recetor, após detetar o startbit, começará a ler os bits que se seguem a uma determinada frequência, denominado *baud rate*, expresso em bits por segundo (bps). Para que a comunicação entre dois dispositivos possa ocorrer sem erros, ambas as UART têm de operar ao mesmo *baud rate*.

Na STM32F767ZI encontram-se disponíveis quatro módulos UART (UART4, UART5, UART7 e UART8) e quatro USART (USART1, USART2, USART3 e USART6). As últimas diferem das primeiras apenas no facto de poderem ser configuradas no modo síncrono. Todos podem ser configurados em full‑duplex (transferência de dados bidirecional simultânea) ou half-duplex(transferência de dados bidirecional não simultânea). A estrutura dos pacotes de dados pode ser modificadas, podendo escolher ter 7,8 ou 9 bits de dados, 1 ou 2 stop bits e uma definição da ordem da transmissão de dados.

Neste projeto, a UART possibilita a comunicação entre a STM e o módulo Bluetooth [ref]. O formato da comunicação série deste módulo é de oito *bits* de dados, um stop *bit* e nenhum de paridade e opera a um *baud rate* de 9600 bits/s. Deste modo, configurou-se a USART1 com as características acima descritas e ativaram-se as interrupções para processamento dos dados

## Descrição de *Software* e Módulos Criados

### MódulosCriados

Na criação de software torna-se indispensável a utilização de métodos de programação modular que consistem na divisão do código em diversos ficheiros, denominados de módulos. Esta abordagem permite estruturar um programa de forma mais clara, agrupando funções e variáveis relacionadas num mesmo ficheiro. Além disso, facilita a reutilização de funções e a manutenção de código.

A Figura 3.10 representa a divisão do *software* criado nos vários módulos e as suas interações, em que, FSM significa *Finite State Machine* (Máquina de estados) e HAL significa *Hardware Abstraction Layer.* De notar que, tal como apresentado na legenda na figura, os módulos estão agrupados por cores e as setas são usadas para representar as interações entre os módulos indicando a relação de dependência. Por exemplo, o módulo FSM depende do módulo Motion, mas Motion não depende do módulo FSM, em que Motion pertence ao grupo de módulos que controlam o movimento (*Movement Modules*).



Figura 3.10 - Divisão do software criado nos vários módulos.

Começando pela descrição dos módulos que controlam o movimento (representados a cor avermelhada, na Figura 3.10), na camada de interação com o *hardware* existemtrês módulos: Motor, QTR e Stop Sensors. O módulo Motor, permite o controlo de um motor através de um *timer* com canal PWM, que efetua a variação da tensão de alimentação média do motor, por modulação de largura de impulso (PWM). O sentido de rotação do motor é definido por dois pinos GPIO, IN1 e IN2, tal como apresentado na Tabela 2.2 do capítulo anterior. Na Figura 3.11 é apresentada a estrutura que define um motor.

Uma imagem com texto

Descrição gerada automaticamente

Figura 3.11 - Estrutura que define um motor.

O módulo QTR inicializa o *array* de sensores do seguidor de linha [ref], possibilitando a obtenção do valor analógico e do valor lógico de cada um dos sensores, sendo o valor lógico alto definido a partir de 2,45 V. A leitura dos valores dos sensores do QTR é feita através do DMA. Na Figura 3.12, encontram-se representados, em forma de um enumerado, todos os sensores do *array* de sensores usados.

Uma imagem com texto

Descrição gerada automaticamente

Figura 3.12 - Enumerado que representa os sensores do QTR utilizados.

O módulo Stop Sensors permite detetar as marcas de paragem (cruzes) e eventuais obstáculos que apareçam no percurso do DWR. Este módulo utiliza dois sensores do módulo QTR para fazer a deteção das marcas de paragem, SENSOR1 e SENSOR8, e um sensor de obstáculos. Tal como no módulo QTR, a leitura dos valores do sensor de obstáculos é feita através do DMA. Definiu-se que o sensor de obstáculos sinaliza a presença de um objeto na trajetória do robô quando este se encontra a, aproximadamente, 15 cm de distância, correspondendo a um valor digital de 0x2000. Este módulo tem dois códigos de erro associados: E\_ST\_CROSS\_FOUND, usado quando deteta uma marca de paragem, e E\_ST\_OBS\_FOUND, usado quando deteta um obstáculo.

O módulo Move controla o movimento dos motores, definindo as suas velocidades e sentidos de rotação. Além disso, este módulo inicializa os dois motores a serem usados: motor direito e motor esquerdo.

O módulo PID implementa o algoritmo do controlador PID, tal como apresentado no Capítulo 3.3.2. Na Figura 3.13, é apresentada a estrutura que define as variáveis utilizadas para realizar o algoritmo PID.

Uma imagem com texto

Descrição gerada automaticamente

Figura 3.13 - Estrutura que agrupa as variáveis de cálculo do algoritmo PID.

O módulo Timeout permite gerar quatro *timeouts,* com duração em segundos, como apresentado na Figura 3.14. Quando um *timeout* termina, a *flag* respetiva será ativa de forma a sinalizar o sucedido. Para isso, este módulo utiliza um *timer* que gera uma interrupção a cada 1 segundo.

|  |  |
| --- | --- |
| Uma imagem com texto  Descrição gerada automaticamente | |
| Uma imagem com texto  Descrição gerada automaticamente |  |

Figura 3.14 - Duração dos vários timeouts, em segundos, e definição das flags respetivas.

O módulo Lfollower implementa o seguidor de linha, através dos sensores do QTR já mencionados, aplicando o algoritmo PID, utilizando o módulo Move para provocar uma alteração na velocidade de rotação dos motores. Além disso, este módulo implementa uma função que permite rodar o robô numa direção, direita ou esquerda, até que o sensor do QTR no lado correspondente à direção de rotação, SENSOR1 e SENSOR8, respetivamente, detetem novamente a linha. Caso nenhum sensor detete a linha durante o movimento de rotação, este será parado ao fim de um *timeout* predefinido, ROTATE\_TIMEOUT apresentado na Figura 3.14. Este módulo tem dois códigos de erro associados: E\_LF\_OFF, usado quando se tenta utilizar o seguidor de linha antes de o inicializar, e E\_LF\_NO\_LINE, usado quando o seguidor de linha não encontra uma linha para seguir. Quando ROTATE\_TIMEOUT termina, é utilizado o código de erro E\_TIMEOUT.

O módulo Motion controla o movimento do robô utilizando o seguidor de linha, os sensores de paragem e o sensor de obstáculos. Na Figura 3.15, está representado um enumerado com os possíveis estados de movimento. Sempre que algum dos erros apresentados anteriormente acontece, é efetuada a mudança do estado do movimento para o estado respetivo.

Uma imagem com texto

Descrição gerada automaticamente

Figura 3.15 - Enumerado com os possíveis estados de movimento.

Este módulo utiliza um *timer* para provocar uma interrupção a cada 10 milissegundos, de forma a aplicar o seguidor de linha e verificar os sensores de paragem. Quando um obstáculo é detetado, o estado do movimento passa a ser MOT\_HOLD, iniciando-se a contagem de um *timeout* com duração HOLD\_TIMEOUT, tal como apresentado na Figura 3.14. Quando este *timeout* acaba, o estado do movimento passa para MOT\_TIMEOUT. Quando não é detetada nenhuma linha durante o movimento, o estado do movimento passa para MOT\_ERR.

