|  |
| --- |
| 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  **AWR-19**  ***Analog Waiter Robot*** |
| Projeto integrador  Laboratórios e Práticas Integradas  Trabalho realizado sob a orientação do  **Professor Luís Barros** |
| dezembro de 2020 |

**Índice**

[Lista de Figuras v](#_Toc63096577)

[Lista de Tabelas vii](#_Toc63096578)

[Acrónimos e Siglas ix](#_Toc63096579)

[Capítulo 1 Introdução 11](#_Toc63096580)

[1.1 Introdução 11](#_Toc63096581)

[1.2 Enquadramento 11](#_Toc63096582)

[1.3 Especificações previstas 11](#_Toc63096583)

[1.4 Testes previstos 12](#_Toc63096584)

[1.5 Estudo da fiabilidade 13](#_Toc63096585)

[1.6 Planeamento inicial 13](#_Toc63096586)

[Capítulo 2 Desenho dos circuitos eletrónicos 15](#_Toc63096587)

[2.1 Introdução 15](#_Toc63096588)

[2.2 Controlo dos motores 15](#_Toc63096589)

[2.3 Isolamento e tratamento dos valores dos sensores 18](#_Toc63096590)

[2.4 Sistema eletrónico para o controlo dos motores 20](#_Toc63096591)

[Capítulo 3 Simulação dos circuitos eletrónicos 29](#_Toc63096592)

[3.1 Controlo dos motores 29](#_Toc63096593)

[3.2 Sistema eletrónico para o controlo dos motores 32](#_Toc63096594)

[Capítulo 4 Desenho da implementação dos circuitos eletrónicos 36](#_Toc63096595)

[4.1 Introdução 36](#_Toc63096596)

[4.2 Controlo dos motores 36](#_Toc63096597)

[4.3 Sensores 39](#_Toc63096598)

[4.4 Máquina de estados 41](#_Toc63096599)

[4.5 Lista de componentes 44](#_Toc63096600)

[Capítulo 5 Desenho do encapsulamento 50](#_Toc63096601)

[Capítulo 6 Possíveis opções alternativas para o desenho dos circuitos 52](#_Toc63096602)

[6.1 Introdução 52](#_Toc63096603)

[Capítulo 7 Evolução do atual desenho para um sistema baseado em microcomputador 53](#_Toc63096604)

[7.1 Introdução 53](#_Toc63096605)

[Referências 54](#_Toc63096606)

Lista de Figuras

[Figura 1- Diagrama de Gantt 14](#_Toc63096528)

[Figura 2 - Módulo Driver L298N 15](#_Toc63096529)

[Figura 3 -Esquemático do gerador de onda PWM 16](#_Toc63096530)

[Figura 4 - Esquemático do circuito estabelece a tensão de comparação 17](#_Toc63096531)

[Figura 5 -Esquemático do circuito para os sensores a ser usados como componentes analógicos 19](#_Toc63096532)

[Figura 6 - Esquemático do circuito para os sensores a ser usados como componentes digitais 19](#_Toc63096533)

[Figura 7 - Máquina de Moore 20](#_Toc63096534)

[Figura 8 - Linha preta usada para os testes 21](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096535)

[Figura 9 – Sensores usados na máquina de estados: S1, S5 e S8 21](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096536)

[Figura 10 - Botão de pressão 21](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096537)

[Figura 11 - Diagrama da máquina de estados 22](#_Toc63096538)

[Figura 12 - Lógica do próximo estado 24](#_Toc63096539)

[Figura 13 - Lógica de saída 24](#_Toc63096540)

[Figura 14 - Desenho da máquina de estados 25](#_Toc63096541)

[Figura 15 - Esquema do Flip-Flop J-k 25](#_Toc63096542)

[Figura 16 - Implementação do Oscilador 26](#_Toc63096543)

[Figura 17 - Resposta do Oscilador de 1 kHz 27](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096544)

[Figura 18 - Circuito RC de reset e a sua resposta no tempo. 27](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096545)

[Figura 19 - Circuito de debounce do botão de pressão (quando premido) 28](#_Toc63096546)

[Figura 20 - Circuito gerador PWM com entrada de 0 V 29](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096547)

[Figura 21 - Circuito gerador PWM com entrada de 2,1 V 29](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096548)

[Figura 22 - Circuito gerador PWM com entrada de 3,6 V 30](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096549)

[Figura 23 - Circuito gerador da tensão de comparação com ambas as entradas a 0 V 30](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096550)

[Figura 24 - Circuito gerador da tensão de comparação com entrada do sensor 1 a 5 V e entrada do sensor 2 a 0 V 31](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096551)

[Figura 25 - Circuito gerador da tensão de comparação com entrada do sensor 1 a 2,5 V e entrada do sensor 2 a 0 V 32](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096552)

[Figura 26 - Estado 0 (Parado) 33](#_Toc63096553)

[Figura 27 - Estado 1 (Anda para a frente) – Botão foi pressionado 33](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096554)

[Figura 28 - Estado 2 (Dá a volta para a esquerda) – S1 e S8 a nível lógico alto 34](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096555)

[Figura 29 - Estado 3 (Dá a volta) - S1 e S8 a nível lógico baixo 34](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096556)

[Figura 30 - Estado 0 (Parado novamente) - S5 = 1 35](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096557)

[Figura 31 - Esquemático PCB controlo dos motores 36](#_Toc63096558)

[Figura 32 - Layout PCB controlo dos motores 37](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096559)

[Figura 33 - Vista bottom PCB controlo dos motores 37](#_Toc63096560)

[Figura 34 - Vista top PCB controlo dos motores 38](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096561)

[Figura 35 -Vista 3D PCB controlo dos motores 38](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096562)

[Figura 36 - Esquemático PCB sensores 39](#_Toc63096563)

[Figura 37 - Layout PCB sensores 39](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096564)

[Figura 38 - Vista bottom PCB sensores 40](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096565)

[Figura 39 - Vista 3D PCB sensores 40](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096566)

[Figura 40 - Esquemático PCB Máquina de Estados 41](#_Toc63096567)

[Figura 41 - Layout PCB Máquina de Estados 42](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096568)

[Figura 42 - Vista bottom PCB máquina de estados 42](#_Toc63096569)

[Figura 43 - Vista top PCB máquina de estados 43](#_Toc63096570)

[Figura 44 - Vista 3D PCB máquina de estados 43](#_Toc63096571)

[Figura 45 - Vista superior do robô 50](#_Toc63096572)

[Figura 46 - Vista inferior do robô 50](#_Toc63096573)

[Figura 47 - Vista lateral do robô 51](#_Toc63096574)

[Figura 48 - Vista traseira do robô 51](#_Toc63096575)

[Figura 49 - Vista frontal do robô 51](#_Toc63096576)

Lista de Tabelas

[Tabela 1- Planeamento do Projeto 14](#_Toc63096523)

[Tabela 2 - Tabela completa para determinação da lógica completa da máquina de estados 23](#_Toc63096524)

[Tabela 3 - Atribuição de estados 24](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096525)

[Tabela 4 - Tabela de excitação do Flip-Flop JK 24](file:///C:\Users\user\Desktop\PI-LPI_I\01Relatorios\Relatorio.docx#_Toc63096526)

[Tabela 5 - Lista de Componentes 44](#_Toc63096527)

Acrónimos e Siglas

|  |  |
| --- | --- |
| **Acrónimo/Sigla** | **Significado** |
|  |  |
|  |  |
|  |  |
| AWR  LED  PWM | *Analog Waiter Robot*  *Light emitting diode*  *Pulse with Modulation* |

Introdução

Introdução

Perante o atual panorama pandémico da Covid-19 [1], pretende-se implementar uma ideia que terá um impacto positivo na vida das pessoas infetadas pela doença bem como dos que as rodeiam.

A maioria das ideias nesta área tem como foco principal a saúde pública da população em geral, tais como, robôs de desinfeção, por exemplo. Há menos projetos com foco individual em pessoas que tenham contraído a doença.

Na China existe um robô (*little peanut*) [2] com a mesma finalidade que foi utilizada num hotel para entregar comida porta a porta a hóspedes com suspeita de infeção.

Tendo em consideração que uma pessoa em isolamento deve evitar o contacto com o mundo exterior, pretende-se desenvolver 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.

## Enquadramento

O *Analog Waiter Robot* (AWR) é um robô seguidor de linha focado na assistência a pessoas em isolamento que não podem contactar com o mundo exterior. Basta colocar no seu suporte o que pretende fazer chegar ao paciente, colocá-lo sobre a linha, ligá-lo e este seguirá o percurso até ao destino.

## Especificações previstas

O AWR terá de deslocar-se entre dois pontos, previamente definidos, com base numa linha instalada nos meios de acesso à divisão em que o paciente em questão se situa.

O AWR terá de ser colocado sobre a linha com os bens essenciais (comida e/ou alimentos) num tabuleiro. Após estar alinhado, deverá ser ligado por um funcionário responsável, para que inicie a sua trajetória. Ao chegar ao destino, demarcado com uma cruz no fim da linha, o robô terá de parar de forma a que o paciente recolha os bens a si destinados. Quando o paciente desejar, poderá acionar o robô de forma a que este reinicie o seguimento da linha de volta ao ponto de partida. Para tal, o AWR fará uma rotação de 180 ° sobre a linha e iniciar a marcha até ao ponto de início. Como a alimentação do robô será a baterias, eventualmente, terá de ser ligado à rede elétrica para ser carregado.

O tipo de desenvolvimento do nosso produto pode ser classificado como “misto”, uma vez que não tem necessariamente características inovadoras, mas é concebido para servir as necessidades particulares de um comprador. O produto será composto na sua totalidade por componentes ou subsistemas que são comprados “*off-the-shelf*”. Isto é, que só precisam de ser instalados e configurados para entrarem em funcionamento.

## Testes previstos

De forma a testar as especificações acima previstas, deverão ser realizados testes.

Para que o AWR siga a linha, será implementado um seguidor de linha através de um *array* de sensores de reflexão de luz. Irão ser realizados ensaios experimentais simulando o ambiente no qual este irá operar. Estes permitirão determinar os valores de referência que servirão para o dimensionamento e calibração dos circuitos de direção do robô. Para isso, este deverá ser colocado em diferentes posições sobre a linha.

