

Universidade de São Paulo
Escola de Educação Física e Esporte de Ribeirão Preto

Thiago Faria dos Santos

**Desenvolvimento de uma esteira ergométrica instrumentada para
avaliações fisiológicas e biomecânicas.**

Ribeirão Preto

2021

Thiago Faria dos Santos

**Desenvolvimento de uma esteira ergométrica instrumentada para
avaliações fisiológicas e biomecânicas.**

Versão Corrigida

Dissertação apresentada à Escola de Educação Física e Esporte de Ribeirão Preto da Universidade de São Paulo, para obtenção de título de Mestre em Ciências, Programa de Pós-Graduação Educação Física e Esporte.

Ribeirão Preto - SP

2021

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo ou pesquisa, desde que citada a fonte.

Catálogo na publicação
Serviço de Biblioteca e Documentação
Escola de Educação Física e Esporte de Ribeirão Preto

Santos, Thiago Faria dos

Desenvolvimento de uma esteira ergométrica instrumentada para avaliações fisiológicas e biomecânicas/ Thiago Faria dos Santos; orientador, Paulo Roberto Pereira Santiago. – 2021
63f.

Dissertação (Mestrado em Educação Física) – Programa de Pós-Graduação Educação Física e Esporte, Universidade de São Paulo, Ribeirão Preto, 2021

Versão Corrigida

1. Esteira ergométrica – 2. Instrumentação – 3. Células de carga – 4. Força de reação do solo

I. Santiago, Paulo Roberto Pereira. II. Título.

Agradecimentos

Primeiramente gostaria de agradecer a Deus, por ter me dado saúde e forças para a produção do trabalho.

Gostaria de agradecer também a meus pais, Maria e José Gaspar, que apesar de não estarem mais entre nós, carregou todos os ensinamentos por eles dados.

A minha esposa Patrícia, que além de incentivo, tem me apoiado e auxiliado diariamente.

Ao meu filho Gabriel; por ele tenho motivação e renovação da juventude, para sempre buscar melhorar.

Ao meu orientador, Prof. Dr. Paulo Roberto Pereira Santiago, por ter aceitado me orientar e por me acompanhar e instruir durante todo o processo.

Aos professores Marcelo Papotti e Renato Moraes, pelas sugestões dadas durante a qualificação.

Agradeço a todos do Laboratório de Biomecânica e Controle Motor (LaBioCom), por todas dúvidas, conselhos e conversas amigáveis.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 "This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-Brasil (CAPES) - Finance Code 001.

Financiamento: processo nº 432259/2018-0, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq); processo nº 2010/20538-7, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), código 19125 processo nº 19.1.359.90.4, Pró-Reitoria Universidade de São Paulo (USP).

Nome: Thiago Faria dos Santos

Título: Desenvolvimento de uma esteira ergométrica instrumentada para avaliações fisiológicas e biomecânicas.

Dissertação apresentada à Escola de Educação Física e Esporte de Ribeirão Preto da Universidade de São Paulo para obtenção de título de Mestre em Ciências.

Avaliado em:

Banca Examinadora

Prof. Dr(a). _____

Instituição: _____

Julgamento: _____

Prof. Dr(a). _____

Instituição: _____

Julgamento: _____

Prof. Dr(a). _____

Instituição: _____

Julgamento: _____

Resumo

SANTOS, Thiago Faria dos. **Desenvolvimento de uma esteira ergométrica instrumentada para avaliações fisiológicas e biomecânicas**. 2021. 63f. Dissertação (Mestrado em Educação Física e Esporte) – Escola de Educação Física e Esporte de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, 2021

Para termos evolução do esporte em seus diversos âmbitos não basta investir somente na formação dos atletas, também é fundamental desenvolver equipamentos para o treinamento, monitoramento e avaliação. Esteiras ergométricas são utilizadas há muito tempo para o treinamento, avaliação e pesquisa com atletas de várias modalidades esportivas e também fora do esporte. Tal uso é decorrente de sua praticidade em atividades de baixa, média e alta intensidade em centros de treinamento ou laboratórios de pesquisa. Existem poucas esteiras devidamente instrumentadas e as poucas existentes são de alto custo, superando em alguns casos os US\$ 170.000,00. Além do alto custo, suas manutenções são complicadas e muitas vezes inviáveis no aspecto financeiro. Também possuem sistema para leitura de dados fechado, sendo compatíveis apenas com os *softwares* do fabricante que, normalmente, não se comunicam com outros sensores ou sistemas de aquisição de dados. Por essas razões, pesquisadores de várias áreas têm dificuldade em adaptar as esteiras atuais às suas necessidades. Este fato ocorre até mesmo quando um laboratório possui quantia suficiente de dinheiro para a aquisição de uma esteira importada devidamente instrumentada. Assim, muitas vezes no primeiro defeito, essas ficam inutilizadas por falta de suporte técnico no país. Deste modo, este projeto tem como objetivo projetar e desenvolver uma esteira ergométrica voltada para uso em laboratórios de pesquisa, que seja versátil, adaptável às necessidades dos pesquisadores e em plataforma aberta (*Open-Source Hardware and software*). A sua automação e instrumentação será em módulos programáveis, fato que permite atualizações e adaptações por qualquer usuário. Além dos recursos convencionais de uma esteira comercial (controle de velocidade e inclinação positiva), o primeiro protótipo da esteira contará com a mensuração da velocidade linear, força de reação do solo, torque no motor, inclinações positiva e negativa, e também será capaz de simular modo de rolagem passiva, com o motor trabalhando apenas para anular as forças de resistência interna da esteira.

Palavras chave: Esteira ergométrica, instrumentação, células de carga, força de reação do solo.

Abstract

SANTOS, Thiago Faria dos. **Development of an instrumented treadmill for physiological and biomechanical evaluations.** 2021. 63f. Dissertação (Mestrado em Educação Física e Esporte) – Escola de Educação Física e Esporte de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, 2021

It is not enough to invest only in the training of athletes, it is also fundamental to develop equipment for training, monitoring and evaluation. Exercise treadmills have long been used for training, assessment and research with athletes of various sports and also outside the sport. Such use is due to its practicality in low, medium and high intensity activities in training centers or laboratories. There are few mats properly instrumented and the few existing ones are high cost, surpassing in some cases the \$ 170,000.00. In addition to the high cost, their maintenance is complicated and often unfeasible in the financial aspect. They also have a closed data reading system and are compatible only with the manufacturer's software, which normally do not communicate with other sensors or data acquisition systems. For these reasons, researchers from various areas have difficulty adapting the current treadmills to their needs. This occurs even when a laboratory has enough money to purchase a properly instrumented imported treadmill. Thus, often in the first defect, these are rendered useless due to lack of technical support in the country. In this way, this project aims to design and develop an ergometric treadmill intended for use in research laboratories, which is versatile, adaptable to the needs of researchers and open source (Open-Source Hardware and software). Its automation and instrumentation will be in programmable modules, fact that allows updates and adaptations by any user. In addition to the conventional features of a commercial treadmill (velocity and positive slope control), the first prototype of the treadmill will count on linear velocity, ground reaction force, motor torque, positive and negative slopes, and will also be able to simulate passive scrolling mode, with the motor only working to override the internal resistance forces of the belt.

Keywords: treadmill, Instrumentation, load cell, ground reaction force.

LISTA DE FIGURAS E TABELAS

Figura 1: Placa de estudo e prototipagem Arduino DUE a esquerda e o Arduino Uno, a direita.	17
Figura 2: ESP32 na versão com 30 I/O.	18
Figura 3: Principais componentes do motor de indução trifásico	21
Figura 4: Funcionamento do motor de corrente alternada trifásico.....	22
Figura 5: Inversor de frequência, formas de controle, IHM (interface homem máquina) e I/O (entradas e saídas digitais e analógicas)	24
Figura 6: Fotoacoplador	26
Figura 7: Sinal de um mesmo sensor ótico usando um osciloscópio, sinal usando o <i>schimit trigger</i> (a) e sinal sem utilizar (b).....	27
Figura 8: Fluxograma geral do protótipo. DAQ NI6009 (conversor analógico digital).....	29
Figura 9: Placa de identificação do motor da esteira Moviment® LX160 (motor Weg W22 plus de 2 cv)	30
Figura 10: Sensor ótico.....	33
Figura 11: Espr32(a), modelo 3d caixa(b), fresamento da placa de circuito impresso(c).....	37
Figura 12: Projeto placa controlador usando ESP32.....	38
Figura 13: Resposta do fotoacoplador forma uma onda que não é quadrada, por volta de 19ms	39
Figura 14: Fluxograma da programação da placa controladora	40
Figura 15: Suporte das células de carga.	42
Figura 16: Localização de cada célula na base da esteira.....	42
Figura 17: Caixa com os plugs do tipo MIKE para entre das 4 células de carga, os 4 amplificadores MKTC-05 e o DAQ NI USB6009.	43
Figura 18: Fluxograma da coleta e processamento dos dados das células de carga executado no MatLab.	46
Figura 19: Software Visual Studio com a primeira página da interface gráfica do software de controle da esteira.	47
Figura 20: Fluxograma software de controle.....	48
Figura 21: Exemplo de protocolo carregado do texto acima (incrementando 1km/k a cada 20seg).	55
Figura 22: Resultados da utilização de tempo de <i>debounce</i> diferentes, 500 μ s, 5000 μ s e 3450 μ s, respectivamente.	57

Figura 23: Dados de velocidade da esteira, quando controlado pela freq. de alimentação.	58
Figura 24: Gráfico velocidade pelo tempo, durante protocolo de testes. Na linha vermelha a velocidade programada e em azul a velocidade medida na lona da esteira.....	59
Figura 25: Imagens FSR do solo (um passo) durante caminhada a 5km/h (fig. a) o somatório do sinal das 4 células de cargas, já na (fig. b) o sinal das 4 células separados).	60
Figura 26: Sinal das 4 células de carga e da somatória delas durante corrida a 10km/h.....	60
Figura 27: FSR, em vermelho o sinal do enquanto o pé direito está em contato com a esteira e em verde o sinal enquanto o pé esquerdo está em contato com o solo.....	61
Figura 28: FSR de cinco contatos dos pés do sujeito com o piso separados um a um. Vertical GRF (força de reação vertical do solo (FSR) em relação a peso corporal do sujeito)	62
Figura 29: Leituras COP da esteira (fig. a) caminhada com a esteira a lona parada lateralmente, fig. b) caminhada com a lona parada de trás para frente e fig. c) corrida tradicional com a lona a 10km/h).....	63

Lista de Abreviaturas e Siglas

A	Amperes
ADC	<i>Analog to Digital Converters</i> (conversor analógico digital)
CA	Corrente Alternada
CC	Corrente Contínua
CI	Circuitos integrados
conversor A/D	Conversor Analógico para Digital
COP	centro de pressão
cv	Cavalo vapor
FSR	Força de reação do solo
HP	Horse Power
I/O	<i>Input/Output</i> (entrada e saída)
IDE	<i>Integrated Development Environment</i>
IF	Inversor de Frequência
IHM	Interface Homem Máquina
IOT	Internet of Things (internet das coisas)
ISR	Interrupt service routine (rotina de interrupção)
Km/h	Quilômetros por hora
LED	Light Emitting Diode (diodo emissor de luz)
mm	milímetros
ms	milissegundos
PCB	Printed Circuit Board (Placa de circuito impresso)
seg	segundos
t	Tonelada
v	volts
µs	microsegundo

Sumário

1	INTRODUÇÃO.....	12
2	OBJETIVOS.....	15
2.1	Objetivos gerais	15
2.2	Objetivos específicos	15
3	REVISÃO DE LITERATURA	16
3.1	Microcontroladores	16
3.1.1	Arduino.....	17
3.1.2	ESP32	18
3.1.3	Comparação entre microcontroladores.....	19
3.1.4	Linguagem C usada nos microcontroladores	20
3.2	Motores de Indução	20
3.3	Inversores de Frequência	23
3.4	Células de carga	24
3.5	Condicionamento do Sinais Analógicos	25
3.5.1	Amplificadores	25
3.5.2	Amplificadores Operacionais	25
3.5.3	Conversores Analógico/Digitais.....	25
3.6	Sensores Acoplador Óptico.....	26
3.7	Sensor Magnético (<i>reed switch</i>)	27
3.8	Linguagem C# e Visual Estúdio	28
4	MATERIAIS E MÉTODOS.....	29
4.1	Motor	30
4.2	Inversor de Frequência.....	31
4.3	Velocidade (sensor ótico)	32
4.4	Construção da placa de controle	37
4.5	Coleta de dados de força de reação do solo (FSR)	42

4.5.1	Células de carga usadas	42
4.5.2	Amplificadores	43
4.5.3	Conversor Analógico/Digital.....	43
4.5.4	Software coleta de dados	44
4.6	Software de controle da esteira	47
4.6.1	Controle de velocidade	51
4.6.2	Controle baseado na frequência de alimentação do inversor	51
4.6.3	Controle baseado na velocidade	52
4.6.4	Realização de protocolos predefinidos	53
4.7	Custo para desenvolvimento do protótipo	56
5	RESULTADOS E DISCUSSÃO	57
5.1	Leituras de velocidade da lona da esteira.	57
5.2	Controle da esteira	58
5.2.1	Controle baseado na velocidade de leitura.	59
5.3	Leituras de forças de reação do solo FSR.....	60
6	POSSIBILIDADES FUTURAS	64
7	CONCLUSÃO.....	65
8	REFERÊNCIA BIBLIOGRÁFICAS	66

1 INTRODUÇÃO

A corrida de rua é um esporte que vem crescendo muito nos últimos anos, sendo procurada tanto por atletas amadores como por atletas de alto rendimento, e os motivos dessa crescente procura são vários, como por exemplo os benefícios físico e mental, a facilidade de iniciar a prática desse esporte (WILLIAMS, 1997) e baixo custo dos equipamentos que envolvem esse esporte. Por outro lado, o índice de lesões osteoarticulares ocasionados por esse esporte apresenta números relativamente altos. (LYSHOLM; WIKLANDER, 1987) acompanhou 60 corredores por um ano; durante esse período 55 lesões foram observadas em 39 dos participantes. O impacto parece estar relacionado ao aumento das chances dessas lesões (MCCORMICK; NWACHUKWU; PROVENCHER, 2012), portanto, analisar a força de reação do solo (FSR) é algo que pode melhorar o entendimento sobre o mecanismo de algumas lesões provenientes da prática da corrida e também pode estar ligado a alguns índices de desempenho do atleta (ANDERSON, 1996; HEISE; MARTIN, 2001; MARTIN; SPIRDUSO, 2001). Sendo assim, o desenvolvimento de métodos de avaliação desse tipo de variável é algo cada vez mais presente nos laboratórios de pesquisa e nos grandes centros de treinamento.

Paralelo a tudo isso, o número de indivíduos com necessidades especiais praticantes de atividades esportivas por recreação ou mesmo de alto rendimento tem aumentado muito na última década, crescendo também a necessidade de promover mais estudos com esse público, gerando a necessidade de esteiras com largura e comprimento diferenciados que permita realizar ajustes para que esse público seja avaliado de forma adequada .

Esteiras também podem ser utilizadas para treinamento e avaliações em outras modalidades que não a corrida, como por exemplo o ciclismo. Citando um exemplo (BOUILLOD; PINOT; VALADE; CASSIRAME *et al.*, 2018), utilizou esteiras de grande porte para avaliar a influência da posição dos ciclistas sobre a mecânica e custo energético durante escaladas simuladas.

Em pesquisa na base de dados da Pubmed¹, acessado em 28 de dezembro de 2020 com as palavras chave *treadmill and exercise* foram encontrados mais de 28 mil artigos, utilizando as palavras chave *treadmill and wheelchair* o número cai para apenas 200 artigos e com *treadmill and cyclist* apenas 65 artigos foram encontrados. Isso é um bom indicador da escassez de estudos com população de não corredores; acreditamos que a falta de esteiras adaptadas para esse público e o alto custo das existentes no mercado sejam fatores importantes na escassez desses estudos. Portanto, o desenvolvimento de uma esteira que seja adaptável, capaz de avaliar diversas tarefas motoras como correr, pedalar, usar cadeira de rodas, etc., é algo que deve auxiliar o desenvolvimento de novas pesquisas.

No mercado existem muitas esteiras, em poucos casos já instrumentadas e/ou com boas dimensões, porém são pouco flexíveis e com custo extremamente elevado. Um exemplo que podemos citar são as esteiras nacionais: a TK35 fabricada pela Cefise® Biotecnologia Esportiva (R\$ 127.370,00)², a Super ATL fabricada pela Imbramed® (R\$ 45.000,00)³, ambas com limitações em dimensão e de soluções de entrada e saída para aquisição de dados analógicos ou digitais.

Fora do Brasil existem opções de esteiras devidamente instrumentadas como, por exemplo, Force-Sensing Tamden⁴, produzida pela AMTI® Force and Motion Inc USA AMTI e a Stratus® Gaitway 3D 150/50, produzida pela H/P/Cosmos® Sport Germany⁵. Apesar da existência desses modelos fora do Brasil, os valores elevados para aquisição e manutenção no país (acima dos U\$ 170.000,00)⁶ dificultam bem sua aquisição e manutenção. Além do custo, estas esteiras especiais também apresentam pouca flexibilidade, pois não permitem grandes adaptações em sua estrutura ou programação.

Para demonstrar as necessidades de adaptação em esteiras comerciais, para que possam realizar suas pesquisas de forma adequada, podemos citar o estudo feito por (LAKOMY, 1987; LAKOMY; CAMPBELL; WILLIAMS, 1987) que inseriu um trilho guia e limitadores à frente

¹ <https://www.ncbi.nlm.nih.gov>.

² <http://www.cefise.com.br/produto/49/40/esteira-tk35> (CEFISE®, 2017), valor orçado em janeiro de 2020.

³ <http://inbrasport.com.br/produto/esteira-ergometrica-super-atl>, (IMBRAMED®, 2017), valor orçado em janeiro de 2020.

