

**UNIVERSIDADE DE SÃO PAULO  
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

**Caio Ian Alher Carrasco**

**Dispositivo de controle dos motores CC da locomoção de  
um robô de combate**

**São Carlos**

**2020**



**Caio Ian Alher Carrasco**

**Dispositivo de controle dos motores CC da locomoção de  
um robô de combate**

Monografia apresentada ao Curso de Engenharia Elétrica com Ênfase em Eletrônica, da Escola de Engenharia de São Carlos, da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Dennis Brandão

**São Carlos  
2020**

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,  
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS  
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da  
EESC/USP com os dados inseridos pelo(a) autor(a).

C257d Carrasco, Caio Ian Alher  
Dispositivo de controle dos motores CC da  
locomoção de um robô de combate / Caio Ian Alher  
Carrasco; orientador Dennis Brandão. São Carlos, 2020.

Monografia (Graduação em Engenharia Elétrica com  
ênfase em Eletrônica) -- Escola de Engenharia de São  
Carlos da Universidade de São Paulo, 2020.

1. Locomoção de robô. 2. Controle de motores CC.  
3. Combate de robôs. 4. Hóquei de robôs. 5. Sabertooth.  
I. Título.

# FOLHA DE APROVAÇÃO

Nome: Caio Ian Alher Carrasco

Título: “Dispositivo de controle dos motores CC da locomoção de um robô de combate”

Trabalho de Conclusão de Curso defendido e aprovado em  
29/10/2020,

com NOTA 10,0 (dez), pela Comissão Julgadora:

Prof. Associado Dennis Brandão - Orientador - SEL/EESC/USP

Prof. Associado Daniel Varela Magalhães - SEM/EESC/USP

Prof. Associado Marcelo Becker - SEM/EESC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:  
Prof. Associado Rogério Andrade Flauzino



*Este trabalho é dedicado aos integrantes da Equipe SEMEAR,  
por todo o crescimento proporcionado,  
e para utilização em seus futuros desenvolvimentos.*



## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, pela saúde, providência constante e presença em minha vida.

Sou muito grato à minha mãe e ao meu pai, Ivany e Claudemir, por me ensinarem os valores que me acompanham, que sempre colocaram as minhas irmãs e a mim à frente de suas próprias vontades, sacrificando-se em incontáveis ocasiões, tanto para atender às nossas necessidades, quanto para nos proporcionar aquilo que eles mesmos não tiveram, como foi com nossa educação.

Agradeço também às minhas irmãs, Carolina e Caren, por me ajudarem incessantemente a ser uma pessoa melhor, a enxergar meus defeitos e reconhecer minhas qualidades; com elas os momentos felizes ficam ainda melhores, e os momentos tristes tornam-se mais suportáveis. À Carolina, especialmente, minha gratidão pela paciência, atenção, suporte e disposição enquanto eu desenvolvia e documentava este projeto. Vocês duas sempre foram e sempre serão minhas melhores amigas!

Meu muito obrigado também direciona-se a três grupos fundamentais durante minha graduação: à República Gato Preto, que me acolheu como parte da família e me ensinou bastante sobre a vida durante esses 5 anos; à turma da Eletrônica de 2014, fundamental nas horas de dedicação e também nas de descontração; ao Grupo SEMEAR e todos os seus integrantes, pois nutriu minha avidez pela eletrônica, construiu minha concepção de time e liderança, e por ser o berço deste projeto.

Por fim, agradeço ao meu orientador, à Universidade de São Paulo, à Escola de Engenharia de São Carlos e ao Departamento de Engenharia Elétrica e da Computação, com todos os seus colaboradores, pelo conhecimento passado, apoio prestado e dedicação empenhada.



*“Capricho: fazer o seu melhor na condição que você tem,  
enquanto não tem condições melhores para fazer melhor ainda.”*

*Mario Sergio Cortella*



## RESUMO

CARRASCO, C.I.A. **Dispositivo de controle dos motores CC da locomoção de um robô de combate.** 2020. 134p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2020.

O combate de robôs vem ganhando cada vez mais espaço nos grupos extracurriculares dos cursos de engenharia, por conta dos desafios que apresenta nos mais diversos aspectos de projeto. No campo da eletrônica, o dispositivo responsável pela aquisição dos sinais do rádio controle e tradução desta informação em movimento para o robô tem um custo comercial elevado, e não possibilita manutenção por parte das equipes por não disponibilizar documentação técnica detalhada. Nesse sentido, este trabalho visa apresentar os passos da implementação da eletrônica de locomoção de um robô de combate (categoria *hobbyweight*) ou de hóquei, ou seja, todos os passos de projeto de um dispositivo que lê os canais de comando do receptor do rádio controle, e a partir disso aciona dois motores de forma independente, responsáveis pela locomoção do robô, com a potência e sentido pretendidos, usando a alimentação proveniente de baterias. Ao final deste trabalho, tem-se detalhes dos projetos funcionais de *hardware* e *software*, para serem reproduzidos como estão, ou como ponto de partida para implementação de melhorias e modificações.

**Palavras-chave:** Locomoção. Controle de motores. Combate de robôs. Hóquei de robôs. Sabertooth.



## **ABSTRACT**

CARRASCO, C.I.A. **DC motor drive responsible for the movement of a combat robot.** 2020. 134p. Undergraduate Thesis - São Carlos School of Engineering, University of São Paulo, São Carlos, 2020.

The robot's combat has been conquering more and more space among engineering extracurricular groups, due to the challenges it presents in the several project approaches. In what concerns the electronics field, the device responsible for acquiring the signals from the radio control receiver, and translate it to robot movement, has a high commercial cost, and does not provide enough detailed data for these groups to perform maintenance themselves. Given this context, this work aims to show the steps regarding the project of the electronics for locomotion control of a combat robot (hobbyweight class) or a hockey robot, that is, all the phases of a device's project which receives command information from the radio control receiver, and from that this device controls the power and direction of two independent locomotion DC motors, using energy of batteries. At the end of this document, hardware and software projects are functional and detailed enough to be implemented as they are, or as a starting point for improvements and design changes.

**Keywords:** Locomotion. Motor control. Combat of robots. Hockey of robots. Sabertooth.



## LISTA DE FIGURAS

Figura 1 – Fotos de dois robôs da Equipe SEMEAR, um de hóquei e outro de combate.	28
Figura 2 – Típicos elementos internos de um robô de hóquei, semelhante ao de combate.	29
Figura 3 – Elementos de um motor CC escovado e de ímãs permanentes.	32
Figura 4 – Circuito equivalente de um motor CC controlado pela armadura.	32
Figura 5 – Circuito básico para exemplificar a modulação PWM.	34
Figura 6 – Tensão média $V_o$ a 30 % de $V_{bat}$ quando $\delta = 30\%$ na modulação PWM.	34
Figura 7 – Circuito básico de uma ponte H utilizando chaves.	35
Figura 8 – Comparação de diferentes frequências de chaveamento do motor, para um mesmo $\delta = 75\%$ .	36
Figura 9 – Ponte H construída com MOSFET's e diodos de proteção.	37
Figura 10 – Circuito grampeador de tensão, ou <i>clamper</i> .	39
Figura 11 – Simulação da saída de um circuito grampeador de tensão.	39
Figura 12 – Estrutura de um circuito de <i>bootstrap</i> .	40
Figura 13 – Diagrama interno do opto-acoplador HCPL-3120, da Avago.	41
Figura 14 – Exemplo de aplicação do HIP4082 no acionamento de uma ponte H.	42
Figura 15 – Terminais correspondentes a cada <i>timer</i> do ATmega328P.	44
Figura 16 – Diagrama temporal dos modos de PWM do ATmega328P.	45
Figura 17 – Registradores TCCR2A e TCCR2B de controle da operação do TMR2.	46
Figura 18 – Rádio controle TURNIGY 5x, de 5 canais, e seu receptor, utilizados neste projeto.	52
Figura 19 – Aspectos do movimento de um aeromodelo controlados por cada canal do rádio controle.	53
Figura 20 – Possível alocação dos canais do rádio controle para os comandos de locomoção e acionamento da arma ativa de um robô de combate.	54
Figura 21 – Dois modos de configuração dos canais do rádio controle TURNIGY 5x.	54
Figura 22 – Exemplo da modulação PPM com dois canais do rádio controle.	55
Figura 23 – Bateria de LiPo comumente utilizada nos robôs de combate.	57
Figura 24 – Diagrama em blocos do circuito.	59
Figura 25 – Módulo do microcontrolador ATmega328P em conjunto com o receptor do rádio controle.	61
Figura 26 – Valores máximos permitidos na operação do ATmega328P.	61
Figura 27 – Implementação do <i>power-on reset</i> do microcontrolador.	63
Figura 28 – Circuito periférico do cristal oscilador com capacitores do ATmega328P.	64
Figura 29 – Medição do sinal do cristal oscilador de 16 MHz do microcontrolador.	64

Figura 30 – Onda quadrada de entrada do <i>Charge pump</i> . . . . .	65
Figura 31 – Características dos pulsos PPM de saída do receptor do rádio controle. . . . .	66
Figura 32 – Comportamento da modulação PPM nos canais ELE e RUD, lidos pelo microcontrolador. . . . .	67
Figura 33 – Módulo do <i>Charge Pump</i> . . . . .	67
Figura 34 – Circuito para simulação do <i>Charge pump</i> no LTspice. . . . .	69
Figura 35 – Simulação da corrente pelos diodos do circuito de <i>Charge pump</i> . . . . .	69
Figura 36 – Simulação da dissipação de potência nos resistores de limitação de corrente do circuito de <i>Charge Pump</i> . . . . .	70
Figura 37 – Simulação do comportamento de $V_{CP}$ no caso de consumo típico do HCPL-3120 nos quatro chips do <i>Gate driver</i> , e bateria descarregada ( $V_{BAT} = 14,8$ V). . . . .	71
Figura 38 – Simulação da tensão $V_{CP}$ no caso de consumo máximo do HCPL-3120 nos quatro chips do <i>Gate driver</i> , e bateria descarregada ( $V_{BAT} = 14,8$ V). . . . .	72
Figura 39 – Simulação do comportamento de $V_{CP}$ no caso de consumo típico do HCPL-3120 nos quatro chips do <i>Gate driver</i> , e bateria carregada ( $V_{BAT} = 16,8$ V). . . . .	73
Figura 40 – Medição da energização do <i>Charge pump</i> sem carga conectada. . . . .	73
Figura 41 – Medição da energização do <i>Charge pump</i> com o <i>Gate driver</i> conectado e bateria carregada ( $V_{BAT} = 16,8$ V). . . . .	74
Figura 42 – Medição da energização do <i>Charge pump</i> com o <i>Gate driver</i> conectado e bateria descarregada ( $V_{BAT} = 14,8$ V). . . . .	74
Figura 43 – Posição dos canais do osciloscópio na medição da tensão dos resistores do <i>Charge pump</i> . . . . .	75
Figura 44 – Medição da tensão nos resistores do <i>Charge pump</i> durante sua energização e pico de corrente. . . . .	75
Figura 45 – Medição da tensão nos resistores do <i>Charge pump</i> durante operação normal. . . . .	76
Figura 46 – Módulo de <i>Gate Driver</i> . . . . .	77
Figura 47 – Simulação do <i>dead-time</i> do <i>Gate Driver</i> . . . . .	78
Figura 48 – Medição dos atrasos que geram o <i>dead-time</i> no <i>Gate driver</i> . . . . .	79
Figura 49 – Medição da queda de tensão interna do HCPL-3120 durante chaveamento de $V_{BAT}$ e $V_{CP}$ . . . . .	80
Figura 50 – Módulo das Pontes H. . . . .	80
Figura 51 – Comparação entre valores de $R_{gate}$ e sua influência no comportamento do chaveamento do transistor. . . . .	82
Figura 52 – Medição da tensão de acionamento $V_{GS}$ dos MOSFET's para $V_{BAT} = 16,8$ V. . . . .	83

Figura 53 – Medição da tensão de acionamento $V_{GS}$ dos MOSFET's para $V_{BAT} = 14,8$ V. . . . .	83
Figura 54 – Medição da tensão de pico sobre $R_{gate}$ , com bateria carregada e no resistor do transistor <i>Low-side</i> . . . . .	84
Figura 55 – Medição do comportamento de $V_{DS}$ durante o chaveamento do MOSFET <i>Low-side</i> . . . . .	85
Figura 56 – Tensão máxima medida em $V_{DS}$ durante a abertura do transistor. . . . .	86
Figura 57 – Medição do ruído de alta frequência em $V_{BAT}$ com o uso do capacitor eletrolítico (1500 $\mu$ F) apenas. . . . .	88
Figura 58 – Medição do ruído de alta frequência em $V_{BAT}$ com o uso do capacitor eletrolítico (1500 $\mu$ F) em paralelo com o capacitor de poliéster (100 nF). . . . .	88
Figura 59 – Módulo do Regulador de Tensão. . . . .	89
Figura 60 – Consumo do ATmega328P com 5 V de alimentação e 16 MHz de frequência de operação. . . . .	90
Figura 61 – Circuito recomendado para o regulador de tensão da família LM78xx. . . . .	91
Figura 62 – Medição do ruído presente nos 5 V gerados - cenário sem acionamento dos motores. . . . .	92
Figura 63 – Medição do ruído presente nos 5 V gerados - cenário com acionamento dos motores em giro livre, sem carga. . . . .	93
Figura 64 – Medição do ruído presente nos 5 V gerados - cenário com acionamento dos motores travados. . . . .	93
Figura 65 – Fluxograma do algoritmo de <i>software</i> . . . . .	95
Figura 66 – Pinos do ATmega328P correlacionados com suas funções e identificação na IDE do Arduino. . . . .	97
Figura 67 – Correlação entre os comandos do rádio controle, o movimento pretendido e o PWM aplicado em cada motor da locomoção. . . . .	102
Figura 68 – Relação entre as variáveis X e Y, e a saída E da Equação 3.3. . . . .	104
Figura 69 – Gráficos representativos do PWM resultante da combinação dos dois canais do rádio controle, para cada um dos motores. . . . .	105
Figura 70 – Medição do efeito do pulso residual na saída do <i>Fast PWM</i> com $OCRnx = 0$ . . . . .	107
Figura 71 – Bancada de desenvolvimento do projeto e dos testes de integração. . . . .	108
Figura 72 – Circuito completo montado em matrizes de contatos ( <i>protoboards</i> ). . . . .	109
Figura 73 – Medições da tensão sobre um motor durante o acionamento para frente em diferentes valores de <i>duty-cycle</i> . . . . .	110
Figura 74 – Medições da tensão sobre um motor durante o acionamento para trás em diferentes valores de <i>duty-cycle</i> . . . . .	110
Figura 75 – Medição do ruído nas tensões constantes do circuito durante o chaveamento do motor em vazio. . . . .	112

Figura 76 – Medição da energização e consumo de corrente do circuito em <i>stand-by</i> , sem acionamento de motores. . . . .	112
Figura 77 – Circuito completo do dispositivo - parte 1/2. . . . .	125
Figura 78 – Circuito completo do dispositivo - parte 2/2. . . . .	126
Figura 79 – Acionamento do motor com PWM de 400 Hz. . . . .	132
Figura 80 – Acionamento do motor com PWM de 4 kHz. . . . .	133
Figura 81 – Acionamento do motor com PWM de 40 kHz. . . . .	133
Figura 82 – Acionamento do motor com PWM de 20 kHz. . . . .	134
Figura 83 – Acionamento do motor com PWM de 7,8 kHz. . . . .	134

## LISTA DE TABELAS

Tabela 1 – <i>Timers</i> do ATmega328P . . . . .	44
Tabela 2 – TMR2 - Parâmetro WGM - <i>Waveform Generation Mode</i> . . . . .	47
Tabela 3 – TMR2 - Parâmetro COM - <i>Compare Output Mode</i> - em operação não-PWM. . . . .	48
Tabela 4 – TMR2 - Parâmetro COM - <i>Compare Output Mode</i> - em operação <i>Fast PWM</i> . . . . .	49
Tabela 5 – TMR2 - Parâmetro COM - <i>Compare Output Mode</i> - em operação <i>Phase correct</i> . . . . .	50
Tabela 6 – TMR2 - Parâmetro CS - <i>Clock Select</i> . . . . .	51
Tabela 7 – Siglas dos canais do rádio controle e sua identificação. . . . .	53
Tabela 8 – Tensões correspondentes aos níveis lógicos do pino de <i>reset</i> do ATmega328P. . . . .	62
Tabela 9 – Valores permitidos para o resistor de <i>pull-up</i> no pino de <i>reset</i> do ATmega328P. . . . .	62
Tabela 10 – Condições de operação recomendadas para o HCPL-3120. . . . .	68
Tabela 11 – Faixa de valores para a tensão de condução direta do LED de controle do opto-acoplador HCPL-3120. . . . .	68
Tabela 12 – Queda de tensão de saída e consumo do HCPL-3120. . . . .	70
Tabela 13 – Pico de corrente de saída permitido pelo opto-acoplador HCPL-3120. .	81
Tabela 14 – Algumas características elétricas do MOSFET IRF1405. . . . .	82
Tabela 15 – Opções de MOSFET para as pontes H. . . . .	86



## LISTA DE ABREVIATURAS E SIGLAS

ADC	<i>Analog-Digital Converter</i>
CC	Corrente Contínua
COM	<i>Compare Output Mode</i>
CS	<i>Clock Select</i>
CTC	<i>Clear Timer on Compare match</i>
DC	<i>Direct Current</i>
EESC	Escola de Engenharia de São Carlos
FET	<i>Field Effect Transistor</i>
FHSS	<i>Frequency Hopping Spread Spectrum</i>
FCEM	Força Contra-Eletromotriz
GPIO	<i>General Purpose Input Output pins</i>
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
LED	<i>Light Emitting Diode</i>
LiPo	<i>Lithium-ion Polymer</i>
MOSFET	<i>Metal-Oxide Semiconductor Field Effect Transistor</i>
OCR	<i>Output Compare Register</i>
PCB	<i>Printed Circuit Board</i>
PPM	<i>Pulse-Position Modulation</i>
PWM	<i>Pulse-Width Modulation</i>
SEMEAR	Soluções em Engenharia Mecatrônica e Aplicações na Robótica
TCC	Trabalho de Conclusão de Curso
TCCR	<i>Timer/Counter Control Register</i>
TVS	<i>Transient Voltage Suppressor</i>

USB      *Universal Serial Bus*

USP      Universidade de São Paulo

WGM      *Waveform Generation Mode*

## LISTA DE SÍMBOLOS

$\Omega$	Ômega maiúsculo, usado para indicar resistência elétrica medida em [ohms]
$\delta$	Delta minúsculo, usado para indicar o ciclo de trabalho, ou <i>duty-cycle</i> , na modulação por largura de pulso (PWM); adimensional ou [%]
$\tau$	Tau minúsculo, usado para representar a constante de tempo de um circuito RC ( $\tau = R.C$ ) ou RL ( $\tau = L/R$ ); [s]
$v_a(t)$	Tensão aplicada na armadura do motor CC, em função do tempo; [V]
$i_a(t)$	Corrente de armadura do motor CC, em função do tempo; [A]
$R_a$	Resistência do enrolamento de armadura do motor CC; [ $\Omega$ ]
$L_a$	Indutância do enrolamento de armadura do motor CC; [H]
$\omega(t)$	Ômega minúsculo, usado para representar a velocidade angular do eixo do motor CC; [rad/s]
$e(t)$	Força contra-eletromotriz, proporcional a $\omega(t)$ , nos terminais do enrolamento de armadura do motor CC; [V]
$J$	Momento de inércia da carga acoplada ao eixo do motor CC; [kg.m <sup>2</sup> ]
$f$	Coeficiente de atrito viscoso do motor CC, proporcional a $\omega(t)$ ; [N.m.s]



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>27</b>
1.1	Contexto e motivação	27
1.2	Objetivos	28
1.3	Organização do trabalho	29
<b>2</b>	<b>EMBASAMENTO TEÓRICO</b>	<b>31</b>
2.1	Motor CC escovado com ímãs permanentes	31
2.2	Modulação por largura de pulso - PWM	33
2.3	Ponte H	35
2.3.1	Charge Pump	38
2.3.2	Bootstrap	40
2.4	Microcontrolador ATmega328P	42
2.4.1	Timers do ATmega328P	43
2.5	Rádio Controle	51
2.5.1	Modulação por posição de pulso - PPM	55
2.6	Bateria de Polímero de Lítio	56
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>59</b>
3.1	<b>Hardware</b>	<b>60</b>
3.1.1	Microcontrolador	60
3.1.2	Charge Pump	66
3.1.3	Gate Driver	76
3.1.4	Pontes H	80
3.1.5	Filtros e o Regulador de Tensão	89
3.2	<b>Software</b>	<b>95</b>
3.2.1	Definições e constantes	96
3.2.2	Bloco A - Configurações iniciais	98
3.2.3	Bloco B - Determinação do zero dos canais	99
3.2.4	Bloco C - Leitura dos canais e adaptação de limites	100
3.2.5	Bloco D - Combinação dos canais e conversão para PWM	100
3.2.6	Bloco E - Aplicação do PWM para cada motor	105
3.3	<b>Integração</b>	<b>108</b>
3.3.1	Testes de integração	109
3.3.2	Modos de falha	113
<b>4</b>	<b>CONCLUSÃO</b>	<b>117</b>

<b>REFERÊNCIAS</b>	<b>121</b>
<b>APÊNDICES</b>	<b>123</b>
<b>APÊNDICE A – CIRCUITO COMPLETO</b>	<b>125</b>
<b>APÊNDICE B – CODIFICAÇÃO DO SOFTWARE</b>	<b>127</b>
<b>APÊNDICE C – FREQUÊNCIAS DE PWM TESTADAS NO ACIONAMENTO DO MOTOR CC</b>	<b>131</b>

# 1 INTRODUÇÃO

## 1.1 Contexto e motivação

Grupos extracurriculares são de grande crescimento para o estudante universitário, pois servem como um complemento prático aos ensinamentos transmitidos a ele ao longo de seu curso. São diversos os tipos de grupos, cada um com suas particularidades; dentre eles encontram-se aqueles voltados às competições de robótica, como é o caso do SEMEAR, participante de eventos como a IEEE Open<sup>1</sup> e a Winter Challenge<sup>2</sup>. Em meio a todas as categorias, o combate de robôs se destaca por exigir um projeto bastante robusto de engenharia, tanto no aspecto mecânico quanto no eletrônico, pois os robôs são submetidos às situações mais adversas e extremas durante a luta, e não podem falhar.

Nesses robôs, a eletrônica tem o principal papel de receber o sinal do receptor do rádio controle e comandar a locomoção e o acionamento da "arma ativa", caso o robô tenha uma, sendo geralmente utilizado um dispositivo para cada propósito. Tratando apenas da locomoção, o desafio se dá pela necessidade de uma eletrônica compacta, sem partes eletromecânicas que podem quebrar pelos impactos da luta, e sem muito espaço para dissipação de potência, porém que seja capaz de controlar ao menos dois motores (um de cada lado) que realizam o movimento do robô na arena, com muita rapidez de resposta e entrega de potência.

Atualmente existem algumas soluções no mercado que atendem a essas condições típicas da luta de robôs; são exemplos a SaberTooth, produto da empresa americana Dimension Engineering<sup>3</sup>, e a Victor 884, da empresa americana Innovation First por meio de sua marca VEX Robotics<sup>4</sup>, produtos que custam acima de 50 dólares, sem contar o preço de importação e dos impostos. Algumas equipes nacionais já procuram desenvolver suas próprias eletrônicas de locomoção, como são exemplos a ThundeRatz<sup>5</sup>, da Escola Politécnica da USP, a Uairrior<sup>6</sup>, da UNIFEI, mas que não as disponibilizam para venda. Além do preço, o problema dessas soluções comerciais é que são produtos fechados, ou seja, as equipes que se utilizam delas dificilmente conseguem realizar manutenção em caso de defeitos ou falhas, por não terem acesso à documentação detalhada de projeto.

A partir de todo este contexto e sendo o autor um membro da equipe SEMEAR, a motivação do trabalho é a criação da base de uma eletrônica de locomoção para os

---

<sup>1</sup> <<http://www.cbrobotica.org/index.php/categorias/>>

<sup>2</sup> <<https://www.robocore.net/eventos/wc14>>

<sup>3</sup> <<https://www.dimensionengineering.com/products/sabertooth2x12rc>>

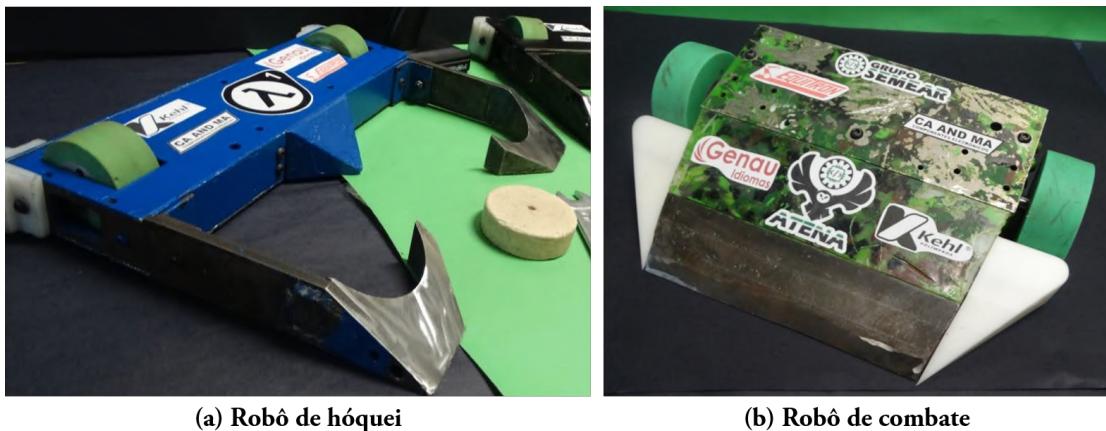
<sup>4</sup> <<http://www.team358.org/files/electrical/V883UsersManual.pdf>>

<sup>5</sup> <<https://thunderatz.org/projects/boards/>>

<sup>6</sup> <<https://uairrior.com.br/index.php/home/eletronicos/>>

robôs de combate - e também de hóquei, outra categoria de competição - que possa suprir a demanda hoje atendida pela compra e importação de produtos estrangeiros, de modo a agregar conhecimento técnico para o grupo, e poder economizar dinheiro, tanto o da aquisição quanto o das manutenções desses dispositivos.

Figura 1: Fotos de dois robôs da Equipe SEMEAR, um de hóquei e outro de combate.



Fonte: Elaborado pelo autor.

## 1.2 Objetivos

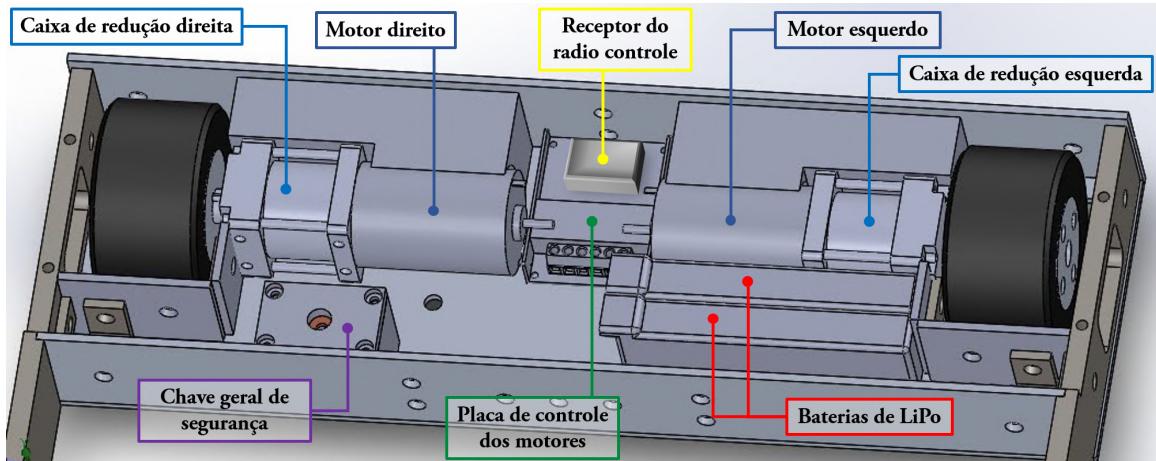
O objetivo deste trabalho consiste na elaboração documentada de um dispositivo base de acionamento de dois típicos motores CC utilizados na locomoção dos robôs de combate da categoria *hobbyweight*<sup>7</sup>, e também dos robôs de hóquei da Equipe SEMEAR, a partir de comandos recebidos pelo receptor do rádio controle. Este utiliza-se de dois canais para controlar a direção e a velocidade do robô, e seu receptor repassa esses comandos por meio de modulação PPM.

Baseado na motivação já explicitada, os requisitos são que o dispositivo deve ser capaz de acionar os dois motores de locomoção individualmente, nos dois sentidos de rotação, cada um com velocidade correspondente à comandada pelo rádio controle. A operação não pode sobreaquecer os componentes da placa, nem expôr nenhum deles às suas temperaturas críticas, e deve ser robusto a ponto de suportar variações bruscas de controle sem falhar, isso para toda a faixa de carga das baterias de alimentação.

Por fim, além da documentação técnica aqui registrada, que abre margem para a manutenção do dispositivo, é esperado que o conhecimento reunido sirva como base para desenvolvimentos futuros, mais elaborados, com aprimoramentos dos circuitos e do algoritmo de controle. Não obstante, a expectativa é que já esta versão aqui projetada

<sup>7</sup> A categoria de peso *hobbyweight* aceita robôs de até 12 libras, que equivale a 5,44 kg.

Figura 2: Típicos elementos internos de um robô de hóquei, semelhante ao de combate.



Fonte: Elaborado pelo autor.

sirva como um substituto completo ao comercial para tratar da locomoção dos robôs de combate e hóquei da equipe SEMEAR.

### 1.3 Organização do trabalho

Este trabalho está dividido essencialmente em quatro capítulos, seguidos da seção de referências bibliográficas, e por fim três apêndices.

O primeiro capítulo é este, da Introdução, onde está apresentada a contextualização do projeto, seus objetivos e breves requisitos.

O segundo capítulo é do Embasamento Teórico, onde é apresentada a revisão dos principais conceitos abordados ao longo do desenvolvimento do projeto, com uma breve introdução de tecnologias e recursos específicos, em alguns momentos.

O terceiro capítulo é o Desenvolvimento do trabalho, onde constam todos os passos para a elaboração, implementação, testes e os resultados do projeto. Por conta da complexidade, o capítulo foi dividido em três subseções, a primeira para explicar aspectos do *hardware*, a segunda para abordar a implementação do *software*, e a terceira para mostrar alguns resultados da integração dos dois primeiros, e expôr os principais modos de falha do dispositivo.

O quarto e último capítulo do corpo do trabalho é a Conclusão, onde está a síntese do que foi alcançado com o projeto, e algumas sugestões de implementações futuras.

Nos apêndices, ao final do texto, estão apresentados o circuito completo do dispositivo, o código do *software*, e as curvas de corrente e tensão do motor para diferentes frequências de acionamento da modulação PWM.



## 2 EMBASAMENTO TEÓRICO

O entendimento completo deste trabalho requer que o leitor tenha conhecimento prévio e básico dos principais conceitos da engenharia elétrica, tais como: tensão e corrente elétricas, circuito elétricos, magnéticos e uma boa noção de ondas eletromagnéticas; comportamento dos elementos passivos (resistor, capacitor e indutor) e como são modelados; familiaridade com semicondutores, diodos, suas construções mais comuns e seu comportamento; o que são transistores, tanto bipolares quanto *FET's (Field Effect Transistors)* e seu uso como chave; o que são microcontroladores e sua funcionalidade; familiaridade com motores de corrente contínua; entre outros tópicos já bastante discutidos durante o curso de engenharia elétrica.

Considerando esta base de conhecimentos, este capítulo traz uma rápida revisão de algumas tecnologias utilizadas durante o desenvolvimento do projeto, destacando aspectos pertinentes ao presente trabalho, e também introduz dispositivos e técnicas mais específicas, como o microcontrolador escolhido e um protocolo de comunicação adotado, a fim de servirem como referência durante a leitura do texto.

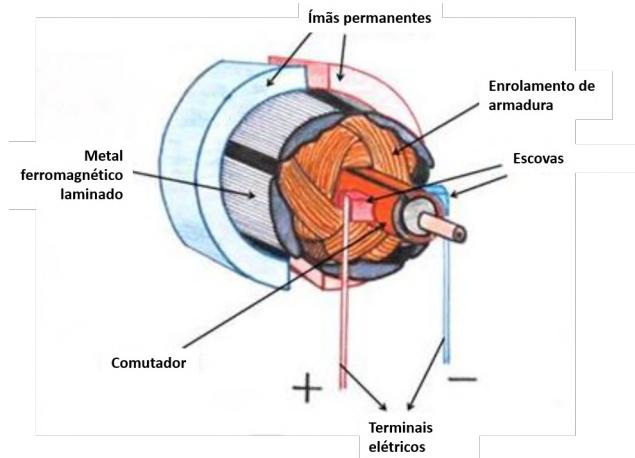
Em todo caso, explicações mais completas e aprofundadas, e outros exemplos de aplicação são recomendados, podendo ser encontrados através da bibliografia citada, ou por pesquisas em sites confiáveis da internet.

### 2.1 Motor CC escovado com ímãs permanentes

O motor elétrico é um dos principais atuadores da robótica por sua capacidade de transformar energia elétrica em torque e movimento, uma interface muito importante para sistemas eletro-mecânicos, com diversas tecnologias e modelos de construção. O motor de corrente contínua, ou motor CC, é amplamente utilizado na robótica principalmente por seu controle de velocidade e torque, precisão e simplicidade de acionamento, e variedade de modelos disponíveis no mercado. Os tratados neste trabalho são os motores CC com ímãs permanentes, compostos por duas partes: uma estática, o estator, onde ficam os dois ímãs, o comutador e os apoios do eixo; e outra móvel girante, o rotor, onde fica o enrolamento de armadura e o eixo do motor. Estes componentes são mostrados na Figura 3.

Apenas para relembrar superficialmente o modo de operação do motor CC, as escovas, geralmente de carvão, e o comutador são os responsáveis pelo contato elétrico dos terminais do motor com o enrolamento de armadura; este, quando conduzindo uma corrente elétrica, gera seu próprio campo magnético proporcional à corrente, que interage com o campo magnético gerado pelos ímãs permanentes e provoca a rotação do motor; o comutador tem sua construção tal que evita o equilíbrio das forças magnéticas, de

Figura 3: Elementos de um motor CC escovado e de ímãs permanentes.

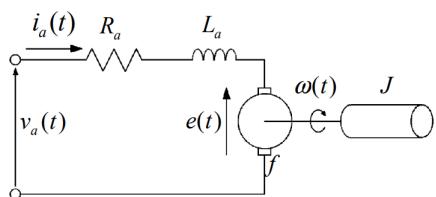


Fonte: Adaptado de ([SUSTEK et al., 2017](#))

modo a sempre gerar um torque no eixo do motor. Outro efeito importante é a força contra-eletromotriz gerada pela rotação do enrolamento de armadura enquanto imerso no campo magnético dos ímãs permanentes: apesar de o campo ser constante, o enrolamento é a referência e gira com o eixo, portanto para ele existe uma variação de fluxo à qual ele se opõe, segundo a Lei de Lenz, resultando assim na força contra-eletromotriz que é dependente da velocidade angular do eixo ([MOURA, 2014](#)). Uma explicação mais aprofundada por ser encontrada em ([VUKOSAVIC, 2013](#)) e ([MCPPERSON, 1981](#)).

Em motores mais potentes, o campo magnético é gerado por um enrolamento próprio, chamado "enrolamento de campo", mas para os motores tratados neste trabalho o campo magnético constante é gerado por ímãs permanentes. Desse modo, o controle de torque e velocidade do motor fica a cargo da corrente de armadura, o enrolamento presente no rotor. A fim de modelarmos o comportamento do motor a partir da tensão aplicada, o circuito equivalente do citado motor é apresentado na Figura 4.

Figura 4: Circuito equivalente de um motor CC controlado pela armadura.



Fonte: Extraído de ([BASILIO; MOREIRA, 2001](#))

Expandindo a nomenclatura usada no modelo elétrico do motor CC (Figura 4), temos:

- $v_a(t)$  = Tensão aplicada na armadura
- $i_a(t)$  = Corrente de armadura
- $R_a$  = Resistência do enrolamento de armadura
- $L_a$  = Indutância do enrolamento de armadura
- $e(t)$  = Força contra-eletromotriz, proporcional a  $\omega(t)$
- $\omega(t)$  = Velocidade angular do eixo
- $J$  = Momento de inércia da carga acoplada ao eixo
- $f$  = Coeficiente de atrito viscoso, proporcional a  $\omega(t)$

A partir do circuito equivalente e utilizando a Lei de Kirchhoff das tensões, obtemos a seguinte equação 2.1:

$$v_a(t) - R_a \cdot i_a(t) - L_a \cdot \frac{d}{dt} i_a(t) - e(t) = 0 \quad (2.1)$$

A força contra-eletromotriz  $e(t)$  é proporcional à velocidade angular  $\omega(t)$  pela constante  $K_e$ , um parâmetro construtivo do motor, de forma que podemos escrever a equivalência  $e(t) = K_e \cdot \omega(t)$ . Portanto, a equação 2.1 pode ser reescrita da seguinte forma:

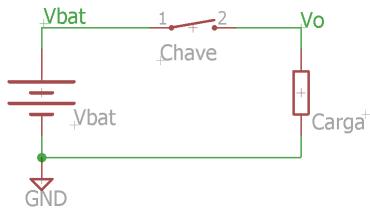
$$v_a(t) = R_a \cdot i_a(t) + L_a \cdot \frac{d}{dt} i_a(t) + K_e \cdot \omega(t) \quad (2.2)$$

A partir da equação 2.2 observa-se que quanto maior a velocidade angular do motor  $\omega(t)$ , maior a força contra-eletromotriz  $e(t)$  produzida e menor a corrente  $i_a(t)$  consumida, portanto veremos um maior consumo de corrente do motor com o eixo travado (chamada "corrente de *stall*") ou durante a partida. Sem estendermo-nos nos aspectos mecânicos, uma breve informação útil para nossa análise é quanto ao torque  $\tau(t)$  do motor, que é proporcional à corrente  $i_a(t)$  pela constante de torque  $K_t$ , portanto  $\tau(t) = K_t \cdot i_a(t)$  ([GASPAR, 2011](#)); logo, durante a partida do motor,  $\omega(t)$  ainda é baixo, consequentemente  $i_a(t)$  é elevada, gerando um maior torque  $\tau(t)$  de partida, característico de motores CC.

## 2.2 Modulação por largura de pulso - PWM

Esta é uma técnica utilizada para gerar valores médios analógicos a partir de variáveis discretas, ou seja, a partir de uma fonte com tensão constante  $V_{bat}$  e uma chave, como mostra o circuito da Figura 5. Mantendo uma frequência constante  $f$  de chaveamento, e alterando o tempo  $t_{on}$  que a chave passa fechada, podemos controlar a tensão média de saída  $V_o$  no intervalo  $0 < V_o < V_{bat}$ .

Figura 5: Circuito básico para exemplificar a modulação PWM.



Fonte: Elaborado pelo autor

A relação entre o tempo de acionamento da chave ( $t_{on}$ ) e o período até ela voltar a ser acionada ( $T$ ) é chamada de "razão de trabalho", ou "*duty-cycle*", na nomenclatura em inglês, e pode ser representada pela letra  $\delta$ .

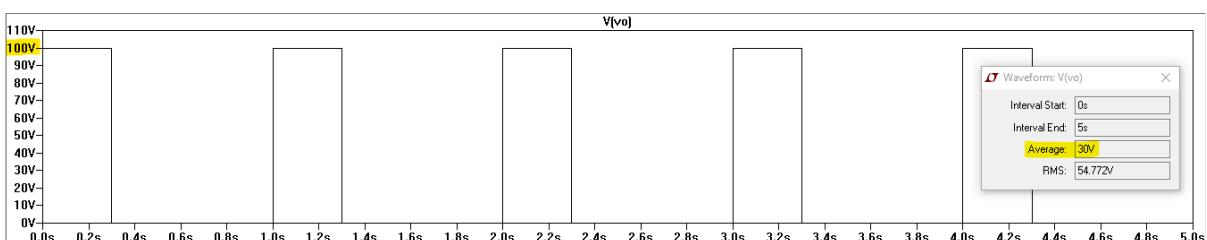
$$\delta = \frac{t_{on}}{T}, \quad \text{onde } T = \frac{1}{f} \quad (2.3)$$

Dessa forma, definindo uma razão  $\delta$  de acionamento da chave, e uma tensão constante  $V_{bat}$  de alimentação do circuito, a tensão média sobre a carga é calculada pela equação 2.4 a seguir:

$$V_o = \delta \cdot V_{bat} \quad (2.4)$$

A título de exemplo, supondo  $V_{bat} = 100$  V e procurando obter uma tensão média de 30 V na carga, calculamos pela equação 2.4 que  $\delta = 30\%$ . Definindo agora um período de acionamento  $T = 1$  s ( $f = 1$  Hz), calculamos por 2.3 que  $t_{on} = 0,3$  s. A medição de  $V_o$  exibida na Figura 6, referente ao circuito da Figura 5, mostra o comportamento da tensão em função do tempo, e a janela flutuante exibe no campo em destaque a tensão média (*Average*) exatamente em 30 V, como previsto.

Figura 6: Tensão média  $V_o$  a 30 % de  $V_{bat}$  quando  $\delta = 30\%$  na modulação PWM.