Depois de determinados os valores de referência, deverão ser testados os circuitos de direção do robô, que usarão controladores para manter os valores dos sensores na gama de referência. Para isso, o AWR deverá ser colocado em diferentes posições sobre a linha para verificar se o controlador cumpre o seu propósito. Isto é, se o robô segue a linha até ao destino final.

Para controlar os diferentes estados de funcionamento do robô, irá ser desenvolvida uma máquina de estados baseada em circuitos sequenciais. De forma a testar o seu funcionamento, esta deverá ser sujeita a diferentes inputs de forma a verificar a comutação entre estados com a utilização de LEDs nas suas saídas. As saídas da máquina de estados irão controlar o estado de funcionamento dos motores.

Consoante as saídas da máquina de estados, cada motor poderá ter três modos de funcionamento: sentido horário, sentido anti-horário ou parado. Nesse momento poderão ser testadas todas as funcionalidades do robô, verificando se segue a linha ou não.

Estudo da fiabilidade

## Planeamento inicial

De forma a testar as especificações acima previstas, deverão ser realizados testes.

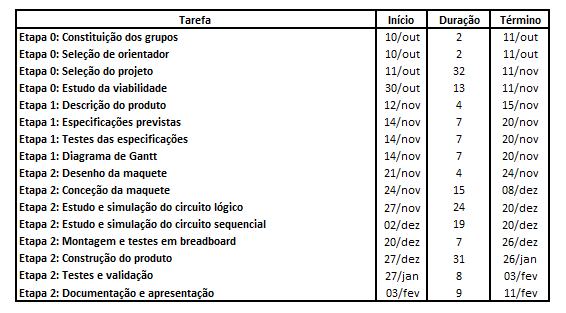
Para que o AWR siga a linha, será implementado um seguidor de linha através de um *array* de sensores de reflexão de luz. Irão ser realizados ensaios experimentais simulando o ambiente no qual este irá operar. Estes permitirão determinar os valores de referência que servirão para o dimensionamento e calibração dos circuitos de direção do robô. Para isso, este deverá ser colocado em diferentes posições sobre a linha.

Depois de determinados os valores de referência, deverão ser testados os circuitos de direção do robô, que usarão controladores para manter os valores dos sensores na gama de referência. Para isso, o AWR deverá ser colocado em diferentes posições sobre a linha para verificar se o controlador cumpre o seu propósito. Isto é, se o robô segue a linha até ao destino final.

Para controlar os diferentes estados de funcionamento do robô, irá ser desenvolvida uma máquina de estados baseada em circuitos sequenciais. De forma a testar o seu funcionamento, esta deverá ser sujeita a diferentes inputs de forma a verificar a comutação entre estados com a utilização de LEDs nas suas saídas. As saídas da máquina de estados irão controlar o estado de funcionamento dos motores.

Consoante as saídas da máquina de estados, cada motor poderá ter três modos de funcionamento: sentido horário, sentido anti-horário ou parado. Nesse momento poderão ser testadas todas as funcionalidades do robô, verificando se segue a linha ou não.

Tabela 1- Planeamento do Projeto



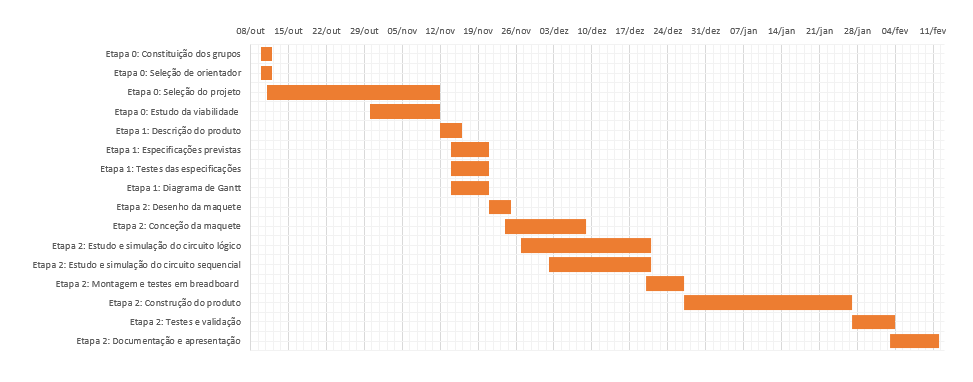


Figura 1- Diagrama de Gantt

Desenho dos circuitos eletrónicos

Introdução

O desenho dos circuitos eletrónicos tem por base um módulo - placa driver de motores com dupla ponte H e base no L298N [3] - que faz a ligação entre os diferentes circuitos.

Este módulo, além das entradas para os motores, possui seis entradas de controlo. Duas controlam a velocidade de rotação do motor respetivo através de uma entrada PWM. As restantes quatro entradas controlam, duas a duas, o estado e sentido de rotação dos motores.

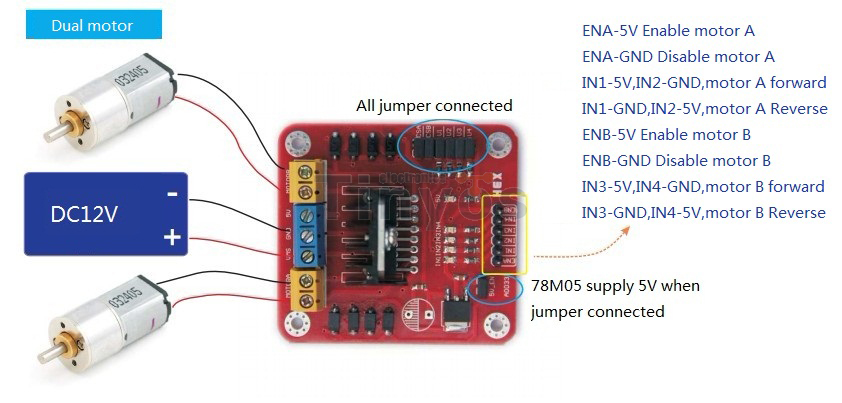


Figura 2 - Módulo Driver L298N

Controlo dos motores

O controlo da velocidade de rotação dos motores está diretamente relacionado com leitura dos valores de tensão obtidos pelo sensor. Quando ambos os sensores estão fora da linha têm de ter velocidade quase máxima. Quando um dos sensores se aproxima da linha preta terá de abrandar sendo o desvio de direção compensado pelo motor oposto.

Como referido anteriormente o controlo de velocidade de rotação dos motores é efetuada através de uma entrada *PWM*. Para gerar este sinal recorre-se a dois circuitos principais. O primeiro estabelece uma tensão que será usada na entrada do segundo como valor de comparação com uma onda triangular e gerar na saída o sinal *PWM* desejado.

Para uma melhor compreensão inicia-se a explicação pelo segundo circuito. Este tem uma finalidade bastante simples: gerar uma onda PWM consoante uma tensão de entrada. Quanto maior o valor deste sinal maior o *duty cycle* à saída. Para realizar este circuito utilizou-*se TL494 Pulse-Width-Modulation Control Circuits* [4]. De realçar que este circuito gera uma onda triangular interna entre 0V e 3,3V (valor teórico - na prática e em simulações verificou-se que este valor se aproxima mais de 3,6V). Sendo assim, a tensão de entrada deste circuito terá de estar compreendida entre estes dois valores para se variar a velocidade do motor. Este circuito tem o seguinte esquemático.

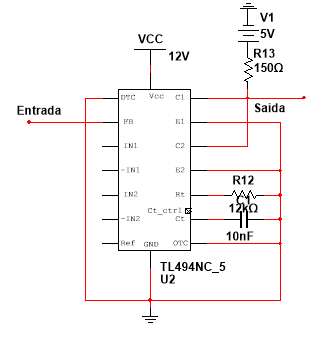


Figura 3 -Esquemático do gerador de onda PWM

A entrada deste circuito é na porta de *feedback.* Às entradas Rt e Ct ligou-se uma resistência de 12KΩ e um condensador de 10 nF, respetivamente. Estes valores definem a frequência da onda triangular gerada. A saída deste circuito tem uma resistência *pull-up* ligada a 5V para que a amplitude da onda PWM gerada tenha este valor.

Para estabelecer a tensão de referência é necessário ter em conta alguns aspetos. O primeiro prende-se com os valores de leitura do sensor. Quando se encontra sobre uma superfície branca mede um valor ligeiramente acima de 0V. Já quando se encontra sobre uma superfície preta mede um valor ligeiramente abaixo de 5V. Sendo um sensor analógico cores intermédias têm um valor lido entre estas duas gamas. Outro aspeto importante está relacionado com a zona de funcionamento do motor. Segundo testes realizados verificou-se que o motor tem uma zona de arranque com *duty cycle* a rondar os 50%. Já depois de se encontrar em funcionamento este o *duty cycle* necessário para manter o motor em funcionamento este valor é ligeiramente inferior. De salientar que estes valores forem obtidos com o motor fora do robô, isto é, sem que a massa deste tivesse influência nas medições. Isto leva a que na prática os valores de *duty cycle* não sejam exatamente estes. Posto isto, a tensão de entrada deste circuito terá de ter no mínimo metade da amplitude da onda triangular gerada internamente pelo circuito integrado para que o motor não pare por completo evitando oscilações. O esquemático deste circuito é o seguinte.