⁴ <http://www.amti.biz/index.aspx>, (AMTI®, 2017).

⁵ <https://www.hpcosmos.com/en/products/sport/biomechanics-gaitway-3d>, (H/P/COSMOS®, 2018).

⁶ Valor orçados em janeiro de 2020

e atrás da esteira, para estudar características fisiológicas de dez cadeirantes com tetraplegia e dois com paraplegia.

Alguns estudos exigem que a velocidade da esteira seja ajustada pelo próprio usuário. Uma solução para que o atleta tenha a liberdade de modular a velocidade por si só, é utilizar uma esteira sem motor para movimentar a lona de rolagem (LAKOMY, 1987; LAKOMY; CAMPBELL; WILLIAMS, 1987). Foi o que utilizou (WOODWAY®, 2019) para analisar o desempenho de corredores durante *sprints* de alta intensidade. Um fato interessante deste estudo é que o autor mensurou a força horizontal de corrida por meio de um transdutor de força fixado na parede e ancorado na cintura dos corredores. Alguns anos após a realização do trabalho, a fabricante da esteira começou a produzir esteiras semelhante à do estudo⁷.

Os fatos apresentados denotam um grande entrave para a realização de muitas pesquisas com o esporte e saúde, visto que as esteiras facilitam as avaliações dentro do ambiente de laboratório. No entanto, as existentes no mercado têm alto custo e/ou apresentam limitações nas possibilidades de adaptação para avaliar outras atividades motoras que não a corrida. Assim, o desenvolvimento de uma esteira com o conceito *Open Source* contribuirá para a inovação e desenvolvimento científico/tecnológico de pesquisas no esporte e saúde. Projetos *Open Source*, têm ainda a vantagem de rotineiramente receber a colaboração de diversos outros pesquisadores e profissionais ao redor do mundo a fim de evoluir o desenvolvimento do mesmo.

⁷ <https://www.woodway.com/products/force>.

2 OBJETIVOS

2.1 Objetivos gerais

Possui o objetivo de atender às necessidades de pesquisadores e treinadores de diversas áreas e modalidades. O objetivo geral desse estudo é desenvolver um protótipo de uma esteira instrumentadas com células de carga e que possa ser controlada de diversas formas a um custo relativamente baixo.

2.2 Objetivos específicos

De forma mais detalhada os objetivos do estudo foram:

- Obter medidas de velocidade diretamente na lona das esteiras, para dessa forma evitar erros relacionados ao escorregamento que possa haver entre a lona e o rotor da esteira.
- Controlar a velocidade da esteira por computador, permitindo a criação de protocolos pré-definidos baseados tanto na frequência de alimentação do motor, como na velocidade medidas diretamente na lona da esteira.
- Obter leituras das forças verticais que o usuário aplica na superfície da esteira.
- Calcular o centro de pressão com base na distribuição das forças verticais aplicadas nas células de carga na base da esteira.
- Utilizar componentes de custo baixo e de fácil acesso, facilitando sua construção, replicação e manutenção.

3 REVISÃO DE LITERATURA

3.1 Microcontroladores

No início do desenvolvimento de circuitos integrados (CI), o componente era desenvolvido com a finalidade de realizar uma única tarefa. Isso gerava um problema principalmente para a prototipagem de novos produtos, pois todo novo teste um novo hardware deveria ser criado. Na busca de resolver esse problema, Frederico Faggin, na época engenheiro da Intel, desenvolveu um CI capaz de acessar um roteiro de tarefas armazenadas em uma memória específica, resultando no microprocessador 4004, lançado em 1971 (DE ALMEIDA; DE MORAES; SERAPHIM, 2017). Um microprocessador ainda precisa de outros componentes para funcionar, pois os mesmos são capazes apenas de realizar cálculos. Para o funcionamento de um microprocessador, diversos componentes eram necessários: unidade de processamento, *clock*, memórias, etc.. Para facilitar ainda mais os fabricantes encapsularam no mesmo CI a unidade de processamento, as memórias e uma série de portas de entrada e saída (I/O).

Algo importante dentro dos microcontroladores é a capacidade de realizar interrupções, que nos sistemas embarcados também é conhecido com ISR (*interrupt service routines*). As interrupções devem ser usadas quando determinado evento não pode ser perdido, como por exemplo a medição de RPM de um motor, etc. De forma simplificada o programador define um evento a ser monitorado; quando esse evento ocorre, o programa principal é pausado, a rotina programada na interrupção é executada e ao finalizar ele retorna de onde parou no programa principal (DE ALMEIDA; DE MORAES; SERAPHIM, 2017). Cada microcontrolador apresenta ou não pinos que podem ser programados como interrupção, o Atmel328 (Arduino Nano e UNO) é possível utilizar os I/O digitais 2 e 3; já o Arduino Mega permite usar os I/O digitais 2, 3, 18, 19 e 21; o Arduino Due, permite usar qualquer pino digital disponível como interrupção (ARDUINO, 2020) e por fim o ESP32 também permite usar todos os I/O digitais como interrupção (ESPRESSIF-SYSTEMS®, 2020).

3.1.1 Arduino

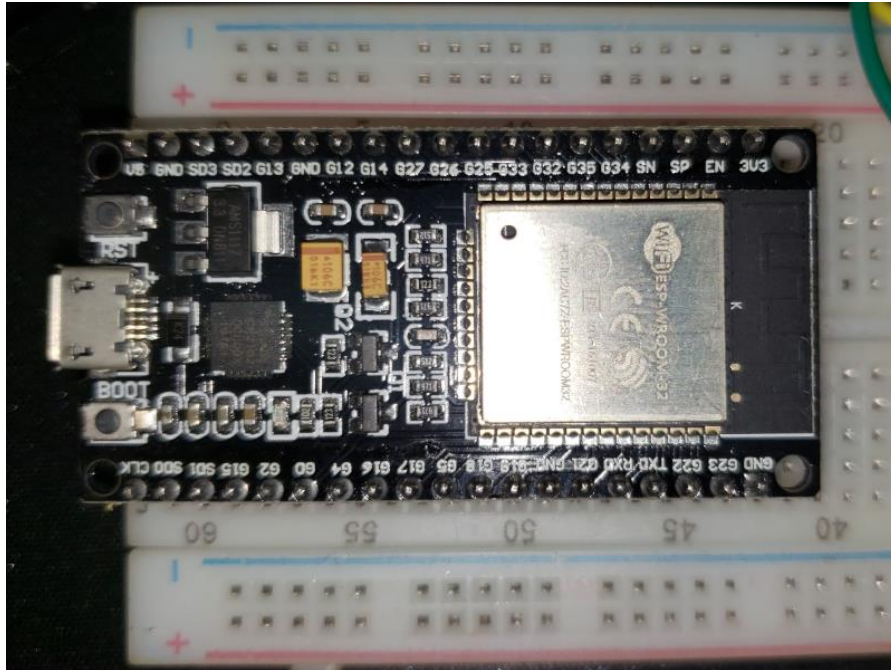
Por volta do ano de 2005, Massimo Banzi criou a placa de controle Arduino, com o propósito de baratear e facilitar o ensino de programação embarcada. Na placa praticamente todos os componentes de um sistema embarcado já estavam presentes. A versão mais comum das placas Arduino é a UNO, formada por um microcontrolador da Atmel, o ATmega 328, de 8bits, 14 portas de entrada/saída digitais, 6 entradas analógicas. Além do microcontrolador, a placa também já contava com um conversor RS232 para USB, cristal oscilatório de 16MHz, responsável pelo *clock* do microcontrolador (STEVAN; SILVA, 2015). Atualmente existe uma diversidade bem grande de placas Arduino, podemos citar entre elas o Arduino Mega DUE e o Arduino Uno (figura 1). As necessidades do projeto é o que vai definir a placa que deverá ser usada. A placa Arduino Due é baseada no microcontrolador Atmel SAM3X8E-Cortex-M3, sendo o primeiro Arduino de 32bits e dois núcleos de processamento diferente, permitindo processamento assíncrono.

Figura 1: Placa de estudo e prototipagem Arduino DUE a esquerda e o Arduino Uno, a direita.



3.1.2 ESP32

Figura 2: ESP32 na versão com 30 I/O.



O microcontrolador ESP32 é fabricado pela Espressif System®. O microcontrolador ESP32 vem ganhando muita popularidade pelo seu baixo custo e conectividades já embutidas em seu encapsulamento, como bluetooth low energy (BLE) e rede sem fio, que são muito bem vindos para o mundo da internet das coisas (IOT).

Uma das principais vantagens do ESP32 é o fato de possuir dois núcleos de processamento, permitindo uma lógica de programação assíncrona, ou seja, a realização de mais de uma tarefa ao mesmo tempo. Esse foi o ponto que nos fez optar pelo ESP32 como sendo a base para a confecção da placa controladora usada em nosso protótipo.

A Espressif System® recomenda o ESP-IDF, como meio de programar e enviar seus programas para o ESP32. Apesar disso, como a plataforma Arduino é muito popular, tanto para prototipagem quanto para ensino, a Espressif System® criou bibliotecas para facilitar a utilização do ambiente de desenvolvimento integrado (IDE) do Arduino como meio de programar o ESP32. Em pesquisas por sites de desenvolvimento, IOT., entre outros do gênero, é muito comum encontrar grande tutoriais utilizando ESP32 e a IDE do Arduino.

3.1.3 Comparação entre microcontroladores

Os microcontroladores têm diferenças consideráveis entre si. O desenvolvedor precisa considerar essas características a fim de encontrar o melhor e mais barato CI que dê conta de cumprir a tarefa de forma eficiente e confiável.

As principais características que temos entre eles é a capacidade de processamento, custo, números de entradas e saídas (I/O), etc. Como exemplo, podemos citar placas de desenvolvimento especializadas no processamento de visão de máquina baseado em rede neural convolucional, como é o caso do Sipeed Maix Bit, que seria uma boa opção pra projetos que envolvem reconhecimento de imagem. Já para projetos que buscam conectividade o ESP32, apresenta em seu encapsulamento rede sem fio e *bluetooth*. Abaixo temos uma tabela com algumas características de placas de desenvolvimento.

Tabela 1: Comparativo entre placas para prototipagem

Comparação de algumas placas de desenvolvimento							
	n° de núcleos	n° de I/O	Bluetooth	Wi-fi	Frequência	ADC	Nível Lógico
Arduino Uno	1	13	não	não	16MHz	6 canais 10bits	5volts
Arduino Leonardo	1	13	não	não	16MHz	12 canais 10bits	5volts
Node MCU ESP8266	1	17	não	HT20	160MHz	1 canal 10bits	3.3volts
Node MCU ESP32	2	36	BLE 4.2	HT40	80MHz	18 canais 12bits	3.3volts
Sipeed Maix – BIT	2	35	não	não	400MHz		3.3volts e 1.8volts

Analisando as características de cada microcontrolador, escolhemos o ESP32 para utilização no protótipo, visto que o mesmo apresenta uma frequência muito superior ao do Arduino e possui dois núcleos de processamento, permitindo dividir os processos entre eles. Além disso, o fato de apresentar conexão de rede sem fio e bluetooth, facilitará melhorias futuras.

3.1.4 Linguagem C usada nos microcontroladores

Boa parte dos microcontroladores podem ser programados utilizando a linguagem C, portanto é importante que programadores de sistemas embarcados dominarem bem essa linguagem. Podemos citar como exemplo o Arduino Uno, que utiliza o microcontrolador Atmel328 tendo como linguagem básica o Assembly (linguagem de montagem), mas pode ser programado utilizando C e o código pode ser convertido para Assembly, antes de ser enviado para o microcontrolador. Dessa forma temos o desenvolvimento agilizado, uma vez que linguagens de alto nível, com C, tem características muito semelhantes à linguagem usada na fala e escrita e falada, sem afetar de forma expressiva a *performance* do programa (STEVAN; SILVA, 2015). A estrutura do código em linguagem C para o Arduino apresenta basicamente três blocos: o primeiro onde as variáveis e constantes globais são declaradas, o segundo onde são configurados alguns parâmetros, essa função é chamada de `setup()`; por fim temos a função `loop()`, como o nome mesmo fala, que corresponde à função que será repetida até que o microcontrolador seja desligado, ou algum erro ocorra.

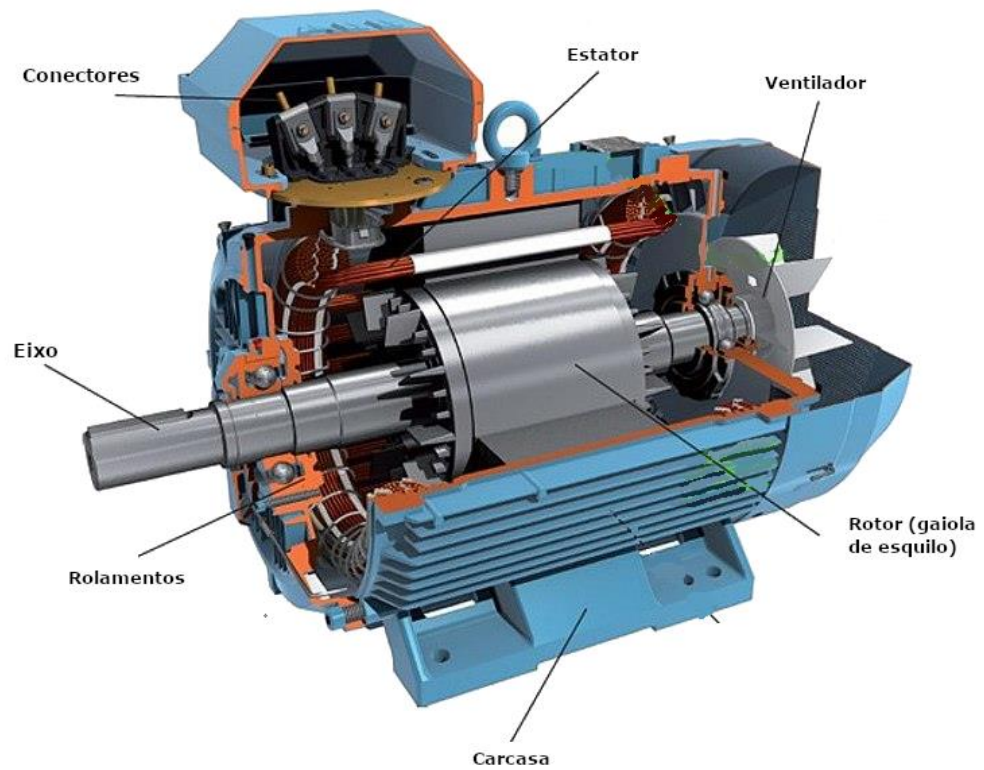
3.2 Motores de Indução

Motores elétricos de indução têm sua rotação gerada pela interação entre dois campos magnéticos. O campo magnético rotativo do estator induz a corrente no rotor (PETRUZELLA, 2013).

Existem no mercado motores de corrente contínua (MCCORMICK; NWACHUKWU; PROVENCHER, 2012) e de corrente alternada (CA). Grande parte dos projetos motores de CA são utilizados devido às suas vantagens em relação aos motores de CC. Entre essas vantagens podemos citar o custo reduzido, maior disponibilidade no mercado, manutenção reduzida e melhor rendimento (FRANCHI, 2009).

Motores de Indução Trifásico são constituídos por diversos componentes, dentre os quais podemos citar: rotor e ventilador, núcleo do estator, carcaça e estrutura, suportes finais e rolamentos (figura 3). O formato e material que cada componente é constituído varia conforme a finalidade de uso do motor (HAND, 2015).

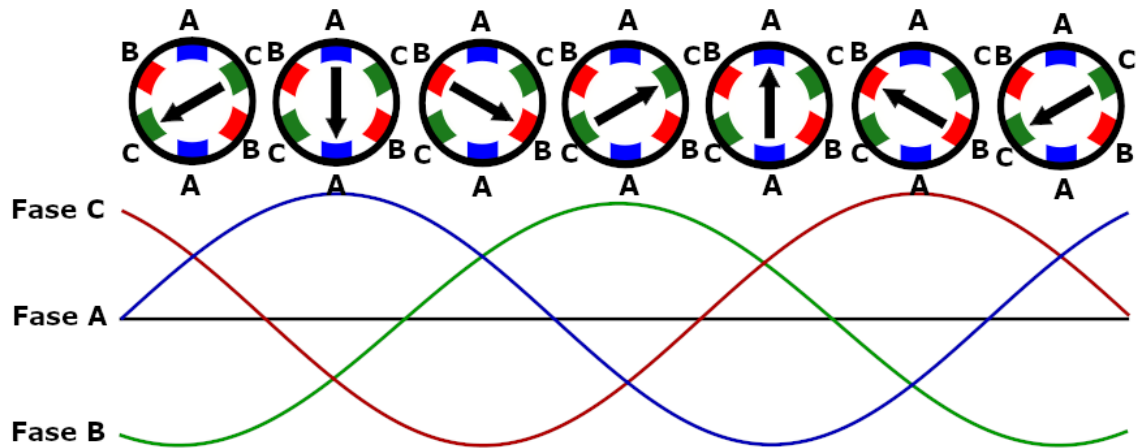
Figura 3: Principais componentes do motor de indução trifásico



Fonte: (Adaptado de Wikimedia Commons – Licencias creative Commons)

De uma forma bem simples, no motor por indução trifásico a corrente em cada bobina do estator varia em forma senoidal, de forma a gerar um campo magnético rotativo que gira o rotor, a (figura 4) ilustra bem como tudo ocorre.

Figura 4: Funcionamento do motor de corrente alternada trifásico.



Fonte: (Adaptado de Wikimedia – *licenses creative commons*)

Podemos variar a velocidade do motor de Indução de CA, variando o número de polos na sua fabricação, ou a frequência da fonte de alimentação em cada fase desse motor, como na fórmula abaixo (PETRUZELLA, 2013). Podemos citar como exemplo um motor de seis polos, sendo alimentado com 60 Hz, apresentaria 3600 RPM. Lembrando que o número de polos de um motor é sempre múltiplo de dois.

$$V = (120 f)/P,$$

Sendo que:

V = Velocidade em RPM do motor.

f = frequência da corrente de alimentação

P = número de polos do motor.