O módulo RFID permite ler um cartão RFID, obtendo-se um CardID, a sua representação em *string* e o seu tipo, tal como apresentado na Figura 3.16 (a). O enumerado da Figura 3.16 (b) representa o estado do leitor RFID. Quando a leitura é bem-sucedida, o seu estado será MI\_OK. Se houver um erro na leitura ou passar demasiado tempo após o início da leitura, o estado do leitor RFID será MI\_ERR ou MI\_TIMEOUT, respetivamente.

|  |  |
| --- | --- |
| Uma imagem com texto  Descrição gerada automaticamente | Uma imagem com texto  Descrição gerada automaticamente |
| **(a)** | **(b)** |

Figura 3.16 - Módulo RFID: a) Estrutura que define um cartão RFID; b) Estado do leitor RFID.

Relativamente aos módulos da comunicação, o módulo Bluetooth é responsável por receber e executar uma trama via UART, que está conectada a um dispositivo Bluetooth. Na Figura 3.17, está representado, em enumerado, o estado do módulo Bluetooth. Quando uma trama for recebida com sucesso o estado será BLUET\_OK. Se uma trama estiver a ser recebida, o estado será BLUET\_RECEIVING, enquanto quando o estiver pronto para receber, o estado será BLUET\_READY.

Uma imagem com texto

Descrição gerada automaticamente

Figura 3.17 - Estado do módulo Bluetooth.

O módulo Commands define a lista de comandos válidos para esta aplicação. Na Figura 3.18 está representada a estrutura que define um comando, sendo composta pela *string* que define o comando, uma *string* com um texto de ajuda para o comando e a função que o executa (*callback*).

|  |  |
| --- | --- |
|  |  |
| Uma imagem com texto  Descrição gerada automaticamente | |

Figura 3.18 - Definição de um callback de um comando e da estrutura que define um comando.

O módulo Parser permite analisar uma trama, dividindo-a em diferentes *strings* a partir de um delimitador. Além disso, se a trama for um comando válido, este módulo executa a função relativa a este comando. O módulo Cmd Parser permite interpretar um comando, definido no módulo Commands, através do módulo Parser, e executá-lo.

O módulo Route permite criar uma rota com vários pontos de paragem (*checkpoints*). Assim, na Figura 3.19, está representada a estrutura que define um ponto de paragem numa rota, contendo uma *string* com o identificador do *checkpoint* (RFID) e a ação a executar nesse ponto de paragem (*action*)*.*

|  |  |
| --- | --- |
| Uma imagem com texto  Descrição gerada automaticamente |  |
|  | |

Figura 3.19 - Ações possíveis a realizar num checkpoint, e, definição de um checkpoint.

O módulo Debounce efetua o *debounce* de um botão de pressão associado a um pino GPIO, por *software*, através de um algoritmo janela deslizante, tal como mostra a Figura 3.20.

Uma imagem com texto

Descrição gerada automaticamente

Figura 3.20 - Definição de uma janela deslizante.

Uma janela deslizante é um algoritmo que permite armazenar um conjunto de valores, neste caso, um conjunto de *bits* numajanela de 8 *bits* (*window*), e também, saber quantos *bits* a nível lógico alto existem na janela. Como definido na estrutura apresentada na Figura 3.21, o módulo Debounce utiliza uma janela deslizante (*sw)* para que, quando o número de *bits* a nível lógico alto for superior a um certo valor, menor do que o tamanho da janela, coloque o *pin*\_*output,* querepresenta o sinal resultante da aplicação do *debounce* ao pino GPIO, a nível lógico alto. Para isso, este módulo utiliza um *timer* que gera uma interrupção a cada 50 milissegundos.

Uma imagem com texto

Descrição gerada automaticamente

Figura 3.21- Definição da estrutura debounce.

O módulo FSM implementa a máquina de estados, descrita no Capítulo 3.1, que controla a evolução do estado de funcionamento do robô. O enumerado ilustrado na Figura 3.22 apresenta os possíveis estados de funcionamento do DWR.

Uma imagem com texto

Descrição gerada automaticamente

Figura 3.22 - Estados da máquina de estados.

Os estados são executados de forma contínua, até que um estímulo específico faça a máquina de estados transitar para outro estado. Para isso, criou-se um *array* de funções, *fsm\_func\_ptr,* com as funções que implementam os diferentes estados da máquina de estados, tal como mostra a Figura 3.23 (a). Este *array* de funções é acedido com base no valor do estado atual, *state.* Após a execução do estado, o valor do próximo estado, *nstate,* é atribuído ao estado atual, como apresentado na Figura 3.23 (b).

|  |  |
| --- | --- |
| Uma imagem com texto  Descrição gerada automaticamente | Uma imagem com texto  Descrição gerada automaticamente |
| **(a)** | **(b)** |

Figura 3.23 - Definição da máquina de estados (a) Array de funções de estado; (b) Execução da máquina de estados.

### Aplicação de Interface

De modo a ser possível realizar as tarefas de inicialização do DWR e gestão dos percursos a efetuar decidiu-se criar uma aplicação de interface com o utilizador. Além destas tarefas, indispensáveis ao funcionamento do robô, a aplicação servirá também o propósito de informação de estado do DWR, tais como, o último cartão de RFID lido e erros que possam ocorrer durante a sua atividade.

A aplicação foi implementada recorrendo à ferramenta *MIT App Inventor[ref]*, um software simples e útil para o desenvolvimento de aplicações para *smartphones*. A interface criada é de utilização intuitiva. O ecrã inicial está representado na Figura 3.24 (a), no qual é pedido para o utilizador se conectar ao DWR. Para isso, o utilizador terá de ativar a funcionalidade *Bluetooth* do seu próprio dispositivo e emparelhá-lo com o dispositivo *Bluetooth* do DWR [ref]. Depois de emparelhado, é necessário pressionar o botão “*Connect to* DWR*”*.

Com a conexão estabelecida, o funcionário terá acesso ao menu principal da aplicação, apresentado na Figura 3.24 (b). Para realizar um pedido, deve ser pressionado o botão “*Request*” e escolher, a partir da lista apresentada, a rota a efetuar. O botão “*Map*” permite visualizar uma imagem da planta do local onde o robô está a operar. Já a botão permite “*Back*”para voltar ao ecrã inicial. Neste, o utilizador poderá iniciar a marcha do robô, pressionando o botão “*Start*”. Após pressionar este botão, o utilizador receberá as mensagens enviadas pelo DWR, tal como mostra a Figura 3.24 (c).

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | Uma imagem com texto  Descrição gerada automaticamente |
| (a) | (b) | | (c) |

Figura 3.24 - Aplicação Interface (a) Ecra incial; (b) Menu principal; (c) Terminal de mensagens

### Circuito de Controlo

Para a implementação do módulo do seguidor de linha é necessário o desenvolvimento e implementação de um controlador.

#### O que é um Controlador?

Um controlador é responsável pelo controlo de processos através de algoritmos específicos. O seu principal objetivo consiste na monitorização, identificação e interpretação de processos, via modelos matemáticos, de forma a produzir uma ação de controlo conveniente. Existem três ações de controlo distintas: proporcional, integral e derivativa, que podem ser conjugadas entre si. A primeira tem uma ação imediata, proporcional ao valor atual do erro, acelera a resposta de um processo controlado, reduz o tempo de subida e o erro máximo. No entanto, aumenta o overshoot, o tempo de estabilização e produz um offset inversamente proporcional ao ganho. A ação integral produz uma ação de controlo gradual proporcional à integral do erro, respondendo, assim, ao passado do erro enquanto este for diferente de zero, elimina o offset e reduz o tempo de subida. Porém, aumenta o overshoot, o período de oscilação e tempo de estabilização, produzindo respostas lentas e oscilatórias. A ação derivativa produz uma ação antecipatória e proporcional à derivada do erro. É usada para acelerar e estabilizar a malha, reduzir o overshoot, o erro máximo e o período de oscilação. Contudo, não é indicada para processos com ruído.