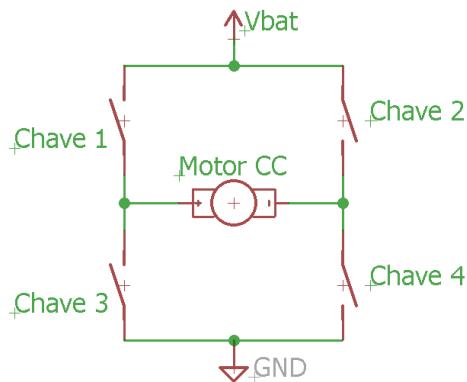


Fonte: Elaborado pelo autor

## 2.3 Ponte H

O nome 'Ponte H' refere-se a uma topologia de conversor DC-DC que permite o acionamento, no caso de motores, em seus dois sentidos de rotação. O esquema básico de uma ponte H pode ser construído utilizando chaves, como ilustrado na Figura 7, e seu funcionamento é relativamente simples: quando as chaves 1 e 4 são fechadas, a corrente flui do positivo para o negativo do motor, causando a rotação do mesmo em um sentido; quando as chaves 2 e 3 são acionadas simultaneamente, por outro lado, a corrente flui do negativo para o positivo do motor, provocando a rotação em sentido contrário ao anterior.

Figura 7: Circuito básico de uma ponte H utilizando chaves.



Fonte: Elaborado pelo autor

Em uma ponte H, duas chaves de um mesmo ramo nunca podem ser acionadas simultaneamente, pois acionar ao mesmo tempo [1 e 3] e/ou [2 e 4] coloca em curto-círcuito a alimentação. Fechar sincronicamente chaves de ramos diferentes, além do que já foi explicado para inversão do sentido do motor, ou seja, os pares [1 e 2] ou [3 e 4], causa o efeito de frenagem no motor, uma vez que coloca seus terminais em curto-círcuito.

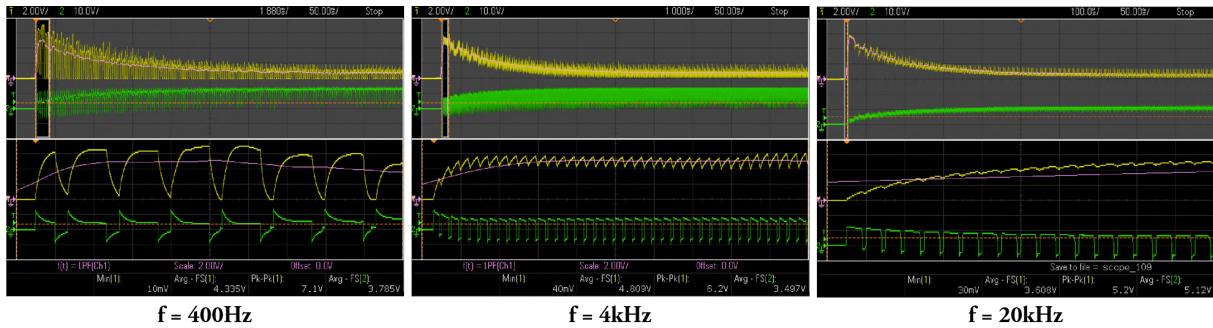
Trazendo o conceito de PWM revisado anteriormente, as chaves podem ser acionadas utilizando esta técnica de modo a controlar a velocidade do motor, pois utilizando a frequência adequada, ele passa a responder apenas ao valor médio da tensão aplicada, algo ajustado pelo *duty-cycle* ( $\delta$ ) do PWM. Este fenômeno ocorre devido ao efeito indutivo do enrolamento do motor, que se torna mais evidente quanto maior é a frequência de chaveamento.

A fim de ilustrar esta situação, um mesmo motor<sup>1</sup> foi chaveado, desde sua partida, com 75 % de *duty-cycle*, mantendo-se constante a tensão de alimentação da bateria, mas variando a frequência entre 3 diferentes valores: 400 Hz, 4 kHz e 20 kHz. A Figura 8 mostra

<sup>1</sup> O motor escolhido é um dos modelos já utilizados nas competições de combate de robôs, o mesmo tipo para o qual este projeto é voltado.

qualitativamente a variação da corrente que passa pelo motor (canal 1 em amarelo) e da tensão nos terminais do mesmo (canal 2 em verde) para cada um dos cenários.

Figura 8: Comparação de diferentes frequências de chaveamento do motor, para um mesmo  $\delta = 75\%$ .



Fonte: Elaborado pelo autor

Nota: A curva em amarelo é a corrente do motor, e a verde indica a tensão sobre ele.

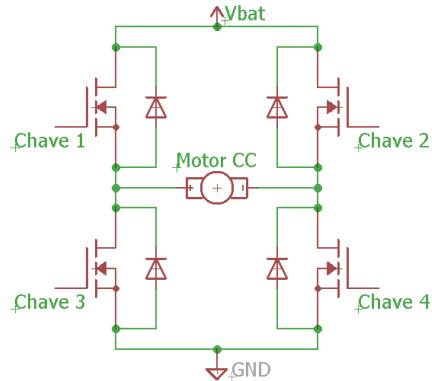
De um modo geral, na metade superior de cada gráfico em que está registrado o comportamento dos sinais até a estabilização, observamos o mesmo comportamento já esperado: uma corrente alta nos primeiros instantes, pois o motor está partindo, seguido de sua queda e estabilização conforme a velocidade aumenta e também acomoda-se. Em todos os três gráficos a ampliação das curvas está focada nos instantes iniciais, em que a corrente é maior: no primeiro caso, com 400 Hz, vemos a corrente com uma variação muito grande de amplitude, indo de zero a um pico em praticamente todos os ciclos de chaveamento, ou seja, o efeito indutivo de seu enrolamento praticamente não está evidente e o que dita o pico da corrente é a resistência do enrolamento; quando a frequência passa para 4 kHz já é possível observar um cenário completamente diferente na corrente, que varia bem menos sua amplitude e responde bem melhor ao valor médio da tensão aplicada, o que evidencia a predominância do efeito indutivo do motor, mas ainda assim é visível um *ripple* de corrente; no último gráfico da direita, a frequência de chaveamento está em 20 kHz e praticamente já não se vê o reflexo de silhueta do PWM aplicado, pois a corrente comporta-se como se houvesse apenas uma tensão constante aplicada sobre o motor.

Componentes eletrônicos que executam bem o papel de chaves são os MOSFET's, transistores desenhados para trabalharem principalmente em suas regiões de corte e saturação, sendo portanto eles a ocuparem os lugares das chaves do circuito da ponte H.

Algo ainda não mencionado aqui, porém de extrema importância para o funcionamento da ponte H, é a tensão gerada nos terminais do motor quando as chaves abrem interrompendo o caminho por onde a corrente passava, tensão esta resultante do efeito indutivo do enrolamento, também conhecida por "força contra-eletromotriz", e abreviada

por FCEM. Um modo de resolver a FCEM gerada no chaveamento é a inserção de diodos em antiparalelo com as chaves; a Figura 9 mostra como fica a Ponte H após a substituição para chaves eletrônicas (MOSFET's) e inserção dos diodos.

Figura 9: Ponte H construída com MOSFET's e diodos de proteção.



Fonte: Elaborado pelo autor

A construção física dos MOSFET's usados nessa aplicação, falando a nível de tratamento do silício, é tal que já produz um diodo em antiparalelo com seus terminais de dreno e fonte, de modo que em muitas vezes não há necessidade em colocar um diodo externo, a não ser que queira-se uma velocidade maior de chaveamento, por exemplo.

A configuração atual da ponte H (Figura 9) nos possibilita discutir as desvantagens de se aumentar muito a frequência de chaveamento, pois até então o aumento só apresentava a melhoria de um controle mais preciso da corrente que passa pelo motor. Pois bem, o transistor possui duas principais perdas: a perda de condução e a de chaveamento; a perda de condução existe pelo fato de que, mesmo em saturação, o canal de condução ainda apresenta uma resistência  $R_{DS_{on}}$ , informada no *datasheet* do componente, que dissipá uma certa potência proporcional ao quadrado da corrente; e a outra, a perda de chaveamento, se dá pelo fato de que o transistor não é uma chave ideal e instantânea, ou seja, ela leva um certo tempo, por menor que seja, para transitar entre seus estados "aberto" e "fechado", momentos em que corrente e tensão alternam, uma indo para seu máximo e outra indo para seu mínimo, o que gera um pico de potência instantânea. É nesta última, a perda por chaveamento, que o aumento da frequência influencia, pois quanto maior a frequência, mais chaveamento e mais significativa fica essa perda correspondente, ou seja, maior a potência dissipada no transistor.

Ainda assim existe um outro elemento que limita a frequência de chaveamento, que é o próprio tempo de resposta do transistor: a partir de uma determinada frequência, o sinal de comando no *gate* pode estar variando mais rápido que a resposta do seu chaveamento, tornando ineficaz o controle. Portanto, a determinação da frequência de chaveamento leva

em conta o compromisso do aumento das perdas por chaveamento e menor precisão do *duty-cycle*, com o melhor controle da corrente circulando pelo motor pela diminuição de sua variação de amplitude (*ripple*).

No circuito de ponte H apresentado anteriormente (Figura 9), podemos dividir os transistores em dois grupos: aqueles da metade superior, representando as chaves 1 e 2, são os *High-side FET's*, e o par da metade inferior, representando as chaves 3 e 4, são os *Low-side FET's*. Esta é uma nomenclatura bastante utilizada quando queremos nos referir à posição do transistor quanto à sua carga.

O acionamento dos MOSFET's até sua condução em saturação requer que uma tensão superior à chamada "tensão de limiar", representada por  $V_{th}$ , seja aplicada em  $V_{GS}$  (*gate - source*, ou porta - fonte). Uma vez que  $V_{th}$  é um valor tipicamente em torno de 2 V a 4 V, basta aplicar, por exemplo, um valor de 5 V em  $V_{GS}$  para colocar o MOSFET em saturação (VERONESE, 2016). Para os transistores da parte inferior, aplicar essa tensão  $V_{GS}$  não representa grandes dificuldades, uma vez que o terminal de fonte (*source*) está ligado diretamente no referencial de terra (*GND*).

Os transistores *High-side*, por outro lado, têm seu *source* conectado à carga, portanto a tensão  $V_S$  varia praticamente entre *GND* (*Low-side* conduzindo) e  $V_{bat}$  (*High-side* conduzindo). Neste caso, aplicar 5 V (referenciados ao *GND*) em  $V_G$  não é eficiente para colocar o transistor em saturação, por conta dessa flutuação de  $V_S$  que influencia  $V_{GS}$ .

Por conta desta situação, foram projetados circuitos e chips a fim de acionar os MOSFET's, os chamados "*gate drivers*". A seguir estão dois exemplos desses circuitos:

### 2.3.1 Charge Pump

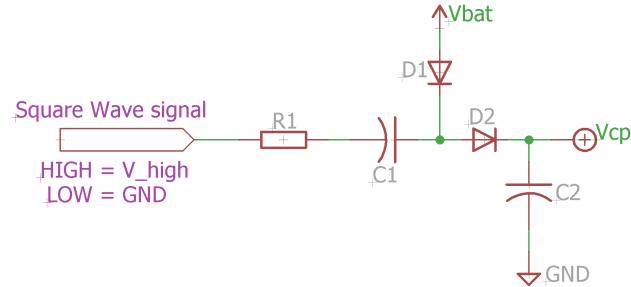
Uma forma de garantir a polarização  $V_{GS} \geq V_{th}$  é aplicar em  $V_G$  uma tensão igual ou maior que  $V_{th} + V_{bat}$ ; desta forma, considerando componentes ideais, na saturação do transistor temos  $V_S = V_{bat}$  e  $V_{GS} \geq V_{th}$ . A ideia de acionar o *gate* do MOSFET com uma tensão absoluta maior que  $V_{bat}$  é a essência do *Charge pump*.

Algo a ser mencionado é que  $V_{GS}$  também possui uma amplitude máxima permitida, que fica por volta de 20 V; acima dela, o componente pode ser permanentemente danificado.

O próximos passos necessários para completar a solução do *Charge pump* são a geração de uma tensão superior a  $V_{bat}$ , e uma forma de aplicar essa tensão nos transistores segundo os tempos do PWM pretendido.

Para o primeiro passo podemos utilizar os circuitos grampeadores de tensão, ou *clampers*, cuja função é disponibilizar uma tensão contínua cuja amplitude é (idealmente) igual à soma de outras duas tensões contínuas - na verdade uma tensão constante e outra onda quadrada. A Figura 10 mostra esse tipo de circuito, com componentes ideais.

Figura 10: Circuito grampeador de tensão, ou *clamper*.

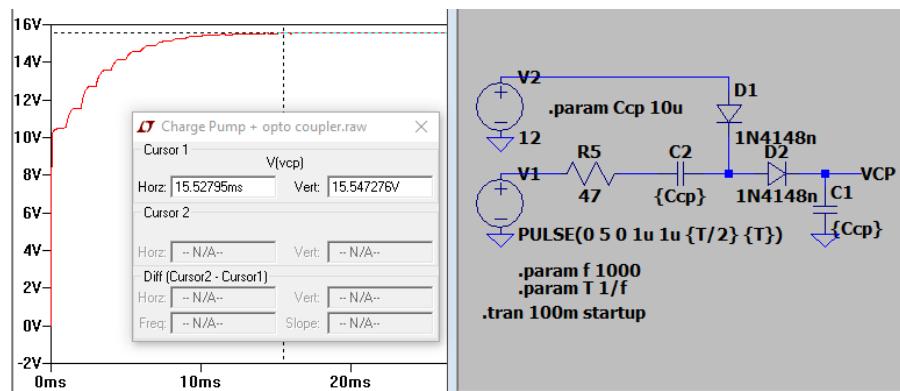


Fonte: Elaborado pelo autor

A onda quadrada varia entre  $V_{high}$ , em nível lógico alto, e 0 V (*GND*), em nível lógico baixo. Uma análise mais aprofundada fica a cargo do leitor, porém destaca-se que após alguns ciclos da onda quadrada, a saída  $V_{CP}$  atinge a amplitude  $V_{high} + V_{bat}$ . O resistor  $R_1$  tem a simples função de limitar a corrente que a onda quadrada pode fornecer, e influencia, assim como a relação de  $C_1$  e  $C_2$ , na quantidade de ciclos necessários para a estabilização de  $V_{CP}$ .

A Figura 11 a seguir mostra o funcionamento do circuito com componentes reais, onde a tensão final é prejudicada principalmente pelas perdas dos diodos (tensões de condução direta). No circuito,  $V_{bat} = 12$  V e  $V_{high} = 5$  V, o que resultaria em uma tensão  $V_{CP} = 17$  V se não fossem as perdas mencionadas que contabilizam 1,5 V a menos, de modo que a tensão final praticamente estabiliza em 15,5 V.

Figura 11: Simulação da saída de um circuito grampeador de tensão.



Fonte: Elaborado pelo autor

A onda quadrada de 5 V tem frequência de 1 kHz, portanto com apenas 15 ciclos já podemos considerar que a tensão de *Charge pump* está disponível para ser utilizada no chaveamento dos transistores.

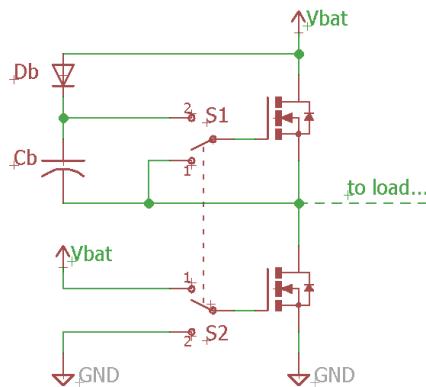
O circuito apresentado nas Figuras 10 e 11 não possui nenhuma carga em sua saída; caso houvesse, ela descarregaria o capacitor de saída conforme ele fosse carregado, influenciando a tensão média final e causando uma oscilação, um *ripple*, em sua amplitude. Por isso é importante que uma carga eventualmente alimentada por essa tensão  $V_{CP}$  seja baixa, e é por isso que este circuito é uma solução tão boa para esta aplicação: os *gates* dos transistores representam apenas uma capacidade parasita, de valor baixo se comparada à dos capacitores utilizados no circuito grampeador, portanto eles são essa baixa carga que atende aos requisitos de consumo da tensão  $V_{CP}$  - desde que o circuito seja bem projetado.

### 2.3.2 Bootstrap

Uma outra forma de garantir a polarização  $V_{GS} > V_{th}$  é por meio de uma tensão "flutuante", ou seja, uma diferença de potencial que pode mudar seu referencial; esta é a proposta do circuito de *bootstrap*, que utilizando-se de um capacitor e um diodo, consegue criar essa tensão flutuante para polarização dos transistores *High-side*.

O circuito da Figura 12 ilustra a estrutura básica do circuito de *bootstrap*, utilizando as chaves S1 e S2 para um propósito didático.

Figura 12: Estrutura de um circuito de *bootstrap*.



Fonte: Elaborado pelo autor

A chaves S1 e S2 têm comutação simultânea, ambas inicialmente ligadas ao seus contatos de número 1. Neste cenário, a porta (*gate*) do transistor superior está em curto com seu terminal de fonte (*source*), colocando-o em corte; a chave S2, por outro lado, aplica  $V_{bat}$  na porta do transistor inferior, de modo que ele conduz em saturação.

Uma vez que o transistor *Low-side* está saturado, isto abre caminho para o capacitor  $C_b$  se carregar com  $V_{bat}$  por meio do diodo  $D_b$  - considerado ideal - quase que instantaneamente, dada a baixa resistência do MOSFET e do diodo em condução.

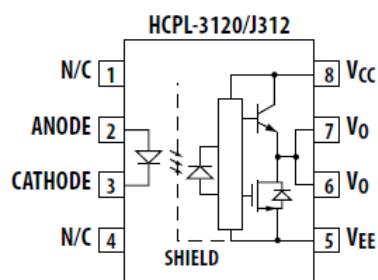
Agora que  $C_b$  está carregado com  $V_{bat}$ , trocamos a posição das chaves para o contato de número 2: o transistor inferior passa a ter aplicado *GND* em sua porta, portanto

encontra-se em corte. O transistor superior, por sua vez, passa a ter o capacitor em paralelo com seus terminais *gate* e *source*, e  $C_b$  estando carregado com  $V_{bat}$  coloca o transistor *High-side* em saturação. A capacitância parasita de entrada do transistor, que tem um valor baixo, absorve uma certa carga de  $C_b$ , diminuindo um pouco sua tensão, porém o mesmo é novamente carregado em um novo ciclo de chaveamento.

O capacitor  $C_b$  faz o papel da tensão "flutuante" mencionada anteriormente, pois seu referencial muda entre *GND* (*High-side* em corte e *Low-side* conduzindo) e  $V_{bat}$  (*High-side* conduzindo e *Low-side* em corte). Algo interessante de se notar é que o diodo  $D_b$  possibilita a carga de  $C_b$  mas impede a descarga, pois fica reversamente polarizado quando o transistor *High-side* está conduzindo.

Na prática, as chaves S1 e S2 podem ser substituídas por opto-acopladores, como o modelo HCPL-3120 da Avago, mostrado na Figura 13, em que um sinal de valor baixo controla o acionamento do LED de entrada, que por sua vez comanda a ligação de  $V_O$ , de saída, com  $V_{CC}$  ou  $V_{EE}$ .

Figura 13: Diagrama interno do opto-acoplador HCPL-3120, da Avago.

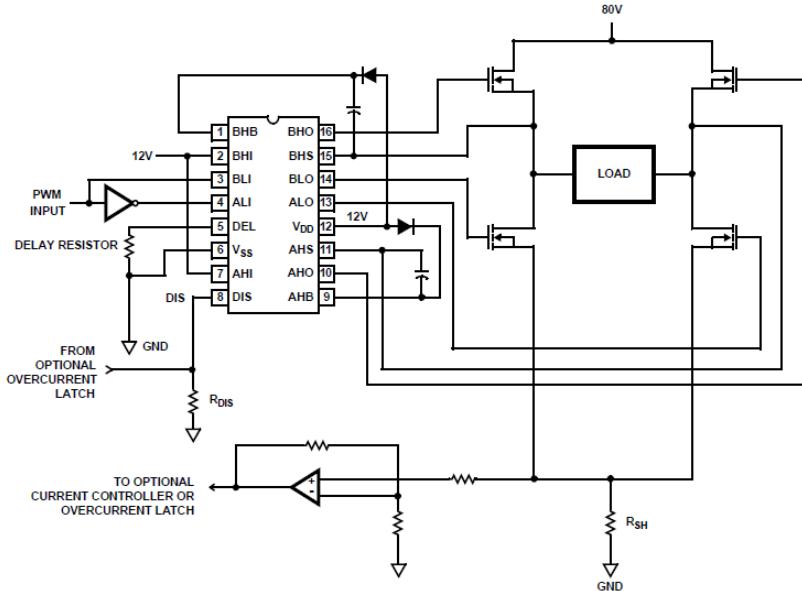


Fonte: Extraído do *datasheet* do HCPL-3120, de 21/03/2016, página 2.

Além dos opto-acopladores, chips específicos foram desenvolvidos para realizar esta interface de chaveamento dos MOSFET's de uma ponte H por meio do *bootstrap*, como é o caso do HIP4082, fabricado pela Intersil. A Figura 14 mostra uma típica aplicação de seu uso, extraída de seu *datasheet*, onde é possível verificar a presença de dois pares de capacitores e diodos de *bootstrap*, a interface de comando onde aplica-se o sinal PWM (indicado na figura por "*PWM input*"), e outras funções adicionais como a medição de sobrecorrente que pode ser acrescentada com o circuito auxiliar do amplificador operacional.

O HIP4082 conta ainda com a proteção contra "shoot through", ou seja, o acionamento dos dois transistores de um mesmo lado/ramo da ponte H, colocando a fonte de alimentação da mesma em curto circuito; no chip foi implementada uma relação de prioridade entre os comandos de um mesmo lado, de modo que, caso ambas recebam o sinal de acionamento, apenas a saída para o *Low-side* seja ligada.

Figura 14: Exemplo de aplicação do HIP4082 no acionamento de uma ponte H.



Fonte: Extraído do *datasheet* do HIP4082, de 03/01/2006

## 2.4 Microcontrolador ATmega328P

O microcontrolador é um chip criado basicamente para receber e executar uma programação; dispondo de uma memória não-volátil (geralmente memória *flash*) para armazenamento do programa a ser executado, uma unidade de processamento interna, pinos de interface (chamados de GPIO) entre o programa e o circuito em que está inserido, temporizadores internos (comumente chamados de *timers*, a nomenclatura em inglês), conversores analógicos-digitais (ADC's), entre diversos outros recursos adicionais, dependendo do modelo de microcontrolador.

Tratando-se de um tópico bastante amplo, aqui focaremos no microcontrolador escolhido para o desenvolvimento deste projeto, o modelo ATmega328P da Atmel/Microchip, utilizado na plataforma Arduino Uno<sup>2</sup>. Suas características, extraídas do *datasheet* do microcontrolador, são principalmente as listadas a seguir:

- 32 kB de memória de programa (*flash*);
- 1 kB de memória EEPROM (armazenamento não-volátil);
- 2 kB de memória RAM (volátil, auxiliar para o processamento);
- *Clock* de até 20 MHz;

<sup>2</sup> Fonte: <<https://store.arduino.cc/usa/arduino-uno-rev3>>, na seção *Tech Specs*; acessado em 05/04/2020.

- 17 pinos GPIO, dentre os quais 6 possuem saídas PWM;
- 6 pinos de entrada para o ADC interno;
- Interface de comunicação SPI pronta para utilização;
- Interface de comunicação serial (UART);
- 3 *timers* internos, com configuração para atuarem como temporizadores, contadores ou máquinas PWM;
- *Watchdog* interno, que basicamente é um recurso para monitorar o bom funcionamento do programa - se implementado;
- Tensão de alimentação de 5 V.

O fato de o ATmega328P integrar o Arduino Uno facilita alguns aspectos do desenvolvimento, uma vez que a plataforma possui uma IDE<sup>3</sup> própria, e não necessita de nenhum gravador específico, ao ponto que o conjunto de comunicação USB, IDE e placa do Arduino são suficientes para atualizações no código. Além disso, a comunicação USB também é bastante útil para interface com o programa, seja para inserção de dados durante sua execução, seja para o simples acompanhamento de algum dado lido (como o *duty-cycle* calculado a partir de uma entrada analógica via potenciômetro, por exemplo). Um último aspecto bastante importante nesse sentido de o microcontrolador ser usado pelo Arduino, é a vasta comunidade de desenvolvedores da plataforma, e a grande quantidade de bibliotecas já criadas para facilitar o desenvolvimento de *software* nos mais diversos projetos e aplicações.

#### 2.4.1 *Timers* do ATmega328P

Um tipo de módulo que compõe o microcontrolador mencionado são os temporizadores/contadores internos, que além de suas funções mais óbvias de contagem de pulsos e cronômetro, também podem ser usados para gerar interrupções programadas, medição da largura de pulsos, e geração de sinais de PWM com frequência e *duty-cycle* ajustáveis (SOUZA, 2015). Essas funcionalidades podem, portanto, ter interface externa ao chip, como ocorre na interação com pulsos, ou não, como é o exemplo da temporização. Para os modos de operação que contam com a interface externa, cada um dos 3 *timers* disponíveis possui um par de terminais do chip relacionados a si, conforme indicado na Tabela 1 e na Figura 15. A Tabela 1 também indica que um dos três possui o dobro de bits para sua operação, ou seja, enquanto dois deles contam até  $2^8 - 1 = 255$ , esse outro consegue contar até  $2^{16} - 1 = 65535$ .

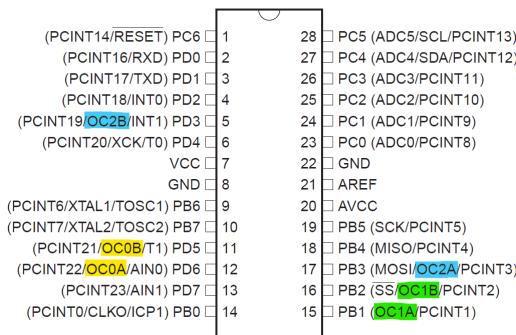
---

<sup>3</sup> *Integrated Development Environment*, ou Ambiente de Desenvolvimento Integrado

Tabela 1: *Timers* do ATmega328P

Identificação	Sigla	Nº de bits	Pinos associados
Timer/Counter 0	TMR0	8	OC0A e OC0B
Timer/Counter 1	TMR1	16	OC1A e OC1B
Timer/Counter 2	TMR2	8	OC2A e OC2B

Fonte: *Datasheet* do ATmega328P, de 11/2015

Figura 15: Terminais correspondentes a cada *timer* do ATmega328P.

Fonte: Extraído do *datasheet* do ATmega328P, de 11/2015

O configuração de cada *timer* se dá por variáveis específicas de cada um, variáveis estas armazenadas em registradores, que são endereços de memória interna. Dado o caráter do projeto, aprofundaremos na função de PWM dos *timers*: cada um deles possui as duas saídas associadas, OCxA e OCxB, onde x indica o número do *timer* utilizado, conforme explicitado pela Tabela 1.

São dois modos de operação para a geração do PWM, modos chamados de "*Fast PWM*" e "*Phase correct*". No modo *Fast PWM* o *timer* dispara sua contagem crescente, de zero até o valor máximo (que depende do *timer* utilizado), e voltando a zero em seguida, pelo estouro da contagem; ao mesmo tempo, um comparador interno é responsável por relacionar o valor do *timer* com um valor previamente estabelecido e armazenado, alterando o valor da saída correspondente assim que eles se igualam<sup>4</sup>, e alterando novamente a saída assim que o *timer* vai a zero. Apenas para esclarecer, a alteração da saída significa a mudança do seu nível lógico: se estava em nível lógico alto, vai para nível lógico baixo, e vice-versa.

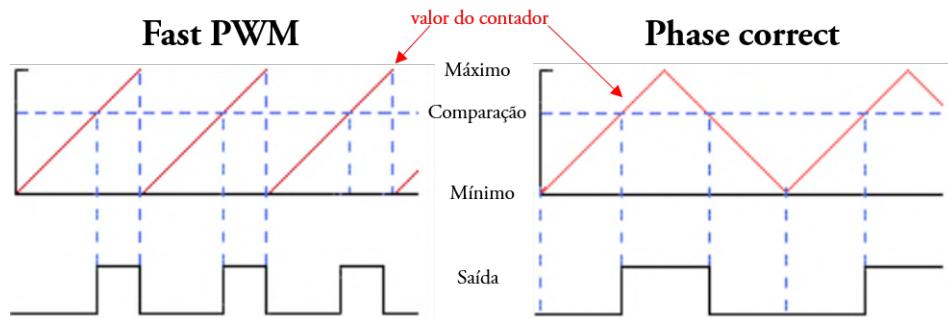
No modo *Phase correct*, por outro lado, o *timer* também dispara de zero até chegar ao seu valor máximo numa contagem crescente, porém ao atingir o máximo, ao invés de retomar do zero, ele inverte a contagem para ser decrescente, chegando até zero e

<sup>4</sup> Na verdade, no modo *Fast PWM* a alteração da saída ocorre na próxima iteração do *timer*; no modo *Phase correct* isso não ocorre.

invertendo para uma contagem crescente novamente, conforme indica a imagem à direita na Figura 16. Quanto à comparação interna, sempre que o valor do *timer* iguala-se ao valor pré-definido, a saída se altera, porém neste modo ela não se altera quando o *timer* chega em zero, como ocorre no *Fast PWM*.

A Figura 16 nos ajuda a entender melhor o mecanismo de funcionamento de cada modo. O eixo horizontal em cada gráfico representa o tempo, o traço em vermelho indica o valor do *timer*, e o eixo vertical possui três indicações: os valores máximos e mínimos de contagem, e o valor de comparação pré-definido, no qual a saída se altera. No diagrama, a saída começa em nível lógico baixo para ambos os modos de funcionamento, como um exemplo, porém este é um parâmetro configurável.

Figura 16: Diagrama temporal dos modos de PWM do ATmega328P.



Fonte: Modificado de <<https://bit.ly/3flLrSQ>>, acesso em 12/07/2020.

Observando o diagrama, notamos que o tempo para contagem do *timer*, entre mínimo e máximo, é o que determina a frequência de saída do PWM, e o valor de comparação estabelecido é responsável por controlar o *duty-cycle* de saída. Para um mesmo *timer*, temos as duas saídas associadas (Tabela 1), e apesar de o valor de contagem ser o mesmo para ambas, portanto a mesma frequência, o valor de comparação de cada uma pode ser alterada por seus registradores individuais; assim também o é com o comportamento destas saídas (começo da contagem em nível lógico alto ou baixo), consequentemente podem ter *duty-cycles* diferentes, ou até complementares, com uma saída invertida em relação à outra.

Entrando um pouco nos parâmetros de configuração de cada *timer*, as variáveis que controlam sua operação estão armazenadas principalmente em dois registradores, TCCRnA e TCCRnB<sup>5</sup>, onde *n* indica sobre qual *timer* o registrador diz respeito. Apesar de terminarem em A e B, é o conjunto dos dois que controla a operação das duas saídas do *timer* em questão, então não são separados em um para cada saída daquele gerador de PWM. Isso tanto é verdade que os principais bits de controle da operação do *timer* estão

<sup>5</sup> TCCR = Timer/Counter Control Register.

divididos entre esses dois registradores, que são os bits de WGM<sup>6</sup>. Além dos registradores citados, os outros de comparação também são muito importantes para o controle de *duty-cycle* das saídas PWM, chamados de OCRnA e OCRnB<sup>7</sup>; estes sim estão designados para suas respectivas saídas, de modo geral<sup>8</sup>.

A fim de ilustrar melhor como os bits de configuração controlam o modo de operação dos *timers* e suas saídas, peguemos o TMR2 como exemplo. Na Figura 17, a seguir, estão apresentados os registradores de configuração do *timer* 2, com os bits de controle armazenados entre eles. O grupo de bits WGM, por exemplo, tem dois bits armazenados em TCCR2A e outro armazenado em TCCR2B.

Figura 17: Registradores TCCR2A e TCCR2B de controle da operação do TMR2.

#### TCCR2A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
(0xB0)	COM2A1	COM2A0	COM2B1	COM2B0	-	-	WGM21	WGM20	TCCR2A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

#### TCCR2B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
(0xB1)	FOC2A	FOC2B	-	-	WGM22	CS22	CS21	CS20	TCCR2B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fonte: Adaptado do *datasheet* do ATmega328P, de 11/2015, páginas 153 e 156.

Começando, então, pelo parâmetro WGM, citando a própria folha de dados do ATmega328P, segundo tradução livre, trata-se dos bits que controlam a sequência de contagem, a referência para seu valor máximo de contagem, e o tipo de geração da forma de onda a ser usado, entre as opções de modo normal (contador), reinício da contagem pela igualdade de comparação (ou CTC - *Clear Timer on Compare Match*), e os dois modos de geração de PWM, *Fast PWM* e *Phase correct*. A Tabela 2, a seguir, expõe as combinações de valores para os bits WGM, e o correspondente modo de operação.

Uma explicação mais detalhada dos modos de operação do TMR2 pode ser encontrada a partir da página 146 do *datasheet* do ATmega328P. Em uma breve abordagem de tais modos, pode-se dizer que o primeiro da Tabela 2, Normal, é simplesmente o *timer* funcionando como um contador ininterrupto, contando de 0 a 255 (0xFF, em hexadecimal), por ter 8 bits, sempre de forma crescente, e sem interação direta com o terminais de saída

<sup>6</sup> WGM = *Waveform Generation Mode*.

<sup>7</sup> OCR = *Output Compare Register*.

<sup>8</sup> Existe um modo de operação da saída PWM pelos *timers* que o valor máximo da contagem é determinado por OCRnA, de modo que esta saída fica indisponível para controle de *duty-cycle*, mas a saída OCnB passa a ter disponível um ajuste mais fino da frequência do PWM.

Tabela 2: TMR2 - Parâmetro WGM - *Waveform Generation Mode.*

Mode	WGM22	WGM21	WGM20	Timer/Counter Mode of Operation	TOP	Update of OCR <sub>x</sub> at	TOV Flag Set on <sup>(1)(2)</sup>
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	-	-	-
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	-	-	-
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Notes:

1. MAX= 0xFF
2. BOTTOM= 0x00

Fonte: Extraído do *datasheet* do ATmega328P, de 11/2015, página 155.

do microcontrolador. Passando para o modo CTC (*Clear Timer on Compare Match*), trata-se da contagem de zero até o valor configurado no registrador OCR2A, sempre crescente também; ao alcançar este valor, o contador é reiniciado e a saídas OC2A e OC2B têm seu comportamento ditado pelos parâmetros COM, explicados mais à frente.

Passando para os modos mais relevantes ao propósito deste trabalho, relacionados à geração de sinal PWM, verificamos que existem duas linhas para cada modo de *Fast PWM* e *Phase correct*. Isto ocorre porque cada modo pode considerar o valor máximo de contagem para o *timer* como o limite dado pelos 8 bits, portanto 255, ou pode ser configurado para contar até o valor armazenado no registrador OCR2A, variando entre 0 e 255. Nas primeiras ocorrências da Tabela 2, ambos os modos são configurados para o TMR2 contar até 255 (0xFF), e nas duas últimas ocorrências (antepenúltima e última linhas) a contagem vai até o valor de OCR2A.

O funcionamento de cada modo de PWM já foi explicado e está ilustrado pela Figura 16. O impacto de alterar o valor máximo de contagem é o aumento da frequência do sinal de saída, pois o *timer* ainda leva o mesmo tempo para incrementar-se em uma unidade (mesmo *clock*), porém agora tem menos valores para contar; há também a diminuição da resolução para controle do *duty-cycle* de saída: com a contagem de 0 a 255, a comparação tinha uma precisão de  $1/255 = 0,39\%$  para o passo de alteração de  $\delta$ , valor este que diminui para  $1/OCR2A$ ,  $0 \leq OCR2A \leq 255$  — ou seja, supondo  $OCR2A = 180$ , o passo de controle de  $\delta$  passa a ser  $0,56\%$ . Uma última desvantagem de se utilizar OCR2A para limitar a contagem é a impossibilidade de utilizá-lo para controlar a saída OC2A, portanto esta não pode mais ser usada como um sinal de PWM controlado, apesar de

ainda poder apresentar um sinal com 50 % de *duty-cycle*, e apenas OC2B ficar disponível, com frequência e *duty-cycle* ajustáveis, respectivamente, por OCR2A e OCR2B.

O próximo parâmetro a ser tratado é o COM, *Compare Output Mode*, que controla o comportamento da sua respectiva saída dentro do modo de operação PWM selecionado. Existem, portanto, dois pares de bits: COM2A, pertinente à saída OC2A, e COM2B, pertinente à saída OC2B.

Dado que o efeito de COM depende do modo de operação selecionado em WGM, entre modo não-PWM, *Fast PWM* e *Phase correct*, cada um dos 3 casos deve ser tratado separadamente. Algo a ser observado de antemão, independentemente do modo escolhido, é que a configuração de COM2x, sendo x igual a A ou B, pode sempre desassociar a saída OC2x do terminal do microcontrolador, de modo que toda a operação ainda continua ocorrendo no TMR2, mas não é aplicada no respectivo pino do ATmega328P, e este funcione como um simples terminal de entrada e saída digital; isso ocorre quando o valor COM2x = 0.

Começando pelo modo de operação que não gera PWM, ou seja, o CTC (pois o modo "normal" não gera sinais de saída), a Tabela 3 nos ajuda a visualizar cada comportamento possível: OC2x pode alternar (*toggle*) o seu valor quando o TMR2 alcança o valor armazenado em OCR2A, ou pode jogar sua saída sempre para nível lógico baixo ou alto.

Tabela 3: TMR2 - Parâmetro COM - *Compare Output Mode* - em operação não-PWM.

<b>COM2A1</b>	<b>COM2A0</b>	<b>Description</b>
0	0	Normal port operation, OC0A disconnected.
0	1	Toggle OC2A on Compare Match
1	0	Clear OC2A on Compare Match
1	1	Set OC2A on Compare Match
<b>COM2B1</b>	<b>COM2B0</b>	<b>Description</b>
0	0	Normal port operation, OC2B disconnected.
0	1	Toggle OC2B on Compare Match
1	0	Clear OC2B on Compare Match
1	1	Set OC2B on Compare Match

Fonte: Extraído do *datasheet* do ATmega328P, de 11/2015, páginas 153 e 154.

Passando para o modo *Fast PWM*, a Tabela 4 indica o comportamento para cada caso possível. Como mencionado anteriormente, a contagem do TMR2 pode ir até o máximo possível pelos 8 bits (WGM22 = 0), ou até o valor definido em OCR2A (WGM22 = 1), interferindo no comportamento das saídas como nota-se pela segunda linha da combinação (COM2A = "01"). Neste caso, OC2A pode ficar desconectado e o terminal

do chip servir como I/O<sup>9</sup> digital ( $WGM22 = 0$ ), ou alternar seu nível lógico sempre que TMR2 atingir o valor de OCR2A ( $WGM22 = 1$ ), resultando em uma onda quadrada com  $\delta = 50\%$ , e com duas vezes a frequência do PWM em OC2B; COM2B = "01" não é permitido.

Além disso, temos as outras combinações para OC2x: "10" configura a respectiva saída para começar a contagem em nível lógico baixo (aprox. 0 V na saída), e trocar seu valor para nível lógico alto (aprox. 5 V na saída) quando a contagem atinge o valor predeterminado em OCR2x; "11" faz o contrário, começa em nível baixo e muda para nível alto ao cruzar o valor especificado.

Tabela 4: TMR2 - Parâmetro COM - *Compare Output Mode* - em operação *Fast PWM*.

<b>COM2A1</b>	<b>COM2A0</b>	<b>Description</b>
0	0	Normal port operation, OC2A disconnected.
0	1	$WGM22 = 0$ : Normal Port Operation, OC2A Disconnected. $WGM22 = 1$ : Toggle OC2A on Compare Match.
1	0	Clear OC2A on Compare Match, set OC2A at BOTTOM, (non-inverting mode).
1	1	Set OC2A on Compare Match, clear OC2A at BOTTOM, (inverting mode).
<b>COM2B1</b>	<b>COM2B0</b>	<b>Description</b>
0	0	Normal port operation, OC2B disconnected.
0	1	Reserved
1	0	Clear OC2B on Compare Match, set OC2B at BOTTOM, (non-inverting mode).
1	1	Set OC2B on Compare Match, clear OC2B at BOTTOM, (inverting mode).

Fonte: Extraído do *datasheet* do ATmega328P, de 11/2015, páginas 153 e 154.

Assim como no modo *Fast PWM*, no modo *Phase correct* também é possível configurar a contagem do TMR2 até o valor de OCR2A ( $WGM22 = 1$ ), e COM2A = "01" depende dessa definição para configurar o comportamento da saída OC2A. Quanto às outras configurações, COM2x = "10" reflete em OC2x em nível lógico alto enquanto a contagem é inferior a OCR2x, e nível lógico baixo enquanto a contagem está superior a OCR2x. Para COM2x = "11" ocorre o contrário, a contagem começa em nível baixo e vai para nível alto quando cruza o valor predefinido, durante a contagem crescente, voltando para nível baixo quando cruza novamente o valor predefinido, durante a contagem decrescente.

Quanto aos bits FOC2x, armazenados no registrador TCCR2B, sua sigla significa *Force Output Compare* e ele só é ativo para modos de operação não-PWM, portanto não

<sup>9</sup> I/O = *Input/Output*, ou seja, terminal de entrada ou saída.

Tabela 5: TMR2 - Parâmetro COM - *Compare Output Mode* - em operação *Phase correct*.

<b>COM2A1</b>	<b>COM2A0</b>	<b>Description</b>
0	0	Normal port operation, OC2A disconnected.
0	1	WGM22 = 0: Normal Port Operation, OC2A Disconnected. WGM22 = 1: Toggle OC2A on Compare Match.
1	0	Clear OC2A on Compare Match when up-counting. Set OC2A on Compare Match when down-counting.
1	1	Set OC2A on Compare Match when up-counting. Clear OC2A on Compare Match when down-counting.
<b>COM2B1</b>	<b>COM2B0</b>	<b>Description</b>
0	0	Normal port operation, OC2B disconnected.
0	1	Reserved
1	0	Clear OC2B on Compare Match when up-counting. Set OC2B on Compare Match when down-counting.
1	1	Set OC2B on Compare Match when up-counting. Clear OC2B on Compare Match when down-counting.

