



**JOSÉ INOCÊNCIO DIAS BALIERO
GILBERTO PEREIRA JUNIOR**

**DESENVOLVIMENTO DE HORÍMETRO DATALOGGER
PARA MONITORAMENTO DE MOTORES ELÉTRICOS**

**LAVRAS – MG
2022**

**JOSÉ INOCÊNCIO DIAS BALIERO
GILBERTO PEREIRA JUNIOR**

**DESENVOLVIMENTO DE HORÍMETRO DATALOGGER PARA
MONITORAMENTO DE MOTORES ELÉTRICOS**

Trabalho de Conclusão de Curso apresentado à
Universidade Federal de Lavras, como parte
das exigências do Curso de Engenharia
Agrícola, para a obtenção do título de Bacharel.

Dr. Giovanni Francisco Rabello

**LAVRAS – MG
2022**

**JOSÉ INOCÊNCIO DIAS BALIERO
GILBERTO PEREIRA JUNIOR**

**DESENVOLVIMENTO DE HORÍMETRO DATALOGGER PARA
MONITORAMENTO DE MOTORES ELÉTRICOS**

**DATALOGGER HOUR METER DEVELOPMENT FOR MONITORING ELECTRIC
MOTORS**

Trabalho de Conclusão de Curso apresentado à
Universidade Federal de Lavras, como parte
das exigências do Curso de Engenharia
Agrícola, para a obtenção do título de Bacharel.

APROVADO em 25 de abril de 2022

Dr. Fábio Domingues de Jesus UFLA

Dr. Carlos Antônio Rufino CEFET

Me. Carlos Eduardo Alves Oliveira UFV

Dr. Giovanni Francisco Rabelo

Orientador

**LAVRAS – MG
2022**

*Aos nossos pais Elizabeth e Inocência; Vanda e
Gilberto e a todos aqueles que, assim como eles,
independentemente de qualquer circunstância, sob
sol e chuva, trabalham com o propósito de prover o
sustento de suas famílias e saciar a fome da
sociedade.*

Homenageio e dedico

AGRADECIMENTOS

Primeiramente a Deus, pelo dom da vida e por nos permitir viver da sua graça.

Aos nossos pais Elizabeth e Inocêncio; Vanda e Gilberto, por não medirem esforços para nos auxiliar em nossos projetos, oferecendo apoio incondicional.

Aos professores docentes do curso de Engenharia Agrícola, pelas contribuições, auxílio, dedicação e compreensão.

Ao nosso orientador, Prof. Dr. Giovanni Francisco Rabelo, por acreditar na proposta do projeto e pelas suas contribuições motivacionais, espirituais e principalmente pela paciência.

Aos nossos eternos amigos de caminhada, Gilmar, Artur, Álvaro, Igor, Silas, Rafael, Otavio, Felipe, Fernando, Luiz, Gabriel, Lucas, Maria, Pâmela, Wigor, Rodolfo, Marco, Murilo, França, Matheus, Luana, João, Pedro, Carlos, Luíza, Thiago, Thulio, Bianca, Izabela, Lunna, Antônio, Allan, Andrey, Jean, Felipe, Vando e André.

À Universidade Federal de Lavras, pelo ambiente, infraestrutura e apoio em nossa graduação.

RESUMO

Nos diversos setores produtivos industriais, alguns fatores como o controle de gastos com manutenção de equipamentos são fundamentais para que as empresas se mantenham no mercado. Independentemente do tamanho da empresa, a manutenção e o acompanhamento são de extrema importância, visto que possibilitam evitar desperdícios, bem como aumentar a lucratividade. Em qualquer área, a coleta de informações é fundamental para desenvolvimento de indicadores. Por meio da realização de estágio dos autores, foi identificada uma falha na captação de informações em uma indústria e verificado que isso se repete na maioria dos lugares que utilizam motores elétricos. A falha citada é referente a falta de informações do tempo de uso de um motor elétrico, não se quantificava o trabalho de um motor e desta maneira dificultava algumas atividades, como o planejamento de manutenção. A proposta de trabalho é apresentar uma solução para quantificação de horas trabalhadas para motores elétricos. A parte eletrônica é composta por uma placa da plataforma Arduino, junto com sensores de corrente elétrica. O projeto apresentou uma alternativa de baixo custo, fácil, ágil e confiável para o monitoramento de motores elétricos. O horímetro desenvolvido é capaz de fazer a leitura de horas trabalhadas e armazenar em uma planilha, desta forma se tornou mais fácil e preciso a coleta dessas informações em qualquer indústria.

Palavras-chave: Arduino. *Datalogger*. Monitoramento. Manutenção.

ABSTRACT

In the various industrial productive sectors, factors such as expenses with equipment's control maintenance are fundamental for companies to remain in the market. Maintaining and monitoring, regardless of the company's size, is crucial, as it avoids waste and increases profitability. Whatever the area, the collection of information is essential for the indicators development. Through the authors' internship, a failure was identified in information capture in the industry and verified that this is repeated in most places that use electric motors. The mentioned failure is related to the lack of information on the time of use of an electric motor, the work of an engine was not quantified and, in this way, it made some activities difficult, such as maintenance planning. The work proposal is to present a solution for the quantification of hours worked for electric motors. The electronics are composed of an arduino platform board, along with electric current sensors. The project presented a low-cost, easy, agile and reliable alternative for monitoring electric motors. The developed hour meter is able to read the hours worked and store it in a spreadsheet, thus making it easier and more accurate to collect this information in any industry.

Keywords: Arduino. Datalogger. Monitoring. Maintenance.

LISTA DE FIGURAS

Figura 1:Placas arduino UNO e Mega 2560.....	10
Figura 2: Diagrama de blocos da sequência de funcionamento de um sketch.	11
Figura 3: Módulo RTC.	13
Figura 4: Módulo MMC.	13
Figura 5: Relé.	14
Figura 6: Exemplos dos dados a serem gravados pelo sistema.	15
Figura 7: Diagrama de blocos: logica de funcionamento do sistema.	16
Figura 8: Percepção com status com uso de contato auxiliar.	17
Figura 9: Percepção do status da carga com o uso do relé.	17
Figura 10: Ensaio utilizando protoboard.	19
Figura 11: Desconexão dos fios na protoboard.	20
Figura 12: Planilha criada pelo sistema após ensaio.	21
Figura 13: Gráfico ilustrativo de funcionamento das cargas testadas neste estudo (bomba, elevador, exaustor e mesa), ao longo do período experimental (09h16min às 21h02min).	22
Figura 14: Horímetro.	23

SUMÁRIO

1.	INTRODUÇÃO	1
1.1.	Problematização	2
1.1.1.	Formulação do problema	2
1.1.2.	Solução proposta	2
1.2.	Objetivos	2
1.2.1.	Objetivo Geral	2
1.2.2.	Objetivos Específicos	3
1.3.	Metodologia	3
2.	REFERENCIAL TEÓRICO	3
2.1.	Manutenção	3
2.2.	Motores elétricos	6
2.2.1.	Vida do motor elétrico	7
2.3.	Sistemas embarcados	7
2.4.	Microcontroladores	8
2.5.	Arduino	8
2.5.1.	<i>Hardware</i> Arduino	9
2.5.2.	Bibliotecas	10
2.5.3.	<i>Shields</i>	10
2.5.4.	Estrutura de um <i>sketch</i>	11
2.5.5.	Pinagem	11
2.6.	Horímetro	12
2.7.	Módulo RTC	12
2.8.	Módulo MMC	13
2.9.	Relé	13
3.	DESENVOLVIMENTO	14

3.1.	Metodologia	14
3.2.	Requisitos funcionais	14
3.3.	Funcionamento da plataforma Arduino	15
3.3.1.	Visão geral.....	15
3.3.2.	Percepção do <i>status</i> da carga.....	16
3.4.	Lógica de funcionamento do código	17
3.5.	Simulação	18
4.	RESULTADOS E DISCUSSÕES	20
4.1.	Testes e simulações	20
4.2.	Orçamento	22
4.2.1.	Horímetro convencional	23
4.2.2.	Horímetro Datalogger.....	23
5.	CONCLUSÕES	24
	REFERÊNCIAS	25
	APÊNDICE 1 - SKETCH DO PROJETO	27

1. INTRODUÇÃO

Arquimedes, filósofo grego, já no século III a.C. estudava a respeito de máquinas, com o objetivo de compreender como elas agregam valor útil à humanidade, a exemplo das alavancas, que eram máquinas com elevada utilidade cotidiana (STRIZHAK; PENKOV; PAPPEL, 2007). No entanto, foi a partir da Revolução Industrial, mais especificamente na década de 1830, que a utilização de máquinas se mostrou como um ponto de inflexão para a humanidade. A partir deste evento, relações de vários âmbitos sofreriam algum tipo de influência, mudando de forma abrupta e permanente a vida das pessoas, principalmente àquelas dos grandes centros urbanos. Estas alterações na sociedade chegaram com a promessa de elevar exponencialmente a produtividade das atividades antes feitas de forma manual, algo que foi alcançado, bem como assegurar produções uniformes e com menor custo. No entanto, muitos aspectos não se concretizaram como previsto, como a melhoria da qualidade de vida dos trabalhadores, pois, uma vez que as máquinas faziam uma parte das atividades que antes eram manuais, logo os trabalhadores teriam menos ocupações e, assim, poderiam ter mais ociosidade, podendo se dedicar a outras atividades, como tempo de qualidade com a família, fato que não se confirmou, porém não foi o suficiente para que a utilização de máquinas se mostrasse inviável (RODRIGUES, 2009).