Na Figura 3.12, estão presentes as três ações de controlo descritas. A Figura 3.12 (a) mostra a resposta de um sistema a uma ação proporcional. A saída deste corresponde à variável de erro multiplicada por uma dada constante. A Figura 3.12 (b) mostra a resposta de um sistema a uma ação integral, onde a saída deste corresponde à integral da variável de erro. Como a integral de uma constante é uma reta com declive não nulo, quando a entrada (variável de erro) é do tipo degrau, a resposta do sistema vai corresponder a uma rampa de declive igual à amplitude da variável de entrada multiplicada por uma constante. A Figura 3.12 (c) mostra a resposta de um sistema a uma ação derivativa, em que a saída deste corresponde à derivada da variável de erro. Como a derivada de uma reta é uma constante, quando a entrada é do tipo rampa, a saída do sistema corresponde a um degrau de amplitude igual à amplitude da rampa multiplicada por uma constante. Um controlador que conjugue as três ações é denominado por controlador proporcional integral derivativo (PID).

|  |  |  |
| --- | --- | --- |
|  |  |  |
| (a) | (b) | (c) |

Figura 3.25 - Ações de controlo (a) ação proporcional; (b) ação integral; (c) ação derivativa

#### Análise do sistema de controlo

O primeiro passo num projeto de controlo está relacionado com a compreensão qualitativa do sistema. É necessário compreender como o sistema a controlar funciona fisicamente, quais as variáveis medidas, a controlar, de atuação, distúrbio e comando, bem como o funcionamento dos atuadores, sensores e controlador.

O sistema a ser controlado assemelha-se a um paralelepípedo com uma roda posicionada na parte central de cada lateral do robô. Sendo o objetivo do sistema seguir uma linha, que pode conter trajetórias retas ou curvilíneas, a velocidade de rotação dos motores terá de variar de modo a ser possível ajustar as velocidades de translação e de rotação do centro de massa do robô. Se se pretender que o robô siga uma trajetória retilínea, os motores terão de rodar à mesma velocidade. Se se pretender que o robô efetue uma trajetória curvilínea, o motor do lado oposto ao que se pretende efetuar a trajetória terá de ter uma velocidade de rotação superior. Ou seja, quando a trajetória é uma curva à esquerda, o motor do lado direito terá de ter uma velocidade de rotação superior comparativamente com a do motor do lado direito. Quando a trajetória é uma curva à direita, passa-se o oposto.

As variáveis medidas são as leituras efetuadas pelos dois sensores do seguidor de linha, o sensor 3 e o sensor 6, Figura 2.2. As variáveis a controlar são a velocidade de translação e velocidade de rotação do centro de massa do robô. As variáveis de atuação são os binários dos motores. As variáveis de comando são a fração de modulação do amplificador PWM de cada motor. A variável de perturbação será o atrito provocado pela superfície (binário de perturbação).

O atuador é composto pelo motor DC (ref), a ponte H(ref) e pelo microcontrolador(ref). O binário produzido pelo motor DC pode ser alterado através da variação da tensão de alimentação do motor (produzida por um amplificador PWM).

O sistema de controlo é responsável pela variação deste parâmetro, de maneira a produzir o binário adequado à velocidade requerida para que o DWR siga a linha corretamente. Em conjunto, o microcontrolador e a ponte H implementam o amplificador PWM. O algoritmo de controlo implementado no microcontrolador produzirá na saída a variável de comando, a fração de modulação do amplificador de PWM, sendo o ganho deste amplificador correspondente a tensão aplicada ao motor DC quando a fração de modulação é um.

O controlador é efetuado através do microcontrolador. A regra de controlo executada por este controlador está implementada numa rotina de serviço à interrupção (ISR), que será despoletada por um *timer* com período igual ao período de amostragem escolhido pelo sistema de controlo. Esta ISR terá como parâmetros de entrada os valores das leituras dos dois sensores a utilizar e como saída a fração de modulação de PWM, que servirá de entrada aos amplificadores PWM.

Na Figura 3.13, está ilustrado o sistema de controlo implementado. Sendo o objetivo principal do DWR o seguimento de uma linha, é necessário manter os sensores na parte exterior da mesma. Quando um dos sensores se aproximar da linha, o motor do lado oposto terá de compensar o desvio da trajetória. Pretende‑se que a diferença entre as leituras dos dois sensores, ou seja, a variável de erro, seja nula, portanto, conclui-se que a variável de referência tem o valor zero.

Se o valor de erro for positivo, significa que o sensor esquerdo se encontra mais próximo da linha do que o sensor direito, implicando que o motor direito tenha uma velocidade de rotação superior. À variável de saída do bloco controlador PID soma-se um valor de offset que servirá de variável de comando ao atuador direito. Ao simétrico da variável de saída do bloco PID soma‑se o mesmo valor de offset, que servirá de variável de comando ao atuador esquerdo. Deste modo, o valor da variável de comando do atuador direito será superior ao valor da variável de comando do atuador esquerdo.

Se o valor do erro for negativo, passa-se o oposto. Se a variável de saída do controlador PID for nula não é necessário fazer ajustes de direção e o DWR seguirá o percurso com velocidade de rotação constante em ambos os motores, com valor igual ao valor de offset*.*



Figura 3.26 - Diagrama de blocos do sistema de controlo

#### Controlador implementado

Um controlador PID contínuo é definido pela equação (3.2), em que , e são referidos, respetivamente, como os ganhos proporcional, integral e derivativo, e como a variável de erro ao longo do tempo. A variável de comando, , corresponde ao somatório das ações proporcional, integral e derivativa.

|  |  |  |
| --- | --- | --- |
|  |  | (3.2) |

Para uma implementação em microcontrolador é necessária uma versão digital do algoritmo de controlo. De entre as diversas famílias das regras PID implementou-se o algoritmo de posição. Esta versão consiste em substituir os termos integral e derivativo, da equação (3.2), pelos seus equivalentes discretos: soma e a diferença divida de 1ª ordem, respetivamente. Assim, obtém-se a equação (3.3), em que e são, respetivamente, a ação de controlo e o erro no instante . Em cada instante é calculado o valor real (de posição) do sinal de saída do controlador. No contexto do problema, a variável de erro corresponde à diferença entre as leituras dos dois sensores, aqui representados, simbolicamente, por e , equação (3.4).

|  |  |  |
| --- | --- | --- |
|  |  | (3.3) |
|  |  | (3.4) |

Para efeitos de cálculo, o valor de pode ser escrito de forma mais adequada, calculando o erro apenas uma vez e colocando em evidência as constantes na aproximação à integral e à derivada (equação (3.5)). Pode-se, agora, redefinir os ganhos proporcional, integral e derivativo, equações (3.6), (3.7) e (3.8), respetivamente, obtendo‑se, por substituição, a equação (3.9).

|  |  |  |
| --- | --- | --- |
|  |  | (3.5) |
|  |  | (3.6) |
|  |  | (3.7) |
|  |  | (3.8) |
|  |  | (3.9) |