Apesar do cálculo acima, existe uma diferença entre a velocidade síncrona do rotor (calculada acima) e a velocidade assíncrona. Podemos calcular a velocidade assíncrona subtraindo a perda do motor (também conhecida como escorregamento), da velocidade síncrona.

Caso a velocidade nominal for igual a velocidade síncrona do motor não teríamos um campo magnético rotativo, pois os polos do estator se alinham com os polos do rotor, o que não geraria torque no rotor e o mesmo ficaria parado (HAND, 2015).

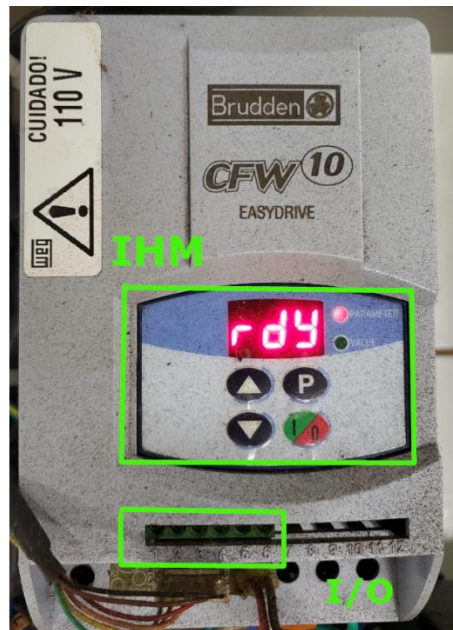
3.3 Inversores de Frequência

Inversores de frequência (IF) são usados amplamente em conjunto com motores a fim de controlar a alimentação do motor. Desta forma é possível definir tempo de aceleração e tempo de parada, bem como controlar sua velocidade de rotação, entre outras funções. A utilização dos IF, além de possibilitar um controle fino do motor, garante maior segurança para o motor e também leva à economia de energia. Os IF possuem uma central de processamento, que recebe parametrizações via uma interface para o usuário ou mesmo interface com computadores (FRANCHI, 2009).

O do IF alterna uma corrente contínua CC nas fases do motor, simulando uma corrente alternada. Como a frequência de alimentação do motor interfere na sua velocidade de rotação, podemos controlar a velocidade e também o sentido de rotação do motor, alterando a frequência de alimentação (PETRUZELLA, 2013).

Alguns modelos de IF apresentam I/O (figura5) principalmente digitais por onde é possível configurar ações (WEG®, 2019), como aumentar ou diminuir a frequência de alimentação ou inverter o giro do motor, entre diversas outras possibilidades. Também é comum que os IF possuam uma interface de controle conhecida como interface homem máquina (IHM), conforme visto na (figura 5).

Figura 5: Inversor de frequência, formas de controle, IHM (interface homem máquina) e I/O (entradas e saídas digitais e analógicas)



3.4 Células de carga

Existem diversas maneiras e utilidades para se medir forças internas e/ou torque. Uma das maneiras mais usadas é a medida de deformação dos materiais, método conhecido como extensometria. Considerando que o material em questão apresenta características isotrópicas, homogêneas e o seu limite elástico não for superado, temos uma relação linear entre a força aplicada sobre ele e sua deformação. Essa relação foi estabelecida por Robert Hooke, em 1678. Willian Thomson em 1856, notou que a resistividade de um material metálico quando submetido à tração, varia de forma inversamente proporcional à seção transversa. Somando as duas informações podemos notar que ao aplicar uma força de tração em uma barra metálica, teremos o alongamento do material e por sua vez a redução da sua área de secção transversa e o resultado é o aumento da resistência desse condutor (FIALHO, 2002). Essas informações foram fundamentais para a criação dos primeiros extensômetros.

Células de cargas, ou transdutores de força, são formados por estruturas normalmente metálicas, projetadas para sofrer deformação em determinados pontos, sendo nesses pontos colados extensômetros, também conhecido como *strain gauges*. Normalmente os *strain gauges* são instalados de forma a quando um for comprimido o outro é estendido, dessa forma as

leituras ficam ainda mais fortes. É possível formar pontes de 1, 2 ou 4 *straing gauges*. Quanto mais *strain gauges*, maior a sensibilidade da célula de carga criada (FRANCHI, 2015).

Atualmente existem células de cargas prontas de diversos formatos, voltados para diversas aplicações, assim como também podemos construir nossas células projetando bem a peça e colando os strain gauges em locais estratégicos.

3.5 Condicionamento do Sinais Analógicos

3.5.1 Amplificadores

Amplificadores correspondem a um tipo de componente eletrônico que recebe um sinal e pode aumentar, ou mesmo diminuir sua amplitude. Comumente os amplificadores apresentam uma entrada e uma saída de sinal.

3.5.2 Amplificadores Operacionais

Os amplificadores operacionais, como o próprio nome diz, são amplificadores capazes de fazer operações matemáticas a fim de melhorar o sinal que se deseja analisar. Diferente dos amplificadores comuns, os amplificadores operacionais apresentam duas entradas de sinal, uma positiva chamada de não inversora e uma entrada negativa, chamada de inversora.

3.5.3 Conversores Analógico/Digitais

Conversores de sinal analógico em sinal digital (conversor A/D), são necessários para que possam transmitir o sinal para um computador, ou visualizar em algum painel. Um conversor A/D apresenta diversas características, as duas principais seriam a frequência de amostragem e a resolução que normalmente é expressa em bits.

Cada projeto precisará de uma frequência de amostragem específica, dependendo do tempo de duração do evento que se deseja observar. Segundo o Teorema de Nyquist, a frequência de amostragem deve ser pelo menos duas vezes maior do que a maior frequência do sinal que se deseja analisar (FIALHO, 2002).

A resolução de um conversor A/D corresponde à menor variação possível que gera uma variação no sinal de saída. Como a conversão será para valores binários, um conversor A/D de

um único bit apresenta dois valores possíveis; já um conversor de 12bits, presente por exemplo nos ESP32, apresenta 4096 possibilidades. O número de possibilidades de um conversor A/D pode ser dado por 2^n , onde n é o número de bits do conversor A/D. Com base na resolução é possível calcular qual seria a quantificação do conversor A/D da seguinte forma:

$$S = 1/2^{(b)} \cdot FE$$

onde:

S = sensibilidade do conversor A/D (menor valor que é possível ser lido)

b = número de bits disponível no conversor A/D

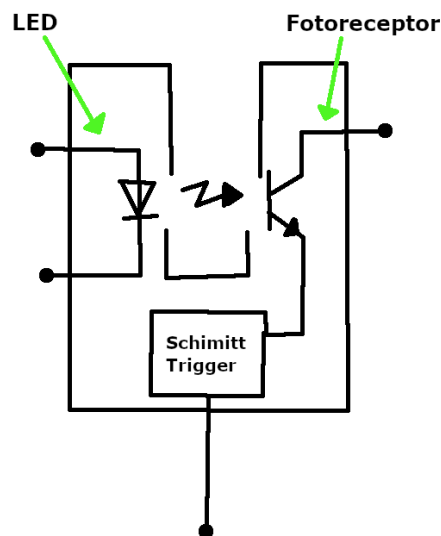
FE = fundo de escala de um do sinal.

(FIALHO, 2002)

3.6 Sensores Acoplador Óptico

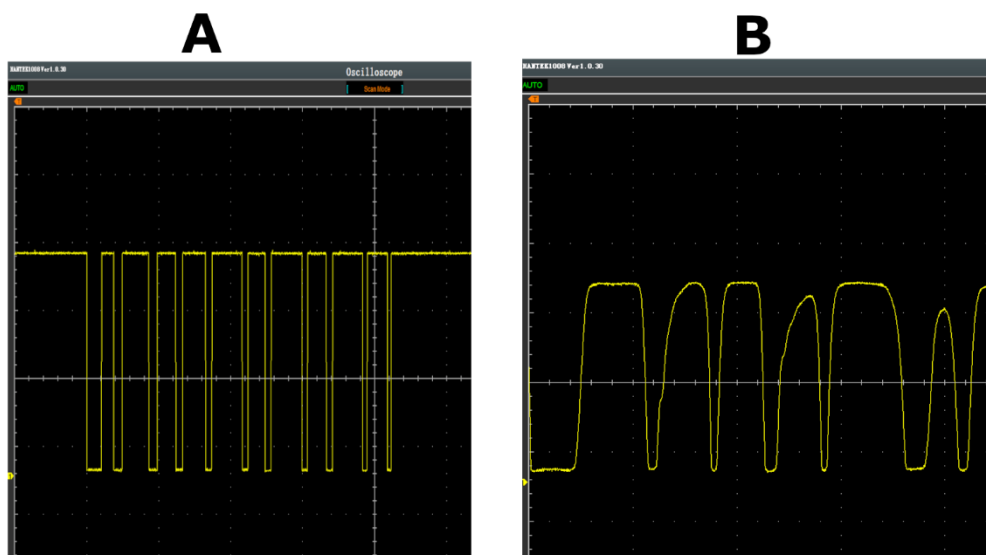
Usado normalmente na indústria para medir velocidade (THOMAZINI; DE ALBUQUERQUE, 2020), na maior parte das vezes, consiste em um disco perfurado ou algo que reflita a luz emitida por um diodo emissor de luz (LED), para um receptor sensível à luz (figura 6). Como o sinal obtido apresenta oscilação, a utilização de um limiar de gatilho, conhecido como *Schmitt Trigger*, tem a finalidade de gerar uma onda quadrada, facilmente interpretando o sinal de forma binária (figura 7(a) e figura 7(b)).

Figura 6: Fotoacoplador



Fonte: Adaptado de (THOMAZINI; DE ALBUQUERQUE, 2020)

Figura 7: Sinal de um mesmo sensor ótico usando um osciloscópio, sinal usando o *schimit trigger* (a) e sinal sem utilizar (b)



É possível construir sensores óticos utilizando um LED infravermelho e um fototransistor (como por exemplo o TIL78) (VERTULO, 2017). Existe a possibilidade da utilização de módulos com o LED infravermelho, fototransistor e o CI LM393 (FAROQI; RAMDHANI; FRASETYIO; FADHIL, 2018), compatíveis com os principais microcontroladores e sendo de baixo custo.

3.7 Sensor Magnético (*reed switch*)

Outra forma de medir velocidade frequentemente utilizada na indústria é usar sensores magnéticos conhecidos como *reed switch*. Esse tipo de sensor é bem simples e de fácil fabricação. Consiste em dois terminais metálicos dentro de uma ampola de vidro e na outra parte móvel existe um ímã permanente. No momento em que o ímã permanente se aproxima da ampola com os terminais, os mesmos fecham contato; quando ímã está distante, o contato permanece aberto (THOMAZINI; DE ALBUQUERQUE, 2020). Na esteira usada no protótipo, por padrão usa-se um sensor do tipo *reed switch* para medir a velocidade de rotação da polia que liga o rotor do motor à correia, porém isso pode gerar alguns erros, posteriormente esse problema será melhor detalhado.

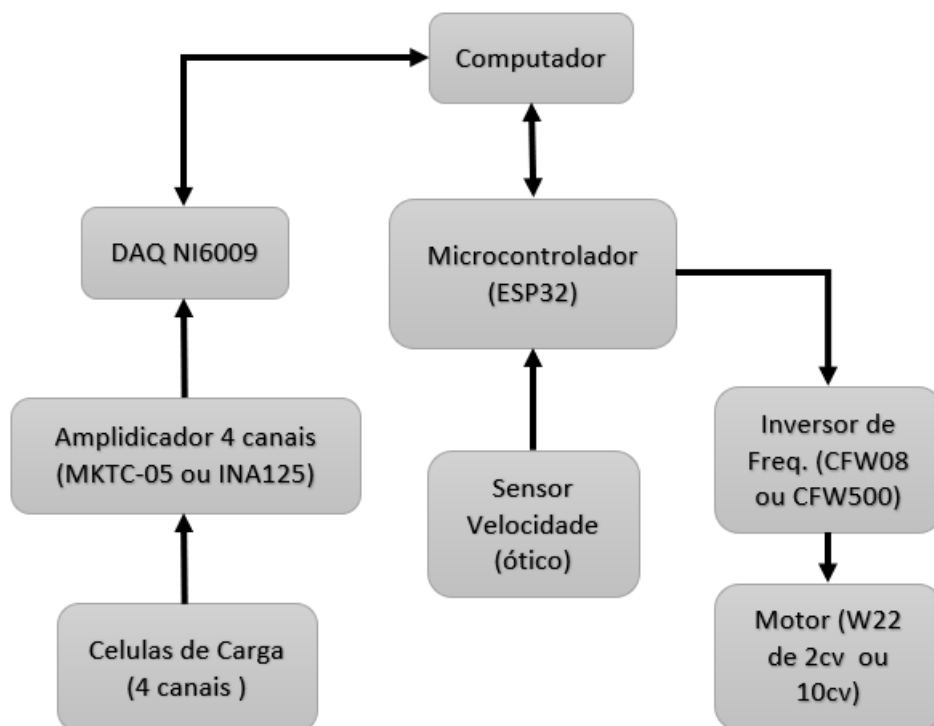
3.8 Linguagem C# e Visual Estúdio

A linguagem C#, foi criada pela Microsoft®, influenciada pelas linguagens C, C++, Pascal e Java. A linguagem C# pode ser usada tanto pelo modo console (terminal ou prompt) ou pelo modo gráfico (interface gráfica). Para o desenvolvimento em linguagem C#, a IDE Visual Studio, disponível em versão livre para desenvolvimento não comercial, pode ser usada para facilitar o desenvolvimento de interfaces gráficas.

4 MATERIAIS E MÉTODOS

Nesse primeiro momento utilizamos a base de uma esteira comercial. Futuramente pretendemos montar toda estrutura da esteira com área útil maior e com melhor capacidade de velocidade. Na descrição dos materiais e métodos, colocaremos informações do que foi feito nessa adaptação, bem como o pretendido e realizado no projeto da esteira de maior tamanho. No fluxograma abaixo (figura 8) temos uma ideia geral de como será cada módulo do projeto. O computador envia e recebe dados do sistema de aquisição (DAQ NI6009) e também envia e recebe dados para a placa controladora (microcontrolador ESP32). O módulo placa controladora recebe o sinal do sensor de velocidade e controla o IF que, por sua vez, controla a velocidade de rotação do motor.

Figura 8: Fluxograma geral do protótipo DAQ NI6009 (conversor analógico digital)



4.1 Motor

A esteira instrumentada originalmente utilizada foi da marca Moviment®, modelo LX160; a mesma não possui inclinação ajustada eletronicamente, portanto, retiramos esse recurso dessa fase de desenvolvimento. O motor presente nessa esteira é da marca Weg 22 Plus, com 2 cavalo vapor (cv), frequência nominal de alimentação de 60 Hz, 2 polos, portanto com rotação nominal de 3380 Hz, como é possível notar na placa de identificação na figura 9. Ressaltando que apesar do motor usado ser trifásico, utilizamos um IF, que permite o uso de uma tomada convencional 220 v como fonte de energia para o protótipo, apenas tomando-se cuidado com a amperagem que a mesma suporta.

Figura 9: Placa de identificação do motor da esteira Moviment® LX160 (motor Weg W22 plus de 2 cv)



4.2 Inversor de Frequência

Para controlar o motor, a esteira possui originalmente um inversor de frequência da marca Weg e modelo CFW08 e características compatíveis com o motor que ele alimenta. O mesmo foi parametrizado para se adequar às características do motor original de 2cv. Além da parametrização básica, modificamos os parâmetros 263 até o 266 (WEG®), referente às I/O digitais para que possam ser controladas pelo microcontrolador que programamos. Os parâmetros foram:

- D1 = Parada de segurança (só permite o funcionamento do motor, se estiver em alta)
- D2 = Aumenta frequência de alimentação
- D3 = Diminui frequência de alimentação
- D4 = Inverte o sentido de giro do motor

Esse IF também permite o controle da velocidade através de um I/O analógico. É possível fazer o *input* de uma tensão de 0 a 12 v, resultando na alteração da frequência de alimentação da frequência mínima até a máxima parametrizada respectivamente (WEG®). Optamos por usar os controles via porta digitais, pois como um dos objetivos foi criar um módulo que permita ser facilmente adaptado para outras esteiras, não são todos os IF que possuem esse I/O analógico.

Outro IF usado foi o CFW500, dimensionado para trabalhar com motores de até 28 A, o suficiente para controlar motores de até 10cv. O IF CFW500 possui uma I/O analógica que pode ser parametrizada para ser uma referência de frequência do motor, ou seja, ele recebe um sinal que vai de 0 a 12 v, e pode ser parametrizado para controlar a frequência de alimentação do motor. Inicialmente utilizamos esse método para fazer o controle via placa controladora ESP32, ou mesmo o Arduino para enviar um sinal PWM para essa porta analógica. Como o nível lógico do ESP32 é 3.3 v e o do Arduino é 5 v, tivemos de utilizar um parâmetro de ganho dentro do próprio IF para chegar nos 12 v do IF CFW500. Apesar dessa lógica ser mais simples, optou-se por mudar pois não é todo IF de frequência que possui esse recurso.

Configuramos os parâmetros desse IF para trabalhar com o motor original da esteira e também com o motor de 10 cv. Notamos uma diferença no tempo de resposta nas entradas digitais, de forma que o CFW08 precisa de aproximadamente 20 milissegundos (ms), enquanto o CFW500 precisa de apenas 7ms para ativar de forma garantida, o que afetou um pouco a

programação do módulo de controle. Foi necessário dar um tempo de espera entre cada acionamento das I/O digitais; dessa forma, tanto o IF como a placa controladora (ESP32) se mantêm sincronizadas com o mesmo valor de frequência.