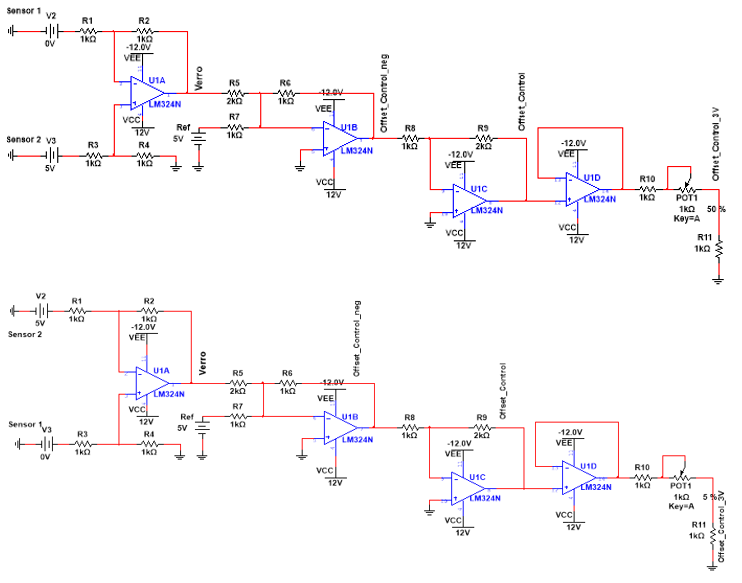


Figura 4 - Esquemático do circuito estabelece a tensão de comparação

Uma vez que cada motor tem de ser controlado individualmente é necessário implementar dois circuitos tal como apresentado na figura acima. Estes circuitos apenas diferem no facto de as entradas de um estarem trocadas em relação às entradas do outro.

Quando nenhum sensor se encontra sobre a linha a saída do circuito será igual nos dois casos. Quando um sensor se começa a aproximar da linha a saída de um dos circuitos aumenta o valor e a do outro diminui permitindo ao robô fazer a compensação devido e continuar o percurso desejado.

O primeiro estágio deste circuito é um subtrator e estabelece a diferença entre a tensão à entrada não inversora com a tensão à entrada inversora, ou seja, a diferença entre os dois sensores. Como cada um destes valores pode variar entre 0V e 5V, o valor da tensão de erro terá um valor máximo de 5V e um valor mínimo de -5V. Tendo em consideração as características de funcionamento do motor acima descritas é necessário que o valor mínimo à saída deste circuito seja aproximadamente metade da amplitude da onda triangular de comparação. Assim, é necessário somar uma tensão de *offset* a estes valores.

O segundo e terceiro estágios são, respetivamente, um somador inversor e um amplificador inversor. Em conjunto, estes dois estágios somam uma tensão de *offset* de 10V à saída do primeiro estágio, fazendo com a que a tensão à saída do terceiro estágio tenha um valor compreendido entre 5V e 15V.

O quarto estágio é um seguidor de tensão e tem como objetivo isolar os dois circuitos.

O último estágio é um divisor de tensão. Permite estabelecer uma tensão de saída do circuito dentro dos valores pretendidos. Colocou-se um potenciómetro com o objetivo de permitir corrigir e compensar diferenças entre os resultados obtidos experimentalmente e os resultados reais devido às tolerâncias dos componentes bem como conseguir um ajuste mais fino da velocidade.

Isolamento e tratamento dos valores dos sensores

Este circuito tem dois objetivos principais. O primeiro objetivo é isolar os valores lidos dos sensores do restante circuito obtendo-se, assim, proteção e valores mais fiáveis. O segundo objetivo é fazer um tratamento dos valores a ser usados pelo circuito digital. Num circuito digital apenas interessam dois tipos de valores: nível lógico alto e nível lógico baixo. Atendendo às características dos circuitos integrados utilizados é considerado nível lógico alto quando os valores de tensão atingem 2V. Uma tensão deste valor lida pelo sensor corresponde a uma cor muito mais próxima do branco do que do preto impossibilitando o robô de fazer o pretendido. Por este motivo, é necessário fazer um tratamento dos valores dos sensores que vão ser usados no circuito digital. Após passar o valor do sensor por um seguidor de tensão introduz-se este valor num comparador não inversor fazendo com que o valor à saída deste comparador seja 0V até um valor de entrada mais próximo do valor de referência da cor preta e seja 5V a partir deste valor.

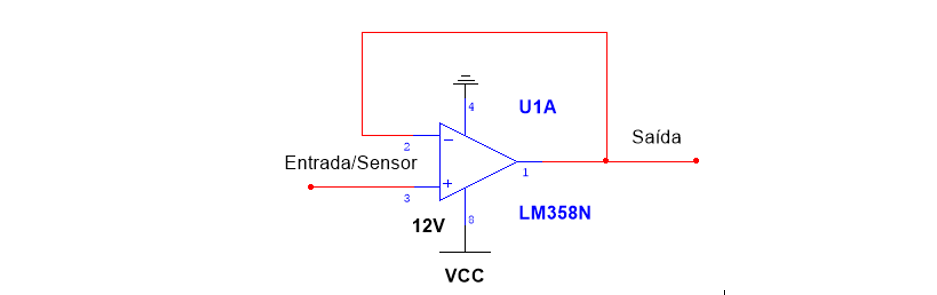


Figura 5 -Esquemático do circuito para os sensores a ser usados como componentes analógicos

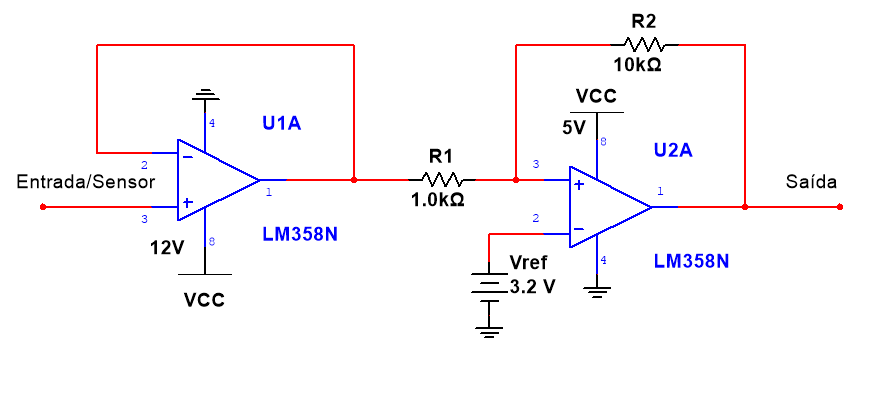


Figura 6 - Esquemático do circuito para os sensores a ser usados como componentes digitais

Fazendo e considerando o valor teórico de . Na prática este valor será ligeiramente inferior, no entanto não é um valor crítico para o funcionamente do circuito uma vez que, tal como referido anteriormente, o nível lógico alto dos circuitos integrados é atingido a partir de um valor de tensão de 2V. Além do mais as medidas dos sensores nunca são ideais. Os cálculos apenas são usados como referência. Após determinadas as resistências a usar e de estimado um valor para a tensão de referência esta pode ser ajustada para um valor que se melhor adapta aos resultados práticos.

Fazendo tem-se que.

Fazendo bem que e , ou seja, a saída do comparador só fica a nível lógico alto quando a leitura do sensor passa os 4V e apenas volta a nível lógico baixo quando a leitura do sensor desce dos 3V.

Sistema eletrónico para o controlo dos motores

De forma a ser possível controlar o estado de funcionamento do robô, ou seja, fazê-lo seguir a linha, parar ou rodar 180 °, desenvolveu-se uma máquina de estados sequencial síncrona, mais especificamente uma máquina de Moore, uma vez que as suas saídas dependem apenas do estado atual.



Figura 7 - Máquina de Moore

Numa máquina de estados, o objetivo é determinar a “Lógica do Próximo Estado” e a “Lógica de Saída” do circuito. Como referido, a “Lógica de saída” depende apenas do estado atual. Já a “Lógica do Próximo Estado” depende do estado atual e das entradas do circuito.

Como referido, o objetivo do robô é seguir uma linha, devendo parar sobre uma linha horizontal colocada no final do percurso (Figura 8) e, nesse momento, o robô deve dar uma volta de 180 ° sobre o seu eixo, voltando a seguir a linha em direção ao ponto de partida, após o acionamento de um botão.



Figura 8 - Linha preta usada para os testes

Denominou-se a linha de final de curso como a linha horizontal e a linha determinante do percurso como linha vertical.

De modo a detetar a linha no final do percurso, usaram-se os dois sensores das pontas do *array* de sensores (S1 e S8, rodeados a amarelo na Figura 9). Usou-se também o sensor do meio (S5, rodeado a verde na Figura 9) de forma a interromper a rotação quando este encontrar a linha preta.



Figura 9 – Sensores usados na máquina de estados: S1, S5 e S8

Nesse momento, o robô poderá ser acionado através de um botão de pressão (B) (Figura 10) para que este volte a seguir a linha.



Figura 10 - Botão de pressão

Os motores são controlados por meio de um *driver* (L298N), podendo estar parados, a rodar no sentido horário ou no sentido anti-horário. O *driver* possui quatro entradas, duas por cada motor.

Para isso, na máquina de estados existem duas saídas por cada motor, sendo que uma controla o movimento no sentido horário e outra no sentido anti-horário.

Como o robô usa dois motores, existem quatro saídas, que quando estão ativas (nível lógico alto) despoletam as seguintes respostas:

• FWR – motor da direita roda no sentido horário (*Forward* *Right*);

• FWL – motor da esquerda roda no sentido horário (*Forward* *Left*);

• BWR – motor da direita roda no sentido anti-horário (*Backwards* *Right*);

• BWL – motor da esquerda roda no sentido anti-horário (*Backwards* *Left*).



Figura 11 - Diagrama da máquina de estados

Na figura seguinte, mostra-se o diagrama da máquina de estados, usada para controlar o modo de funcionamento do robô. FW representa os valores de FWR e FWL e BW representa os valores de BWR e BWL.

Inicialmente, no estado “A”, o robô está parado (FW = 0 e BW = 0), e espera pelo acionamento do botão para iniciar a marcha. Após o acionamento do botão, B a nível lógico alto, no estado “B”, o robô inicia a sua trajetória, seguindo a linha com os dois motores a rodar no sentido horário (FW = 1 e BW = 0). Quando o *array* de sensores estiver posicionado sobre a linha horizontal preta (fim do trajeto), os sensores S1 e S8 estarão a nível lógico alto, e ocorrerá a transição para o estado “C”. Neste estado, o robô inicia a rotação para a esquerda (FWR = 1, FWL = 0, BWR = 0 e BWL = 1) e não transita de estado enquanto os sensores S1 e S8 estiverem a nível lógico alto, isto é, enquanto o *array* de sensores estiver sobre a linha horizontal. Quando o *array* de sensores sair desta, S1 e S8 estarão a nível lógico baixo, avançando para o estado “D”, no qual o robô continua a rotação, mantendo as suas saídas, até que o sensor S5 esteja sobre a linha vertical, S5 a nível lógico alto. Assim posicionado, o robô volta ao estado inicial.