Para calcular o somatório dos erros, presente na equação (3.9), define-se a variável , equação (3.10), que pode ser escrita de forma recursiva, equação(3.11). Juntando todas as transformações efetuadas, obtém-se, a equação (3.12), uma nova versão da expressão inicial, equação (3.3).

|  |  |  |
| --- | --- | --- |
|  |  | (3.10) |
|  |  | (3.11) |
|  |  | (3.12) |

Uma forma de reduzir os problemas com erros de medida prende-se com a utilização de um filtro passa-baixo na ação derivativa. Na equação (3.13), apresenta-se a expressão desta ação. Calculando a sua transformada em obtém-se a equação (3.14). A aproximação à derivada introduz um zero em e um polo em , no plano. Pode-se deslocar o polo para a direita, no plano , com um filtro passa‑baixo resultando as equações (3.15) a (3.17). A função de transferência tem um zero em , como no caso anterior, mas o polo situa-se em , e não em . A implementação desta função de transferência está representada na equação (3.18). Pode-se então redefinir como apresentado na equação (3.19). Adicionando a componente do filtro passa-baixo à equação (3.12), chega-se à equação (3.20). A segunda reduz-se à primeira se *a* for zero, ou seja, a ação derivativa sem filtro de passa-baixo é obtida com , o que implica que .

|  |  |  |
| --- | --- | --- |
|  |  | (3.13) |
|  |  | (3.14) |
|  |  | (3.15) |
|  |  | (3.16) |
|  |  | (3.17) |
|  |  | (3.18) |
|  |  | (3.19) |
|  |  | (3.20) |

Um dos problemas de algoritmo de posição está relacionado com a possível saturação do valor de Definem-se, então, limites de saturação superior () e inferior (). Se o resultado do cálculo do valor de for superior ao valor do limite de saturação superior, ou inferior ao valor do limite de comparação inferior, é necessário limitar o mesmo aos valores de saturação impostos. Assim, modifica‑se a interrupção para detetar se o transpôs os valores de saturação e, caso tal aconteça, fixa-se o valor de nos mesmos. Além disso, a saturação do atuador pode dar origem a um aumento brusco dos valores dos somatórios dos erros levando a oscilações prejudiciais no valor da variável controlada.

Tendo em conta os aspetos mencionados acima e fazendo uso da equação (3.20), é possível desenhar o fluxograma que permite a implementação do algoritmo de controlo no microcontrolador, Figura 3.14. Como se pode verificar, a atualização do somatório dos erros só é efetuada quando a variável se encontra dentro dos limites de saturação. Caso contrário, a atualização é anulada, o que equivale, em tempo contínuo, a parar a integração do erro, fixando‑se o valor de no valor de saturação respetivo.



Figura 3.27 - Serviço de Rotina à Interrupção do algoritmo PID

Tendo em conta que, para um valor da variável de comando baixo, os motores estão parados, torna‑se necessária a soma de um *offset* a esta variável, de forma que, quando o erro for nulo, ou seja, quando , a variável de comando não seja nula. Por exemplo, para um valor de *offset* igual a 0,7 e , a variável de comando será igual a e a variável de comando será igual a . Desta forma, garante-se que a variável de comando nunca será nula.

## Simulações

# Lista de Componentes

Na Tabela 4.1, listam-se todos os componentes usados para o DWR, bem como a quantidade e o preço associado. É de realçar que nem todos os componentes foram adquiridos pelo preço indicado. Alguns foram emprestados, oferecidos ou adquiridos em condições mais favoráveis. No entanto, optou-se por especificar o preço de mercado.

Tabela 4.1 - Lista de componentes.

|  | Material | | Loja | Quantidade | Preço unitário | Preço total |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Motor Bot'n Roll ONE 300 rpm |  | Bot'n Roll | 2 | 17,50 € | 35,00 € |
| 2 | Par de rodas Bot'n Roll ONE A (D67 mm) |  | Bot'n Roll | 1 | 11,50 € | 11,50 € |
| 3 | Par de hubs Bot'n Roll ONE A |  | Bot'n Roll | 1 | 4,50 € | 4,50 € |
| 4 | Rodas livres  (D25 mm) |  | Leroy Merlin | 2 | 1,69 € | 3,38 € |
| 5 | QTR-8A |  | Bot'n Roll | 1 | 19,90 € | 19,90 € |
| 6 | Sensor Sharp 2Y0A21YK |  | Bot'n Roll | 1 | 11,95 € | 11,50 € |
| 7 | Módulo RFID RC522 | Uma imagem com texto, eletrónica  Descrição gerada automaticamente | Bot'n Roll | 1 | 4,90 € | 4,90 € |
| 8 | Módulo Bluetooth HC-05 | Uma imagem com eletrónica, circuito  Descrição gerada automaticamente | Bot'n Roll | 1 | 6,80 € | 6,80 € |
| 9 | Módulo *driver* de motores L298N |  | Bot'n Roll | 1 | 12,90 € | 12,90 € |
| 10 | Fonte de alimentação 5 V / 3,3 V | Uma imagem com eletrónica  Descrição gerada automaticamente | Bot'n Roll | 1 | 6,5 € | 6,5 € |
| 11 | Step-Down para UBS 5 V | Uma imagem com eletrónica  Descrição gerada automaticamente | Bot'n Roll | 1 | 9,70€ | 9,70€ |
| 12 | Cabo USB A p/ micro USB B |  | Bot'n Roll | 2 | 1,60 € | 3,20 € |
| 13 | Suporte para uma pilha 18650 c/fios |  | Bot'n Roll | 6 | 0,85 € | 5,10 € |
| 14 | BMS para proteção baterias 18650 3S  12,6 V 20 A |  | Bot'n Roll | 2 | 5,50 € | 11,00 € |
| 15 | Pilha LI-ION 18650 3,7 V 2200 mAh 18X65 mm – 22 A MAX |  | Bot'n Roll | 6 | 3,90 € | 23,40 € |
| 16 | Carregador baterias 18650 | Uma imagem com assobio, adaptador, chave  Descrição gerada automaticamente | Bot'n Roll | 1 | 8,90 € | 8,90 € |
| 17 | Cabo DC com conector reto 9,5 mm | Uma imagem com seta  Descrição gerada automaticamente | Bot'n Roll | 1 | 1,70 € | 1,70 € |
| 18 | Suporte Fusível Auto |  | AlfaElektor | 1 | 1,22 € | 1,22 € |
| 19 | Fusível Auto 3A | Uma imagem com isqueiro, caso  Descrição gerada automaticamente | AlfaElektor | 1 | 0,10 € | 0,10 € |
| 20 | STM32F767ZI | Uma imagem com eletrónica, circuito  Descrição gerada automaticamente | Bot'n Roll | 1 | 30,90 € | 30,90 € |
| 21 | PCB Shield STM32F767ZI | Uma imagem com texto, eletrónica  Descrição gerada automaticamente | Guimo circuito | 1 | 63,96 € | 63,96 € |
| 22 | Conectores Buchanan 284517-4 | 4P TERMIBLOK HDR DIR POL-284517-4 | TE Connectivity | 12 | 1,19 € | 14,26 € |
| 23 | Conectores Buchanan 284507-4 | Uma imagem com brinquedo  Descrição gerada automaticamente | TE Connectivity | 12 | 2,77 € | 33,24 € |
| 24 | Barra (2x8) fêmea 2,54 mm PCB | Uma imagem com texto  Descrição gerada automaticamente | Mauser | 1 | 0,30 € | 0,30 € |
| 25 | Barra (2x10) fêmea 2,54 mm PCB |  | Mauser | 1 | 0,40 € | 0,40 € |
| 26 | Barra (2x16) fêmea 2,54 mm PCB | Uma imagem com música  Descrição gerada automaticamente | Mauser | 1 | 0,62 € | 0,62 € |
| 27 | Barra (2x17) fêmea 2,54 mm PCB | Uma imagem com texto  Descrição gerada automaticamente | Mauser | 1 | 0,69 € | 0,69 € |
| 28 | Barra (2x3) fêmea 2,54 mm PCB | Uma imagem com contentor, caixa  Descrição gerada automaticamente | Mauser | 2 | 0,16 € | 0,32 € |
| 29 | Barra (1x6) fêmea 2,54 mm PCB |  | Mauser | 1 | 0,09 € | 0,09 € |
| 30 | Bloco terminal 12 conetores |  | Gmlux | 1 | 0,40 € | 0,40 € |
| 31 | Condensador poliéster |  | AlfaElektor | 4 | 0,10 € | 0,40 € |
| 32 | Botão de painel 12 mm | Uma imagem com luz  Descrição gerada automaticamente | Bot'n Roll | 1 | 0,60 € | 0,60 € |
| 33 | Interruptor de painel redondo | Uma imagem com eletrónica, preto  Descrição gerada automaticamente | Bot'n Roll | 1 | 1,20 € | 1,20 € |
| 34 | Alumínio 2 mm | Alumínio: Chapa de ALUMÍNIO em fms.pt | **-** | **-** | **-** | 35,00 € |
| 35 | Placa madeira | Placa de contraplacado 2500x1220x10mm - Leroy Merlin Portugal | **-** | 1 | **-** | 5,00 € |
| 36 | Placa de acrílico policarbonato |  | **-** | 1 | - | 10,00 € |
| 37 | Parafuso 4 mm |  | Ferritrofa | 50 | 0,05 € | 2,50 € |
| 38 | Parafuso 3 mm cabeça chata | Uma imagem com utensílios de metal, parafuso  Descrição gerada automaticamente | Ferritrofa | 13 | 0,05 € | 0,65 € |
| 39 | Parafuso 6 mm | Uma imagem com utensílios de metal, parafuso  Descrição gerada automaticamente | Ferritrofa | 6 | 0,05 € | 0,30 € |
| 40 | Parafuso 3 mm auto-roscante | Uma imagem com parafuso, utensílios de metal  Descrição gerada automaticamente | Ferritrofa | 6 | 0,04 € | 0,24 € |
| 41 | Porca com asas 4 mm |  | Ferritrofa | 8 | 0,10 € | 0,80 € |
| 42 | Porca 4 mm |  | Ferritrofa | 32 | 0,05 € | 1,60 € |
| 43 | Porca 3 mm | Uma imagem com projetor  Descrição gerada automaticamente | Ferritrofa | 13 | 0,05 € | 1,15 € |
| 44 | Rebites |  | Ferritrofa | 18 | 0,10 € | 1,80 € |
| 45 | Anilhas | Resultado de imagem para anilha | Ferritrofa | 12 | 0,01 € | 0,12 € |
| 46 | Kit Conectores 2,54 mm | Uma imagem com eletrónica  Descrição gerada automaticamente | Bot'n Roll | *pack* | 10,95 € | 10,95 € |
| 47 | Conjunto condutores flexíveis (1 m) |  | Bot'n Roll | 4 | 1,00 € | 4,00 € |
| Total: | | | | | | 367,69 € |