Desde a Revolução Industrial, a história da manutenção de máquinas tem acompanhado o avanço das mesmas. As máquinas evoluíram e se tornaram complexas em demasia e, conseqüentemente, suas demandas por manutenção também. As manutenções são primordiais para a qualidade, produtividade e viabilidade dos resultados, justificando a necessidade da adoção de critérios cada vez mais rígidos para avaliação de suas metodologias. Do ponto de vista da gestão administrativa, busca-se que a prática de manutenção possibilite maior eficiência no uso de recursos, favorecendo os resultados da instituição em seu mercado de atuação (FERREIRA, 2019).

Nas últimas décadas, são notórios os desenvolvimentos nos campos das máquinas, gestão de recursos e técnicas de manutenção. Grandes indústrias têm grande eficiência em suas atividades de produção, justificadas, dentre outros motivos, pelos investimentos em corpo técnico e em equipamentos, que possibilitam realizar manutenções corretas nos equipamentos. Um dos critérios para a manutenção de equipamentos é por meio de horas de operação, em que a cada intervalo de tempo é feita a revisão da máquina utilizada (ROMANO, 2003).

Todavia, ainda hoje, indústrias menores e pequenos produtores carecem de recursos técnicos para estimar o momento adequado para intervir nos equipamentos, a fim de realizar a

sua manutenção, o que pode acarretar prejuízos por revisões em menores prazos, culminando em maiores frequências, ou por maiores prazos, ocasionando em desgastes desnecessários ao maquinário (CHAPMAN, 1989). Para estes agentes, uma ferramenta que compute o tempo de atividade de máquinas e que, ao mesmo tempo, seja prática e economicamente viável, pode ser bastante útil. Considerando esse cenário, este trabalho se propôs em desenvolver uma ferramenta para a computação de horas de atividade de equipamentos elétricos, um horímetro, de modo que fosse de produção de baixo custo, fácil utilização e eficiente.

1.1. Problematização

1.1.1. Formulação do problema

O problema detectado é a falta da quantificação de uso dos motores elétricos em indústrias. A falta destes dados inviabiliza uma correta manutenção, otimização do uso da energia elétrica e gestão de processos. O que pode ocasionar maiores custos na indústria, tanto com manutenção, muitas vezes feita de maneira tardia e com maiores custos, quando com os próprios custos de produção.

1.1.2. Solução proposta

Diante do problema descrito, este trabalho propõem a concepção de horímetro de baixo custo utilizando a plataforma Arduino, visando reduzir os custos mencionadas, contribuir para a maior eficiência do uso de motores elétricos, bem como possibilitar subsídios que promovam uma maior disponibilidade das máquinas para trabalho.

1.2. Objetivos

1.2.1. Objetivo Geral

O objetivo geral deste trabalho foi conceber um horímetro, baseado na plataforma Arduino, que seja capaz de contabilizar as horas de funcionamento de uma máquina por meio da quantificação do tempo de funcionamento dos motores elétricos, elemento motriz deste tipo de máquina.

O projeto teve como objetivo apresentar-se como uma alternativa de baixo custo e que, além de tornar fácil, ágil e confiável o monitoramento das cargas, torna menos trabalhosa e mais simples a coleta e tratamento dos dados, uma vez que o horímetro convencional monitora uma carga por vez, não é capaz de registrar os dados em uma planilha, não registra o status da

carga ao longo do dia, é necessário a disponibilidade de um responsável para fazer a leitura desses dados periodicamente, e, a depender do tamanho da indústria (quantidade de cargas), esta atividade pode ser dispendiosa de tempo e estar sujeita a erros de transcrição.

1.2.2. Objetivos Específicos

Foram considerados objetivos específicos para este trabalho:

- Montar plataforma Arduino capaz de monitorar o status de várias cargas ao mesmo tempo;
- Programar um código capaz de fazer o monitoramento das cargas e registrar o status em uma planilha ao longo do tempo;
- Obter a informação de status das máquinas de maneira mais simples possível, com o mínimo de invasão;
- Ajustar a integração dos componentes; e
- Realizar simulações que permitam observar os resultados do trabalho.

1.3. Metodologia

A metodologia utilizada para a realização do projeto foi dividida em duas etapas para melhor compreensão, que estão listadas abaixo:

- **Fundamentação teórica:** constitui em pesquisar e estudar os conceitos necessários.
 - Fornecer uma introdução a manutenção;
 - Fornecer uma introdução a motores;
 - Fornecer uma introdução a sistemas embarcados e microcontroladores;
 - Fornecer uma introdução a Arduino.
- **Desenvolvimento:** essa etapa apresenta a montagem da plataforma Arduino, programação e maneiras de identificar o funcionamento do motor.

2. REFERENCIAL TEÓRICO

2.1. Manutenção

Em qualquer indústria, as linhas de produção dispõem de diversos equipamentos, que funcionam de forma associada ou individual, e precisam ser reparados de tempos em tempos. As operações de reparo são necessárias devido problemas como desgaste, enguiços, quebras, fraturas, e mais uma vastidão de acidentes e incidentes que podem ocorrer durante a produção

(NEPOMUCENO, 1989). Em se tratando de qualidade e produtividade, a manutenção é de vital importância, pois pode evitar que os equipamentos sofram possíveis paradas não programadas, ou mesmo que funcionem fora do padrão (NOGUEIRA; GUIMARÃES e da SILVA, 2012).

Segundo PINTO e XAVIER (2009, p. 1 a 6), a história da manutenção pode ser dividida em quatro gerações. A primeira se passa no período antes da Segunda Guerra mundial, no qual a indústria era pouco mecanizada e, devido a conjuntura econômica da época, a produtividade não era prioridade, e não era necessária uma manutenção sistematizada. Nesta época, eram feitos apenas serviços de limpeza e manutenção pós-quebra, pois não se tinha uma visão de que os equipamentos se desgastavam com a idade e quebravam, então se almejava apenas mão de obra técnica, com habilidades voltadas ao reparo.

Segundo os mesmos autores, a segunda geração ocorre entre os anos 1950 e 1970, se passando após a Segunda Grande Guerra, no qual houve um aumento na demanda por todo tipo de produto, ao mesmo tempo que era observada uma redução sensível da mão-de-obra industrial, havendo então forte crescimento da mecanização e complexidade nas instalações industriais. Diante disso, tornava-se necessário que as máquinas tivessem maior disponibilidade e confiabilidade, e as indústrias se tornavam bastante dependentes do bom funcionamento das máquinas e, portanto, foi levantada a ideia de que as falhas poderiam e deveriam ser evitadas. As manutenções preventivas eram feitas em intervalos fixos de intervenção, e passam a ser onerosas, fazendo com que se aumente as buscas por sistemas de planejamento e controle.

A terceira geração tem início após a década de 1970, a partir da intensificação dos processos de mudança nas indústrias, com destaque para a ocorrência de paralizações, que reduzem a capacidade produtiva e a qualidade dos produtos, além de aumentar os custos e gerar preocupações. A partir desta época, tendeu-se ao uso de sistemas *Just-in-time*, e estoques reduzidos, combinados com a produção em andamento, significavam que pequenas pausas na produção poderiam paralisar a fábrica. Os sistemas começam a ficar cada vez mais mecanizados e automatizados, e, a partir de então, disponibilidade e confiabilidade passam a ser postos-chave nos mais diversos setores econômicos da sociedade. Porém, apesar de novos projetos buscarem maior confiabilidade, ainda existia falta de interação entre as áreas de engenharia, manutenção e operação, o que impedia que os resultados fossem melhores (PINTO e XAVIER, 2009).

Os mesmos autores apontam que a quarta geração teve início em meados de 1995, na qual várias expectativas da terceira geração continuam a existir, mas são consolidadas atividades de engenharia de manutenção na estrutura organizacional da indústria, e as práticas de manutenção preditiva e de monitoramento de condição de equipamentos são cada vez mais utilizadas, com o objetivo de intervir o mínimo na produção.

Nos diversos setores produtivos industriais, alguns fatores como o controle de gastos com manutenção de equipamentos são fundamentais para que as empresas se mantenham no mercado. Manutenção e acompanhamento, independentemente do tamanho da empresa, são de extrema importância, visto que podem evitar desperdícios e ainda aumentar a lucratividade (FERREIRA, 2019).

Segundo Capelli (2013, citado por Ferreira, 2019, p. 13), “ toda empresa necessita de um sistema de monitoramento contínuo, para que se obtenha o maior controle do processo aumentando sua competitividade, melhorando os pilares qualidade, redução de custo operacional e desempenho de produção [...]”.

Define-se “manutenção” como “reunião daquilo que se utiliza para manter e/ou conservar alguma coisa, garantindo, assim, seu bom funcionamento: manutenção de máquinas; manutenção de ferramentas” (DICIO, 2022). A ABNT (1994), por meio da norma NBR 5462, define manutenção como o conjunto de todas as ações técnicas e administrativas, incluindo as de supervisão, com o propósito de manter ou recolocar um item no estado em que consiga desempenhar sua função definida. Em qualquer fábrica ou instalação industrial, a manutenção é essencial, já que em algum momento uma máquina irá quebrar e, portanto, será necessário algum técnico para reparar (NEPOMUCENO, 1989).

