

**MEC - SETEC
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE MINAS GERAIS – CAMPUS FORMIGA**

**PROPOSIÇÃO DE SISTEMA DE TRANSMISSÃO DE LONGA
DISTÂNCIA PARA CONTROLE DE VELOCIDADE DE MOTOR DE
INDUÇÃO TRIFÁSICO**

Bruno Gonçalves Silva

Orientadora: Prof^ª. MSc. Ana Paula Lima dos
Santos.

Coorientador: Prof. Dr. Gustavo Lobato
Campos

FORMIGA - MG

2019

BRUNO GONÇALVES SILVA

**PROPOSIÇÃO DE SISTEMA DE TRANSMISSÃO DE LONGA DISTÂNCIA PARA
CONTROLE DE VELOCIDADE DE MOTOR DE INDUÇÃO TRIFÁSICO**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica do Instituto Federal de Minas Gerais, como requisito para obtenção do título de bacharel em Engenharia Elétrica.

Orientadora: Prof^ª. Msc. Ana Paula Lima dos Santos.

Coorientador: Prof. Dr. Gustavo Lobato Campos.

FORMIGA - MG

2019

Silva, Bruno Gonçalves

S586p Proposição de sistema de transmissão de longa distância para controle de velocidade de motor de indução trifásico / Bruno Gonçalves Silva.
-- Formiga : IFMG, 2019.
80p. : il.

Orientador: Prof. MSc. Ana Paula Lima dos Santos

Co-orientador: Prof. Dr. Gustavo Lobato Campos

Trabalho de Conclusão de Curso – Instituto Federal de Educação,
Ciência e Tecnologia de Minas Gerais – *Campus* Formiga.

1. Arduino. 2. Inversor de frequência. 3. IoT. 4. LoRaWAN
5. Motor de indução. I. Santos, Ana Paula Lima dos. II. Campos, Gustavo Lobato. III. Título.

CDD 621.3

BRUNO GONÇALVES SILVA

**PROPOSIÇÃO DE SISTEMA DE TRANSMISSÃO DE LONGA DISTÂNCIA PARA
CONTROLE DE VELOCIDADE DE MOTOR DE INDUÇÃO TRIFÁSICO**

Trabalho de Conclusão de Curso apresentado ao
Curso de Engenharia Elétrica do Instituto Federal
de Minas Gerais como requisito para obtenção do
Título de Bacharel em Engenharia Elétrica.

Avaliado em: 02 de dezembro de 2019.

Nota: 99

BANCA EXAMINADORA

Ana Paula Lima dos Santos

Orientadora: Prof^a. MSc. Ana Paula Lima dos Santos

Gustavo Lobato Campos

Coorientador: Prof. Dr. Gustavo Lobato Campos

Danielle Costa de Oliveira

Prof^a. MSc. Danielle Costa de Oliveira

Lucas Frederico Jardim Meloni

Prof. MSc. Lucas Frederico Jardim Meloni

Gabriel Ramos Teixeira

Bacharel em Engenharia Elétrica Gabriel Ramos Teixeira

Dedico esse trabalho aos meus pais José Maria e Maria Irene por todo amor, apoio, incentivo e dedicação que me deram durante toda minha vida e jornada acadêmica.

AGRADECIMENTOS

Primeiramente agradeço a Deus pela saúde, paz, paciência e força a mim concedidas, que me proporcionou superar e concluir mais uma etapa em minha vida.

Aos meus pais José Maria e Maria Irene, por terem me apoiado e por me proporcionarem a realização deste sonho.

A minha irmã Brenda e meu irmão Bernardo pelo carinho e por terem me apoiado nesta jornada.

A minha namorada Hellen por todo incentivo, amor, paciência e carinho para superar diversos desafios.

A minha orientadora, Prof^a. Msc. Ana Paula Lima dos Santos, e meu coorientador, Prof. Dr. Gustavo Lobato Campos, pela disponibilidade e orientações no desenvolvimento deste trabalho.

A todos meus amigos pela paciência, compreensão, companheirismo e amizade.

Por fim, agradeço a todos professores e funcionários IFMG – *Campus* Formiga, que de algum modo colaboraram para a conclusão desta jornada.

“Se você quiser descobrir os segredos do Universo, pense em termos de energia, frequência e vibração”.

(Nikola Tesla)

RESUMO

As mudanças ocorridas no panorama industrial nos últimos dez anos conduziram a um cenário de inovações e fusões de tecnologias, hoje caracterizadas como Indústria 4.0. A 4ª Revolução Industrial elevou o nível de automação, inteligência artificial e o uso da tecnologia IoT (Internet das Coisas), que permitem a fusão de mundos físicos, digitais e biológicos, e, consequentemente, admitem descentralização, interoperabilidade e virtualização de processos. Nestes sistemas são empregados vários tipos de comunicação sem fio, que possam atender às necessidades do processo impostos, tais como as tecnologias Wi-Fi, *Bluetooth* e as LPWAN (*Low Power Wide Area Networks*), que são divididas em redes licenciadas (LTE, EC-GSM e NB-IoT) e não licenciadas (IEEE 802.11ah, Weightless, Ingenu, SigFox e LoRa). A tecnologia LoRa, gerida pela empresa Alliance LoRa, se adequa a todos os requisitos das LPWAN, com longo alcance de transmissão e baixo consumo de energia. Esta rede é muito empregada na transmissão de dados oriundos de sensores, tais como velocidade, temperatura, e pressão, assim como pode transmitir sinais para a operação de atuadores na indústria. Com o propósito de apresentar uma aplicação para o uso desta tecnologia, este trabalho propõe o desenvolvimento de um protótipo em escala reduzida de um sistema embarcado, para o controle de partida e velocidade de um motor de indução trifásico (MIT). Para este projeto foi desenvolvido um ambiente de monitoramento, por meio do *software* Processing e comunicação serial com um *Dragino LoRa Shield* acoplado a um Arduino Uno (caracterizados como *gateway*), que juntos transmitem os comandos a outro circuito receptor (*gateway* e *end-device*, também formado com um *Dragino LoRa Shield* acoplado a um Arduino Uno). O circuito receptor é conectado a um inversor de frequência, para que dessa forma as entradas digitais do inversor possam ser utilizadas para o controle remoto da velocidade do motor. Os testes de controle da velocidade do MIT foram realizados no Laboratório de Máquinas do Instituto Federal de Minas Gerais – *Campus* Formiga e o alcance máximo conseguido na transmissão dos dados foi de 806,91m. Os resultados obtidos com o desenvolvimento deste projeto mostraram que é possível realizar o controle da partida e velocidade de um MIT de forma remota com comunicação sem fio e a longa distância.

Palavras chave: Arduino, Inversor de frequência, IoT, LoRaWAN, Motor de indução.

ABSTRACT

The changes have taken place in an industrial landscape in the last ten years, they have occurred in a scenario of innovations and mergers of technologies, today characterized as Industry 4.0. The 4th Industrial Revolution, the level of automation, artificial intelligence and use of IoT (Internet of Things) technology, which uses the fusion of physical, digital and biological backgrounds and, consequently, allows for decentralization, interoperability and process virtualization. Nested systems employ various types of wireless communication, which may meet imposed process requirements, such as Wi-Fi, Bluetooth, and LPWAN (Low Power Wide Area Networks) technologies, which are divided into licensed networks (LTE, EC - GSM and NB-IoT) and unlicensed (IEEE 802.11ah, Weightless, Ingenu, SigFox and LoRa). LoRa technology, managed by Alliance LoRa, meets all LPWAN requirements with long transmission range and low power consumption. This network is widely used in the transmission of data from sensors, such as speed, temperature and pressure, as well as can transmit signals for the operation of industry agents. In order to present an application for the use of this technology, this work applied the development of a prototype of an embedded system to control the start and speed of the three phase induction motor (MIT). For this project, a monitoring environment was developed, through software Processing and serial communication with a Dragino LoRa Shield activated for an Arduino Uno (characterized as a gateway), which transmits together the commands of another receiving circuit (gateway and end device, also formed with a Dragino LoRa Shield included with Arduino Uno). The receiver circuit is connected to a frequency inverter, so that how the inverter digital inputs can be used for remote motor speed control. MIT speed control tests were not performed at the Machinery Laboratory of the Federal Institute of Minas Gerais - Formiga Campus and the maximum range achieved in data transmission was 806.91m. The results obtained from the development of this project show that it is possible to perform remote start and speed control of an MIT with long distance wireless communication.

Keywords: Arduino, Frequency Inverter, Induction Motor, IoT, LoRaWAN.

LISTA DE ILUSTRAÇÕES

Figura 1 - Arquitetura LoRa	25
Figura 2 - Arquitetura LoRaWAN.....	26
Figura 3 - Formato de mensagem LoRaWAN.....	28
Figura 4 - Construção geral de uma máquina de indução.	29
Figura 5 - Desenho esquemático de um rotor gaiola de esquilo.....	30
Figura 6 - Interface Processing 3.5.2	32
Figura 7 - Placa de composição do Arduino UNO.....	34
Figura 8 - Interface Arduino IDE 1.8.6	34
Figura 9 - Mapeamento de pinos <i>Dragino LoRa Shield</i> v1.4.....	35
Figura 10 - Blocos dos componentes básicos de um inversor de frequência	37
Figura 11 - Diagrama da etapa de potência do inversor de frequência	38
Figura 12 - Multispeed	39
Figura 13 - Montagem do <i>Dragino LoRa Shield</i> com Arduino.....	42
Figura 14 - Circuito receptor com módulo relés e inversor.....	43
Figura 15 - Funções para comunicação Serial entre Processing e Arduino	44
Figura 16 - Função para enviar informação pela porta Serial	45
Figura 17 - Função para ler as informações da porta Serial	46
Figura 18- Ambiente de monitoramento do controle de velocidades.....	46
Figura 19 - Funções que enviam pacotes de dados pelo <i>Dragino LoRa Shield</i>	47
Figura 20 - Função para leitura de sinais com o <i>Dragino LoRa Shield</i>	48
Figura 21 - Circuito transmissor e receptor com botões e LEDs.	49
Figura 22 – Circuito transmissor (a) e circuito receptor (b)	49
Figura 23 - Placa com as características do MIT.....	50
Figura 24 - Parametrização do MIT	51
Figura 25 - Parametrização da função <i>Multispeed</i>	51
Figura 26 - Parametrização das entradas digitais DI1 e DI4 a DI6	52
Figura 27 - Parametrização das velocidades.....	52
Figura 28 - Montagem para controle de velocidade do MIT por meio da comunicação sem fio	53
Figura 29 - Circuito escravo com inversor de frequência e motor trifásico	54
Figura 30 - Mapa das distâncias de transmissão com 512,49m.	54
Figura 31 - Mapa da transmissão de dados com 595,79m.	55
Figura 32 - Mapa da transmissão de dados com 806,91m.	55

LISTA DE TABELAS

Tabela 1 - Revisão bibliométrica do trabalho.....	18
Tabela 2 - Sensibilidade do receptor LoRa Semtech SX1276 em diferentes BW e SF.	28
Tabela 3 - Especificações técnicas do <i>Dragino LoRa Shield</i>	36
Tabela 4 - Seleção da função <i>Multispeed</i> via entradas digitais.	39
Tabela 5 - Referência <i>Multispeed</i>	39
Tabela 6 - Custos estimado para elaboração do projeto	56

LISTA DE ABREVEATURAS E SIGLAS

BW – *Bandwidth*

Chirp – *Compressed High Intensity Radar Pulse*

CPS – *Sistemas Ciber Físicos*

CPU – *Unidade Central de Processamento*

CR – *Rate Code*

CSS – *Spread Spectrum Chirp*

f – *Frequência*

GSM – *Global System Mobile*

IHM – *Interface Homem Máquina*

IoT – *Internet of Things*

LAN – *Local Area Network*

LoRa – *Long Range*

LPWAN – *Low Power Wide Area Network*

LTE – *Long Term Evolution*

MAC – *Controle de Acesso à Mídia*

MIT – *Motor de Indução Trifásico*

n – *Velocidade do Rotor*

n_s – *Velocidade Síncrona*

NB-IoT – *Narrowband Internet of Things*

NFC – *Near Field Communication*

p – *Número de polos*

PAN – *Personal Area Network*

PER – *Packet Error Rate*

PWM – *Modulação por Largura de Pulso*

RFID – Radio Frequency IDentification

RPMA – Random Phase Multiple Access

RSSF – Redes de Sensores Sem Fio

SF – Spreading Factor

SIG – Special Interest Group

SS – Spread Spectrum

UNB – Ultra Narrow Band

USB – Universal Serial Bus

WPAN – Wireless Personal Area Network

SUMÁRIO

1 INTRODUÇÃO	15
1.1 Considerações gerais	15
1.2 Justificativa	17
1.3 Objetivos geral e específicos	18
1.4 Estrutura do trabalho	19
2 REVISÃO BIBLIOGRÁFICA	20
2.1 Internet das Coisas (IoT)	20
2.2 Redes <i>Low Power Wide Area Network</i>	21
2.2.1 Redes Licenciadas	22
2.2.1.1 LTE	22
2.2.1.2 EC-GSM	22
2.2.1.3 NB-IoT	22
2.2.2 Redes Não Licenciadas	23
2.2.2.1 IEEE 802.11ah	23
2.2.2.2 Weightless	23
2.2.2.3 Ingenu	23
2.2.2.4 SigFox	24
2.2.2.5 LoRa	24
2.3 LoRaWAN	25
2.3.1 Camada Física LoRa	27
2.4 Motor de Indução Trifásico	28
3 METODOLOGIA	32
3.1 O <i>software</i> Processing	32
3.2 Arduino	33
3.3 Dragino LoRa <i>Shield</i>	35

3.4 Inversor de Frequência	36
3.4.1 Unidade Central de Processamento (CPU)	36
3.4.2 Interface Homem/Máquina (IHM)	36
3.4.3 Circuito de Potência	37
3.5 Proposta	40
4.1 Monitoramento	44
4.2 Comunicação sem fio com <i>Dragino LoRa Shield</i>	47
4.3 Controle de velocidade	50
4.4 Distância	53
4.5 Custos	56
4.6 Produção Bibliográfica	56
5 CONSIDERAÇÕES FINAIS	57
5.1 Sugestão para trabalhos futuros	58
REFERÊNCIAS BIBLIOGRÁFICAS	59
ANEXO II	74
ANEXO III	78

1 INTRODUÇÃO

1.1 Considerações gerais

A quarta revolução industrial iniciou-se a partir dos anos 2000 e caracteriza-se pelo conjunto de tecnologias que permitem a fusão do mundo físico, digital e biológico, sendo conhecida como a Indústria 4.0. As principais tecnologias da Indústria 4.0 são a Manufatura Aditiva, a Inteligência Artificial, a Internet das Coisas (IoT), a Biologia Sintética e os Sistemas Ciber Físicos (CPS). A tecnologia IoT possibilita a conexão de objetos físicos à Internet para realizar de forma coordenada uma determinada ação, por meio de dispositivos eletrônicos embarcados que admitem a coleta e a troca de dados, ou seja, informações (ABDI, 2018).