# Circuito Mecânico Implementado

O objetivo do DWR é transportar um tabuleiro com alimentos e/ou medicamentos. Sendo a forma dos tabuleiros tradicionais retangular, optou-se por definir a forma da armadura do robô como um paralelepípedo, com dimensões aproximadas às dos tabuleiros: 300 mm de largura por 420 mm de comprimento. De forma a ser possível aceder e ver os circuitos no interior do DWR, usou-se, na parte superior da armadura, uma placa de acrílico policarbonato transparente e removível. Para a construção da restante armadura usou-se uma folha de alumínio de 2 mm. Para a fixação de todas as peças, usaram-se parafusos de 3 mm e 5 mm, porcas de 5 mm, porcas com asas de 5 mm, anilhas e rebites.

Relativamente às rodas, foram usadas duas rodas motoras e duas rodas livres. As rodas motoras, com 67 mm de diâmetro, são usadas nas laterais do robô e estão acopladas aos dois motores [REF1]. Os seus pneus são de borracha, com 26,5 mm de largura. As rodas livres (com 25 mm de diâmetro), tal como o próprio nome indica, rodam livremente em qualquer sentido e têm o propósito de dar estabilidade ao robô. Os suportes destas rodas foram colocados na dianteira e traseira da armadura, permitindo o ajuste da distância das rodas livres ao chão. Foi usado um suporte de altura ajustável para prender o *QTR-8A* à armadura do robô. Como *array* de sensores deve estar na parte inferior da dianteira do robô, o suporte deste foi fixo na dianteira da armadura. O suporte, de altura ajustável, para o leitor RFID está posicionado no centro da parte inferior da base do robô. O botão de interação com o utilizador, de 12 mm de diâmetro, está colocado na parte superior esquerda da lateral direita da armadura do robô. Na parte inferior da lateral direita, existe, ainda, a entrada do carregador [REF2] e um botão ON/OFF [REF3], com 20 mm de diâmetro, que permite desligar o DWR.

Na Figura 6.1 (a), é possível observar a vista superior do robô, onde a borda de 20 mm da armadura permite fixar a placa de acrílico através de parafusos. Na Figura 6.1 (b), mostra-se a fixação dos suportes das rodas, motores e sensor de linha e leitor RFID. Na Figura 6.1 (c), é possível observar o mecanismo de ajuste da altura das rodas livres, do sensor de linha e do leitor RFID. Além disso, vê-se uma placa colocada no fundo da armadura com o propósito de suportar todos os circuitos e impedir contactos destes com a armadura. No canto superior esquerdo da armadura do DWR, está localizado o botão de interação com o utilizador e no canto inferior esquerdo estão localizados a entrada de carregamento e o botão ON/OFF. Nas Figura 6.1 (d) e Figura 6.1 (e), podem ver-se a vista dianteira e vista traseira do robô, respetivamente.

|  |  |
| --- | --- |
|  | Uma imagem com texto, interior, forno, minibar  Descrição gerada automaticamente |
| **(a)** | |
|  | Uma imagem com texto, de madeira, madeira  Descrição gerada automaticamente |
| **(b)** | |
|  | Uma imagem com texto, interior, altifalante  Descrição gerada automaticamente |
| **(c)** | |
|  |  |
| **(d)** | |
|  |  |
| **(e)** | |

Figura 5.1 - Desenho (à esquerda) e imagem real (à direita) do DWR (a) vista superior; (b) vista inferior; (c) vista lateral direita; (d) vista dianteira; (e) vista traseira.