No século XVIII, a Revolução Industrial na Inglaterra foi responsável por grandes mudanças, que se refletem atualmente. Este período foi marcado por grande desenvolvimento tecnológico, resultando no surgimento da indústria. A partir disso, o modo de viver humano mudou, e iniciou-se a busca pelas inovações tecnológicas e de organização da sociedade, como o movimento trabalhista, no qual se busca melhores condições de trabalho. Por meio da adoção de métodos de serviços mais adequados, que se refletem em melhores ambientes de trabalho e processos indiretos de controle, a manutenção contribui para a melhoria nas condições de trabalho (NEPOMUCENO, 1989). Desta forma, a manutenção não só contribui para a eficiência produtiva, mas também dá grande passos para a humanização do trabalho.

A ABNT (1994), por meio da norma NBR 5462, divide a manutenção em três tipos:

- Manutenção preventiva: efetuada antes da falha, em intervalos predeterminados ou de acordo com critérios. Este tipo de manutenção é destinado a reduzir a probabilidade de ocorrer alguma falha ou deterioração do funcionamento de algum item.
- Manutenção corretiva: efetuada após a falha, é realizada de forma imediata, para a resolução do problema. Tem por objetivo recolocar o item em condições normais de trabalho.

- Manutenção preditiva: é a que permite assegurar uma qualidade de serviço desejada, com base na aplicação sistemática de técnicas de análise e dados, utilizando-se de meios de supervisão, para reduzir ao mínimo as manutenções preventiva e corretiva.

A manutenção é de grande importância em qualquer atividade fabril, pois, além de reparar os estragos e aumentar a vida útil dos equipamentos, possibilita que as máquinas continuem trabalhando. Segundo Nepomuceno (1989, p. 2), um estudo detalhado mostra que quando existe uma organização preparada, os custos de manutenção são suprimidos, quanto comparados ao lucro obtido, por conseguir manter a capacidade produtiva. Além disso, a realização de manutenções possibilita prevenir falhas inesperadas, que causam impactos financeiros, acarretando prejuízos na produção.

2.2. Motores elétricos

Segundo Franchi (2007, p. 17), motor elétrico é uma máquina que transforma energia elétrica em energia mecânica. Com a simples passagem de corrente elétrica, o resultado é o movimento em um eixo, que pode ser utilizado de diversas maneiras. Motores elétricos são um assunto de grande importância econômica, e estima-se que o seu mercado mundial seja da ordem de dezena de bilhões de dólares. Nas indústrias, estima-se que cerca de 80% da energia elétrica consumida seja transformada em energia mecânica, por meio de motores elétricos.

Os motores elétricos são capazes de transformar energia elétrica em energia mecânica com algumas perdas de energia (GUEDES, 2004). Como essas perdas de energia são pequenas, o motor elétrico pode ser considerado uma máquina de rendimento elevado. O progresso tecnológico permitiu o desenvolvimento de controle dos motores elétricos, que evita dissipação de energia e contribui para a utilização racional da energia elétrica.

Segundo Gonzalez (2007, p. 18), entre todos os tipos de motores, o motor elétrico é o mais utilizado, pois combina uma série de vantagens, quando comparados com outros tipos. Dentre as vantagens, pode-se citar baixo custo, simplicidade de comando, adaptação e bom rendimento. Além disso, vale destacar que utilizam energia elétrica, que pode ser obtida de maneira sustentável.

Para Mamede Filho (2017, p. 235), os motores elétricos, de acordo com o tipo de fonte de alimentação, podem ser divididos em motores de corrente contínua e de corrente alternada:

- Motores de corrente contínua: são acionados a partir de uma fonte de corrente contínua. Possuem capacidade de um controle fino de velocidade e, por este motivo, são aplicados em situações que exigem tais características.

- Motores de corrente alternada: são acionados a partir de uma fonte de corrente alternada. São os mais utilizados na maioria das aplicações industriais. São classificados em duas categorias: síncronas e de indução. Nas máquinas síncronas, as correntes são fornecidas através de contatos, e já nas máquinas de indução, as correntes são induzidas (FITZGERALD, 2006).

Segundo Gonzalez (2007, p. 18), o motor de indução é o mais utilizado devido a sua simplicidade, robustez e baixo custo. É considerado adequado para quase todos os tipos de máquinas. Normalmente, opera com uma velocidade constante, que pode variar ligeiramente com a carga mecânica aplicada no eixo.

Para aplicação nas indústrias brasileiras, o motor que melhor tem se adaptado é o de indução trifásico com gaiola de esquilo. São motores de construção robusta, que exigem menor manutenção, com rendimento na casa de 90% e com baixo custo de aquisição, quando comparado aos demais. Possuem poucas desvantagens, como não variar a velocidade e exigir alta corrente de partida (GARCIA, 2003).

2.2.1. Vida do motor elétrico

Gonzalez (2007, p. 1), aponta que a vida útil do motor de indução depende quase que exclusivamente da isolação dos enrolamentos, já que outros fatores são de construção simples e robusta. A isolação dos enrolamentos é afetada por diversos fatores, tais como umidade, vibrações, ambientes corrosivos, dentre outros. Dentre todos os fatores, um dos mais importantes é a temperatura, pois um aumento de 8°C a 10°C na temperatura permitida pela isolação é capaz de reduzir a vida útil do motor pela metade.

É importante ressaltar que a vida útil da isolação refere-se também ao envelhecimento gradual do isolante, que com o passar do tempo se torna ressecado e perde o poder isolante, produzindo um curto-circuito.

2.3. Sistemas embarcados

Sistema embarcado pode ser conceituado como capacidade computacional dentro de um circuito integrado, equipamento ou sistema. Um sistema embarcado não é apenas um simples computador, mas um sistema completo e independente, preparado para realizar uma determinada tarefa. O usuário final de um sistema embarcado, não tem acesso ao programa instalado no dispositivo, mas interage por meio das interfaces projetadas para o sistema, como teclados, displays etc. (CUNHA, 2007)

Diferente de computadores, em que são utilizados diversos aplicativos, um sistema embarcado realiza tarefas predefinidas, normalmente com requisitos específicos. Devido a característica de ser um sistema simples, com uso de engenharia pode-se otimizar projetos, possibilitando redução de custos e recursos computacionais. De maneira geral, o sistema não pode ter sua funcionalidade alterada durante o uso. Caso queira-se modificar o propósito de uso, é necessário reprogramar todo o sistema (VILLAÇA, 2012).

2.4. Microcontroladores

No sistema embarcado, há necessidade de algum mecanismo para gerenciar todo o funcionamento. Essa função é dada aos microcontroladores que, de maneira geral, executam o sistema embarcado. Os microcontroladores são capazes de fazer a leitura de sinais externos, executar o programa projetado, processar os sinais e executar a função, para entregar o resultado esperado (CUNHA, 2007).

Oliveira (2010, p. 35), descreve que os microcontroladores são ideais para aplicações que necessitam menores dimensões, tempo e custos. Estão inseridos neles, em um único chip, todos os periféricos necessários, mesmo que pequenos. Os microcontroladores estão presentes em qualquer situação que seja necessário o controle de um dispositivo eletrônico. São amplamente utilizados na automação industrial e de eletrodomésticos. Como exemplo, pode citar o elevador: quando alguém aperta o botão para ir até um andar específico, o microcontrolador recebe a informação, interpreta e aciona os motores para chegar até ao andar, parar e abrir as portas.

2.5. Arduino

O Arduino nasceu como uma ferramenta de prototipagem fácil e rápida, destinada a estudantes sem formação em eletrônica e programação (ARDUINO, 2022). O mesmo autor ainda aponta algumas vantagens do microcontrolador Arduino: barato, multiplataforma, com ambiente de programação simples e claro, *software* e *hardware* de código aberto e extensível.

Arduino (2022, em seu web site), descreve a placa Arduino como uma plataforma eletrônica, de código aberto baseada em *hardware* e *software* fáceis de usar. O Arduino é capaz de ler entradas diversas (como interruptores, sensores, mensagens em sites da web etc.), transformando-as em saídas (acionar um motor, publicar algo na web, ligar um LED etc.), basta “dizer à placa o que fazer”, enviando um conjunto de instruções.

Monk (2013, p. 1) descreve o Arduino como uma pequena placa de microcontrolador contendo um plugue USB, que permite a conexão com um computador e/ou diversos outros terminais, que possibilitam a conexão com dispositivos externos, como motores, relés, sensores luminosos, diodos a laser, alto-falantes etc.. As placas Arduino podem ser energizadas por meio do computador ou por uma bateria de 9 V, e podem ser controladas por um computador, ou ser programada no computador e posteriormente desconectada, permitindo que trabalhe de maneira independente.

Segundo Arduino (2022), site oficial da plataforma Arduino, para enviar as instruções à placa, é necessário usar o Arduino IDE (ambiente de desenvolvimento integrado), que contém um editor de texto para a escrita de códigos, e uma barra de ferramentas, como de praxe em *softwares* computacionais de desenvolvimento. O IDE se conecta com o *hardware* Arduino, para envio dos programas e interação (um programa escrito no IDE é chamado de *sketch*).