Para realizar a conexão de objetos físicos a outros objetos, com troca de dados ou coordenação de uma ação, tem-se a comunicação sem fio como método, a qual oferece à indústria grande flexibilidade, permitindo a comunicação de diferentes componentes, desde que atrelados a uma mesma rede (SANTOS; VOLANTE, 2018). A conectividade dos sistemas por meio da comunicação sem fio tem crescido, segundo Santos (2018), dados de pesquisas realizadas em 2017 mostram que a conectividade sem fio dentro das indústrias já corresponde a um quarto de toda conectividade.

Para o uso deste tipo de comunicação é necessário cumprir requisitos quanto à confiabilidade e à segurança (ROHDE & SCHWARZ, 2018). Portanto, ao passar dos anos foram criados vários protocolos de comunicação para atender aos requisitos e necessidades dos processos industriais. Estes protocolos encontram-se dentro das redes LAN (do Inglês *Local Area Network* – rede de área local), PAN (do Inglês *Personal Network Area* – rede de área pessoal) e LPWAN (do Inglês *Low Power Wide Area Networks* – redes de grande alcance e baixa potência) (OLIVEIRA, 2016).

Os protocolos contidos na rede LPWAN são geralmente aplicados em Redes de Sensores Sem Fio (RSSF), que são sistemas embarcados, com unidade de processamento, comunicação sem fio e sensores. Estes sistemas requerem equipamentos que gerem pouco tráfego de dados e com baixo consumo de energia (MOTTOLA; PICCO, 2011).

Dentre os diversos de equipamentos que são utilizados nas indústrias, os motores elétricos possuem destaque. Conforme Ben (2018), o setor industrial representou um consumo

de 31,8% de toda a energia elétrica consumida no Brasil durante o ano de 2017, dos quais 49% é realizado por motores elétricos.

Os motores elétricos são capazes de converter energia elétrica em energia mecânica (movimento) e para sua operação existem sistema de acionamento que mantem o processo de conversão sob controle. Tais sistemas são utilizados quando é necessário acionar uma carga ou equipamento, cujo movimento deve ser controlado, por exemplo a velocidade de rotação de uma bomba. Os acionamentos elétricos modernos são formados por três elementos principais: motor elétrico, dispositivo eletrônico (inversor de frequência ou *soft-starter*) e transmissão mecânica (WEG, 2005).

O motor de indução trifásico (MIT) rotor gaiola é o mais utilizado na indústria, isso se deve a fatores como custo relativo baixo, robustez, confiabilidade, eficiência e capacidade de adaptar o motor aos mais variados tipos de cargas (MELLO, 20013). Tal motor opera com velocidade constante, tendo uma queda de velocidade de poucos por cento quando passa da condição de ausência de carga para plena carga, (FITZGERALD, 2006).

O uso do inversor de frequência é uma das técnicas para controle de velocidade mais utilizadas na indústria, devido à possibilidade de controlar a velocidade dentro de toda a faixa operacional do motor, além de garantir proteção elétrica e mecânica, evitando desgastes mecânicos e garantindo que o motor opere dentro da sua capacidade operacional. O inversor pode ser operado localmente, através de sua IHM (Interface Homem Máquina), ou remotamente, por meio das suas entradas digitais e analógicas.

Para o controle da velocidade do motor de indução remotamente, por meio das entradas digitais do circuito inversor, pode ser utilizada a tecnologia dos sistemas embarcados. Uma configuração possível é o emprego da plataforma microcontrolada Arduino para processar as informações recebidas de um *software* de monitoramento, que possibilita a seleção de diferentes velocidades para o MIT. Além de conter dispositivos para transmissão e recepção dos dados. Assim, o controle do processo pode ser realizado a partir de uma sala de controle localizada longe dos atuadores.

Neste âmbito, o presente trabalho de conclusão de curso propõe o uso de uma rede de comunicação sem fio LPWAN para o controle remoto da velocidade de um motor de indução acionado por meio de um inversor de frequência, utilizando sistema embarcado e módulos de comunicação sem fio.

1.2 Justificativa

Em indústrias de grande, médio ou pequeno porte são inseridos avanços tecnológicos em busca do aumento da produtividade. Tais avanços podem ocorrer de diversas formas, como na mudança de um processo de produção, troca de equipamentos obsoletos, inserção de tecnologias nestes equipamentos ou aquisição de equipamentos com tecnologia avançada, para que, assim, o sistema produtivo seja otimizado.

Atualmente, os sistemas embarcados são cada vez mais empregados nas indústrias para automação de máquinas, assim como também no envio e processamento de informações oriundas do processo de produção para um centro de gerenciamento. Dessa forma, aumentam a eficiência na alocação de recursos, fornecem resposta rápida à demanda do mercado e reduzem o custo do trabalho e da logística (WANG; HE; XU, 2017).

Além disso, a comunicação sem fio nas indústrias abre novos recursos para soluções flexíveis e eficazes de automação. Possibilita a substituição de cabos e fibras ópticas, os quais demandam espaço para alocação, possuem alto custo de implantação e manutenção. As redes sem fio possibilitam realocação e posicionamento, normalmente são portáteis, a sua manutenção está mais ligada à troca das baterias, além de romper a barreira física dos cabos quando se trata de comunicação de longa distância (MÜLLER, 2016). Tal aplicação possui destaque em ambientes com riscos de explosão, tóxicos, de difícil acesso e distantes (PILON, 2009).

No decorrer do desenvolvimento deste trabalho foi concretizada a revisão bibliométrica, para analisar quantitativamente os trabalhos já desenvolvidos e publicados que abordam as tecnologias utilizadas para o desenvolvimento do protótipo em escala reduzida. A revisão foi realizada por meio do Google Acadêmico, com pesquisas de monografia, dissertações, teses e artigos, tanto em português quanto em inglês (GOOGLE ACADÊMICO, 2019). Primeiramente a busca foi realizada com temas mais abrangentes e em seguida foram incluídos os temas mais específicos das tecnologias empregadas neste trabalho. Os resultados são exibidos na Tabela 1.

Mediante a revisão bibliométrica realizada, foi observado que não existem trabalhos desenvolvidos com o emprego de sistemas embarcados, juntamente com a tecnologia LoRaWAN e controle de velocidade de um MIT para, por fim, obter um protótipo que compõe o controle de velocidade de MIT por meio de comunicação sem fio.

Tabela 1 - Revisão bibliométrica do trabalho

Temas de estudo	Resultados em Português	Resultados em Inglês
Sistemas Embarcados	6900	54400
LoRaWAN	121	6080
MIT	106000	8790000
Inversor de Frequência	4530	19300
Controle de Velocidade	6200	824000
Sistemas Embarcados e LoRaWAN	30	602
Sistemas Embarcados e MIT	933	47100
Sistemas Embarcados e Inversor de Frequência	111	37
Sistemas Embarcados, LoRaWAN e MIT	0	0
Sistemas Embarcados, LoRaWAN, MIT e Inversor de Frequência	0	0
Sistemas Embarcados, LoRaWAN, MIT e Controle de Velocidade	0	0

Fonte: Acervo do Autor.

1.3 Objetivos geral e específicos

O objetivo geral desta monografia é o desenvolvimento de um protótipo em escala reduzida de um sistema embarcado, para controle da partida e variação da velocidade de um MIT. Neste sistema, o acionamento do motor é realizado pelo inversor de frequência, o qual receberá o sinal de controle da plataforma Arduino, com as trocas de informações realizadas por meio de comunicação sem fio, possibilitadas pelo uso da tecnologia LoRaWAN.

Para consolidar o objetivo geral, alguns objetivos específicos devem ser alcançados:

- Realizar pesquisa bibliográfica para conhecer a tecnologia LoRaWAN, a qual faz parte das redes LPWAN;
- Realizar pesquisa bibliográfica sobre controle de velocidade de um MIT a partir de um inversor de frequência;
- Apresentar características técnicas da tecnologia LoRaWAN;
- Efetuar troca de informações entre *Network server* e *end-device*;
- Desenvolver um *software* de monitoramento para a comunicação entre “Homem e Máquina”;

- Efetuar testes de comunicação Arduino e inversor de frequência;
- Efetuar testes de partidas e controle de velocidade de motores a partir de dados da comunicação sem fio;
- Realizar o levantamento de custos estimados para a realização do protótipo.

1.4 Estrutura do trabalho

Este trabalho é composto por cinco capítulos. O Capítulo 2 inclui revisão bibliográfica sobre MITs, controle de velocidade de MIT com inversor de frequência, redes LPWAN e características da tecnologia de comunicação a ser utilizada no protótipo, LoRaWAN. O Capítulo 3 apresenta a metodologia detalhada utilizada para o desenvolvimento do protótipo. O Capítulo 4 apresenta os resultados obtidos e são realizadas discussões pertinentes. Por fim, o Capítulo 5 apresenta as conclusões do trabalho e sugestões para futuros trabalhos.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo são apresentados os estudos referentes ao tema abordado, de modo a identificar as principais teorias aplicadas na execução do protótipo. Estão demonstradas como são classificadas as Redes LPWAN, a arquitetura LoRaWAN, as características de um motor de indução trifásico e as possíveis formas de realizar o seu controle de velocidade.

2.1 Internet das Coisas (IoT)

Comumente as diferentes definições do termo “Internet das Coisas” (IoT) citam a relação desta como a integração entre o mundo físico e o mundo virtual através da Internet. Esta conexão à Internet pode-se referir às redes de sensores, automóveis, computadores e máquinas, dentre outras.

Para monitorar e interagir com um ou mais sistemas e fazer a conexão com a Internet são requeridos dispositivos de comunicação. Os dispositivos podem ser conectados ou embutidos nos próprios sistemas ou podem ser instalados no ambiente das “coisas” a serem monitoradas. Exemplos típicos de dispositivos incluem leitores de RFID (do Inglês *Radio Frequency IDentification*), sensores e atuadores, computadores embarcados, bem como telefones celulares (HALLER, 2010).

Na indústria, os dispositivos - como as várias classes de sensores, com ou sem capacidades de atuação - são usados para o monitoramento e provimento de dados de modo que um processo possa ser controlado remotamente. Isso pode ser tão simples como a utilização de vários sensores para posicionar as caixas de vários tamanhos em uma correia transportadora, para que uma máquina de rotulagem possa aplicar corretamente as etiquetas a elas. Essa tarefa pode ser realizada *on-line*, enviando os dados para um computador remoto, analisando-os e atuando no processo de forma automática, sem qualquer intervenção humana (KARIMI; ATKINSON, 2014).

Os principais obstáculos que impedem o desenvolvimento do conceito de Internet das Coisas são:

- **Conectividade:** consiste em assegurar a interconectividade entre os dispositivos construídos por diferentes fabricantes. A utilização de um único protocolo de comunicação não é uma

opção viável. Portanto, o principal objetivo é garantir a interconectividade dos vários formatos de dados em uma escala mundial. Igualmente vital é a análise dos padrões utilizados hoje em dia no conceito de Internet das Coisas (LAVRIC; POPA, 2017);

- **Gestão eficiente da energia:** a maioria dos módulos é alimentada por bateria, logo existe uma necessidade constante para reduzir o consumo de energia. Como o custo destes dispositivos é baixo, a substituição da bateria pode não ser considerada eficiente, já que esta operação seria mais cara do que a substituição de todo o módulo. Portanto, o ciclo de vida útil da bateria deverá ser de anos (LAVRIC; POPA, 2017);
- **Segurança:** o fornecimento de altos níveis de segurança deve ser considerado. Assim, é altamente necessário desenvolver novos protocolos de autenticação e criptografia (LAVRIC; POPA, 2017).

Para serem controlados remotamente, devem ser utilizados dispositivos de comunicação, para realizar a interação entre os sistemas. Estas redes são construídas por cabos de cobre, fibras ópticas ou sem fio (CASSIOLATO, 2019). Porém a comunicação para IoT emprega alguns protocolos padrões sem fio como o NFC (do Inglês *Near Field Communication*), RFID, ZigBee, Bluetooth, Wi-Fi e as LPWANs, sendo estas últimas referentes a uma nova tecnologia para comunicação de longo alcance (SANT'ANA, 2017).

2.2 Redes *Low Power Wide Area Network*

A comunicação sem fio entre objetos, sistemas de controle ou outras aplicações necessitam de redes que consigam atender suas devidas necessidades. No mercado de IoT havia somente tecnologias de redes sem fio de curto alcance para WPAN (do Inglês *Wireless Personal Area Network*) como ZigBee e *Bluetooth*, as WLANs (do Inglês *Wireless Local Area Network*) e até o uso de redes como LTE (do Inglês *Long Term Evolution*) e GSM (do Inglês *Global System for Mobile*). Outras tecnologias surgiram visando um baixo consumo de energia e longo alcance, porém com baixa taxa de dados, denominadas de Redes de Grande Alcance e Baixa Potência (do Inglês *Low Power Wide Area Networks – LPWAN*) (OLIVEIRA, 2016).

O uso da tecnologia LPWAN permite implantar uma única estação base e cobrir uma instalação completa, podendo ser um hospital, *campus* universitário, estacionamentos ou na indústria, que pode incluir vários processos, por exemplo medição de consumo de energia

elétrica, o monitoramento de gás e água, ou ficar encarregado de uma rede de sensores de um determinado processo (PETÄJÄJÄRVI et al., 2016).

As redes LPWAN são classificadas em duas categorias, sendo a categoria com o espectro de frequência licenciado e a de espectro não licenciado. Para cada país existe uma regulamentação que realiza a homologação de quais faixas são licenciadas ou não e para quais aplicações podem ser utilizadas.

2.2.1 Redes Licenciadas

As redes licenciadas tem um custo maior em comparação às não licenciadas, devido à necessidade de pagamento para transmitir nas larguras de banda reservadas para tais transmissões. Além disso, os dispositivos podem necessitar da utilização de *chip* (cartão SIM), que consequentemente aumenta o custo dos dispositivos (SOUZA, 2018). Estas redes são a LTE-M, EC-GSM e a NB-IoT.

2.2.1.1 LTE

A rede LTE é uma rede de comunicação móvel que surge como uma evolução das redes 2G e 3G e foi alterada incluindo restrições para reduzir os custos e o consumo de energia, assim obteve-se a versão LTE-M que opera na faixa de 1,4 MHz. Esta tecnologia é fundamentada na rede de celular 4G, porém com um alcance maior devido à tecnologia GSM também empregada (SOUZA, 2018).

2.2.1.2 EC-GSM

EC-GSM é uma tecnologia LPWAN em desenvolvimento pela 3GPP, que usa o mesmo projeto GSM, porém com mudanças que são necessárias a fim de reforçar as condições relacionadas com a LPWAN, ou seja, de alta capacidade, de longo alcance e baixo consumo de energia. Também conhecida como EC-GSM-IoT, usa 200 KHz de largura de banda por canal, para uma largura de banda total do sistema de 2,4 MHz. A taxa de transferência do dispositivo EC-GSM varia de 350bps para 70Kbps, dependendo da classe de cobertura atualmente em uso (FINNEGAN; BROWN, 2018).