Fonte: Extraído do *datasheet* do ATmega328P, de 11/2015, páginas 154 e 155.

interessam ao propósito deste projeto. Segundo recomendação do *datasheet* do ATmega328P (de 11/2015, página 156), ambos (FOC2A e FOC2B) devem ser escritos como zero durante modos de operação PWM.

A última variável armazenada nos registradores principais são os três bits do *Clock Select*, CS, que configuram o *prescaler*, que é o fator de divisão do *clock* principal para ser usado como frequência de incremento do *timer*; por exemplo, aplicando um *clock* de 16 MHz no ATmega328P, e selecionando um *prescaler* de 32, significa que a frequência do *timer* é 32 vezes menor que a de referência, portanto  $16 \cdot 10^6 / 32 = 500$  kHz, ou um incremento de contagem a cada  $2 \mu\text{s}$ .

Cada *timer* deste microcontrolador possui diferentes opções de valores para o *prescaler*, porém ele é sempre configurado pelos três bits do CS. Pegando o exemplo do TMR2, as opções disponíveis estão exibidas na Tabela 6.

A primeira linha da Tabela 6 desativa o incremento do *timer* e para a contagem; da segunda linha em diante, o valor do *prescaler* aumenta de 1 (próprio *clock* do microcontrolador) até 1024. É importante destacar que a frequência calculada aqui é a de incremento da contagem, e não a de saída do PWM, quando o *timer* está configurado neste modo.

O cálculo da frequência do PWM de saída dos modos de operação do TMR2, que não utilizam OCR2A como máximo, portanto com ambas as saídas (OC2A e OC2B) habilitadas, está apresentado na equação 2.5 para o modo *Fast PWM*, e na equação 2.6 para o modo *Phase correct*<sup>10</sup>.

<sup>10</sup> Retiradas do *datasheet* do ATmega328P, de 11/2015, páginas 148 e 149.

Tabela 6: TMR2 - Parâmetro CS - *Clock Select*.

<b>CS22</b>	<b>CS21</b>	<b>CS20</b>	<b>Description</b>
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$\text{clk}_{\text{T2S}}/(No \text{ prescaling})$
0	1	0	$\text{clk}_{\text{T2S}}/8$ (From prescaler)
0	1	1	$\text{clk}_{\text{T2S}}/32$ (From prescaler)
1	0	0	$\text{clk}_{\text{T2S}}/64$ (From prescaler)
1	0	1	$\text{clk}_{\text{T2S}}/128$ (From prescaler)
1	1	0	$\text{clk}_{\text{T2S}}/256$ (From prescaler)
1	1	1	$\text{clk}_{\text{T2S}}/1024$ (From prescaler)

Fonte: Extraído do *datasheet* do ATmega328P, de 11/2015, páginas 156 e 157.

$$f_{\text{PWM}} = \frac{f_{\text{clock}}}{256.\text{Prescaler}} \quad \longrightarrow \text{Fast PWM} \quad (2.5)$$

$$f_{\text{PWM}} = \frac{f_{\text{clock}}}{510.\text{Prescaler}} \quad \longrightarrow \text{Phase correct} \quad (2.6)$$

Cada um dos *timers* tem suas características específicas: número de bits, terminais de saída no chip do microcontrolador, modos de operação adicionais, valores de *prescaler*; entretanto, o TMR2 explorado aqui serve como uma boa base de referência para o funcionamento da geração de sinais PWM com o ATmega328P. Informações mais aplicadas e a manipulação prática desse recurso pode ser encontrada na seção de desenvolvimento deste trabalho.

## 2.5 Rádio Controle

Muito utilizado nas categorias de aeromodelismo, nautimodelismo e automodelismo, o rádio controle é a interface de comando para o operador acionar os movimentos do robô, isto é, por meio das alavancas e botões disponibilizados no rádio controle, o operador determina a direção e a velocidade de locomoção. Dependendo dos recursos do modelo utilizado, como a quantidade de canais, outros aspectos além do movimento também podem ser controlados, como a rotação de uma "arma ativa", por exemplo, no caso do combate de robôs.

Uma característica muito importante do rádio controle é sua quantidade de canais. Um canal é uma variável a ser transmitida, podendo ser analógica<sup>11</sup>, como a posição de uma alavanca, ou digital binária, como o estado de uma chave ou botão. A partir disso,

<sup>11</sup> Na verdade, apesar de a alavanca e o sensor serem analógicos, o circuito interno realiza uma discretização para transmissão.

o movimento de um robô no plano pode ser controlado por dois canais: um destinado a controlar a velocidade de deslocamento para frente e para trás, e outro canal relacionado às curvas para a direita e a esquerda, e quanto acentuadas elas devem ser.

Os canais são combinados, multiplexados e modulados para transmissão por rádio-frequência, captados pelo receptor instalado dentro do robô, demodulados, demultiplexados e disponibilizados para leitura por outro sistema, o qual é responsável pelo processamento e tradução daqueles sinais em movimento efetivo. A tecnologia usada em todo esse processo de transmissão e recepção é bastante complexa para ser tratada com profundidade neste texto, mas seus fundamentos são abordados durante o curso de engenharia elétrica, assim como brevemente nas referências ([MEGGIOLARO, 2006](#)) e ([MEGGIOLARO, 2009](#)).

A fim de exemplificar esse funcionamento do rádio controle de forma mais prática, tomemos o modelo utilizado neste projeto, o "TURNIGY 5x" de cinco canais, identificado na Figura 18. Suas duas alavancas podem ser movidas nas duas direções (horizontal e vertical), sendo cada uma dessas direções responsável por um canal analógico, de modo que ambas concentram quatro dos cinco canais deste rádio controle. O último canal disponível é controlado pela chave na parte superior esquerda, indicada pela escrita CH5, que tem duas posições apenas.

Figura 18: Rádio controle TURNIGY 5x, de 5 canais, e seu receptor, utilizados neste projeto.



Fonte: Elaborado pelo autor.

O receptor, mostrado no detalhe à esquerda da Figura 18, exibe cinco linhas de 3 terminais cada, em que cada linha é responsável por disponibilizar um canal para leitura. Da esquerda para a direita, os terminais seguem o padrão: "S" para o sinal com o valor do canal para leitura, "+" como o terminal para se aplicar o positivo da tensão de alimentação (corrente contínua de 4,8 V a 6 V, segundo o manual), e "-" para se aplicar o negativo da tensão de alimentação. Vale dizer que o terminal central e o da direita, pinos responsáveis pela alimentação do receptor, estão todos curto-circuitados entre si, ou seja, "+" com "+",

e "—" com "—", então a ligação de apenas um par deles já é suficiente para alimentar o dispositivo. Na verdade, caso os canais sejam lidos por sistemas diferentes que proveem tensão de alimentação para o receptor (caso comum para este padrão de conector triplo), apenas um deve conter todos os três terminais e alimentar o receptor, enquanto que os outros devem conter apenas o terminal para leitura, "S", e o terminal do negativo, "—", para servir de referencial de tensão; caso o terminal positivo dos outros também seja ligado com uma tensão diferente daquela que está alimentando o receptor, cria-se um paralelo de fontes de alimentação, e aquelas com tensão menor passam a drenar corrente ao invés de fornecê-la.

Os canais são disponibilizados individualmente segundo a sequência apresentada na Tabela 7, e sua nomenclatura segue o padrão de controle de aviões, dado que é um controle voltado para o aeromodelismo.

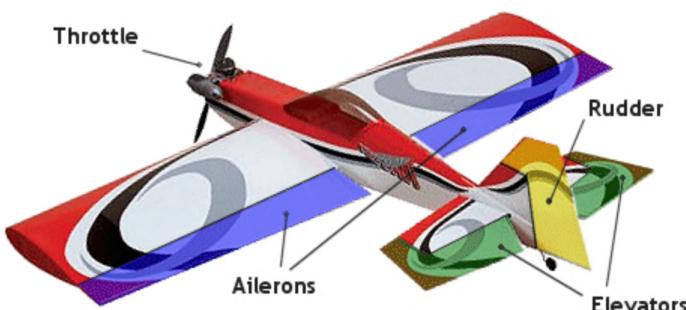
Tabela 7: Siglas dos canais do rádio controle e sua identificação.

Sigla	Identificação
CH5	<i>Channel 5</i>
RUD	<i>Rudder</i>
THR	<i>Throttle</i>
ELE	<i>Elevator</i>
AIL	<i>Aileron</i>

Fonte: Elaborado pelo autor com base no manual do TURNIGY 5x.

Apenas a título de curiosidade, a Figura 19 mostra a correlação entre os canais do rádio controle e quais são os aspectos do movimento do aeromodelo controlados por cada canal. O canal CH5 geralmente é utilizado para o recolhimento do trem de pouso após a decolagem.

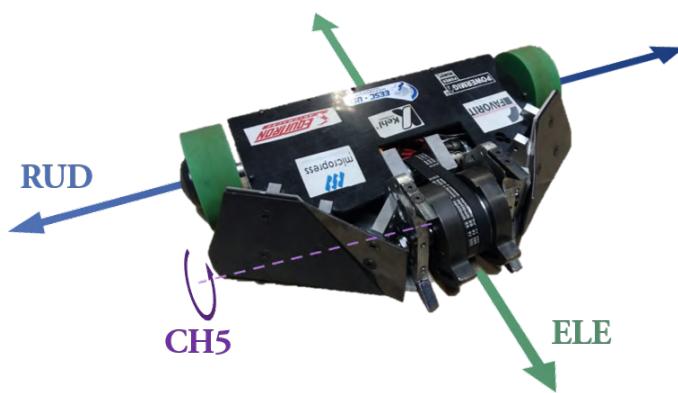
Figura 19: Aspectos do movimento de um aeromodelo controlados por cada canal do rádio controle.



Fonte: Extraído de <<https://bit.ly/3kWvVQe>>, em 01/07/2020.

Da mesma forma, podemos determinar os canais disponíveis para cada direção do movimento de um robô de combate, como pode ser visto exemplificado na Figura 20, que conta inclusive com o CH5 acionando a arma ativa.

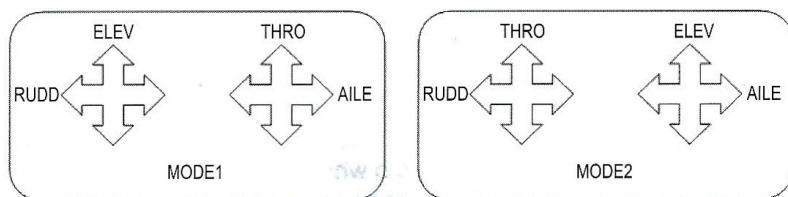
Figura 20: Possível alocação dos canais do rádio controle para os comandos de locomoção e acionamento da arma ativa de um robô de combate.



Fonte: Elaborado pelo autor.

A correlação de alavancas com os canais pode seguir dois modos, de acordo com o manual do controle. A chave, por outro lado, sempre comanda o canal CH5. A alteração de um modo para outro pode ser feita no próprio controle, com uma chave específica para isso, e ambos estão expostos na Figura 21.

Figura 21: Dois modos de configuração dos canais do rádio controle TURNIGY 5x.



Fonte: Extraído do manual do TURNIGY 5x.

Além desses citados, o rádio controle TURNIGY 5x dispõe de outros recursos, como o ajuste fino de desvio (*offset*) dos canais, ou a tecnologia FHSS<sup>12</sup>, que permite a troca da frequência portadora, aumentando a segurança e robustez contra interferência (MOTLAGH, 2011), ou ainda o emparelhamento entre transmissor e receptor por meio do botão *bind*; contudo, tais recursos não interessam ao propósito deste trabalho, portanto não serão abordados com maior profundidade neste texto.

<sup>12</sup> FHSS = Frequency Hopping Spread Spectrum.

### 2.5.1 Modulação por posição de pulso - PPM

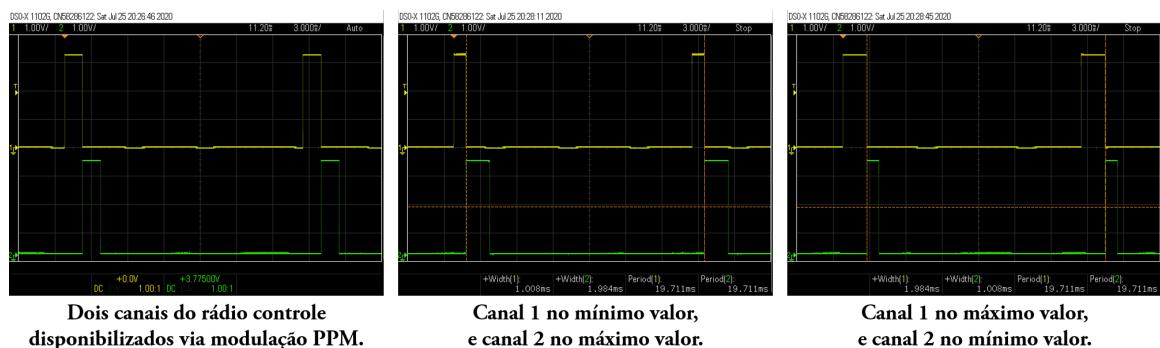
A disponibilização da informação dos canais não é feita de modo que volta a seu valor no momento da aquisição, mas ela segue um tipo de modulação chamado PPM, explicado a seguir. Em outras palavras, a posição da alavanca gera um sinal analógico de tensão, passa por todo o processo de transmissão e recepção, mas não aparece no respectivo canal de saída com o mesmo valor analógico inicial, mas sim seguindo a citada modulação PPM.

O nome PPM significa *Pulse-Position Modulation*, e seu funcionamento no rádio controle pode ser descrito como uma concatenação de pulsos. Cada pulso representa um canal (e pode ser lido no terminal correspondente), seu período é fixo de aproximadamente 20 ms, e sua largura (tempo em nível lógico alto) varia entre 1 ms e 2 ms, em uma relação linear com o valor analógico de comando. O cascamenteamento é perceptível quando olha-se para a leitura de todos os canais simultaneamente: no momento em que o primeiro canal vai para nível lógico baixo, o segundo vai para nível lógico alto, e assim por diante; passados 20 ms da subida de borda do primeiro canal (AIL, neste caso), o ciclo se repete.

A fim de esclarecer a linearidade do comando, caso a alavanca do controle no canal THR esteja em 40 % da sua excursão, por exemplo, o pulso do canal correspondente ficará em nível alto por 40 % da "excursão" possível entre 1 ms e 2 ms, ou seja, 1,4 ms.

A Figura 22 mostra a aplicação prática do controle dos canais ELE (canal 1 do osciloscópio, em amarelo) e THR (canal 2 do osciloscópio, em verde) do rádio controle TURNIGY 5x. Na primeira captura, à esquerda, vemos o aspecto geral da leitura dos respectivos terminais desses canais no receptor, onde fica claro o cascamenteamento anteriormente citado. Caso o osciloscópio tivesse mais entradas, mais canais do receptor poderiam ser lidos seguindo este mesmo cascamenteamento, com o canal AIL terminando logo antes do canal ELE, em amarelo, e o canal RUD começando logo ao fim do canal THR, em verde, seguido do pulso de CH5.

Figura 22: Exemplo da modulação PPM com dois canais do rádio controle.



Fonte: Elaborado pelo autor.

Nas outras capturas, ao centro e à esquerda, vemos a variação de largura de cada canal entre seus valores mínimo (1 ms) e máximo (2 ms), e como o cascataamento se mantém, assim como o período de aproximadamente 20 ms de ambos os sinais.

## 2.6 Bateria de Polímero de Lítio

A fonte da alimentação do robô deve ser por baterias, dado que não é possível mantê-lo ligado por fios a uma fonte externa. Dentre as opções disponíveis, as baterias de Lítio-Polímero, abreviadas por Li-Po ou simplesmente LiPo, graças aos progressos recentes na tecnologia de fabricação, são as opções comerciais que mais atendem aos requisitos de nossa aplicação: alta densidade de carga, sem "efeito de memória"<sup>13</sup>, alta corrente de descarga e prolongada vida útil.

Abordar o processo eletroquímico de funcionamento das baterias de LiPo não é de interesse deste trabalho, porém algumas características devem ser destacadas. A primeira é a tensão de operação de cada célula, de 3,7 V nominais, podendo variar de 4,2 V, quando completamente carregada, a até 3 V, quando descarregada; exceder tais valores, para cima ou para baixo, não é recomendado por diminuir a vida útil da célula. Pelo fato de apresentarem um valor relativamente baixo de tensão, é comum que as baterias sejam associadas em série, a fim de aumentar o valor de tensão final, e/ou em paralelo, com o objetivo de aumentar o valor de corrente que o conjunto pode fornecer.

Em seguida temos a capacidade de carga da célula, especificada em Wh (watt-hora), mAh (miliampere-hora) ou Ah (ampere-hora). Este parâmetro varia muito de acordo com a aplicação da bateria e seu tamanho físico, até dentro do combate de robôs depende de qual categoria de peso estamos tratando. Dada a grande gama de opções, tratemos daquela utilizada neste projeto, mostrada na Figura 23, que possui duas células associadas em série e 2200 mAh, portanto 1100 mAh em cada célula individualmente.

A Figura 23 mostra ainda, além da tensão de 7,4 V nominais resultante da associação série de duas células de 3,7 V, a indicação de sua característica 30C. Tal marcação refere-se à corrente de curto-círcuito da bateria, parametrizada pelo fabricante em relação à capacidade C de carga da bateria. No nosso caso, por exemplo, a bateria pode ser descarregada com uma corrente 30 vezes maior que 2200 mA, portanto 66 A, apenas por um curto período de tempo, a fim de evitar sobreaquecimento (WOLLINGER, 2020).

Uma última coisa a se notar na Figura 23 é que saem dois cabos do encapsulamento (*package*) da associação série, um mais grosso, com dois fios, na parte superior, e um mais

<sup>13</sup> "Este efeito consiste no fato de que a bateria 'memoriza' a carga adquirida no último processo de carga e não consegue ultrapassá-lo. Se a bateria, uma vez que seja, for carregada com apenas uma pequena parcela de sua carga total, nas cargas seguintes ela não consegue mais adquirir a carga completa por mais tempo que a deixemos no carregador", extraído do blog Instituto Newton C. Braga, acessado por <<https://bitly.com/8jc7B>> em 26/07/2020.

Figura 23: Bateria de LiPo comumente utilizada nos robôs de combate.



Fonte: Elaborado pelo autor.

fino, com três fios, na parte inferior. O mais grosso corresponde à saída da bateria, onde deve conectar-se a carga, por ter maior capacidade de corrente (fio com secção transversal maior) e apresentar a tensão resultante da associação série das duas células internas. O cabo da parte inferior é utilizado para recarga e balanceamento das células, que podem não se descarregar de maneira uniforme e serem recarregadas com cargas diferentes; isso gera uma descarga mais acentuada que pode danificar a célula e prejudicar o desempenho de bateria como um todo.

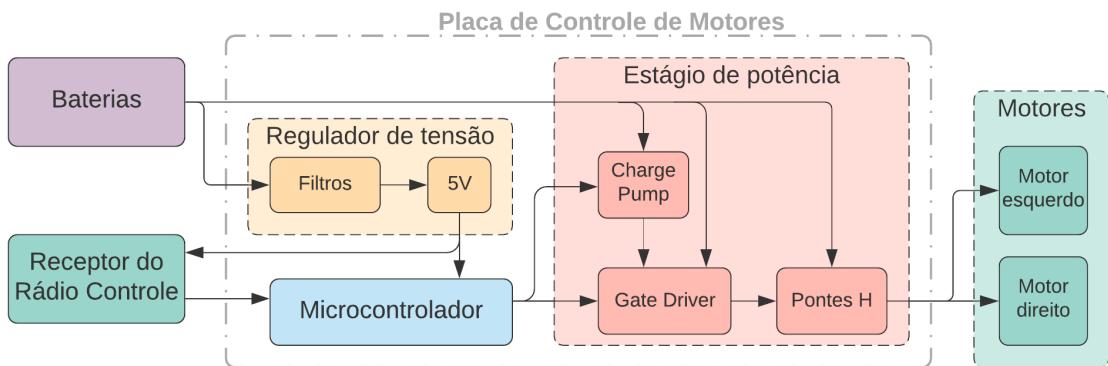


### 3 DESENVOLVIMENTO

Após a apresentação dos principais conceitos técnicos aos quais este projeto recorreu, este capítulo discorre sobre a implementação, com o apoio de tais técnicas e tecnologias, de um sistema de controle de motores CC, conforme descrito anteriormente no Objetivo. Faremos isso primeiramente nos aspectos de *hardware*, ou seja, de circuito eletrônico, e em seguida nos aspectos de *software*, isto é, da elaboração do programa do microcontrolador, seguido da integração de ambos e realização de alguns testes, levantando também seus principais modos de falha.

A fim de organizar melhor o desenvolvimento e a explicação do circuito, este foi dividido em módulos dispostos no diagrama em blocos da Figura 24, o qual indica também as interfaces da placa eletrônica com os dispositivos periféricos - baterias (em série), motores de locomoção e o receptor do rádio controle.

Figura 24: Diagrama em blocos do circuito.



Fonte: Elaborado pelo autor.

Pelo diagrama observa-se que o sinal do receptor do rádio controle é lido pelo microcontrolador, cuja função é interpretar sua informação e acionar o *Gate Driver*, que por sua vez chaveia os transistores de cada ponte H, os quais entregam a potência exigida em cada motor, conforme o comando inicial do rádio controle; além disso, o microcontrolador também provê a onda quadrada necessária no circuito de *Charge Pump*. Na parte de alimentação do circuito temos as baterias como fonte de energia, duas células de LiPo em série, cuja tensão é filtrada e regulada para proporcionar os 5 V de tensão da parte lógica do circuito; o módulo do *Charge Pump* utiliza a tensão da bateria como referência para aumentá-la ainda mais, por meio do multiplicador de tensão, e disponibilizá-la para o circuito de *Gate Driver*. Por fim, a tensão da bateria é aplicada ainda ao circuito do próprio *Gate Driver* e o das pontes H, que realiza a interface com os motores segundo a

modulação PWM.

A seguir discutiremos, uma a uma, a implementação em circuito de cada um dos módulos, seguido da integração final de todos conforme mostra o diagrama. Apesar de o desenvolvimento do *software* ser discutido após esta apresentação dos blocos de *hardware*, na prática ambos caminharam em paralelo, pois era importante saber se as propostas de um lado eram viáveis do outro, como a frequência de chaveamento do PWM das pontes H, por exemplo.

### 3.1 **Hardware**

O roteiro de apresentação dos tópicos segue uma certa lógica de dependência, ou seja, começaremos com o circuito periférico do microcontrolador e sua disposição de pinos de entrada e saída, depois partiremos para a geração da tensão de *Charge Pump* no respectivo módulo, que depende de um sinal PWM do microcontrolador, e que depois de implementado possibilita o dimensionamento do módulo de *Gate Driver*, o qual depende desses dois outros módulos citados, e assim por diante.

Reiterando, na prática o caminho de elaboração e projeto não foi tão linear quanto aqui descrito, diversos testes de compatibilidade foram feitos para conferir se um módulo seria compatível com outro de sua dependência. Contudo, uma vez atingido o objetivo de se ter um sistema funcionando, a apresentação mais clara e sucinta foi encontrada nesta forma citada e disposta a seguir.

#### 3.1.1 Microcontrolador

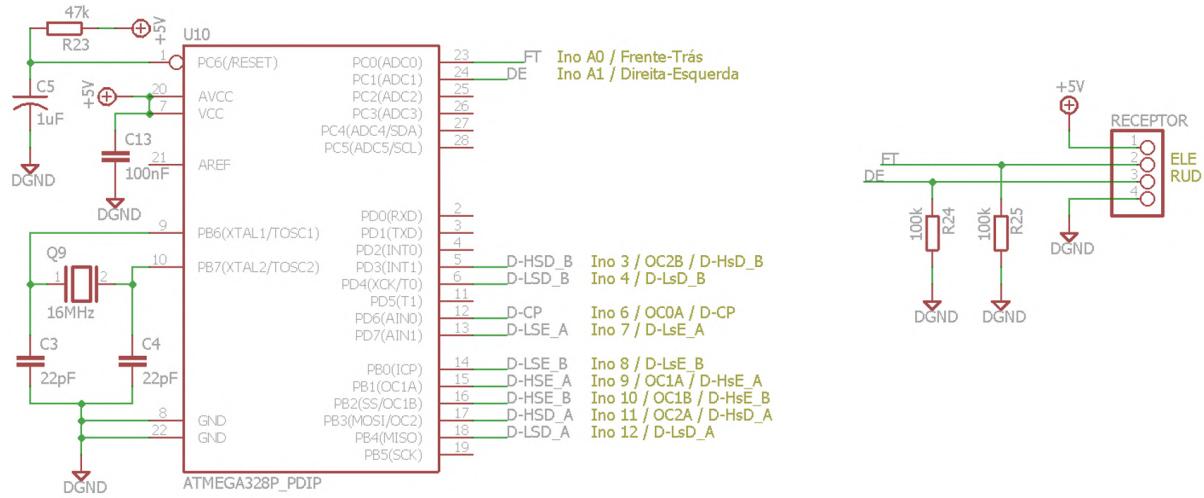
O primeiro módulo abordado é o do microcontrolador, exposto na Figura 25, que exibe os pinos do receptor do rádio controle, o ATmega328P e seu circuito periférico: ligação dos pinos de alimentação, oscilador a cristal piezoelétrico de 16 MHz, circuito RC de *power-on reset* e atribuição de alguns pinos às suas funções específicas.

Começando pelos pinos de alimentação, o ATmega328P permite uma alimentação máxima de 6 V, conforme indicado na Figura 26, mas utilizamos o valor típico de 5 V para esta parte lógica. O capacitor de 100 nF, posicionado próximo à alimentação, é um filtro contra ruídos e distúrbios, e no desenho da placa (PCB) deve ser posicionado o mais próximo possível do chip. Por fim, os terminais de GND são conectados ao *ground* digital.

O pino  $AV_{CC}$  corresponde à alimentação do conversor analógico-digital (ADC), e se este não for utilizado, como é o nosso caso, pode ser conectado diretamente junto ao  $V_{CC}$ , conforme o trecho extraído da página 5 do *datasheet* do ATmega328P, de 11/2015:

*"AV<sub>CC</sub> is the supply voltage pin for the A/D converter, PC3:0, and ADC7:6. It should be externally connected to V<sub>CC</sub>, even if the ADC is not used."*

Figura 25: Módulo do microcontrolador ATmega328P em conjunto com o receptor do rádio controle.



Fonte: Elaborado pelo autor.

Figura 26: Valores máximos permitidos na operação do ATmega328P.

30.1 Absolute Maximum Ratings*	
Operating Temperature .....	-55°C to +125°C
Storage Temperature .....	-65°C to +150°C
Voltage on any Pin except <u>RESET</u> with respect to Ground .....	-0.5V to V <sub>CC</sub> +0.5V
Voltage on <u>RESET</u> with respect to Ground .....	-0.5V to +13.0V
Maximum Operating Voltage .....	6.0V
DC Current per I/O Pin .....	40.0mA
DC Current V <sub>CC</sub> and GND Pins .....	200.0mA

Fonte: Extraído do *datasheet* do ATmega328P, de 11/2015, página 313.

Em seguida, há a implementação do *power-on reset* por meio do circuito RC (Resistor - Capacitor) no pino 1, cujo desenho do pino possui uma bolinha para indicar a lógica invertida de operação, isto é, barrada, acionada em nível lógico baixo. O *power-on reset*, também conhecido por *reset* de energização, consiste no acionamento do *reset* do microcontrolador durante um período inicial na energização do circuito, de modo que a sua operação só comece depois dos transientes iniciais, como a estabilização dos 5 V ou o funcionamento normal do receptor do rádio controle, por exemplo, mesmo que o chip já esteja energizado.

Para implementarmos o *power-on reset*, utilizamo-nos do circuito RC pelo fato de o capacitor estar descarregado antes da energização do circuito, o qual leva um tempo calculável para se carregar acima do valor de tensão máxima ainda considerado como nível lógico baixo pelo pino de *reset* do ATmega328P; além disso, depois de carregado ele não volta a se descarregar, a não ser pelo desligamento da tensão de alimentação, portanto sem risco de um acionamento indevido do *reset*. Apenas por curiosidade, em outras aplicações em que um botão de *reset* é disponibilizado para o usuário, ele geralmente é do tipo "normalmente aberto" (NA, ou NO, em inglês), e fica em paralelo com o capacitor deste circuito RC.

O valor de tensão máxima no pino de *reset* que ainda é considerado nível lógico baixo é de  $0,1V_{CC}$ , conforme está exibido na Tabela 8.

Tabela 8: Tensões correspondentes aos níveis lógicos do pino de *reset* do ATmega328P.

**Table 30-1. Common DC characteristics  $T_A = -40^\circ\text{C to } 105^\circ\text{C}$ ,  $V_{CC} = 1.8\text{V to } 5.5\text{V}$  (unless otherwise noted)**

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
$V_{IL2}$	Input Low Voltage, RESET pin	$V_{CC} = 1.8\text{V - }5.5\text{V}$	-0.5		$0.1V_{CC}^{(1)}$	V
$V_{IH2}$	Input High Voltage, RESET pin	$V_{CC} = 1.8\text{V - }5.5\text{V}$	$0.9V_{CC}^{(2)}$		$V_{CC} + 0.5$	V

Fonte: Adaptado do *datasheet* do ATmega328P, de 11/2015, página 313.

Com  $V_{CC} = 5\text{ V}$  temos  $V_{Low_{MAX}} = 0,1V_{CC} = 0,5\text{ V}$ . Por meio da Equação 3.1 de carga do capacitor, conseguimos calcular o período de *power-on reset*.

$$V_{capacitor}(t) = V_{CC} \cdot (1 - e^{-\frac{t}{R \cdot C}}) \quad (3.1)$$

O valor do resistor de *pull-up*, ou seja, do terminal de *reset* até o  $V_{CC}$ , deve seguir a recomendação dada na Tabela 9: entre  $30\text{ k}\Omega$  e  $60\text{ k}\Omega$ .

Tabela 9: Valores permitidos para o resistor de *pull-up* no pino de *reset* do ATmega328P.

**Table 30-1. Common DC characteristics  $T_A = -40^\circ\text{C to } 105^\circ\text{C}$ ,  $V_{CC} = 1.8\text{V to } 5.5\text{V}$  (unless otherwise noted) (Continued)**

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
$R_{RST}$	Reset Pull-up Resistor		30		60	$\text{k}\Omega$

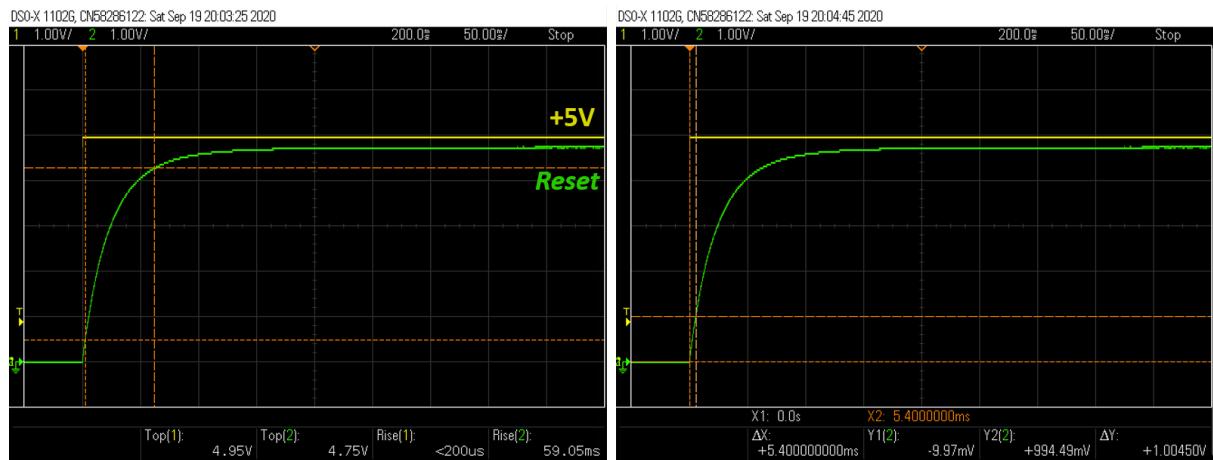
Fonte: Adaptado do *datasheet* do ATmega328P, de 11/2015, página 313.

Tomando  $R = 47\text{ k}\Omega$ ,  $V_{CC} = 5\text{ V}$  e  $V_{capacitor} = 0,5\text{ V}$ , e estipulando um período mínimo de  $t = 5\text{ ms}$  de *power-on reset*, a capacidade calculada resulta em  $C = 1\text{ }\mu\text{F}$ , conforme consta no esquemático apresentado. Portanto, o microcontrolador não começa a executar seu programa antes dos 5 ms iniciais de energização do circuito, pois nesse

período o ATmega328P seguramente lê a tensão de entrada no pino de *reset* como nível baixo. Considerando a informação de limiar de tensão mínima para nível alto, e a mesma equação de carga de capacitor, o tempo máximo de permanência do *power-on reset* é de aproximadamente 100 ms.

Na Figura 27 podemos ver o resultado desta implementação no circuito real do módulo, durante a energização do circuito, com a curva em amarelo do canal 1 exibindo a tensão de 5 V, que alimenta o microcontrolador, e a curva em verde do canal 2 apresentando a tensão sobre o capacitor, a qual é aplicada no pino de *reset*. No gráfico da direita observa-se a tensão atingindo 1 V no intervalo de aproximadamente 5 ms, conforme projetado.

Figura 27: Implementação do *power-on reset* do microcontrolador.



Fonte: Elaborado pelo autor.

Outro circuito presente neste módulo é o do oscilador externo, que consiste em um cristal piezoelétrico de 16 MHz e dois capacitores de 22 pF, dispostos e dimensionados conforme as indicações do *datasheet* expostas na Figura 28. Uma recomendação de desenho para a placa é que tais componentes devem ficar o mais próximos possíveis dos terminais do microcontrolador, e com trilhas curtas para suas ligações.

A medição deste sinal oscilante na implementação efetiva do circuito exige certos cuidados, pois qualquer efeito de carregamento do equipamento de medição é suficiente para distorcer tal sinal. A Figura 29 exibe uma tentativa de medição que foi bastante distorcida: nos canais 1 e 2 temos os sinais XTAL1 (pino 9) e XTAL2 (pino 10), e a função matemática em rosa é a subtração dos dois, a qual exibe o sinal oscilante de 16 MHz, da forma que foi possível capturá-lo.

Figura 28: Circuito periférico do cristal oscilador com capacitores do ATmega328P.

Figure 9-2. Crystal Oscillator Connections

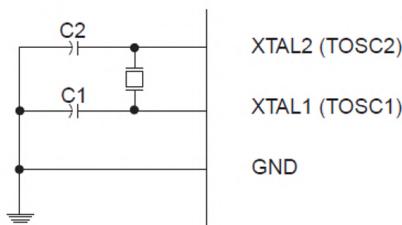
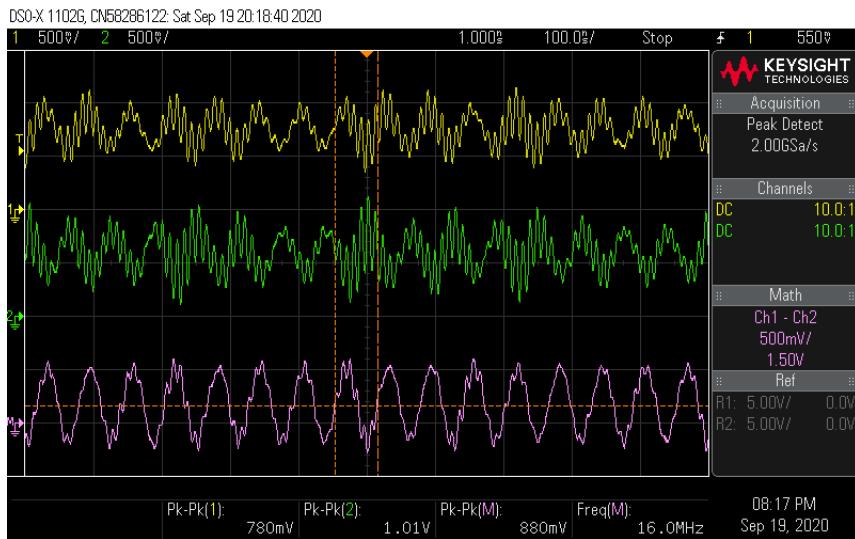


Table 9-3. Low Power Crystal Oscillator Operating Modes<sup>(3)</sup>

Frequency Range (MHz)	Recommended Range for Capacitors C1 and C2 (pF)	CKSEL3...1 <sup>(4)</sup>
0.4 - 0.9	–	100 <sup>(2)</sup>
0.9 - 3.0	12 - 22	101
3.0 - 8.0	12 - 22	110
8.0 - 16.0	12 - 22	111

Fonte: Adaptado do *datasheet* do ATmega328P, de 11/2015, páginas 29 e 30.

Figura 29: Medição do sinal do cristal oscilador de 16 MHz do microcontrolador.



Fonte: Elaborado pelo autor.

Por fim, existe a atribuição de pinos às funções do sistema, dispostas na lista a seguir, mas cuja explicação de função individual é melhor entendida com o contexto dos próximos módulos e a definição pelo *software*. Os terminais utilizados foram:

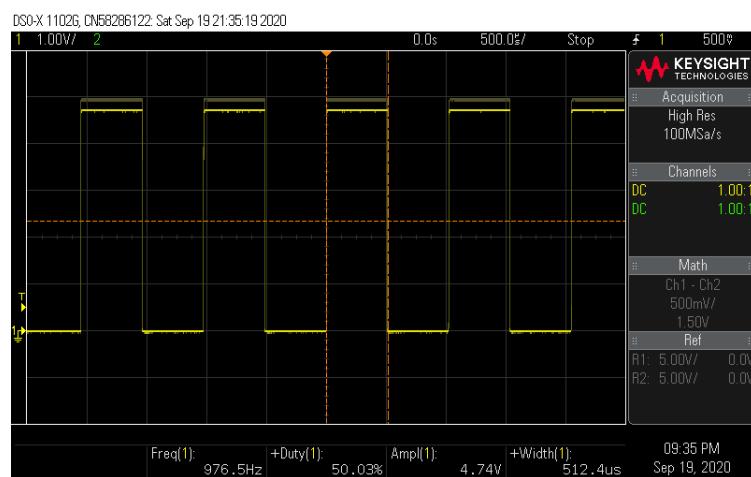
- 2 entradas para leitura dos canais do rádio controle, nos pinos PC0 e PC1 (terminais 23 e 24);
- 1 saída PWM para gerar a onda quadrada do *Charge pump*, no pino OC0A (terminal

12);

- 4 saídas PWM, um par para cada ponte H, para controlarem seus transistores *High-side*, nos pinos OC1A, OC1B, OC2A e OC2B (terminais 15, 16, 17 e 5);
- 4 saídas digitais, um par para cada ponte H, para controlarem seus transistores *Low-side*, nos pinos PD7, PB0, PB4 e PD4 (terminais 13, 14, 18 e 6).

Apenas a título de exemplificação do sinal de um dos pinos de saída do microcontrolador, a Figura 30 traz a medição da onda quadrada gerada para acionamento do *Charge pump*. Nela podemos observar uma curva sombreada, que representa a tensão do terminal do microcontrolador antes de ser ligado ao circuito do *Charge pump*, por volta de 0,2 V mais alta, o que evidencia o pequeno carregamento que tal circuito causa na saída do ATmega328P - nada preocupante para a aplicação. O sinal tem aproximadamente 1 kHz e  $\delta = 50\%$ , conforme é esperado no citado bloco posterior.

Figura 30: Onda quadrada de entrada do *Charge pump*.



Fonte: Elaborado pelo autor.

O outro elemento deste bloco é o receptor do rádio controle, alimentado pelos mesmos 5 V que o microcontrolador. De todas as suas saídas disponíveis, conforme mostrado no capítulo do Embasamento Teórico, utilizamo-nos de apenas duas, uma para cada direção do movimento do robô. Encaramos o receptor como uma "caixa preta", da qual sabe-se o comportamento esperado nos sinais de saída, mas não se tem conhecimento de seu funcionamento interno. Algumas características de tais sinais estão mostradas no gráfico da Figura 31, obtido pela medição do circuito real, em que o canal 1 do osciloscópio (amarelo) representa a saída ELE do receptor, responsável pelo movimento para frente e para trás, e o canal 2 (verde) representa a saída RUD, que comanda o movimento para a direita e para a esquerda.

Figura 31: Características dos pulsos PPM de saída do receptor do rádio controle.



Fonte: Elaborado pelo autor.

Na Figura 31 nota-se que a amplitude do sinal de saída é de aproximadamente 3,3 V, o que não é um problema para leitura do ATmega328P, o qual entende como nível alto valores de tensão a partir de  $0,6V_{CC} = 3$  V. Os resistores de  $100\text{ k}\Omega$  têm função de *pull-down*, ou seja, garantir que, na ausência da conexão do receptor ao circuito, a tensão em cada entrada não fique flutuando, podendo causar uma falsa leitura.