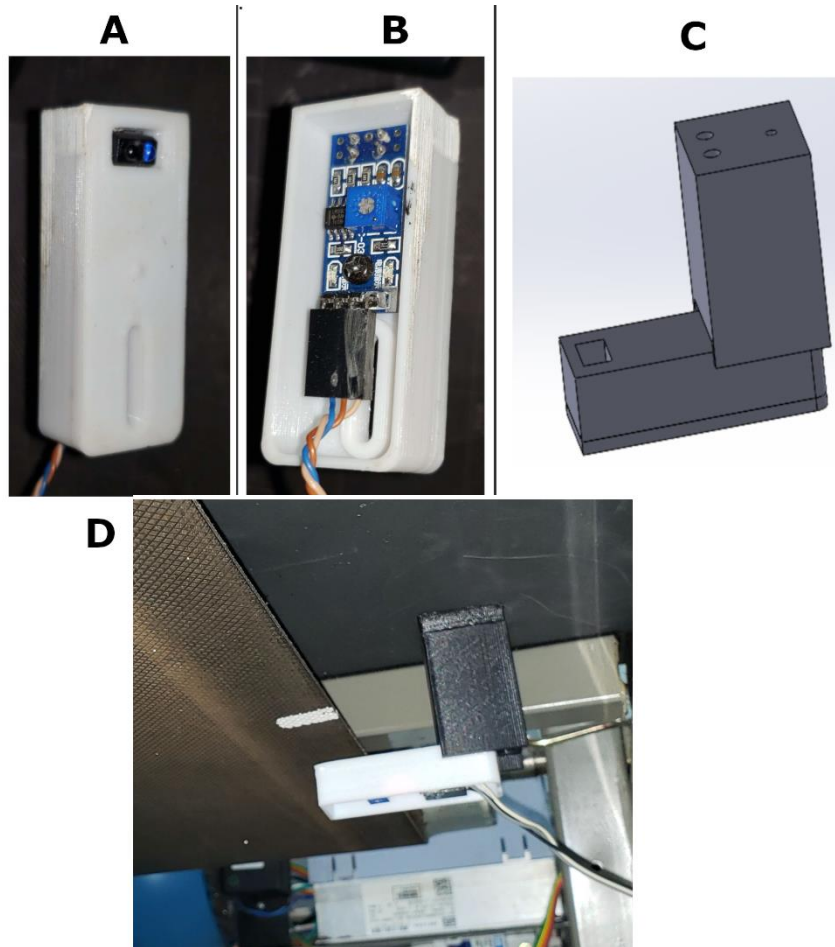
4.3 Velocidade (sensor ótico)

As esteiras disponíveis no mercado controlam e mensuram a velocidade no eixo de rotação do motor. Este procedimento pode apresentar um valor de erro sistemático (*bias*) considerável para testes de alto desempenho, pois essa é uma medida indireta da velocidade linear da lona da esteira, que em muitos casos apresenta uma frenagem em cada contato do pé durante o ciclo de corrida. Uma forma de solucionar esse problema é medir a velocidade diretamente na lona da esteira.

Para a construção do sensor ótico, preferimos utilizar um módulo TCRT5000 (figura 10 (a) e (b)). Além de fácil de ser construído, o custo do módulo é muito baixo, inferior a R\$10,00 (dez reais)⁸. Fabricamos uma case em impressão 3D (figura 10 (c)) para acomodar o módulo que foi instalado embaixo da esteira (figura 10(d)), a uma distância de aproximadamente 5mm da lona. Os cabos foram trançados para minimizar o risco de interferência externa. O módulo tem a facilidade de apresentar um potenciômetro que tem a função de regular o *Schmitt Trigger* e enviar as leituras de forma binária para o microcontrolador. Na programação do microcontrolador, é necessário tomar o cuidado de utilizar as interrupções para que as leituras do sensor ótico tenham prioridade e não ocorra falhas pelo fato do microcontrolador estar realizando outra tarefa.

⁸ Custo baseado em pesquisa feita em dezembro de 2020

Figura 10: Sensor ótico



Na esteira adaptada, marcamos a lona com uma caneta permanente de cor branca, da marca Pentel®, modelo Paint Marker, com um espaçamento de 288mm. A luz infravermelha emitida pelo módulo é refletida pelas marcas brancas na lona da esteira e o fototransistor presente no TCRT5000 envia um sinal para a placa controladora (ESP32).

Inicialmente usamos uma placa de desenvolvimento Arduino Nano, e a leitura era feita por meio de uma interrupção que está relacionada ao I/O digital 2. Como além da medida do sinal do sensor ótico o microcontrolador precisava também controlar o inversor de frequência e se comunicar com o computador (receber e enviar dados via serial), mudamos a placa para o nodeMCU ESP32, que apresenta dois núcleos de processamento e consegue lidar com dois processamentos simultaneamente de forma assíncrona, assim conseguimos trabalhar com todas tarefas constantemente. O nodeMCU ESP32 permite que praticamente todos os I/O possam ser usados nas interrupções, portanto usamos o I/O 26. Entre outras vantagens, é possível

configurar a função interrupção como IRAM_ATTR, que faz com que esse trecho do código fique armazenado no barramento de instruções da memória RAM, e não na memória Flash, o que aumenta a velocidade.

Em ambos os casos utilizamos a configuração de interrupção como sendo a borda de descida do sinal do sensor ótico (FALLING). Utilizamos duas formas de calcular a velocidade. Na primeira delas, a cada interrupção uma variável contadora incrementava um; no loop principal a cada 1000ms, o contador é zerado. O valor do contador representa a frequência em Hz e outra variável soma o número total de incremento da sessão; desses valores é possível calcular a velocidade e a distância percorrida de forma bem simples com nas fórmulas abaixo:

$$Vel = Freq(Hz) * Dist(m)$$

Onde:

Vel = velocidade em m/s

Freq = n° de counts a cada segundo

Dist = distância entre cada marcador na lona da esteira.

$$D = SC * Dist(m)$$

Onde:

D = distância total percorrida em metros

SC = soma total do n° de counts

Dist = distância entre cada marcador da lona da esteira.

O inconveniente desta forma é que necessário aguardar um tempo e como o número de leituras apresentava frequência baixa, acabava gerando muitas oscilações na resultante do cálculo de velocidade, além de uma baixa taxa de atualização da leitura.

Para contornar o problema, adotamos a estratégia de medir o tempo entre duas ou mais leituras geradas pelas interrupções, e dessa forma conseguir atualizações mais frequentes e taxa mais elevadas de atualização.

Outro cuidado que tomamos foi usar a estratégia de debounce, para evitar leituras maiores do que a correta, em função de oscilação do sinal dos sensores óticos (DE ALMEIDA; DE

MORAES; SERAPHIM, 2017). Abaixo segue parte dos códigos utilizado no ESP32, o primeiro a interrupção e o segundo a forma que usamos para calcular a velocidade da lona da esteira.

```
1. #define debounceOtico 3450 // define tempo de debounce em microseg
2. //Função interrupção
3. portMUX_TYPE mux = portMUX_INITIALIZER_UNLOCKED;
4. //IRAM_ATTR para rodar na memoria ram, ao invés da flash
5. static unsigned long lastInterrupt = 0;
6. unsigned long interruptTime = 0;
7. void IRAM_ATTR interruptHz() {
8.
9.     interruptTime = micros();
10.    if(interruptTime - lastInterrupt > debounceOtico){
11.        portENTER_CRITICAL_ISR(&mux);
12.        countSensorOtico++;
13.        portEXIT_CRITICAL_ISR(&mux);
14.    }
15.    lastInterrupt = interruptTime;
```

```
1. //função que faz leitura da velocidade
2. #define time2zeroOtico 5000//define tempo de espera para próx leitura
3. #define nOtico 2 // define n de leituras para calcular velocidade
4. int Hz;
5. volatile int countSensorOtico;
6. unsigned long dist;
7. unsigned long timePreOtico = 0;
8. unsigned long timeDeltaOtico = 0;
9. void Speed(){
10.   int countExt = 0;
11.   portENTER_CRITICAL_ISR(&mux); // início da seção crítica
12.     countExt = countSensorOtico;
13.   portEXIT_CRITICAL_ISR(&mux); // fim da seção crítica
14.   if(countExt >= nOtico){
15.     timeDeltaOtico = millis() - timePreOtico;
16.     timePreOtico = millis();
17.     Hz = (nOtico * 100000.00 ) / timeDeltaOtico;
18. // valor irá com multiplo de 100
19.     dist += nOtico;
20.     vel =( Hz * 36.00 ) * ( passo / 1000.00 ) ;
21.     portENTER_CRITICAL_ISR(&mux); // início da seção crítica
22.       countSensorOtico = 0;
23.     portEXIT_CRITICAL_ISR(&mux); // fim da seção crítica
24.   }
25.   if(millis() - timePreOtico > time2zeroOtico){
26.     Hz = 0;
27.     vel = 0;
28.   }
29. }
```

4.4 Construção da placa de controle

Para a construção da placa de controle geral e conexão com o computador, inicialmente utilizamos o microcontrolador Arduino Nano. No decorrer do projeto, mudamos para o ESP32 figura 11 (a)), pois o mesmo apresenta performance muito superior e o custo é muito semelhante. A principal característica do ESP32 é possuir dois núcleos de processamento, os seja, ele consegue executar duas tarefas ao mesmo tempo. Desta forma conseguimos fazer as leituras, os cálculos, enviar os dados via serial e enviar os comandos para o inversor de frequência sem grandes dificuldades. Também montamos uma caixa em impressão 3d, para acomodação da placa de forma segura (figura 11 (b)). Para projetar a placa usamos o software Eagle (Autodesk®), juntamente com o plugin PCBgcode (LUCAS, 2019), para desenhar a placa de circuito impresso (PCB) e gerar o código para fresamento da PCI (figura 12). Utilizamos o software bCNC (VLACHOUDIS, 2017), para interpretar o código gerado pelo PCBgcode, e comandar a fresadora CNC projetada para o estudo. A placa foi fresada em uma fresadora CNC, que montamos baseadas online (figura 11 (c)) (CONRRADO., 2019).

Figura 11: Espr32(a), modelo 3d caixa(b), fresamento da placa de circuito impresso(c).

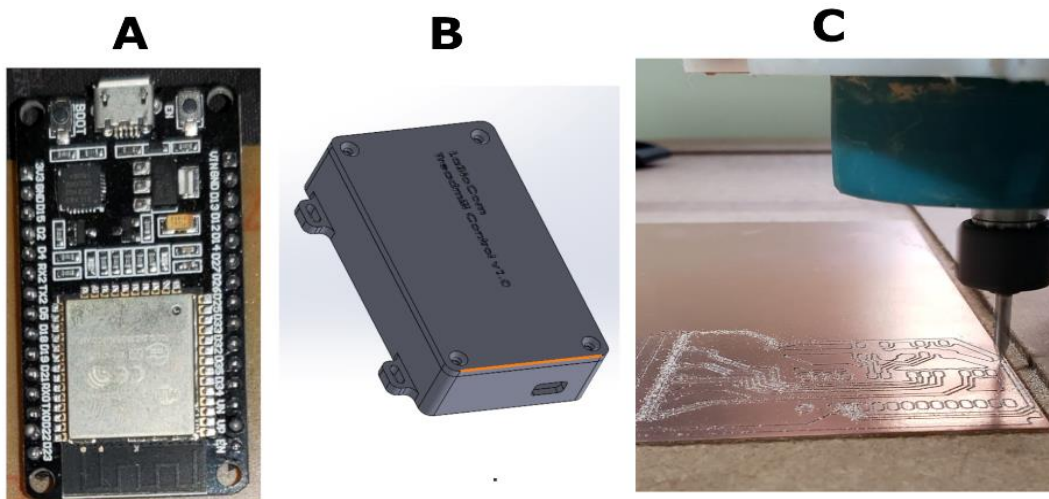
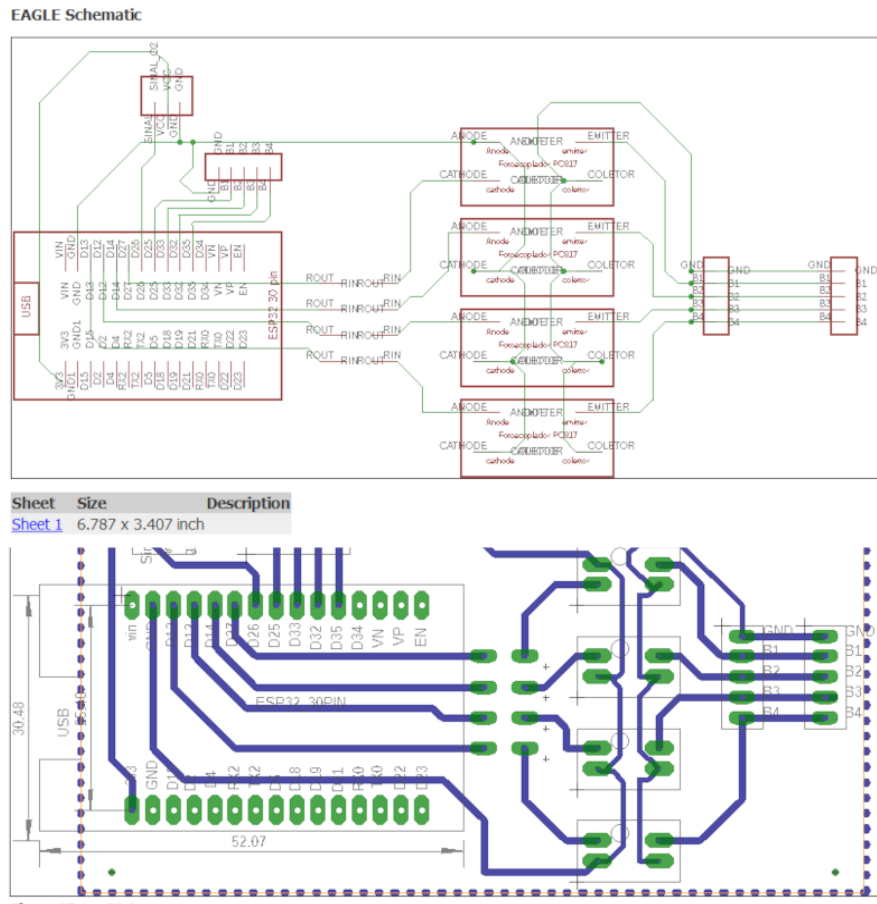
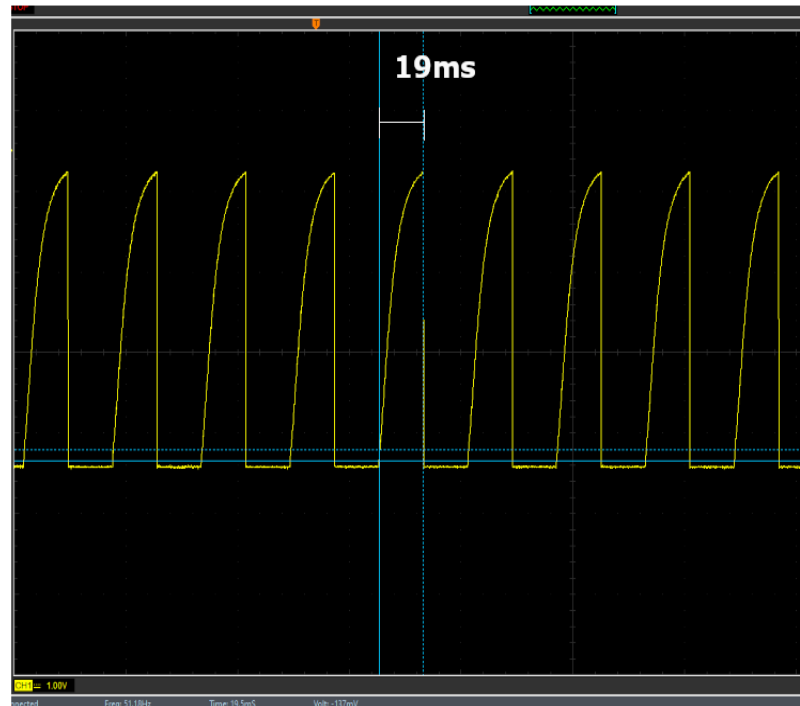