2.2.1.3 NB-IoT

Narrowband Internet of Things (NB-IoT) é uma das soluções que juntamente com a EC-GSM, formadas pela 3GPP. É construída a partir de LTE com simplificações de processo, como redução de largura de banda para 180 KHz (em comparação a 1,4 MHz da LTE) e sua taxa de dados chega a 60Kbps. A melhora na cobertura e redução do consumo de energia é conseguida

em troca de latência, uma taxa de dados mais baixa. O preço do *chip* também é reduzido, por meio da utilização de uma faixa de banda mais estreita (FINNEGAN; BROWN, 2018).

2.2.2 Redes Não Licenciadas

Os tipos de redes LPWAN que operam nas faixas de banda não licenciadas são as IEEE 802.11ah, Weightless, Ingenu, SigFox e LoRa.

2.2.2.1 IEEE 802.11ah

O padrão 802.11ah permite a comunicação em distâncias de até 1000 m e utiliza faixas isentas de licença, abaixo de 1 GHz, para fornecer um alcance estendido às redes Wi-Fi. Esta rede permite uma transmissão com taxas de dados a partir de 160Kbps e até 346Mbps, que dependem do modo de operação. Ela é baseada em camadas PHY e MAC do modelo 802.11ac (CROTTI, 2017).

2.2.2.2 Weightless

Weightless representa um conjunto de tecnologias LPWAN definidas e geridas pelo Weightless – SIG (*Special Interest Group*). Este conjunto contém três padrões diferentes: Weightless – N, que é focado no custo baixo, que é análogo a SigFox e com alcance de quilômetros, porém uma taxa pequena de 100bps; Weightless – W, que ocupa parte do espectro em branco utilizado para TV e tem um alcance de até 5km e uma taxa de dados de 1Kbps a 10Mbps; e Weightless – P, que se concentra na alta performance, com taxas de 200bps a 100kbps, e transmite no espectro de 12,5KHz, podendo suportar abrangências de 2km em centros urbanos (FINNEGAN; BROWN, 2018).

2.2.2.3 Ingenu

Ingenu é uma solução para redes LPWAN de espalhamento espectral baseada na tecnologia de Acesso Múltiplo Aleatório por Fase (do Inglês *Random Phase Multiple Access – RPMA*) que opera na banda ISM de 2,4 GHz. A cobertura alcançada por essa tecnologia deve-se principalmente ao aumento da potência de transmissão disponível nesta banda, que pode atingir um alcance de 4km. Uma implantação de O RPMA utiliza 1MHz da banda de 80MHz, a qual permite várias implantações simultâneas ou, alternativamente, o uso de múltiplos canais para suportar uma rede. Além disso, não há regulamentos a respeito do *duty cycle* (ciclo de trabalho) a serem seguidas na Europa na banda de 2,4 GHz. Sua taxa de dados máxima gira em torno de 8Kbps (FINNEGAN; BROWN, 2018).

2.2.2.4 SigFox

A tecnologia da Sigfox opera nas sub bandas de 1GHz e utiliza uma técnica de modulação específica, que se refere a banda ultra estreita (do Inglês *Ultra Narrow Band* – UNB), a qual deve atender aos regulamentos a respeito do *duty cycle*. Ela se baseia no protocolo MAC ALOHA (NOLAN; GUIBENE; KELLY, 2016). O sistema é uma abordagem baseada em nuvem, pois os usuários necessitam adquirir apenas os *end-devices* (dispositivos finais), onde todos os dados recebidos pelos *gateways* são enviados para um servidor *back-end*, o qual pode ser acessado pelo cliente através de um portal web. Os clientes podem, então, implementar retornos de chamada para terem seus dados retransmitidos para o seu próprio sistema. A estação base Sigfox pode cobrir uma gama de 20-50km em áreas rurais, e 3-10km em áreas urbanas. Esta tecnologia utiliza um modelo que limita a quantidade de mensagens a serem enviadas por dia, que é comprada por cada usuário (NOLAN; GUIBENE; KELLY, 2016).

2.2.2.5 LoRa

LoRa (*Long Range*) é uma técnica de modulação que permite a comunicação de longa distância, de baixa potência e baixo rendimento. Opera-se nas bandas 433, 868 e 915MHz ISM, dependendo da região de implementação. No Brasil a faixa de banda livre de licenciamento é de 915MHz segundo ANATEL (2017). A modulação LoRa foi patenteada pela Semtech Corporation. LoRa é um tipo de modulação *Spread Spectrum* (SS) (LAVRIC; POPA, 2017).

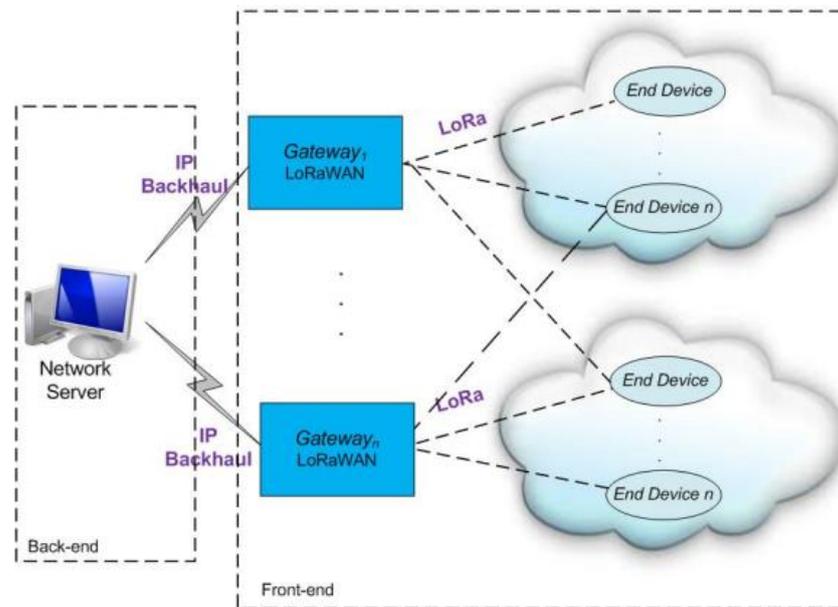
A rede LoRa é implementada usando a topologia de rede em estrela. Sua arquitetura pode ser separada em *back-end* e uma parte de *front-end*. A parte *back-end* consiste no servidor de rede que armazena as informações recebidas dos sensores. A *front-end* consiste de módulos de *gateway* e os nós do dispositivo final. Os módulos de *gateway* atuam como uma ponte entre os nós do dispositivo final e o servidor de rede. As informações entre o servidor de rede e os módulos de *gateway* são enviadas através da conexão IP. A Figura 1 apresenta a arquitetura LoRa (LAVRIC; POPA, 2017).

Os módulos do dispositivo final não têm capacidade de roteamento e as mensagens são enviadas diretamente para o concentrador de *gateway* através de um mecanismo de transmissão. As taxas de transferência da tecnologia LoRa variam entre 0,3Kbps e 50Kbps (LAVRIC; POPA, 2017).

De acordo com Lavric e Popa (2017) e Nolan, Guibene e Kelly (2016), a tecnologia LoRa é promissora e para aplicações que exigem baixas taxas de transmissão é uma opção

adequada. Além disso, segundo Al-Kashoash e Kemp (2017), seu nível de segurança é maior que outras tecnologias, comparada à SigFox, pois possui mecanismos de criptografia. Portanto, este trabalho será baseado na tecnologia LoRa. Na seção 2.3 a seguir serão apresentados os principais aspectos desta tecnologia.

Figura 1 - Arquitetura LoRa



Fonte: LAVRIC e POPA (2017).

2.3 LoRaWAN

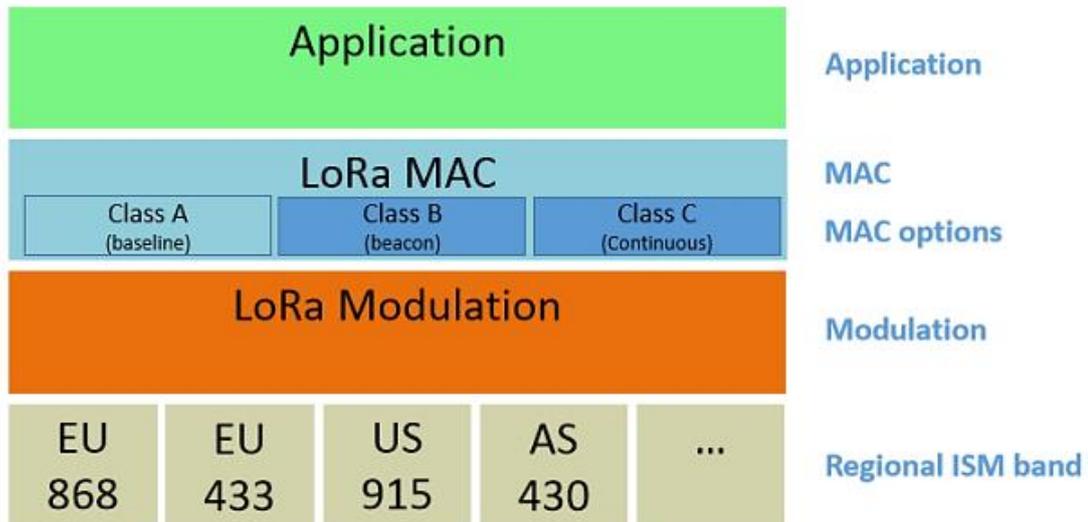
LoRaWAN é um protocolo de comunicação de longo alcance e pertence às Redes *Low Power Wide Area* (LPWAN), que utilizam comunicação sem fio de rede aérea e que são projetadas para cobrir grandes áreas e minimizar o consumo de energia, porém oferecem uma taxa de bits relativamente baixa (LAVRIC; POPA, 2017).

O protocolo LoRaWAN é gerido pela Alliance LoRa, sendo um modelo livre de taxa de inscrição, simplificando e barateando a gestão da infraestrutura com uma implementação simples (NOREEN; KEMP; CLAVIER, 2017).

Os dispositivos LoRa não utilizam endereço IP, sendo que a comunicação entre eles e os *gateways* ocorre pelo protocolo LoRaWAN e a segurança se dá por chaves criptografadas AES (do Inglês *Advanced Encryption Standard*), utilizando um identificador IEEE EU164. A

arquitetura LoRaWAN pode se referir a duas camadas distintas, conforme a Figura 2: (i) um protocolo de camada MAC (LoRaWAN), embora o sistema de comunicações LoRa também implica uma arquitetura de rede de acesso específico; e (ii) uma camada física que utiliza uma técnica de modulação por rádio, *Spread Spectrum Chirp* (CSS) (NOREEN; KEMP; CLAVIER, 2017).

Figura 2 - Arquitetura LoRaWAN.



Fonte: ALLIANCE (2017).

Esta rede contém vários componentes que são definidos na especificação e necessários para formar a rede LoRaWAN, que são os *end-device*, *gateways*, e *network server*.

- *End-device*: são os sensores ou atuadores que comunicam com os *gateways* operados por LoRa.
- *Gateway*: são os dispositivos intermediários, responsáveis por encaminhar pacotes provenientes dos *end-devices* para o *network server* através de uma interface de *backhaul*.
- *Network server*: responsável por duplicar e decodificar os pacotes enviados pelos dispositivos e gerar os pacotes que devem ser enviados de volta aos mesmos.

O LoRaWAN possui três classes diferentes de dispositivos finais para atender às diversas necessidades de aplicativos (NOREEN; KEMP; CLAVIER, 2017).

- Classe A: dispositivos de classe A têm capacidade de comunicação bidirecional. O espaço de tempo durante o qual o dispositivo está transmitindo é seguido por dois intervalos de tempo curto durante o qual o dispositivo pode receber informações. Dispositivos incluídos nesta categoria podem receber dados a partir do servidor somente depois que tenham

enviado as informações. Assim, os dispositivos de classe A asseguram a mais alta eficiência de energia;

- Classe B: dispositivos da Classe B têm capacidade de comunicação bidirecional e um *slot* de tempo adicional que lhes permite receber dados. Dispositivos finais Classe B abrem recebimento de janelas extras em horários programados;
- Classe C: dispositivos classe C têm capacidade de comunicação bidirecional, não simultâneas e intervalos de tempo durante o qual eles podem receber informações ilimitadas. O único momento em que um dispositivo de classe C não pode receber informações é quando ele envia a informação.

2.3.1 Camada Física LoRa

LoRa é uma modulação de CSS, que utiliza uma frequência *Chirps* (do Inglês (*Compressed High Intensity Radar Pulse* - Pulso de Radar de Alta Intensidade Comprimido) com uma variação linear da frequência que varre toda a largura de banda ao longo do tempo, de modo a codificar a informação. Devido à linearidade dos impulsos *Chirp*, desvios de frequência entre o receptor e o transmissor são equivalentes à temporização dos deslocamentos, facilmente eliminados no decodificador. Isto também faz com que esta modulação seja imune ao efeito *Doppler*, equivalente a um desvio de frequência. O desvio de frequência entre o transmissor e o receptor pode chegar a 20% da largura de banda sem impactar no desempenho da decodificação. Isso ajuda a reduzir o preço de transmissores LoRa (NOREEN; KEMP; CLAVIER, 2017).

Vários parâmetros estão disponíveis para a personalização da modulação LoRa: Largura de Banda (do Inglês *Bandwidth* – BW), Fator de Propagação (do Inglês *Spreading Factor* – SF) e Taxa de Código (do Inglês *Rate Code* – CR).

A largura de banda é o parâmetro mais importante da modulação LoRa. Um aumento de largura de banda reduz a sensibilidade do receptor, enquanto que o aumento do SF aumenta a sensibilidade do receptor. Diminuir o CR ajuda a reduzir a taxa de erro de pacote (do Inglês *Packet Error Rate* – PER), na presença de rajadas curtas de interferência, isto é, um pacote transmitido com um código de taxa 4/8 vai ser mais tolerante à interferência do que um sinal transmitido com um código de taxa 4/5. A Tabela 2, feita a partir da folha de dados SX1276, informa a sensibilidade do receptor para diferentes larguras de banda (NOREEN; KEMP; CLAVIER, 2017).

Tabela 2 - Sensibilidade do receptor LoRa Semtech SX1276 em diferentes BW e SF.

BW	SF					
	7	8	9	10	11	12
125KHz	-123	-126	-129	-132	-133	-136
250KHz	-120	-123	-125	-128	-130	-133
500KHz	-116	-119	-122	-125	-128	-130

Fonte: NOREEN; KEMP; CLAVIER (2017).