Há diversas alternativas ao Arduino no mercado, porém, a plataforma Arduino apresenta algumas facilidades tais como: facilidade de encontrar no mercado, grande disponibilidade de projetos e documentações na internet, facilidade de encontrar periféricos e *shields* no mercado nacional, facilidade de uso e preço acessível (RODRIGO, 2011; SANTOS, 2011; WERNECK, 2009).

Como vantagens de se utilizar o Arduino em relação a outros microcontroladores, Santos (2021, p. 10) aponta: possibilidade de ser alimentado pela porta USB; O IDE pode ser instalado em ambiente Windows, Linux ou MacOS; o IDE é gratuito; total compatibilidade entre o IDE e a placa, não sendo necessário configurações complexas; na maioria dos outros microcontroladores, a compilação e a gravação do programa no microcontrolador são realizadas em *softwares* diferentes; não precisa de alimentação externa; preço mais acessível; é uma plataforma já pronta para trabalhar; trabalha com a linguagem C++; existem inúmeras bibliotecas e exemplos prontos, reduzindo o tempo de desenvolvimento de sua prototipação utilizando *shields*.

2.5.1. *Hardware* Arduino

Há no mercado uma grande diversidade de plataformas Arduino, sendo o Arduino Uno a mais comum. A placa Arduino Uno conta com 14 pinos digitais, que podem ser configurados como entrada ou saída, sendo que seis destes podem ser configurados para fornecer uma saída de modulação de largura de pulso (PWM). O Uno conta ainda com seis entradas analógicas e, para projetos que exigem mais portas de entrada e saída, há o Arduino Mega, que conta com 54

pinos digitais, sendo 14 com opção PWM e 16 pinos de entrada analógica (EVANS; NOBLE; HOCHENBAUM; 2013). As placas Arduino Uno e Mega estão ilustrados na Figura 1.

Figura 1: Placas arduino UNO e Mega 2560.



Legenda: A: Placa arduino Mega 2560. B: Placa arduino UNO.

2.5.2. Bibliotecas

Para tornar mais enxutos e menos repetitivos os códigos escritos nos *sketches*, como é feito nas linguagens de programação convencionais, no Arduino IDE também é possível fazer o uso de bibliotecas. As bibliotecas são uma coleção de subprogramas ou rotinas que o usuário pode importar no seu programa, e que realizam operações comuns e complexas, até mesmo interações com *shields*, ajudando a simplificar o código e diminuir o uso de memória pelo programa (GAIER, 2011; GER, 2009; HACHOUCHE).

Santos (2021, p. 175) descreve as bibliotecas como um conjunto de códigos prontos, que são disponibilizados por outros programadores para realizar determinadas tarefas, podendo ainda ser desenvolvidas para trabalhar com dispositivos, tais como módulos para motores, GPS etc. O mesmo autor aponta que a utilização de bibliotecas ajuda na organização do código fonte, reduz o tempo necessário para revisar o código, e auxilia o programador na organização do projeto, fazendo deixar-se de lado o uso de códigos que não precisam ser conhecidos ou editados.

2.5.3. Shields

É possível aumentar as capacidades de controle e interação físicas e de *hardware* do Arduino. Para isso, podem ser conectados *shields*, placas que otimizam as capacidades e que podem ser facilmente plugadas ao Arduino, a fim de adicionar diversas formas de conectividade

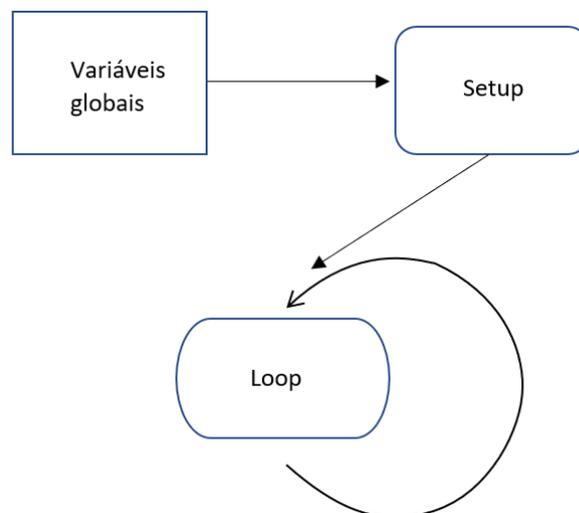
e controle. De modo geral, as *shields* contam com bibliotecas que facilitam na programação (GAIER, 2011).

2.5.4. Estrutura de um *sketch*

Geralmente, um *sketch* típico consiste em duas partes ou rotinas: a primeira rotina chamada de *setup*, é de inicialização, e a segunda, chamada de *loop*, é a que contém o código principal (EVANS; NOBLE; HOCHENBAUM, 2013).

Antes de qualquer função ou procedimento, são declaradas as variáveis globais, que podem ser acessadas de qualquer parte do código e, somente após suas declarações, é que são declaradas as funções e procedimentos do código (SANTOS, 2021). Segundo Evans, Noble e Hochenbaum (2013, p. 42) e Santos (2021, p. 91), após a declaração de variáveis globais, se encontram as rotinas *setup* e *loop*. O *setup* é executado apenas uma vez, e nele são iniciados e configurados os dispositivos de *hardware*, como definição das portas e configuração dos *shields*. No *loop* são executadas as linhas de comando, de maneira indefinida, ocorrendo leitura de sensores e acionamento de atuadores. No diagrama de blocos na Figura 2 está ilustrada a sequência de funcionamento de um *sketch*.

Figura 2: Diagrama de blocos da sequência de funcionamento de um *sketch*.



Fonte: Dos autores (2022).

2.5.5. Pinagem

A fim de se conectar com os periféricos, o Arduino dispõe de diversos pinos para conexão com sensores e atuadores. As portas são divididas em dois tipos: analógicas e digitais, sendo o último tipo dividido em binários e PWM (FBS; HACHOUCHE).

Os pinos são configurados no *sketch*, de acordo com a necessidade, como entradas ou saídas de sinais. As portas digitais trabalham com apenas dois valores de tensão, sendo 0V e 5V. Já as analógicas permitem a leitura não apenas de 0V e 5V, mas de quaisquer valores entre estes, em uma escala de 0 a 1023, na qual o 0 representa 0V e 1023 representa 5V. Estas portas são mais utilizadas para ler valores de sensores (FBS; HACHOUCHE).

Ademais, a plataforma Arduino também conta com conectores para alimentação, comunicação de dados com o computador, e pinagens de fornecimento de energia, sendo negativo (GND), dois níveis de tensão (5V e 3,3V) e alguns pinos digitais reservados para comunicação, que são responsáveis pela recepção e transmissão de dados com sensores (MONK, 2013).

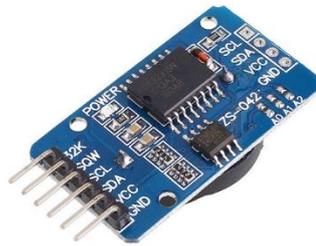
2.6. Horímetro

O horímetro é um instrumento, não resetável, que mostra a quantidade de tempo acumulado que uma máquina ou veículo funcionou. A grande maioria dos dispositivos trabalha em horas e centésimos de horas, e o acompanhamento do dispositivo possibilita, de maneira precisa, o planejamento da manutenção preventiva, a redução dos custos, o monitoramento do desempenho, da vida útil e da segurança do equipamento, além de facilitar cálculos de indicadores, como custos, gasto de energia e tempo de operação (DE SOUSA et al, 2021).

2.7. Módulo RTC

O RTC (*real time clock*) é um módulo de relógio de tempo real de alta precisão, capaz de fornecer informações de segundo, minuto, hora, dia, mês e ano. O módulo, ilustrado na Figura 3, é capaz de diferir meses com 28, 30 ou 31 dias e conta ainda com uma bateria para que este continue a funcionar mesmo em faltas de energia (WALKER, 2022).

Figura 3: Módulo RTC.

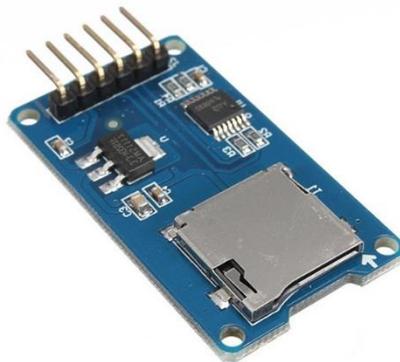


Fonte: Walker (2022).

2.8. Módulo MMC

O MMC (*micro memory card*), ilustrado na Figura 4, é um módulo de cartão micro SD, de fácil conexão, que permite fazer a leitura e gravação de dados, em cartões micro SD, utilizando o Arduino (FILIFEFLOP, 2022).

Figura 4: Módulo MMC.



Fonte: FilipeFlop (2022).

2.9. Relé

O relé, ilustrado pela Figura 5, é um dispositivo eletromecânico, constituído por um magneto móvel, que se desloca unindo dois contatos metálicos (MORIMOTO, 2010).

Figura 5: Relé.