Figura 12: Projeto placa controlador usando ESP32



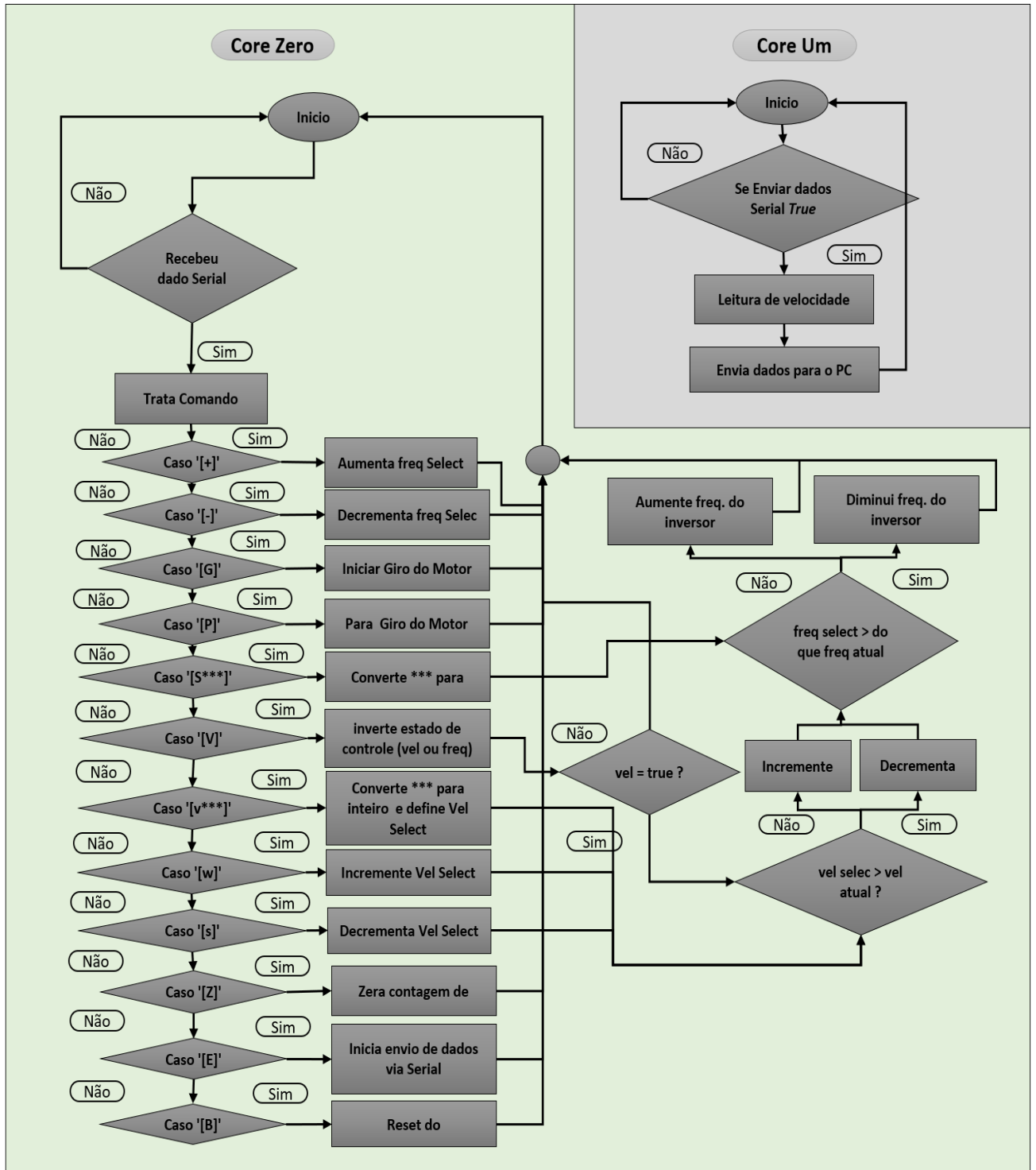
A interface do ESP32 com o inversor de frequência, foi realizada através das portas digitais do ESP32 (D12, D13, D14 e D27), com as portas digitais do CFW08 ou CFW500 (D1, D2, D3 e D4). Como o CFW08 trabalha normalmente em alta e o nível lógico do inversor é de 12v e o ESP32 tem nível lógico de 3.3 v, utilizamos um fotoacoplador PC817 da Sharp® para comunicação entre eles. Quando um I/O digital do ESP32 é colocado em alta, um LED dentro do encapsulamento do fotoacoplador é acionado, o que excita o fotorreceptor interno ao fotoacoplador, fechando o contato entre os dois pinos de saída do fotoacoplador. Dessa forma o ESP32, com nível lógico de 3,3 v, fica separado do CFW08, que tem nível lógico de 12 v. O fotoacoplador PC817 tem uma resposta que não forma uma onda quadrada exata (figura 13), por esse motivo é necessário um tempo de espera para que o acionamento do inversor de frequência seja realizado com sucesso.

Figura 13: Resposta do fotoacoplador forma uma onda que não é quadrada, por volta de 19ms



Como o microcontrolador usado possui dois núcleos de processamentos, é possível trabalhar de forma assíncrona. Utilizamos um dos núcleos para receber e tratar os dados oriundos da serial e comandar o inversor de frequência, já o outro núcleo ficou responsável por fazer as leituras do sensor ótico e enviar os dados para o computador, via porta serial. Segue fluxograma geral da programação da placa de controle (figura 14).

Figura 14: Fluxograma da programação da placa controladora



Como visto no fluxograma, a placa controladora responde a alguns comandos recebidos via serial a partir do software controlado pelo usuário. Esses comandos são tradados, como por exemplo, no trecho de código abaixo:

```
1. /*[+] incremente freq
2. *[-] decrementa freq
3. *[G] gira motor
4. *[P] para motor
5. *[S] coloca freq de 0 a 1000, motor seria o int de 0 a 1000
6. *[Z] zera referência de tempo
7. *[B] reseta via software
8. While(true){
9.     if(Serial.available()){
10.        texto[c_texto] = Serial.read();
11.        if(texto[0] != '[') c_texto = 0;
12.        else{
13.            if(texto[c_texto] == ' '){
14.                c_texto = 0;
15.                if(texto[0] == '[' && texto[2] == ' '){
16.                    switch(texto[1]){
17.                        case '+':
18.                            incremente(5); // incremente 0,5Hz no inversor
19.                            break;
20.                        case '-':
21.                            decremente(5); // decrementa 0,5Hz no inversor
22.                            break;
23.                        case 'G':
24.                            digitalWrite(pinGiraPara, HIGH);
25.                            flagMotor = 1;
26.                            break;
27.                        case 'P':
28.                            digitalWrite(pinGiraPara, LOW);
29.                            flagMotor = 0;
30.                            break;
31.                        case 'S':
32.                            valFreqSelect = Serial.parseInt();
33.                            velFlag = false;
34.                            if(freqSelect < valFreqSelect){
35.                                int diffFreq = valFreqSelect - freqSelect;
36.                                incremente(diffFreq);
37.                            }
38.                            else{
39.                                int diffFreq = freqSelect - valFreqSelect;
40.                                decremente(diffFreq);
41.                            }
42.                            break;
```

4.5 Coleta de dados de força de reação do solo (FSR)

4.5.1 Células de carga usadas

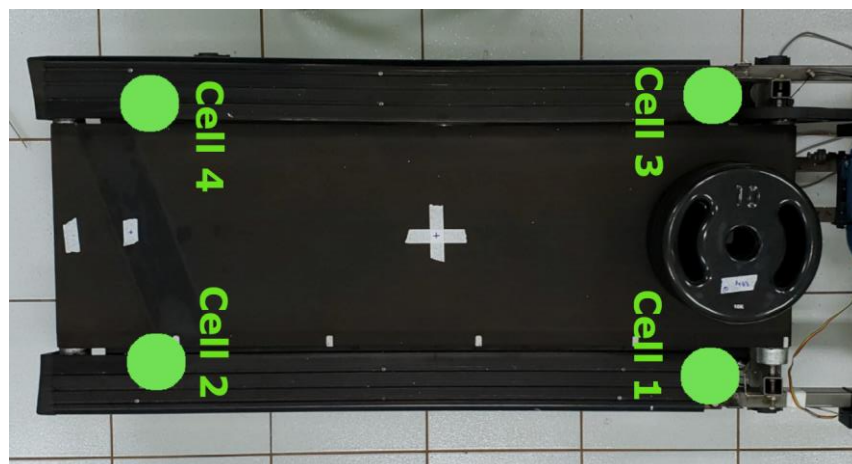
A medida das forças verticais é uma das informações mais relevantes a ser retirada do sistema. Os valores de forças verticais servem de base para diversos cálculos de análises que permitem identificar o padrão de corrida e marcha. Para isso, foram utilizadas quatro células de cargas CTR 1 toneladas força (MK Controle e Instrumentação LTDA - São Paulo, Brasil) dispostas na base da esteira.

Para proteger as células de carga e garantir que as mesmas fiquem em paralelo com o piso, foram confeccionados apoios articulados para serem instalados entre a esteira e as células de carga (figura 15). As células foram dispostas conforme visto na figura 16.

Figura 15: Suporte das células de carga.



Figura 16: Localização de cada célula na base da esteira.



4.5.2 Amplificadores

Como o sinal da célula de carga é pequeno, utilizamos em cada célula de carga um amplificador da marca MKcontrole, modelo MKTC-05. Esse amplificador permite o ajuste de offset e SPAN (ajustados por meio de trimpots), além de já possuir fonte de alimentação interna. Os condicionadores de sinal foram dispostos de uma caixa e os conectores fêmeas do tipo Mike de 4 vias foram instaladas para facilitar a conexão com as células de carga e também a utilização do condicionado de sinal em outros projetos (figura 17).

Outro condicionador de sinal foi montado utilizando quatro amplificadores instrumentais do tipo INA125 (Texas Instruments Incorporated®) (FIALHO, 2002); FIALHO; DA CUNHA; HIRT; INSTRUMENTAÇÃO *et al.* (2014).

Figura 17: Caixa com os plugs do tipo MIKE para as 4 células de carga, os 4 amplificadores MKTC-05 e o DAQ NI USB6009.



4.5.3 Conversor Analógico/Digital

Como o sinal gerado pelas células de carga e condicionado pelos amplificadores é analógico, para que possamos trabalhar com esses dados foi necessário convertê-lo para digital.

O próprio microcontrolador possui 18 entradas analógicas de 12 bits, ou seja, permite a leitura de 2^{12} valores diferentes, ou seja 4096. Apesar de apresentar uma boa performance geral, obtivemos baixa performance na transmissão desses dados para o computador, possivelmente pelo protocolo de comunicação desenvolvido.

Para facilitar o processo utilizamos o USB data acquisition card da National Instruments USB-6009 (DAC NI USB6009). O mesmo apresenta oito canais de entrada analógica de 14bits, portanto apresenta a possibilidade de 16384 possibilidades, 4 vezes mais do que o conversor do ESP32. Além disso apresenta uma taxa de amostragem de 48khz, e algumas bibliotecas que facilitam a programação de rotinas em linguagens como a do Matlab e Labview para coleta desses dados. Para que o dispositivo seja reconhecido pelo computador foi necessário utilizar o driver fornecido pelo próprio fabricante, o NI-DAQmx (NATIONAL-INSTRUMENTS®, 2020).

4.5.4 Software coleta de dados

Desenvolvemos uma função utilizando o MatLab para coletar os dados do DAQ NI USB6009: processar o sinal, salvar os dados e plotar gráficos das variáveis de interesse. A função é determinada por uma sobrecarga que corresponde ao tempo em segundos que deseja coletar. Uma sessão com o DAQ NI USB6009 é criada com 4 canais analógicos, em seguida é perguntado para o usuário se deseja realizar a tara das leituras das células; caso responda afirmativamente, é realizada a coleta de 2 segundos, o sinal é filtrado e o valor de média de cada célula é armazenando em uma matriz de tara. É perguntado para o avaliador se deseja continuar com as coletas, se responder positivamente, é coletado o tempo informado na sobrecarga da função inicial. Dos dados coletados, é passado o filtro, subtraído o valor, adicionado o valor referente a soma de trocas as células, desses dados calcula-se o centro de pressão. O fluxograma representado na figura 18 representas os passos do processo de coleta de dados.

Em um estudo uma fórmula para cálculo de COP foi sugerida para uma plataforma de equilíbrio (HUANG; SUE; ABBOD; JIANG *et al.*, 2013), porém como a esteira tem diversas cargas sobre ela e o sistema de aquisição de dados montado com o DAQ NI USB6009 também pode ser utilizado para outros tipos de coletas, desenvolvemos uma rotina de calibração que visa saber o quanto cada célula é deformada quando uma carga é aplicada no centro da esteira. Com as porcentagens de contribuição de cada célula feitas, podemos calcular o centro de pressão da seguinte forma:

$$cell(x) = \frac{Soma}{valCell(x)}$$

Onde:

cell(x) = valor normalizado da célula de carga x

soma = soma da leitura das 4 células de carga em volts

valCell(x) = valor de leitura da célula x em volts

$$COPx = \left(\left(\frac{cell(3) + cell(4)}{2} \right) - \left(\frac{cell(1) + cell(2)}{2} \right) \right) * LX$$

Onde:

COPx = eixo x do centro de pressão

cell(x) = valor normalizado da leitura da célula x (obtida na equação anterior)

LX = comprimento do eixo x em cm (largura da lona da esteira).

$$COPy = \left(\left(\frac{cell(3) + cell(1)}{2} \right) - \left(\frac{cell(4) + cell(2)}{2} \right) \right) * LY$$

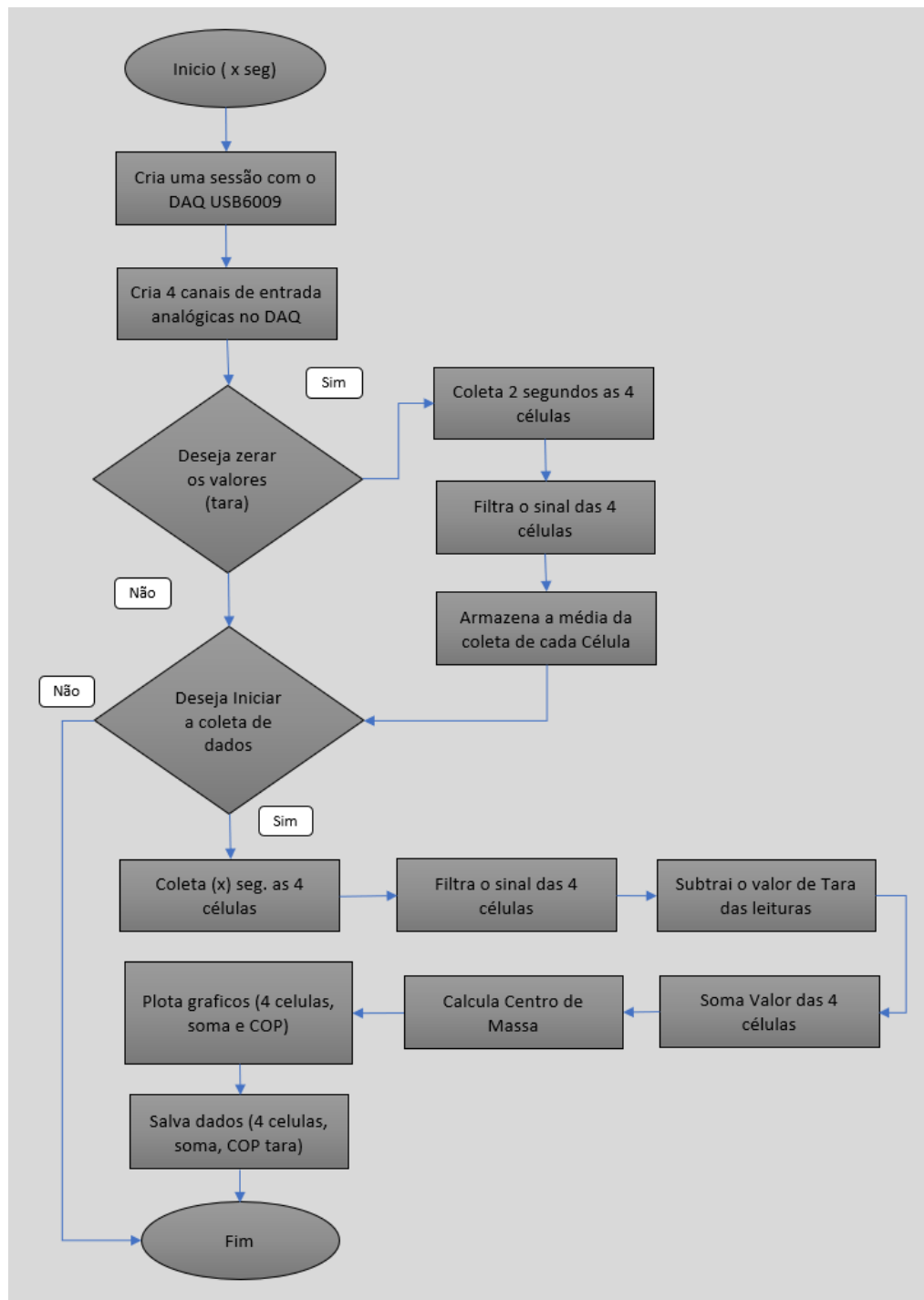
Onde:

COPy = eixo y do centro de pressão

cell(x) = valor normalizado da leitura da célula x (obtida na equação anterior)

LY = comprimento do eixo y em cm (largura da lona da esteira).

Figura 18: Fluxograma da coleta e processamento dos dados das células de carga executado no MatLab.



4.6 Software de controle da esteira

Para desenvolvimento do software de controle da esteira, utilizamos o software Visual Studio da empresa Microsoft®, na versão Community e a linguagem C#. A escolha foi feita devido às facilidades obtidas no desenvolvimento de interfaces gráficas e a sua similaridade com a linguagem C e C++. O fluxograma abaixo demonstra a versão do software atual.

Na primeira página do software temos alguns locais para selecionar a porta de comunicação serial, onde o ESP32 da placa controladora esteja conectado, e também a velocidade de comunicação serial em que a placa controladora se comunica com o computador (baudrate) (figura 19). Após conectar, outro botão dentro do software deve ser pressionado para que a placa controladora inicie o envio de dados para software no computador.

Figura 19: Software Visual Studio com a primeira página da interface gráfica do software de controle da esteira.