De forma a implementar a máquina de estados, de acordo com o diagrama de estados, determinou-se toda a lógica necessária, construindo a Tabela 2.

Tabela 2 - Tabela completa para determinação da lógica completa da máquina de estados



Adicionaram-se duas colunas que indicam os bits do estado atual (B1 e B0) e também as entradas da máquina de estados (S1, S5, S8 e B). Como existem quatro entradas, existem dezasseis (24) combinações possíveis para cada estado, o que resulta em sessenta e quatro possibilidades no total (24+2 = 26 = 64) que será o número de linhas da Tabela 2

A seguir, “E” representa o estado atual e “E\*” o estado seguinte. “B1\*” e “B0\*” representam os bits do estado seguinte. Estas colunas devem ser preenchidas de acordo com a tabela ao lado (Tabela 3).

Tabela 3 - Atribuição de estados



Sabendo que cada *flip-flop* permite a existência de dois estados e que existem quatro estados (A – D), selecionaram-se dois *flip-flops* do tipo J-K, que apresentam a tabela de excitação ao lado (Tabela 4). De acordo com esta tabela, foram preenchidas as colunas “J” e “K” para cada *flip-flop* (Ja e Ka, relativo a B0; Jb e Kb, relativo a B1). Para este efeito, determinou-se a lógica do próximo estado, em função das quatro entradas da máquina de estados, simplificada através de mapas de Karnaugh. Resolveram-se quatro mapas de Karnaugh (um para cada entrada J e K dos *flip-flops*) e obteve-se os resultados apresentados na Figura 12



Tabela 4 - Tabela de excitação do Flip-Flop JK



Figura 12 - Lógica do próximo estado

Adicionaram-se as colunas das saídas (FWR, FWL, BWR e BWL), as quais foram preenchidas de acordo com o estado atual. Para cada uma destas determinou-se a lógica de saída, que também foi simplificada através de mapas de Karnaugh. Os resultados obtidos foram os da Figura 13



Figura 13 - Lógica de saída

De acordo com a lógica determinada, desenhou-se o circuito que implementa a máquina de estados, o qual foi testado usando a ferramenta *MultiSim*, verificando-se a sua funcionalidade.

Lógica de próximo estado

Entradas



Lógica de saída

Saídas

Flip-Flops

Estado atual

Lógica de Entrada

Figura 14 - Desenho da máquina de estados

Tal como mostra a Figura 15, cada *flip-flop* J-K necessita de um sinal *clock* como entrada. O *clock* é um sinal em forma de onda quadrada com um *duty* *cycle* de, idealmente, 50 %. Dá-se uma transição entre estados sempre que, a cada pulso de *clock*, uma das entradas (J ou K) estiver a nível lógico alto.



Figura 15 - Esquema do Flip-Flop J-k

Para este efeito, implementou-se um oscilador usando o integrado NE555 numa montagem como multivibrador astável. O condensador C1 carrega até 2\*Vcc/3 através das resistências RA e RB (saída = 1). Quando C1 atinge este valor, irá descarregar por RB (saída=0), até uma tensão de Vcc/3 e o ciclo repete-se.



Figura 16 - Implementação do Oscilador

Através dos seguintes cálculos, dimensionaram-se os componentes para obter uma onda quadrada de frequência 1 kHz, *duty cycle* de 50 %, aproximadamente.

Para e :

Foram selecionados os valores standard das resistências (Série E24) e do condensador (Série E12) mais próximos dos obtidos:

A Figura 17 mostra a saída do circuito obtida em simulação com a ferramenta *TINA*.

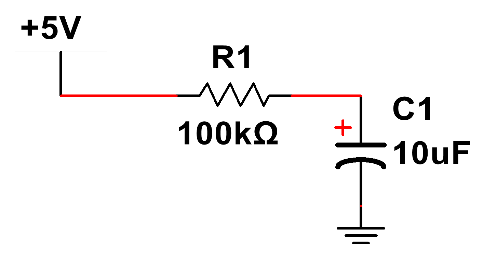
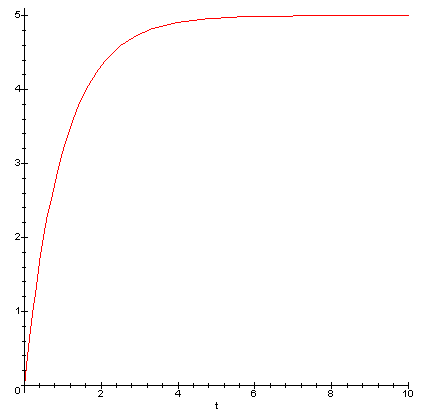


Figura 17 - Resposta do Oscilador de 1 kHz

As entradas Clear dos Flip-Flops J-K utilizados são ativas a nível lógico baixo, isto é, se ligarmos esta entrada ao *ground*, ocorrerá um *reset* no *flip-flop* e sua a saída Q virá a nível lógico baixo, independentemente das suas entradas J e K. Assim, se na máquina de estados as entradas *Clear* forem ligadas ao *GND*, B0 e B1 ficarão a nível lógico baixo.

Experimentalmente, verificou-se que, quando alimentados, os flip-flops mantêm as saídas do estado anterior. Para que o robô inicie no primeiro estado de funcionamento (estado “A”), utilizou-se um circuito para efetuar um *reset* aos *flip-flops*, tal como mostra a figura 10.

Figura 18 - Circuito RC de reset e a sua resposta no tempo.



Sabendo que a saída deste circuito estará a nível lógico baixo enquanto o condensador não carregar, a entrada *reset* dos *flip-flops* estará também a nível lógico baixo, o que resulta na ação de *reset*. Assim, logo que se alimentar a máquina de estados, os *flip-flops* apresentarão nas saídas “Q” o nível lógico baixo que representa o primeiro estado de funcionamento do robô. Após o condensador carregar, a entrada *reset* estará a nível lógico alto durante o resto do tempo de funcionamento, não provocando um *reset* indesejado.

Foi usado um botão de pressão para que o robô inicie a marcha. No entanto, sabe-se que, quando uma saída provém diretamente de um botão de pressão, esta é afetada por problemas de vibração mecânica (*bounce*) quando o contacto se fecha. De forma a garantir que a mudança de nível lógico se faça sem oscilações, torna-se necessário implementar um circuito de *debounce* como, por exemplo, o apresentado na Figura 19

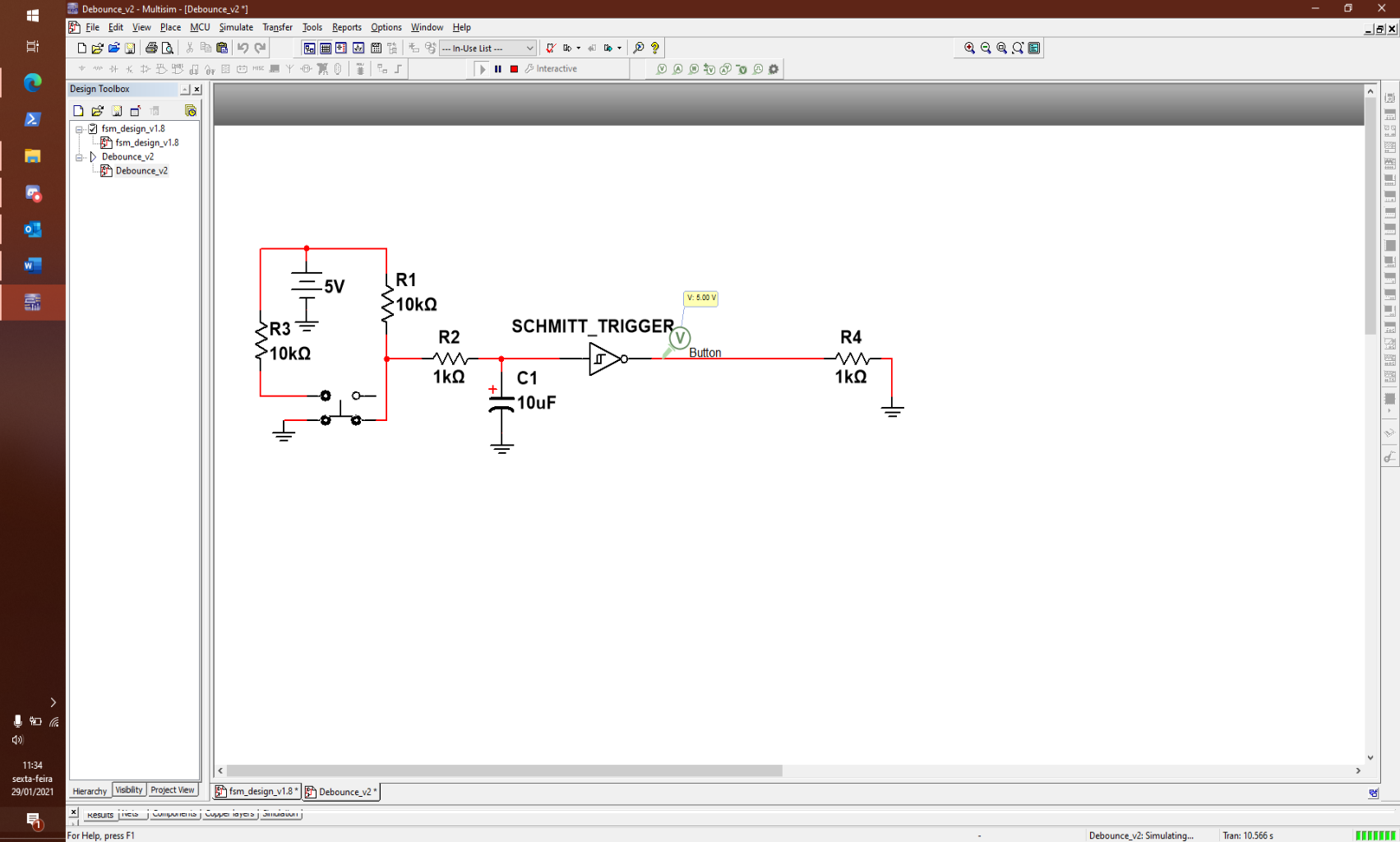


Figura 19 - Circuito de debounce do botão de pressão (quando premido)

O condensador em conjunto com a resistência funciona como filtro passa-baixo, que ajuda a reduzir o ruído elétrico provocado pelas vibrações mecânicas do botão. Assim, quando o botão não está pressionado, os terminais do condensador apresentarão 5 V e, quando pressionado, apresentarão 0 V. Uma vez que, num circuito sequencial, o nível lógico alto corresponde ao acionamento de uma entrada (botão, neste caso), será necessário utilizar uma lógica inversora na saída do condensado. Para isso, usou-se uma porta lógica Schmitt-Trigger (integrado 74HCT14) que funciona como um circuito comparador inversor com histerese centrado em 0 V. Assim, aumenta a imunidade ao ruído e, como inverte, a saída do circuito estará a nível lógico alto quando o botão for pressionado.