Fonte: Dos autores (2022).

3. DESENVOLVIMENTO

3.1. Metodologia

Este projeto foi idealizado durante a realização de estágio dos autores, em uma unidade de armazenamento de grãos e em uma unidade de beneficiamento de algodão. Foi percebido que nas unidades não era utilizado nenhum indicador de manutenção preciso para a realização da manutenção dos equipamentos. Diante disso, surgiu a ideia de se fazer a instalação de horímetros nos motores, porém, o instrumento disponível no mercado tinha um custo considerável e era capaz de monitorar apenas um motor por vez. Dessa forma, surge a ideia de implementar um microcontrolador, instalado no centro de comando de motores, para que, com apenas um dispositivo, fosse possível monitorar e registrar os dados de todos em apenas uma planilha, sem a necessidade de delegar/atribuir e criar uma rotina diária de coleta de dados em horímetros individuais.

3.2. Requisitos funcionais

Como requisito funcional, foi estabelecido que o sistema, ao ser instalado num CCM (centro de controle de motores), fosse capaz de acompanhar o *status* de funcionamento das cargas ao longo do tempo. Sendo capaz de fazer coletas de informações pontuais, em um intervalo de tempo configurável na programação do microcontrolador, e que, a cada coleta, grave em uma planilha em um cartão de memória, o *status* de cada carga, seguido do dia da semana e das informações de hora/minuto/segundo e dia/mês/ano no instante da coleta, como na Figura 6.

Figura 6: Exemplos dos dados a serem gravados pelo sistema.

	A	B	C	D	E
25	Qua	16/02/2022	10:32:31	Elev1	1
26	Qua	16/02/2022	10:32:31	Elev2	1
27	Qua	16/02/2022	10:32:31	Cor1	0
28	Qua	16/02/2022	10:32:31	Cor2	0
29	Qua	16/02/2022	10:32:32	Elev1	1
30	Qua	16/02/2022	10:32:32	Elev2	1
31	Qua	16/02/2022	10:32:32	Cor1	1
32	Qua	16/02/2022	10:32:32	Cor2	0
33	Qua	16/02/2022	10:32:33	Elev1	1
34	Qua	16/02/2022	10:32:33	Elev2	1
35	Qua	16/02/2022	10:32:33	Cor1	1
36	Qua	16/02/2022	10:32:33	Cor2	0
37	Qua	16/02/2022	10:32:34	Elev1	1
38	Qua	16/02/2022	10:32:34	Elev2	1
39	Qua	16/02/2022	10:32:34	Cor1	1
40	Qua	16/02/2022	10:32:34	Cor2	0
41	Qua	16/02/2022	10:32:35	Elev1	1

Fonte: Dos autores (2022).

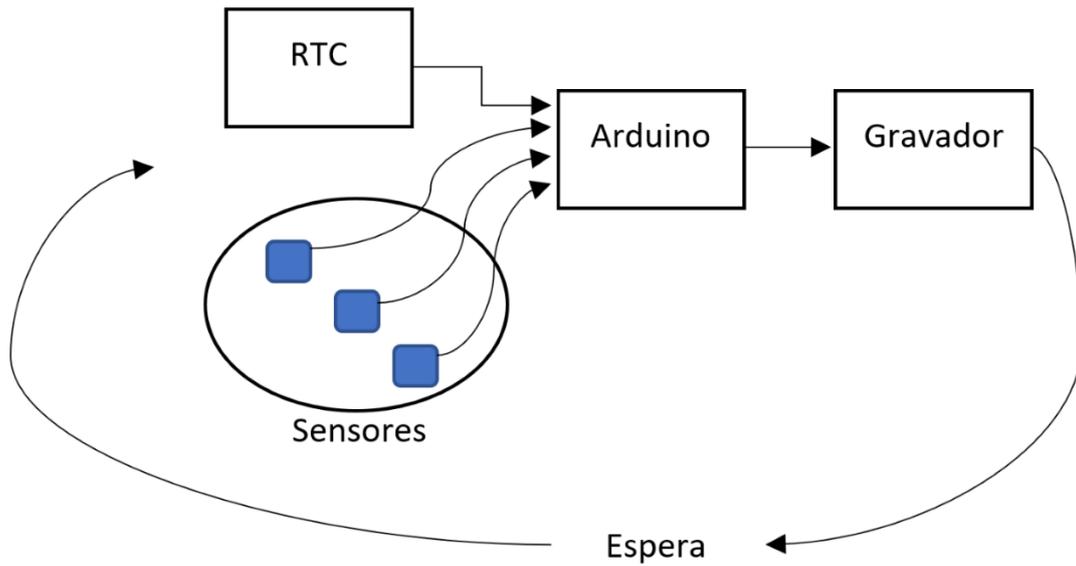
3.3. Funcionamento da plataforma Arduino

3.3.1. Visão geral

De maneira geral, o sistema foi constituído por: um sensor que permite que o Arduino acompanhe o status da carga; um relógio que, sempre que solicitado, envia informações de tempo para o Arduino; três LEDs, que quando acesos, de maneira individual, indicam respectivamente falha no módulo de cartão micro SD, falha no RTC e efetuação da leitura das cargas e gravação; e um microcontrolador Arduino que, ao ser programado com um código, fica a cada intervalo de tempo, atribuído no código, solicitando que os sensores verifiquem o *status* de cada carga e as informações de tempo do RTC, agrupa as informações de tempo com o estado da máquina e, através do gravador de cartão micro SD, registra essas informações em uma planilha eletrônica dentro do cartão.

Em um primeiro momento era feita a coleta de informações do RTC e, na sequência, era feito um ciclo de leitura seguida de gravação de uma carga por vez, até que fossem lidas todas e, em seguida, era esperado um tempo e o ciclo se repete em *looping*. A lógica de funcionamento está ilustrada no fluxograma de informações da Figura 7.

Figura 7: Diagrama de blocos: logica de funcionamento do sistema.



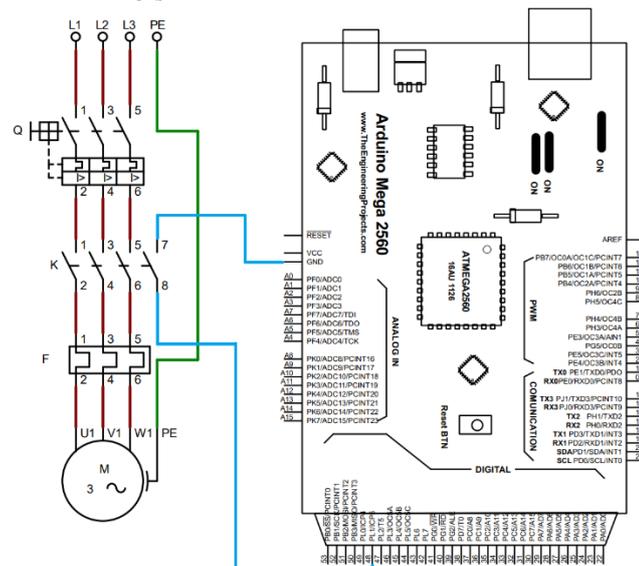
Fonte: Dos autores (2022).

3.3.2. Percepção do *status* da carga

Podem serem adotados diversos métodos para realizar o monitoramento da carga utilizando a plataforma Arduino. Uma maneira muito simples é configurar uma porta digital desejada como entrada, vinculá-la a uma carga, por meio do código, de maneira que essa porta sempre corresponda à carga em questão, e assim o microcontrolador irá interpretar essa entrada como um indicativo do estado da carga. As referências de estado são o GND (0V) e o VCC (5V), e basta utilizar um sensor que feche curto com a entrada digital e uma referência de tensão, para que o Arduino converta a leitura da porta no *status* da carga. Como estratégia para assegurar que sempre que o sensor estiver com circuito aberto o Arduino não confunda a leitura, podem ser utilizadas técnicas de resistores *pull-up* ou *pull-down*.

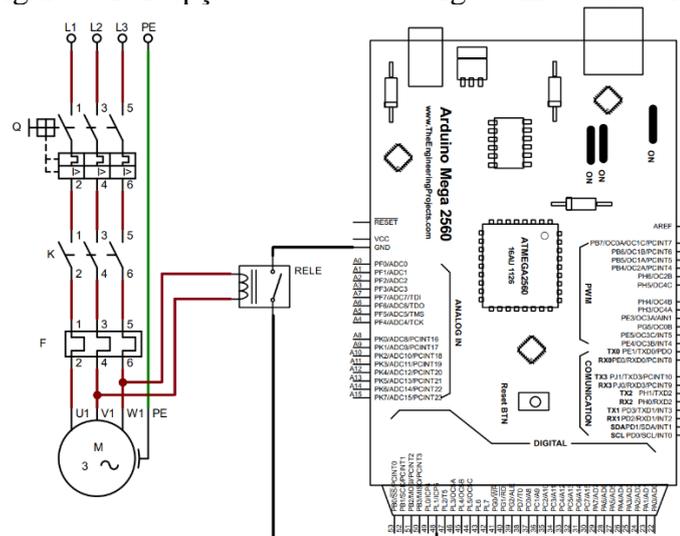
Para a percepção do estado do motor, pode ser aproveitado um contato auxiliar de algum contator utilizado no acionamento do mesmo, como ilustrado na Figura 8, no qual o contato auxiliar está conectado com o GND e com o pino digital 48 do Arduino, ou ainda, pode-se instalar em paralelo com os terminais do motor um relé, que fechará curto quando os terminais do motor estiverem energizados, Figura 9.