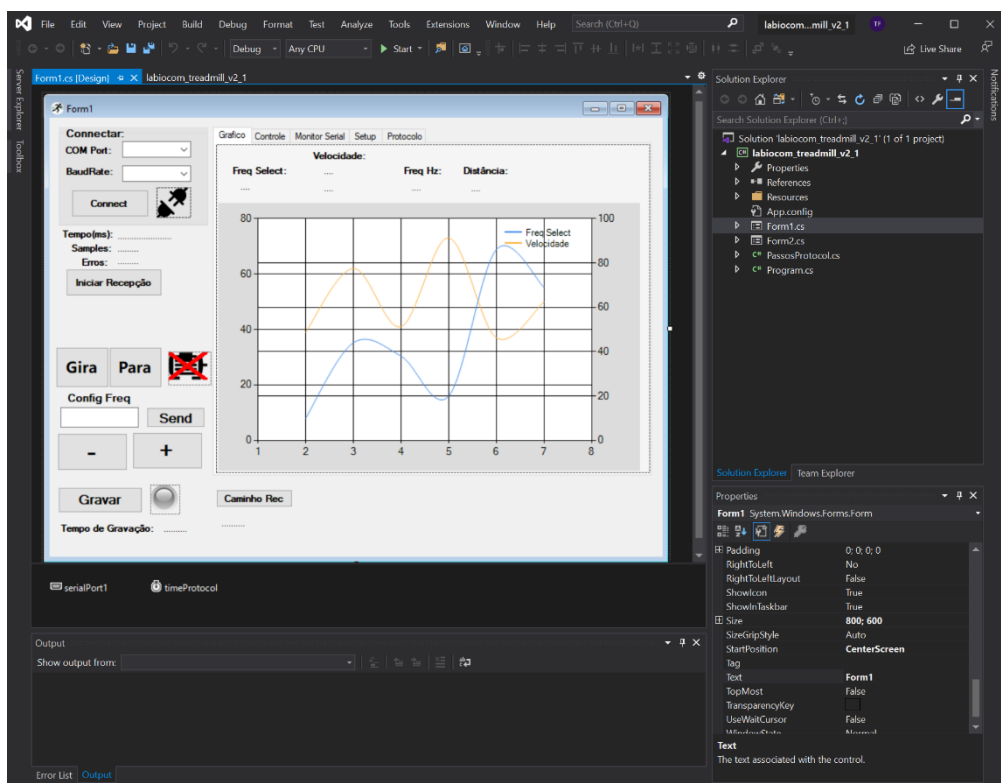
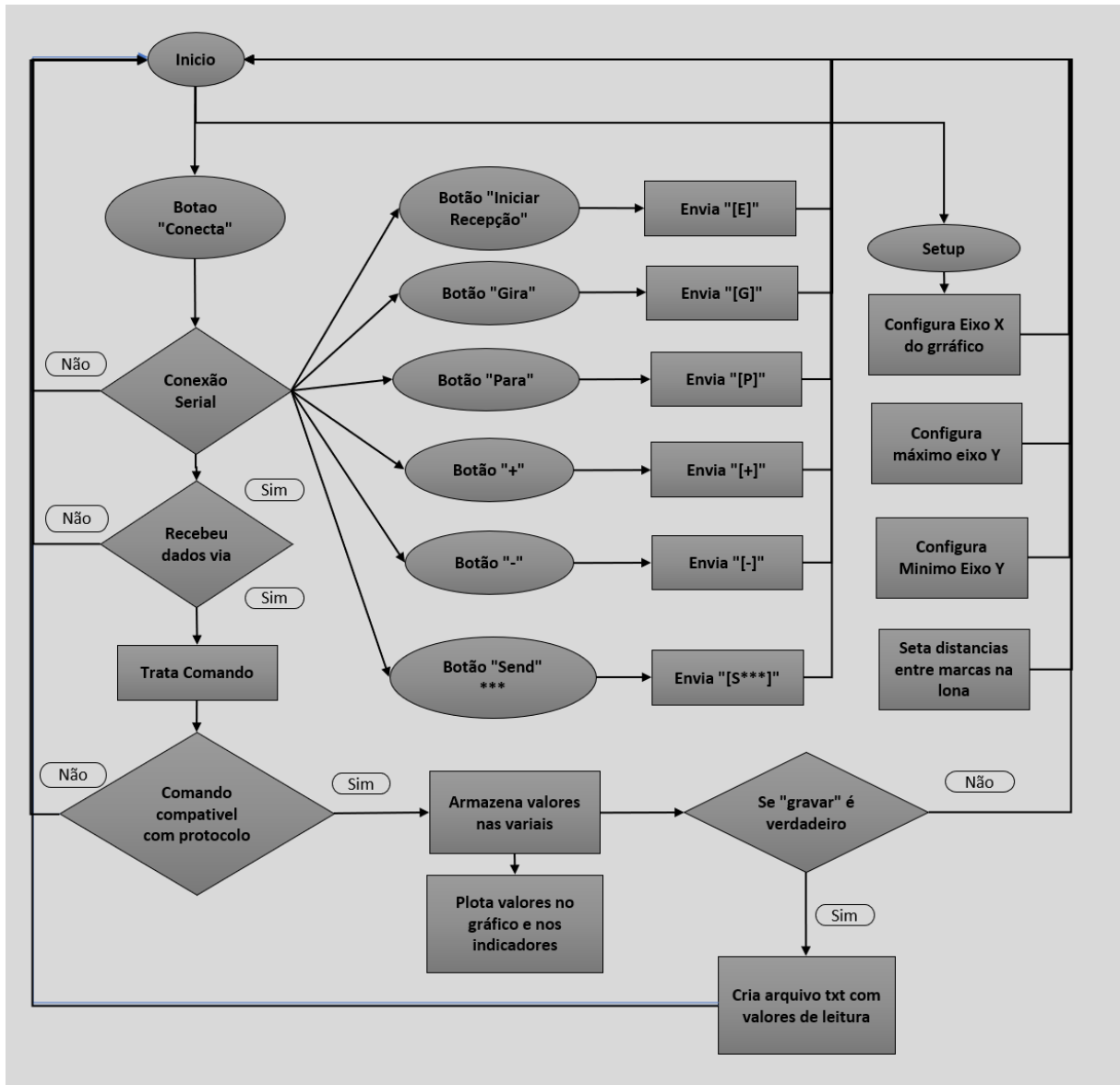


Figura 20: Fluxograma software de controle



Uma função foi criada para registrar toda vez que um dado seja recebido pela c serial, o dado recebido é tratado e salvo em suas variáveis específicas. Para isso uma outra função do tipo *delegate* é criada, desta forma o evento de receber dados ocorre, os dados recebidos são direcionados para a função que irá tratá-los conforme suas especificidades. O fluxograma esboçado na figura 20 demonstra o fluxo do programa criado para controle e coleta de dados de velocidade da esteira.

Para que os dados recebidos sejam organizados, o tratamentos dos mesmos segue os seguintes passos: os dados são lidos do buffer da serial, ao testarmos verificamos que o buffer deve ter pelo menos 60 caracteres, se não tiver ele retorna e soma mais dados, caso tiver, é testado se o buffer recebido tem no índice 0 a string “[“, no índice 1 a string “T”, no índice 12, a string “F”... e no índice 60 a string “]”, no final espera-se ter a seguinte string:

```
[T000005847,F0000,D000000,R00000,L00288,V00000,v00000,M0,C0,]
```

O primeiro grupo de caracteres está relacionado ao tempo decorrido no microcontrolador, o segundo à frequência de alimentação do motor, o terceiro à distância percorrida, o quarto ao número de leituras do sensor óptico por segundo, o quinto ao comprimento do passo (espaço entre cada marca na lona em mm), o sexto à velocidade lida, o sétimo à velocidade selecionada pelo usuário, o oitavo ao estado motor (1 ligado, 0 desligado) e o nono ao tipo de controle (1 para velocidade e 0 para frequência do inversor). Abaixo segue um trecho do código como exemplo:

```

1. //=====Função recepção
2.     public void RecepcaoSerial()
3.     {
4.         try
5.         {
6.             buffStr += serialPort1.ReadExisting();
7.             if (buffStr.Length < 61)
8.             {
9.                 return;
10.            }
11.            else
12.            {
13.                if (cbMonitorSerial.Checked == true)
14.                {
15.                    txtReceive.Text += buffStr;
16.                    //print valor lido da serial no txtReceive
17.                    if (txtReceive.Text.Length > 800) // zera
18.                        a caixa de texto Receiver, quando estiver cheia
19.                    {
20.                        txtReceive.Clear();
21.                    }
22.                    if (buffStr.Substring(0, 1).Equals("[") &&
23.                        buffStr.Substring(1, 1).Equals("T") &&
24.                        buffStr.Substring(12, 1).Equals("F") &&
25.                        buffStr.Substring(18, 1).Equals("D") &&
26.                        buffStr.Substring(26, 1).Equals("R") &&
27.                        buffStr.Substring(33, 1).Equals("L") &&
28.                        buffStr.Substring(40, 1).Equals("V") &&
29.                        buffStr.Substring(60, 1).Equals("]")
30.                    )
31.                    {
32.                        if (buffStr.Length >= 61)
33.                        {
34.                            str = buffStr;
35.                            // tempo / freq / dist / hz / passo /
36.                            vel / velsel / motor / tipo de controle
37.                            //      //
38.                            [T000025779,F0000,D000000,R00000,L00288,V00000,v00000,M0,C0,]
39.                            (...)
40.                        }
41.                    }
42.                }
43.            }
44.        }
45.    }

```

Com base na frequência de leituras do sensor óptico recebidas, calculamos a velocidade de deslocamento da lona da esteira, baseado na distância de cada marcador, no caso 288mm. Essa distância é setada pelo usuário na parte de configurações do software de controle da esteira; abaixo a fórmula de como calcular a velocidade de deslocamento da lona.

$$Vel = Fq * \left(\frac{Passo}{1000} \right)$$

Onde:

Vel = velocidade em m/s

Fq = frequência de leituras por segundo do sensor óptico

Passo = distância entre cada marca na lona da esteira em mm.

Utilizando o botão gravar, é possível gravar os dados em um arquivo no formato “.txt” para consultas futuras, nesse caso um novo cronômetro é iniciado.

4.6.1 Controle de velocidade

O software pode controlar a velocidade da esteira de duas formas: através da modificação da referência de frequência de alimentação do motor ou através da velocidade de referência da lona. A placa controladora compara o valor da frequência de alimentação do inversor desejada (enviada pelo software), com a que estava no momento, caso estivesse menor, o I/O responsável por aumentar a frequência de alimentação do inversor alterna entre alta e baixa, até atingir a frequência desejada, caso estiver maior, o I/O responsável por diminuir a frequência alterna entre alta e baixa, até chegar na frequência desejada. Para o controle com base na velocidade ocorria o mesmo, nesse caso a velocidade da lona é responsável por aumentar ou diminuir a frequência de alimentação do motor.

4.6.2 Controle baseado na frequência de alimentação do inversor

A forma mais simples de controlar a esteira é enviando sinal para as entradas digitais do inversor de frequência, responsável por elevar ou diminuir a frequência de alimentação do motor. A placa controladora, poderia receber a *string*:

- [+], para elevar a freq. do inversor.
- [-], para diminuir a freq. do inversor.
- [S] + valor interior de 0 a 1000, para colocar a freq. entre 0 e 100Hz

Na placa controladora, o código abaixo foi usado no núcleo responsável pelo controle do IF:

```

1. // função que incrementa valor de frequência no inversor de
   frequência
2. void incremente(int val){
3.     countMais = 0;
4.     for(int i = 0; i <= (val - 1); i++){
5.         if(freqSelect < 1000){
6.             digitalWrite(pinMais, HIGH);
7.             vTaskDelay(timeBotDown);
8.             digitalWrite(pinMais, LOW);
9.             vTaskDelay(timeBotUp);
10.            freqSelect += 1;
11.        }
12.    }
13. }
14.
15. //Função que decrementa valor de freq no inversor de frequência
16. void decrete(int val){
17.     countMenos = 0;
18.     for(int i = 0; i <= (val - 1); i++){
19.         if(freqSelect > 0){
20.             digitalWrite(pinMenos, HIGH);
21.             vTaskDelay(timeBotDown);
22.             digitalWrite(pinMenos, LOW);
23.             vTaskDelay(timeBotUp);
24.            freqSelect -= 1;
25.        }
26.    }
27. }
28.

```

A função tanto de incremento como de decremento, recebe uma sobrecarga, que é o número de acionamento do pino de aumento ou de redução de frequência de alimentação do inversor que se deseja. Entre cada mudança de estado dos I/O de entrada ou de saída é colocado um tempo de espera para que se tenha uma resposta efetiva dos fotoacopladores, resultado no acionamento dos I/O responsáveis pelo aumento ou redução da frequência no IF.

4.6.3 Controle baseado na velocidade

Outra forma de controlar a esteira é por meio da definição de uma velocidade e por meio do *feedback* das leituras da velocidade, onde a placa controladora decide se aumenta ou diminui a freq. de alimentação do inversor. A placa controladora pode receber as *strings*:

- [V], para ativar ou desativar o modo de controle baseado na velocidade.
- [w], para aumentar a velocidade alvo.
- [s], para diminuir a velocidade alvo.
- [v] + valor inteiro entre 0 e 200, para definir a velocidade entre 0 e 20km/h.

Definimos uma faixa de aceitação de velocidade por comandos citados acima, e um tempo de espera antes de fazer novas medições e decidir se aumenta ou diminui a freq. No caso usamos como tempo de espera 50ms e margem de velocidade aceitável sendo 0,2km/h para mais ou para menos. Essa função acaba utilizando a função anterior para aumentar ou diminuir a freq. do inversor. Segue uma parte do código.

```
int ePasso[5] = {0, 0, 0, 0, 0}; //compr das marcas na esteira salva na
//eeprom
int passo;
int vel;
int velFlag = 0;
int velSelect;
#define timeDelayVel 50 // define tempo de resposta das leituras de
//velocidade
#define velMargem 200 //define margem para segurar vel 1000=1km/h
unsigned long timePreVel = 0;

// função que autoregula valor de alimentação do motor com base na
velocidade da lona da esteiravoid controlSpeed(){
    if(velFlag == 1){
        if(velSelect > vel - velMargem && millis() - timePreVel >
timeDelayVel){ // valor com uma decima, porem sem a virgula, 0,5 de
//margem
            timePreVel = millis();
            incremente(5);
        } |
        if(velSelect < vel + velMargem && millis() - timePreVel >
timeDelayVel){ //valor com uma decima, porem sem a virgula, 0,5 de margem
            timePreVel = millis();|
            decremente(5);
        }
    }
}
```

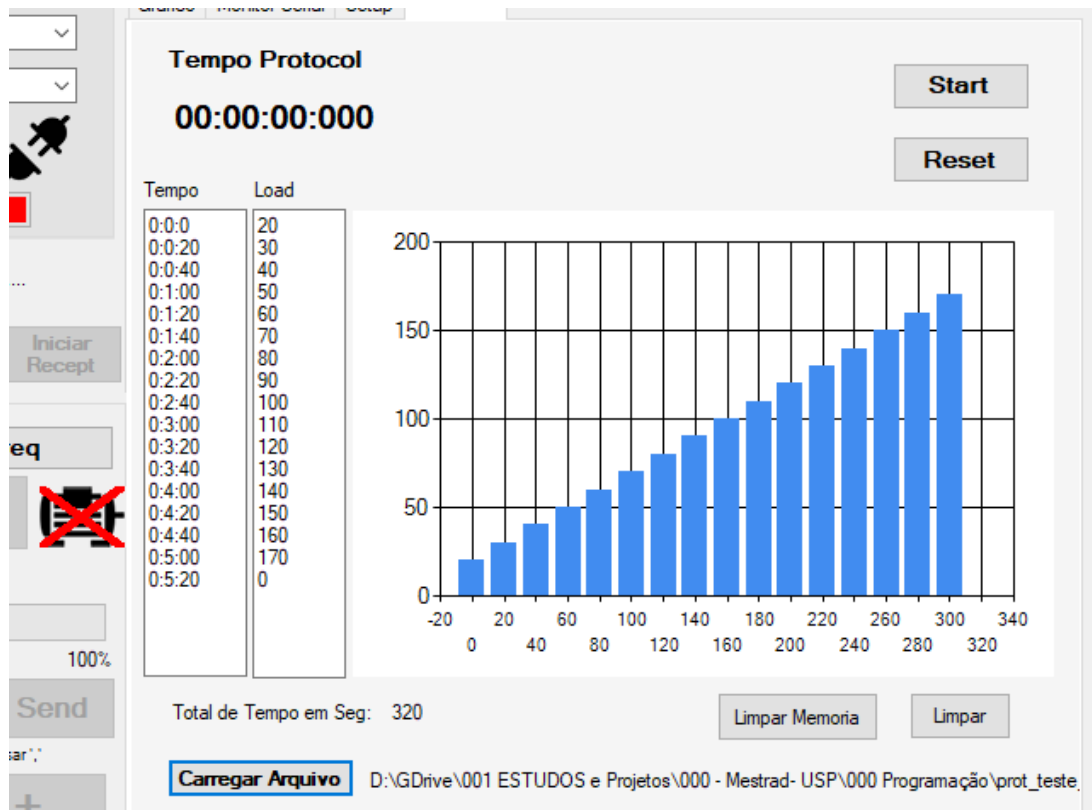
4.6.4 Realização de protocolos predefinidos

É importante para uma esteira poder criar protocolos predefinidos, seja para avaliação ou para treinamento. Para isso o usuário pode carregar um arquivo de texto com um formato específico, discriminando o tempo desejado para modificar as cargas, seja ela baseada na freq. de alimentação do inversor ou na velocidade da lona da esteira.

O arquivo de texto deve conter as seguintes informações separadas por ‘, ‘: horas, minutos, segundos, carga e tipo de carga. Inclusive, o usuário pode misturar o controle de carga baseado em velocidade ou em freq. de alimentação. Segue um exemplo de protocolo onde a cada 20seg, 20km/h são elevados e no 5’20’’ muda o controle para freq. e zera a freq. de alimentação. Concluindo o protocolo o usuário pode salvar os dados em um arquivo de texto para futuras análises. Abaixo temos um exemplo de código interpretado pelo software para execução de um protocolo de avaliação e na figura 21, os dados do mesmo protocolo já carregados.

```
1. 0,0,0,20,v
2. 0,0,20,30,v
3. 0,0,40,40,v
4. 0,1,00,50,v
5. 0,1,20,60,v
6. 0,1,40,70,v
7. 0,2,00,80,v
8. 0,2,20,90,v
9. 0,2,40,100,v
10. 0,3,00,110,v
11. 0,3,20,120,v
12. 0,3,40,130,v
13. 0,4,00,140,v
14. 0,4,20,150,v
15. 0,4,40,160,v
16. 0,5,00,170,v
17. 0,5,20,0,f
18.
```

Figura 21: Exemplo de protocolo carregado do texto acima (incrementando 1km/k a cada 20seg).