O formato de mensagem LoRa começa com um preâmbulo de oito *Chirps*. Este preâmbulo é seguido por um início do delimitador de quadros (SFD). O SFD também codifica a chamada “sincronização palavra” usada para distinguir entre público / privado a Rede LoRaWAN. A Semtech não divulga os detalhes do PHY *header*. Por fim, o CRC protege a integridade do *payload* PHY, pois é aplicado para checar a integridade do pacote, ou seja, usado para *uplinks*, mas normalmente não está presente nos quadros de *downlink*. O formato desta mensagem é demonstrado pela a Figura 3 (RIZZI et al., 2017).

Figura 3 - Formato de mensagem LoRaWAN



Fonte: RIZZI et al. (2017).

2.4 Motor de Indução Trifásico

Os motores de indução trifásicos são as máquinas rotativas mais empregadas atualmente nos acionamentos industriais. Tal fato se deve a sua aplicabilidade aos mais diversos tipos de cargas mecânicas, tais como: compressores, bombas, esteiras, transportadores, prensas, exaustores, entre outros. Além disso, suas características operacionais garantem confiabilidade, robustez, simplicidade e baixo custo. A forma construtiva de um motor de indução de forma geral é apresentada pela a Figura 4.

O motor de indução possui duas partes principais: estator e rotor. O estator (armadura) é formado por um enrolamento trifásico, como bobinas idênticas, localizadas nas ranhuras do núcleo ferromagnético e distribuídas ao longo da circunferência de entreferro. Os enrolamentos de armadura são posicionados de forma que seus eixos magnéticos sejam separados de 120°

geométricos entre si e são alimentados por tensões trifásicas equilibradas, com um defasamento de 120° elétricos (FITZGERALD, 2006).

Dessa forma, é criado um campo magnético que gira à velocidade síncrona, a qual é dada pela seguinte equação (KOSOW, 1982):

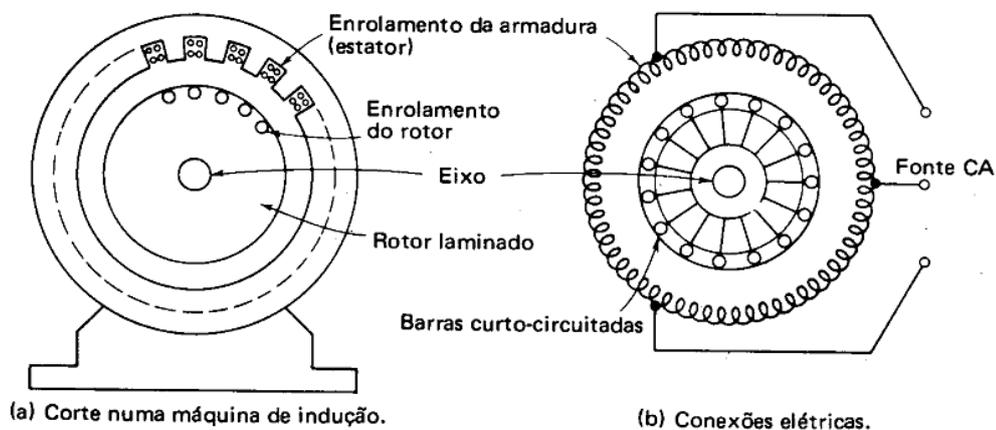
$$n_s = \frac{120 \times f}{p} \quad (1)$$

em que f é a frequência da fonte de alimentação, em hertz, e p é o número de polos magnéticos da máquina, parâmetro construtivo.

A forma construtiva da parte rotativa (rotor) destas máquinas os diferencia em dois tipos: rotor bobinado e rotor gaiola de esquilo (KOSOW, 1982). As máquinas de indução de rotor bobinado possuem enrolamento de rotor semelhante ao enrolamento de estator. Os terminais destes enrolamentos são conectados a anéis deslizantes e escovas de carvão apoiadas sobre estes, para que os terminais do enrolamento de rotor se tornem acessíveis externamente. Por isso, estas máquinas são menos robustas, mais caras e exigem maior manutenção.

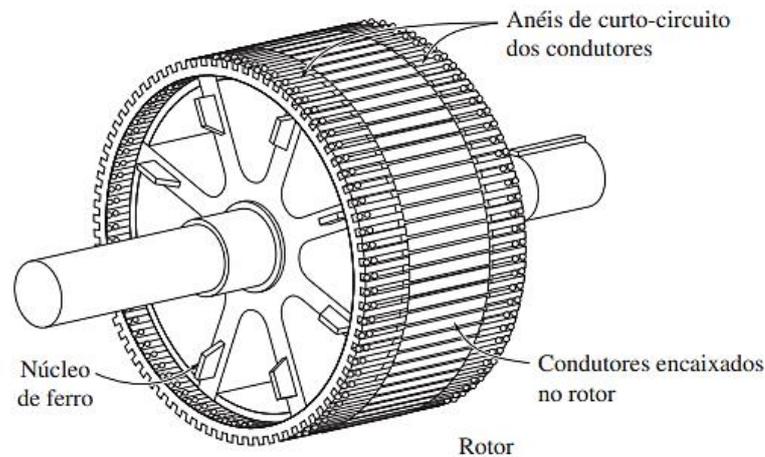
Os motores de indução do tipo gaiola de esquilo possuem o rotor formado por várias barras condutoras que são acomodadas dentro de ranhuras na superfície do rotor e curto-circuitadas nas extremidades por anéis (CHAPMAN, 2013). Este rotor é apresentado na Figura 5. Devido à simplicidade, robustez e conseqüente menor demanda por manutenção, o torna o tipo de motor mais utilizado (FITZGERALD, 2006).

Figura 4 - Construção geral de uma máquina de indução.



Fonte: KOSOW (1982).

Figura 5 - Desenho esquemático de um rotor gaiola de esquilo.



Fonte: CHAPMAN (2013).

O rotor do motor de indução gira a uma velocidade n e no mesmo sentido que o campo girante do estator (n_s). Porém, existe uma diferença entre essas velocidades, denominada “escorregamento” (s), apresentada nas equações (2) e (3) (FITZGERALD, 2006).

$$s = n_s - n \quad (2)$$

$$s = (n_s - n) \times 100 \quad (3)$$

Portanto, a velocidade do rotor do motor de indução trifásico em rpm é expressa pela equação (4):

$$n = (1 - s) \times n_s \quad (4)$$

Especificamente o motor de indução rotor gaiola de esquilo é um motor de velocidade constante, a qual depende da frequência da rede de alimentação e do número de polos magnéticos da máquina (parâmetro construtivo).

Alterando a velocidade síncrona do motor de indução pode-se alterar a velocidade do rotor e para isso existem algumas formas, como: mudança do número de polos, variação da frequência de linha e modificação do escorregamento pela inserção de resistência externa, no rotor, possível apenas no caso do motor rotor bobinado. A alteração do número de polos magnéticos é uma variação discreta e depende da forma construtiva do estator do motor, logo a alteração da frequência da tensão de alimentação do motor é a técnica mais flexível e aplicada.

O controle de velocidade com a variação da frequência na armadura se baseia na equação (1), que altera a frequência de linha f e consequentemente altera-se a velocidade do rotor, como visto nas equações (2) e (4). Para esse feito, são utilizados os inversores de frequência.

Os circuitos inversores controlam a frequência assim como a amplitude da tensão aplicada, pra que as condições físicas do MIT sejam mantidas. Com essa característica é possível manter o fluxo magnético e consequentemente o torque constante para a aplicação. Alguns inversores de frequência contam com padrões pré-estabelecidos, de curvas de conjugado de partida, dentre os quais pode ser escolhido aquele que melhor se adequa ao conjugado requerido pela carga. Estes padrões são divididos em uso geral, conjugado elevado de partida e baixo conjugado de partida (FITZGERALD, 2006).

3 METODOLOGIA

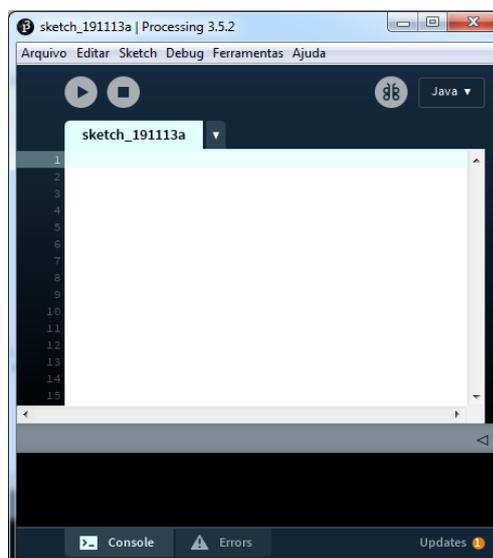
Neste capítulo é descrita a metodologia empregada para alcançar o objetivo proposto, que envolve as seguintes etapas: sistema de monitoramento, transmissão de dados, parametrização do inversor de frequência e controle da velocidade do MIT. Logo após, é apresentado um panorama geral do protótipo proposto.

3.1 O *software* Processing

O *software* Processing foi inicialmente criado para servir como um “caderno de desenho” de *software* e ensinar os fundamentos da programação em um contexto visual, portanto é uma plataforma *open-source*, a qual pode ser utilizada nos sistemas Mac, Windows e Linux (FONDATION, 2019).

A criação de protótipos de *softwares* ou visualizações de dados são as principais áreas exploradas pelo Processing. Empresas como Google e Intel utilizaram o Processing para desenvolver novas interfaces e serviços, enquanto General Eletric e Nokia aplicaram-no para visualizar seus dados internos. Tais *softwares* são desenvolvidos com as linguagens Java, Java Script e Python. Um exemplo da interface do ambiente de desenvolvimento do Processing é apresentado na Figura 6.

Figura 6 - Interface Processing 3.5.2



Fonte: Acervo do Autor.

3.2 Arduino

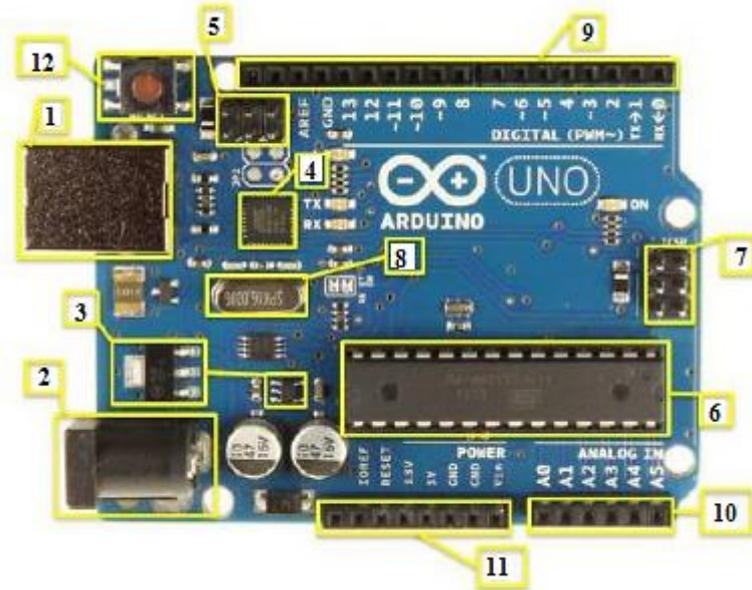
O Arduino é uma plataforma de prototipagem *open-source* que se destaca em meio as outras plataformas devido à facilidade de utilização. Utiliza um microcontrolador programável geralmente usado para construção e programação de eletrônicos. É possível realizar troca de informações entre dispositivos por vários tipos de comunicação, desde serial, I2C e até mesmo através da Internet.

O *hardware* do Arduino contém uma série de componentes que combinados permitem construir protótipos. Os principais componentes da plataforma são descritos a seguir e ilustrados pela a Figura 7 (ZACARONI, 2019):

1. Entrada USB que permite realizar o *upload* de um programa para o microcontrolador e ainda contém uma alimentação de 5V regulada para o Arduino;
2. Fonte de alimentação externa que contém uma tensão regulada de 7 a 12V;
3. Reguladores de tensão para 3,3V e 5V;
4. Microcontrolador ATMEL ATMEGA 16U2, que possibilita a comunicação via porta USB do Arduino com o computador;
5. Conector ICSP do ATMEGA16U2;
6. Microcontrolador ATMEL ATMEGAA328 que envia, recebe e comanda os sistemas de controle;
7. Conector para gravação ICSP do ATMEGAA328;
8. Cristal oscilador de 16MHz;
9. Pinos referentes às entradas/saídas digitais;
10. Pinos referentes às entradas/saídas analógicas;
11. Pinos de alimentação;
12. Reset: ao pressionar este botão, o Arduino redefine os parâmetros para a forma inicial, como havia sido carregada no microcontrolador.

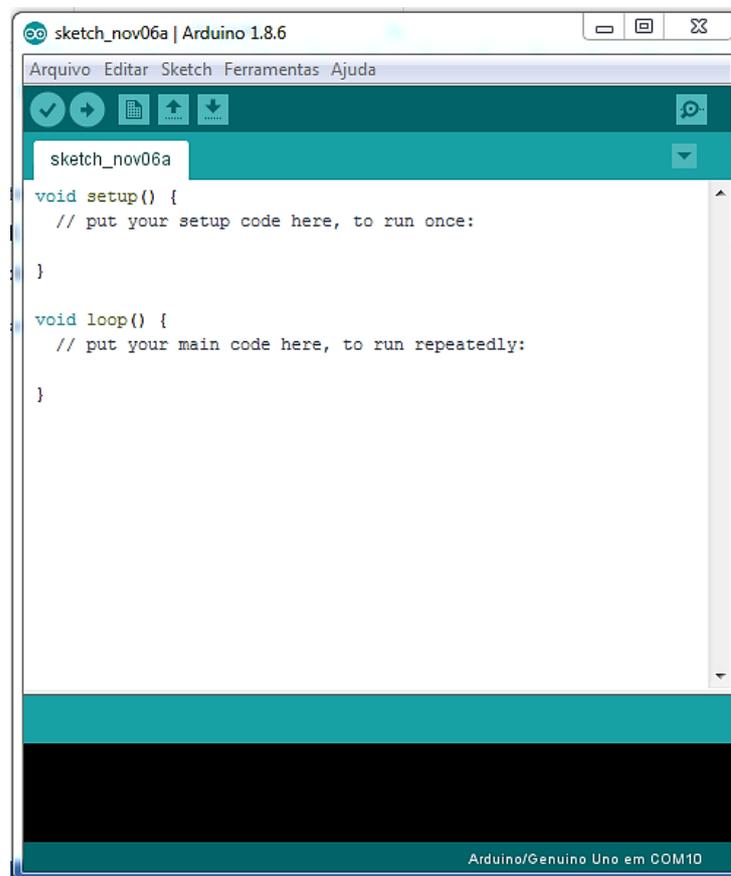
Esta placa pode ser programado por meio de computadores que possuam sistemas operacionais Windows, Linux ou Mac. Esta programação é realizada com o *software* IDE, o qual utiliza a linguagem C/C++ que formam os *sketches* a serem compilados para o Arduino (ARDUINO, 2017). A interface do *software* IDE é mostrada pela Figura 8.

Figura 7 - Placa de composição do Arduino UNO



Fonte: ZACARONI (2019).