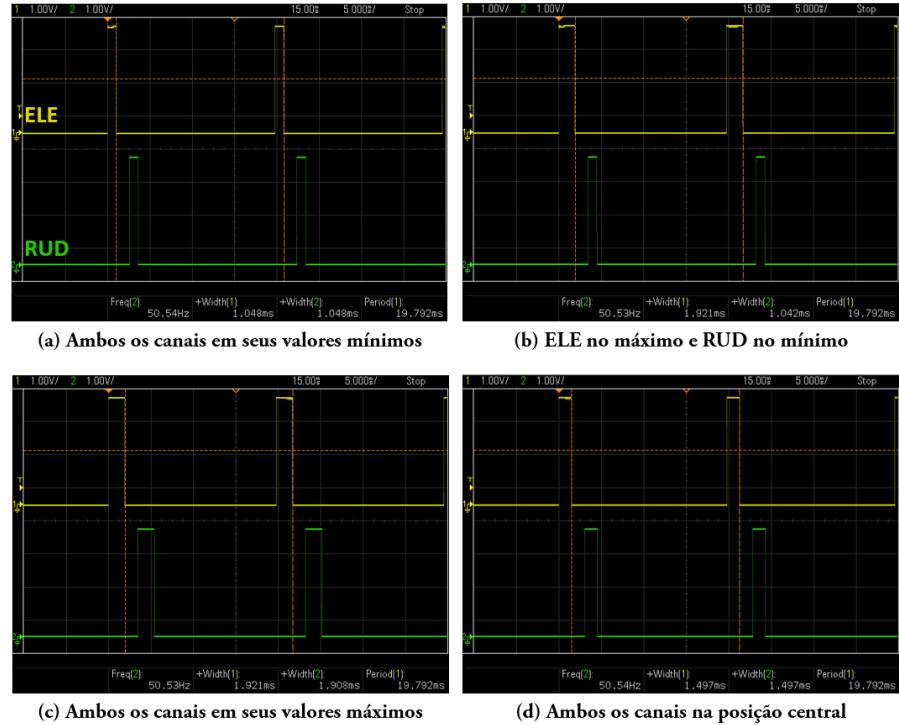
Quanto à excursão de cada canal de saída do receptor, isto é, a faixa de larguras de pulso possível, a Figura 32 exibe quatro medições feitas no circuito conforme a manipulação da alavanca do rádio controle, com combinações de posições mínimas e máximas, e também a posição central, que deve ser a de comando para os motores permanecerem sem acionamento (na prática, para deixar o robô parado).

### 3.1.2 Charge Pump

Como já explicado no capítulo do embasamento teórico, uma das técnicas para se chavear os transistores *High-side* da ponte H envolve a aplicação de uma tensão superior à da alimentação dela, que neste caso é a tensão proveniente das baterias. Essa tensão mais elevada é gerada pelo circuito de *Charge Pump*, exibido na Figura 33.

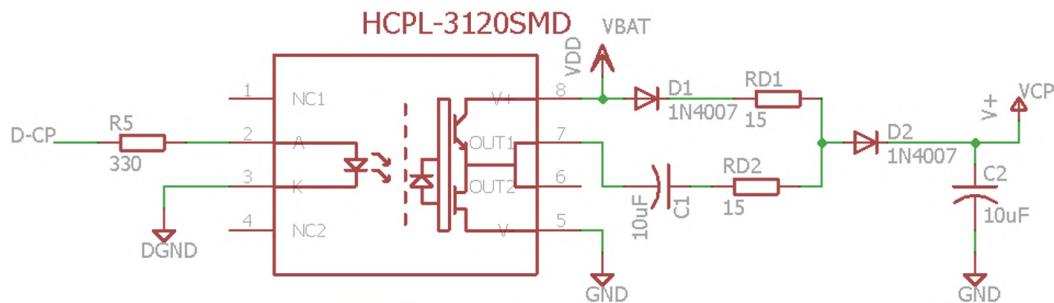
O opto-acoplador utilizado foi brevemente exibido na Figura 13, a qual contém o nome de seus terminais. O comando da operação é feito pelos pinos à esquerda, de anodo e catodo do LED interno, e a respectiva resposta ocorre no terminal  $V_o$ : quando o LED é acionado/aceso,  $V_o$  conecta-se a  $V_{CC}$ , e quando ele fica apagado,  $V_o$  conecta-se a  $V_{EE}$ . O HCPL-3120 ainda recomenda que  $15\text{ V} \leq (V_{CC} - V_{EE}) \leq 30\text{ V}$ , conforme a Tabela 10, o que não é um problema aqui, pois a associação série das duas baterias LiPo, de duas células cada, resulta em 16,8 V quando completamente carregadas, e 14,8 V quando descarregadas, apenas um pouco abaixo do valor recomendado. Após alguns experimentos variando a tensão  $V_{CC} - V_{EE}$ , observou-se que o chaveamento ocorreu sem deformações

Figura 32: Comportamento da modulação PPM nos canais ELE e RUD, lidos pelo microcontrolador.



Fonte: Elaborado pelo autor.

Figura 33: Módulo do *Charge Pump*.



Fonte: Elaborado pelo autor.

até próximo dos 12 V, mas abaixo disso foi notada a atenuação do sinal de saída, conforme a página 2 do *datasheet* também aponta.

O comando, como dito anteriormente, é uma onda quadrada de 1 kHz vinda do microcontrolador, de 5 V de amplitude com referência ao *ground* digital (DGND). O resistor R5 serve como limitador de corrente, e seu dimensionamento foi feito baseado nas informações do *datasheet* do componente, indicadas na Tabela 10, a qual diz que a corrente do LED deve estar entre 7 mA e 16 mA, e na Tabela 11, que mostra uma queda

Tabela 10: Condições de operação recomendadas para o HCPL-3120.

**Recommended Operating Conditions**

Parameter	Symbol	Min.	Max.	Units
Power Supply Voltage	$(V_{CC} - V_{EE})$	15	30	Volts
Input Current (ON)	$I_F(ON)$	7	16	mA
		10		
Input Voltage (OFF)	$V_{F(OFF)}$	-3.6	0.8	V
Operating Temperature	$T_A$	-40	100	°C

Fonte: Extraído do *datasheet* do HCPL-3120, de 21/03/2016, página 13.

de tensão típica de 1,5 V no LED, mas que pode variar entre 1,2 V e 1,8 V.

Tabela 11: Faixa de valores para a tensão de condução direta do LED de controle do opto-acoplador HCPL-3120.

Parameter	Symbol	Device	Min.	Typ. <sup>a</sup>	Max.	Units	Test Conditions	Fig.	Note
Input Forward Voltage	$V_F$	HCPL-3120	1.2	1.5	1.8	V	$I_F = 10 \text{ mA}$	16	
		HCPL-J312 HCNW3120		1.6	1.95				

Fonte: Adaptado do *datasheet* do HCPL-3120, de 21/03/2016, página 14.

Utilizando  $R_5 = 330 \Omega$ , e considerando 5 V de amplitude da onda quadrada do microcontrolador, para o caso de 1,2 V de tensão de condução direta no LED, este consumiria 11,5 mA, e para o caso de 1,8 V a corrente seria de 9,7 mA; em ambos os casos, a corrente fica dentro da margem recomendada para o controle do opto-acoplador.

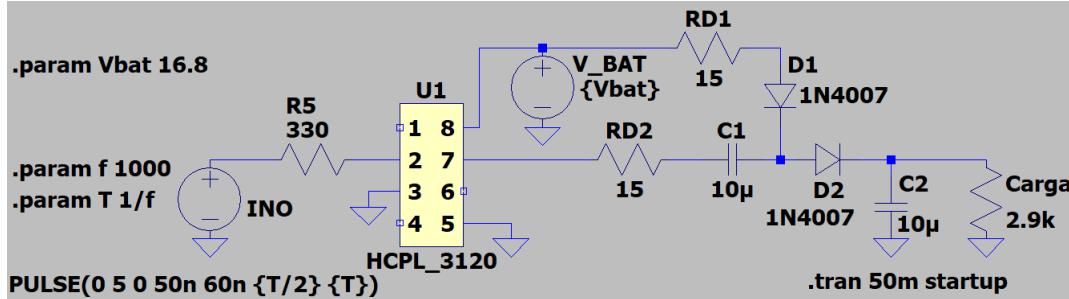
Na saída temos o circuito grampeador de tensão, ou *clamper*, ou ainda, neste caso, multiplicador de tensão, dado que a amplitude da onda quadrada de saída é a mesma da bateria. O conceito de funcionamento foi brevemente abordado no capítulo do embasamento teórico, e aqui nos aprofundaremos mais no dimensionamento de cada componente.

No momento da energização do circuito, tanto C1 quanto C2 estão descarregados, e para limitar a corrente inicial de carga, também chamada de "*inrush current*", os resistores RD1 e RD2 são colocados. Os diodos D1 e D2 são da família 1N400x ( $x = 1,2,3\dots7$ ), que possuem a mesma capacidade de condução de corrente, 1 A contínuo e 30 A de pico não repetitivo, mas diferem na tensão reversa máxima suportada, que aumenta quanto maior o número final ( $x$ ).

A fim de verificar o funcionamento do circuito e seu dimensionamento correto, foi realizada a simulação no LTspice, cujo circuito utiliza os modelos disponibilizados para os

diodos e para o opto-acoplador<sup>1</sup>. O circuito da simulação está apresentado na Figura 34, e sua única diferença para o módulo real é a presença de um resistor de carga na saída, para representar o consumo que a tensão de *Charge pump* deve estar preparada para suprir.

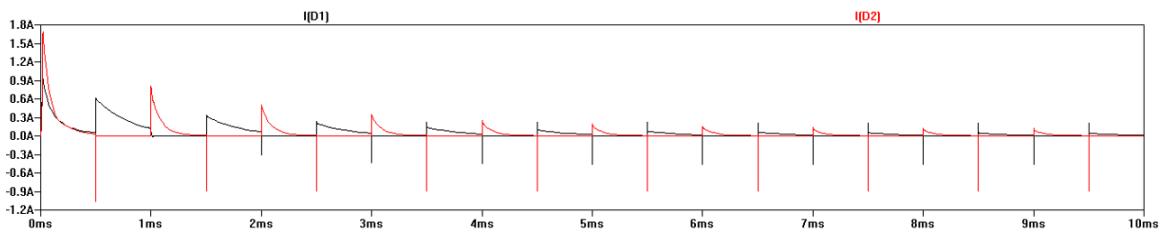
Figura 34: Circuito para simulação do *Charge pump* no LTspice.



Fonte: Elaborado pelo autor.

Executando a simulação e medindo a corrente que passa pelos diodos nos primeiros 10 ms, obtemos os gráficos exibidos na Figura 35; nela observa-se um pico inicial em ambos D1 e D2, o qual não ultrapassa 1,8 A e, portanto, não representa risco para a família 1N400x, que permite até 30 A de pico não-repetitivo. Após esse período inicial de 10 ms de energização do circuito, a corrente fica abaixo de 300 mA mesmo durante os picos, os quais ocorrem brevemente em cada novo ciclo da onda quadrada de comando. Outro comportamento observável é a ocorrência de picos negativos de corrente, como se o diodo conduzisse reversamente, porém não é exatamente isso o que acontece, e sim a reação da capacitância parasita entre anodo e catodo em função da aplicação de tensão reversa; essa capacitância tem por volta de 5 pF, e tal pico de corrente "reversa" tem aproximadamente 150 ns.

Figura 35: Simulação da corrente pelos diodos do circuito de *Charge pump*.



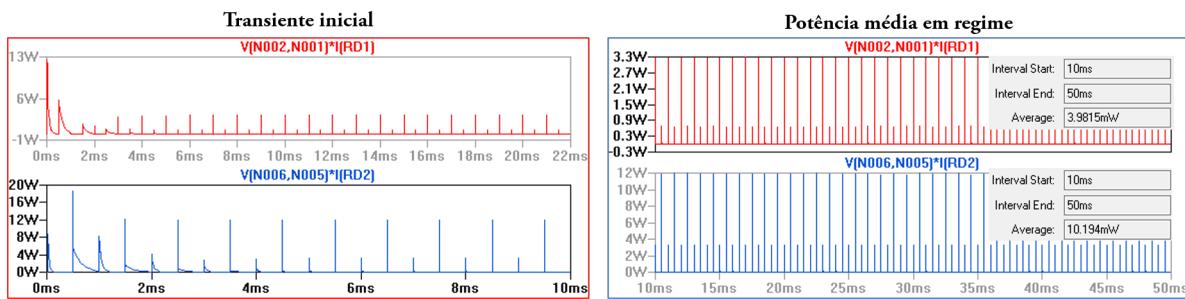
Fonte: Elaborado pelo autor.

Uma vez garantida a operação segura dos diodos, partimos para a análise dos resistores RD1 e RD2, que também conduzem a corrente dos diodos. A Figura 36 mostra os

<sup>1</sup> Disponível em <[https://docs.broadcom.com/docs/AvagoHCPL3120\\_MOD.txt](https://docs.broadcom.com/docs/AvagoHCPL3120_MOD.txt)>, acessado em 23/08/2020.

gráficos de potência instantânea, tanto durante o transiente de energização, quanto depois da acomodação para operação em regime. A conclusão da análise é que, mesmo com picos de potência instantânea bastante elevados, a potência média responsável pelo aquecimento do componente é baixa, sendo de 4 mW para RD1 e 10 mW para RD2, aproximadamente, conforme mostram as janelas de medição na própria Figura 36.

Figura 36: Simulação da dissipação de potência nos resistores de limitação de corrente do circuito de *Charge Pump*.



Fonte: Elaborado pelo autor.

Legenda: RD1 em vermelho, RD2 em azul.

Por fim, devemos analisar o dimensionamento dos capacitores. Pela Tabela 12, sabe-se que, mesmo com a saída em aberto, o chip do opto-acoplador consome tipicamente 2,5 mA de  $V_{CC}$ , mas pode puxar até 5 mA. O módulo do *Gate Driver* será abordado em seguida, e nele veremos a tensão de *Charge pump* alimentando quatro de seus oito opto-acopladores; disto, podemos calcular um consumo típico de  $4 \times 2,5 \text{ mA} = 10 \text{ mA}$  da tensão  $V_{CP}$ , podendo chegar a até  $4 \times 5 \text{ mA} = 20 \text{ mA}$ .

Tabela 12: Queda de tensão de saída e consumo do HCPL-3120.

Parameter	Symbol	Device	Min.	Typ. <sup>a</sup>	Max.	Units	Test Conditions	Fig.	Note
High Level Output Voltage	$V_{OH}$		$(V_{CC} - 4)$	$(V_{CC} - 3)$		V	$I_O = -100 \text{ mA}$	1, 3, 19	d, e
Low Level Output Voltage	$V_{OL}$			0.1	0.5	V	$I_O = 100 \text{ mA}$	4, 6, 20	
High Level Supply Current	$I_{CCH}$			2.5	5.0	mA	Output Open, $I_F = 7 \text{ to } 16 \text{ mA}$	7, 8	
Low Level Supply Current	$I_{CCL}$			2.5	5.0	mA	Output Open, $V_F = -3.0 \text{ to } +0.8 \text{ V}$		

Fonte: Adaptado do *datasheet* do HCPL-3120, de 21/03/2016, página 14.

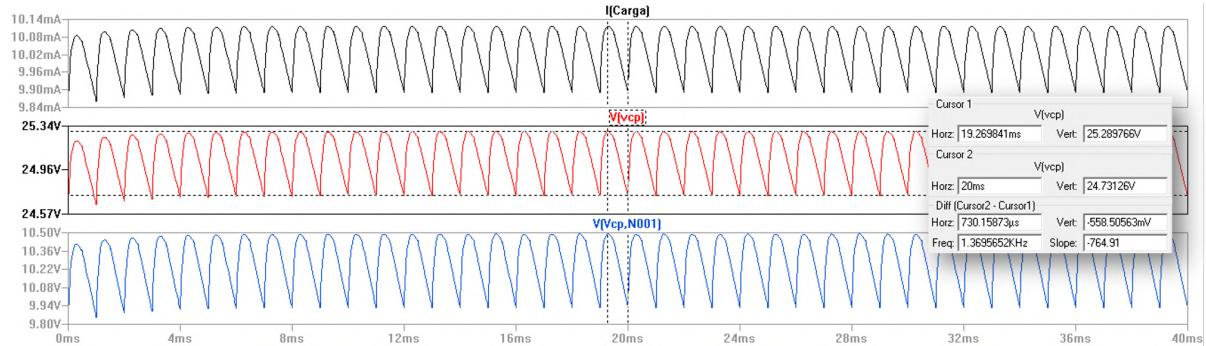
A Tabela 12 ainda contém outra informação relevante quanto às perdas do HCPL-3120 no que se refere à queda de tensão interna quando acionado, isto é, sua tensão  $V_{CC}$  pode ficar até 4 V menor quando disponibilizada da saída  $V_o$ , sendo a perda típica de 3 V nessa ligação interna.

A tensão  $V_{CP}$  depende da carga do capacitor C2, e de como o capacitor C1 o recarrega em cada novo ciclo da onda quadrada de comando (vinda do microcontrolador). Desse modo, um circuito drenando a carga de C2 causa uma queda da tensão de saída, que só volta a subir com um novo ciclo de carga por C1; esta oscilação de tensão é comumente chamada de *ripple*, a qual não pode ser muito significativa, senão pode prejudicar o chaveamento dos transistores *High-side*.

A fim de simular o *ripple* da tensão  $V_{CP}$ , dois cenários são propostos: um leva em conta o consumo típico do HCPL-3120 de 2,5 mA, e outro considera o pior caso, com consumo de 5 mA por opto-acoplador. Em ambos os casos, usaremos a tensão de bateria descarregada,  $V_{BAT} = 14,8$  V, e observaremos o comportamento em regime, após os 10 ms de transientes iniciais de energização.

No caso típico, o resistor que representa a carga tem valor  $R_{Carga} = 2,5$  k $\Omega$ , equivalente aos 10mA drenados, e o comportamento da saída é mostrado na Figura 37.

Figura 37: Simulação do comportamento de  $V_{CP}$  no caso de consumo típico do HCPL-3120 nos quatro chips do *Gate driver*, e bateria descarregada ( $V_{BAT} = 14,8$  V).



Fonte: Elaborado pelo autor.

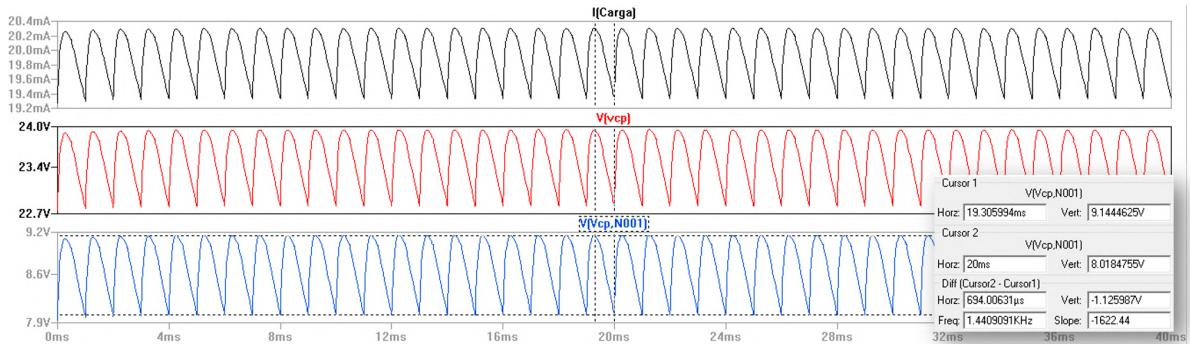
Legenda: O primeiro gráfico, em preto, mostra a corrente da carga por volta de 10 mA; o segundo, em vermelho, mostra o *ripple* de 0,55 V da tensão  $V_{CP}$ ; o terceiro é a diferença entre  $V_{CP}$  e  $V_{BAT}$ .

Das curvas apresentadas na Figura 37, a última é a que nos interessa mais, pois mostra a tensão  $V_{CP}$  pelo menos 9,9 V acima de  $V_{BAT}$ , mesmo nos pontos mais baixos. Contando com outros 3 V de queda de tensão interna no opto-acoplador do *Gate Driver*, ainda restam 6,9 V para acionar o MOSFET, cuja tensão  $V_{th_{MAX}} = 4$  V, portanto sem problemas para o chaveamento.

O outro cenário é o do consumo máximo de 5 mA por opto-acoplador; neste caso, a carga tem o valor de  $R_{carga} = 1,18$  k $\Omega$ , e o comportamento da saída está mostrado na Figura 38.

Observando novamente a última curva, em azul, vemos que seu ponto mais baixo

Figura 38: Simulação da tensão  $V_{CP}$  no caso de consumo máximo do HCPL-3120 nos quatro chips do *Gate driver*, e bateria descarregada ( $V_{BAT} = 14,8$  V).



Fonte: Elaborado pelo autor.

Legenda: O primeiro gráfico, em preto, exibe a corrente da carga por volta de 20 mA; o segundo, em vermelho, mostra a tensão  $V_{CP}$ ; o terceiro é a diferença entre  $V_{CP}$  e  $V_{BAT}$ , com um *ripple* de 1,1 V.

é de 8 V, e descontando-se 3 V de queda de tensão interna ainda restam 5 V para o acionamento do MOSFET, mais que os 4 V de  $V_{th}$  máximo. Sendo assim, o pior caso, que combina a bateria descarregada e o consumo máximo dos quatro opto-acopladores, não representa risco para o chaveamento dos transistores *High-side*.

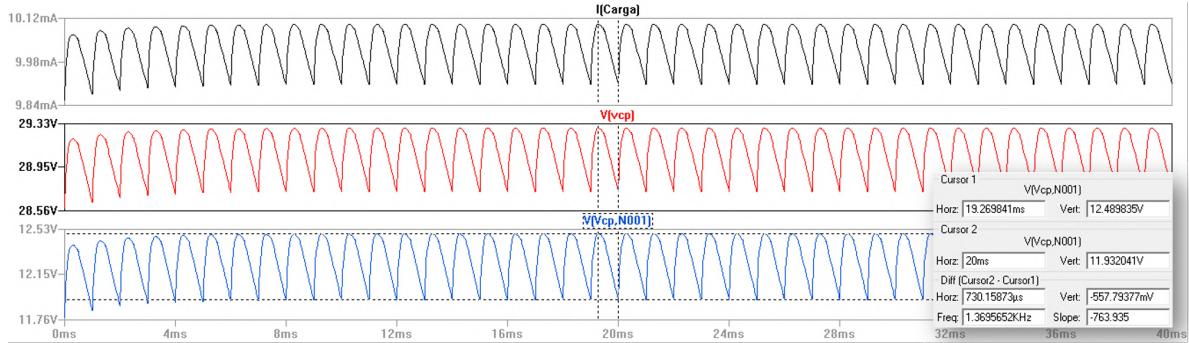
Um último cenário interessante para abordarmos é aquele que tem a bateria completamente carregada e o consumo típico dos componentes, a fim de verificarmos a amplitude do *ripple* de saída e o valor máximo da tensão de *Charge pump*. Neste cenário,  $V_{BAT} = 16,8$  V e  $R_{carga} = 2,9 \text{ k}\Omega$ , e os gráficos resultantes estão na Figura 39.

Neste último cenário, observamos que o valor máximo de  $V_{CP} = 29,3$  V, e sabendo que ele será aplicado ao  $V_{CC}$  do HCPL-3120, que recomenda  $(V_{CC} - V_{EE}) \leq 30$  V e um valor máximo de 35 V, confirma-se que não teremos problemas com a geração de uma tensão acima disso.

Uma vez completada a simulação do circuito e o projeto do dimensionamento correto dos seus componentes, passamos à verificação prática da implementação por meio de medições do circuito. Começando pelo primeiro cenário em que não há carga nenhuma conectada à tensão  $V_{CP}$ , a fim de verificar o potencial máximo do circuito em cada extremo da tensão de bateria, estão as duas medições na Figura 40. A partir delas extrai-se que os valores máximos para bateria carregada e descarregada são respectivamente 30,6 V e 26,6 V, portanto dentro do limite de 35 V para alimentação do HCPL-3120.

Pela Figura 40 também pode-se medir o tempo de até 10 ms para que a tensão  $V_{CP}$  entre em regime e mantenha um valor estável, conforme esperamos na estabilização

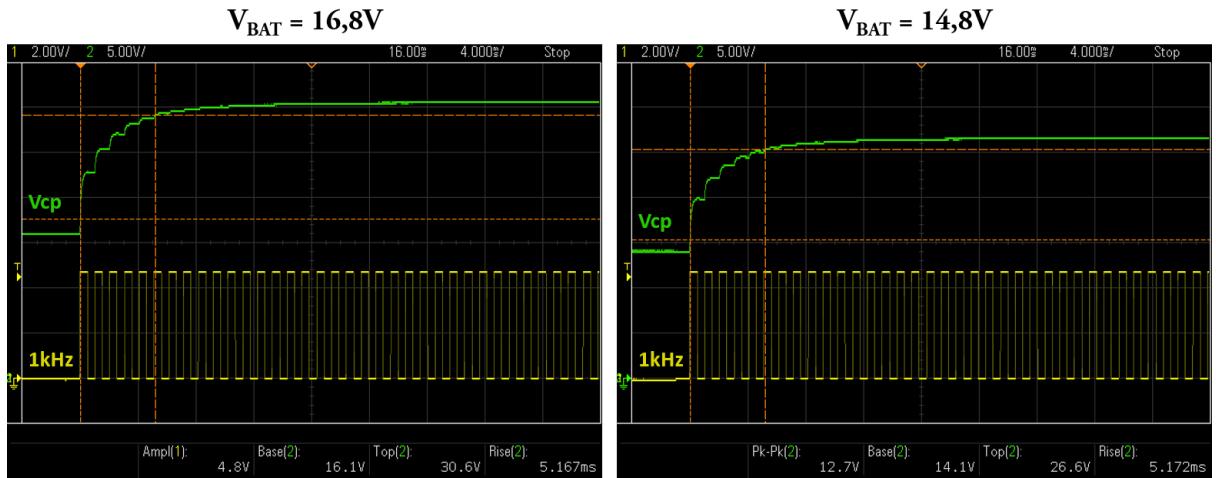
Figura 39: Simulação do comportamento de  $V_{CP}$  no caso de consumo típico do HCPL-3120 nos quatro chips do *Gate driver*, e bateria carregada ( $V_{BAT} = 16,8$  V).



Fonte: Elaborado pelo autor.

Legenda: O primeiro gráfico, em preto, exibe a corrente da carga por volta de 10 mA; o segundo, em vermelho, mostra a tensão  $V_{CP}$ , com um valor máximo de 29,3 V; o terceiro é a diferença entre  $V_{CP}$  e  $V_{BAT}$ , com um *ripple* de 0,56 V.

Figura 40: Medição da energização do *Charge pump* sem carga conectada.



Fonte: Elaborado pelo autor.

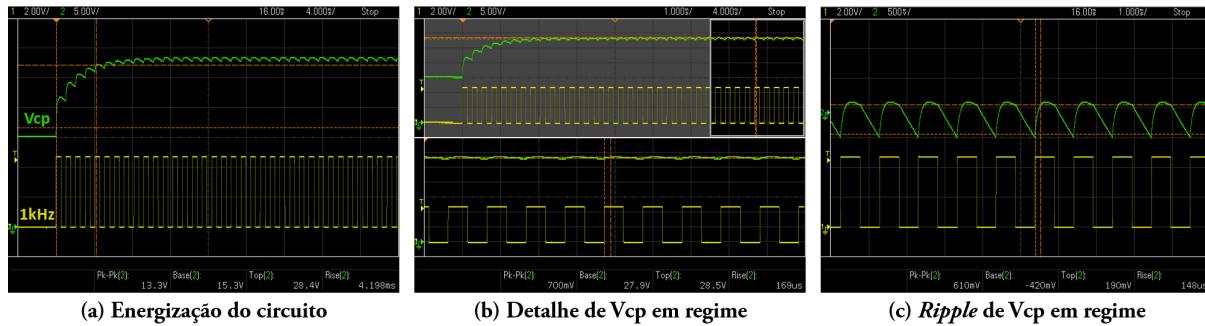
da simulação, portanto coerente com a mesma.

Nas Figuras 41 e 42 constam as medições de transiente e estabilização da tensão de *Charge pump*, para os casos de sua ligação ao módulo do *Gate driver* (e alimentação dos seus quatro opto-acopladores) nos extremos de carga da bateria. Em ambos notam-se pequenos desvios em relação à simulação: no caso de  $V_{BAT} = 16,8$  V a simulação indica um máximo de 29,3 V, porém a medição do circuito real mostra o topo da curva no valor de 28,5 V, 0,8 V abaixo; no caso de  $V_{BAT} = 14,8$  V, da mesma forma, a tensão mínima

esperada era 24,7 V e na realidade ela chegou a 24 V, 0,7 V abaixo. Estes pequenos desvios não geram preocupação quanto à aplicação pois o valor absoluto de  $V_{CP}$  continua sendo alto o suficiente em ambos os casos.

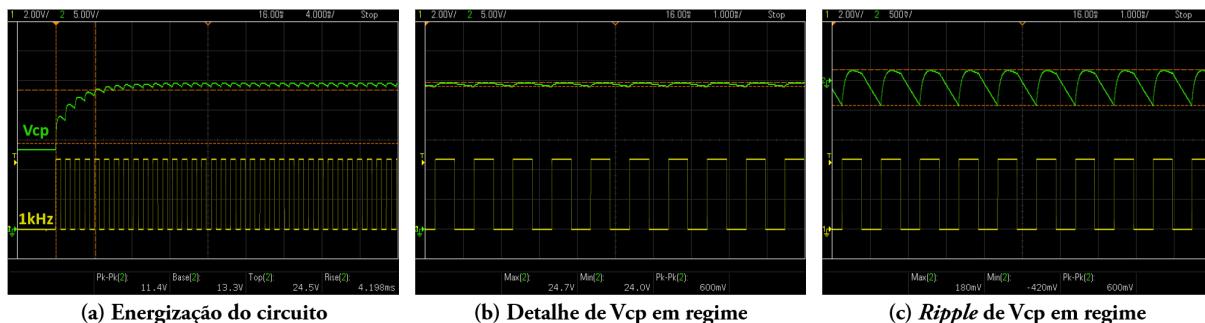
Algo interessante para se notar é que o *ripple* observado na simulação é praticamente o mesmo da situação real, não só em forma como também em amplitude.

Figura 41: Medição da energização do *Charge pump* com o *Gate driver* conectado e bateria carregada ( $V_{BAT} = 16,8$  V).



Fonte: Elaborado pelo autor.

Figura 42: Medição da energização do *Charge pump* com o *Gate driver* conectado e bateria descarregada ( $V_{BAT} = 14,8$  V).

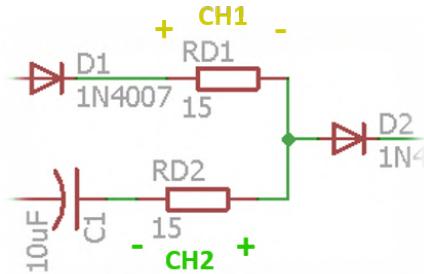


Fonte: Elaborado pelo autor.

Outra medição pertinente a ser apresentada é a da corrente pelos resistores RD1 e RD2, feita tanto durante a energização do circuito, quanto com o *Charge pump* em regime. A medição de corrente é feita indiretamente pelo osciloscópio, que mede o comportamento da tensão sobre o resistor de valor conhecido, possibilitando o cálculo da corrente. A posição das pontas de prova dos dois canais do instrumento está mostrada no detalhe do circuito na Figura 43.

Utilizando a bateria carregada e uma chave geral, realizamos a medição da Figura 44, que exibe à esquerda o comportamento geral durante a energização do circuito, e à

Figura 43: Posição dos canais do osciloscópio na medição da tensão dos resistores do *Charge pump*.



Fonte: Elaborado pelo autor.

direita a ampliação do período que corresponde à corrente máxima pelo resistor RD1. É possível notar no detalhe uma oscilação da tensão, que é resultante da trepidação da chave geral da bateria ao ser fechada, natural por ser um componente mecânico - efeito conhecido como *bouncing*.

Figura 44: Medição da tensão nos resistores do *Charge pump* durante sua energização e pico de corrente.

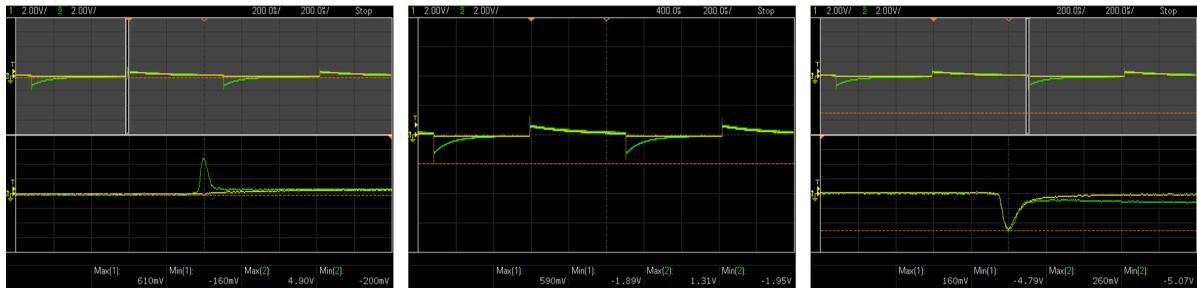


Fonte: Elaborado pelo autor.

O pico da tensão sobre o resistor RD1 de  $15 \Omega$  é de 18 V, portanto a corrente que passa por ele naquele instante é de 1,2 A, valor que se aproxima do observado na simulação, de aproximadamente 1,7 A. Quanto ao RD2, sua tensão máxima é de 4 V nesta etapa de energização, portanto conduz uma corrente de apenas 0,27 A.

Passando para a medição das mesmas formas de onda, agora com o circuito em regime normal de operação, capturamos os seguintes resultados apresentados na Figura 45. Na parte central temos o comportamento geral da forma de onda, e nas laterais estão apresentados os detalhes dos transientes.

Figura 45: Medição da tensão nos resistores do *Charge pump* durante operação normal.



Fonte: Elaborado pelo autor.

Os picos de corrente não ultrapassam os 350 mA, e duram menos de 200 ns, conforme pode ser observado nos transientes ampliados na Figura 45, portanto sem riscos para o dimensionamento dos componentes.

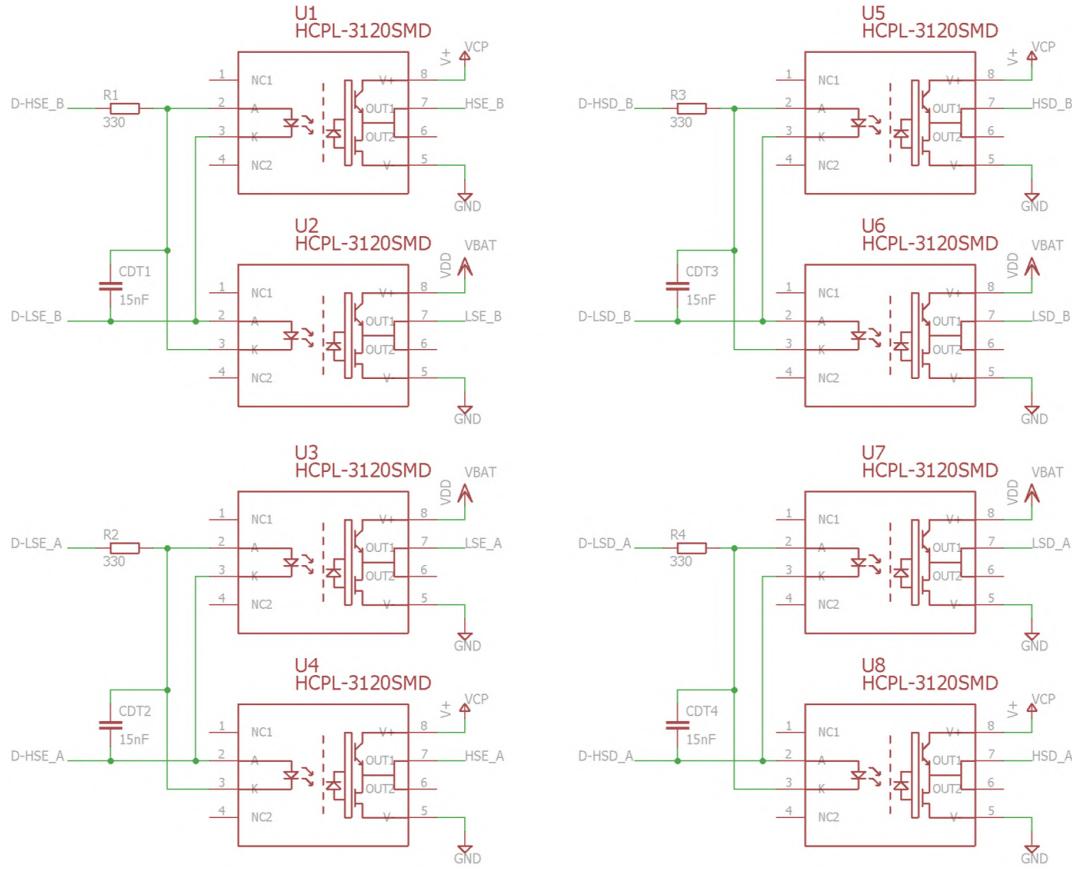
### 3.1.3 Gate Driver

O módulo de *Gate Driver* é a interface entre o controle de baixa potência, proveniente do microcontrolador, e as pontes H de saída para os motores de locomoção. É ele o responsável por chavear os transistores com as tensões adequadas, prover seus picos de corrente de *gate* e, neste caso, impedir via circuito que o par de transistores de um mesmo lado da ponte H seja acionado simultaneamente, o chamado "*shoot through*". O módulo completo está apresentado na Figura 46.

A primeira coluna de opto-acopladores corresponde à ponte H do motor esquerdo, e a segunda corresponde à ponte H do motor direito. Seus circuitos são idênticos, a não ser pela atribuição dos pinos de comando e de saída. Além disso, cada par de opto-acopladores (U1 e U2, U3 e U4,...) corresponde ao controle dos transistores de um mesmo lado da ponte H: um deles com o  $V_{CC}$  do HCPL-3120 conectado à tensão  $V_{CP}$ , correspondente ao acionamento do transistor *High-side*, e o outro recebendo alimentação de  $V_{BAT}$  para acionamento do MOSFET *Low-side*, ambos em referência ao GND (pinagem do HCPL disponível na Figura 13).

Agora que passamos pelas saídas do *Gate Driver*, voltemos a atenção ao circuito de comando de cada par de opto-acopladores. Os LED's internos de acionamento de cada componente estão ligados em antiparalelo, de modo que apenas um deles podem ficar aceso por vez, impedindo - via circuito - que os dois transistores de um mesmo ramo sejam acionados ao mesmo tempo. Para que um LED acenda e seu respectivo opto-acoplador seja acionado, portanto, um pino do microcontrolador deve estar em nível alto e o outro em nível baixo; caso ocorra o acionamento simultâneo por problemas no código (*software*), a diferença de potencial volta a ser zero e ambos os LED's internos apagam.

Figura 46: Módulo de *Gate Driver*.



Fonte: Elaborado pelo autor.

Apenas um resistor é necessário para limitar a corrente dos dois LED's pelo fato de eles estarem em antiparalelo, e seu dimensionamento segue os mesmos parâmetros já apresentados no módulo do *Charge pump*.

Os capacitores indicados por  $C_{DT}$  são os responsáveis por garantir o "dead-time" do *Gate Driver*, que consiste em um período em que nenhum dos dois opto-acopladores está acionado. Tal período é importante para garantir que um transistor de um mesmo lado da ponte H esteja completamente em corte antes que o outro entre em condução. Os diodos em antiparalelo evitam o acionamento dos transistores simultaneamente, porém a sobreposição de condução por conta dos tempos de chaveamento do MOSFET não são evitados por este método, e sim pela inserção de um capacitor que deve ser completamente descarregado e depois carregado para possibilitar a alternância de acionamento dos opto-acopladores.

O dimensionamento do capacitor de *dead-time*,  $C_{DT}$ , pode ser estimado pela Equação 3.2 (COLTON, 2009), e leva em conta a tensão de saída do microcontrolador,  $V_{comando} = 5$  V, resistência para limitação de corrente,  $R_S = 330 \Omega$ , a tensão de condução direta do LED, tipicamente  $V_{LED} = 1,5$  V, e o período de atraso que se deseja entre o fim

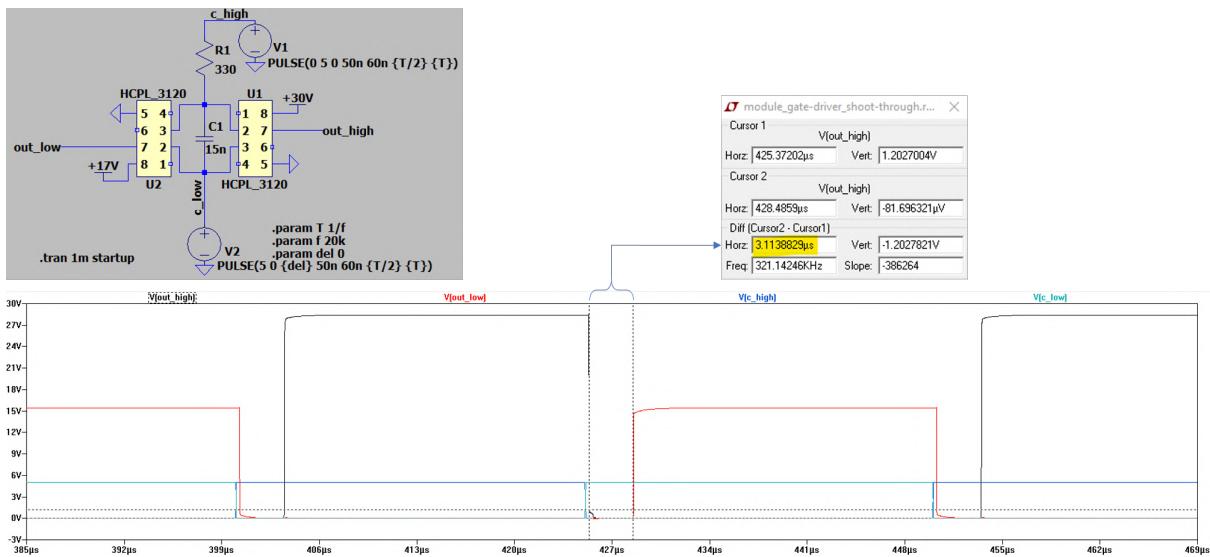
do acionamento de um opto-acoplador e o início do próximo,  $t_{atraso}$ .

$$C_{DT} = \frac{t_{atraso}}{R_S} \cdot \frac{V_{comando}}{2 \cdot V_{LED}} \quad (3.2)$$

É importante que o atraso seja algumas vezes superior aos tempos de chaveamento do transistor: para o MOSFET escolhido, o IRF1405, foram medidos os tempos  $t_{subida} = 121,5$  ns e  $t_{descida} = 51$  ns para sua tensão  $V_{DS}$ . Sendo assim, o atraso determinado foi de  $t_{atraso} = 3 \mu s$ , resultando em um capacitor  $C_{DT} \approx 15$  nF, segundo cálculo pela Equação 3.2 e conforme consta no circuito do módulo.