4.7 Custo para desenvolvimento do protótipo

Tabela 2: Custo final

Custo para produção do protótipo e testes			
Item	qt	val. Uni.	valo total
Motor Trifásico Alta Rotação 10cv	1	R\$ 2.873,52	R\$ 2.873,52
Inversor de Freq. Weg CFW 500	1	R\$ 3.069,00	R\$ 3.069,00
DAQ NI USB 6009	1	R\$ 1.300,00	R\$ 1.300,00
Célula de carga CTR 1t	4	R\$ 1.140,00	R\$ 4.560,00
Amplificador MK controle	4	R\$ 475,00	R\$ 1.900,00
plug mike 4 vias	4	R\$ 11,50	R\$ 46,00
Microcontrolador NodeMCU esp32	2	R\$ 54,99	R\$ 109,98
Modulo sensor ótico	2	R\$ 18,90	R\$ 37,80
Suporte para células de carga (fabricado)	4	R\$ 100,00	R\$ 400,00
placa fenolite	2	R\$ 31,50	R\$ 63,00
cabos	1	R\$ 50,00	R\$ 50,00
plug, conectores e terminais kk	50	R\$ 1,00	R\$ 50,00
Extensor USB	1	R\$ 80,00	R\$ 80,00
Esteira Usada	1	R\$ 4.000,00	R\$ 4.000,00
Peças impressão 3D	2	R\$ 50,00	R\$ 100,00
caneta branca	1	R\$ 30,00	R\$ 30,00
Estanho solda	1	R\$ 15,90	R\$ 15,90
fresa de 0,1mm	3	R\$ 12,00	R\$ 36,00
		Total:	R\$18.721,20

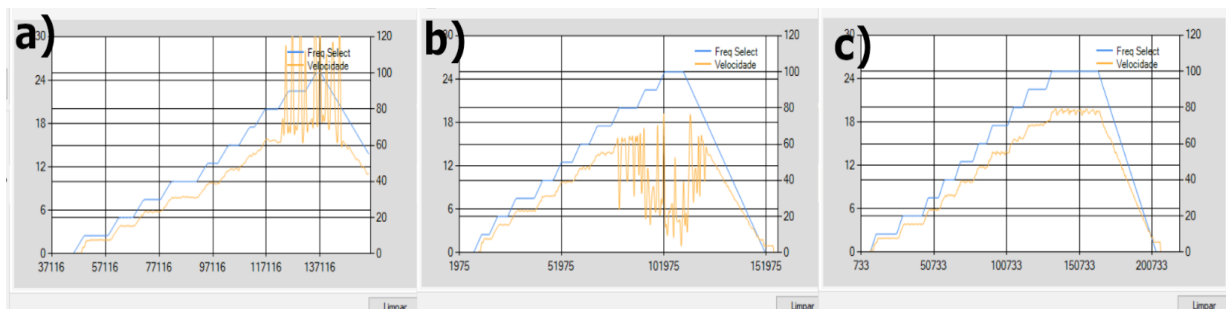
5 RESULTADOS E DISCUSSÃO

5.1 Leituras de velocidade da lona da esteira.

O cálculo da velocidade da lona realizado a partir dos números de leituras demonstrou valores mais precisos e taxa de amostragem maiores, fator importante para o desenvolvimento da parte de controle da esteira baseado na velocidade da lona.

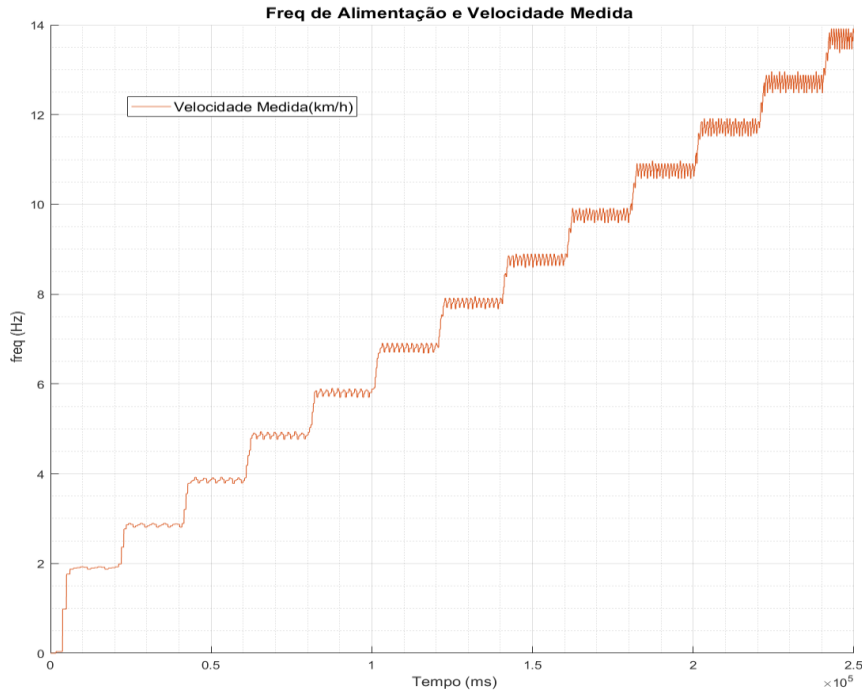
A utilização do *debounce* para garantir que as leituras de interrupção sejam precisas foram fundamentais. Nos testes utilizamos o *debounce* de 3450 microssegundos (μs) para obter os melhores resultados. Pelo medido no osciloscópio seria possível utilizar valores maiores, porém nos testes funcionais foi o que funcionou melhor. Na figura 22 (a), é possível notar o resultado de valores de *debounce* muito baixo, no caso 500 μs , resultando em leituras maiores do que o real. Na figura 22 (b) temos um resultado de *debounce* muito elevado, no caso 5000 microssegundos; leituras são perdidas e notou-se também resultados em erro de leitura. Na figura 22 (c), temos a configuração com melhor resultado sem erros de leitura, usando *debounce* de 3450 μs .

Figura 22: Resultados da utilização de tempo de *debounce* diferentes, 500 μs , 5000 μs e 3450 μs , respectivamente.



Gravamos um protocolo de aumento de carga de 10 em 10hz, a cada 20seg, chegando até aproximadamente 14km/h e tivemos o resultado plotado no gráfico da figura 23:

Figura 23: Dados de velocidade da esteira, quando controlado pela freq. de alimentação.



A oscilação do sinal notada, possivelmente ocorre devido a variações nas marcações feitas na lona da esteira, já que as mesmas foram feitas manualmente.

5.2 Controle da esteira

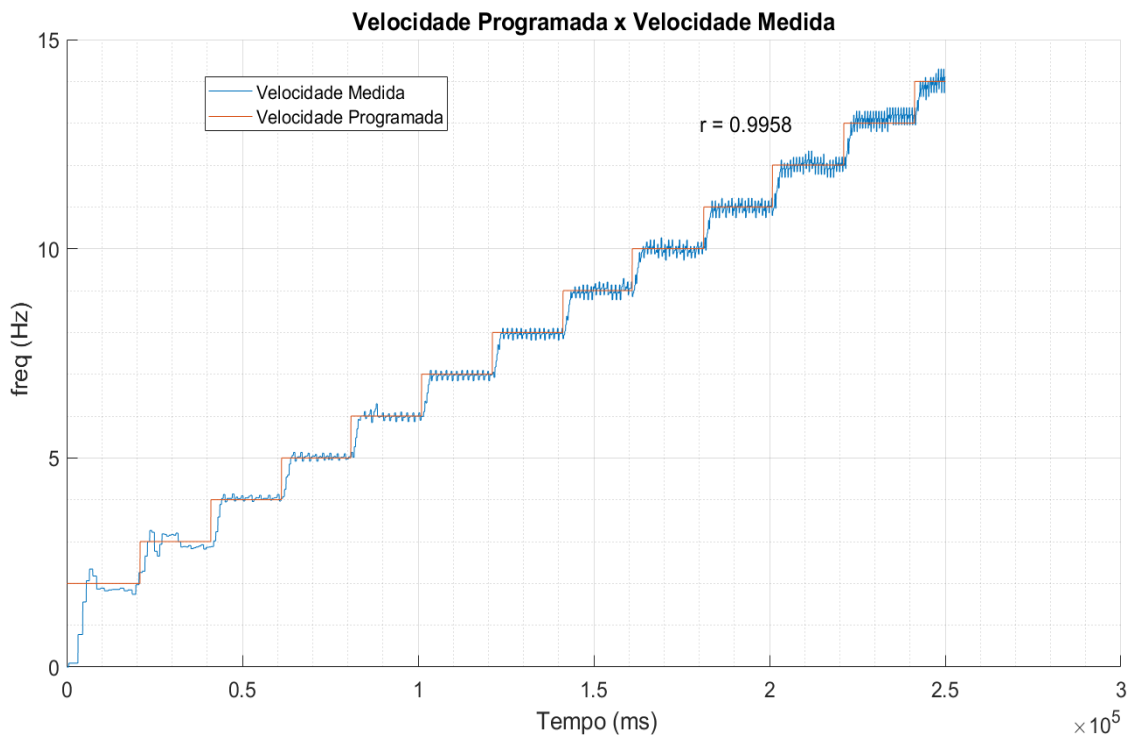
A placa de controle juntamente com o software desenvolvido foi muito eficaz em realizar a tarefa de controle e leituras de velocidade. Utilizar as interrupções para fazer a leitura do sensor óptico garantiu uma leitura precisa, evitando erros relacionados ao escorregamento que pode ocorrer entre a lona, o rotor da esteira e a polia do motor. Estava prevista a possibilidade de inverter o sentido de movimentação da lona. Apesar da modificação do sentido de giro do motor ser algo muito fácil de ser realizado, utilizado o inversor de frequência nos testes que fizemos ocorria um deslocamento lateral da lona quando o sentido de rotação era invertido, portanto essa função foi retirada do projeto atual.

Os fotoacopladores utilizados para separar o inversor da placa de controle garantiu segurança elétrica para a placa de controle e fez com que a mesma seja facilmente adaptada a um grande número de esteiras já presentes no mercado.

5.2.1 Controle baseado na velocidade de leitura.

O controle baseado na velocidade de leitura funcionou muito bem. Foi trabalho encontrar um bom tempo de espera nas leituras e a margem aceitável de velocidade. No caso usamos 50ms de tempo de espera e margem de velocidade de 0,2km/h. Isso deixou o tempo de aumento de carga um pouco lento, mas um controle bem preciso. Comparando o gráfico de velocidade programada com o de velocidade medida, temos uma correlação de 0,9958, como visto na figura 24.

Figura 24: Gráfico velocidade pelo tempo, durante protocolo de testes. Na linha vermelha a velocidade programada e em azul a velocidade medida na lona da esteira.



5.3 Leituras de forças de reação do solo FSR.

O sinal do componente vertical da FSR foi semelhante ao que se esperava, com a resolução de uma redução na diminuição do primeiro pico de contato, possivelmente em função dos amortecedores que estão presentes base da placa de suporte da lona embaixo da esteira. Na figura 5 (a) a resultante do somatório do sinal das quatro células de cargas durante uma caminhada a 5km/h, na figura 29 b) o sinal das quatro células separadas na mesma atividade. Na figura 25 o sinal tanto da somatória das 4 células de carga, com o sinal das 4 células individualmente, durante uma corrida a 10km/h.

Figura 25: Imagens FSR do solo (um passo) durante caminhada a 5km/h (fig. a) o somatório do sinal das 4 células de cargas, já na (fig. b) o sinal das 4 células separados).

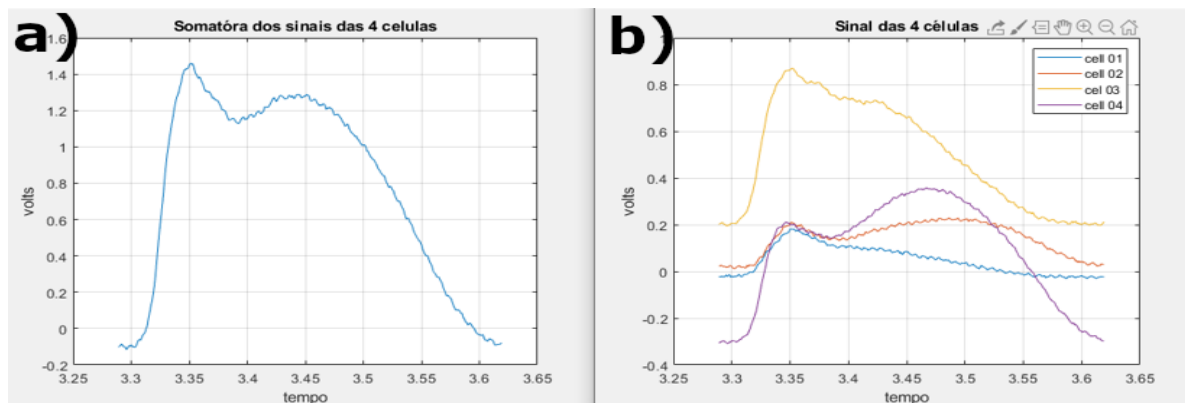
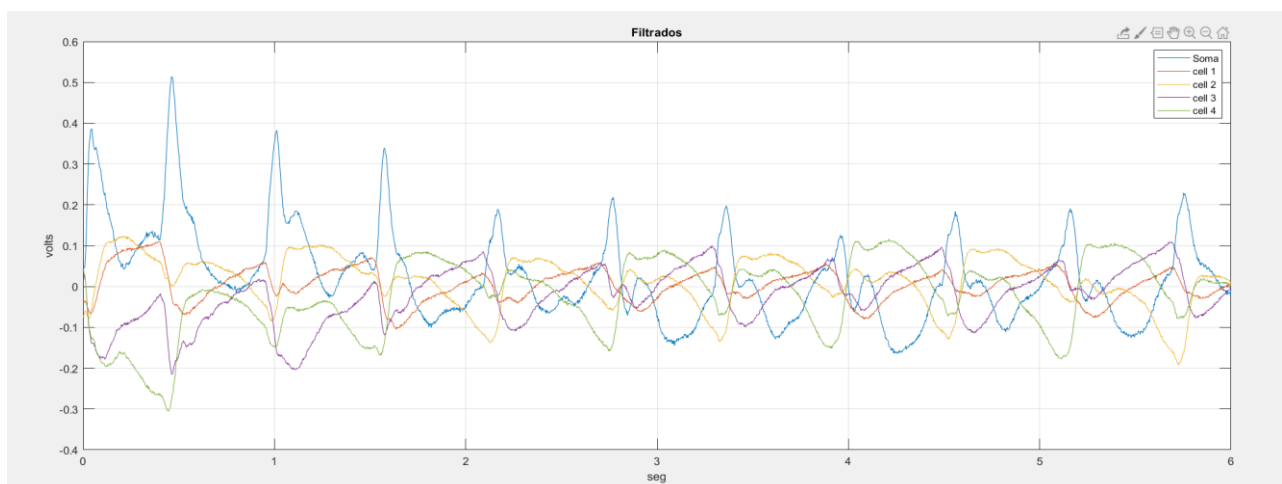
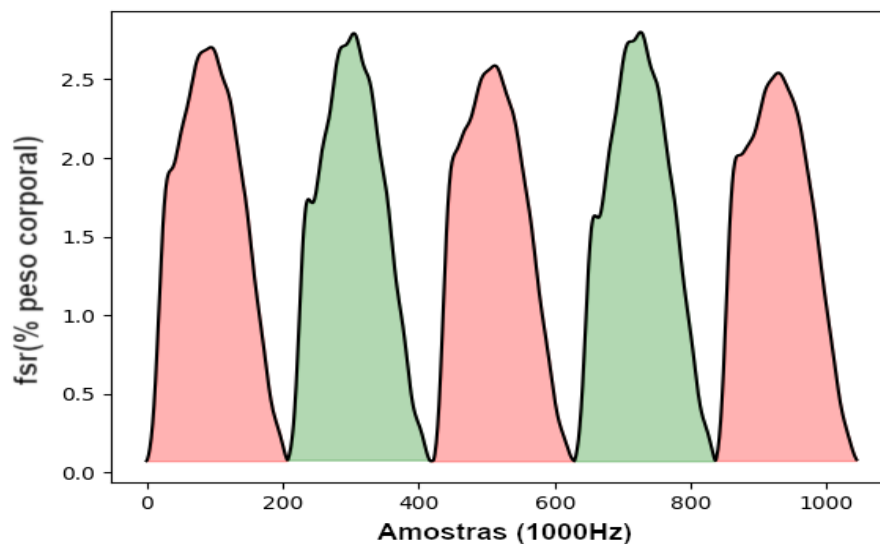


Figura 26: Sinal das 4 células de carga e da somatória delas durante corrida a 10km/h.