Figura 8 - Interface Arduino IDE 1.8.6



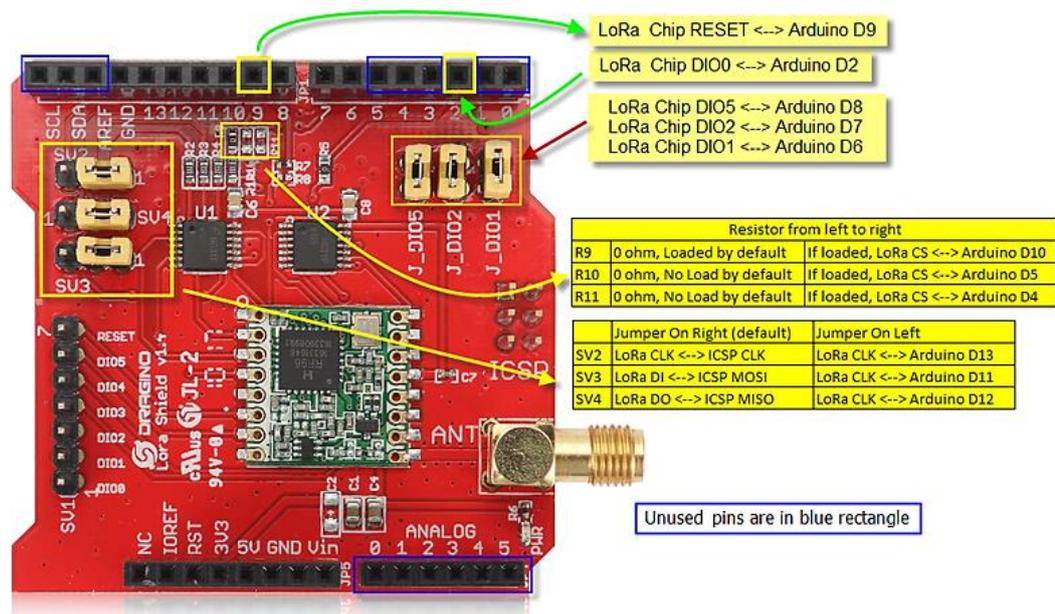
Fonte: Acervo do Autor.

3.3 Dragino LoRa Shield

O *Dragino LoRa Shield* é um transceptor de longo alcance que utiliza a técnica de modulação LoRa patenteada pela Semtech, o qual é utilizado para construir redes LPWAN e podem atingir uma sensibilidade superior a -148dBm com materiais de baixo custo. A alta sensibilidade combinada ao amplificador de potência integrado, de $+20\text{dBm}$, torna-o adequado para aplicações que exijam alcance (DRAGINO, 2017).

Baseados nos *chips* SX1276 / SX1278 da Semtech, são utilizados em aplicações que necessitam de comunicação sem fio, como uma rede de sensores sem fio, automação predial, cidades inteligentes, sistemas de irrigação, entre outros. O *Shield* é ilustrado pela a Figura 9, juntamente com o mapeamento de seus pinos (DRAGINO, 2017).

Figura 9 - Mapeamento de pinos Dragino LoRa Shield v1.4



Fonte: DRAGINO (2017).

Devido ao formato e às configurações do *Dragino LoRa Shield*, ele é compatível com algumas plataformas de prototipagem, que incluem o Arduino Leonardo, Uno, Mega e DUE. O *Shield* encontra-se na versão v1.4 e em comparação a versão v1.0, foi implementada a conexão das entradas digitais ao Arduino, alteração do conector *i-pex* para um SMA, a mudança da tensão de alimentação de 5V para 3,3V e inserção de resistores R9/R10/R11 para possibilitar alterações do pino LoRa CS para outras entradas. Algumas especificações técnicas são apresentadas na Tabela 3 (NERY, 2018).

Tabela 3 - Especificações técnicas do Dragino LoRa Shield.

Dragino LoRa Shield	
Tensão de alimentação I/O	3,3 ou 5V
Banda de Frequência	433MHZ/ 868MHZ/ 915MHZ
Sensibilidade máxima	-148dBm
Potência de saída	110mW
Modulação	FSK, GFSK, MSK, GMSK, LoRaTM e OOK
Alcance	1,5km

Fonte: Adaptado (NERY, 2018).

3.4 Inversor de Frequência

O uso do inversor de frequência nos acionamentos industriais vem da necessidade de controlar a velocidade do motor para adequá-la às necessidades da carga acionada, tais como bombas, ventiladores, esteiras transportadoras, sistemas de dosagem, entre outros. O inversor de frequência é o equipamento que possibilita realizar o controle da velocidade da máquina de indução por meio da modificação da frequência aplicada aos seus terminais, conforme mostra a eq. (1). A tensão e a frequência são modificadas de forma que mantenham o fluxo magnético constante na máquina e o controle destes parâmetros é realizado por técnicas de modulação de largura de pulso (PWM).

Um inversor de frequência é constituído basicamente por: Circuito de Controle, Unidade Central de Processamento (CPU), Interface Homem/Máquina (IHM) e Circuito de Potência. O diagrama de blocos de um inversor CFW11 da WEG é representado pela Figura 10 (WEG, 2019).

3.4.1 Unidade Central de Processamento (CPU)

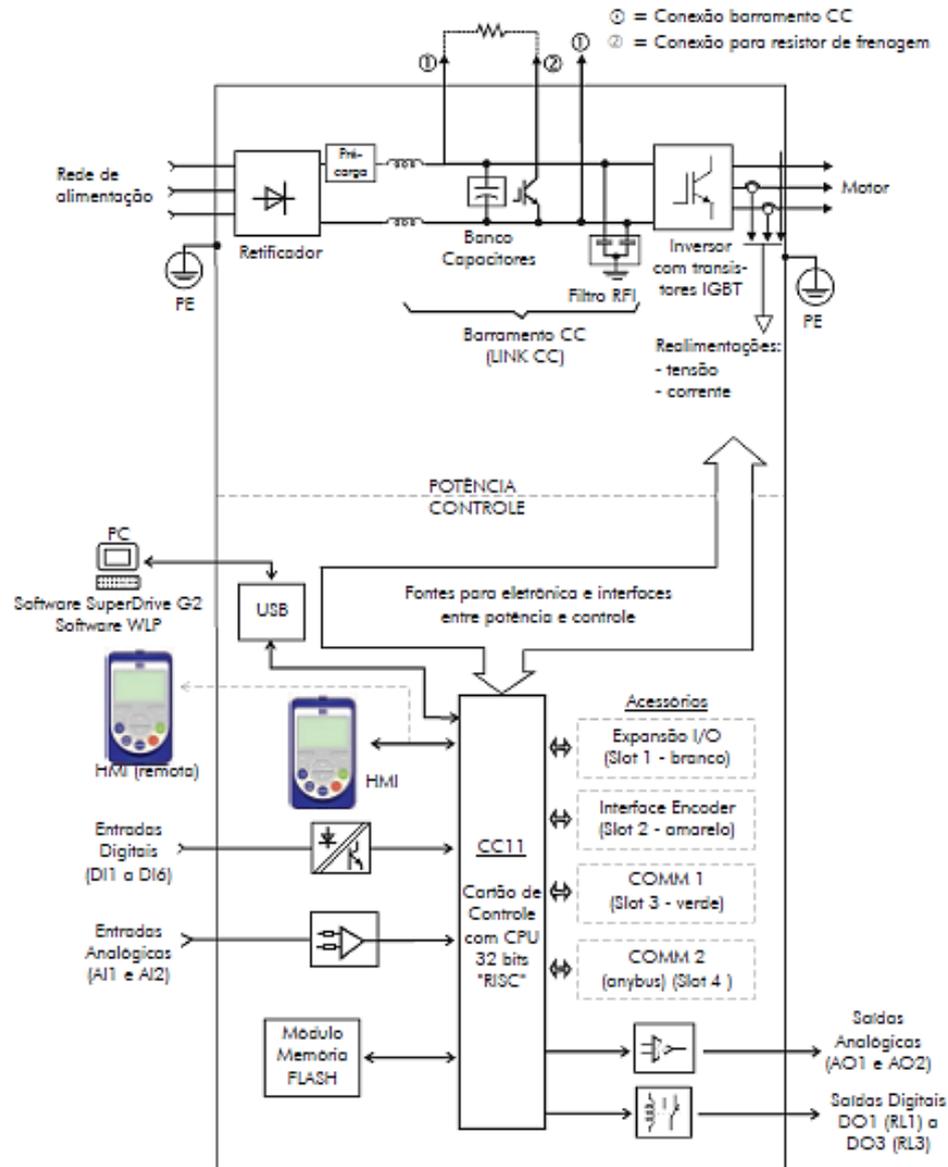
A CPU é o bloco responsável por armazenar todas as informações e parâmetros, por meio da sua memória integrada, e também executa a função de gerar pulsos de disparo, a partir da lógica de controle para os IGBTs (Transistores Bipolares de Portas Isoladas, do Inglês *Insulated Gate Bipolar Transistor*) (WEG, 2019).

3.4.2 Interface Homem/Máquina (IHM)

A parametrização e monitoramento do inversor são realizadas de forma local pela IHM. A parametrização consiste em inserir os parâmetros do motor a ser controlado, como valores

de tensão, corrente, potência e velocidade nominal, além da programação da função de controle que o inversor implementará. Com o monitoramento é possível observar os valores dos parâmetros de leitura (WEG, 2019).

Figura 10 - Blocos dos componentes básicos de um inversor de frequência



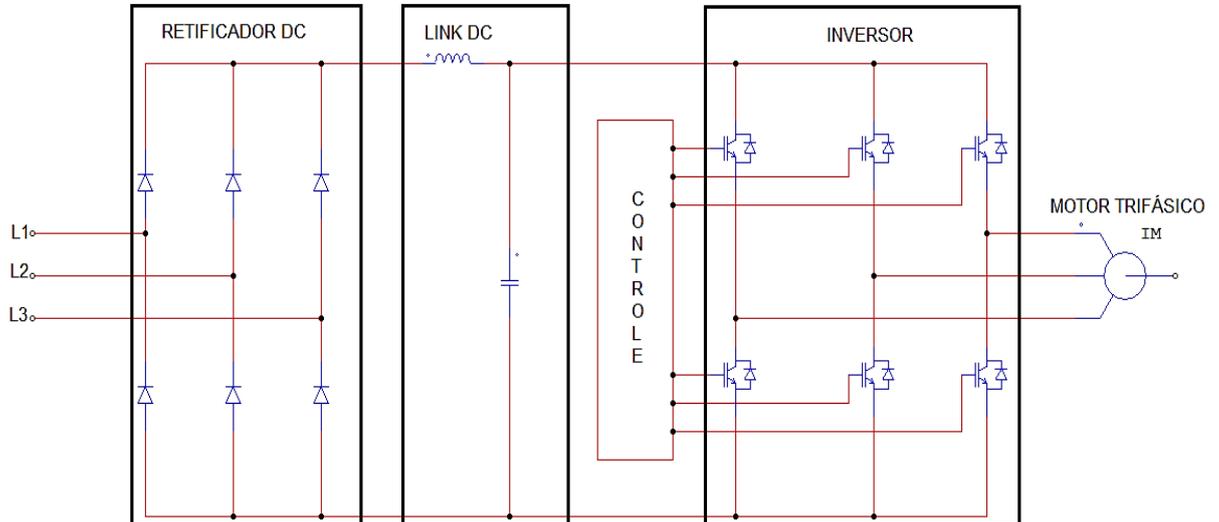
Fonte: WEB (2019).

3.4.3 Circuito de Potência

Constituído por um circuito retificador que recebe a alimentação da rede com a função de transformar tensão CA em CC, o circuito normalmente é constituído por diodos. Um filtro ou *Link DC* regula a tensão retificada com um banco de capacitores. A etapa final trata-se de um circuito inversor, que realiza a conversão de CC para CA com os IGBTs mediante a lógica

do controle de chaveamento (PWM). A etapa de potência é ilustrada pela Figura 11 (WEG, 2019).

Figura 11 - Diagrama da etapa de potência do inversor de frequência



Fonte: Acervo do Autor.

O inversor de frequência CFW-11 realiza o controle de torque e velocidade de motores de indução. Este dispositivo possibilita o controle escalar e vetorial, sendo que no primeiro a relação V/f é mantida constante, assim como o torque, já o controle vetorial é empregado quando se necessita de um controle dinâmico e preciso do torque, assim como resposta rápida frente a uma faixa extensa de condições operacionais (WEG, 2005).

O controle remoto da velocidade do motor pode ser realizado pelo CFW-11 por meio das suas entradas analógicas ou digitais. O controle pelas entradas analógicas pode ser realizado seja por sinal de corrente (de 4 a 20mA) ou tensão (0 a 10V_{cc}), sendo capaz de modificar a velocidade do motor ao longo de toda sua faixa de operação. O controle por entradas digitais recebe sinais lógicos, nível alto $\geq 18V$ e nível baixo $\leq 3V$, permitindo um controle discreto da velocidade (WEG, 2019).

Neste trabalho é empregado o controle remoto da velocidade de MIT por meio das entradas discretas do inversor CFW-11 da WEG. A função utilizada para tal controle foi a *Multispeed*, a qual permite selecionar até oito velocidades diferentes, que são comandadas pelas entradas digitais (DI4, DI5, DI6). A velocidade do motor será estabelecida nas velocidades de referências programadas no inversor. Para ativar esta função é necessário configurar o parâmetro P0221=8 e/ou P0222=8 (Seleção de Referência) e ativar as entradas digitais, conforme a Tabela 4.

Tabela 4 - Seleção da função Multispeed via entradas digitais.

DIx habilitada	Programação
DI4	P0266=13
DI5	P0267=13
DI6	P0268=13

Fonte: WEG (2019).

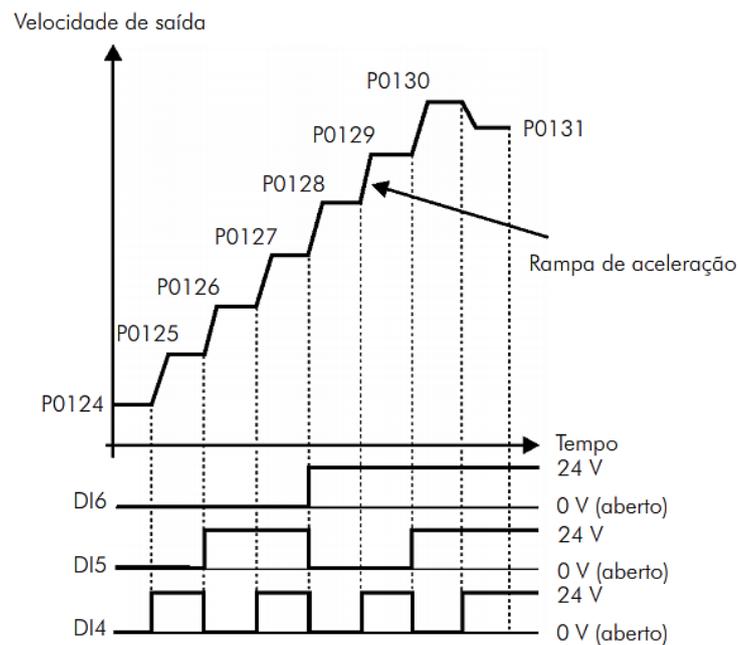
As velocidades de referências são as velocidades pré-estabelecidas entre os parâmetros P0124 ao P0131 e de acordo com a alimentação nas entradas digitais será selecionada a velocidade, como pode ser visto na Tabela 5 e na Figura 12.