A fim de comprovar o funcionamento do *dead-time* e o correto dimensionamento do respectivo capacitor, uma simulação em LTspice foi realizada utilizando o modelo do HCPL-3120, e está exibida na Figura 47, juntamente com a ampliação de um período de chaveamento dos opto-acopladores, e a janela com os valores dos cursores de medição do intervalo.

Figura 47: Simulação do *dead-time* do *Gate Driver*.



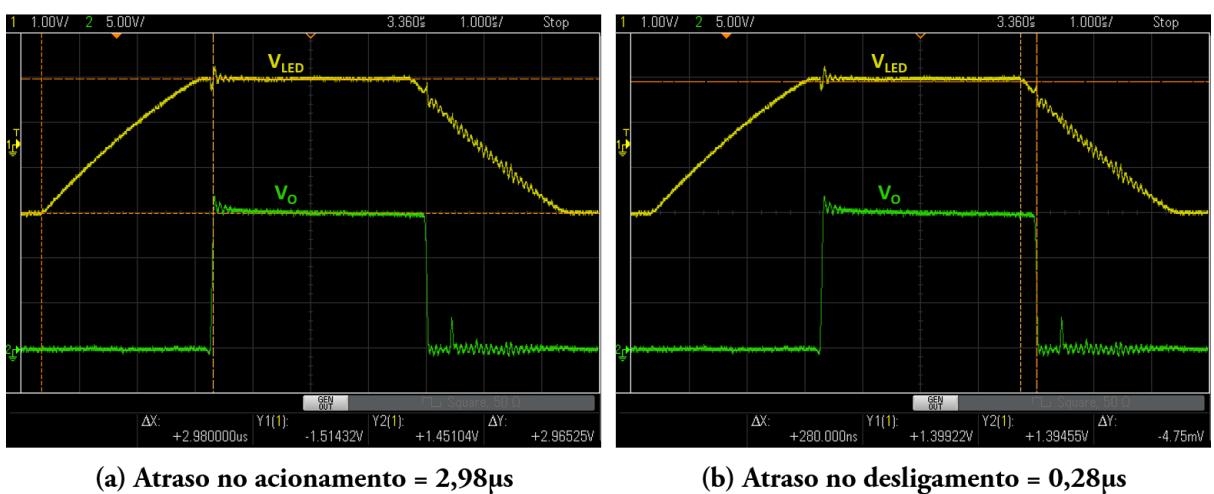
Fonte: Elaborado pelo autor.

Conforme projetado, apesar de os comandos de entrada inverterem-se quase que instantaneamente, como pode ser observado nas curvas  $V(c\_high)$  e  $V(c\_low)$ , as saídas têm um atraso de aproximadamente  $3 \mu s$  entre o fim de um acionamento e o início de outro, de acordo com o apontado pelos cursores entre a borda de descida da saída  $V(out\_high)$ , e a borda de subida da saída  $V(out\_low)$ .

Na implementação prática do circuito, as medições da Figura 48 foram realizadas a fim de verificar em *protoboard* o funcionamento do projeto do *dead-time* do módulo, e o resultado foi satisfatório. No canal 1 (em amarelo) temos a entrada de um opto-acoplador,

portanto a tensão sobre seu LED interno, e no canal 2 (em verde) temos sua respectiva saída, portanto a tensão em  $V_o$ . Na medição da esquerda (a) podemos observar a tensão saindo de -1,5 V, o que indica que o outro opto-acoplador, cuja entrada está em antiparalelo com este, estava acionado; a tensão sobe de forma praticamente linear pois corresponde à carga do capacitor  $C_{DT}$ , até atingir o valor de 1,5 V, momento em que a saída é ativada. Desse modo, o atraso entre a troca do comando do microcontrolador e a efetiva mudança na entrada do opto-acoplador foi medido em  $2,98 \mu s$ , praticamente o valor esperado pelo dimensionamento dos componentes.

Figura 48: Medição dos atrasos que geram o *dead-time* no *Gate driver*.



Fonte: Elaborado pelo autor.

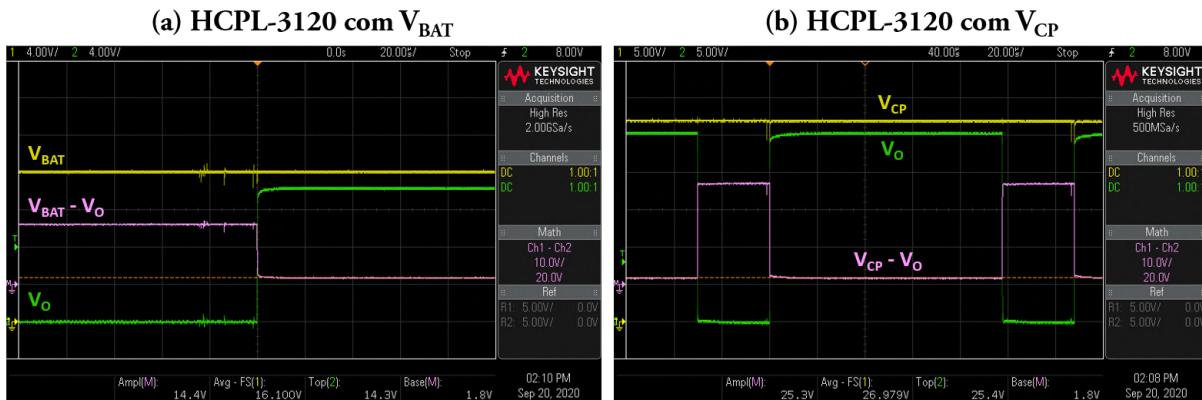
Entretanto, na Figura 48 à direita (b), temos outra medição interessante, a qual mostra um certo atraso no desligamento da saída, quando o comando de entrada se inverte, no valor de  $0,28 \mu s$ , e que deve ser levado em conta pois diminui o intervalo entre desligamento de um MOSFET e acionamento de outro no mesmo ramo. Considerando este atraso, o valor real do *dead-time* passa a ser  $2,98\mu s - 0,28\mu s = 2,7 \mu s$ , o que ainda fica dentro do calculado e é suficiente para evitar o *shoot through* nos ramos das pontes H.

Com o circuito do *Gate driver* que chaveia tanto a tensão da bateria para acionar os transistores *Low-side*, quanto a tensão do *Charge pump* para os *High-side*, foi medida a queda de tensão interna do HCPL-3120 nessas duas condições, com  $V_{CP}$  e  $V_{BAT}$ . Os resultados estão apresentados na Figura 49, e mostram uma queda de tensão interna de apenas 1,8 V, menor que o cenário crítico de 4 V considerado no projeto do circuito do *Charge pump*.

Explicando um pouco a Figura 49, o canal 1 (amarelo) representa a tensão aplicada no  $V_{CC}$  do HCPL-3120, o canal 2 (verde) representa a saída  $V_o$  do mesmo, e a curva em rosa é o canal M, uma função matemática interna do osciloscópio utilizado que plota a

diferença entre os canais 1 e 2. Desse modo, quando o opto-acoplador é acionado, a tensão em  $V_{CC}$  aparece em  $V_o$  descontada de uma pequena queda de tensão interna, medida pela diferença mostrada no nível baixo do canal M.

Figura 49: Medição da queda de tensão interna do HCPL-3120 durante chaveamento de  $V_{BAT}$  e  $V_{CP}$ .

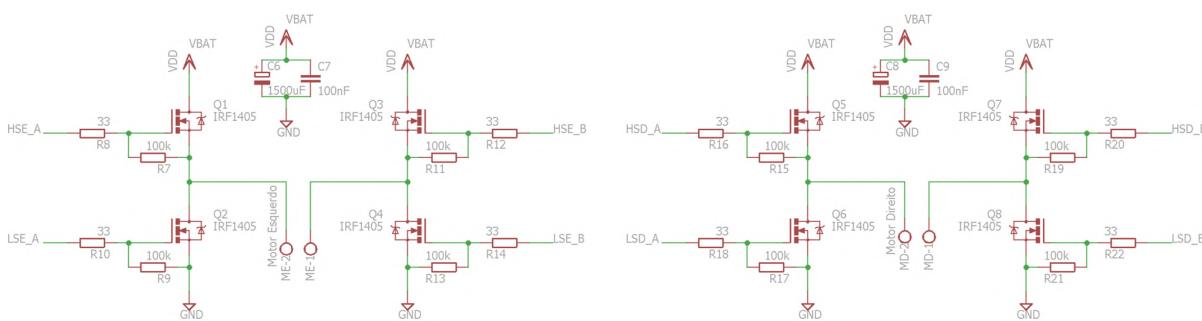


Fonte: Elaborado pelo autor.

### 3.1.4 Pontes H

O módulo de Pontes H é o de maior potência do circuito, consistindo na interface de acionamento dos motores de locomoção com a tensão da bateria, segundo o PWM e o sentido calculados pelo microcontrolador, cujos sinais são transferidos pelo *Gate Driver*. O conceito básico da ponte H já foi apresentado no capítulo do Embasamento Teórico, e o módulo das Pontes H está apresentado na Figura 50, contendo alguns incrementos de circuito que serão explicados em seguida.

Figura 50: Módulo das Pontes H.



Fonte: Elaborado pelo autor.

Podemos notar que as duas pontes H são idênticas, exceto pela atribuição dos pinos de comando, e o fato de uma ser dedicada ao acionamento do motor esquerdo, e outra ao

do motor direito. Associado a cada transistor notamos dois resistores, um entre o sinal de acionamento e o *gate*,  $R_{gate}$ , no valor de  $33\ \Omega$ , e outro entre porta e fonte (*gate* e *source*),  $R_{GS}$ , no valor de  $100\ k\Omega$ .

O primeiro,  $R_{gate}$ , tem a função de limitar o pico de corrente que ocorre no transiente das bordas de subida e descida dos pulsos de PWM. Em um primeiro momento, pode causar estranheza falar de corrente de acionamento em um transistor de efeito de campo, como é o MOSFET, contudo vale lembrar que ele possui capacitâncias parasitas intrínsecas de sua construção física, portanto podemos encarar o acionamento do MOSFET, de modo bastante simplificado, como a carga e descarga de um capacitor, cujo valor é quantificado no *datasheet* do componente como  $C_{iss}$  (*Input capacitance*)<sup>2</sup>. Sendo assim, no instante inicial, tanto de carga quanto de descarga, pode-se considerar que a tensão de acionamento está toda sobre o resistor, e a corrente de pico é calculada desta forma. O HCPL-3120, opto-acoplador do *Gate driver*, possui uma limitação de corrente de saída, exposta no rodapé da Tabela 13, no valor de  $2\ A$  ( $2,5\ A$  do corpo da tabela supõe condições diferentes de operação), e considerando a tensão de acionamento<sup>3</sup> como  $V_{BAT} = 16,8\ V$ , sem levar em conta a perda de tensão interna no HCPL, teríamos um valor mínimo de  $R_{gate} = 16,8V/2A = 8,4\ \Omega$ .

Tabela 13: Pico de corrente de saída permitido pelo opto-acoplador HCPL-3120.

Parameter	Symbol	Min.	Max.	Units	Note
"High" Peak Output Current	$I_{OH(PEAK)}$		2.5	A	b
"Low" Peak Output Current	$I_{OL(PEAK)}$		2.5	A	b

b. Maximum pulse width =  $10\ \mu s$ , maximum duty cycle = 0.2%. This value is intended to allow for component tolerances for designs with IO peak minimum =  $2.0\ A$ . See Applications section for additional details on limiting IOH peak.

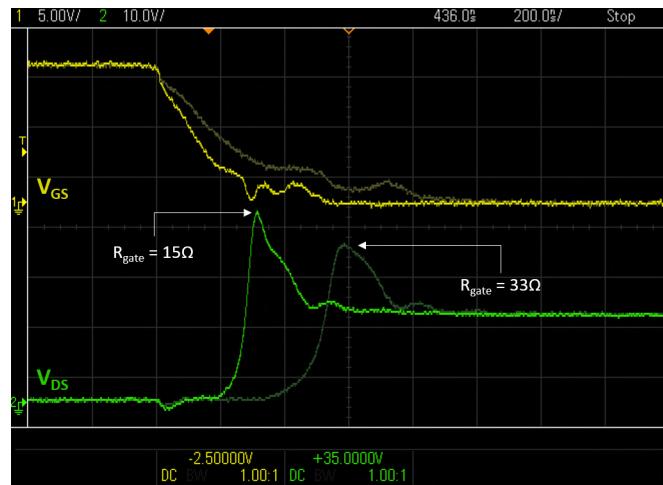
Fonte: Adaptado do *datasheet* do HCPL-3120, de 21/03/2016, página 13.

Utilizando um valor aproximadamente quatro vezes maior, isto é,  $R_{gate} = 33\ \Omega$ , temos um pico de  $0,51\ A$ , portanto uma boa margem para o máximo permitido pelo componente. Tomando  $C_{iss} = 5,48\ nF$ , temos uma constante RC de  $\tau = 181\ ns$ , da mesma ordem de grandeza dos tempos de subida (*rise time*,  $190\ ns$ ) e descida (*fall time*,  $110\ ns$ ) do transistor IRF1405 escolhido (Tabela 14). Além disso, um tempo maior para acionamento do transistor contribui para diminuição de oscilações e *overshoot* nos transientes de  $V_{DS}$ , como mostra a medição da Figura 51, com um mesmo circuito de carga e chaveamento, mas diferentes valores para  $R_{gate}$ .

<sup>2</sup>  $C_{iss} = C_{GD} + C_{GS}$ , segundo a fabricante TOSHIBA; disponível em <<https://bitly.com/rjIXK>>, acesso em 30/08/2020.

<sup>3</sup> A tensão de acionamento foi considerada única, apesar da tensão de *Charge pump*, pois quando esta é aplicada no *gate* do transistor, seu referencial passa a ser  $V_{BAT}$ , e dado que idealmente  $V_{CP} \leq 2.V_{BAT}$ , a consideração de tensão única é válida.

Figura 51: Comparaçāo entre valores de  $R_{gate}$  e sua influêncāo no comportamento do chaveamento do transistor.



Fonte: Elaborado pelo autor.

Tabela 14: Algumas características elétricas do MOSFET IRF1405.

#### Electrical Characteristics @ $T_J = 25^\circ C$ (unless otherwise specified)

	Parameter	Min.	Typ.	Max.	Units	Conditions
$V_{(BR)DSS}$	Drain-to-Source Breakdown Voltage	55	—	—	V	$V_{GS} = 0V, I_D = 250\mu A$
$\Delta V_{(BR)DSS}/\Delta T_J$	Breakdown Voltage Temp. Coefficient	—	0.057	—	V/ $^{\circ}C$	Reference to $25^\circ C, I_D = 1mA$
$R_{DS(on)}$	Static Drain-to-Source On-Resistance	—	4.6	5.3	$m\Omega$	$V_{GS} = 10V, I_D = 101A$ ④
$V_{GS(th)}$	Gate Threshold Voltage	2.0	—	4.0	V	$V_{DS} = 10V, I_D = 250\mu A$
$t_{d(on)}$	Turn-On Delay Time	—	13	—	ns	$V_{DD} = 38V$
$t_r$	Rise Time	—	190	—		$I_D = 101A$
$t_{d(off)}$	Turn-Off Delay Time	—	130	—		$R_G = 1.1\Omega$
$t_f$	Fall Time	—	110	—		$V_{GS} = 10V$ ④
$C_{iss}$	Input Capacitance	—	5480	—		$V_{GS} = 0V$

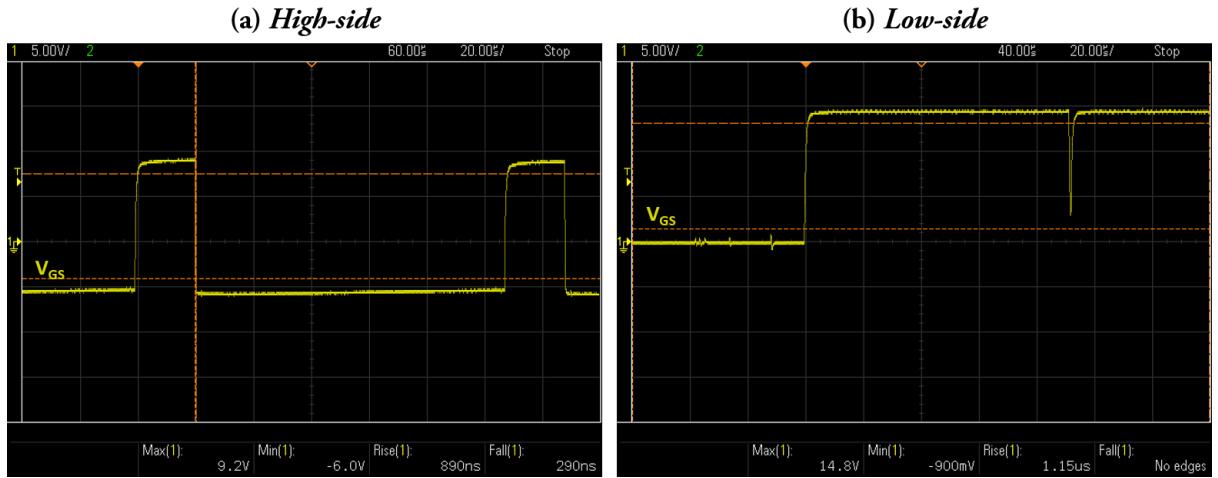
Fonte: Adaptado do *datasheet* do IRF1405, de 12/07/2004, página 2.

No sentido de verificar o dimensionamento do componente e seus efeitos, alguns resultados experimentais podem ser apresentados. O primeiro deles é a verificação da tensão de acionamento  $V_{GS}$  nos terminais de cada MOSFET, portanto após o resistor  $R_{gate}$ , tanto no *High-side* quanto no *Low-side*, e para os extremos de carga da bateria; tais gráficos estão apresentados nas Figuras 52 e 53 a seguir.

Pela Figura 52, cenário que mostra a bateria carregada, podemos notar a tensão máxima de 9,2 V no  $V_{GS}$  do transistor superior, suficiente para seu chaveamento, mas bem abaixo de  $V_{BAT}$ , valor esperado idealmente; no MOSFET inferior temos um valor de acionamento de 14,8 V, mais próximo do esperado porém ainda um pouco abaixo. É interessante observar que o *duty-cycle* utilizado para acionamento do *High-side* é menor que 25 %, escolhido para não representar uma carga mais elevada para  $V_{CP}$  e permitir que

sua tensão seja a maior possível.

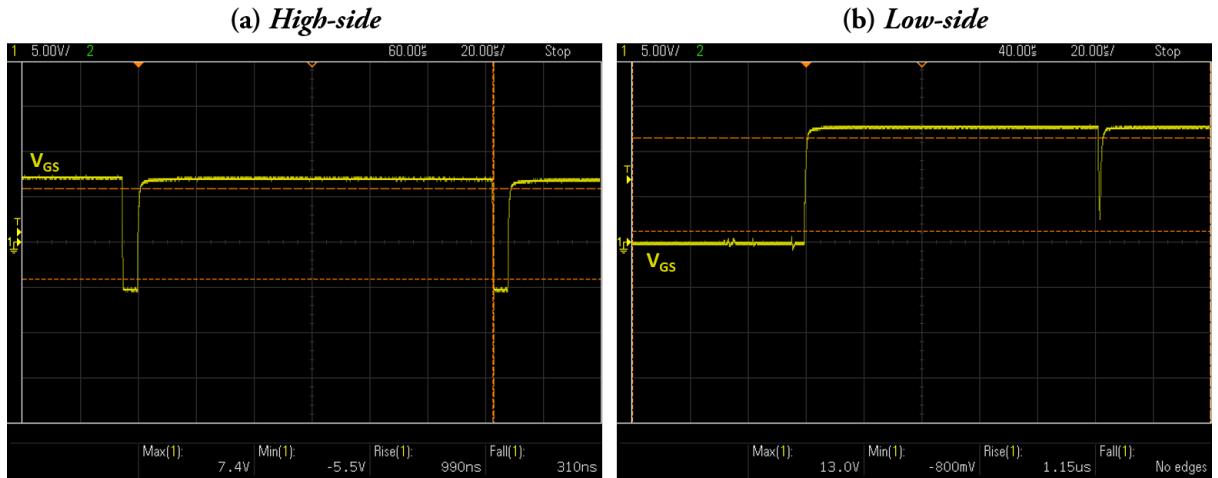
Figura 52: Medição da tensão de acionamento  $V_{GS}$  dos MOSFET's para  $V_{BAT} = 16,8$  V.



Fonte: Elaborado pelo autor.

Na Figura 53 temos o contrário, um cenário com bateria descarregada e o *duty-cycle* de acionamento de quase 100 %, para chegar ao extremo mínimo da tensão  $V_{CP}$ , que já é mais baixo por conta da bateria descarregada, e ainda deve suportar o acionamento quase ininterrupto dos transistores superiores. Diante disso, temos um valor máximo de 7,4 V em  $V_{GS}$  dos MOSFET's *High-side*, e de 13 V nos *Low-side*, os quais merecem as mesmas considerações anteriores.

Figura 53: Medição da tensão de acionamento  $V_{GS}$  dos MOSFET's para  $V_{BAT} = 14,8$  V.



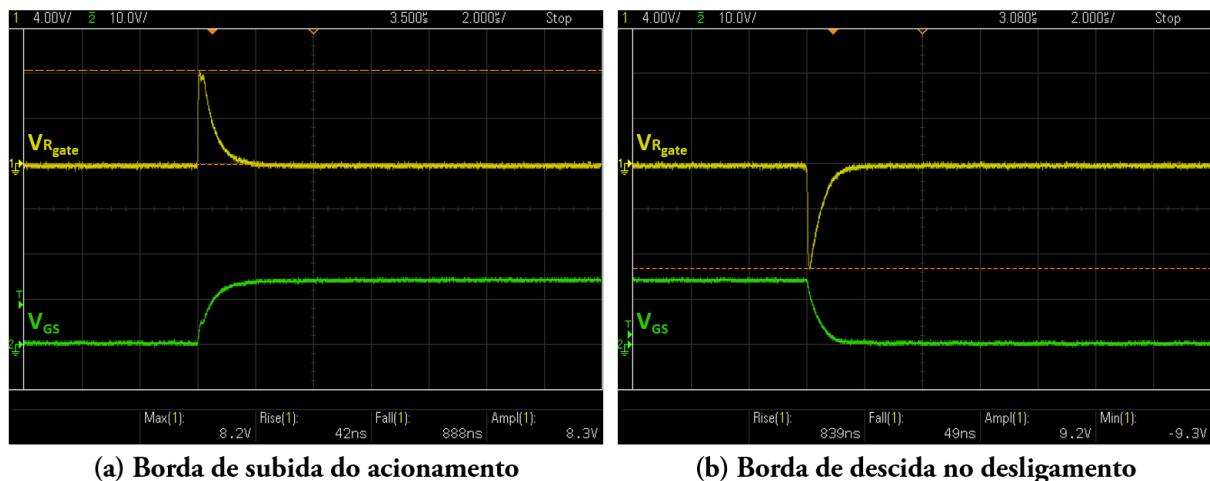
Fonte: Elaborado pelo autor.

Duas últimas notas quanto à tensão  $V_{GS}$  dos transistores superiores: mesmo no pior caso de carga da bateria e acionamento quase ininterrupto do componente, ainda

sobra uma margem de mais de 3 V para o  $V_{th_{MAX}} = 4$  V, de modo que seu chaveamento completo (região de saturação) não corre risco por conta de má polarização. Outro aspecto da tensão  $V_{GS}$  nos MOSFET's *High-side* é que seu nível baixo não tem valor de 0 V, como nos outros, mas tem um valor negativo, por volta dos -6 V, o que não é um problema, de acordo com o *datasheet* do componente, o qual informa que  $V_{GS_{MAX}} = \pm 20$  V.

Experimentalmente, a próxima etapa seria a medição da tensão sobre  $R_{gate}$  no circuito, a fim de calcularmos sua corrente de pico em cada acionamento e desativação do transistor. Como o objetivo é medir a corrente máxima que o resistor deve suportar, a condição de medição é aquela com bateria carregada, e o  $R_{gate}$  escolhido é o de um transistor *Low-side*, que recebe uma tensão de acionamento  $V_{GS}$  maior. As medições de acionamento e desativação estão apresentadas na Figura 54.

Figura 54: Medição da tensão de pico sobre  $R_{gate}$ , com bateria carregada e no resistor do transistor *Low-side*.



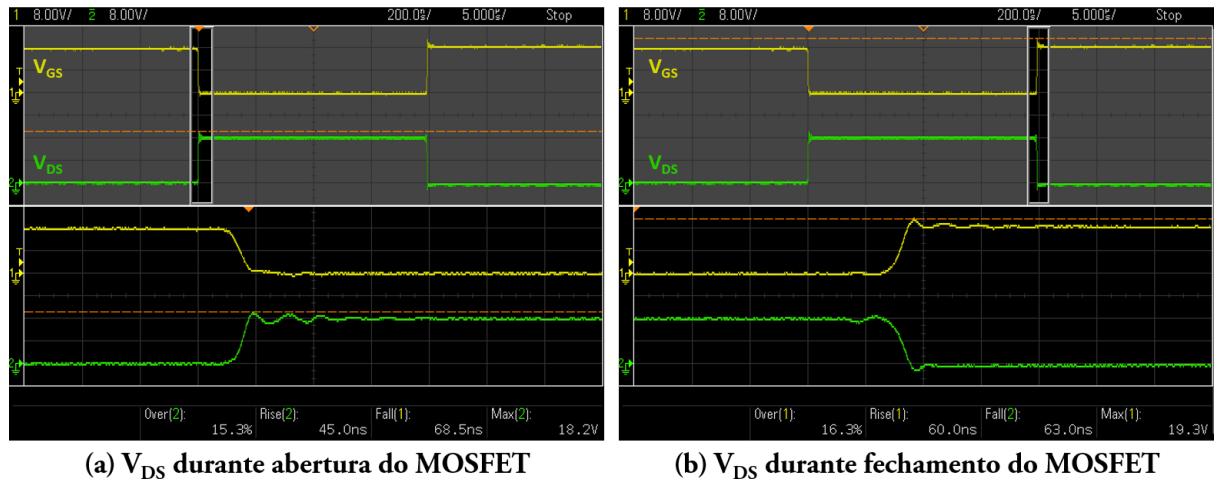
Fonte: Elaborado pelo autor.

Conforme esperado, o perfil da tensão corresponde à carga e descarga de um circuito RC, isto é, um pico de corrente positiva na carga do capacitor, e um pico de corrente negativa em sua descarga, com um perfil exponencial ( $e^{-\tau}$ ) de acomodação. Quanto aos valores absolutos em si, relembrando que  $R_{gate} = 33 \Omega$ , temos um pico positivo de 8,2 V, portando um pico de corrente de  $8,2V/33\Omega = 0,25$  A, e um pico negativo de tensão de -9,2 V, resultando em um pico de descarga de  $-9,2V/33\Omega = -0,28$  A. Tais correntes são praticamente a metade do valor esperado durante o dimensionamento teórico de  $R_{gate}$ , e ficam bastante longe do limite de 2 A que o HCPL-3120 pode fornecer, de modo que, olhando apenas para a limitação de corrente, o resistor ainda poderia ser reduzido à metade sem risco de expor o opto-acoplador.

Entretanto, um dos benefícios citados em se aumentar a resistência de *gate* é a diminuição do *overshoot* de tensão em  $V_{DS}$  durante o chaveamento, conforme exemplificado

pela Figura 51, e esta tensão é importante ser mantida baixa pois, se ultrapassar o valor máximo permitido para o componente, pode levar a uma falha permanente do mesmo. A medição à esquerda (a) da Figura 55 mostra o *overshoot* de  $V_{DS}$  em 15 %, que foi o típico observado durante o chaveamento do motor travado, um valor não muito alto, mas que aumentaria com a diminuição de  $R_{gate}$ .

Figura 55: Medição do comportamento de  $V_{DS}$  durante o chaveamento do MOSFET *Low-side*.



Fonte: Elaborado pelo autor.

Aproveitando as mesmas medições da Figura 55, vemos que os tempos de subida e descida de  $V_{DS}$  estão bem inferiores aos informados na Tabela 14, e isso deve-se às condições para as quais estes foram estabelecidos na folha de dados do componente:  $V_{DD} = 38$  V e  $I_D = 101$  A, que são valores bem superiores aos usados neste projeto,  $V_{BAT} = 16,8$  V e  $I_{stall} = 15$  A.

Quanto ao outro resistor associado a cada MOSFET das pontes H,  $R_{GS}$  atua como um *pull-down*, ou seja, um resistor com a função de conectar o terminal de *gate* ao terminal de *source*. Sua utilização tem ganho para o caso de a saída do opto-acoplador falhar e ficar em aberto, de modo que a carga do gate se manteria por um tempo bem mais longo caso  $R_{GS}$  não estivesse presente para absorvê-la. Sua função também é a de dissipar algum acoplamento ou energia estática que possa colocar o transistor em condução caso não seja absorvida. O valor de  $R_{GS}$  deve ser alto para não tornar-se uma carga significativa para a tensão de acionamento no *gate*, geralmente acima de  $10\text{ k}\Omega$ , mas não tão alta que compare-se a um circuito aberto, podendo alcançar até alguns  $\text{M}\Omega$ ; nesta faixa, o valor escolhido foi  $100\text{ k}\Omega$ .

Passando para o componente principal, o transistor foi escolhido com base em requisitos técnicos, preço e disponibilidade. Foram compilados em uma lista todos os MOSFET's disponíveis em uma das principais lojas de eletrônica de São Paulo, filtrados

pelas maiores capacidades de corrente e de tensão de operação, ainda num preço inferior a R\$ 10,00 por unidade. Os finalistas estão apresentados na Tabela 15, em ordem crescente do preço consultado em 05/02/2020.

Tabela 15: Opções de MOSFET para as pontes H.

MOSFET:	<b>IRLB3813</b>	<b>IRF2807</b>	<b>IRF1405</b>	<b>IRFB3206</b>
<i>Preço:</i>	R\$ 4,80	R\$ 5,50	R\$ 7,60	R\$ 9,50
$R_{DS_{ON}}(max)$	1,95 mΩ	13 mΩ	5,3 mΩ	3 mΩ
$I_{D_{max}}(package)$	120 A	75 A	75 A	120 A
$V_{DSS}$	30 V	75 V	55 V	60 V
$C_{iss}$	8,42 nF	3,82 nF	5,48 nF	6,54 nF

Fonte: Elaborado pelo autor com base em informações de *datasheet*, e preços consultados em 05/02/2020.

A primeira opção, o IRLB3813, foi inicialmente utilizada e parecia ser a melhor delas, porém sua tensão máxima de operação mostrou-se ser um problema. Na Figura 55 temos o comportamento típico de  $V_{DS}$ , em que o pico de tensão chega a 18,2 V, porém durante alguns testes de operação da Ponte H, chaveando com a bateria carregada e uma carga de caráter indutivo, foi detectado um pico de tensão de 32 V, conforme mostrado na Figura 56. Para o componente citado,  $V_{DS}$  não poderia ultrapassar 30 V; o resultado foi que, depois de alguns minutos de operação, o IRLB3813 queimou, entrando em curto - e levando alguns componentes consigo.

Figura 56: Tensão máxima medida em  $V_{DS}$  durante a abertura do transistor.



Fonte: Elaborado pelo autor.

Diante deste resultado, o IRLB3813 foi descartado como uma das opções possíveis. Comparando as outras opções, a próxima mais barata, IRF2807, tem um  $R_{DS_{ON}}$  maior que o dobro do valor do mesmo parâmetro do IRF1405, isso pelo custo de R\$ 2,10 a mais.

O  $R_{DS_{ON}}$  determina o valor da potência de condução do MOSFET, portanto influencia no quanto ele deve esquentar; ponderando menos da metade da potência com o aumento de preço citado, o IRF1405 foi o escolhido entre os dois. Sua tensão de operação é um pouco mais baixa, como podemos ver na Tabela 15, porém superior ao valor máximo do pico medido e exibido na Figura 56, portanto aceitável para utilização no circuito.

Por fim, temos a comparação dos dois últimos, IRF1405 e IRFB3206: o segundo seria R\$ 1,90 mais caro, teria uma queda de 43 % no  $R_{DS_{ON}}$ , e a tensão máxima de ambos é praticamente a mesma. Neste caso, aumentar ainda mais o valor desse componente da ponte H pareceu não compensar comparado ao ganho nas perdas de condução, então ficamos com a economia de R\$ 15,20 no preço da placa. O componente escolhido, portanto, é o MOSFET IRF1405, mas caso testes na prática com os robôs de combate mostrem que um  $R_{DS_{ON}}$  ainda menor é necessário, já fica disponível o IRFB3206 como opção, cuja pinagem e encapsulamento são idênticos aos do IRF1405.

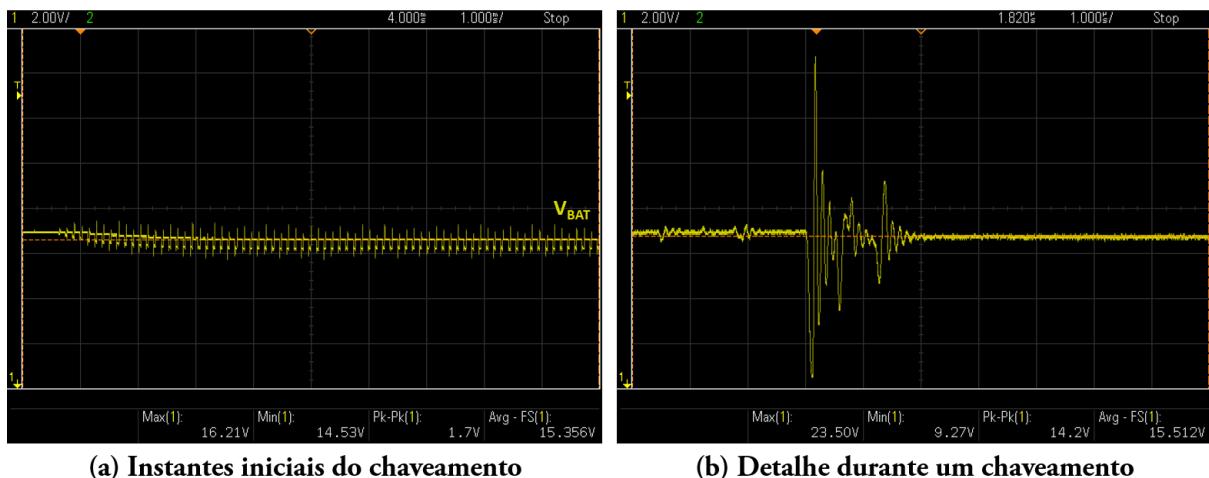
Os últimos componentes ainda não abordados no módulo são os capacitores em paralelo, um de 1500  $\mu\text{F}$  e outro de 100 nF, um par para cada ponte H. Sua presença é importante para suprir a corrente nos transientes do acionamento do motor, quanto ocorre um "surto" de demanda. Temos dois em paralelo pois o de maior valor é do tipo eletrolítico, que consegue atingir capacitâncias maiores para encapsulamentos menores, porém tem uma resistência série mais elevada e não consegue responder com tanta velocidade aos transientes; para isso usamos o segundo em paralelo, que utiliza-se de outra tecnologia com resistência série menor, como o de cerâmica ou poliéster, neste caso, para filtrar os ruídos de alta frequência causados pelo chaveamento.

Dada a aplicação, a regra de dimensionamento é quase que "quanto maior, melhor", pois mais carga eles teriam para fornecer no caso dos surtos de corrente, e menos a tensão oscilaria; por outro lado, o tamanho físico começa a ser um problema, ainda mais porque a tensão de operação de cada um deve ser de, pelo menos, 35 V, e a placa não deve ser muito volumosa, dada a restrição do espaço interno do robô, exposta na Introdução. Outro aspecto de uma capacitância muito elevada em paralelo com a bateria é sua corrente inicial de carga, ou seja, o pico de corrente proveniente da bateria quando conectada em paralelo com tais capacitores descarregados. O melhor encontrado foi um capacitor de 1500  $\mu\text{F}$  com 16 mm de diâmetro, a ser posicionado perto de cada ponte H, juntamente com o capacitor de poliéster de 100 nF, conforme indicado na Figura 50 do módulo.

Na implementação prática, tentou-se obter um gráfico do ruído na tensão de bateria, ante o chaveamento dos motores, sem a utilização de capacitor algum, porém tal ruído tinha amplitude tão alta que  $V_{BAT}$  ultrapassava os 35 V máximos de alimentação do HCPL-3120 e chegou a queimar dois chips. Diante disso, foi possível comparar apenas a utilização ou não do segundo capacitor, de poliéster, na função de filtro dos ruídos de chaveamento. Tal comparação pode ser conferida a partir das Figuras 57 e 58 a seguir.

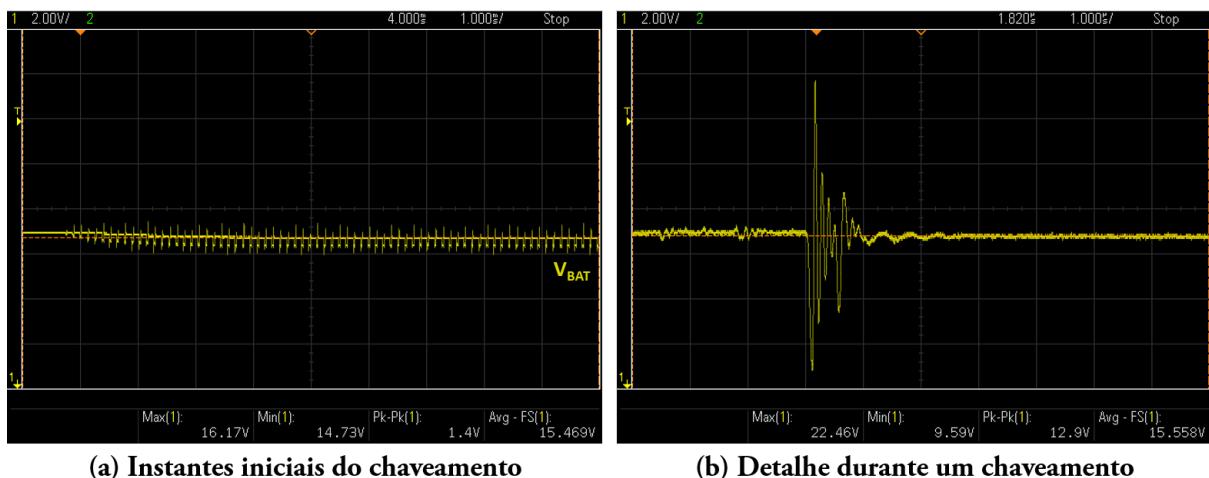
A bateria não estava completamente carregada durante os testes de ruído, portanto a tensão de alimentação era por volta de 15,5 V, conforme nota-se pelo valor médio ("Avg-FS") indicado nos gráficos das figuras. Em ambas temos, à esquerda (a), uma visão geral da alimentação nos primeiros instantes de chaveamento, com o motor ainda parado, e à direita (b) a ampliação do ruído causado no momento de algum acionamento ou desativação de ponte H segundo seu PWM, com mais resolução nos eixos vertical e horizontal.

Figura 57: Medição do ruído de alta frequência em  $V_{BAT}$  com o uso do capacitor eletrolítico (1500  $\mu$ F) apenas.



Fonte: Elaborado pelo autor.

Figura 58: Medição do ruído de alta frequência em  $V_{BAT}$  com o uso do capacitor eletrolítico (1500  $\mu$ F) em paralelo com o capacitor de poliéster (100 nF).



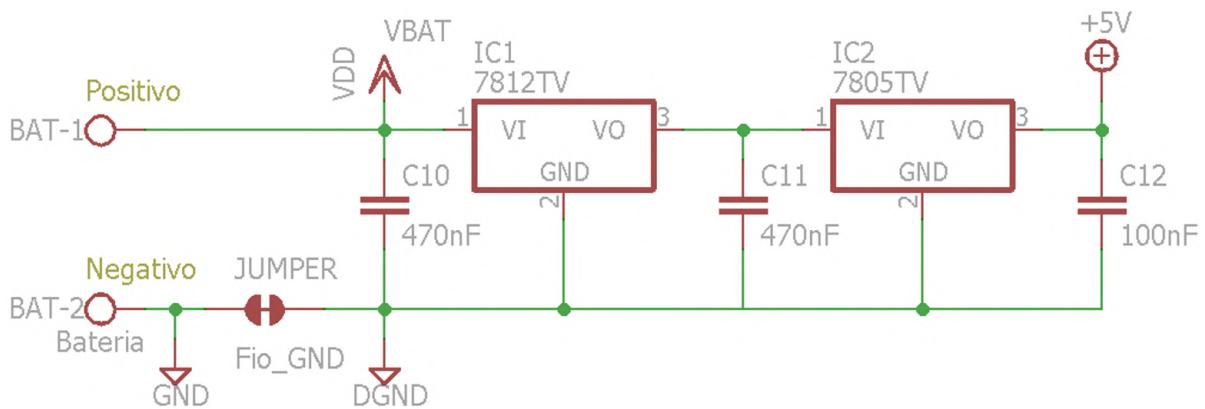
Fonte: Elaborado pelo autor.