Figura 8: Percepção com status com uso de contato auxiliar.



Fonte: Dos autores (2022).

Figura 9: Percepção do status da carga com o uso do relé.



Fonte: Dos autores (2022).

3.4. Lógica de funcionamento do código

O algoritmo utilizado no Arduino, para este projeto, pode ser resumido em três etapas, e um resumo de cada uma delas está descrito abaixo:

Primeira etapa:

- Foram importadas as bibliotecas necessárias para trabalhar com os módulos MMC e RTC; e
- Foram declaradas as variáveis responsáveis por fazer as leituras, armazenar o nome de cada carga monitorada e os dados gravados no cartão SD.

Segunda etapa (*setup*):

- Foram inicializados e verificada a efetividade de cada modulo e, caso encontrada alguma falha, o Arduino entrava em *loop*, que fazia com que um LED correspondente à falha detectada fosse acionado;
- Na primeira inicialização do modulo RTC, era utilizado o comando *rtc.adjust*, inserindo nele as informações de data e hora atuais;
- Nesta etapa, também era declarada a função de cada porta digital a ser utilizada, sendo de saída para os LEDs e entrada para as cargas.

Terceira etapa (*loop* infinito): Etapa em que o Arduino fazia a verificação do *status* de cada carga e gravava o resultado na planilha eletrônica:

- No início do *loop*, era feita a aquisição das informações de tempo do RTC;
- Na sequência, era realizada a abertura da planilha eletrônica no cartão SD, ou criada uma nova planilha, caso não fosse encontrada nenhuma;
- Se por algum motivo, ocorresse um erro na abertura da planilha, o código caía em um *loop* infinito, que fazia um LED piscar, indicando a respectiva falha;
- Seguindo tudo normal, o código acendia um LED que indicava o momento em que estava sendo feita a gravação de dados no cartão SD e, na sequência, era feita uma rotina em que a leitura de cada carga era realizada de forma individual, e o resultado da leitura era armazenado em uma variável do tipo texto. Posteriormente, os dados contidos na variável eram gravados na primeira linha vazia da planilha, e essa rotina era feita até que fosse feita a leitura de todas as cargas contidas no código, quando a planilha era fechada;
- Finalizada a etapa anterior, o código mandava desligar o LED indicativo de gravação no cartão; e
- Por fim, iniciava-se uma contagem de tempo, configurável no código, para iniciar novamente o *loop*.

Vale ressaltar que o tempo de espera para iniciar novamente o *loop*, neste estudo, era configurável dentro do código, uma vez que o protótipo não contava com uma interface para comunicação com o usuário e, para desconectar o cartão SD, era necessário desligar a fonte de alimentação do Arduino, procedimento que poderia ser feito com o uso de um interruptor.

3.5. Simulação

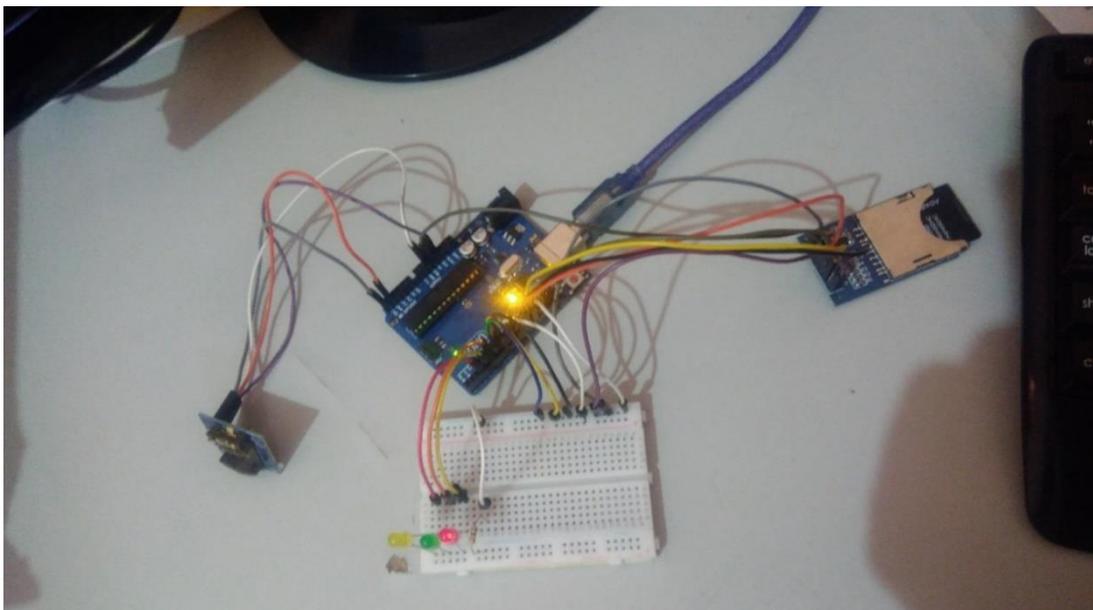
Para verificar a estabilidade e o funcionamento da ferramenta desenvolvida, foi realizado um ensaio em protoboard, conforme ilustrado na Figura 10, no qual não se utilizou o relé, uma vez que sua função era apenas de fechar contato entre a porta digital do Arduino e o GND. Para tal, foi feito de maneira manual, apenas conectando e desconectando os respectivos fios no trilho GND da *protoboard*, como ilustrado pela Figura 11.

No código, é importante determinar um intervalo de tempo, de acordo com a precisão necessária, que correspondo à frequência que o sistema irá fazer as leituras do status das cargas e gravação de dados no cartão de memória. Este tempo deve ser adequado, de forma que não seja muito grande e resulte em omissão de dados, mas também não deve ser muito pequeno, de maneira que acabe realizando coletas muito próximas, resultando numa certa “redundância” de informações, mas também criando uma planilha de dados extremamente extensa, que dificulta a utilização e tratamento destas informações. Para realização da etapa de testes, foi adotado, arbitrariamente, um intervalo de 5 minutos.

Para avaliação da ferramenta desenvolvida, foram avaliados a estabilidade geral de funcionamento do sistema, verificando a possibilidade de ocorrência de falhas para um dado tempo de funcionamento, a coerência da lógica de funcionamento do código com o acionamento dos LEDs de indicação, e foi feita uma comparação do tempo marcado pelo RTC com o tempo corrido no relógio do computador.

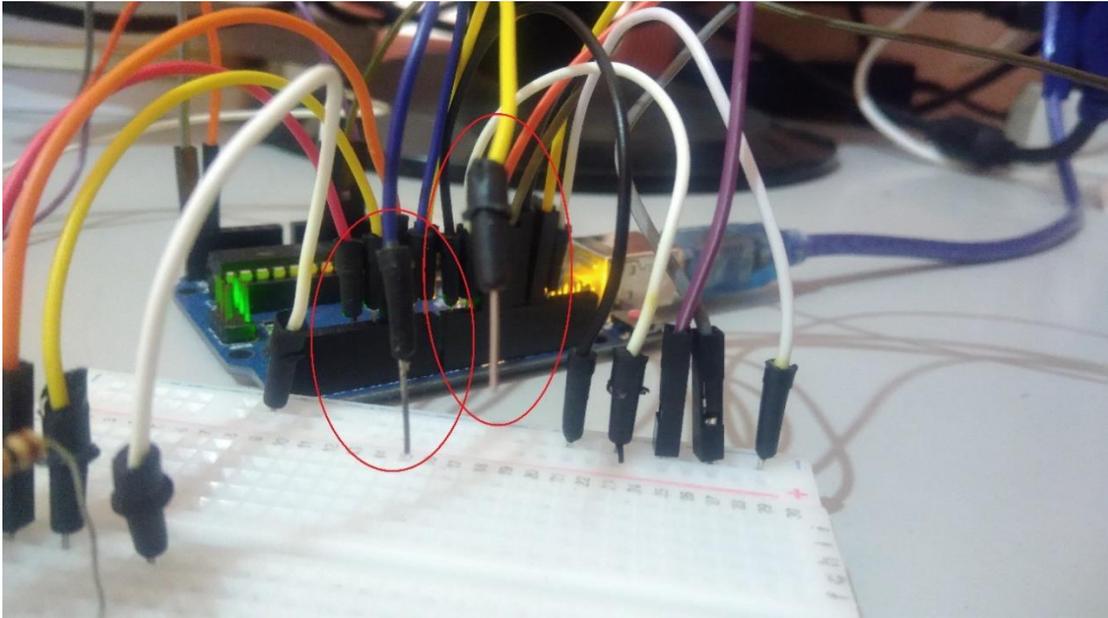
Por fim, foi feito também um orçamento, com objetivo de verificar os custos de aquisição para a elaboração do projeto e comprar com a utilização de horímetros convencionais.

Figura 10: Ensaio utilizando protoboard.



Fonte: Dos autores (2022).

Figura 11: Desconexão dos fios na protoboard.



Fonte: Dos autores (2022).