Tabela 5 - Referência Multispeed.

8 velocidades			
		4 velocidades	
		2 velocidades	
DI6	DI5	DI4	Ref. de Veloc.
0 V	0 V	0 V	P0124
0 V	0 V	24 V	P0125
0 V	24 V	0 V	P0126
0 V	24 V	24 V	P0127
24 V	0 V	0 V	P0128
24 V	0 V	24 V	P0129
24 V	24 V	0 V	P0130
24 V	24 V	24 V	P0131

Fonte: WEG (2019).

Figura 12 - Multispeed



Fonte: WEG (2019).

3.5 Proposta

O sistema sugerido de controle remoto de velocidade de motor de indução tem como objetivo atender às aplicações específicas com utilização de MIT nas indústrias, as quais exigem o acionamento e controle de velocidade remoto de equipamentos que estão a uma grande distância do centro de controle.

Um sistema de monitoramento é responsável por realizar a comunicação “Homem e Máquina”, promove a seleção e monitoramento da velocidade do MIT, bem como possibilita sua parada. Este sistema é desenvolvido por meio do *software* Processing.

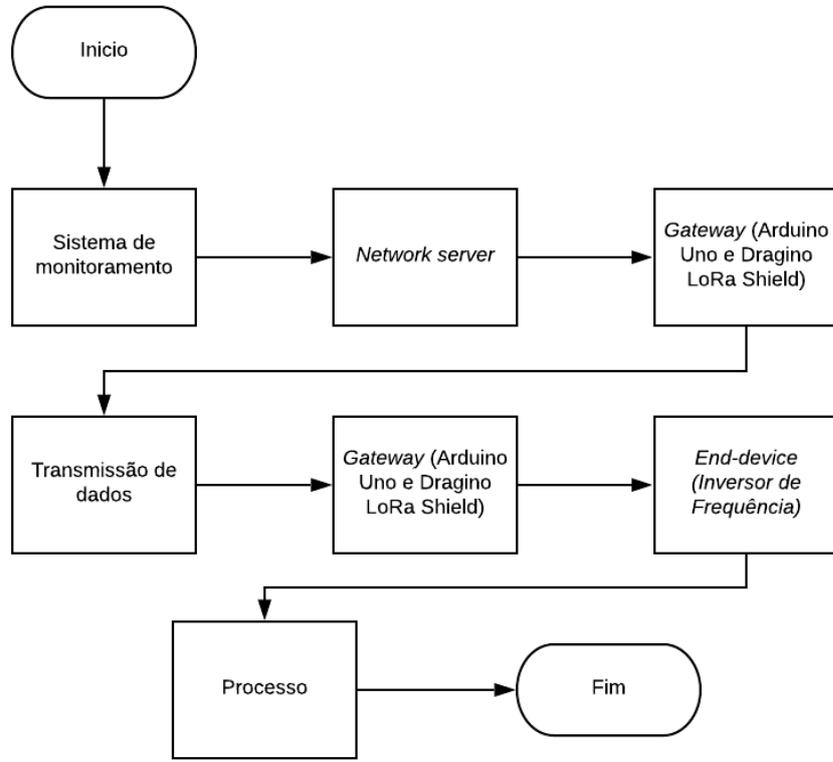
Para a interação entre o monitoramento e a tecnologia de comunicação sem fio, utilizou-se a plataforma Arduino UNO, sendo necessário o emprego de um transmissor e um receptor dos dados, o *Dragino LoRa Shield*.

O sistema de monitoramento pode ser realizado por um *Network server*, que tem a função de definir a ação do *end-device*, ou seja, controlar a velocidade. O Arduino UNO e o *Dragino LoRa Shield*, circuito transmissor, são definidos como um *gateway*, em que nesta rede é realizada uma troca de protocolo da comunicação Serial, vinda do monitoramento, para o protocolo LoRaWAN para transmitir para o *end-device*. As informações são recebidas por meio de um *gateway* interligado com o *end-device*, que será responsável por interpretar as informações e atuar, conseqüentemente controlar a velocidade por meio do Inversor de Frequência. Este processo é representado pelo Fluxograma 1.

Com base no Fluxograma 1, pode-se dividir o processo em etapas para o controle de velocidade de um MIT com comunicação sem fio, que também é representada pelo Diagrama em Blocos 1:

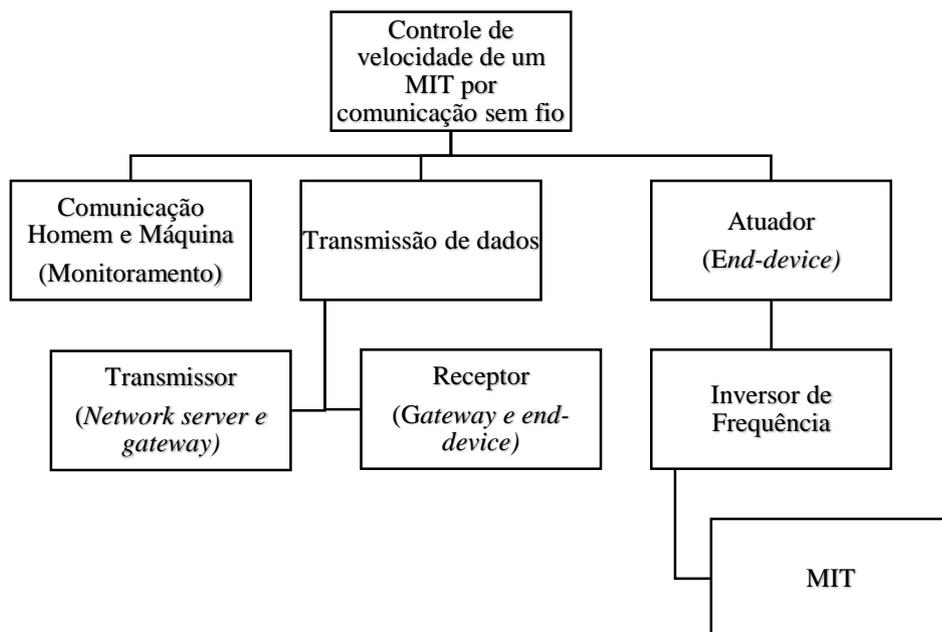
- Monitoramento: ambiente que o operador seleciona a velocidade desejada.
- Circuito transmissor e receptor: responsáveis por enviar e receber as informações, assim como realizar o tratamento das mesmas. Os circuitos são compostos por Arduino Uno e *Dragino LoRa Shield*.
- Atuador: sistema responsável por controlar a velocidade do MIT. Este sistema consiste no uso do Inversor de Frequência.

Fluxograma 1: Sistema de controle de velocidade sem fio.



Fonte: Acervo do Autor.

Diagrama de Blocos 1: Etapas do processo do sistema embarcado de controle.

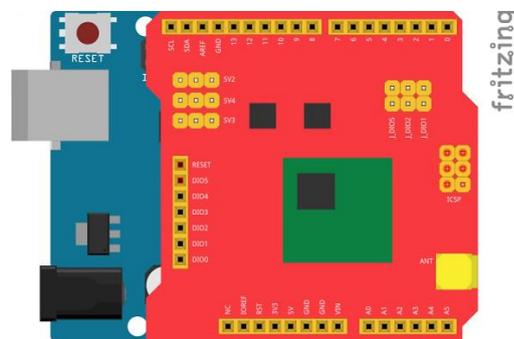


Fonte: Acervo do Autor.

Para que o operador possa seleccionar e monitorar a velocidade é criado um ambiente de monitoramento no *software* Processing. Por meio deste, é possível ligar e desligar o motor, além de escolher dentre 8 velocidades diferentes para o equipamento. Ao ligá-lo, ele parte com uma velocidade de referência configurada no parâmetro P0124 e posteriormente é possível seleccionar dentre as demais velocidades referentes aos parâmetros configurados do P0125 ao P0131.

O *software* Processing permite comunicação serial entre o *sketch*, do ambiente em Processing, e a plataforma Arduino. Então, de acordo com os comandos enviados do *sketch* ao Arduino, as informações são tratadas e são criados os pacotes de informações a serem transmitidas pelo circuito transmissor ao circuito receptor. Os pacotes de informações se traduzem em quais portas digitais do inversor deverão ser habilitadas nível lógico alto para obter as respectivas velocidades do MIT escolhidas. O circuito transmissor (*gateway*) é montado com o acoplamento do *Dragino LoRa Shield* ao Arduino UNO, conforme ilustra a Figura 13.

Figura 13 - Montagem do Dragino LoRa Shield com Arduino



Fonte: Acervo do Autor.

O circuito receptor (*gateway* e *end-device*) executa a recepção dos dados e realiza a lógica para o controle de velocidades, ou seja, realiza a mudança das combinações entre as entradas digitais DI4, DI5 e D6 com níveis lógicos altos e baixos, conforme a Tabela 3. As entradas digitais do inversor exigem uma tensão contínua de 24V, enquanto o Arduino UNO tem apenas uma tensão contínua de 5V, logo é necessário utilizar uma fonte de 24V, geralmente disponível no próprio inversor, e um módulo relé com quatro canais, assim é possível controlar o chaveamento pelo Arduino e aplicar a tensão necessária no inversor. A ilustração da Figura 14 apresenta a ligação do circuito receptor com o módulo relé e a comunicação pelas portas digitais do inversor.

4 RESULTADOS E DISCUSSÕES

Nesta sessão são apresentados os resultados alcançados com a validação do protótipo no Laboratório de Máquinas e Acionamentos do Instituto Federal de Minas Gerais – *Campus Formiga*. São, ainda, retratados os códigos e como foi resultado do monitoramento realizado, a transmissão dos dados, o controle de velocidade e a distância alcançada. Por último são apresentados os custos do projeto.

4.1 Monitoramento

Como é necessário um ambiente de monitoramento, assim como a seleção das velocidades, foi criado um *sketch* que permite utilizar a comunicação pelas portas seriais, através do *software* Processing. As funções que estão demarcadas na Figura 15 são responsáveis pela a comunicação Serial entre o Processing e o Arduino.

Figura 15 - Funções para comunicação Serial entre Processing e Arduino

```

sketch_8velocidades | Processing 3.5.2
Arquivo Editar Sketch Debug Ferramentas Ajuda

sketch_8velocidades
22 boolean stat_vel1 = false;
23 boolean stat_vel2 = false;
24 boolean stat_vel3 = false;
25
26 //Importa a biblioteca para abrir uma comunicação Serial
27 import processing.serial.*;
28 //Instância a biblioteca para a comunicação Serial
29 Serial port;
30
31
32 void setup() {
33
34   size(817, 511);
35   printArray(Serial.list());
36   port = new Serial(this, Serial.list()[0], 9600);
37   fundo = loadImage("Fundo1.jpg");
38   off = loadImage("off.jpg");
39   on = loadImage("on.jpg");
40   offgeral = loadImage("offgeral.jpg");

```

Could not parse -1 for --display [0] "COM11"

Fonte: Acervo do Autor.

Como as informações são enviadas para o Arduino por meio da comunicação Serial, foi necessário utilizar uma função que pudessem “ler” a porta Serial e filtrar o que seria útil para ser utilizado. A informação enviada pela porta Serial através do Processing é demarcada na Figura 16.

A função para “ler” a porta Serial no Arduino está demarcada na Figura 17, que ao receber informações relevantes, como um “a” (Uma variável declarada como *string*) acende um LED e ao receber um “b” apaga o LED.

Figura 16 - Função para enviar informação pela porta Serial

```

sketch_8velocidades | Processing 3.5.2
Arquivo Editar Sketch Debug Ferramentas Ajuda

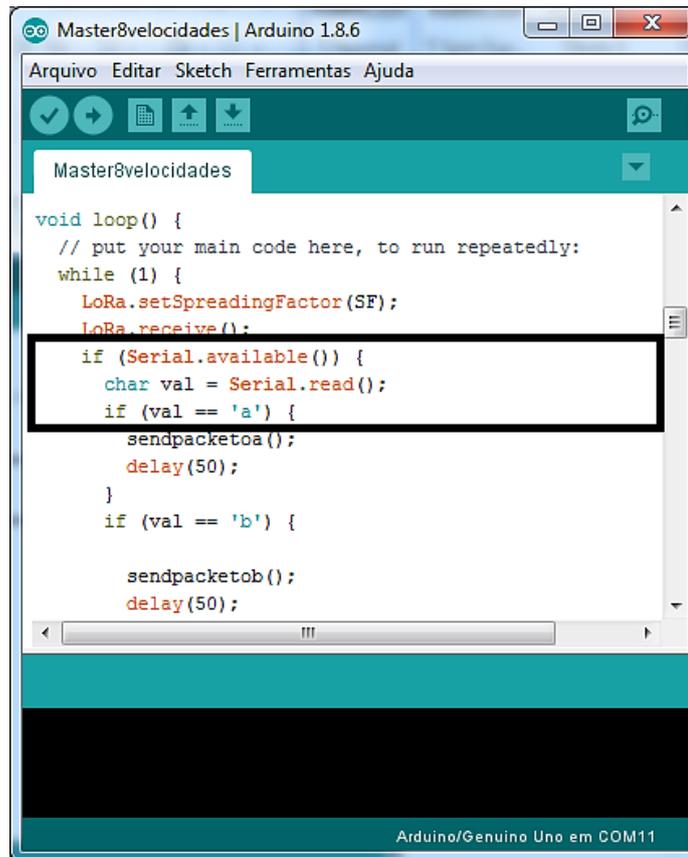
sketch_8velocidades
223   if(aux1==1){ // auxiliar para permitir desligar motor 1
224     offgeral_um();
225   }
226 // Ao pressinar o botao on motor 1, permite desligar motor 1
227   if(m1_on==1){
228     m1_on=0;
229     aux1=1;
230     port.write('a'); //Liga D1 no inversor
231   }
232 // Ao pressinar o botao off motor 1, permite ligar motor 1
233   if(m1_off==1){
234     m1_off=0;
235     aux1=0;
236     port.write('b');//Desliga D1 no inversor
237   }
238   aux_v1=0;
239   aux_v2=0;
240   aux_v3=0;
241   aux_v4=0;
242   aux_v5=0;
243   aux_v6=0;

Could not parse -1 for --display
[0] "COM11"
Console Errors Updates
  
```

Fonte: Acervo do Autor.

Com as principais funções, foi elaborado o *sketch* no Processing, o qual permite ligar e desligar o MIT, assim como selecionar sete velocidades diferentes. A interface para monitoramento e seleção das velocidades é apresentada na Figura 18, o código é apresentado no Anexo I.