Simulação dos circuitos eletrónicos

Controlo dos motores

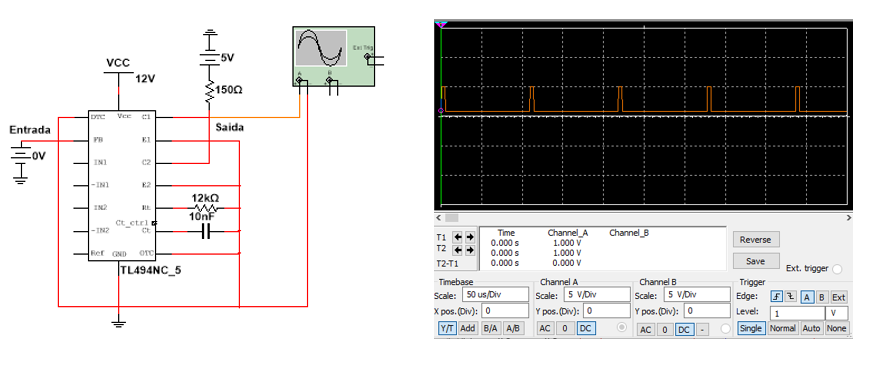
Tal como no desenho e explicação do circuito no capítulo anterior ir-se-á começar a simulação dos circuitos eletrónicos pela segunda parte do circuito, ou seja, pelo circuito que gera a onda *PWM* quando lhe é fornecida uma tensão de entrada.

Figura 20 - Circuito gerador PWM com entrada de 0 V

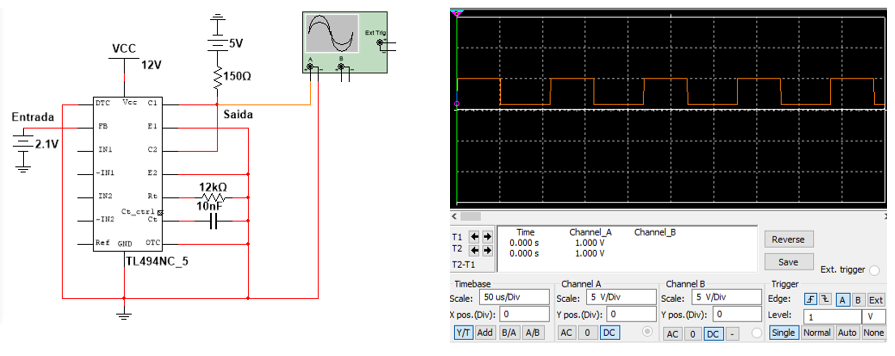


Figura 21 - Circuito gerador PWM com entrada de 2,1 V

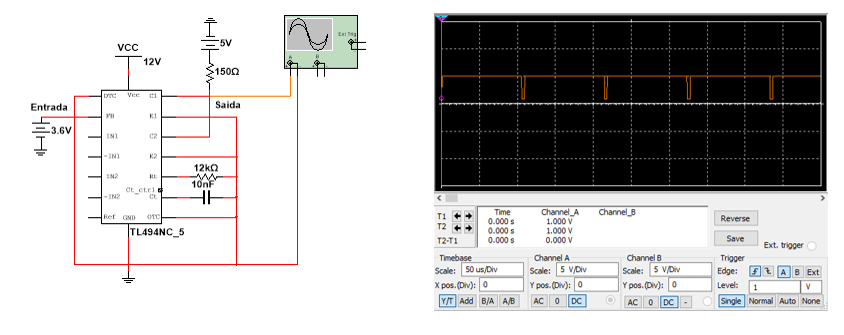
 Através das simulações acima pode-se concluir que quanto maior a tensão de entrada maior o *duty cycle.* Quando a entrada é zero o *duty cycle* é aproximadamente zero. Quando é próxima da amplitude da onda triangular interna do circuito integrado TL494 o *duty cycle* é praticamente 100 %.

Figura 22 - Circuito gerador PWM com entrada de 3,6 V

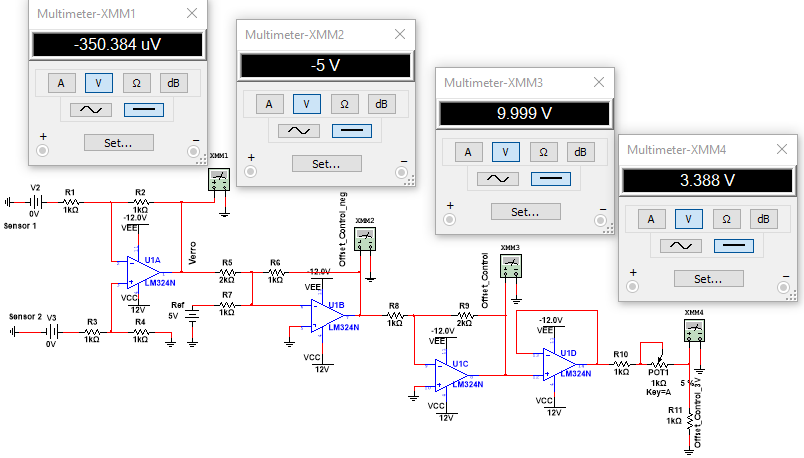
Apresenta-se de seguida as simulações do circuito que geram a tensão de entrada a ser usada no circuito acima.

Figura 23 - Circuito gerador da tensão de comparação com ambas as entradas a 0 V

Como podemos ver na figura acima, quando nenhum dos sensores está sobre a linha preta a tensão de saída tem um valor próximo do máximo o que implicará um *duty cycle* próximo dos 100% e o robô seguirá a mesma trajetória.

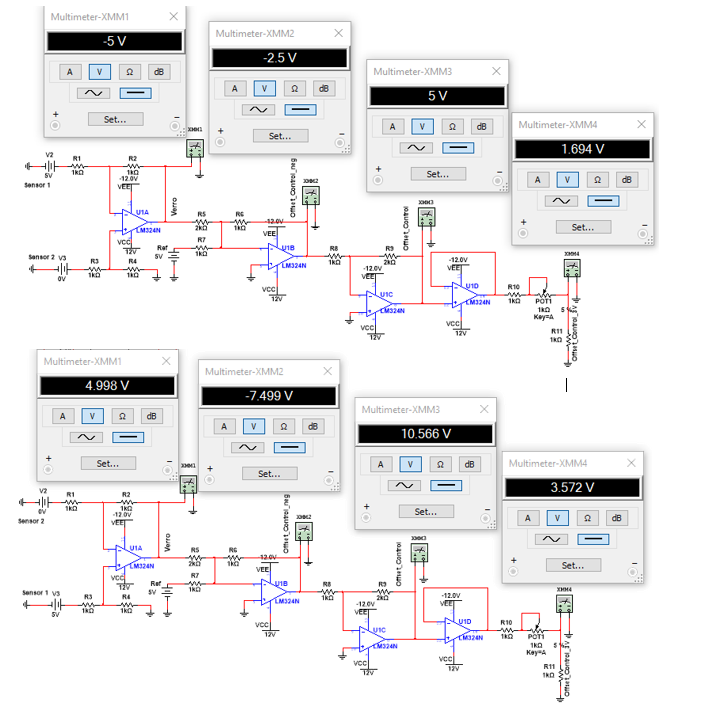


Figura 24 - Circuito gerador da tensão de comparação com entrada do sensor 1 a 5 V e entrada do sensor 2 a 0 V

Quando um dos sensores se aproxima da linha preta – sensor 1 neste caso – o motor que se encontra do lado deste sensor terá de abrandar para que o robô se desvie da linha. Como se pode verificar, o valor da tensão de saída do primeiro circuito desce e o motor tende a abrandar. No lado oposto, como as entradas dos circuitos estão trocadas, verifica-se o contrário. O motor terá de acelerar (ou manter a velocidade) para que se aproxime da linha. É de referir que a saída *offset\_control* tem um valor de 10,566 V uma vez que que se encontra limitado pela saturação do **Amp**. **Op**. O valor esperado seriam 15 V, no entanto em nada interfere com o normal funcionamento.

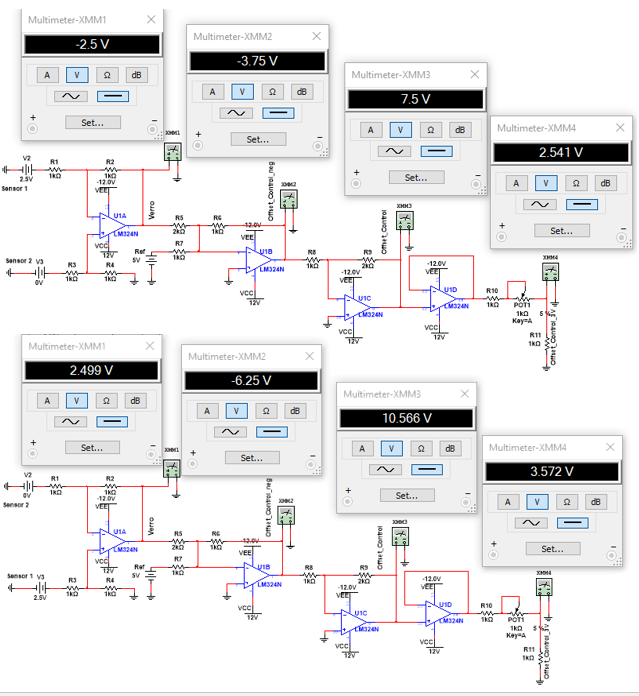


Figura 25 - Circuito gerador da tensão de comparação com entrada do sensor 1 a 2,5 V e entrada do sensor 2 a 0 V

Este caso demonstra uma situação mais real, visto que o circuito corrige antes de chegar a situações de valores extremos, e pode ser analisado como no último caso.