4. RESULTADOS E DISCUSSÕES

Após a programação e montagem do protótipo, foi realizado um teste, com o objetivo de verificar a lógica de funcionamento do algoritmo.

4.1. Testes e simulações

Foram testados o funcionamento dos LEDs de indicação de falhas e gravação de dados, fazendo-se o desacoplamento dos módulos a fim de testar a efetividade do código, o funcionamento do modulo RTC, verificando coerência com o relógio do computador, e a assiduidade do modulo MMC ao gravar os dados no cartão SD.

Primeiramente, por meio de um teste foi avaliada a estabilidade do código, mais precisamente em relação às repostas de comunicação dos módulos. Foi constatado adequado funcionamento dos LEDs de indicação de falhas e gravação de dados.

Na etapa seguinte, o protótipo foi colocado para trabalhar de maneira efetiva, iniciando às 09h16min (relógio do computador) do mesmo dia e, finalizando, no mesmo dia, às 21h02min, com duração de 11h46min horas. Esta etapa foi importante para comparar a precisão do modulo RTC. Comparando-o com o relógio do computador, a fim de obter uma planilha de dados criada pela ferramenta desenvolvida e verificar a estabilidade de funcionamento do sistema como um todo. O procedimento teve como principal resultado a própria planilha que o

sistema criou. Com ela em mãos, foi possível observar a estabilidade de funcionamento do código, por meio da quantidade de linhas criadas na planilha, Figura 12.

Uma vez que o código estava configurado para registrar dados em intervalos de 5 em 5 minutos e ele monitorava 4 cargas, era esperado que o número de linhas criadas na planilha fosse igual a quatro vezes o número de horas trabalhadas dividido por 5 minutos, o resultado desta conta seria 564 linhas. Porém, como o sistema fazia o primeiro registro imediatamente após ser iniciado, deve-se somar mais quatro leituras, totalizando 568 linhas, estando em conformidade com o número de linhas criadas na planilha, conforme ilustrada na Figura 12. Este resultado, por si só, confirma a estabilidade geral de funcionamento do sistema como um todo e a precisão do módulo RTC, se comprado ao relógio do computador.

O resultado das horas registradas pelo sistema também pode ser facilmente observado por meio da Figura 13, em que é possível verificar que houve conformidade entre as horas de início e fim de operação, reforçando a estabilidade do módulo RTC, quando comprado ao relógio do computador.

Figura 12: Planilha criada pelo sistema após ensaio.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
545	Seg	18/04/2022	20:36:36	Mesa	1															
546	Seg	18/04/2022	20:36:36	Elevador	1															
547	Seg	18/04/2022	20:36:36	Bomba	0															
548	Seg	18/04/2022	20:36:36	Exaustor	0															
549	Seg	18/04/2022	20:41:36	Mesa	1															
550	Seg	18/04/2022	20:41:36	Elevador	1															
551	Seg	18/04/2022	20:41:36	Bomba	0															
552	Seg	18/04/2022	20:41:36	Exaustor	0															
553	Seg	18/04/2022	20:46:36	Mesa	1															
554	Seg	18/04/2022	20:46:36	Elevador	1															
555	Seg	18/04/2022	20:46:36	Bomba	0															
556	Seg	18/04/2022	20:46:36	Exaustor	0															
557	Seg	18/04/2022	20:51:36	Mesa	1															
558	Seg	18/04/2022	20:51:36	Elevador	1															
559	Seg	18/04/2022	20:51:36	Bomba	0															
560	Seg	18/04/2022	20:51:36	Exaustor	0															
561	Seg	18/04/2022	20:56:36	Mesa	1															
562	Seg	18/04/2022	20:56:36	Elevador	1															
563	Seg	18/04/2022	20:56:36	Bomba	0															
564	Seg	18/04/2022	20:56:36	Exaustor	0															
565	Seg	18/04/2022	21:01:36	Mesa	1															
566	Seg	18/04/2022	21:01:36	Elevador	1															
567	Seg	18/04/2022	21:01:36	Bomba	0															
568	Seg	18/04/2022	21:01:36	Exaustor	0															
569																				
570																				
571																				
572																				
573																				

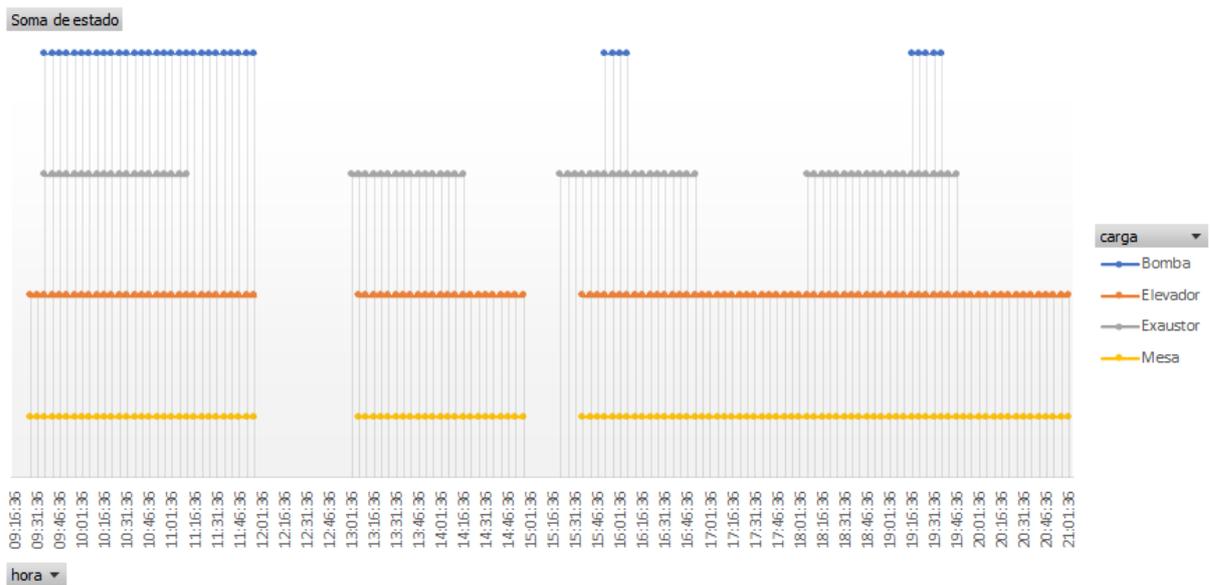
Fonte: Dos autores (2022).

A fim de se obter um resultado de melhor visualização, os dados da planilha foram trabalhados a fim de criar o gráfico da Figura 13.

Diante do gráfico ilustrado na Figura 13, possibilitou-se verificar o *status* de funcionamento de cada carga no decorrer do dia, bem como os momentos em que todas as

cargas estavam ligadas ou, em outras palavras, os instantes de tempo em que houve maior consumo de energia. Estas informações, dispostas de maneira visual, facilitam nas tomadas de decisões para otimizar o consumo de energia elétrica nas indústrias, nas maneiras de buscar melhores efetividades no tempo de funcionamentos das máquinas, na identificação das falhas no ciclo de produção, no cálculo dos custos de produção, dentre outros.

Figura 13: Gráfico ilustrativo de funcionamento das cargas testadas neste estudo (bomba, elevador, exaustor e mesa), ao longo do período experimental (09h16min às 21h02min).



Fonte: Dos autores (2022).

4.2. Orçamento

Para verificação dos custos de montagem e aquisição da ferramenta proposta, estão apresentados a seguir os orçamentos para a simulação de instalação de 20 horímetros. Os dados utilizados para montagem dos orçamentos foram obtidos via pesquisa na internet, com base em três fornecedores. Para cálculo de frete, como alguns itens podem ser encontrados em um mesmo vendedor, foi simulado um valor de 10% sobre o total.

São dois orçamentos. O primeiro orçamento é para os horímetros convencionais do mercado. O segundo é com base na plataforma apresentada no projeto com sensor de corrente encontrado no mercado.

4.2.1. Horímetro convencional

O modelo de horímetro cotado foi semelhante ao da Figura 14.

Figura 14: Horímetro.



Fonte: Dos autores (2022).

Tabela 1- Orçamento para Horímetro Convencional.

Descrição	Quantidade (un)	Preço médio (R\$)	Preço total (R\$)
Horímetro – Totalizador Horário	20	R\$ 113,32	R\$ 2.226,20

Fonte: Dos autores (2022).

Vale ressaltar, que o valor de compra do material com o frete simulado a 10%, seria de aproximadamente R\$ 2.448,82.

4.2.2. Horímetro Datalogger

No orçamento a seguir, foram cotados os valores para todos componentes e materiais que foram utilizados na confecção da ferramenta de baixo custo desenvolvida neste estudo (horímetro com Arduino).

Tabela 2 - Orçamento para horímetro DATALOGGER

Item	Descrição	Quantidade (un)	Preço Médio (R\$)	Preço total (R\$)
I.	Arduíno MEGA 2560 PRO MINI	1	R\$ 139,43	R\$ 139,43
II.	RTC DS1307	1	R\$ 13,08	R\$ 13,08
III.	Módulo cartão micro SD	1	R\$ 10,85	R\$ 10,85
IV.	Fonte DC 9V 1A	1	R\$ 18,43	R\$ 18,43
V.	Bateria LIR20332	1	R\$ 2,17	R\$ 2,17
VI.	Cartão micro SD 2GB	1	R\$ 16,77	R\$ 16,77
VII.	Relé acoplador interface	20	R\$ 24,30	R\$ 485,93
				754,60

Fonte: Dos autores (2022).