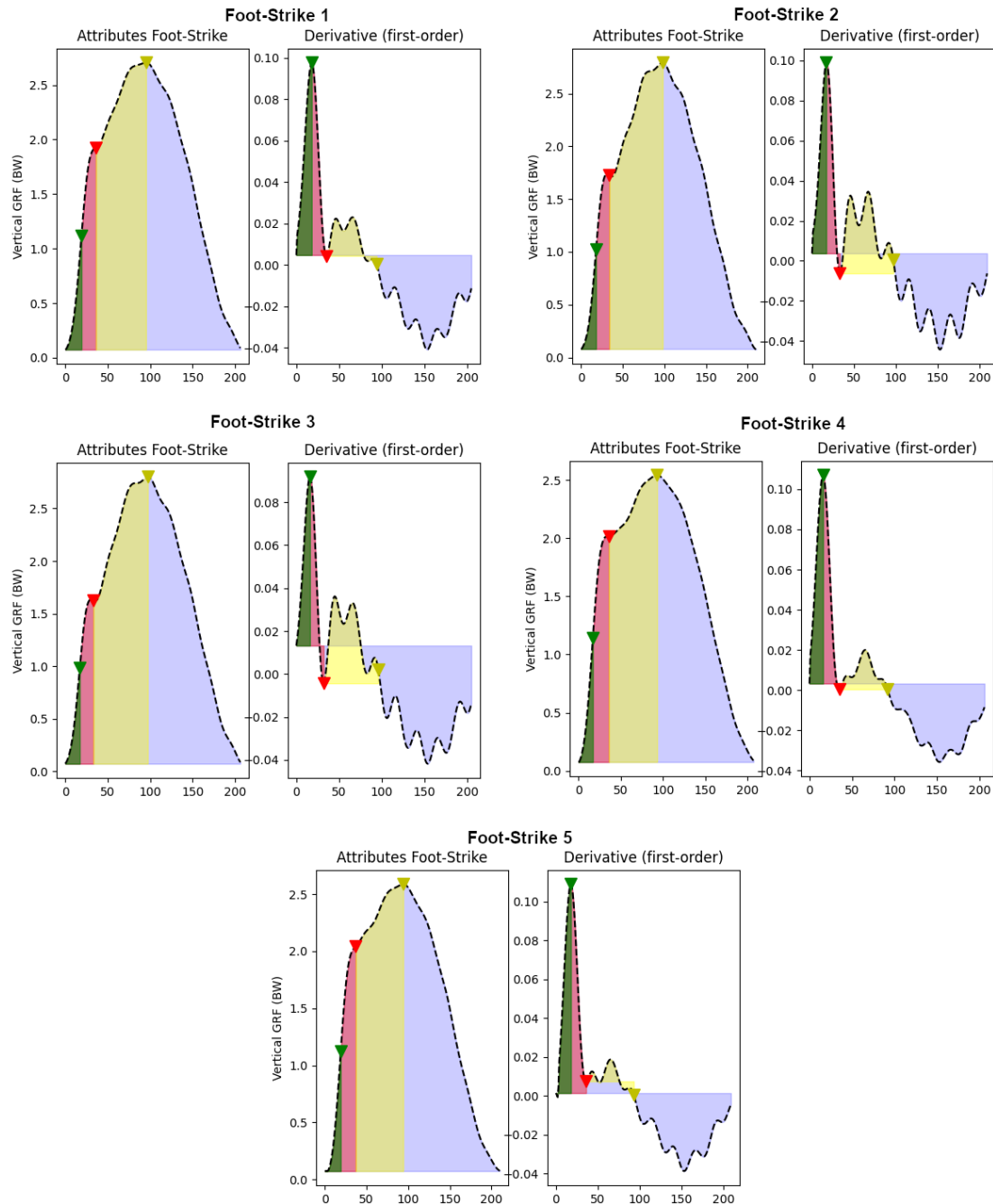
Na figura 27 estão representados o sinal da FSR de cinco passos. No gráfico as partes hachuradas em vermelho representam a FSR, enquanto o pé direito está em contato com a esteira, enquanto as partes hachuradas em verde, representa a FSR, enquanto o pé esquerdo está em contato com a esteira. O sinal está com as referências de calibração convertendo a saída de volts para kg, sendo o valor comparado com a porcentagem do peso corporal do sujeito.

Figura 27: FSR, em vermelho o sinal do enquanto o pé direito está em contato com a esteira e em verde o sinal enquanto o pé esquerdo está em contato com o solo.



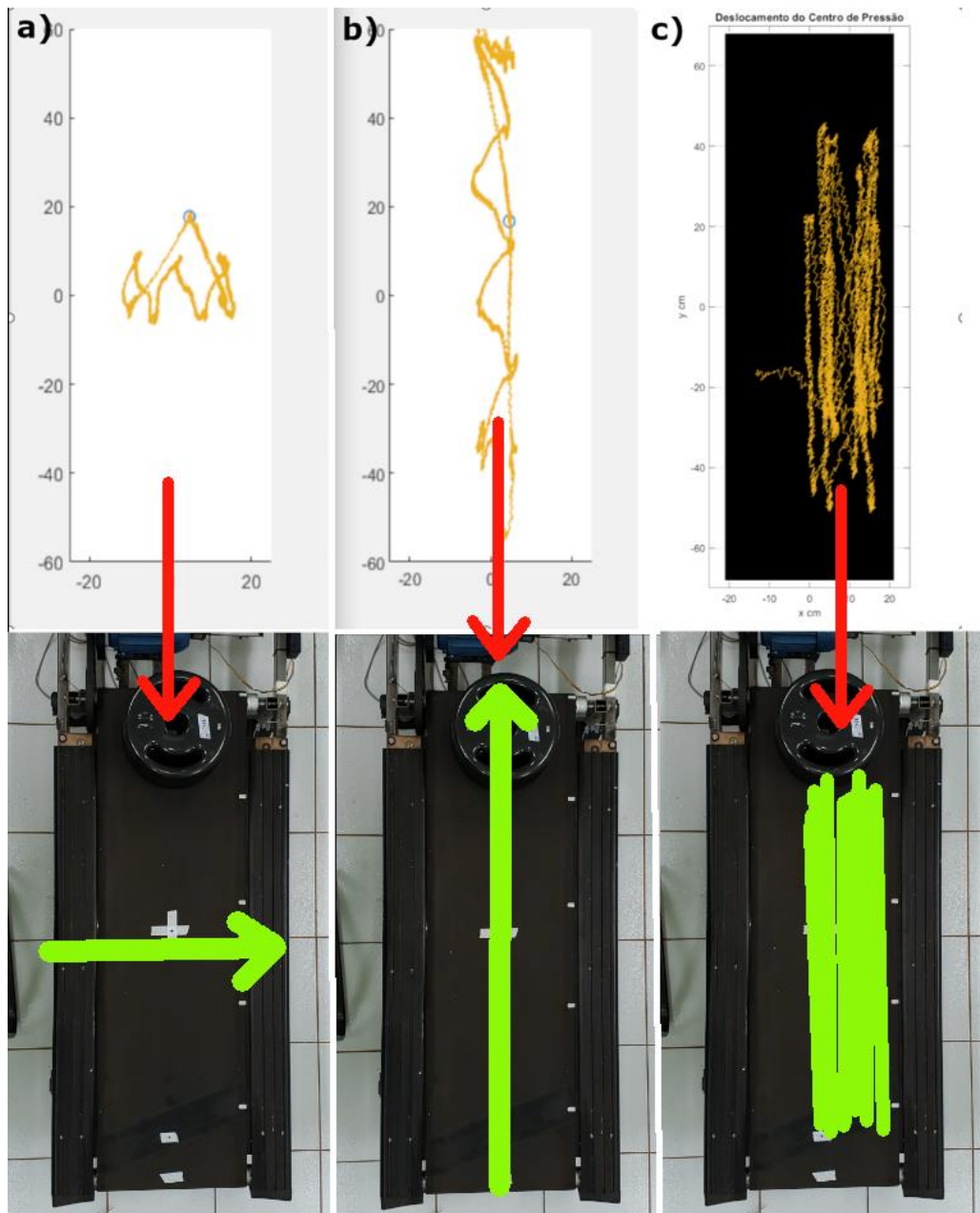
Na figura 28 tem os mesmo cinco contatos, mostrados separadamente, inclusive com cada fase de contato hachurados de cores diferentes, sinal esse semelhante ao visto por (LIEBERMAN; VENKADESAN; WERBEL; DAOUD *et al.*, 2010). A derivativa de primeira ordem do sinal de FSR, dá uma ideia de cada uma das fases da variação da FRS durante o contato de cada pé com a esteira.

Figura 28: FSR de cinco contatos dos pés do sujeito com o piso separados um a um. Vertical GRF (força de reação vertical do solo (FSR) em relação a peso corporal do sujeito)



Com os dados também foi possível realizar boas leituras do centro de pressão, segundo o algoritmo relatado anteriormente. Na figura 29(a) é possível verificar o sinal de uma caminhada com a esteira parada da esquerda para a direita, na figura 29 (b)) uma caminhada com a esteira parada, da parte de trás para frente e na figura 29 (c)) uma corrida a 10km/h.

Figura 29: Leituras COP da esteira (fig. a) caminhada com a esteira a lona parada lateralmente, fig. b) caminhada com a lona parada de trás para frente e fig. c) corrida tradicional com a lona a 10km/h).



6 POSSIBILIDADES FUTURAS

Com os resultados que obtivemos nas leituras e controle da esteira, podemos desenvolver várias análises, tanto em ambiente de pesquisa como em ambiente profissional. A possibilidade de adaptar esteiras já existentes, é muito interessante, pois diminui o custo para usuários que já possuem uma esteira comum.

Mesmo com a utilização do *debounce* para melhorar as leituras das marcas, para ler velocidades superiores a 18km/h precisamos melhorar a marcação feita na esteira, pois assim é possível utilizar menos o *debounce* e com isso conseguir ler velocidades bem mais elevadas.

Outra forma de melhorar a leitura de velocidade é modificar o tipo de sensor ótico utilizado. Ao invés de usar o de reflexo, alterar para o de perfuração e realizar pequenas perfurações na lona da esteira, espaçadas da mesma forma que as marcas feitas com a caneta branca. Dessa maneira teremos sinais mais claros e excluindo o risco de desgaste que as marcas feitas com tinta possam sofrer ao longo do tempo.

É possível melhorar a forma de realizar a aquisição, para isso pretendemos coletar e tratar os sinais das células de carga no mesmo software supervisor de controle da esteira. Isso permitirá a visualização do sinal em tempo muito próximo do real. Essa integração também possibilitará a utilização do sinal da FRS como feedback para controle da velocidade da lona da esteira.

O termo “*gamification*”, que já foi abreviada para gamificação, tem por definição adicionar elemento de jogos dentro de outras aplicações (WIKIPEDIA, 2021). Isso tem por objetivo aumentar o engajamento do usuário em determinada atividade. Dentro do esporte, temos aplicativos com Strava®, Garmin®Connect, entre outros que utilizam princípios da *gamification* para incentivar a prática e o compartilhamento de resultados com outros praticantes. Como boa parte das esteiras presentes no mercado possui funcionamento semelhante à utilizada nesse trabalho, implementar pelo menos a parte de controle do inversor de frequência e as leituras da lona, ou mesmo do sensor *reed switch* original no rotor da esteira, é fácil e barato. Dessa forma podemos desenvolver um aplicativo com melhorias e compartilhamento de resultados, o que faz da *gamification* um excelente incentivador da prática de esportes.

7 CONCLUSÃO

Foi possível coletar a velocidade diretamente na lona da esteira utilizando marcações feitas manualmente diretamente na lona da esteira. Também conseguimos controlar a velocidade da esteira a partir de um sistema supervisorio no computador, que permite o controle da velocidade da lona da esteira com base na frequência de alimentação do motor ou pelo feedback da própria velocidade da lona.

As leituras do componente vertical da FRS coletados nos ensaios realizados apresentou características semelhantes ao descrito na literatura. Com base no sinal de cada célula de carga conseguimos calcular o COP do indivíduo em movimento sobre a esteira e obtivemos dados semelhantes ao descrito na literatura.

O custo total do projeto, adaptando uma esteira comercial de baixo custo foi de pouco mais de R\$18.000,00, representando uma fração do custo de esteiras com características semelhantes. Todos os componentes usados são encontrados facilmente no mercado nacional; sendo assim, o projeto é de fácil replicação e manutenção.

Por fim, grandes melhorias ainda podem ser realizadas, tanto em software como em hardware, possivelmente a mais relevante delas seja a adequação para acesso dos dados e controle remoto via rede sem fio, já que o CI do ESP32 permite esse tipo de acesso.

8 REFERÊNCIA BIBLIOGRÁFICAS

AMTI®, P. **BIOMECHANICS OVERVIEW**. 2017. Disponível em: BIOMECHANICS OVERVIEW.

ANDERSON, T. Biomechanics and running economy. **Sports Med**, 22, n. 2, p. 76-89, 1996.

ARDUINO. **Arduino Reference attachInterrupt()**. 2020. Disponível em: <https://www.arduino.cc/reference/pt/language/functions/external-interrupts/attachinterrupt/>.

BOUILLOD, A.; PINOT, J.; VALADE, A.; CASSIRAME, J. *et al.* Influence of standing position on mechanical and energy costs in uphill cycling. 72, p. 99-105, 2018.

CEFISE®, B. E.

Esteira TK35 2017. Disponível em: <http://www.cefise.com.br/produto/49/40/esteira-tk35>.

CONRRADO., R. **Atividade Maker**. 2019. Disponível em: <http://atividademaker.com.br/cnc-revolution-2>.

DE ALMEIDA, R. M. A.; DE MORAES, C. H. V.; SERAPHIM, T. d. F. P. **Programação de Sistemas Embarcados: Desenvolvendo Software para Microcontroladores em Linguagem C**. Elsevier Brasil, 2017. 8535285199.

ESPRESSIF-SYSTEMS®. **ESP32 Series Datasheet**. Shanghai, 2020. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

FAROQI, A.; RAMDHANI, M.; FRASETYIO, F.; FADHIL, A., 2018, **DC Motor Speed Controller Design using Pulse Width Modulation**. IOP Publishing. 012205.

FIALHO, A. B. **Instrumentação industrial: conceitos, aplicações e análises**. Saraiva Educação SA, 2002. 853652541X.

FIALHO, L. F. L.; DA CUNHA, V. C.; HIRT, W. E. D.; INSTRUMENTAÇÃO, A. *et al.* Simulação e projeto de células de carga. **Citado na**, p. 48, 2014.

FRANCHI, C. M. **Inversores de frequência: teoria e aplicações**. Saraiva Educação SA, 2009. 8536511656.

FRANCHI, C. M. **Instrumentação de Processos Industriais - Princípios e Aplicações**. 2015.

H/P/COSMOS®. **Biomechanics - gaitway 3D**. 2018. Disponível em: <https://www.hpcosmos.com/en/products/software-measuring-technology/biomechanics-gaitway-3d>.

HAND, A. **Motores Elétricos, Manutenção e Solução de Problemas**. Tradução RIZZATO, F. A. P. 2 ed. Porto Alegre: 2015 2015. 978-85-8260-267-6.

HEISE, G. D.; MARTIN, P. E. Are variations in running economy in humans associated with ground reaction force characteristics? **Eur J Appl Physiol**, 84, n. 5, p. 438-442, 2001.

HUANG, C.-W.; SUE, P.-D.; ABBOD, M. F.; JIANG, B. C. *et al.* Measuring center of pressure signals to quantify human balance using multivariate multiscale entropy by designing a force platform. 13, n. 8, p. 10151-10166, 2013.

IMBRAMED®. <http://inbrasport.com.br/produto/esteira-ergometrica-super-atl/>. 2017. Disponível em: <http://inbrasport.com.br/produto/esteira-ergometrica-super-atl/>.

LAKOMY, H. The use of a non-motorized treadmill for analysing sprint performance. **Ergonomics**, 30, n. 4, p. 627-637, 1987.

LAKOMY, H.; CAMPBELL, I.; WILLIAMS, C. Treadmill performance and selected physiological characteristics of wheelchair athletes. **British journal of sports medicine**, 21, n. 3, p. 130-133, 1987.

LIEBERMAN, D. E.; VENKADESAN, M.; WERBEL, W. A.; DAOUD, A. I. *et al.* Foot strike patterns and collision forces in habitually barefoot versus shod runners. 463, n. 7280, p. 531-535, 2010.

LUCAS, H. **Pcb Gcode Eagle 9**. 2019. Disponível em: <https://pcbdesignsdl.blogspot.com/1969/12/pcb-gcode-eagle-9.html>. Acesso em: 02/05/2020.

LYSHOLM, J.; WIKLANDER, J. Injuries in runners. **The American journal of sports medicine**, 15, n. 2, p. 168-171, 1987.

MARTIN, J.; SPIRDUSO, W. Determinants of maximal cycling power: crank length, pedaling rate and pedal speed. **Eur J Appl Physiol**, 84, n. 5, p. 413-418, 2001.

MCCORMICK, F.; NWACHUKWU, B. U.; PROVENCHER, M. T. Stress fractures in runners. **Clinics in sports medicine**, 31, n. 2, p. 291-306, 2012.

NATIONA-INSTUMENTES®. **NI-DAQmx**. 2020. Disponível em: <https://www.ni.com/pt-br/support/downloads/drivers/download-ni-daqmx.html#348669>.

PETRUZELLA, F. **Motores Elétricos e Acionamentos: Série Tekne**. Bookman Editora, 2013. 8580552583.

STEVAN, S. L.; SILVA, R. A. **Automação e instrumentação industrial com Arduino: teoria e projetos**. Saraiva Educação SA, 2015. 853651566X.

THOMAZINI, D.; DE ALBUQUERQUE, P. U. B. **Sensores industriais: fundamentos e aplicações**. Saraiva Educação SA, 2020. 8536533242.

VERTULO, R. **Laboratório de Eletronica, Sensor de linha com arduino**. 2017. Disponível em: <http://labdeeletronica.com.br/sensor-de-linha-arduino/>.

VLACHOUDIS, V. **bCNC (github)**. 29/12/2020 2017. Disponível em: <https://github.com/vlachoudis/bCNC>.

WEG®. **Instalation, Configuration and Operation Guide CFW-08 Frequency Inverter, Document: 0899.5547 / 02**. Disponível em: <https://static.weg.net/medias/downloadcenter/hce/h47/WEG-cfw-08-multipump-system-technical-manual-0899.5547-manual-english.pdf>.

WEG®. **User's Manual Frequency Inverter CFW500**. 2019.

WIKIPEDIA. **Gamification**. 2021. Disponível em: <https://en.wikipedia.org/wiki/Gamification>.

WILLIAMS, P. T. Relationship of distance run per week to coronary heart disease risk factors in 8283 male runners The National Runners' Health Study. **Archives of Internal Medicine**, 157, n. 2, p. 191, 1997.

WOODWAY®, I. **Treadmill Force**. 2019. Disponível em: <https://www.woodway.com/products/force>.

ZELLER, S.; ABEL, T.; SMITH, P. M.; STRUEDER, H. K. Influence of noncircular chainring on male physiological parameters in hand cycling. **J Rehabil Res Dev**, 52, n. 2, p. 211 - 220, 2015.