Sistema eletrónico para o controlo dos motores

Foi feito o teste em simulação do circuito da máquina de estados através da ferramenta *MultiSim*. Os resultados obtidos apresentam-se nas figuras abaixo. Verificou-se, portanto, a boa funcionalidade do circuito dimensionado.



Figura 26 - Estado 0 (Parado)

Figura 27 - Estado 1 (Anda para a frente) – Botão foi pressionado



Figura 28 - Estado 2 (Dá a volta para a esquerda) – S1 e S8 a nível lógico alto



Figura 29 - Estado 3 (Dá a volta) - S1 e S8 a nível lógico baixo



Figura 30 - Estado 0 (Parado novamente) - S5 = 1



Desenho da implementação dos circuitos eletrónicos

Introdução

Todos os circuitos acima descritos serão implementados em placas de circuito impresso. Para o efeito fez-se uso de duas ferramentas: o *PADS LOGIC* e *PADS LAYOUT*. Para simular o design da PCB em 3D utilizou-se a ferramenta KiCad.

Controlo dos motores

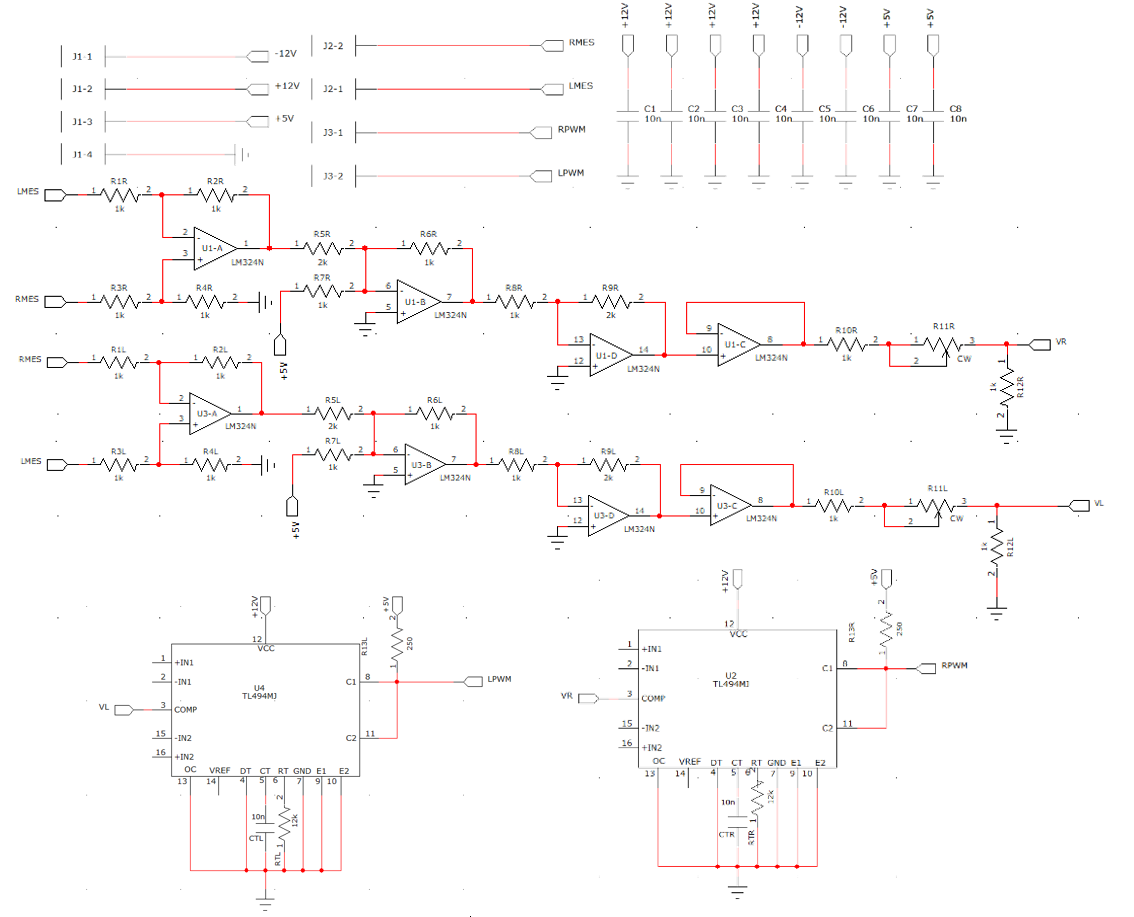


Figura 31 - Esquemático PCB controlo dos motores

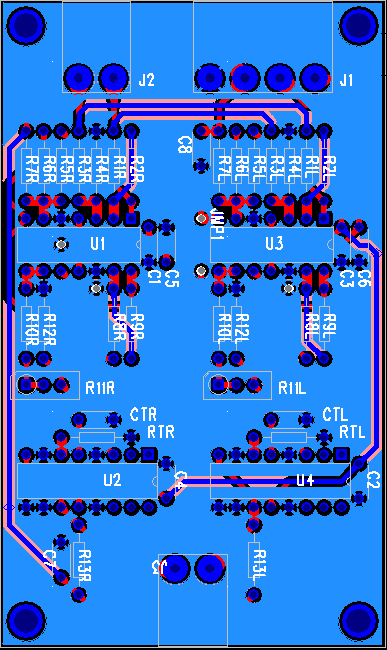


Figura 32 - Layout PCB controlo dos motores

Figura 33 - Vista bottom PCB controlo dos motores

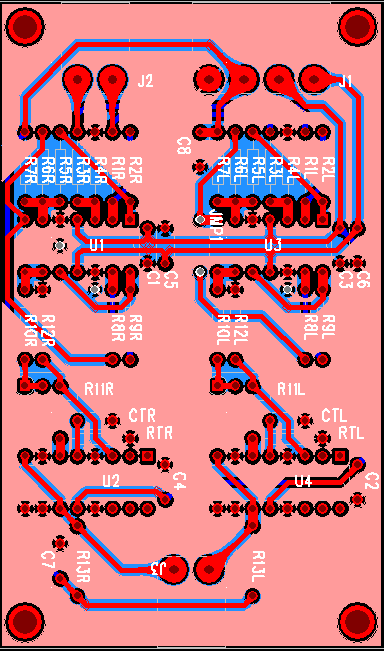


Figura 34 - Vista top PCB controlo dos motores



Figura 35 -Vista 3D PCB controlo dos motores

Sensores

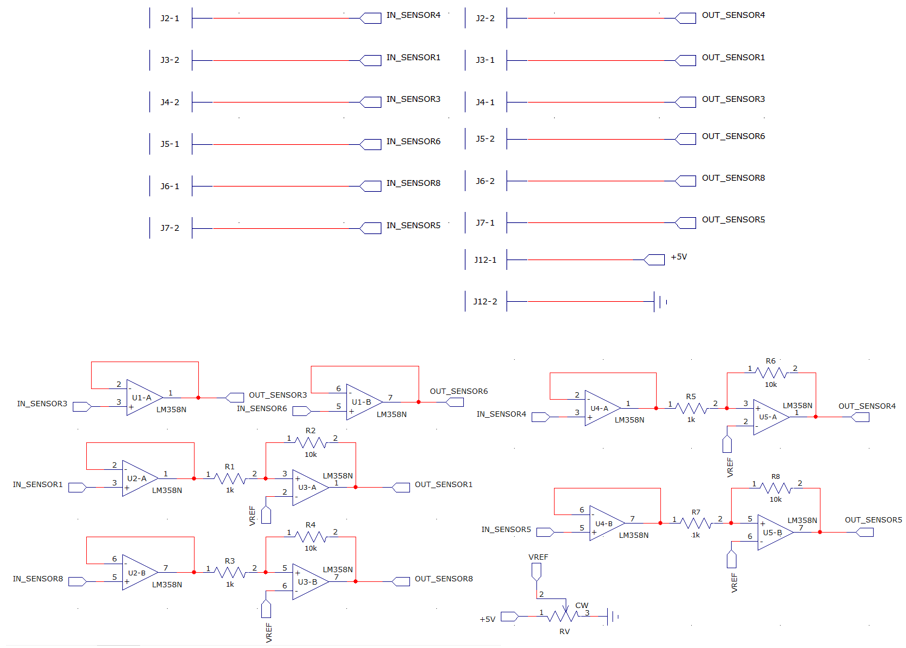


Figura 36 - Esquemático PCB sensores



Figura 37 - Layout PCB sensores

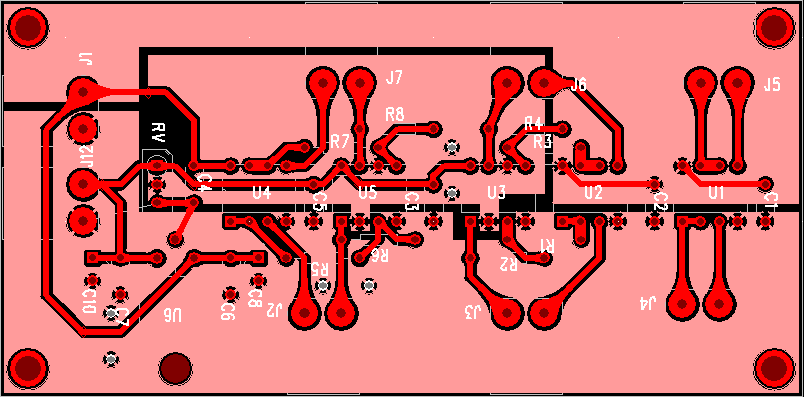


Figura 38 - Vista bottom PCB sensores



Figura 39 - Vista 3D PCB sensores

Máquina de estados

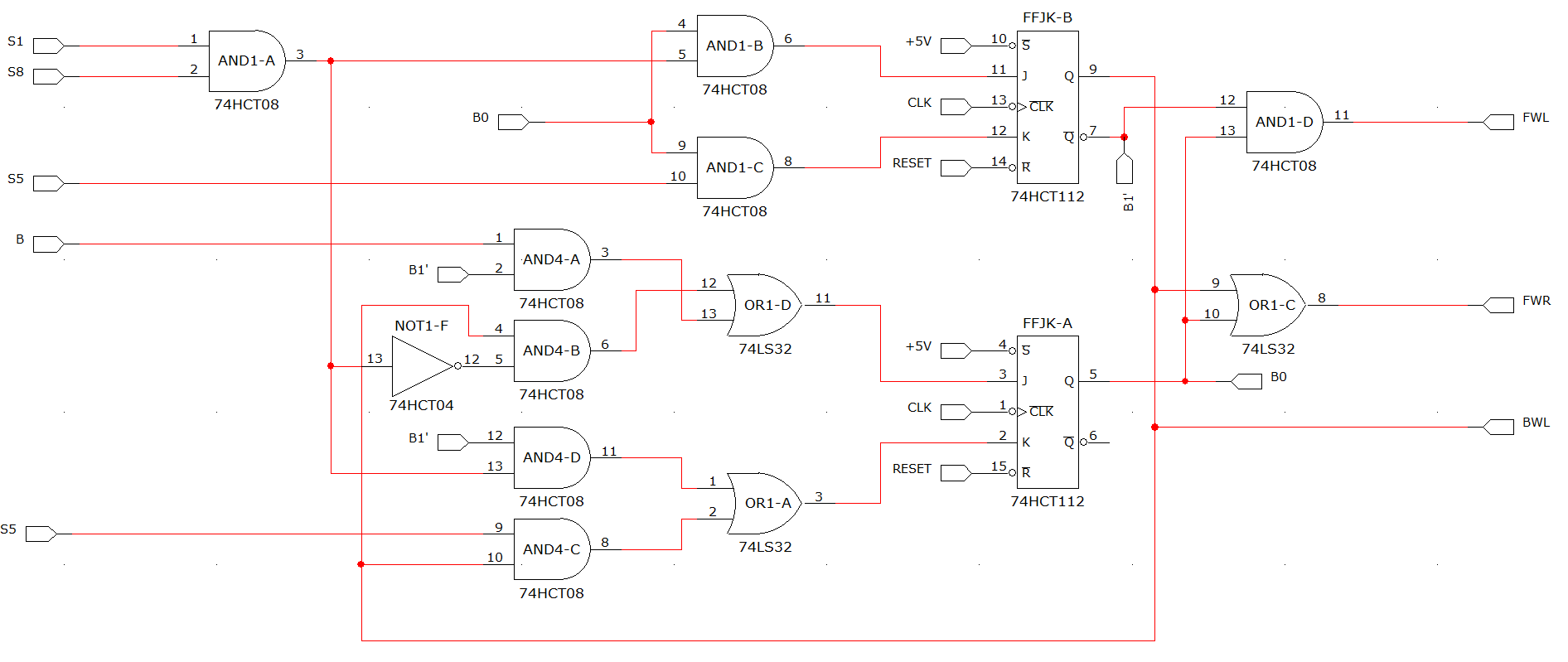
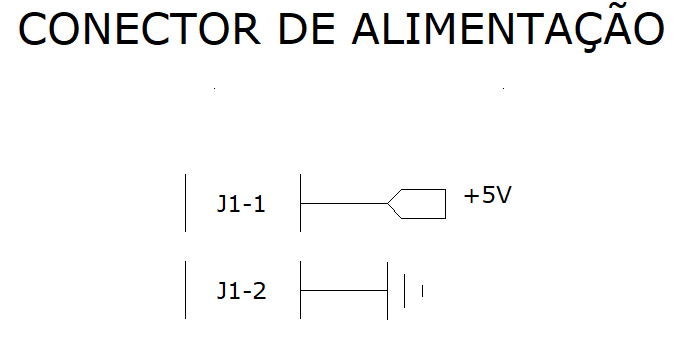
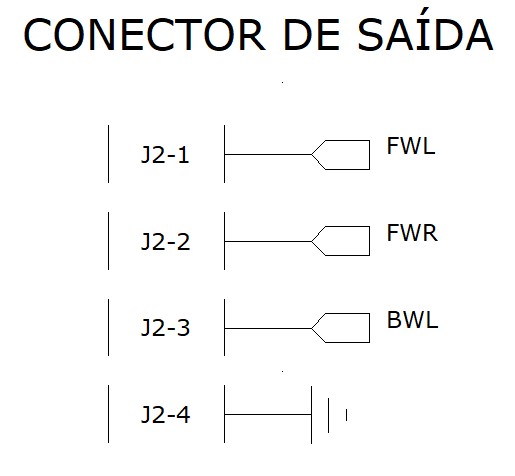
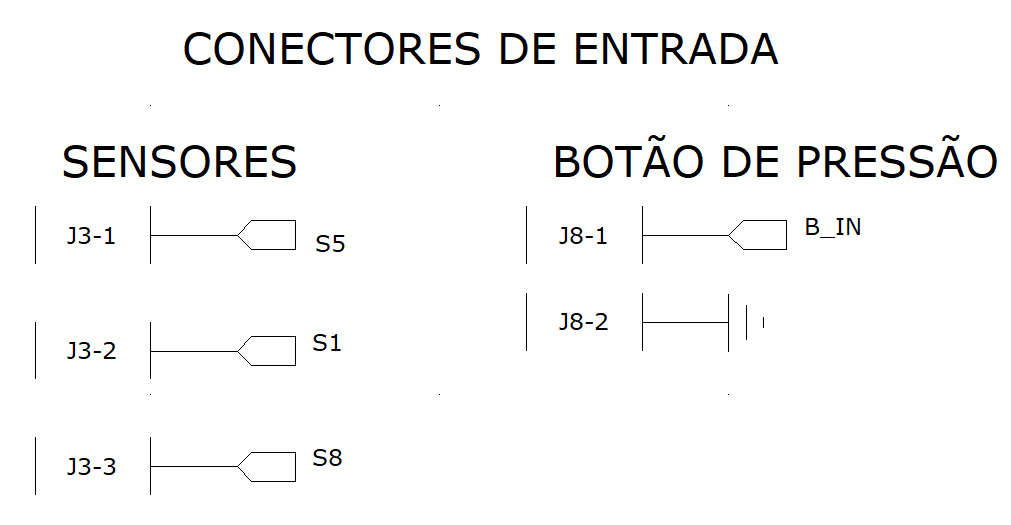


Figura 40 - Esquemático PCB Máquina de Estados