Figura 17 - Função para ler as informações da porta Serial



```

Master8velocidades | Arduino 1.8.6
Arquivo Editar Sketch Ferramentas Ajuda

Master8velocidades

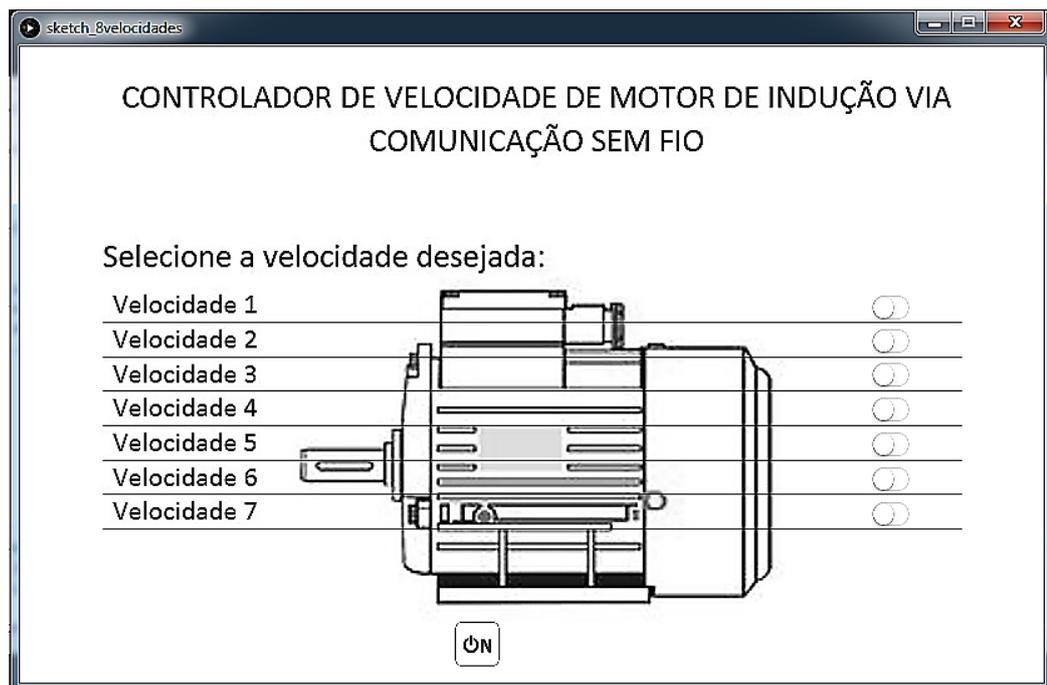
void loop() {
  // put your main code here, to run repeatedly:
  while (1) {
    LoRa.setSpreadingFactor(SF);
    LoRa.receive();

    if (Serial.available()) {
      char val = Serial.read();
      if (val == 'a') {
        sendpacketoa();
        delay(50);
      }
      if (val == 'b') {
        sendpacketob();
        delay(50);
      }
    }
  }
}

Arduino/Genuino Uno em COM11
  
```

Fonte: Acervo do Autor.

Figura 18- Ambiente de monitoramento do controle de velocidades



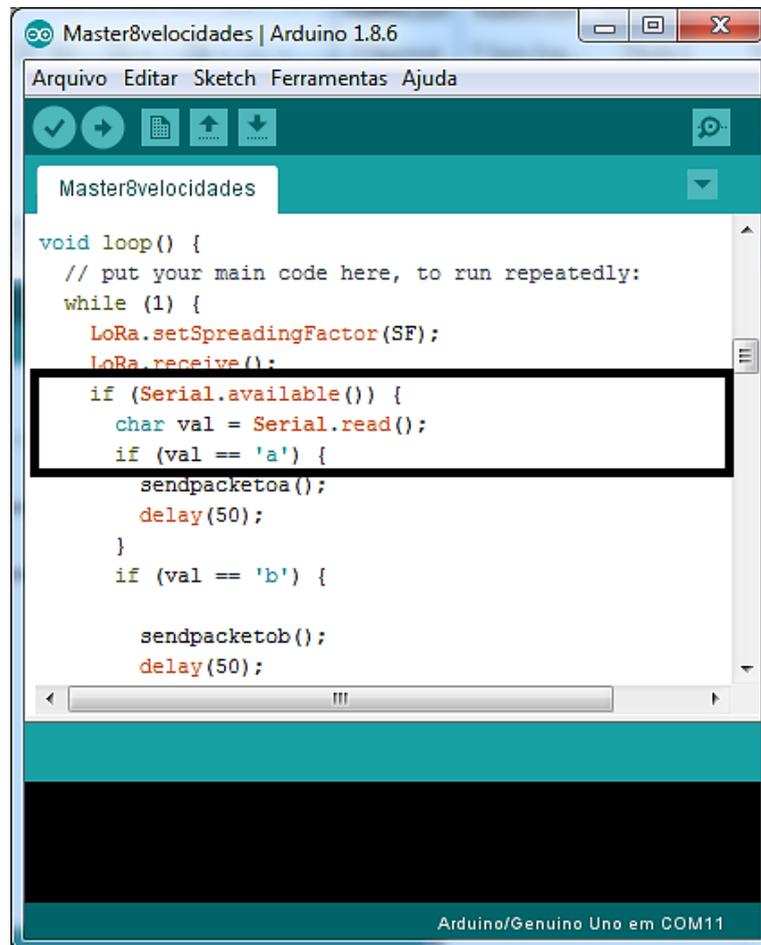
Fonte: Acervo do Autor.

4.2 Comunicação sem fio com *Dragino LoRa Shield*

A rede de comunicação criada pelos *Dragino LoRa Shields* envolve o *gateway*, denominados como circuito transmissor, e o receptor, denominado como *gateway* e *end-device*. Como teste de comunicação e para familiarizar as funções e criações de pacotes de informações, foi implementado um sistema que consiste em transmitir sinais de botões do circuito transmissor para o circuito receptor, o qual acenderia os LEDs respectivos, ou seja, ao pressionar o primeiro botão acenderia o primeiro LED, o segundo botão acenderia o segundo LED, o terceiro botão iria acender o terceiro LED e o quarto botão apagaria todos os LEDs.

Para o circuito transmissor enviar as informações para o circuito receptor, é necessário que sejam criados pacotes de informações. Os botões são conectados ao Arduino e quando são pressionados, são criados os pacotes respectivos para cada botão e enviados para o circuito receptor, os pacotes criados estão demarcados na Figura 19.

Figura 19 - Funções que enviam pacotes de dados pelo *Dragino LoRa Shield*.



```
Master8velocidades | Arduino 1.8.6
Arquivo Editar Sketch Ferramentas Ajuda

Master8velocidades

void loop() {
  // put your main code here, to run repeatedly:
  while (1) {
    LoRa.setSpreadingFactor(SF);
    LoRa.receive();

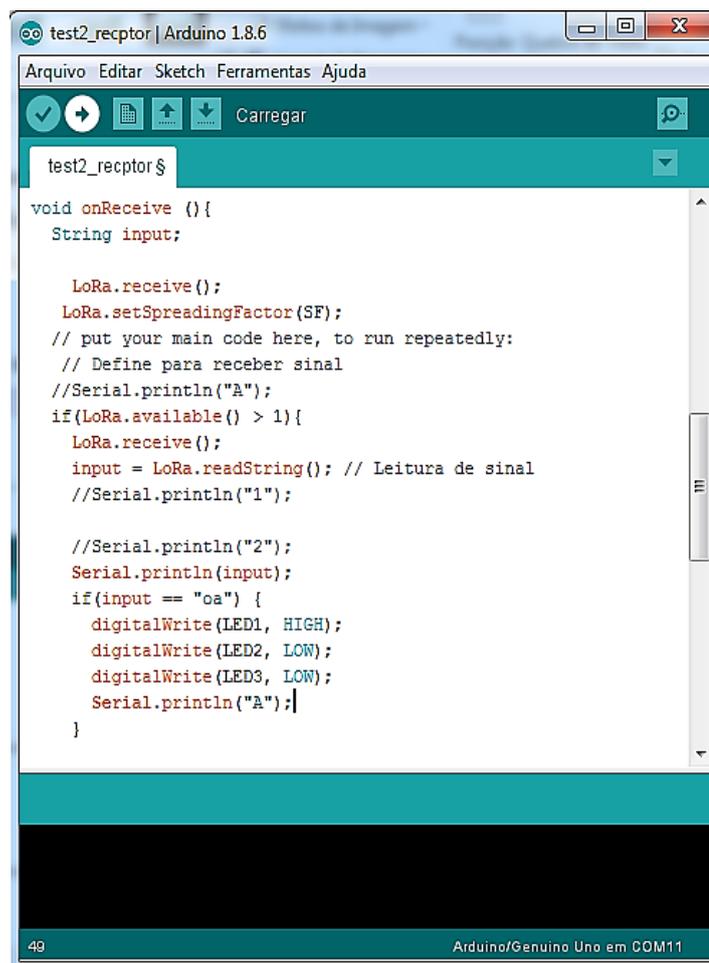
    if (Serial.available()) {
      char val = Serial.read();
      if (val == 'a') {
        sendpacketoa();
        delay(50);
      }
      if (val == 'b') {
        sendpacketob();
        delay(50);
      }
    }
  }
}

Arduino/Genuino Uno em COM11
```

Para receber os pacotes enviados pelo circuito transmissor, o circuito receptor deve estar no modo de recepção de pacotes, assim realizar a leitura dos sinais e por fim atuar e acender ou apagar os LEDs, conforme os pacotes recebidos. A função do circuito criada está apresentada na Figura 20.

Por fim os circuitos foram montados em *protoboard* para validar a comunicação entre o transmissor e o receptor, os quais são representados na Figura 21.

Figura 20 - Função para leitura de sinais com o Dragino LoRa Shield.



```

test2_receptor | Arduino 1.8.6
Arquivo Editar Sketch Ferramentas Ajuda
test2_receptor$
void onReceive () {
  String input;

  LoRa.receive();
  LoRa.setSpreadingFactor(SF);
  // put your main code here, to run repeatedly:
  // Define para receber sinal
  //Serial.println("A");
  if(LoRa.available() > 1){
    LoRa.receive();
    input = LoRa.readString(); // Leitura de sinal
    //Serial.println("1");

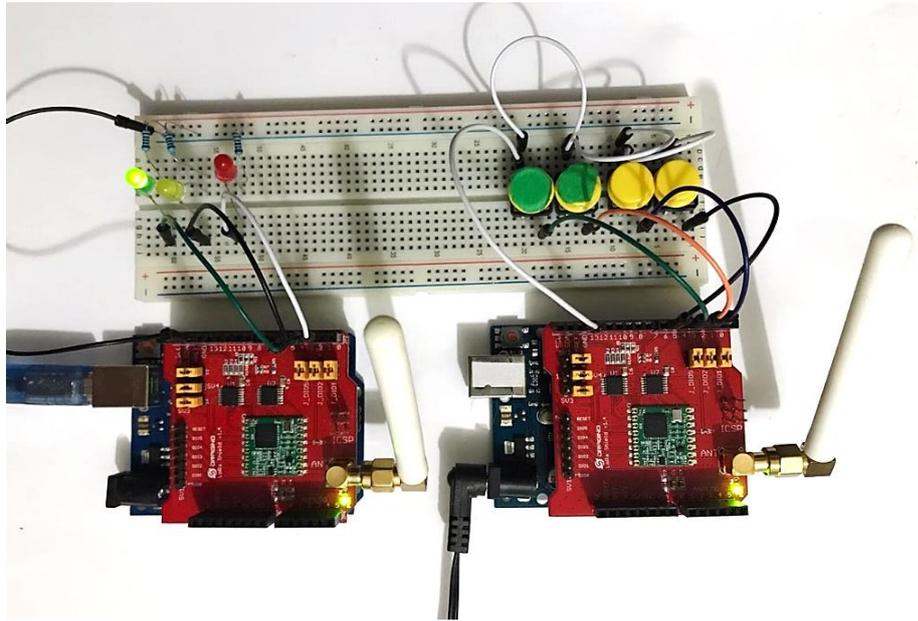
    //Serial.println("2");
    Serial.println(input);
    if(input == "oa") {
      digitalWrite(LED1, HIGH);
      digitalWrite(LED2, LOW);
      digitalWrite(LED3, LOW);
      Serial.println("A");
    }
  }
}
49 Arduino/Genuino Uno em COM11

```

Fonte: Acervo do Autor.

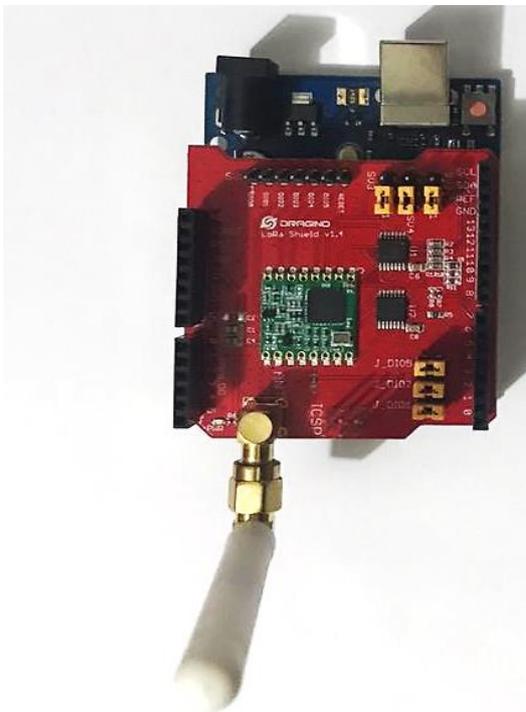
Com êxito na comunicação entre os circuitos transmissor e receptor, com botões e LEDs, foi elaborado um *sketch* no *software* IDE, para o circuito transmissor, que fosse capaz de receber os dados do Processing e enviá-los para o circuito receptor, este código encontra-se no Anexo II. O código versão final do circuito receptor localiza-se no Anexo III. O circuito transmissor é apresentado na Figura 22 (a), enquanto o circuito receptor na Figura 22 (b).

Figura 21 - Circuito transmissor e receptor com botões e LEDs.

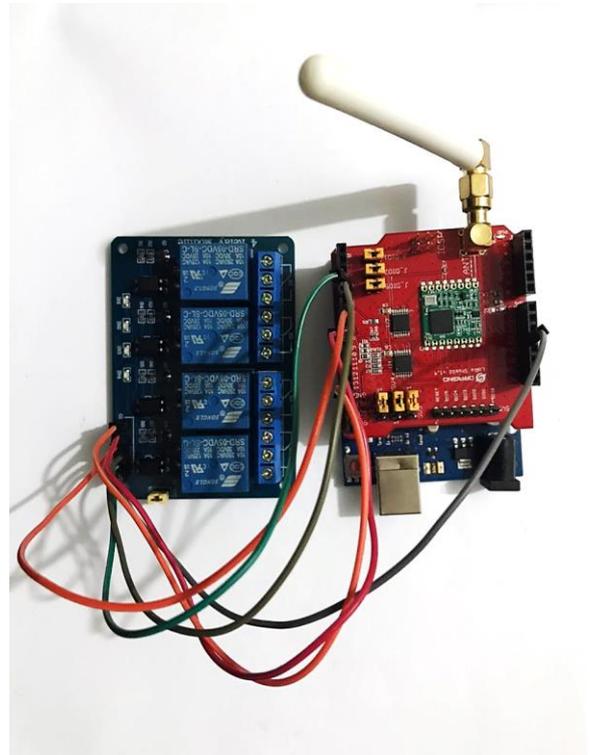


Fonte: Acervo do Autor.