Da Figura 57, que mostra o ruído de alta frequência presente com o uso apenas do capacitor de  $1500 \mu\text{F}$ , para a Figura 58, a qual exibe o ruído depois do acréscimo do capacitor de poliéster de  $100 \text{nF}$ , podemos notar duas melhorias: o valor de pico a pico diminuiu de  $14,2 \text{ V}$  para  $12,9 \text{ V}$ , portanto  $1,3 \text{ V}$  a menos, e a duração do mesmo caiu, em valores aproximados, de  $2 \mu\text{s}$  para  $1 \mu\text{s}$ . Estas diferenças mostram como é importante acrescentar um capacitor de resposta mais rápida em paralelo com o de maior capacitância; seria ainda melhor se o valor de capacitância do de poliéster fosse maior, porém este tipo de tecnologia não comporta valores maiores em encapsulamentos adequados para a aplicação em questão. Cabe ao módulo de filtros, discutido a seguir, absorver este ruído e impedir que ele chegue a circuitos e componentes mais sensíveis, como o microcontrolador e o circuito do receptor do rádio controle.

### 3.1.5 Filtros e o Regulador de Tensão

O último módulo de *hardware* é o responsável por filtrar a tensão da bateria e produzir a tensão de  $5 \text{ V}$  utilizada pelo microcontrolador e pelo receptor do rádio controle, a qual deve ser a mais estável e livre de ruídos possível, principalmente diante do cenário de chaveamentos exibido anteriormente. Também participa dos filtros do sistema o par de capacitores em cada ponte H, apresentados há pouco no módulo anterior. O circuito correspondente a este bloco está apresentado na Figura 59.

Figura 59: Módulo do Regulador de Tensão.



Fonte: Elaborado pelo autor.

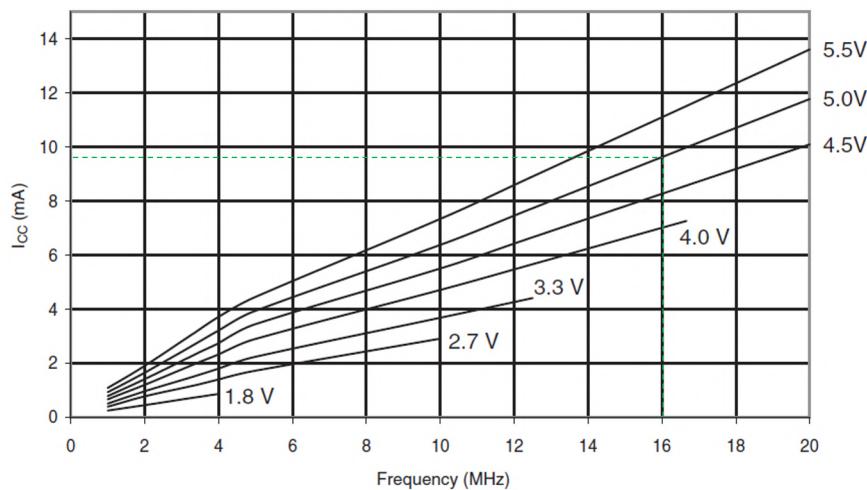
A primeira coisa a se notar é a conexão do terminal positivo da bateria no que denominamos  $V_{BAT}$  anteriormente, e o negativo no que corresponde ao GND. A segunda coisa digna de atenção é a diferenciação entre GND e DGND, este tomado como referencial da parte digital e de baixa potência, e aquele para o resto do circuito, como referencial da parte de maior potência, ligados por apenas uma trilha na placa, representada pelo jumper "Fio\_GND". O propósito desta ligação única entre ambos os referenciais é evitar

os chamados "*ground loops*", flutuações no referencial de terra que podem acontecer devido ao chaveamento de cargas e altas correntes instantâneas, de modo que dois pontos que deveriam ter a mesma tensão passam a ter uma diferença, interferindo no funcionamento do circuito. Dada a importância de se manter uma tensão de alimentação "limpa" no circuito do microcontrolador, apenas ele e a entrada dos opto-acopladores utilizam-se desse referencial digital (DGND), além, é claro, do receptor do rádio controle.

A geração dos 5 V é feita por reguladores lineares de tensão; a escolha foi feita por diminuir a complexidade do circuito, que seria maior com um conversor DC-DC chaveado, como um conversor *buck*. A desvantagem são as perdas em potência, que aumentam junto do consumo de corrente por conta da queda de tensão em cada regulador. Olhando para a tensão 5 V, ela deve prover aproximadamente 10 mA para o microcontrolador, conforme o gráfico na Figura 60, 15,3 mA para o receptor do rádio controle, conforme medição feita em operação normal do mesmo, e 53 mA do máximo de cinco entradas dos opto-acopladores acionados simultaneamente<sup>4</sup>, portanto um total de 78,3 mA.

Figura 60: Consumo do ATmega328P com 5 V de alimentação e 16 MHz de frequência de operação.

**Figure 31-286. ATmega328: Active Supply Current vs. Frequency (1-20MHz)**



Fonte: Adaptado do *datasheet* do ATmega328P, de 11/2015, página 468.

Utilizando reguladores lineares, a queda de tensão sobre eles é a diferença entre a tensão de entrada, 16,8 V no máximo, e a de saída, 5 V, portanto 11,8 V de queda, e proporcionando uma corrente de aproximadamente 80 mA (desconsiderando o consumo do próprio regulador), temos uma potência máxima de 0,94 W perdida na conversão. Utilizar apenas um regulador para abaixar a tensão é possível no aspecto de tensão de entrada permitida, porém ele esquentaria muito dada a potência calculada. A solução encontrada

<sup>4</sup> A corrente de acionamento de cada LED do opto-acoplador tem seu valor típico calculado por  $(V_{comando} - V_{LED})/R_{LED} = (5 - 1,5)/330 = 10,6 \text{ mA}$ .

foi a de utilizar dois reguladores em cascata, um que abaixa a tensão para 12 V, resultando em uma potência de  $(16,8 - 12) \times 80m = 0,38$  W sobre ele, e outro para abaixar a tensão de 12 V para 5 V, ficando com  $(12 - 5) \times 80m = 0,56$  W. Além disso, algum ruído que venha a passar pelo primeiro regulador ficará atenuado e, provavelmente, será absorvido pelo segundo, contando com o efeito dos capacitores de filtro.

Os reguladores escolhidos foram o LM7812 (para 12 V) e o LM7805 (para 5 V). Seu *datasheet* - da família de reguladores lineares 78xx - indica um circuito recomendado para operação a fim de garantir estabilidade na tensão de saída, o qual está exposto na Figura 61. Nela, notamos o acréscimo de dois capacitores, um na entrada, de 330 nF, e outro na saída, de 100 nF.

Figura 61: Circuito recomendado para o regulador de tensão da família LM78xx.

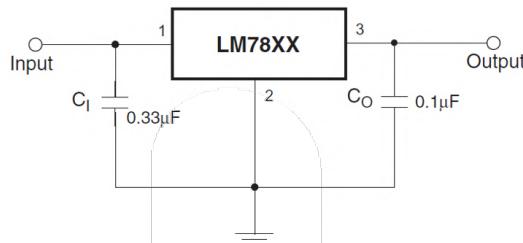


Figure 9. Fixed-Output Regulator

Fonte: Extraído do *datasheet* da Fairchild da série LM78xx, de 09/2014, página 19.

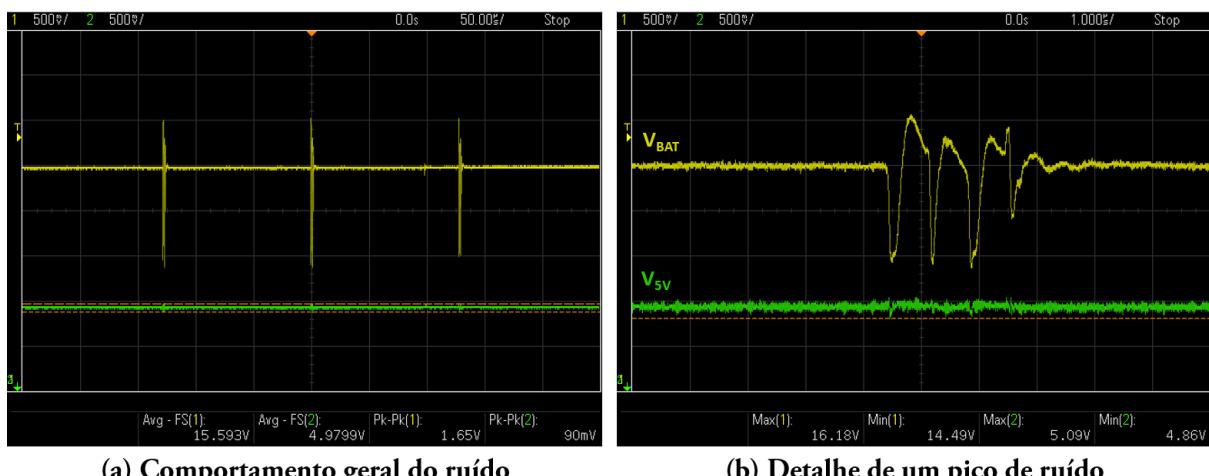
Baseando-se no circuito recomendado, foi montado o cascamenteamento dos reguladores de tensão com um capacitor um pouco mais alto na entrada, de 470 nF, por conta da presença dos ruídos de chaveamento advindos da parte de potência e já exibidos; um capacitor de mesmo valor é posicionado no ponto que é a saída do primeiro mas a entrada do segundo regulador, e um último na saída do de 5 V, de acordo com o valor recomendado de 100 nF.

A verificação dessa implementação de filtros e geração dos 5 V foi realizada com medições em três etapas progressivas na intensidade de ruído em  $V_{BAT}$ : a primeira, com o circuito energizado mas sem chaveamento do motor; a segunda, com o acionamento do motor livre, sem carga alguma no eixo além de sua caixa de redução; e a terceira, com o acionamento dos motores travados mecanicamente, sem possibilidade de rotação. Tais etapas estão apresentadas sucessivamente nas Figuras 62, 63 e 64 a seguir.

Começando pela condição de medição sem acionamento dos motores, cujos gráficos estão na Figura 62, observa-se um ruído de baixa amplitude em  $V_{BAT}$ , de 1,7 V de pico a pico, e em uma frequência de aproximadamente 8 kHz, próxima dos 7,8 kHz do PWM de acionamento de cada MOSFET *High-side*. A explicação mais aprofundada para este pulso residual, mesmo quando não pede-se o acionamento dos transistores, será feita a seguir,

na seção de *software*; por agora, basta saber que este pulso residual não é suficiente para colocar o transistor em condução, mas por acionar o opto-acoplador do *Gate driver*, já é suficiente para gerar um distúrbio na alimentação e resultar no ruído observado. À direita da Figura 62, o gráfico (b) nos mostra que esta perturbação não se transmite de forma alguma para a tensão de 5 V, portanto não há com o que se preocupar neste estado de *stand-by* do sistema, em espera.

Figura 62: Medição do ruído presente nos 5 V gerados - cenário sem acionamento dos motores.



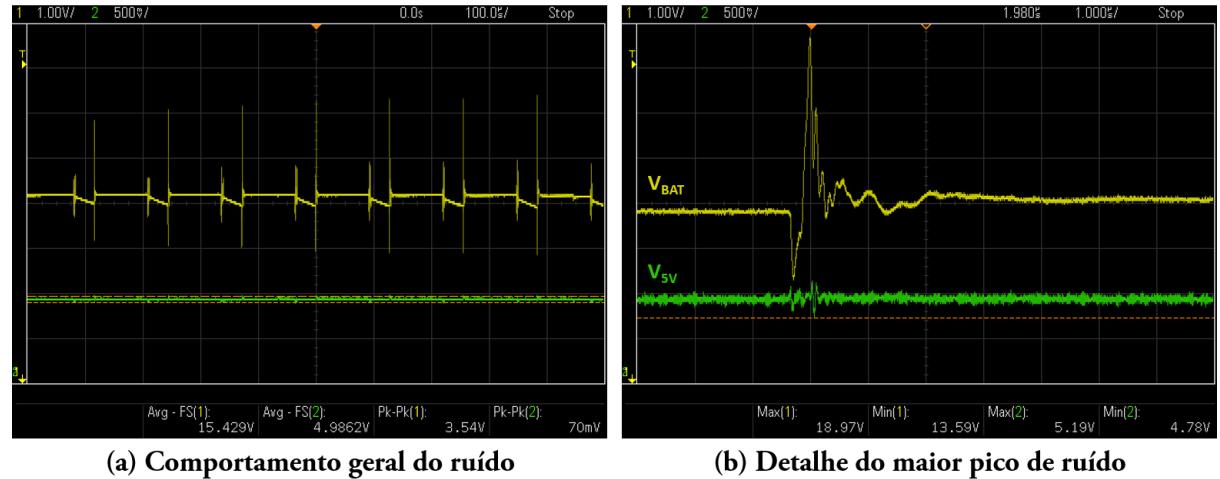
Fonte: Elaborado pelo autor.

A Figura 63 traz medições do cenário de acionamento dos motores em vazio e em regime, portanto consumindo sua mínima corrente. Neste caso, o ruído chega a ter 5,4 V de pico a pico em  $V_{BAT}$ , cuja intensidade é maior por um período de  $1 \mu s$ . Ao contrário do cenário anterior, aqui nota-se uma discreta variação da tensão 5 V simultaneamente ao ruído, cuja amplitude chega a 0,4 V de pico a pico.

O último cenário analisado é o de maior estresse para o circuito de acionamento, pois toma os dois motores travados, em *stall*, consumindo sua maior corrente possível. O ruído, por consequência, tem o maior valor de pico a pico, medido em 14,2 V, e aparecendo com maior intensidade na tensão 5 V. Nela, temos um ruído de até 500 mV de pico a pico por aproximadamente  $0,5 \mu s$ , e neste intervalo um pico maior, bem curto, de máximo em 6,13 V e mínimo em 4,28 V (portanto 1,85 V de pico a pico).

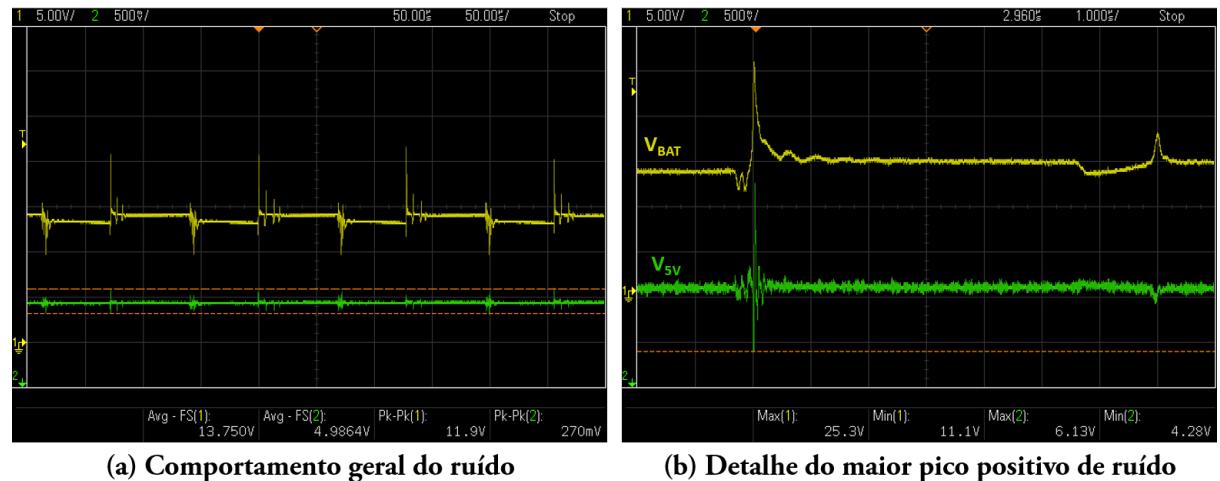
A tensão de 5 V alcançando instantaneamente o pico de 6,13 V ultrapassa o valor máximo de alimentação, de 6 V, permitido para o ATmega328P, porém não foi notado dano algum ou mau funcionamento do microcontrolador durante os testes; portanto, é provável que a duração desse pulso, ultrapassando 0,13 V do limite, não seja suficiente para causar algum dano imediato ao chip, mas no longo prazo pode degradá-lo mais rapidamente e diminuir sua vida útil. Quanto ao valor mínimo de 4,28 V, ele precisaria se

Figura 63: Medição do ruído presente nos 5 V gerados - cenário com acionamento dos motores em giro livre, sem carga.



Fonte: Elaborado pelo autor.

Figura 64: Medição do ruído presente nos 5 V gerados - cenário com acionamento dos motores travados.



Fonte: Elaborado pelo autor.

manter por mais tempo para acionar o *reset* do microcontrolador, que consiste em um circuito RC - o tempo necessário seria algo por volta de 7 ms, portanto não representa um problema.

Retomando a análise de todos os cenários de medição, o único que traz alguma preocupação para o funcionamento do circuito é o último, cuja condição de acionamento dos motores não representa a mais frequente de funcionamento do circuito, isto é, os motores ao serem acionados saem da inércia e passam a movimentar as rodas e locomover

o robô, de modo que têm uma certa carga em seu eixo, mas não algo extremo como a completa imobilização realizada no teste.

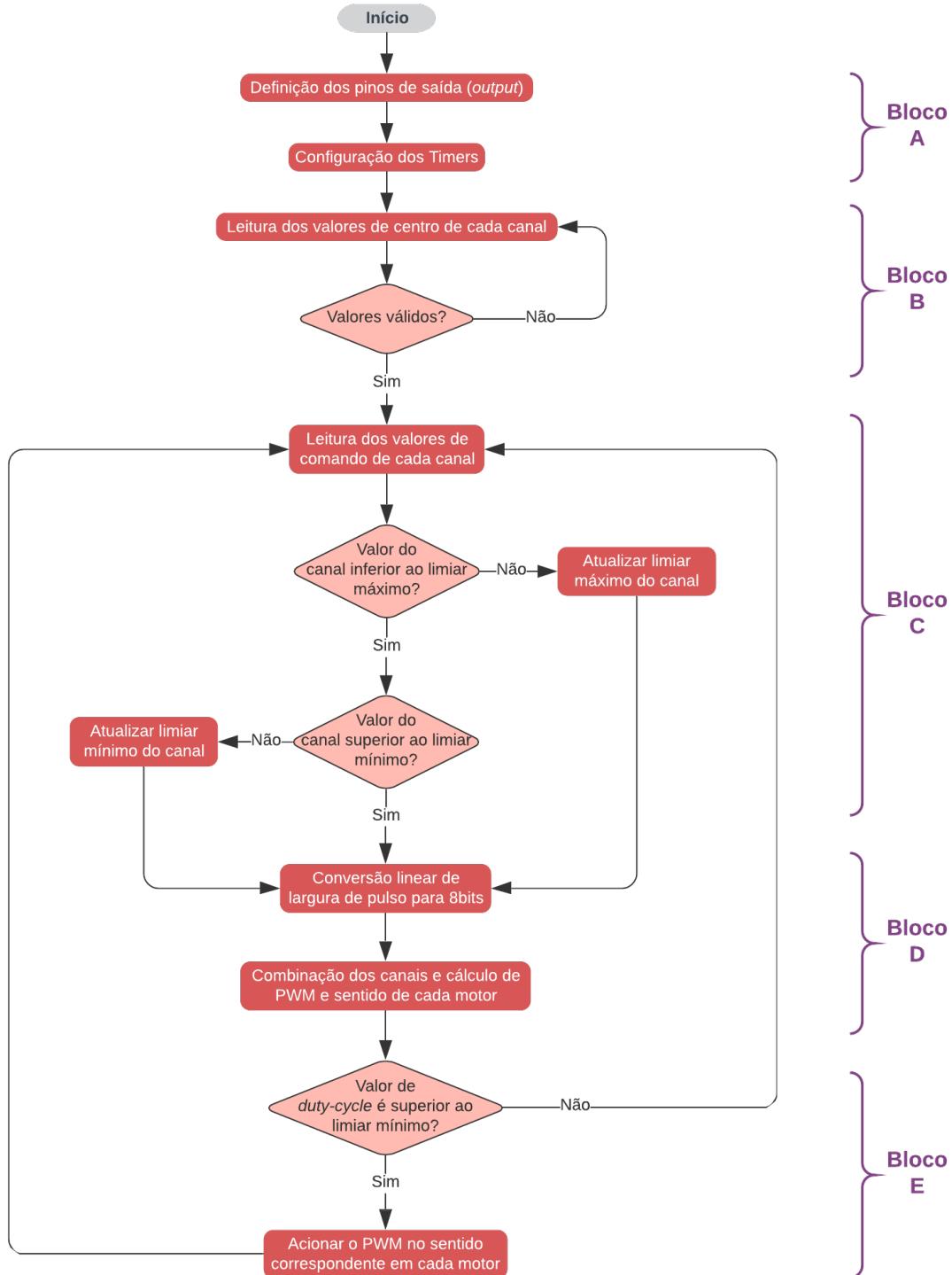
De qualquer forma, as medições realizadas levantam o alerta quanto à eficácia deste módulo, que precisa ser melhorada para condições extremas, como a exibida.

Com esta última análise, fechamos o desenvolvimento dos módulos de *hardware*, e o circuito completo do dispositivo pode ser encontrado no Apêndice A.

### 3.2 Software

O algoritmo a ser executado pelo microcontrolador está apresentado na forma de um fluxograma na Figura 65 a seguir.

Figura 65: Fluxograma do algoritmo de *software*.



Fonte: Elaborado pelo autor.

O fluxograma foi dividido em cinco partes, de acordo com os blocos indicados, a fim de facilitar sua explicação. Sua elaboração já levava em conta os recursos de *hardware*, principalmente os geradores de PWM do microcontrolador e *timers* internos, porém ele poderia ser facilmente codificado em outras plataformas que não a do Arduino Uno e seu ATmega328P. A explicação dada a seguir, sobre a função e implementação de cada bloco, é feita com base no microcontrolador utilizado, de modo a tornar mais concreta a visualização de cada processo pela correspondência com os circuitos apresentados. O código elaborado consta no Apêndice B, após o final deste texto.

### 3.2.1 Definições e constantes

A primeira etapa é definir alguns valores de referência, valores constantes atribuídos a nomes que possam ser utilizados no momento da codificação. Por exemplo, a modulação PPM, que será lida em dois canais do receptor, trabalha com pulsos de 1 ms a 2 ms, ao ponto que podemos considerar um desvio de leitura e definir os valores de 0,9 ms e 2,1 ms como as durações limite mínima e máxima que um pulso deve ter para ser considerado válido, plausível.

Ainda no campo do receptor do rádio controle, sabemos que a alavanca de controle ("joystick") fica em repouso na metade da excursão possível de cada canal<sup>5</sup>, de modo que o pulso correspondente deve ter aproximadamente 1,5 ms de largura, outro valor de referência definido. Nesta etapa, também estabelecemos a referência de 1,8 ms como a duração máxima do pulso, e 1,2 ms como a duração mínima em cada canal, valores que serão substituídos caso a leitura identifique que essas referências foram ultrapassadas pela leitura do canal, mas ainda estejam dentro dos limites de plausibilidade já adotados, conforme será explicado posteriormente.

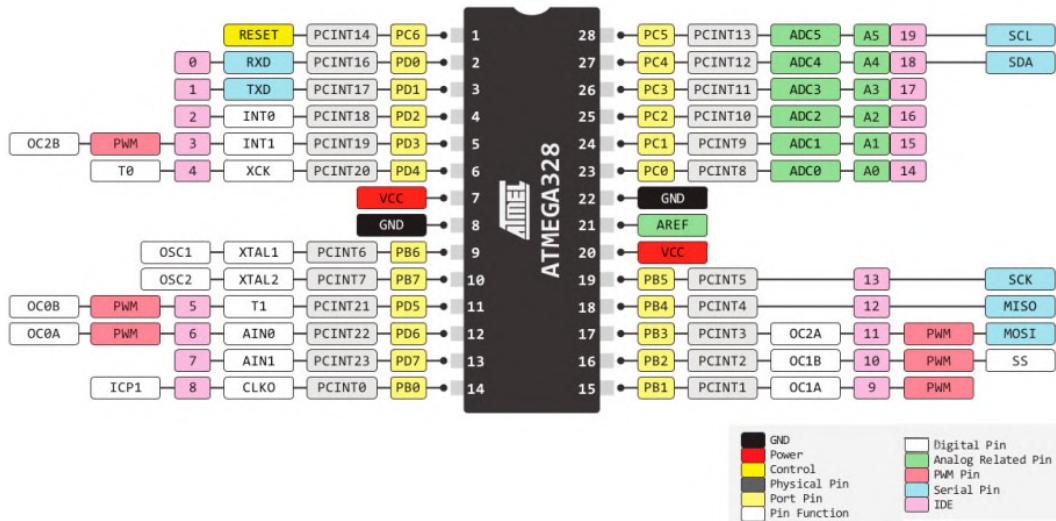
Outro valor a ser definido para a leitura dos sinais PPM do receptor do rádio controle é o de *timeout* do pulso, ou seja, por quanto tempo a entrada de leitura pode ficar esperando pelo surgimento de um pulso antes de declarar que ele está ausente, falhou. Para este parâmetro foi determinado 30 ms, que é uma vez e meia o intervalo de aplicação do pulso por canal (20 ms).

Uma penúltima definição importante é a dos pinos de entrada e saída, ou seja, a correlação entre o pino físico do ATmega328P, seu número de identificação na IDE, e um mnemônico que simbolize sua função. Desta forma, fica mais simples e clara a utilização das entradas e saídas durante o desenvolvimento do código. A Figura 66 mostra diversas informações relacionadas aos pinos do ATmega328P, dentre elas a identificação dos pinos pela IDE do Arduino, e a posição das saídas "OCnx" de cada *timer*. Os mnemônicos foram

<sup>5</sup> Existe uma exceção, a qual será explicada mais à frente no texto; para os propósitos do combate de robôs, consideraremos que o repouso dos canais deve ser na metade de excursão possível.

construídos segundo as seguintes premissas: temos duas pontes H, uma para cada motor, direito (D) e esquerdo (E); e em cada ponte H temos dois ramos verticais, com um par de transistores cada (*High* e *Low-side*), sendo o ramo A o da esquerda, e B o da direita (verificar Figura 50). Deste modo, para comandar o MOSFET *High-side* do ramo A da ponte H do motor direito, acionaremos o pino `HsD_A`, assim como o pino definido por `LsE_B` comanda o transistor *Low-side* no ramo B da ponte H do motor esquerdo.

Figura 66: Pinos do ATmega328P correlacionados com suas funções e identificação na IDE do Arduino.



No caso do ATmega328P, o *Timer 1* ficou reservado para controle da ponte H do motor esquerdo, suas saídas `OC1A` e `OC1B` foram nomeadas `HsE_A` e `HsE_B`, respectivamente. O motor direito teve sua ponte H comandada pelos pinos `OC2A` e `OC2B` do *Timer 2*, que passaram a ser referenciados por `HsD_A` e `HsD_B`, respectivamente. O *Timer 0* também teve seu pino `OC0A` utilizado para gerar a onda quadrada de comando do *Charge pump*, portanto foi atrelado ao nome `CP_PWM`. Todos os pinos de controle dos transistores *Low-side* são utilizados na sua função de saída digital, escolhidos dentre os disponíveis.

Quanto aos pinos de entrada, foram utilizados os terminais de leitura analógica, definidos como FT para o canal ELE do rádio controle, responsável pelo movimento para Frente e para Trás do robô, e DE para o canal RUD do rádio controle, responsável pelos movimento de curva do robô, à Direita e à Esquerda.

Por fim, foi definido um valor mínimo de 15 % de *duty-cycle* calculado para que o PWM seja efetivamente aplicado na saída, ou seja, na respectiva ponte H. Transpondo para um controle de *duty-cycle* em 8 bits, o limiar corresponde ao valor absoluto de 38 da faixa que vai até 255. O motivo deste limiar é que, antes dele, o PWM estaria sendo

aplicado em um motor que, com toda a carga mecânica em seu eixo, não teria torque suficiente para resultar em nenhum movimento; a eletrônica, portanto, faria o chaveamento de um motor em *stall*, travado, cenário em que consome sua maior corrente, causando um aquecimento de componentes e consumo de corrente sem nenhum benefício prático. O valor definido pode - e deve - ser ajustado conforme testes com a locomoção instalada no robô.

Os comentários ao longo do código são de grande valia neste momento, por trazerem uma breve explicação da atribuição feita e o que aquele valor definido representa, e podem ser conferidos no Apêndice B.

### 3.2.2 Bloco A - Configurações iniciais

Com todas as definições feitas, passamos à primeira etapa da execução do código, que é a definição dos pinos como entradas ou saídas, conforme sua função, nos registradores do microcontrolador. Além disso, é preciso configurar o modo de operação dos *timers* utilizados.

O estabelecimento da frequência de operação dos *timers* foi feito a partir de testes práticos com o motor, pois precisava-se equilibrar o controle de corrente e torque com as perdas de chaveamento no transistor; isto é, se por um lado o aumento da frequência proporciona um controle melhor da corrente do motor, por outro ele resulta em um aumento das perdas por chaveamento, e chega até a não permitir um controle muito sensível ao *duty-cycle*, devido aos tempos de chaveamento do componente que começam a ser mais significativos se comparados ao período do PWM. O Apêndice C contém os gráficos de *ripple* de corrente para todas as frequências testadas, além da tensão de chaveamento sobre o motor, e o resultado dos experimentos práticos foi que a frequência de 7,8 kHz é a que apresenta melhor relação de controle e perdas - entre as opções disponíveis para ambos os *timers* 1 e 2.

Uma vez definida a frequência, basta configurar os *timers* para gerarem um PWM de 7,8 kHz. No TMR1, de 16 bits, o modo de operação é o 5, *Fast PWM* de 8 bits, portanto contando sempre no sentido crescente, de 0 a 255. O *prescaler* selecionado tem um fator 8, de modo que a frequência de incremento do *timer* é  $16\text{MHz}/8 = 2\text{ MHz}$  (2 milhões de incrementos por segundo), o que resulta em uma frequência de estouro da contagem de  $2\text{MHz}/256 = 7,8\text{ kHz}$ , a frequência do PWM. O comportamento das saídas OC1A e OC1B foi definido como começando a contagem em nível alto, e alterando para nível baixo assim que ela ultrapassar o valor do respectivo registrador OCR1x.

No TMR2, de 8 bits, a configuração de funcionamento é igual à do anterior: modo de operação 3, *Fast PWM*, com contagem de 0 a 255, *prescaler* de fator 8 e comportamento das saídas com início em nível alto.

Aqui vale registrar a resolução de ajuste do *duty-cycle* do sinal PWM de cada um desses *timers*: com 8 bits, portanto 256 valores, para controlar a variação de 0 a 100 % do *duty-cycle*, temos um passo de  $(1/256 \times 100) = 0,39\%$ /incremento.

Quanto ao *Timer 0*, sua configuração também é de *Fast PWM* de 0 a 255, porém seu *prescaler* definido por padrão (depois de um *reset* do microcontrolador) é um fator de 64, de modo que o PWM de saída tem a frequência no valor de 977 Hz, próximo o suficiente do 1 kHz esperado para o acionamento do *Charge pump*. Na verdade, uma vez que o TMR0 é usado para outras funções relacionadas a tempo no código, como o *pulseInLong()*, não é recomendado alterar sua frequência de operação quando pretende-se utilizar esta classe de funções.

O TMR1 e o TMR2 são inicializados com OCRnx = 0, o que corresponde ao valor de  $\delta = 0\%$ , a fim de que nenhum motor seja acionado antes de uma leitura do sinal do rádio controle. O TMR0 é inicializado e mantido com OCR0A = 127, o que corresponde a  $\delta = 50\%$ , a fim de produzir uma onda quadrada fixa na saída, usada para acionamento do circuito de *Charge pump*.

### 3.2.3 Bloco B - Determinação do zero dos canais

Olhando para a Figura 18, com a foto do rádio controle, é possível verificar que as alavancas podem se movimentar vertical e horizontalmente, porém algo não representado na imagem é que, em uma delas, um sistema de molas puxa sempre a alavanca para o centro, e em outra o movimento vertical pode ser deixado em repouso em qualquer ponto de sua excursão - no aeromodelismo, trata-se do canal que controla a potência do motor de propulsão. Para nossa aplicação, é mais prático utilizar a alavanca que volta ao ponto de centro, situação na qual a locomoção deve ficar completamente parada; entretanto, dado que os canais seguem todos a mesma modulação PPM na saída do receptor, quaisquer dois poderiam ser usados neste projeto.

Esta contextualização do parágrafo anterior serve para explicar a próxima etapa do algoritmo: a leitura de "zero" dos canais. É esperado que, no momento que a eletrônica de locomoção do robô é ligada, o rádio controle não esteja enviando nenhum comando de movimento, de modo que a primeira leitura de cada canal do receptor será associada ao centro da alavanca do controle, isto é, à metade da excursão possível para cada um dos canais. É por este motivo que fica mais prático utilizar a alavanca com retorno automático para o centro, uma vez que a outra está susceptível ao erro manual de posicionamento.

No caso do ATmega328P, e utilizando a IDE do Arduino, a função *pulseInLong()* foi a responsável pela leitura do sinal PPM em cada canal, configurada para retornar o valor da duração do pulso em nível alto, em microsegundos, dentro de um limite de espera dado no campo das definições, o chamado *timeout*. Esta mesma função, com as mesmas configurações, é a utilizada para ler o valor dos canais do receptor do rádio controle em

outras ocasiões do código.

A verificação de validade do valor lido analisa apenas se o *timeout* ocorreu, resultando em valor zero para a variável correspondente, ou não. Uma sugestão para a implementação futura é a comparação do valor lido com o valor de referência definido (sugestão de 1,5 ms) para o centro de cada canal, a fim de verificar a coerência e confiabilidade da leitura. Caso o valor lido seja zero, portanto resultante de um *timeout* - causado pelo controle desligado, por exemplo -, uma nova leitura é feita, e assim sucessivamente até que valores válidos sejam lidos e armazenados em variáveis para referência de "centro" de cada canal.

### 3.2.4 Bloco C - Leitura dos canais e adaptação de limites

Uma vez feita a leitura dos valores de zero (ou de centro) de cada canal, passamos ao *loop* principal do código, à rotina que o microcontrolador executará ininterruptamente, começando pela leitura dos canais de comando. Tudo o que será descrito a seguir para um canal acontece também com o outro, na mesma sequência: o pulso em PPM é lido e comparado com os valores de referência, mínimo e máximo, declarados no campo das definições; caso o valor lido extrapole algum valor de referência, mas ainda mantenha-se dentro dos limites definidos, então o valor de referência passa a ser esta quantia lida no canal.

A fim de ilustrar este comportamento, vamos supor que o valor lido para o canal FT (frente e trás) seja de  $1920\ \mu s$  de pulso em nível alto. É útil relembrar os valores de referência adotados inicialmente, de  $1200\ \mu s$  (1,2 ms) para mínimo, e  $1800\ \mu s$  para máximo, e os valores de plausibilidade limite superior de  $2100\ \mu s$ , e inferior de  $900\ \mu s$ . O canal com um pulso de  $1920\ \mu s$  supera o valor de referência máximo ( $1800\ \mu s$ ), e ainda está dentro do limite superior de coerência PPM ( $2100\ \mu s$ ), portanto o valor de referência máximo passa a ser  $1920\ \mu s$ .

O motivo dessa implementação de faixa adaptativa é devido aos valores de máximo e mínimo reais de cada canal do rádio controle poderem variar ligeiramente, e manter um teto ou piso fixos de referência poderia ocasionar em uma "saturação" ou subutilização do canal. A adaptação da faixa de excursão é individual para cada canal de entrada, e completa-se após alguns movimentos do *joystick* do rádio controle.

### 3.2.5 Bloco D - Combinação dos canais e conversão para PWM

Uma vez realizada a etapa anterior, para cada canal temos agora a referência do valor de zero, do valor de extremo máximo e do valor de extremo mínimo - que podem ou não ser simétricos em relação ao centro. Por exemplo, o canal FT pode ter seu valor de centro em  $1450\ \mu s$ , de mínimo em  $1050\ \mu s$ , e de máximo em  $1980\ \mu s$ , portanto uma

excursão de  $530 \mu\text{s}$  para frente, e apenas  $400 \mu\text{s}$  para trás.<sup>6</sup>

Talvez o leitor já tenha reparado que o controle de um motor não está relacionado ao valor de apenas um canal do receptor, e o controle do outro motor apenas ao valor do outro canal, mas é a combinação de ambos que deve ser traduzida para a aplicação de uma potência em cada motor da locomoção. O exemplo mais simples disto é observado quando a alavanca do rádio controle é acionada completamente para frente, situação em que ambos os motores devem ser acionados com potência máxima para que o robô ande para frente, e não apenas um deles.

A tradução da intensidade de acionamento de cada canal em intensidade de acionamento de cada motor ocorre por meio de duas equações de duas variáveis, uma equação para cada motor. Por intensidade de acionamento de canal lê-se valor da largura de pulso (PPM) em relação ao centro, e por intensidade de acionamento de cada motor lê-se valor de PWM aplicado em determinado sentido da ponte H.

Ora, as unidades não estão compatíveis, de um lado temos quantias em microssegundos, e do outro temos valores baseados em quantização de 8 bits. A fim de trazer todos para uma mesma base, decidiu-se pela conversão da largura de pulso para um valor de 8 bits, pois o resultado da equação de combinação dos canais para cada motor já poderia ser utilizado nos registradores que controlam os sinais de PWM.

A conversão lança mão de um recurso interessante e muito útil para a parte matemática: quando um canal é acionado acima de seu valor de zero até o extremo máximo, a transformação gera um valor de 8 bits positivo, de 0 a 256; por outro lado, quando o acionamento ocorre para um valor abaixo da referência do zero até o extremo mínimo, a conversão gera um valor de 8 bits negativo, de 0 a -256. A conta desta operação é feita transpondo proporcionalmente o valor do canal, em sua faixa de centro até o máximo ou o mínimo, para a mesma proporção na faixa de 8 bits, positivo ou negativo segundo a regra citada. Tomando uma situação mais concreta de um canal com  $1000 \mu\text{s}$  de pulso mínimo,  $1500 \mu\text{s}$  de valor de zero, e  $1900 \mu\text{s}$  de pulso máximo, temos: caso a leitura seja de  $1250 \mu\text{s}$ , portanto abaixo do valor de zero, resultado da conversão seria  $\frac{(1250-1500)}{(1000-1500)} \cdot (-256) = -128$ ; em outro caso, se a leitura seja de  $1800 \mu\text{s}$ , acima do valor de zero, a conversão resultaria em  $\frac{1800-1500}{1900-1500} \cdot 256 = 192$ .

Esta conversão é implementada na IDE do Arduino por meio da função *map()*, e sua sintaxe pode ser conferida tanto no código apresentado no Apêndice B, quanto na documentação *online* de referência do Arduino.

Como padrão, definiu-se no canal FT que o valor positivo de 8 bits deve ocorrer quando a locomoção do robô é para frente, e a faixa de valores negativos é usada quando ele deve ir para trás; no canal DE temos algo similar, sendo a faixa positiva destinada ao

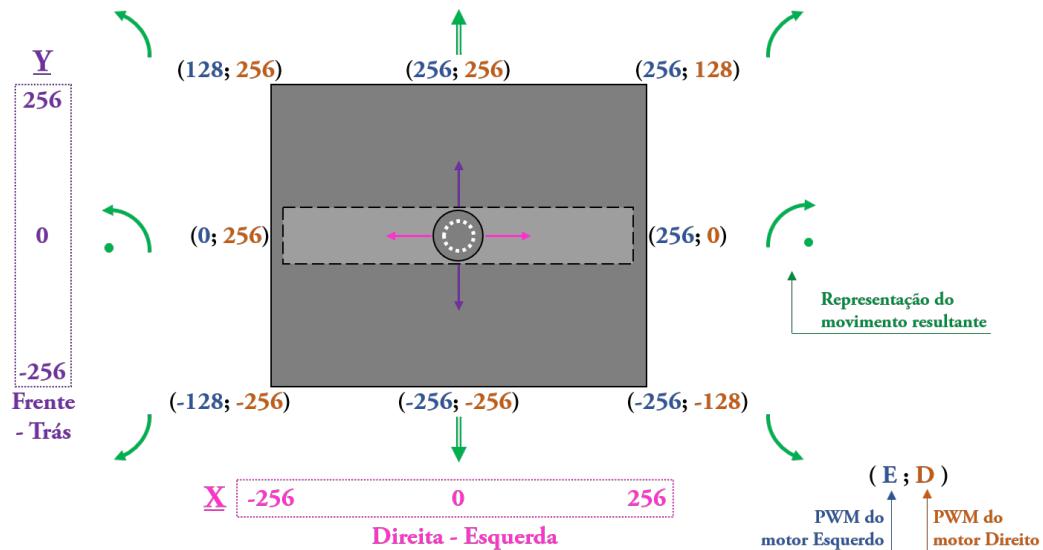
---

<sup>6</sup> Já considerando que o canal FT aumenta seu pulso PPM conforme o robô vai para frente.

movimento para a direita, e a faixa negativa para o movimento à esquerda.

Para auxiliar na visualização dessa conversão de canais, temos na Figura 67 uma representação da alavancagem do rádio controle, cujo movimento está mapeado pelos eixos X, em magenta, e Y, em roxo. O eixo X representa o canal DE depois da conversão, e o canal Y representa o canal FT convertido para valores de 8 bits. Observe que alguma assimetria que existia entre o centro e as referências de extremos dos canais já não existe mais, todos possuem agora a mesma excursão de 256 valores; a faixa anterior de variação só dita, a partir deste momento, a sensibilidade da respectiva faixa convertida, ou seja, uma excursão de  $500 \mu\text{s}$  tem maior precisão de controle sobre os valores em 8 bits quando comparado a uma excursão de apenas  $300 \mu\text{s}$ .

Figura 67: Correlação entre os comandos do rádio controle, o movimento pretendido e o PWM aplicado em cada motor da locomoção.



Fonte: Elaborado pelo autor.

O padrão de definição quanto ao valor em 8 bits ser positivo ou negativo também é aplicável ao PWM dos motores: caso o valor seja positivo, o motor deve girar de modo que o robô se move para frente, e se o valor atribuído ao PWM for negativo, o motor deve ser acionado tal que o robô move-se para trás - qualquer um dos motores. É claro que não existe tal coisa como um PWM negativo, mas adotar esta tratativa facilita a resolução da equação que determina o PWM a partir dos valores dos canais, e no próximo bloco do *software* será o indicativo do sentido que a ponte H, e portanto o motor, devem ser acionados.