Figura 41 - Layout PCB Máquina de Estados

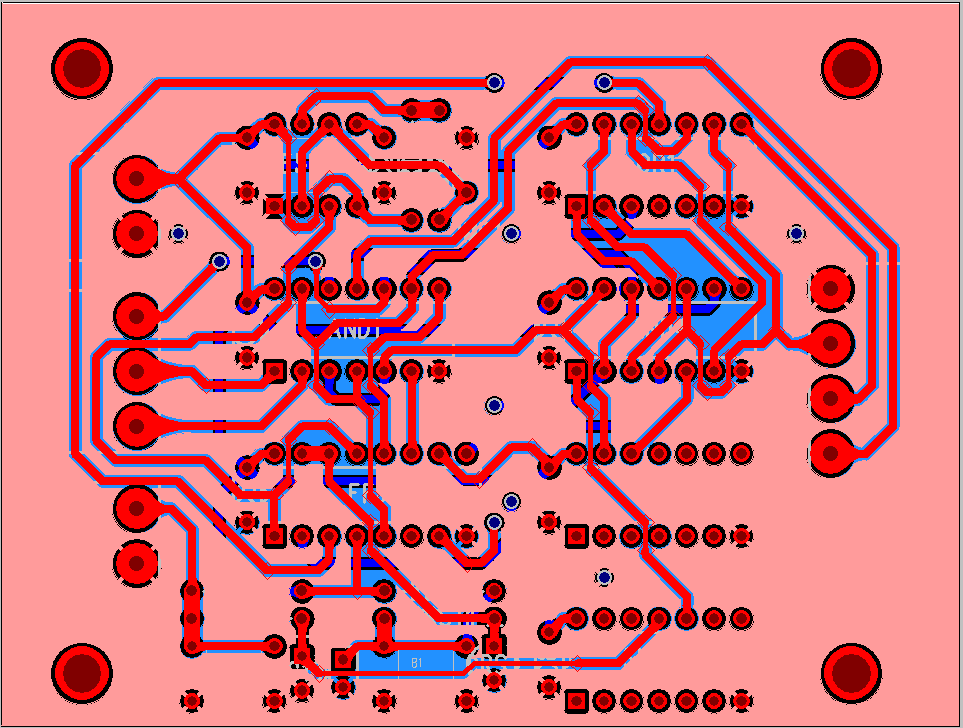


Figura 42 - Vista bottom PCB máquina de estados



Figura 43 - Vista top PCB máquina de estados



Figura 44 - Vista 3D PCB máquina de estados

Lista de componentes

Tabela 5 - Lista de Componentes

| Material | | Loja | Quantidade | Preço unitário | Preço total |
| --- | --- | --- | --- | --- | --- |
| Motor Bot'n Roll ONE 300rpm |  | Botnroll | 2 | 17,5 € | 35 € |
| Par de rodas Bot'n Roll ONE A |  | Botnroll | 1 | 11,5 € | 11,5 € |
| Par de hubs Bot'n roll ONE A |  | Botnroll | 1 | 4,5 € | 4,5 € |
| Placa driver de motores L298N |  | Botnroll | 1 | 12,9 € | 12,9 € |
| Sensor QTR8A |  | Botnroll | 1 | 19,9 € | 19,9 € |
| Suporte para uma pilha 18650 c/fios |  | Botnroll | 6 | 0,85 € | 5,1 € |
| BMS para proteção baterias 18650 3S 12,6V 20A |  | Botnroll | 2 | 5,5 € | 11 € |
| Pilha LI-ION 18650 3,7V 2200mAh 18X65mm - 22A MAX |  | Botnroll | 6 | 3,9 € | 23,4 € |
| LM7805 |  | Botnroll | 1 | 0,5 € | 0,5 € |
| Rodas Livres  (D25MM) |  | Leroy Merlin | 2 | 1,69 € | 3,38 € |
| TL494 |  | Aquário | 2 | 1,70 € | 3,40 € |
| LM324 | Texas Instruments LM324 LM324N DIP14 Quadruple Operational Amplifier DIP14  1 Pack: Amazon.com: Industrial & Scientific | Aquário | 2 | 1,95 € | 3,9 € |
| LM358 | TEXAS INSTRUMENTS LM358P Lm358 Dual Operational Amplifier, For Electronics,  8 Pin Dip, Rs 5 /piece | ID: 21935084862 | Aquário | 5 | 0,74 € | 3,7 € |
| Bloco Terminal 2 Pinos |  | Aquário | 12 | 0,4 € |  |
| Potenciómetro trimmer multivolta |  | Aquário | 2 | 2,2 € | 4,4 € |
| Condensador Cerâmico |  | Aquário | 19 | 0,1 € |  |
| Condensador eletrolítico |  | Aquário | 2 | 0,25 € |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Desenho do encapsulamento