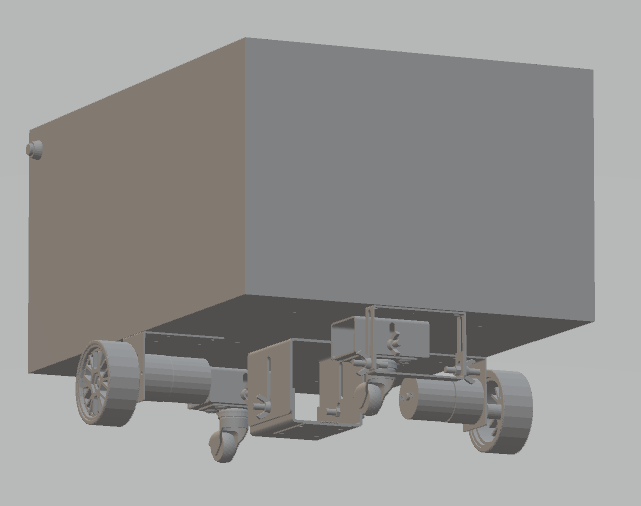


Figura 5.2 - Modelo 3D do DWR

# Resultados Experimentais

De forma a testar as especificações acima previstas, foram realizados ensaios experimentais simulando o ambiente no qual o robô irá operar.

Começou-se por testar o sensor e verificou-se que as medidas dos sensores não atingem nem 0 V nem + 5 V.

Em seguida, montou-se o circuito de atuação dos sinais PWM de modo a testar o seu funcionamento. Para isso, variou-se o sinal na entrada de *feedback* do TL494, através do ajuste de um potenciómetro, obtendo-se os valores de *duty cycle* à saída em função da tensão de entrada, como apresentado na Tabela 6.1.

Tabela 6.1 - *Duty cycle* do sinal PWM de saída em função da tensão de entrada.

|  |  |
| --- | --- |
| Vcomando [V] | *Duty cycle* [%] |
| 1,15 | 0 |
| 1,39 | 10 |
| 1,9 | 30 |
| 2,3 | 50 |
| 2,9 | 70 |
| 3,35 | 90 |
| 3,6 | 97 |

Depois de validado este circuito, ligou-se a saída deste ao *driver* [10] para testar a resposta do motor. Concluiu-se que o motor apenas possui binário de arranque suficiente quando o *duty cycle* do sinal PWM ronda os 50 %. Já com o motor em funcionamento, o *duty cycle* pode atingir um mínimo de 30 % sem que os motores parem. Estes dados foram retirados com os motores em vazio.

# Análise do Produto

## Introdução

Em todos os projetos práticos é necessária uma análise do produto, em áreas como a fiabilidade, segurança e certificação do sistema. Esta análise permite identificar pontos de falha do equipamento e os potenciais perigos a estes associados, de forma a informar o utilizador.

## Fiabilidade

Um sistema ou equipamento diz-se “*fiável quando está livre de erros catastróficos, é capaz de recuperar de erros e apresenta resultados previsíveis (determinismo)*” [15]. Apesar de, neste caso, não ser possível fazer uma avaliação do tipo quantitativa, realizou-se uma avaliação do tipo qualitativa por meio do estudo dos modos de falha e as suas consequências para o sistema e, também, influência do ambiente e do tempo.

Relativamente aos modos de falha, o desenho do sistema deve ter em conta as condições em que o robô se desvia da linha preta. Assim, este deverá ter uma velocidade adequada, uma vez que o sistema pode não responder suficientemente rápido a uma variação. Para diminuir a probabilidade de erros no seguimento da linha, deve ser favorecido o uso de linhas retas em todo o percurso e, quando necessário mudar de direção, devem ser feitas curvas ligeiras. Em acrescento, a grossura da linha deve ser ligeiramente inferior à distância entre os dois sensores usados para seguir a linha, ou seja, 5 cm no máximo.

Do ponto de vista do ambiente que envolve o robô, devem ter-se em conta as condições climatéricas, interações com o utilizador, condições do equipamento e obstrução da via de passagem do robô. O percurso não deverá ser obstruído com objetos, sob pena de danificar tanto o robô como o objeto que o obstrui ou pessoas que se encontrem próximas. Apesar de o DWR ter um sensor de obstáculos, este não cobre toda a área frontal do robô, não garantindo, por isso, total fiabilidade neste quesito. Além disso, sendo o robô um equipamento eletrónico, este não deverá estar sujeito a um ambiente húmido, tal como água ou neve, sujidade e temperaturas extremas. As condições anteriores podem provocar um mau funcionamento dos circuitos constituintes do robô, podendo, em situações extremas, danificar permanentemente o equipamento.

Em relação ao tempo de operação, sabe-se que os componentes têm um tempo médio de vida previsto pelos fabricantes. Visto que não existe informação suficiente para calcular o consumo dos circuitos do DWR, torna-se difícil fazer uma previsão do tempo de vida dos seus componentes com exatidão. Num circuito eletrónico, os componentes com maior probabilidade de falha são os condensadores, semicondutores, baterias e motores. Para quantificar o número de horas previstos de funcionamento médio de um componente, existe um parâmetro denominado *Mean Time Between Failures* –MTBF. Os componentes enunciados têm os MTBFs apresentados na Tabela 7.1. Anormalidades nos circuitos poderão diminuir estes tempos consideravelmente, exigindo uma manutenção ou reparação precoce do robô.

Tabela 7.1 - Tempo de vida dos componentes com maior probabilidade de falha, usados no AWR.

|  |  |
| --- | --- |
| Componente | MTBF |
| Baterias 18650 | 300 – 500 ciclos |
| Semicondutores | 10 anos |
| Condensadores Poliéster | 11,4 anos |
| Motores | 1000 a 3000 horas |

Como calculado no Capítulo 2.6, a autonomia prevista do robô é de duas horas e dez minutos. Se o DWR for usado, por exemplo, na distribuição de alimento, assumindo uma hora de funcionamento por refeição e seis refeições por dia, a bateria deverá ser recarregada, aproximadamente a cada duas refeições, ou seja, três vezes por dia. Calcula-se, assim, que as baterias terão de ser substituídas ao fim de, aproximadamente, três meses, podendo durar, no máximo, até cinco meses e meio.

Conclui-se que as baterias deverão ser os elementos que requerem maior atenção, pois, além das deficiências acima apresentadas, com o aumento do tempo de utilização, estas descarregam, conduzindo a alterações dos resultados práticos.

## Segurança

A interação do robô com o utilizador deve ser realizada em segurança.

Os sistemas elétricos e fios condutores não devem estar em contacto com a armadura do robô que, sendo metálica, é condutora, evitando curto-circuitos ou choques elétricos que podem ser fatais ao sistema e prejudiciais à saúde do utilizador. Para garantir o isolamento da armadura do robô, usou-se, por questões económicas e para validação de conceitos, uma placa de madeira para suportar todos os circuitos do robô. Os motores, o *array* de sensores e leitor RFID, que ficam no exterior da armadura, estão expostos ao utilizador. De forma a isolar os motores eletricamente, usou-se fita isoladora envolvendo todos os terminais destes. Quanto ao *array* de sensores e ao leitor RFID, estes são ligados por conectores do tipo Dupont [REF], ou seja, o isolamento é garantido. Portanto, todos os componentes do robô estão encapsulados e isolados do utilizador. Todavia, sabe-se que este equipamento é um dispositivo sensível à eletricidade estática (frequentemente abreviado como ESD – *Eletrostatic-Sensitive Device*), o que significa que possui componentes que podem ser danificados por cargas elétricas estáticas que se acumulam em pessoas, ferramentas e outros materiais não condutores [16]. Se o utilizador violar o encapsulamento do produto, poderá representar perigos para este ou uma avaria no produto (Figura 7.2 (h)).