O valor total do orçamento foi de R\$ 686,67. Então, ao simular o frete em 10% do valor da compra, estima-se que será gasto cerca de R\$ 754,60 na compra dos componentes e materiais.

5. CONCLUSÕES

Por meio do presente estudo, foi possível desenvolver e validar uma ferramenta de baixo custo e fácil aplicação, com base na plataforma Arduino. Esta ferramenta se apresenta como uma alternativa prática para coleta das informações de horas trabalhadas por motores elétricos, frente aos horímetros convencionais disponíveis no mercado.

O sistema proposto foi capaz de registrar dados em tempo real, quando as cargas estavam ligadas ou desligadas, gerando um histórico preciso, que permitiu saber se a máquina estava parada ou funcionando para um dado momento. Essas informações podem servir de base para a tomada de decisão de diversos processos importantes em uma indústria.

A implantação do sistema se mostrou viável economicamente, quando comparada com a utilização de horímetros para indústrias, pois permite o monitoramento de diversas cargas simultaneamente.

REFERÊNCIAS

- ABNT. **ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS: Confiabilidade e manutenibilidade.** NBR ISSO 5462. Rio de Janeiro, 1994.
- ARDUINO. **Site Oficial.** Disponível em: < <https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 01 de março de 2022.
- CAPELLI, A. **Automação Industrial: controle do movimento e processos contínuos.** 3 ed. São Paulo: Érica, 2013.
- CHAPMAN, S. J. **Fundamentos de Máquinas Elétricas.** v. 53. 1989.
- CUNHA, A. **Sistemas Embarcados.** Revista Saber Eletrônica, 2007.
- DE SOUSA, V. M. R.; et al. **Gestão da Manutenção Preventiva de Empilhadeiras em uma empresa de pequeno porte utilizando uma ferramenta IOT integrada a um software.** 2021.
- DICIO. **Significado de manutenção.** 2022. Disponível em: <<https://www.dicio.com.br/manutencao/>>. Acesso em: 16/03/2022.
- EVANS, M.; NOBLE, J.; HOCHENBAUM, J. **Arduino em ação.** Novatec Editora, 2013.
- FBS. **Apostila Arduino UNO.** Fbseletronica. Disponível em: <www.fbseletronica.com.br> Acesso em: 06 de fevereiro de 2022.
- FERREIRA, M. I. J. **Planejamento da Manutenção de Sistemas de Distribuição Baseado em Índices de Confiabilidade.** Rio de Janeiro, 2019.
- Fitzgerald, A. E. et al. **Máquinas elétricas.** tradução Anatólio Laschuk. 6. ed. Porto Alegre: Bookman, 2006.
- FILIPEFLOP. **Módulo cartão micro sd.** Disponível em: < <https://www.filipeflop.com/produto/modulo-cartao-micro-sd/>> Acesso em: 22 de março de 2022.
- FRANCHI, C. M. **Acionamento Elétricos.** 1 ed. São Paulo, 2007.
- GAIER. **Aprendendo a programar em Arduino.** PET: IFMT: Campus Cuiabá, 2011.
- GARCIA, A. P. **Impacto da Lei de Eficiência Energética para Motores Elétricos no Potencial de Conservação de Energia na Indústria.** Rio de Janeiro, 2003.
- GER. **Introdução ao Arduino.** Grupo de estudo em robótica: Unicamp: Campinas, 2009.
- GONÇALEZ, F. G. **Estudo do motor de indução trifásico e desenvolvimento de um dispositivo de proteção efetiva de motores operando em condições anormais: rotor bloqueado e falta de fase.** Florianópolis: SC, 2007.

HACHOUCHE, A. S. **Apostila Arduino Básico V1.0**. Eletrogate. Disponível em: <www.eletrogate.com> Acesso em 06 de fevereiro de 2022.

MAMEDE, F. J. **Instalações elétricas industriais: de acordo com a norma brasileira**. 9. ed. Rio de Janeiro, 2017.

MONK, S. **Programação com Arduino: começando com sketches**. Porto Alegre: Bookman, 2013.

MORIMOTO, C. E. **Dicionário de termos de informática**. 3. ed. Guia do hardware.net. Guarulhos: SP, 2010.

NEPOMUCENO, L. X. **Técnicas de manutenção preditiva**. São Paulo: Bluncher, 1989.

NOGUEIRA, C. F.; GUIMARÃES, L. M.; da SILVA, M. D. B. **Manutenção Industrial: implementação da manutenção produtiva total (TPM)**. Belo Horizonte, 2012.

PINTO, A. K.; XAVIER, J. A. N. **Manutenção Função Estratégica**. 3. ed. Rio de Janeiro: Qualitymark, 2009.

RODRIGO, L. **Plataforma Arduino, Hardware e visão geral**. 2011.

RODRIGUES, L. M. **Trabalhadores, sindicatos e industrialização**. 2009.

ROMANO, L. N. **Modelo De Referência Para O Processo De Desenvolvimento De Máquinas Agrícolas**. 2003.

SANTOS, C. R. B. **Fundamentos de Projetos Eletrônicos Envolvendo o Arduino**. Rio de Janeiro: Ciência Moderna Ltda., 2021.

STRIZHAK, V.; PENKOV, I.; PAPPEL, T. **Evolution of Design, Use and Strength Calculations of Screw Threads and Threaded Joints**. 2007.

WALKER, M. **Electronic Shop**. Disponível em: <<https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-modulo-real-time-clock-rtc-ds3231>> Acesso em 20 de março de 2022.

WERNECK, P. **Introdução ao Arduino**. 2009.

APÊNDICE 1 - SKETCH DO PROJETO

```
// Inclusão das bibliotecas do RTC e MMC e, definição do pino CS.
#include <Wire.h>
#include "RTCLib.h"
#include <SPI.h>
#include <SD.h>
#define CS_pin 10

// Vetores que armazenam as cargas e as portas nas respectivas ordens
byte Portas[27] = {6, 7, 8, 9};
String Cargas[27] = {"Mesa", "Elevador", "Rosca", "Bomba"};
String linha = "";

// Inicialização do modulo RTC.
RTC_DS1307 rtc;

// Atribuição das abreviações dos dias da semana.
char daysOfTheWeek[7][12] = {"Dom", "Seg", "Ter", "Qua", "Qui", "Sex", "Sab"};

void setup () {
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  Serial.begin(9600);

  //Verifica se o rtc está conectado.
  if (! rtc.begin()) {
    while (1){
      Serial.println("RTC não conectado");
      digitalWrite(3,HIGH);
      delay(500);
      digitalWrite(3,LOW);
      delay(500);
    }
  }
}
```

```

    }
}

// verifica se é possível comunicar com o rtc.
if (! rtc.isrunning()) {
    Serial.println("RTC comunica");
    digitalWrite(3,HIGH);
    delay(500);
    digitalWrite(3,LOW);
    delay(500);
    rtc.adjust(DateTime(2022, 1, 21, 10, 13, 10));
}

// faz a declaração de cada porta digital de leitura.
for (int p = 5; p <= 9; p++){
    pinMode(p, INPUT_PULLUP);
}

//Verifica se o SD está conectado
if (!SD.begin(11)) {
    while (1){
        Serial.println("SD não conectado");
        digitalWrite(4,HIGH);
        delay(500);
        digitalWrite(4,LOW);
        delay(500);
    }
}

void loop () {
    //Faz a coleta das informações de data e hora do rtc.
    DateTime now = rtc.now();

```

```

//Caso não exista, cria a planilha e a vincula dentro de uma variável.
File myFile = SD.open("Horímetro.csv", FILE_WRITE);
linha = String(daysOfTheWeek[now.dayOfTheWeek()]+";"+ String(now.day()) + "/" +
String(now.month()) + "/" + String(now.year()+";"+ String(now.hour()) + ":" +
String(now.minute()) + ":" + String(now.second()+";"+ String(Cargas[0]+";"+
String(!digitalRead(Portas[6]));
Serial.println(linha);

//Abre a planilha eletrônica contida na raiz do cartão sd.
if(myFile)
{
Serial.println("SD gravou");
//LED que indica gravação do cartão sd.
digitalWrite(5,HIGH);
for (int p = 0; p <= 3; p++){
linha = String(daysOfTheWeek[now.dayOfTheWeek()]+";"+ String(now.day()) + "/" +
String(now.month()) + "/" + String(now.year()+";"+ String(now.hour()) + ":" +
String(now.minute()) + ":" + String(now.second()+";"+ String(Cargas[p]+";"+
String(!digitalRead(Portas[p]));
myFile.println(linha);
}
myFile.close();
digitalWrite(5,LOW);
}

else{
Serial.println("Falha ao gravar cartão SD!");
while(1){
Serial.println("SD não gravou");
digitalWrite(4,HIGH);
delay(500);
digitalWrite(4,LOW);
delay(500);
}
}

```

```
}  
//tempo de espera (em milissegundos) para reiniciar o loop.  
delay(300000);  
}
```