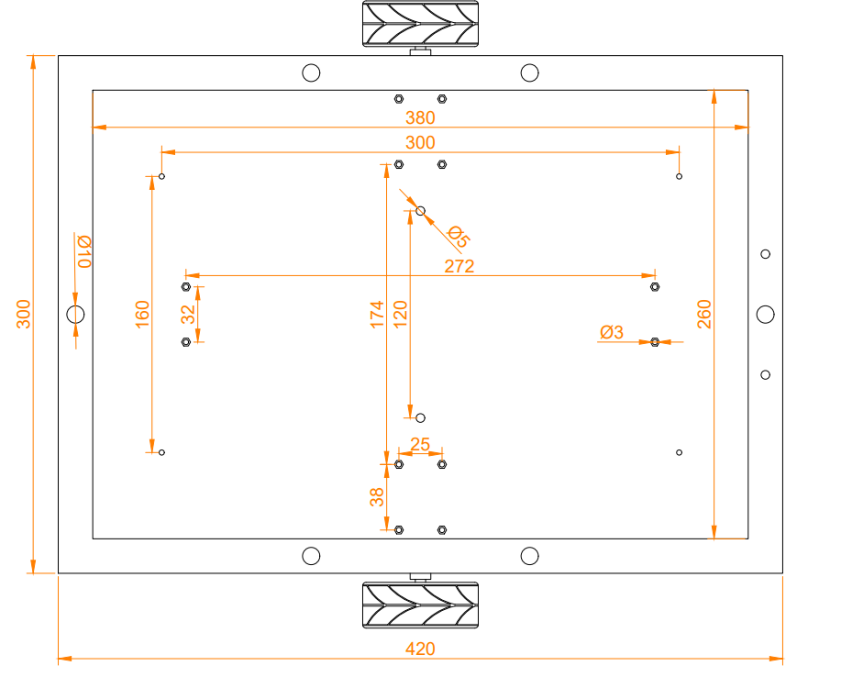


Figura 45 - Vista superior do robô

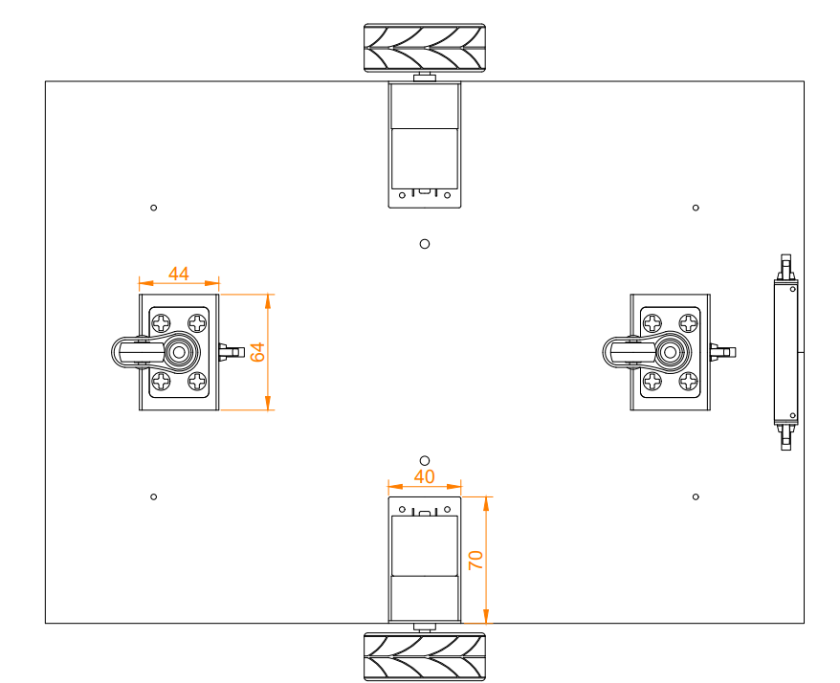


Figura 46 - Vista inferior do robô

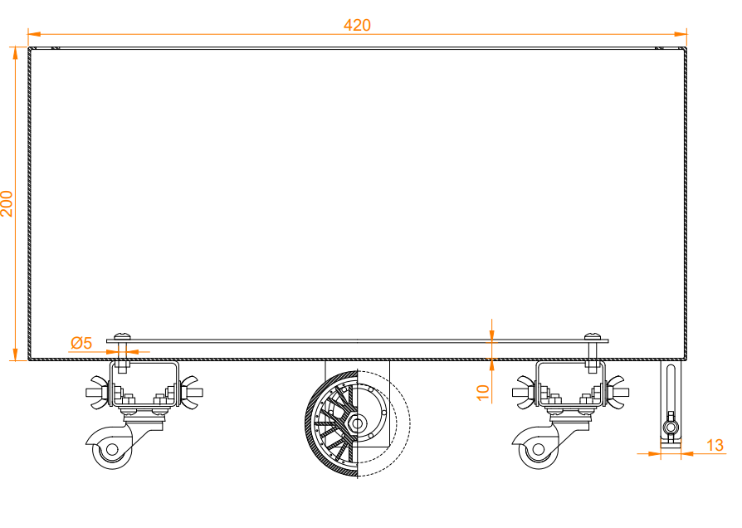


Figura 47 - Vista lateral do robô

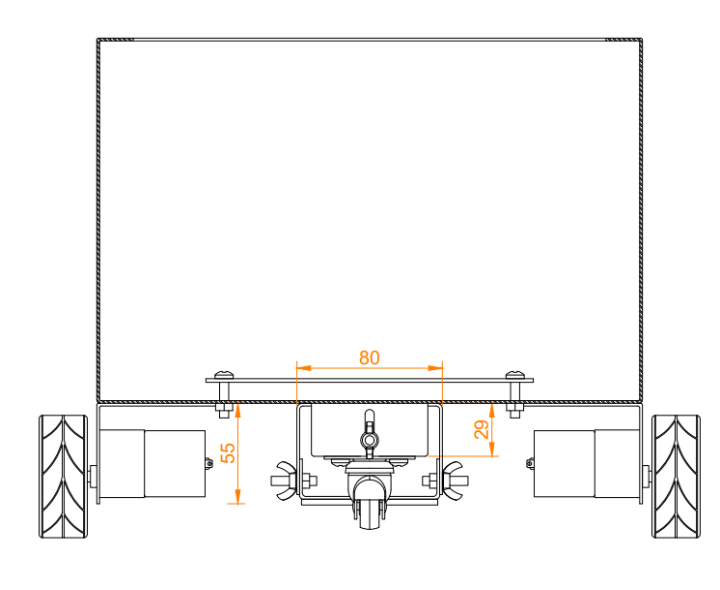


Figura 48 - Vista traseira do robô

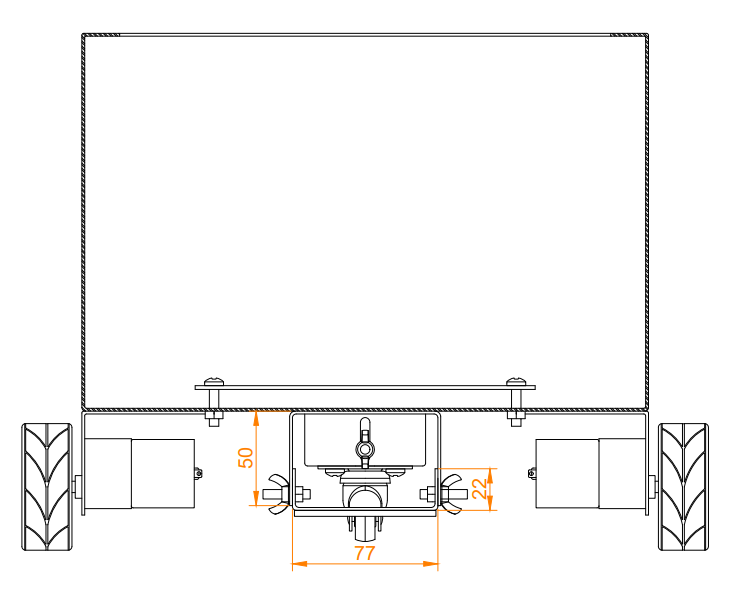


Figura 49 - Vista frontal do robô

Possíveis opções alternativas para o desenho dos circuitos

Introdução

Evolução do atual desenho para um sistema baseado em microcomputador

Introdução

# 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] | 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]. |
| [3] | STMicroelectronics, “DUAL FULL-BRIDGE DRIVER”. Patente L298 datasheet, janeiro 2000. |
| [4] | T. Instruments, “TL494 Pulse-Width-Modulation Control Circui”. Patente TL494 datasheet, janeiro 1983. |

|  |  |
| --- | --- |
| [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] | 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]. |
| [3] | STMicroelectronics, “DUAL FULL-BRIDGE DRIVER”. L298 datasheet, Jan. 2000. |
| [4] | T. Instruments, “TL494 Pulse-Width-Modulation Control Circui”. TL494 datasheet, janeiro 1983 [revisto março 2017]. |