Deverá ter-se em conta que o robô possui elementos que apresentam perigos para o utilizador. A madeira usada para isolar os circuitos eletrónicos da armadura do robô é considerada um material comburente (Figura 7.2 (c)) e inflamável (Figura 7.2 (d)). As baterias usadas nos circuitos do DWR, são componentes inflamáveis (Figura 7.2 (d)), explosivos (Figura 7.2 (e)), tóxicos (Figura 7.2 (f)), perigosos para o ambiente (Figura 7.2 (a)), corrosivos (Figura 7.2 (b)) e comburentes (Figura 7.2 (c)). Assim, o robô não deverá estar exposto a qualquer condição ambiental extrema, tal como referido no subcapítulo anteriorrelacionado com a fiabilidade. Devido aos perigos acima descritos, deve ser vigiado o manuseamento do robô por crianças.

## Certificação

Nenhum produto poderá ser exposto ao comércio antes de ser certificado. A certificação é o modo pelo qual uma entidade competente dá uma garantia escrita de que um produto está em conformidade com os requisitos especificados.

A marcação CE, apresentada na Figura 7.1, é um indicativo de conformidade obrigatória para diversos produtos comercializados no Espaço Económico Europeu. Esta marca indica que um produto respeita a legislação da União Europeia em requisitos como segurança, higiene e proteção ambiental, estando, desta forma, credenciado a circular por todo Espaço Económico Europeu [17].



Figura 7.1 - Marcação CE.

De acordo com DIRETIVA 2014/35/UE [18] ANEXO I, que informa os principais elementos dos objetivos de segurança para o material elétrico usado no equipamento, devem ser previstas medidas de ordem técnica a fim de que:

1. As pessoas e os animais domésticos fiquem protegidos de forma adequada contra os riscos de ferimentos ou de outros acidentes resultantes de contactos diretos ou indiretos;
2. Não se produzam temperaturas, descargas ou radiações que possam provocar perigo;
3. As pessoas, os animais domésticos e os bens sejam protegidos de forma adequada contra os riscos de natureza não elétrica provenientes do material elétrico que a experiência venha a revelar;
4. O isolamento seja adequado aos condicionamentos previstos.

Como já foi apresentado, o robô cumpre com os pontos a), b) e d). Uma vez que este projeto serve fins académicos, ou seja, tem o propósito de validar conceitos, apenas foi implementado um sensor de obstáculos na parte dianteira, não garantindo que não embata no objeto, visto que não cobre todo o raio de ação do DWR. Assim, o ponto c) não é cumprido, pelo que o certificado CE não poderá ser atribuído ao DWR.

Tratando-se o DWR de um veículo autónomo, este deve cumprir os requisitos de segurança ditados pelas normas europeias EN ISO-12100 [ref] e ISO-3691-4:2020 [ref], referentes a veículos autónomos. Estes documentos definem vários requerimentos, tais como o nível de segurança do produto, estudo dos riscos e perigos, condições de operação, documentação necessária para a sua utilização e integração, bem como as funcionalidades do sistema de controlo. Dado que o DWR não cumpre algumas destas normas, este não pode obter o certificado europeu de máquinas autónomas.

Devido aos perigos enunciados no subcapítulo anterior, relacionado com a segurança, deverão ser apresentados vários símbolos ao utilizador.

|  |  |  |  |
| --- | --- | --- | --- |
| Uma imagem com texto, símbolo, gráficos de vetor  Descrição gerada automaticamente | Uma imagem com texto, símbolo, exterior, volta  Descrição gerada automaticamente |  |  |
| (a) | (b) | (c) | (d) |
| Uma imagem com texto, símbolo, relógio  Descrição gerada automaticamente | Uma imagem com texto, símbolo  Descrição gerada automaticamente |  |  |
| (e) | (f) | (g) | (h) |

Figura 7.2 - Símbolos de perigo: (a) perigoso para o ambiente; (b) corrosivo; (c) comburente; (d) inflamável; (e) explosivo; (f) tóxico; (g) vários perigos; (h) *Eletrostatic Sensitive Device - ESD*.

O robô será constituído por circuitos e componentes eletrónicos que não podem ser enviados para o lixo doméstico. Estes devem ser encaminhados para locais próprios de reciclagem (Figura 7.3).



Figura 7.3 - Símbolo WEEE.

# Conclusões

## Conclusão

O projeto integrador da unidade curricular de LPI II pôs à prova algumas competências adquiridas ao longo do curso. Aplicaram-se conhecimentos das unidades curriculares das áreas de controlo, eletrónica, máquinas elétricas, instrumentação e sensores, entre outras. Foi um projeto ímpar comparativamente a projetos já realizados porque envolveu um maior conhecimento de várias áreas e requereu muito trabalho autónomo.

Ao longo da conceção do projeto foram sugeridas várias ideias. Algumas destas, foram descartadas numa fase inicial, ao passo que outras apenas foram postas em causa já durante o desenvolvimento do projeto. Num primeiro momento, chegou-se a pensar em fazer a distinção dos quartos e cruzamentos através de um sensor de cores. No entanto, além de ser uma abordagem que poderia trazer resultados pouco precisos, era muito limitadora em relação ao número de combinações únicas. Pensou-se, ainda, em usar a tecnologia RFID apenas nos cruzamentos e marcas de deteção (não identificação) de quartos. Esta marca seria detetada per um dos sensores do *array* do QTR-8A, reduzindo o número de etiquetas RFID que seriam necessárias adquirir. Com base no número de quartos detetados saber-se-ia o quarto em que seria necessário fazer paragem. Porém, chegou-se à conclusão que era uma abordagem que acrescentaria alguma complexidade ao sistema e era mais passível de falhas de leitura ao longo do período de operação do DWR, muito difíceis de identificar e controlar. Para evitar estes problemas, estabeleceu-se que se iria usar a tecnologia RFID em todos os cruzamentos e quartos, permitindo uma identificação precisa e um número de combinações únicas praticamente infinita.

Relativamente ao tempo despendido no desenvolvimento do projeto, visto que todo o grupo se juntou para trabalhar em horários definidos, todos os elementos do grupo trabalharam o mesmo número de horas. Após o desenho e conceção que foi desenvolvido em conjunto, o grupo foi dividido em duas equipas de trabalho, sendo que uma se dedicou à implementação do controlo dos motores e outra ao desenvolvimento da máquina de estados que controla o estado de funcionamento do robô. Assim, foram contabilizadas, individualmente, um total de 250 horas, como mostrado na Tabela 8.1. Em suma, este projeto foi desafiante, revelando a sua importância na formação enquanto futuros engenheiros.

Tabela 8.1 - Número de horas despendidas por elemento

|  |  |
| --- | --- |
| Nome | Número de Horas |
| Bruno Silva | 250 |
| Diogo Fernandes | 250 |
| Duarte Rodrigues | 250 |
| Francisco Salgado | 250 |
| João Miranda | 250 |
| José Abreu | 250 |

## Sugestões de Trabalho Futuro

Durante a realização deste projeto, foram feitas opções que ditaram um rumo. Poder-se‑iam ter tomado outras opções que modificariam os resultados obtidos. Este projeto não representa o fim de uma ideia, é, apenas, uma implementação de um conceito. Assim, neste subcapítulo são feitas algumas sugestões de, não só, melhorias à implementação desenvolvida, mas também de novas abordagens para esta ideia.

Uma das possíveis alterações seria usar células de carga que detetam a colocação de um tabuleiro sobre o DWR ao invés de usar de um botão de pressão para iniciar a marcha. Estas detetariam a colocação e/ou remoção de produtos no DWR enviando sinais ao sistema que agiria em conformidade.