Sabendo disso, podemos interpretar completamente a Figura 67: cada extremidade de movimento da alavanca contém um par ordenado, que o detalhe no canto inferior direito da figura indica como sendo os valores de PWM para os motores esquerdo e direito,

respectivamente. Desse modo, tomemos o caso em que a alavanca está no ponto mais baixo do seu movimento vertical: nesta situação o robô deve mover-se para trás com potência total, e assim indicam os valores de PWM dos dois motores, ambos em -256. A posição central não possui um par de valores PWM, mas como dito anteriormente, deve manter ambos os motores parados (PWM = 0).

A Figura 67 contém ainda algumas setas em verde para ilustrar o movimento pretendido para o robô em determinadas posições da alavanca de acionamento. As mais simples são a locomoção para frente e para trás, representadas pelas setas retas para cima e para baixo. Nos vértices temos a representação de uma curva mais aberta, como no vértice superior esquerdo, por exemplo: quando a alavanca vai para esta posição é esperado que o robô faça uma curva aberta para a esquerda, e neste ponto o par ordenado entre parênteses indica que o motor direito está em sua potência máxima (PWM = 256), porém o motor esquerdo tem metade da potência aplicada nele (PWM = 128), o que resulta na esperada curva aberta para a esquerda.

Talvez os movimentos menos óbvios sejam com a alavanca completamente para a direita ou para a esquerda, pois nestas situações poderia se esperar que o robô girasse em torno do próprio eixo, com um motor girando para frente e outro para trás, ambos em potência máxima. A escolha para a implementação foi diferente, decidiu-se manter uma roda completamente parada e a outra girar para frente com torque máximo, de modo que o eixo de rotação passa a ser esta roda parada. Tomando o caso da alavanca completamente para a esquerda, sem deslocamento vertical, por exemplo, o motor esquerdo permanece parado e o direito é acionado com PWM = 256. O motivo da implementação desta forma é a intenção de dar mais controle para o piloto do robô quando pretende fazer uma curva fechada.

Uma vez entendido o sistema de conversão dos canais, e também o mapa de PWM esperado para cada motor em algumas posições da alavanca de controle, passamos à elaboração das equações que combinam as entradas dos dois canais. Vale relembrar que o canal FT, depois da conversão, é designado pela letra Y, e o DE passa a ser identificado pela letra X, ambos com seus valores de -256 a 256.

Pois bem, sabe-se que cada motor, esquerdo (E) e direito (D), têm seu seu PWM determinado em função das variáveis de comando, portanto  $E = f(X, Y)$  e  $D = g(X, Y)$ , e cada um também possui nove valores de referência para determinadas combinações de X e Y, conforme indicado por cada par ordenado na Figura 67. Sendo assim, podemos montar a Equação 3.3 de duas variáveis, com nove coeficientes a serem determinados.

$$E(X, Y) = (a + b.Y + c.Y^2).X^2 + (d + e.Y + f.Y^2).X + (i + g.Y + h.Y^2) \quad (3.3)$$

Mas por que não montamos também uma equação para  $D = g(X, Y)$ ? Se observarmos

o PWM determinado para E e D em cada posição de X e Y, é fácil notar que os valores são iguais em função de Y, mas invertidos em relação a X. Desse modo, montamos a relação indicada em 3.4:

$$D(X, Y) = E(-X, Y) \quad (3.4)$$

Portanto basta resolver o sistema para E(X,Y) e encontrar seus coeficientes, que para D muda-se apenas  $D = f(-X, Y)$ . Para visualizarmos melhor as combinações de entradas X e Y no sistema linear, e os valores de saída E, a Figura 68 exibe com clareza tais combinações.

Figura 68: Relação entre as variáveis X e Y, e a saída E da Equação 3.3.

<u>Y</u>	<u>E</u>		
256	128	256	256
0	0	0	256
-256	-128	-256	-256
<u>X</u>	<u>-256 0 256</u>		

Fonte: Elaborado pelo autor.

Os valores para os coeficientes encontrados estão apresentados a seguir (3.5), escritos todos em base 2. Caso tivéssemos optado por seguir com os valores dos canais em microssegundos, os coeficientes não apareceriam desta forma, e o resultado ainda teria que ser convertido para os 8 bits que serão atribuídos aos registradores que controlam o PWM.

$$a = 2^{-9}; b = -2^{-18}; c = -2^{-25}; d = 2^{-1}; e = 2^{-10}; f = -2^{-17}; g = 2^0; h = 0; i = 0 \quad (3.5)$$

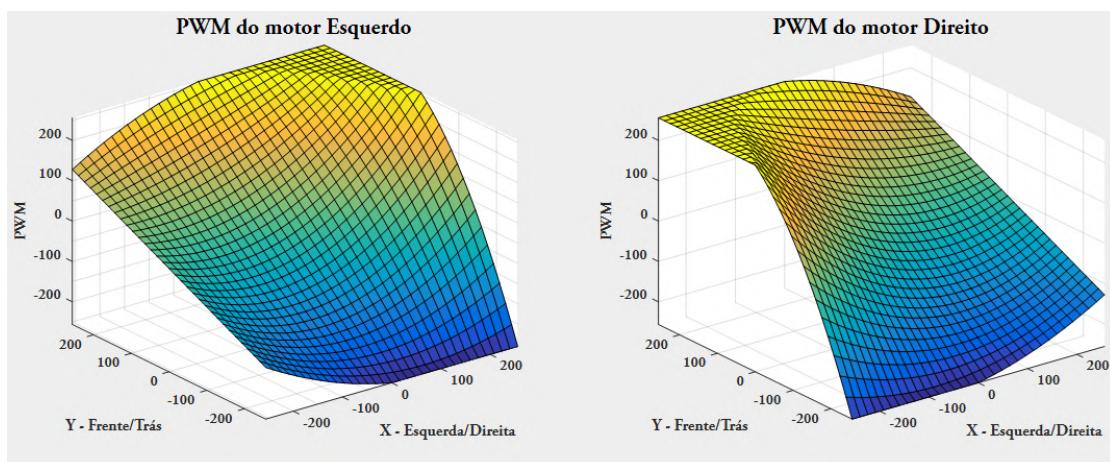
Substituir os coeficientes de 3.5 na Equação 3.3 e resolvê-la em apenas uma linha de código não é recomendado, pois muito do que é calculado aqui pode ser reutilizado para D(X,Y). A saída é quebrar a equação em pedaços menores, e utilizar variáveis auxiliares para armazenar seu resultado. A implementação desta estratégia pode ser verificada no código presente no Apêndice B.

Ademais, nos registradores não é possível atribuir o valor 256 ou superior, uma vez que sua faixa de 256 valores começa no 0 e vai até 255, portanto a etapa final desse cálculo é restringir o valor de E(X,Y) e D(X,Y) para que tenham um valor absoluto máximo de

255. Na IDE do Arduino, esta operação é efetuada pela função *constrain( )*, que substitui pelo valor limite estipulado caso o valor de entrada seja maior (valores absolutos).

A fim de ilustrar o PWM correspondente em qualquer posição da alavanca do rádio controle, a Figura 69 exibe dois gráficos em três dimensões, um para cada motor. Com ele é possível não só confirmar que os valores de PWM de referência estão sendo seguidos para as "coordenadas" de referência, mas também pode-se estimar o movimento do robô para outras posições ainda não mapeadas da posição da alavanca.

Figura 69: Gráficos representativos do PWM resultante da combinação dos dois canais do rádio controle, para cada um dos motores.



Fonte: Elaborado pelo autor.

Uma observação interessante sobre a Figura 69 é a superfície plana nos valores mais altos de PWM, que indicam tanto que a limitação de faixa é importante para que o valor de 255 não fosse excedido, quanto que naquela área toda de posição da alavanca o motor ficará girando para frente em potência máxima, e a curva será feita pela diminuição da potência do outro motor.

### 3.2.6 Bloco E - Aplicação do PWM para cada motor

Os valores de PWM para ambos os motores direito (D) e esquerdo (E) foram calculados e restritos para a faixa de -255 a 255, na qual o movimento para trás do motor corresponde a um valor de PWM negativo, e o giro que resulta em um movimento para frente do robô é aquele designado por um valor positivo de PWM.

Como já comentado na seção das definições, foi estabelecido um limiar de 15 % de *duty-cycle* mínimo para que as saídas e a ponte H sejam acionadas; na base de 8 bits (0 a 255),  $\delta = 15\%$  corresponde ao valor 38. Desse modo, caso o valor absoluto de E seja menor que 38, nenhuma saída PWM é acionada, o são apenas as saídas que controlam os transistores *Low-side*, a fim de colocar os terminais do motor em curto-circuito pelo

GND do circuito, o que constitui o chamado "freio motor" elétrico; o mesmo ocorre caso o valor absoluto de D seja menor que 38, ou qualquer que tenha sido o novo valor definido no cabeçalho do código. Em seguida, o código segue para uma nova leitura dos canais de entrada, conforme indica o fluxograma na Figura 65.

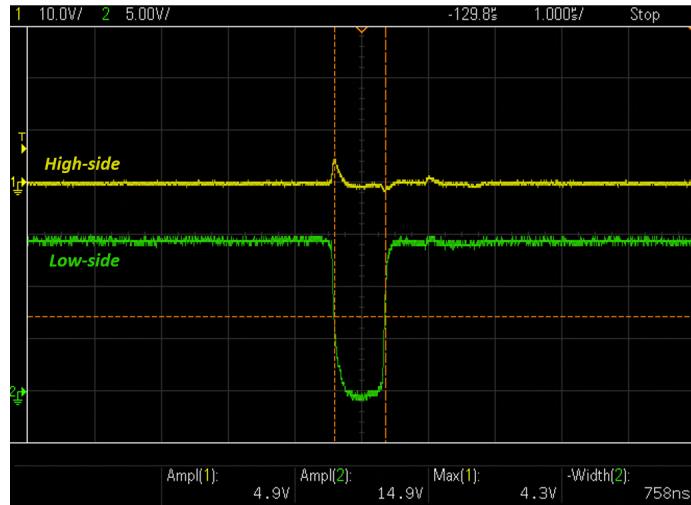
Aqui cabe mencionar uma particularidade do modo *Fast PWM*: mesmo que o registrador OCRnx esteja em zero, como a saída OCnx vai para nível alto no estouro da contagem do *Timer*, ela passa o primeiro período de incremento de contagem neste estado, e só no próximo é atualizada para nível baixo. Com a frequência em 7,8 kHz, um incremento ocorre a cada aproximadamente 500ns ( $\frac{7,8k^{-1}}{256}$ ), portanto as saídas dos OCnx, de acionamento dos transistores *High-side*, ficam aproximadamente 500 ns em nível alto simultaneamente às saídas que acionam os transistores *Low-side*. Se não fosse pela estratégia de *hardware*, com os diodos dos opto-acopladores em anti-paralelo no *Gate driver*, e a implementação do capacitor de *dead-time*, este acionamento simultâneo poderia causar o *shoot through* já mencionado.

O que ocorre na prática, contudo, é o exposto na Figura 70, que mostra a saída dos opto-acopladores do *Gate driver* no acionamento do *High-side*, no canal 1 (amarelo), e do *Low-side*, no canal 2 (verde), de um mesmo ramo da ponte H. No gráfico, observa-se a desativação do *Low-side* esperada pelo acionamento das duas saídas de comando de um mesmo ramo, e a largura do sinal em nível baixo é apenas um pouco maior que o período de 500 ns do incremento do contador, por conta do efeito do capacitor de *dead-time*, e do atraso natural de acionamento entre entrada e saída do opto-acoplador. Quanto à saída do *High-side*, ela nem chega a ficar em nível alto, devido também ao capacitor de *dead-time*, apenas tem um pico curto em 4,9 V que é insuficiente para polarizar o transistor da metade superior da ponte H.

Para o acionamento dos motores, um novo padrão teve de ser definido: quando o motor tiver que girar para frente, os transistores *High-side* do ramo da esquerda (Hs\_A) e *Low-side* do ramo da direita (Ls\_B) devem ser acionados; por outro lado, quando o motor deve girar para trás, os MOSFET's *High-side* do ramo da direita (Hs\_B) e *Low-side* do ramo da esquerda (Ls\_A) dever ser os acionados. Utilizando apenas os mnemônicos estabelecidos para resumir o padrão, e omitindo-se a terceira letra que indica de qual motor se trata, temos a rotação do motor para frente pelo acionamento das saídas Hs\_A e Ls\_B, e quando ele for acionado para trás as saídas Hs\_B e Ls\_A devem ser ativadas.

Com esta padronização estabelecida, vamos supor que E = 100, maior que 38, portanto requer ativação da ponte H. Dado que é um valor positivo, HsE\_A e LsE\_B devem ser acionados, este último sem aplicação PWM, apenas acionado até uma próxima intervenção no sentido do movimento (ou cessação dele), e o primeiro por meio da atribuição do valor de E ao registrador OCR1A; por garantia, tanto OCR1B, que corresponde a HsE\_B, quanto LsE\_A recebem o valor zero, dado que o motor pode estar passando por

Figura 70: Medição do efeito do pulso residual na saída do *Fast PWM* com  $OCRnx = 0$ .



Fonte: Elaborado pelo autor.

uma inversão de movimento.

Outro exemplo pode ser  $D = -200$ : sendo seu valor absoluto maior que 38, o acionamento do PWM é requisitado. Por ser negativo, o sentido de rotação é para trás, assim os pinos acionados são HsD\_B e LsD\_A, enquanto que HsD\_A e LsD\_B devem ficar desativados. O registrador OC2B recebe o valor de D multiplicado por -1, para que seja válido para o byte do registrador, e o pino LsD\_A fica acionado ininterruptamente - até uma próxima inversão de sentido de rotação.

A estratégia de deixar os transistores *Low-side* acionados ininterruptamente, enquanto apenas os *High-side* chaveiam o PWM requisitado, foi implementada para reduzir as perdas de chaveamento de um transistor, e deixar um caminho aberto para a corrente do motor quanto o PWM do *High-side* estiver em nível baixo. Neste caso, a corrente continua fluindo pelo *Low-side* que está conduzindo, e a FCEM gerada pelo motor polariza o diodo intrínseco em antiparalelo no *Low-side* do outro ramo.

Uma vez que a saída de cada sinal de controle dos motores foi atualizada, o código volta à etapa de leitura dos canais do receptor do rádio controle, e o processo todo a partir daí é reiniciado.

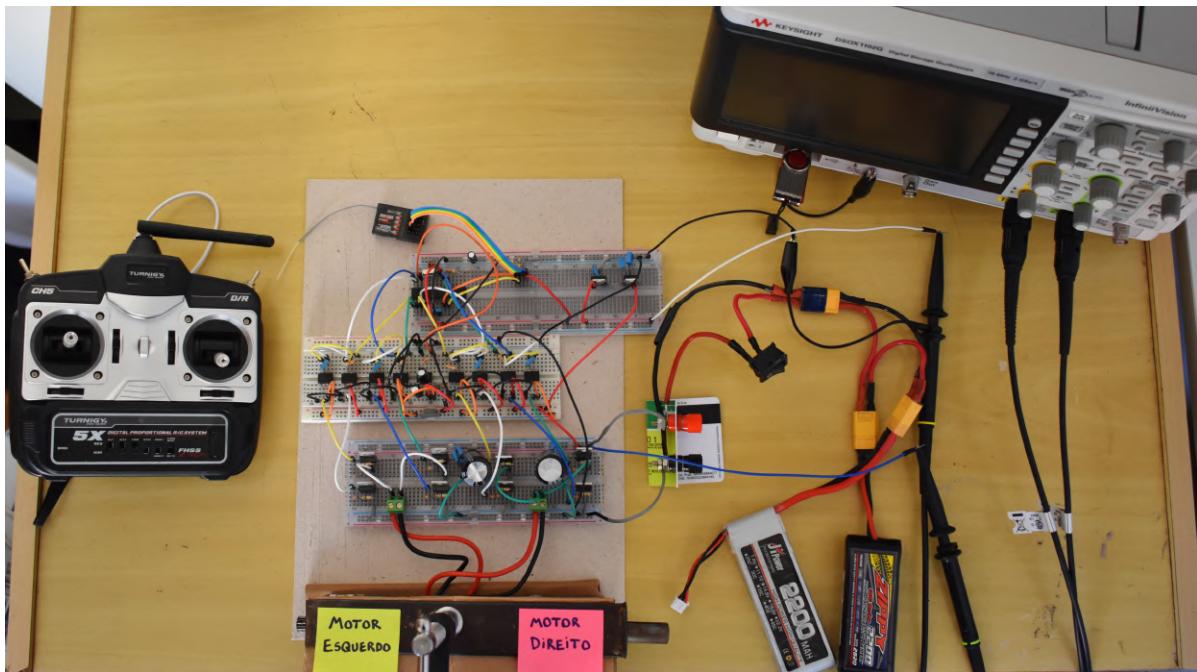
### 3.3 Integração

Uma vez que os módulos de *hardware* já estão projetados e interfaceados entre si (Apêndice A), e o algoritmo de *software* já está codificado (Apêndice B), resta testar a integração de ambos no controle dos motores de locomoção, e levantar uma lista dos modos de falha do dispositivo.

Infelizmente, na data de conclusão deste trabalho, uma placa de circuito impresso (PCB) ainda não havia sido projetada e integrada ao robô, de modo que os testes feitos contam com a utilização de *protoboard*. Além disso, a melhor forma de apresentar o funcionamento do dispositivo seria com um vídeo demonstrativo, o qual não pode ser incluso nesta documentação, por motivos óbvios, mas está disponível no YouTube<sup>7</sup>.

A Figura 71 exibe uma foto da bancada utilizada para o desenvolvimento do projeto e para os testes de integração, com o circuito montado em *protoboard* conectado ao receptor do rádio controle e às baterias em série, e suas duas saídas ligadas aos dois motores de locomoção, presos à bancada. Ainda estão presentes o rádio controle, à esquerda, e o osciloscópio para medições e capturas, à direita, de modelo DSOX1102G, da Keysight.

Figura 71: Bancada de desenvolvimento do projeto e dos testes de integração.



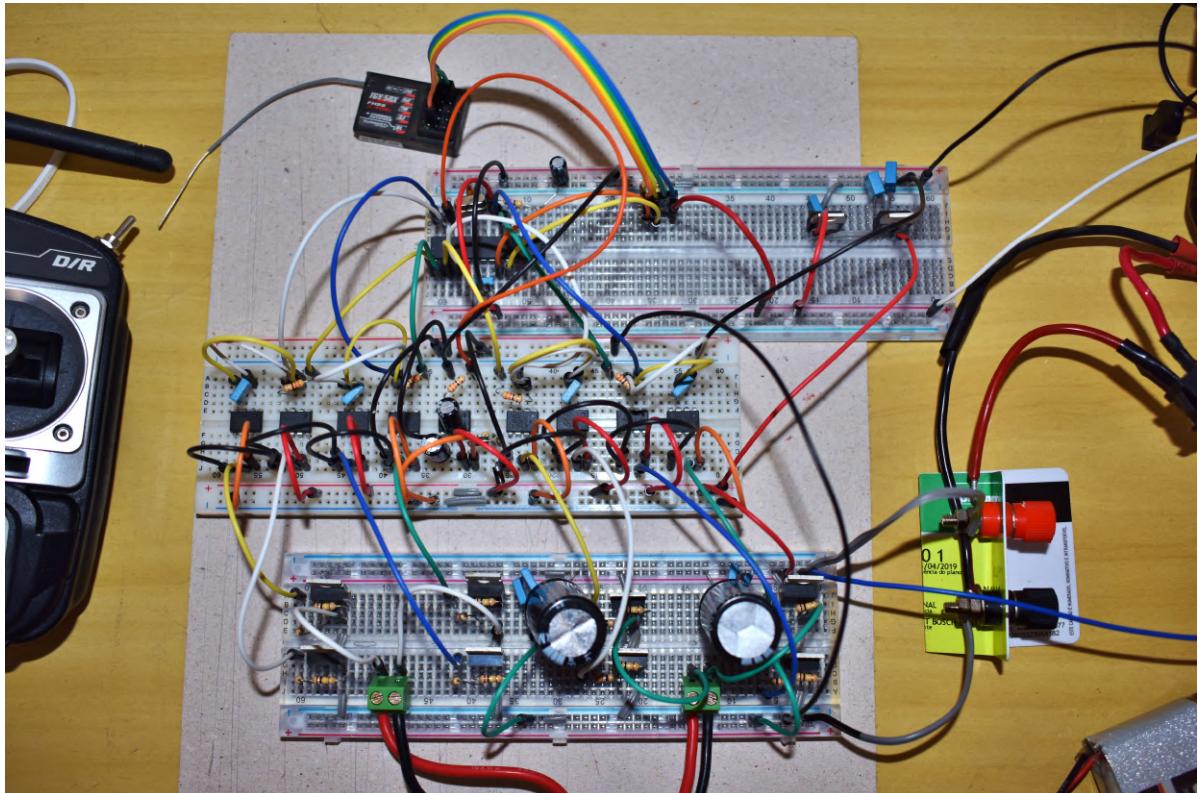
Fonte: Elaborado pelo autor.

A Figura 72 apresenta ampliadas as matrizes de contatos utilizadas na montagem do circuito completo. Na primeira, de cima para baixo, estão os módulos do regulador

<sup>7</sup> Nome do vídeo demonstrativo: "Caio Carrasco TCC exibicao", disponível pelo link: <<https://youtu.be/nJZ4PjlKctk>>.

de tensão, à direita, e do microcontrolador, à esquerda; na *protoboard* do meio estão os módulos do *Charge pump*, ao centro, e do *Gate driver* nas laterais; a matriz inferior abriga o módulo das pontes H.

Figura 72: Circuito completo montado em matrizes de contatos (*protoboards*).



Fonte: Elaborado pelo autor.

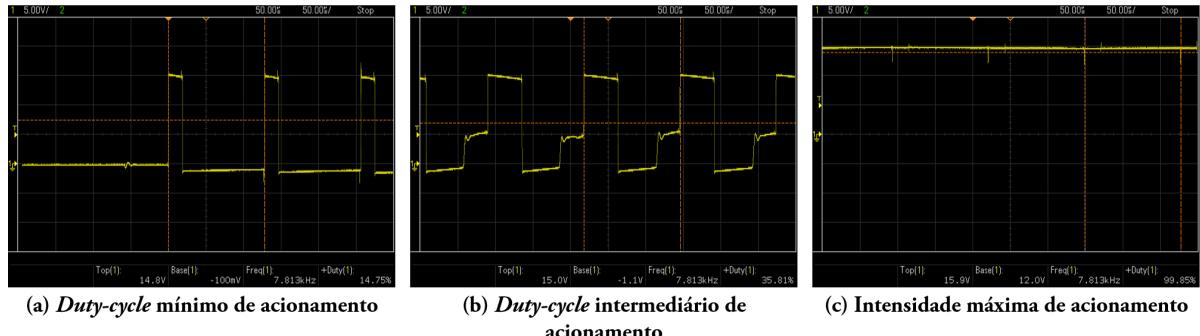
Após a apresentação dos resultados de alguns testes complementares, serão listados os modos de falha do sistema, a fim de que, na futura implementação, eles já sejam conhecidos e talvez até solucionados.

### 3.3.1 Testes de integração

O primeiro teste de integração mostra do resultado da implementação do controle de PWM em um dos motores, demonstrando a tensão resultante do acionamento para frente, na Figura 73, e para trás, na Figura 74, e em três intensidades diferentes: a primeira (a) indica o mínimo PWM aplicado, a segunda (b) corresponde a um comando de intensidade intermediária na alavanca do rádio controle, e o último (c) apresenta a potência máxima aplicada no motor, praticamente a aplicação de  $V_{BAT}$  diretamente. Vale citar que o motor estava em vazio, portanto sem carga em seu eixo no teste em questão, e a bateria não estava completamente carregada.

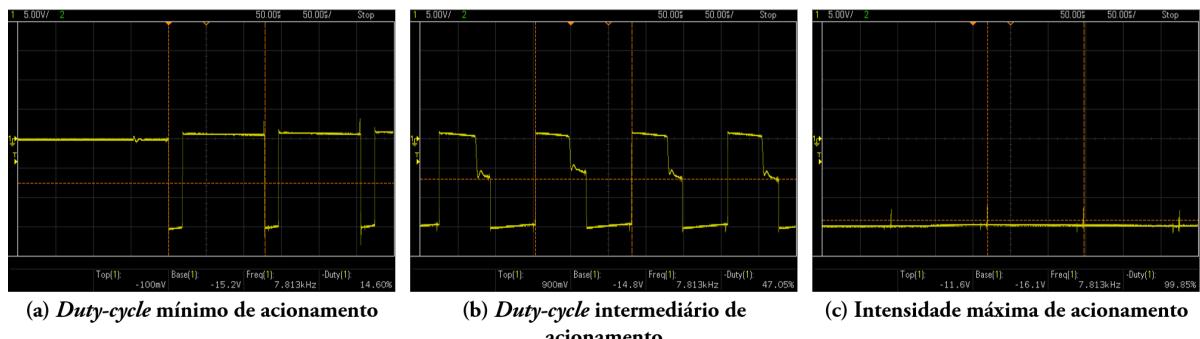
Comparando as Figuras 73 e 74, observa-se que uma é praticamente o espelhamento vertical da outra, ou seja, ambas têm o mesmo comportamento no tempo mas a amplitude de sinal invertida, o que é exatamente o esperado pela simetria de implementação das pontes H e lógica do algoritmo apresentado.

Figura 73: Medições da tensão sobre um motor durante o acionamento para frente em diferentes valores de *duty-cycle*.



Fonte: Elaborado pelo autor.

Figura 74: Medições da tensão sobre um motor durante o acionamento para trás em diferentes valores de *duty-cycle*.



Fonte: Elaborado pelo autor.

A implementação também definiu que nenhum PWM seria aplicado enquanto o valor mínimo de  $\delta = 15\%$  não fosse atingido, e é exatamente isso o que as medições em (a) das Figuras 73 e 74 apresentam: os primeiros pulsos observados no motor com um *duty-cycle* de aproximadamente 15 %.

A medição em (b) mostra o perfil de tensão do motor girando em vazio, em três etapas que se repetem. A primeira etapa consiste na aplicação de  $V_{BAT}$  nos terminais do motor, por um período que corresponde ao *duty-cycle* calculado; no caso da Figura 74, por ter o sentido invertido, é aplicado  $-V_{BAT}$ . A segunda etapa decorre da abertura do

transistor *High-side* pelo fim do pulso PWM naquele período, de modo que a tensão sobre o motor cai a um valor próximo de zero; neste intervalo, o comportamento indutivo do motor se destaca, pois, para que sua corrente possa se "descarregar", uma tensão gerada em seus terminais, de polaridade invertida à do acionamento, atravessa o transistor *Low-side* que já está conduzindo e, por meio da ligação elétrica do GND, polariza diretamente o diodo intrínseco do outro MOSFET *Low-side*, fechando o circuito para a passagem dessa corrente, e de modo que essa tensão próxima de zero observada é praticamente a tensão de polarização direta do diodo intrínseco citado. A última etapa consiste em um patamar de tensão com nível entre 0 V e  $V_{BAT}$ , que começa assim que a corrente do motor vai a zero, e termina com um novo ciclo de acionamento PWM; esse patamar de tensão corresponde à atuação do motor como gerador, uma vez que está em vazio e sua inércia mecânica o mantém girando. Caso o motor estivesse com mais carga mecânica em seu eixo, sua corrente aumentaria e o tempo dessa última fase do perfil diminuiria, podendo até não existir, no caso de o motor estar travado e/ou a corrente subir mais, a ponto de nem se descarregar completamente dentro de um ciclo de PWM.

O último destes acionamentos é a aplicação do *duty-cycle* máximo, apresentada em (c) como praticamente a própria tensão da bateria nos terminais do motor, conforme o sentido de rotação pretendido.

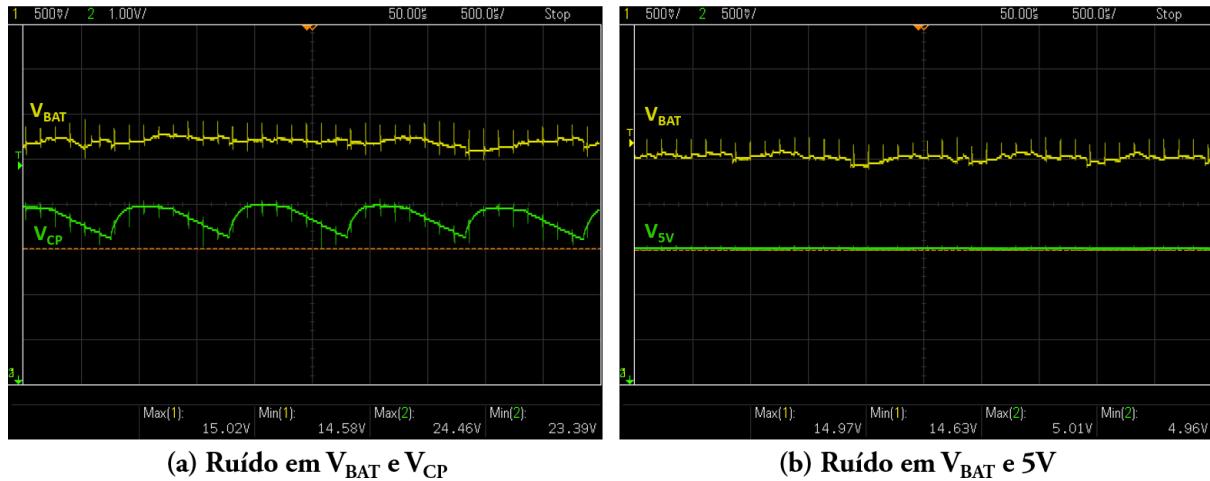
Outro resultado que vale a pena ser apresentado é o ruído presente das tensões constantes do circuito,  $V_{BAT}$ ,  $V_{CP}$  e  $V_{5V}$ , ante o acionamento dos motores em vazio, e com a bateria descarregada. Tais medições estão apresentadas na Figura 75, à esquerda (a) relacionando  $V_{BAT}$  e  $V_{CP}$ , e à direita (b) relacionando  $V_{BAT}$  e  $V_{5V}$ , com a maior resolução de amplitude que foi possível se obter no osciloscópio, e com uma resolução temporal que permite visualizar o comportamento em regime de cada sinal.

É interessante notar que o chaveamento dos motores são a principal causa do ruído nos sinais, gerando os picos de maior amplitude a cada  $125 \mu s$ , aproximadamente, por conta dos 7,8 kHz do PWM. Na tensão de *Charge pump*, à esquerda (a), o *ripple* é causado principalmente pelo consumo dos opto-acopladores, como já verificado anteriormente, e a pequena carga utilizada no acionamento dos MOSFET's apenas gera um certo ruído, porém não é suficiente para causar um distúrbio considerável que coloque em risco o chaveamento dos transistores *High-side*; mesmo os 23,4 V de tensão mínima medida ainda ficam acima do mínimo necessário para chaveamento do máximo  $V_{th}$  do transistor selecionado.

Olhando para a tensão de 5 V, à direita (b), o cenário é ainda mais positivo, com uma tensão completamente estável e sem ruídos percebidos. Tal resultado é esperado, como analisado no desenvolvimento do módulo do regulador de tensão, pois os ruídos começam a ser uma maior preocupação quando a carga do motor é mais intensa, causando o aumento de sua corrente.

Por fim, um último resultado aqui exibido é a energização e os instantes iniciais do

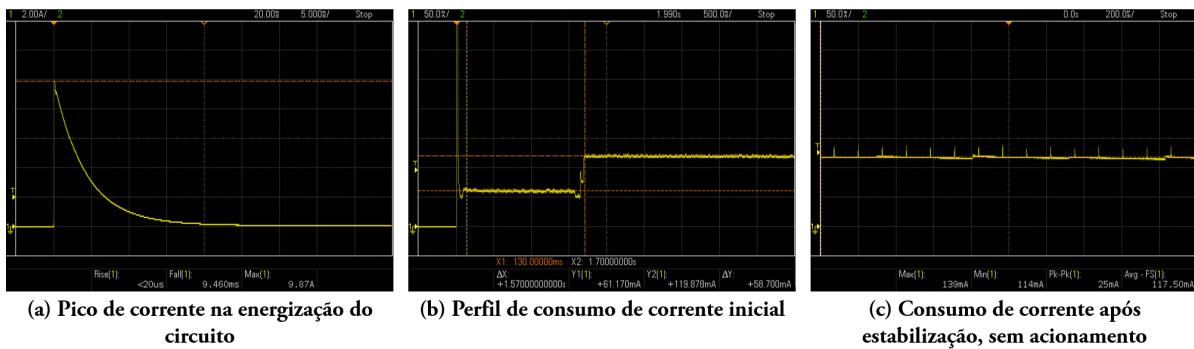
Figura 75: Medição do ruído nas tensões constantes do circuito durante o chaveamento do motor em vazio.



Fonte: Elaborado pelo autor.

consumo de corrente do circuito apresentado. O método de medição foi a utilização de um resistor de  $1 \Omega \pm 5\%$  em série com a associação de baterias de alimentação, e captura da queda de tensão sobre ele, convertida no próprio osciloscópio para a unidade de corrente (conversão  $1 \text{ V} = 1 \text{ A}$ ). Os gráficos obtidos estão expostos na Figura 76 a seguir.

Figura 76: Medição da energização e consumo de corrente do circuito em *stand-by*, sem acionamento de motores.



Fonte: Elaborado pelo autor.

A primeira medição observada (a) indica o pico de corrente, de quase 10 A, que ocorre na energização do circuito, momento em que os capacitores estão descarregados e, por alguns estarem literalmente em paralelo com a tensão da bateria, a limitação de corrente é muito baixa, e que aqui foi aumentada pelo resistor inserido para medição; caso este não estivesse presente, um pico ainda maior poderia ser observado, como é a situação real da aplicação, porém de menor duração.

O segundo gráfico (b) mostra o perfil do consumo de corrente alterando-se nos primeiros segundos de operação: nota-se o pico de corrente da energização estourando a escala de medição, mas logo caindo a um patamar de consumo por volta de 60 mA, o qual dura aproximadamente 1,5 segundo, e depois aumenta para 120 mA, mantendo-se nessa faixa durante a operação posterior.

A terceira figura (c) apresenta o perfil da corrente consumida ainda sem ativação dos motores, mas com o circuito já energizado e em espera (*stand-by*). Apesar dos picos de 25 mA na frequência de 7,8 kHz, decorrente do acionamento residual do *Fast PWM* já comentado, o consumo médio fica na faixa dos 117 mA. Tal valor foi confirmado por medição do mesmo circuito com multímetro, em sua função de amperímetro, cujo visor apresentou consumo de 114 mA, próximo o suficiente para confirmar a média do valor medido.

### 3.3.2 Modos de falha

Por mais que o sistema seja projetado para funcionar corretamente, alguns procedimentos, condições e desgastes podem levar ao comportamento inadequado do dispositivo, falha intermitente ou até dano permanente.

A seguir estão dispostas diversas causas desses modos de falha e suas principais consequências, a fim de preparar melhor o usuário do dispositivo para o caso de um deles ocorrer.

- a) Energização da placa com as alavancas do rádio controle fora do centro
  - Uma das primeiras coisas que o *software* faz quando começa a ser executado é a leitura dos valores de centro das alavancas dos canais que vão comandar a locomoção do robô. Caso uma ou as duas estejam fora do centro durante esta etapa de execução do código, a precisão dos comandos fica prejudicada, pois passa a sofrer de um pequeno desvio; dependendo de onde a alavanca de controle estava no início, ao voltar para o centro, posição na qual o robô deveria ficar parado, o *duty-cycle* mínimo de 15 % pode ter sido superado e ele passe a se mover.
- b) Sobreaquecimento dos opto-acopladores
  - A máxima temperatura do IRF1405 é 175°C, a do ATmega328P e dos reguladores 78xx é 125°C, e a do HCPL-3120 é 100°C, segundo os respectivos *datasheet's*, portanto, no aquecimento da placa, os primeiros componentes passíveis de mau funcionamento são os opto-acopladores, presentes tanto no *Gate driver* quanto no *Charge pump*. Neste último, uma falha poderia derrubar a tensão  $V_{CP}$  pela ausência do chaveamento que recarrega seus capacitores, de modo que os transistores *High-side* não mais teriam tensão suficiente

para serem acionados. Por outro lado, caso algum dos opto-acopladores do *Gate driver* seja danificado, temos duas opções: sua saída fica em aberto, sem possibilidade de acionamento do respectivo transistor, portanto falha de chaveamento de um motor em um sentido; ou, no pior caso, a saída entra em curto com o pino de  $V_{CC}$ , acionando ininterruptamente o MOSFET, independentemente do comando de entrada, causando um curto circuito da bateria ao primeiro acionamento intencional do outro transistor do mesmo ramo, o conhecido *shoot through*.

c) Pico de tensão na tensão de alimentação

- Como visto nos testes dos módulos das pontes H e reguladores de tensão, durante o chaveamento do motor surgem ruídos com picos bastante elevados na tensão de alimentação. Para os reguladores 78xx e os HCPL-3120, esta tensão não deve ultrapassar os 35 V, de acordo com suas respectivas folhas de dados, e tal valor não é alcançado nem com o motor travado, situação de maior consumo de corrente, de acordo com os testes em bancada. Contudo, na dinâmica de utilização do robô durante um combate ou uma partida de hóquei, supondo que  $V_{BAT}$  supere os 35 V, pode ocorrer a queima do opto-acoplador, resultando nos cenários já apresentados no item anterior, e/ou pode ocorrer a queima do primeiro regulador de tensão, LM7812, de modo que a geração de 5 V não é mais possível, o microcontrolador fica desenergizado e o sistema para de funcionar.
- Caso a tensão supere os 55 V máximos de operação do transistor IRF1405, o MOSFET provavelmente se danificará em um dos dois cenários possíveis, em aberto ou em curto, causando no mínimo a perda de chaveamento do motor em um sentido, e no máximo o *shoot through* no seu ramo da ponte H, e possivelmente a queima do seu respectivo opto-acoplador.

d) Ruído de alta frequência na tensão 5 V

- No módulo dos reguladores de tensão, foi observado, no caso mais extremo de acionamento dos motores travados, que algum ruído em  $V_{BAT}$  pode passar pelos filtros implementados e aparecer nos 5 V. Também foi dito que eles não eram suficientes para causar algum mau funcionamento do microcontrolador nem um *reset* acidental, dado que isso não foi observado durante os testes práticos. Contudo, outro dispositivo que utiliza-se dos 5 V gerados no circuito é o receptor do rádio controle, o qual é uma caixa preta em nossa aplicação, ao ponto que podemos assumir que algum ruído em 5 V, decorrente de condições mais extremas de chaveamento dos motores, pode causar interferência em seu circuito interno, e até sua queima. No primeiro caso, é previsível um comportamento dos canais diferente do comandado pela alavanca de comando,

seja uma deformação do pulso ou até a ausência dele como forma de proteção do dispositivo em questão; no outro caso, a queima pode simplesmente desligar o receptor, causando a falha descrita no item a seguir, ou colocar os 5 V em curto, desenergizando o microcontrolador e podendo queimar inclusive o regulador linear LM7805.

- e) Perda do sinal do receptor do rádio controle
  - Caso um ou os dois canais do receptor do rádio controle parem de ser enviados, por uma desconexão elétrica, por exemplo, o último comando lido de cada canal é mantido indefinidamente, ou até ocorrer o *reset* do microcontrolador. Isso ocorre por conta da implementação do algoritmo, que verifica se o sinal lido está dentro da faixa válida para os valores da modulação PPM e, caso não esteja, descarta aquela leitura e não atualiza o valor da variável no código. Desse modo, no caso de uma desconexão elétrica de um dos canais ou queima do receptor do rádio controle, nenhum pulso será lido dentro do intervalo de *timeout* definido, a leitura retorna zero e este não é um valor aceitável para pulso PPM, entrando na condição explicada anteriormente.
- f) Sobreaquecimento dos capacitores eletrolíticos
  - Os capacitores eletrolíticos possuem um eletrólito interno que pode ir ressecando e se degradando quando submetido a altas temperaturas, na faixa de 85 a 100°C, fazendo com que sua capacidade diminua em relação ao valor nominal, e o seu efeito de filtro fique prejudicado, aumentando o nível de ruído do circuito e podendo ocasionar os problemas já citados anteriormente.



## 4 CONCLUSÃO

Retomando a proposta definida nos Objetivos (seção 1.2), pode-se concluir que ela foi executada ao longo deste projeto aqui documentado, uma vez que a versão inicial de um sistema de controle de motores de locomoção foi desenvolvida com algoritmo funcional, o circuito foi capaz de atender às expectativas de controle individual dos motores sem sobreaquecimento, e o sistema como um todo está apto para ser utilizado no sistema de locomoção de um robô de combate ou de hóquei da equipe SEMEAR.

Assim como citado anteriormente, este projeto visa proporcionar uma base teórica e prática - no sentido de que também resulta de testes experimentais - de consulta para aqueles que desejam desenvolver uma eletrônica de locomoção. Tanto o *hardware* quanto o *software* apresentados no desenvolvimento não são os mais robustos que poderia-se alcançar, mas são um bom ponto de partida para se ter um dispositivo funcionando, e a partir dele ir implementando-se as modificações e melhorias.