Um dos pontos que poderia ser melhorado, prende-se com o algoritmo de controlo do módulo seguidor de linha. Quando o DWR se encontra numa trajetória reta, a velocidade de rotação dos motores está longe de ser a máxima permitida, mas, com o controlador atual, a velocidade reduzida é necessária para ser possível fazer a compensação nas trajetórias curvas. Num trabalho futuro, seria possível aumentar a velocidade de rotação dos motores quando os últimos valores do erro fossem próximo de nulo para valores mais próximos da velocidade máxima, e reduzir para valores que permitem o ajuste da trajetória quando os últimos valores do erro não fossem próximos de zero. Outras melhorias seriam substituir a tecnologia *Bluetooth* por tecnologia WI-FI, permitindo um maior alcance nas comunicações, e adicionar mais sensores de obstáculos de modo a cobrir uma maior área de deteção.

Além de modificações e melhorias poder-se-iam acrescentar novas funcionalidades ao DWR. Uma delas seria a implementação de um mecanismo de controlo remoto, através do qual uma pessoa responsável controlaria o percurso do robô até ao local pretendido. Teria de ser adicionada uma câmara ao robô, sendo a imagem transmitida para o comando, ou algum dispositivo eletrónico com ecrã, que o utilizador estivesse a utilizar. Outro acréscimo prende-se com a criação de um servidor para uma maior automatização do processo e de algoritmos de mapeamento de percursos com base nas paragens a efetuar. Sendo assim, seria possível controlar uma frota de robôs em simultâneo e no mesmo percurso. Seria também interessante que o DWR pudesse estimar se carga atual das baterias seria ou não suficiente para completar a trajetória pretendida.

Referências

[1] SNS - Serviço Nacional de Saúde, “Covid-19 | Pandemia,” 11 março 2020. [Online]. Available: https://www.sns.gov.pt/noticias/2020/03/11/covid-19-pandemia/. [Acedido em 16 dezembro 2020].

[2] “Conheça Jaci: o robô de desinfecção que auxilia no combate a Covid-19,” Tecnopuc, 29 abril 2020. [Online]. Available: https://www.pucrs.br/tecnopuc/2020/04/29/conheca-jaci-o-robo-de-desinfeccao-que-auxilia-no-combate-covid-19/. [Acedido em 8 fevereiro 2021].

[3] E. L. Brand Talk, “Pandemic and the Smarter World: A Future of Robots?” 5 maio 2020. [Online]. Available: https://www8.gsb.columbia.edu/articles/brand-talk/pandemic-and-smarter-world-future-robots. [Acedido em 10 fevereiro 2021].

[4] R. K. Erico Guizzo, “How Robots Became Essential Workers in the COVID-19 Response,” IEEE SPECTRUM, 30 setembro 2020. [Online]. Available: https://spectrum.ieee.org/robotics/medical-robots/how-robots-became-essential-workers-in-the-covid19-response. [Acedido em 10 fevereiro 2021].

[5] J. D'Onfro, “Robots To The Rescue: How High-Tech Machines Are Being Used To Contain The Wuhan Coronavirus,” 2 fevereiro 2020. [Online]. Available: https://www.forbes.com/sites/jilliandonfro/2020/02/02/robots-to-the-rescue-how-high-tech-machines-are-being-used-to-contain-the-wuhan-coronavirus/?sh=73364f201779. [Acedido em 16 dezembro 2020].

[6] P. Semiconductors, “74HCT14 Hex inverting Schmitt trigger”. 74HCT14 datasheet, 26 agosto 1997 [revisto a 30 outubro 2003].

[7] T. Instruments, “TL494 Pulse-Width-Modulation Control Circui”. TL494 datasheet, janeiro 1983 [revisto a março de 2017].

[8] STMicroelectronics, “DUAL FULL-BRIDGE DRIVER”. L298 datasheet, janeiro 2000.

[9] Texas Instruments, “SN54/74HCT CMOS Logic Family Applications and Restrictions,” maio 1996. [Online]. Available: https://www.ti.com/lit/an/scla011/scla011.pdf?ts=1612201599681&ref\_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCD74HCT251. [Acedido em 2 fevereiro 2021].

[10] T. Corporation, “TENERGY 18650 2200 mAh Li-Ion Cell”. Tenergy 18650 datasheet.

[11] “BMS PARA PROTECÇÃO BATERIAS 18650 3S 12,6V 20A,” [Online]. Available: https://www.botnroll.com/pt/acessorios/2558-bms-para-protec-o-baterias-18650-3s-12-6v-20a.html. [Acedido em 10 fevereiro 2021].

[12] Z. Electromotor. ZGB37RG datasheet.

[13] T. Instruments, “LM340, LM340A and LM78xx Wide VIN 1.5-A Fixed Voltage Regulators”. LM7805 datasheet, setembro 2000 [revisto a julho de 2016].

[14] P. Carvalhal, “Fiabilidade e boas práticas de projeto,” 2014. [Online]. Available: https://elearning.uminho.pt/bbcswebdav/pid-1045855-dt-content-rid-3987823\_1/courses/2021.9305O4\_1/FiabilidadeBoasPraticasProjeto\_PCarvalhal.pdf. [Acedido em 3 fevereiro 2021].

[15] “Electrostatic-sensitive device,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Electrostatic-sensitive\_device. [Acedido em 8 fevereiro 2021].

[16] TUR CERT, “O que é o certificado CE?” [Online]. Available: https://www.ceisaret.com/pt/ce-sertifikasi-nedir/. [Acedido em 2 fevereiro 2021].

[17] Jornal Oficial da União Europeia, “DIRETIVA 2014/35/UE DO PARLAMENTO EUROPEU E DO CONSELHO,” 29 março 2014. [Online]. Available: https://eur-lex.europa.eu/legal-content/PT/TXT/PDF/?uri=CELEX:32014L0035&from=EN. [Acedido em 8 fevereiro 2021].

[18] A. G. e. I. Cardoso, “Fecha tudo. Escolas e universidades em casa a partir de sexta-feira.,” Jornal de Notícias, 21 janeiro 2021. [Online]. Available: https://www.jn.pt/nacional/fecha-tudo-escolas-e-universidade-em-casa-a-partir-de-sexta-feira-13256762.html. [Acedido em 3 fevereiro 2021].

[19] P. Corporation, “pololu.com,” 2001-2014. [Online]. Available: https://www.pololu.com/docs/pdf/0J12/QTR-8x.pdf. [Acedido em 3 fevereiro 2021].

[20] G. Bocock, “Electrolytic Capacitors Determine The Lifetime Of A Power Supply,” XP Power, [Online]. Available: https://www.xppower.com/resources/blog/electrolytic-capacitor-lifetime-in-power-supplies. [Acedido em 11 fevereiro 2021].

[21] “How can I estimate the life of ceramic capacitors?” Taiyo Yuden, [Online]. Available: https://www.yuden.co.jp/eu/product/support/faq/q020.html. [Acedido em 11 fevereiro 2021].

[22] “How to Prolong the Life of an 18650 Battery,” instructables ciruits, [Online]. Available: https://www.instructables.com/How-to-Prolong-the-Life-of-an-18650-Battery/. [Acedido em 11 fevereiro 2021].

[23] A. Perzan, “Brushed vs. brushless DC motors,” drive.tech, [Online]. Available: https://drive.tech/en/stream-content/brushed-vs-brushless-dc-motors. [Acedido em 11 fevereiro 2021].

[24] T. Instruments, “Calculating Useful Lifetimes of Embedded Processors,” 2014. [Online]. Available: https://www.ti.com/lit/an/sprabx4b/sprabx4b.pdf?ts=1612984192026. [Acedido em 11 fevereiro 2021].