Aliás, durante o projeto e testes deste dispositivo aqui documentado, diversas ideias de incrementos foram sendo levantadas, porém não implementadas a fim de não aumentar muito a complexidade do trabalho; são elas:

- Adição de supressor de transiente de tensão, o TVS, para absorver algum pico de tensão que pode ocorrer no circuito pelo seu caráter de carga indutiva, e danificar não só os MOSFET's, mas também outros componentes mais sensíveis;
- Projeto de um circuito *snubber*, com o objetivo de diminuir os ruídos de alta frequência causados pelo chaveamento dos transistores;
- Substituição dos reguladores lineares por conversores CC-CC na geração de 5 V, diminuindo o desperdício de carga da bateria e esquentando menos a placa;
- Implementação do *Gate Driver* com outros componentes além do opto-acoplador, utilizando ainda o *Charge pump* para manter o acionamento máximo do motor ( $\delta = 100\%$ ), mas substituindo a interface de acionamento por outro componente, como o HIP4082, por exemplo, por conta do preço elevado do HCPL-3120, a quantidade utilizada e a área demandada na placa (PCB);
- Acrédito de um circuito de monitoramento de corrente para cada motor, incluindo proteções de sobrecorrente e estratégias de acionamento quando o motor estiver travado (pode acontecer no ambiente do combate de robôs);

- No campo do *software*, verificação de coerência do valor de centro do canal para complementar o possível erro de *timeout*, já detectável, conforme sugerido anteriormente na seção 3.2.3 de explicação do Bloco B;
- Implementação de perfis não lineares de controle no *software*, ou seja, a transformação da excursão do canal de linear para exponencial, por exemplo, aumentando a precisão do controle próximo da posição de centro da alavanca;
- Implementação do *fail-safe*, recurso que zera o acionamento dos motores de locomoção no caso da ausência de sinais do receptor do rádio controle;
- Outras melhorias que sejam observadas necessárias diante do uso na prática do dispositivo de controle da locomoção.

Quanto ao custo da placa, citado anteriormente como um dos motivadores para o desenvolvimento desta alternativa, tomando a soma de todos os componentes mais caros (exceto a própria PCB, cujo *layout* não foi projetado), o custo de material foi estimado em cerca de R\$ 180,00<sup>1</sup>, bastante competitivo se comparado com os R\$ 400,00<sup>2</sup> que custaria um produto da mesma categoria, e sem a possibilidade de manutenção por parte da equipe.

A placa de circuito impresso em si não foi desenhada e nem produzida até a conclusão deste trabalho, de modo que os gráficos e os resultados aqui apresentados e discutidos, provenientes da parte experimental, foram medidos em bancada - quando não simulados, conforme indicação - utilizando matrizes de contatos (*protoboards*) e outros mecanismos de prototipação rápida, porém sempre com os componentes reais, indicados ao longo do texto. Todos os módulos foram testados, tanto separadamente quanto depois da integração completa no circuito final (Apêndice A), e o sistema final mostrou-se capaz de controlar o movimento de ambos os motores conforme os comandos do rádio controle, sem atrasos perceptíveis entre comando e resposta, sem sobreaquecimento dos componentes nem falhas de funcionamento do sistema. O dispositivo, contudo, pela ausência de uma placa, não foi montado no robô nem testado em condições reais de uso, que envolve a carga mecânica da locomoção de um robô, de aproximadamente 6 kg, nos eixos dos motores. Este seria um próximo passo sugerido para continuidade deste projeto.

Dado que este é um Trabalho de Conclusão de Curso, vale citar algumas das disciplinas que mais contribuíram para sua elaboração, tanto teóricas quanto práticas. As matérias de "Circuitos Elétricos" e "Circuitos Eletrônicos", evidentemente, foram essenciais para a elaboração do projeto dos módulos de *hardware*, assim como as aulas de "Conversão

<sup>1</sup> Estimando pelos custos, na segunda metade de 2020, dos 8 MOSFET's (R\$ 7,60 cada), 9 opto-acopladores (R\$ 9,50 cada), microcontrolador (R\$ 15,30), 2 capacitores de 1500  $\mu$ F por 63 V (R\$ 6,30 cada) e 2 reguladores de tensão (R\$ 1,50 cada).

<sup>2</sup> Preço de uma Sabertooth 2x12, consultada em <<https://www.robocore.net/loja/drivers-de-motores/controlador-motor-sabertooth-2x12-rc>> em 04/09/2020.

Eletromecânica de Energia" foram fundamentais para entendimento do modelo do motor CC e seu controle. No que se refere ao *software*, a base foi dada principalmente em "Linguagens de Programação e Aplicações", "Introdução à Organização de Computadores" e "Sistemas Digitais", tanto nas aulas expositivas quanto em laboratório. Nomeadas estas, ficam representadas todas as outras disciplinas que compõem o curso de Engenharia Elétrica com ênfase em Eletrônica da EESC, cada uma contribuindo de alguma forma para a formação profissional do autor.

Finalmente, apesar de voltado para o combate e o hóquei de robôs, o dispositivo e o material expostos aqui podem ser utilizados e adaptados para outras aplicações, conforme necessidade e conhecimento técnico do leitor.



## REFERÊNCIAS

- BASILIO, J. C.; MOREIRA, M. V. Experimentos para estimação dos parâmetros de motores de corrente contínua. **Cobenge**, Porto Alegre, Rio Grande do Sul, Brasil, 2001. Disponível em: <<http://www.dee.ufrj.br/controleAutomatico/artigos/cobenge2001.pdf>>. Acesso em: 23 de maio de 2020.
- COLTON, S. Simple modular half-bridge. p. 16, 2009. Disponível em: <[http://web.mit.edu/first/kart/controller\\_rev1.pdf](http://web.mit.edu/first/kart/controller_rev1.pdf)>. Acesso em: 23 de agosto de 2020.
- GASPAR, L. S. **Plataforma para medição de velocidade em motor CC sem o uso de sensor acoplado ao eixo**. 2011. Trabalho de Conclusão de Curso — EESC - USP, São Carlos, 2011. Disponível em: <<http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-30032012-095415/?&lang=br>>. Acesso em: 24 de maio de 2020.
- MCPHERSON, G. **An introduction to electrical machines and transformers**. Wiley, Nova Iorque: [s.n.], 1981.
- MEGGIOLARO, M. A. Tutorial em robôs de combate. p. 190, 2006. Disponível em: <<https://www.riobotz.com/tutorial-riobotz>>. Acesso em: 4 de novembro de 2020.
- \_\_\_\_\_. Riobotz combot tutorial. p. 367, 2009. Disponível em: <<https://www.riobotz.com/riobotz-combot-tutorial>>. Acesso em: 4 de novembro de 2020.
- MOTLAGH, N. H. **Advanced Trends in Wireless Communication**. Finlândia: IntechOpen, 2011. Capítulo 10: Frequency Hopping Spread Spectrum: An Effective Way to Improve Wireless Communication Performance.
- MOURA, M. A. S. **Modelação e Identificação de Motor DC**. 2014. Dissertação (Mestrado em Engenharia Eletrotécnica de Computadores) — Instituto Superior de Engenharia do Porto, Porto, Portugal, 2014.
- SOUZA, F. Timers do atmega328 no arduino. **Embarcados**, 2015. Disponível em: <<https://www.embarcados.com.br/timers-do-atmega328-no-arduino/>>. Acesso em: 15 de junho de 2020.
- SUSTEK, M. et al. DC motors and servo-motors controlled by Raspberry Pi 2B. **CSCC**, 2017. Disponível em: <[https://www.researchgate.net/publication/320204209\\_DC\\_motors\\_and\\_servo-motors\\_controlled\\_by\\_Raspberry\\_Pi\\_2B](https://www.researchgate.net/publication/320204209_DC_motors_and_servo-motors_controlled_by_Raspberry_Pi_2B)>. Acesso em: 23 de maio de 2020.
- VERONESE, P. R. **MOSFET - Resumo da Teoria**. São Carlos, 2016. 29 p.
- VUKOSAVIC, S. N. **Electrical Machines**. [S.l.: s.n.], 2013.
- WOLLINGER, L. Baterias de lítio-Íon: Um guia completo. **Embarcados**, 2020. Disponível em: <<https://www.embarcados.com.br/baterias-de-litio-ion-um-guia-completo/>>. Acesso em: 22 de junho de 2020.

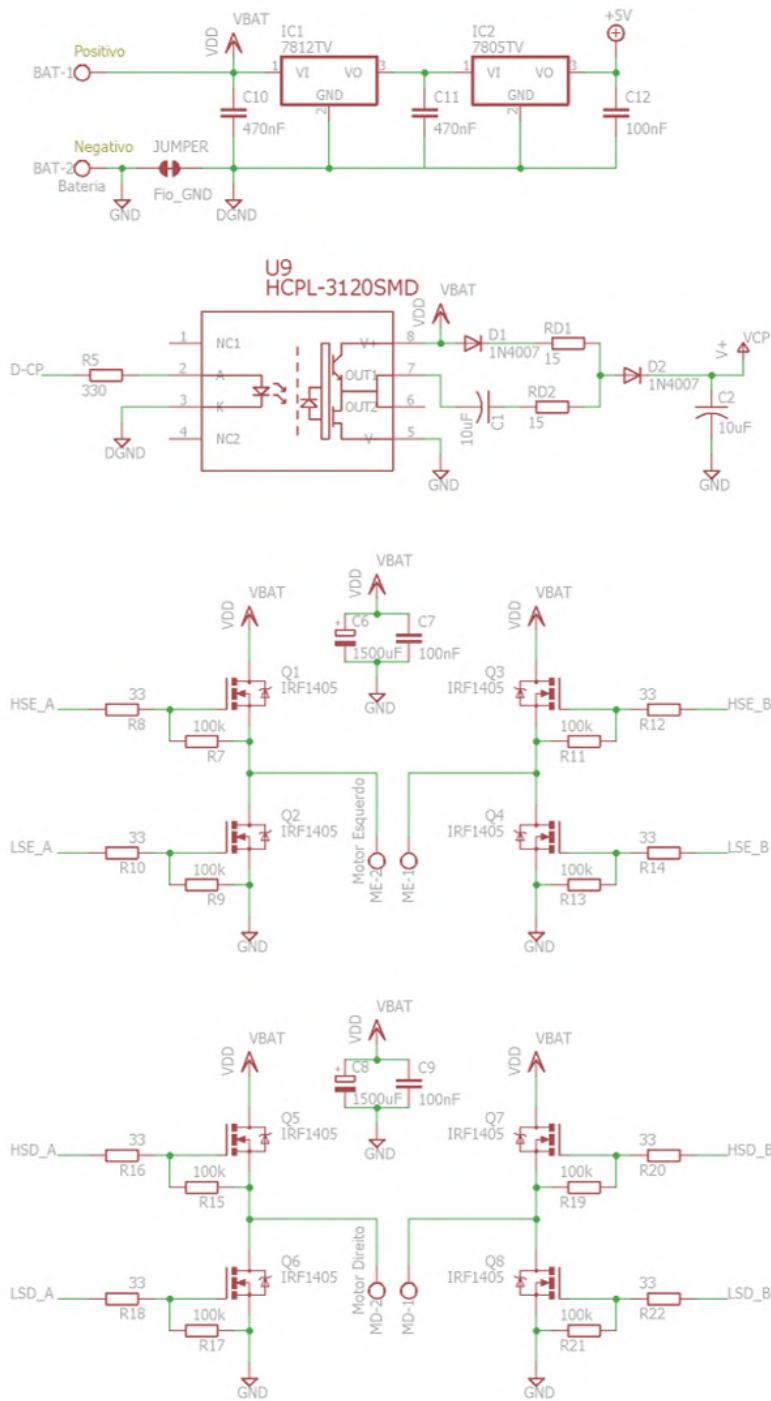


## **Apêndices**



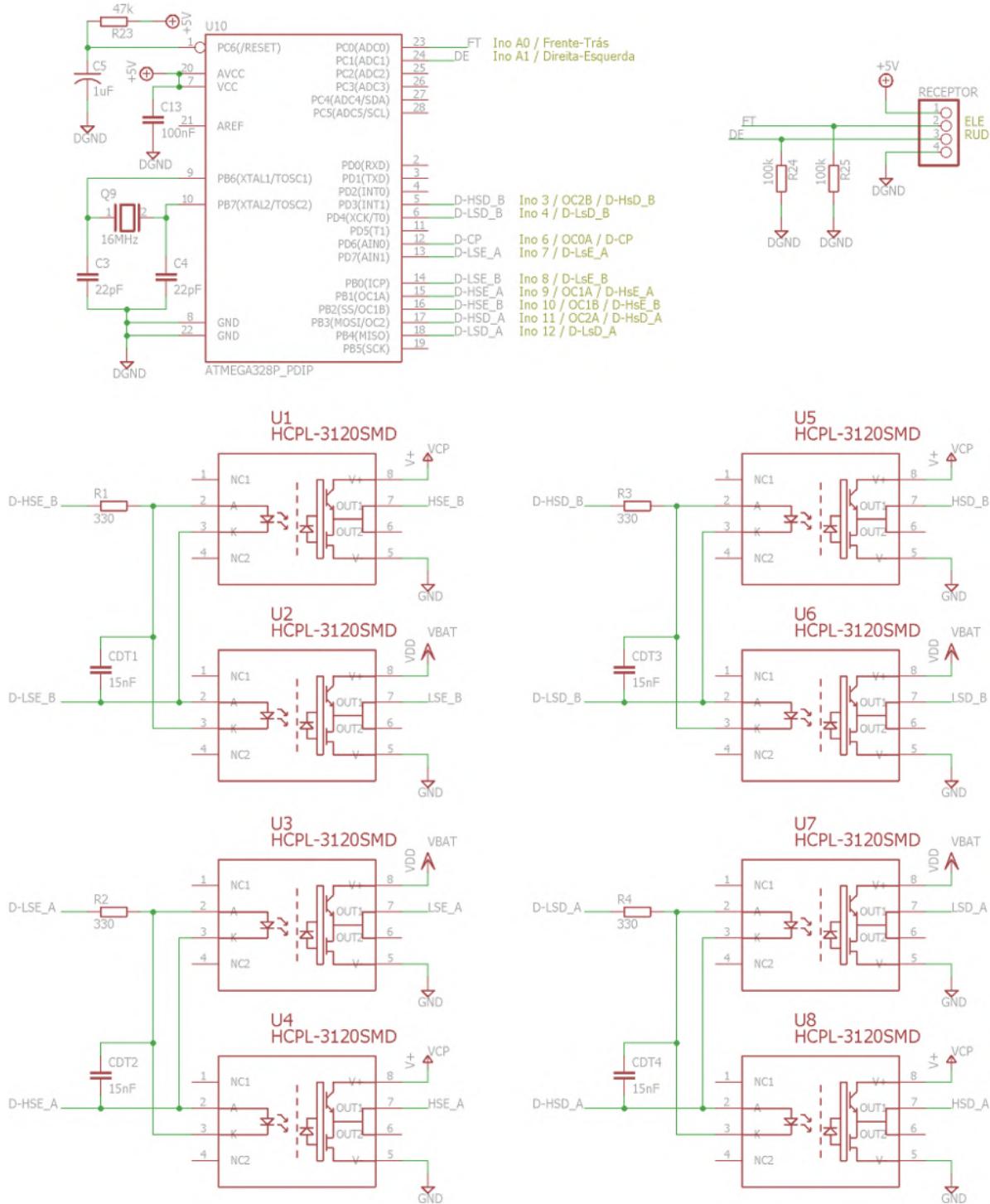
## APÊNDICE A – CIRCUITO COMPLETO

Figura 77: Circuito completo do dispositivo - parte 1/2.



Fonte: Elaborado pelo autor.

Figura 78: Circuito completo do dispositivo - parte 2/2.



Fonte: Elaborado pelo autor.

## APÊNDICE B – CODIFICAÇÃO DO SOFTWARE

A seguir está apresentado na íntegra o código a ser gravado no ATmega328P por meio da IDE do Arduino.

```

1  /*
2   - SOFTWARE ELABORADO PARA CONTROLE DOS MOTORES DE LOCOMOCÃO DE UM ROBO DE DUAS RODAS
3   - TCC: Caio Carrasco
4   - Controle linear: não damos ganho não-linear em nenhum canal
5   - Medição dos pulsos PPM com ajuste automático de range
6   - ao começar, o código le qual é o zero de cada canal,
7   - ou seja, nem pra frente nem pra trás, nem pra direita, nem pra esquerda;
8   - referências: MIN = 1.2ms, MAX = 1.8ms, MEIO = 1.5ms
9   - Limites para os pulsos do PPM: MIN = 900us, MAX = 2100us
10  - Aquisição do pulso e conversão para 256, sem normalização
11  - nem ajuste não linear de ganho
12 - Charge pump implementado com 1kHz e 50% d.c.
13 - Limiar de duty-cycle mínimo para acionamento do motor: 15%
14 */
15
16 #define FT AO // ELE normal
17 #define DE A1 // RUD invertido
18 #define PPM_REF_MAX 1800 // 1.8ms |
19 #define PPM_REF_MIN 1200 // 1.2ms | valores de referência para o controle inicial
20 #define PPM_REF_MED 1500 // 1.5ms |
21 #define PPM_LIM_MAX 2100 // limite máximo de um pulso válido da modulação PPM = 2100us
22 #define PPM_LIM_MIN 900 // limite mínimo de um pulso válido da modulação PPM = 900us
23 #define TO 30000 // timeout do pulseInLong de 30000us = 30ms
24
25 #define CP_PWM 6 // OC0A
26
27 #define HsE_A 9 // OC1A
28 #define HsE_B 10 // OC1B
29 #define LsE_A 7 // Pino digital do Low-side do ramo A, fechado junto com Hs1_B (OCR1B)
30 #define LsE_B 8 // Pino digital do Low-side do ramo B, fechado junto com Hs1_A (OCR1A)
31 // Timer1 = Motor da esquerda
32 //      OCR1A = positivo | OCR1B = negativo
33
34 #define HsD_A 11 // OC2A
35 #define HsD_B 3 // OC2B
36 #define LsD_A 12 // Pino digital do Low-side do ramo A, fechado junto com Hs2_B (OCR2B)
37 #define LsD_B 4 // Pino digital do Low-side do ramo B, fechado junto com Hs2_A (OCR2A)
38 // Timer2 = Motor da direita
39 //      OCR2A = positivo | OCR2B = negativo
40
41 #define LIMIAR_DC 38 // mínimo duty-cycle para acionar a saída (evitar ruídos) = 15%
42
43 void setup() {
44     // Definição de saída dos pinos
45     pinMode(HsE_A, OUTPUT);
46     pinMode(HsE_B, OUTPUT);
47     pinMode(HsD_A, OUTPUT);
48     pinMode(HsD_B, OUTPUT);
49     pinMode(LsE_A, OUTPUT);
50     pinMode(LsE_B, OUTPUT);
51     pinMode(LsD_A, OUTPUT);

```

```

52 pinMode(LsD_B, OUTPUT);
53 pinMode(CP_PWM, OUTPUT);
54
55 // PWM do Charge Pump
56 // Configuracao do Timer 0 – onda quadrada para o Charge pump no pino 6 / OC0A
57 // FastPWM ate 0xFF | High -> Low | Frequencia = 977Hz
58 TCCROA = (1 << COM0A1) | (0 << COM0A0) | (1 << WGM01) | (1 << WGM00);
59 TCCR0B = (0 << CS02) | (1 << CS01) | (1 << CS00) | (0 << WGM02);
60 OCR0A = 127; // duty-cycle de 50% fixo
61
62 // Configuracao do Timer 1 – motor esquerdo
63 // FastPWM ate TOP=0xFF (8 bits) | Clock = 7.81kHz | Saidas High -> Low
64 TCCR1A = (1<<COM1A1) | (0<<COM1A0) | (1<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (1<<WGM10);
65 TCCR1B = (0 << CS12) | (1 << CS11) | (0 << CS10) | (0 << WGM13) | (1 << WGM12);
66 OCR1A = 0; // sem PWM inicial
67 OCR1B = 0;
68
69 // Configuracao do Timer 2
70 // FastPWM ate 0xFF | Clock = 7.81kHz | Saidas High -> Low
71 TCCR2A = (1<<COM2A1) | (0<<COM2A0) | (1<<COM2B1) | (0<<COM2B0) | (1<<WGM21) | (1<<WGM20);
72 TCCR2B = (0 << CS22) | (1 << CS21) | (0 << CS20) | (0 << WGM22);
73 OCR2A = 0; // sem PWM inicial
74 OCR2B = 0;
75 }
76
77
78 void loop() {
79     // variaveis do PPM
80     unsigned long pulso_ft_max=PPM_REF_MAX, pulso_ft_min=PPM_REF_MIN, pulso_ft_med=PPM_REF_MED;
81     unsigned long pulso_de_max=PPM_REF_MAX, pulso_de_min=PPM_REF_MIN, pulso_de_med=PPM_REF_MED;
82     unsigned long pulso_ft = PPM_REF_MED, pulso_de = PPM_REF_MED;
83     // variaveis de saida
84     int dc_ft = 0, dc_de = 0;
85     double x = 0.0000, y = 0.0000, dc_e = 0.0000, dc_d = 0.0000;
86     double aux1 = 0.0000, aux2 = 0.0000, Vx = 0.0000, Vy1 = 0.0000, Vy2 = 0.0000;
87
88
89     // determinacao dos pontos de zero de cada canal
90     pulso_ft_med = pulseInLong(FT, HIGH, TO);
91     pulso_de_med = pulseInLong(DE, HIGH, TO);
92
93
94     while((pulso_ft_med != 0) && (pulso_de_med != 0)){
95         // aquisicao dos pulsos de comando
96         pulso_ft = pulseInLong(FT, HIGH, TO);
97         pulso_de = pulseInLong(DE, HIGH, TO);
98         // ajuste de escala maxima
99         if(pulso_ft > pulso_ft_max && pulso_ft < PPM_LIM_MAX)
100             pulso_ft_max = pulso_ft;
101         if(pulso_de > pulso_de_max && pulso_de < PPM_LIM_MAX)
102             pulso_de_max = pulso_de;
103         // ajuste de escala minima
104         if(pulso_ft < pulso_ft_min && pulso_ft > PPM_LIM_MIN)
105             pulso_ft_min = pulso_ft;
106         if(pulso_de < pulso_de_min && pulso_de > PPM_LIM_MIN)
107             pulso_de_min = pulso_de;
108
109         // — conversao para duty-cycle
110         // Frente = positivo | Tras = negativo

```

```

112     if(pulso_ft >= pulso_ft_med)
113         dc_ft = (double) map(pulso_ft, pulso_ft_med, pulso_ft_max, 0, 256);
114     else
115         dc_ft = (double) map(pulso_ft, pulso_ft_med, pulso_ft_min, 0, -256);
116     // Direita = positivo | Esquerda = negativo
117     if(pulso_de >= pulso_de_med)
118         dc_de = (double) map(pulso_de, pulso_de_med, pulso_de_max, 0, 256);
119     else
120         dc_de = (double) map(pulso_de, pulso_de_med, pulso_de_min, 0, -256);
121
122
123     // — Complexidade intermediaria (curva fechada quando vira tudo)
124     // E = [128 | 256 | 256; 0 | 0 | 256; -128 | -256 | -256]
125
126     // Troca de variaveis para o calculo – nao necessaria, mas deixa mais legivel
127     x = (double) dc_de;
128     y = (double) dc_ft;
129
130     // Variaveis recorrentes
131     Vx = x / 16.0000;
132     Vx = Vx * Vx;
133
134     Vy1 = y / 1024.0000;
135
136     Vy2 = y / 256.0000;
137     Vy2 = Vy2 * Vy2;
138
139     // Componentes da equacao principal
140     aux1 = 1.0000 - Vy2;
141     aux2 = aux1;    // aproveitando a conta ja feita
142     aux1 = aux1 - 2.0000*Vy1;
143     aux1 = aux1 * Vx/2.0000;
144     aux1 = aux1 + y;
145
146     aux2 = aux2 + 2.0000*Vy1;
147     aux2 = aux2 * x/2.0000;
148
149
150     // Combinacao dos componentes na equacao principal
151     dc_e = (int) (aux1 + aux2);
152     dc_d = (int) (aux1 - aux2);
153
154
155     // Garantindo que eles estao dentro da faixa dos registradores
156     // (ainda a ser convertido no negativo)
157     dc_e = constrain(dc_e, -255, 255);
158     dc_d = constrain(dc_d, -255, 255);
159
160
161     // Comando para os PWMs
162     // Motor Esquerdo
163     if( abs(dc_e) <= LIMIAR_DC){
164         OCRIA = 0; // nenhum comando de PWM acionado
165         OCRIB = 0;
166         digitalWrite(LsE_A, HIGH); // todos os Low-side acionados
167         digitalWrite(LsE_B, HIGH); // assim o motor fica em curto
168     } else if(dc_e > 0){ // motor deve ser acionado por Hs_A e Ls_B
169         OCRIB = 0;
170         digitalWrite(LsE_A, LOW);
171         digitalWrite(LsE_B, HIGH);

```

```
172     OCRIA = dc_e;
173 } else { // motor deve ser acionado por Hs_B e Ls_A
174     OCRIA = 0;
175     digitalWrite(LsE_B, LOW);
176     digitalWrite(LsE_A, HIGH);
177     OCRIB = -1 * dc_e; // o valor de dc_e esta negativo, precisa ser invertido
178 }
179
180 // Motor Direito
181 if( abs(dc_d) <= LIMIAR_DC){
182     OCR2A = 0; // nenhum comando de PWM acionado
183     OCR2B = 0;
184     digitalWrite(LsD_A, HIGH); // todos os Low-side acionados
185     digitalWrite(LsD_B, HIGH); // assim o motor fica em curto pelo GND
186 } else if(dc_d > 0){ // motor deve ser acionado por Hs_A e Ls_B
187     OCR2B = 0;
188     digitalWrite(LsD_A, LOW);
189     digitalWrite(LsD_B, HIGH);
190     OCR2A = dc_d;
191 } else { // motor deve ser acionado por Hs_B e Ls_A
192     OCR2A = 0;
193     digitalWrite(LsD_B, LOW);
194     digitalWrite(LsD_A, HIGH);
195     OCR2B = -1 * dc_d; // o valor de dc_d esta negativo, precisa ser invertido
196 }
197
198 }
199
200 }
```

## APÊNDICE C – FREQUÊNCIAS DE PWM TESTADAS NO ACIONAMENTO DO MOTOR CC

Como já citado no embasamento teórico, existe um compromisso entre as perdas de chaveamento nos transistores, o controle de corrente do motor, e até a sensibilidade à variação do *duty-cycle* quando pretende-se escolher a frequência de PWM para se controlar um motor CC. Este apêndice mostra algumas frequências exploradas para o PWM dos motores de combate, na situação de partida de um dos motores utilizados no combate de robôs (o mesmo utilizado ao longo de todo o projeto), mas sem nenhuma carga em seu eixo - a carga inicial é a própria inércia mecânica do eixo e seus atritos.

As frequências foram inicialmente três, a fim de compararmos os impactos da escolha: 400 Hz, 4 kHz e 40 kHz. Depois de analisar os resultados de aquecimento dos componentes com 40 kHz, e um controle muito pobre da corrente com 400 Hz, mais duas frequências foram testadas: 20 kHz e 7,8 kHz. Como em 20 kHz ainda notava-se um certo aquecimento dos componentes, mesmo sem carga, e uma sensibilidade menor à variação do *duty-cycle* por conta de os tempos de chaveamento do transistor já significativos perante o período de PWM, optamos pela frequência de 7,8 kHz para acionamento do motor.

Em todas as capturas de tela a seguir estão presentes as curvas do acionamento com  $\delta = 25\%$  e  $\delta = 75\%$ , tanto no início do acionamento, com a carga mecânica citada e o pico de corrente que isso causa, quanto depois de um certo período em que o motor já está em regime. Em todas as figuras, a curva em amarelo representa o comportamento da corrente, e a curva em verde representa a tensão sobre o motor.

- Na Figura 79 temos  $f = 400$  Hz, e fica visível que a corrente chega a zerar em alguns momentos, tem um comportamento ruidoso e o controle de torque fica muito precário nesta situação.

- Na Figura 80 temos  $f = 4$  kHz, a corrente tem um *ripple* relativamente alto, porém ela não chega a zerar; quanto aos transistores, também não mostraram aquecimento nenhum nestas condições.

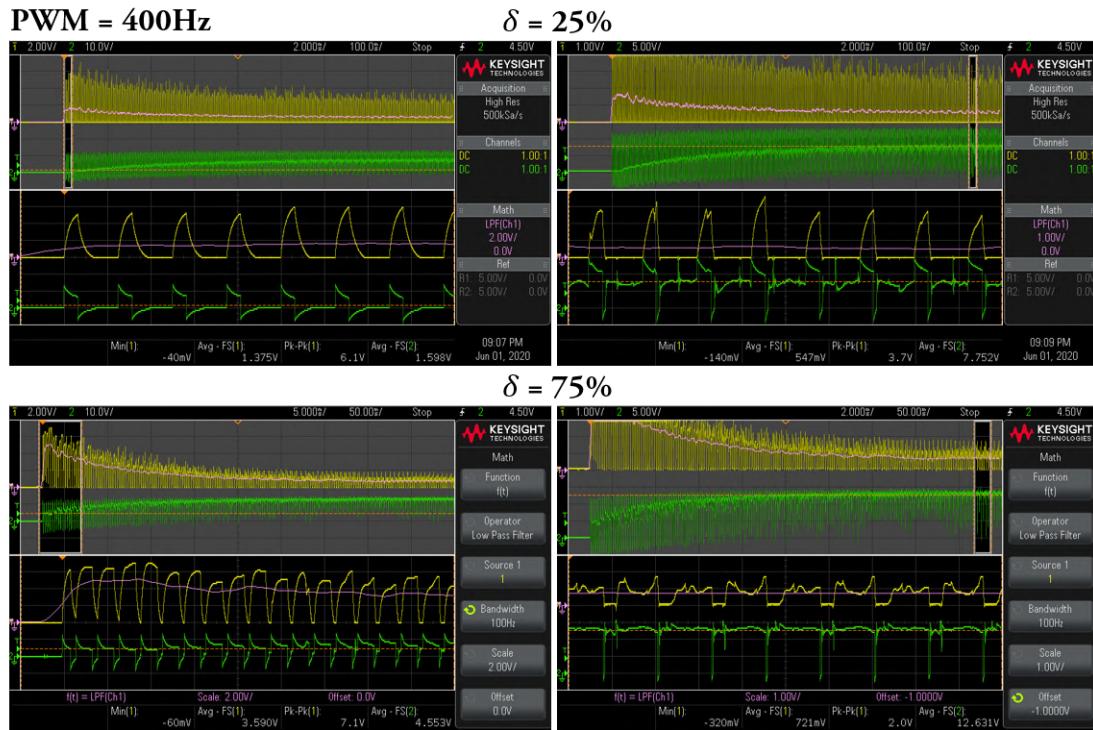
- Na Figura 81 temos  $f = 40$  kHz, frequência na qual não se observa qualquer oscilação de corrente, o que significa um controle mais preciso do torque, porém perde-se precisão no controle de *duty-cycle* pois o chaveamento do transistor começa a ser afetado, como podemos ver pelas bordas arredondadas de subida e descida da tensão (em verde); além disso, os MOSFET's esquentaram bastante já nesta situação de motor sem carga, que é a de menor "estresse".

- Na Figura 82 temos  $f = 20$  kHz, onde observa-se um *ripple* bastante suave na corrente, porém ainda fica visível que é uma frequência muito alta quando nota-se o

comportamento da tensão, cujos sinais não estão compatíveis com os *duty-cycles* aplicados; nesta frequência também observou-se um aquecimento dos componentes, o que não seria indicado para esta situação sem carga.

- Na Figura 83 temos  $f = 7,8$  kHz, a corrente com um *ripple* significativo porém não exagerado, fica em torno de um valor médio, a tensão de controle corresponde bem ao *duty-cycle* aplicado, e os componentes não esquentaram durante o teste, portanto a situação de equilíbrio entre ganhos e perdas.

Figura 79: Acionamento do motor com PWM de 400 Hz.



Fonte: Elaborado pelo autor.

Figura 80: Acionamento do motor com PWM de 4 kHz.

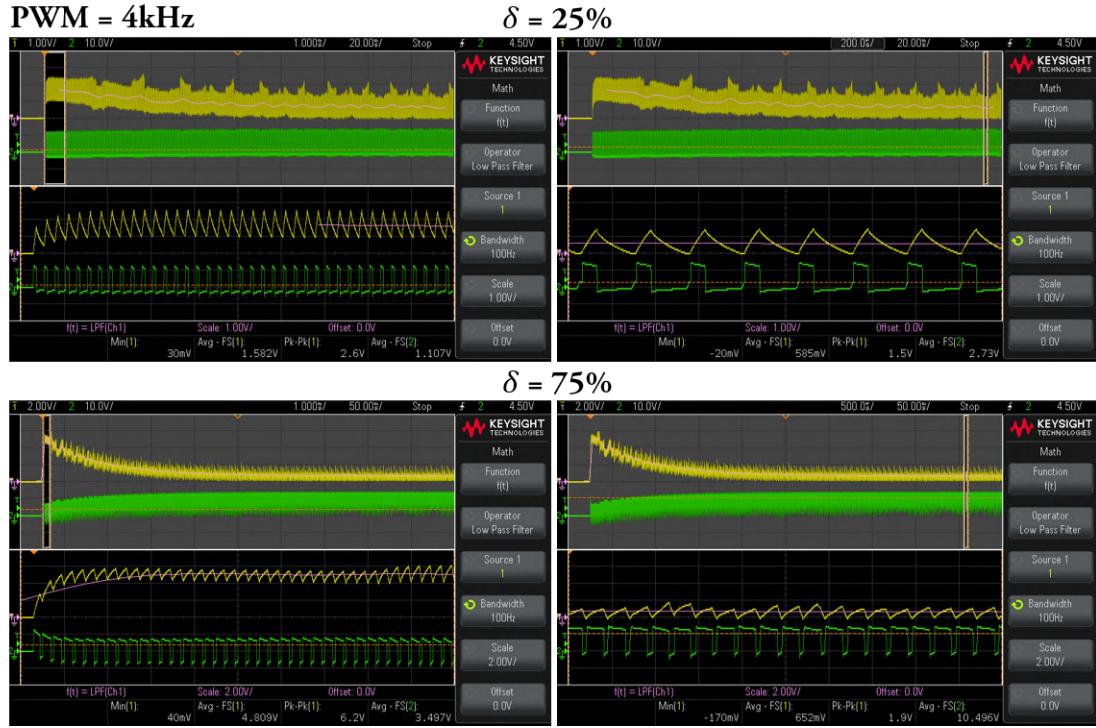


Figura 81: Acionamento do motor com PWM de 40 kHz.

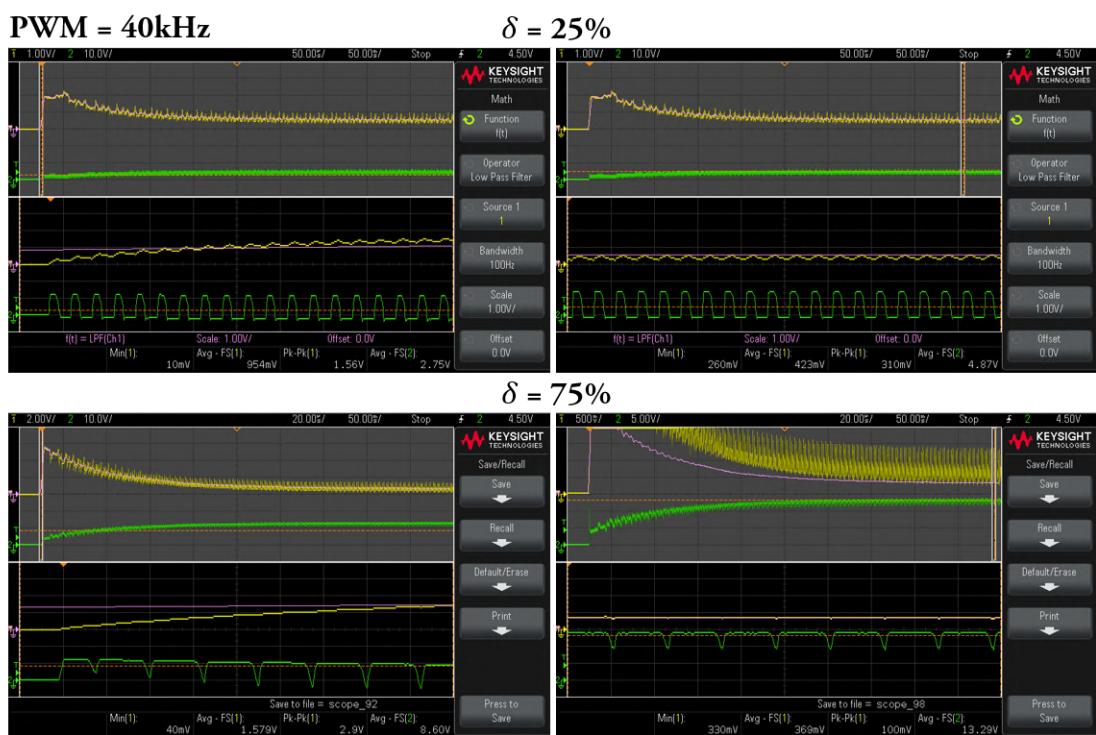
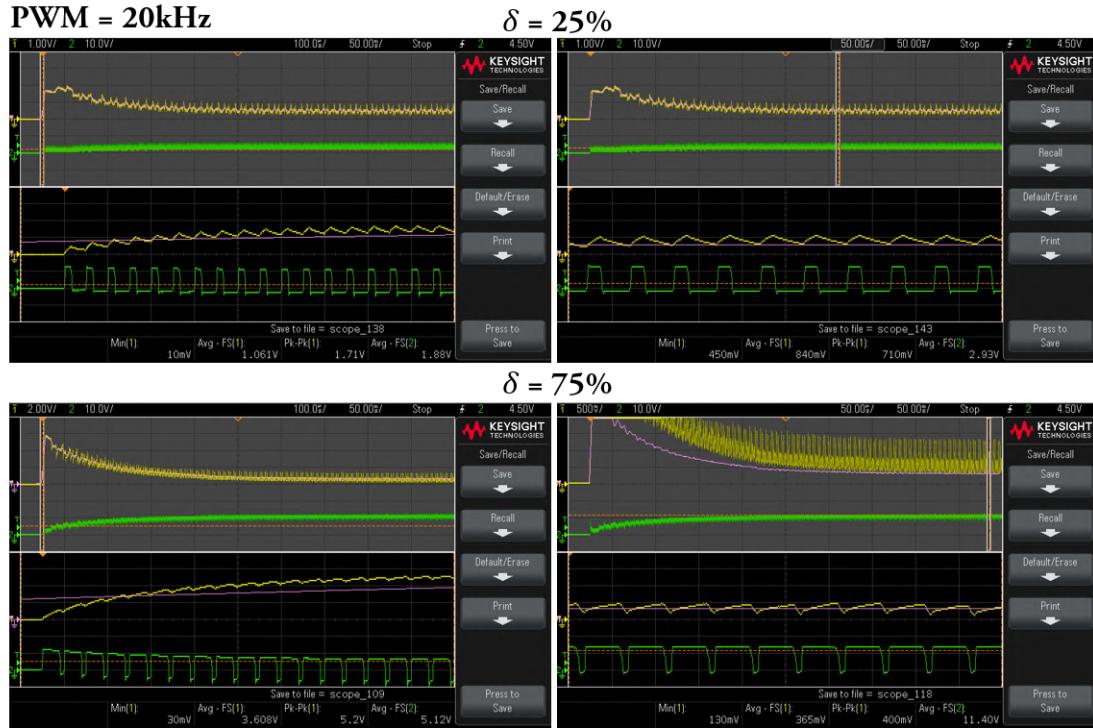
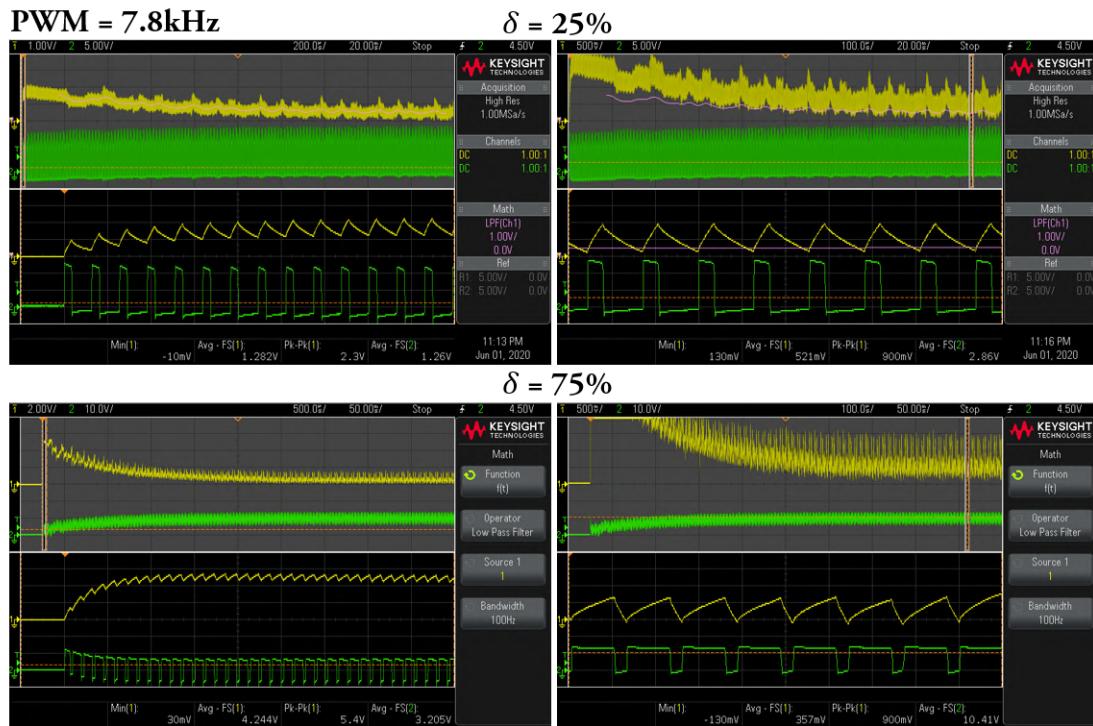


Figura 82: Acionamento do motor com PWM de 20 kHz.



Fonte: Elaborado pelo autor.

Figura 83: Acionamento do motor com PWM de 7,8 kHz.



Fonte: Elaborado pelo autor.