Figura 22 – Circuito transmissor (a) e circuito receptor (b)



(a)



(b)

Fonte: Acervo do Autor.

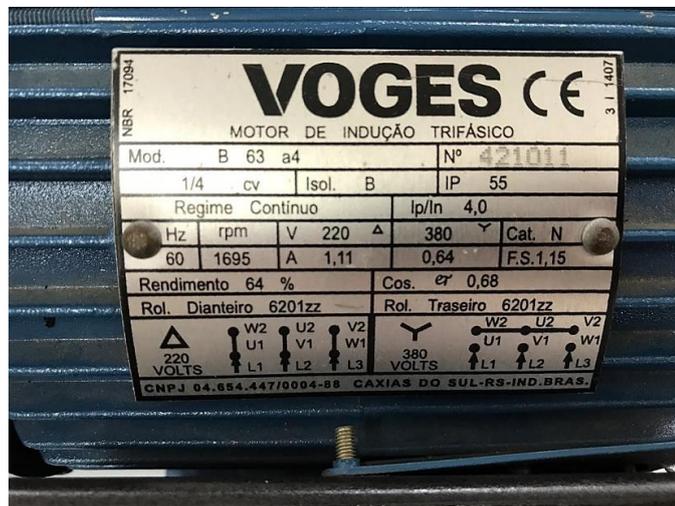
4.3 Controle de velocidade

Com o sistema de transmissão de dados para o controle de velocidade concluído, foi possível realizar o controle de velocidade de um MIT. O circuito receptor foi ligado às entradas digitais por meio do módulo relé com quatro canais, em que o primeiro relé é ligado à porta DI1, responsável por ligar e desligar o MIT e nas portas DI4, DI5 e DI6 para controle de velocidade, conforme apresentado na seção 3.5.

Para habilitar a função *Multispeed* do inversor foi necessário configurar seus parâmetros, para isso foi utilizado o *software* SuperDrive G2, disponibilizado em WEG (2019). Com este *software* é possível alterar os parâmetros do inversor e monitorá-los de forma remota, por meio da entrada USB do inversor de frequência.

Primeiramente, foram alterados os parâmetros relacionados à operação nominal do motor, como tensão, corrente, frequência, potência e velocidade, de acordo com os dados de placa do motor, os quais são apresentados na Figura 23. Portanto, os principais parâmetros foram alterados como demonstra a Figura 24.

Figura 23 - Placa com as características do MIT



Fonte: Acervo do Autor.

Em seguida foi parametrizado o inversor para ativar a função *Multispeed*, vista na Figura 25, e as entradas digitais DI1 e DI4, DI5 e DI6 foram habilitadas, de acordo com a Figura 26.

Por último foram parametrizadas as velocidades de referência, de acordo com a demarcação na Figura 27.

Figura 24 - Parametrização do MIT

Editor de Parâmetros Offline: Endereço 1 - Multispeedfinal.par

Localizar: Localizar próxima Coincidir conteúdo da célula inteira

Número	Função	Mínimo	Máximo	Ajuste de Fábrica	Ajuste do Usuário	Unidade	Comparação
384	Temper. F/A Sensor 4	-20	200	130	130	°C	
385	Tipo do Sensor PTC5	0	1	1: PTC Triplo	1: PTC Triplo		
386	Conf. F/A Sensor 5	0	7	1: Falha/Al./Cabo	1: Falha/Al./Cabo		
387	Temper. F/A Sensor 5	-20	200	130	130	°C	
388	Temperatura Sensor 1	-20	200	0	0	°C	
389	Temperatura Sensor 2	-20	200	0	0	°C	
390	Temperatura Sensor 3	-20	200	0	0	°C	
391	Temperatura Sensor 4	-20	200	0	0	°C	
392	Temperatura Sensor 5	-20	200	0	0	°C	
393	Maior Temp. Sensores	-20	200	0	0	°C	
397	Compens. Escala Regen.	0	1	1: Auto	1: Auto		
398	Fator Serviço Motor	1	1,5	1	1,15		Alterado
399	Rendimento Nom. Motor	50	99,9	67	67	%	
400	Tensão Nominal Motor	0	690	220	220	V	
401	Corrente Nom. Motor	0	9,1	7	1,1	A	Alterado
402	Rotação Nom. Motor	0	18000	1750	1695	rpm	Alterado
403	Frequência Nom. Motor	0	300	60	60	Hz	
404	Potência Nom. Motor	0	58	4: 1.5 CV	0: 0.33 CV		Alterado
405	Número Pulsos Encoder	100	9999	1024	1024	ppr	
406	Ventilação do Motor	0	2	0: Autoventilado	0: Autoventilado		
407	Fator Pot. Nom. Motor	0,5	0,99	0,68	0,68		
409	Resistência Estator	0	9,999	0	0	ohm	
410	Corrente Magnetização	0	8,8	4,7	4,7	A	
411	Indutância Dispersão	0	99,99	0	0	mH	
412	Constante Tr	0	9,999	0	0	s	
413	Constante Tm	0	99,99	0	0	s	
431	Número de Pólos	2	24	6	6		
433	Indutância Lq	0	100	0	0	mH	
434	Indutância Ld	0	100	0	0	mH	
435	Constante Ke	0	400	100	100		
438	Ganho Prop. To	0	1,99	0,8	0,8		

Offline

Fonte: Acervo do Autor.

Figura 25 - Parametrização da função Multispeed

Editor de Parâmetros Offline: Endereço 1 - Multispeedfinal.par

Localizar: Localizar próxima Coincidir conteúdo da célula inteira

Número	Função	Mínimo	Máximo	Ajuste de Fábrica	Ajuste do Usuário	Unidade	Comparação
215	Fundo Escala Leitura3	0	200	100	100	%	
216	Contraste Display HMI	0	37	27	27		
217	Bloqueio por Vel. Nula	0	1	0: Inativo	0: Inativo		
218	Saída Bloq. Vel. Nula	0	1	0: Ref. ou Veloc.	0: Ref. ou Veloc.		
219	Tempo com Veloc. Nula	0	9,99	0	0	s	
220	Seleção Fonte LOC/REM	0	14	2: Teda LR (LOC)	1: Sempre REM		Alterado
221	Sel. Referência LOC	0	13	0: HMI	8: Multispeed		Alterado
222	Sel. Referência REM	0	13	1: AI1	8: Multispeed		Alterado
223	Seleção Giro LOC	0	16	2: Teda 00 (0)	2: Teda 00 (0)		
224	Seleção Giro/Pára LOC	0	6	0: Tedas I,O	0: Tedas I,O		
225	Seleção JOG LOC	0	7	1: Teda JOG	1: Teda JOG		
226	Seleção Giro REM	0	16	4: Dix	4: Dix		
227	Seleção Giro/Pára REM	0	6	1: Dix	1: Dix		
228	Seleção JOG REM	0	7	2: Dix	2: Dix		
229	Seleção Modo Parada	0	4	0: Por Rampa	0: Por Rampa		
230	Zona Morta (AIs)	0	1	0: Inativa	0: Inativa		
231	Função do Sinal AI1	0	7	0: Ref. Veloc.	0: Ref. Veloc.		
232	Ganho da Entrada AI1	0	9,999	1	1		
233	Sinal da Entrada AI1	0	3	0: 0 a 10V/20mA	0: 0 a 10V/20mA		
234	Offset da Entrada AI1	-100	100	0	0	%	
235	Filtro da Entrada AI1	0	16	0	0	s	
236	Função do Sinal AI2	0	7	0: Ref. Veloc.	0: Ref. Veloc.		
237	Ganho da Entrada AI2	0	9,999	1	1		
238	Sinal da Entrada AI2	0	4	0: 0 a 10V/20mA	0: 0 a 10V/20mA		
239	Offset da Entrada AI2	-100	100	0	0	%	
240	Filtro da Entrada AI2	0	16	0	0	s	
241	Função do Sinal AI3	0	7	0: Ref. Veloc.	0: Ref. Veloc.		
242	Ganho da Entrada AI3	0	9,999	1	1		
243	Sinal da Entrada AI3	0	3	0: 0 a 10V/20mA	0: 0 a 10V/20mA		
244	Offset da Entrada AI3	-100	100	0	0	%	
245	Filtro da Entrada AI3	0	16	0	0	s	

Offline

Fonte: Acervo do Autor.

Figura 26 - Parametrização das entradas digitais DI1 e DI4 a DI6

Editor de Parâmetros Offline: Endereço 1 - Multispeedfinal.par

Localizar: Localizar próxima Coincidir conteúdo da célula inteira

Número	Função	Mínimo	Máximo	Ajuste de Fábrica	Ajuste do Usuário	Unidade	Comparação
254	Função da Saída AO2	0	23	5: Corrente Saída	5: Corrente Saída		
255	Ganho da Saída AO2	0	9,999	1	1		
256	Sinal da Saída AO2	0	3	0: 0 a 10V/20mA	0: 0 a 10V/20mA		
257	Função da Saída AO3	0	71	2: Veloc. Real	2: Veloc. Real		
258	Ganho da Saída AO3	0	9,999	1	1		
259	Sinal da Saída AO3	0	6	4: 0 a 10 V	4: 0 a 10 V		
260	Função da Saída AO4	0	71	5: Corrente Saída	5: Corrente Saída		
261	Ganho da Saída AO4	0	9,999	1	1		
262	Sinal da Saída AO4	0	6	4: 0 a 10 V	4: 0 a 10 V		
263	Função da Entrada DI1	0	31	1: Gira/Pára	1: Gira/Pára		
264	Função da Entrada DI2	0	31	8: Sentido Giro	8: Sentido Giro		
265	Função da Entrada DI3	0	31	0: Sem Função	0: Sem Função		
266	Função da Entrada DI4	0	31	0: Sem Função	13: Multispeed		Alterado
267	Função da Entrada DI5	0	31	10: JOG	13: Multispeed		Alterado
268	Função da Entrada DI6	0	31	14: 2a. Rampa	13: Multispeed		Alterado
269	Função da Entrada DI7	0	31	0: Sem Função	0: Sem Função		
270	Função da Entrada DI8	0	31	0: Sem Função	0: Sem Função		
271	Função Saída DO1(RL1)	0	42	19: Sem Função	19: Sem Função		
276	Função Saída DO2(RL2)	0	42	2: N > Nx	2: N > Nx		
277	Função Saída DO3(RL3)	0	42	1: N* > Nx	1: N* > Nx		
278	Função da Saída DO4	0	42	0: Sem Função	0: Sem Função		
279	Função da Saída DO5	0	42	0: Sem Função	0: Sem Função		
281	Frequência Fx	0	300	4	4	Hz	
282	Histerese Fx	0	15	2	2	Hz	
283	Tempo para DO2 ON	0	300	0	0	s	
284	Tempo para DO2 OFF	0	300	0	0	s	
285	Tempo para DO3 ON	0	300	0	0	s	
286	Tempo para DO3 OFF	0	300	0	0	s	
287	Histerese Nx/Ny	0	900	18	18	rpm	
288	Velocidade Nx	0	18000	120	120	rpm	
289	Velocidade Ny	0	18000	1800	1800	rpm	

Offline

Fonte: Acervo do Autor.

Figura 27 - Parametrização das velocidades

Editor de Parâmetros Offline: Endereço 1 - Multispeedfinal.par

Localizar: Localizar próxima Coincidir conteúdo da célula inteira

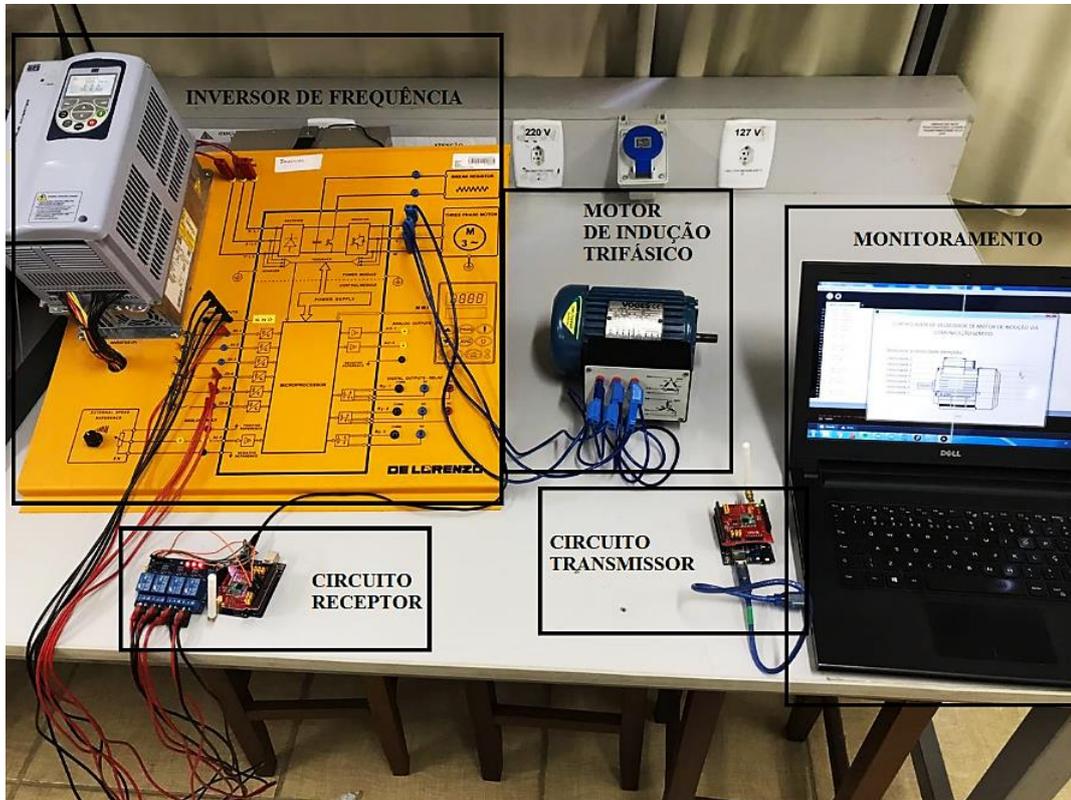
Número	Função	Mínimo	Máximo	Ajuste de Fábrica	Ajuste do Usuário	Unidade	Comparação
101	Tempo Desaceleração	0	999	20	20	s	
102	Tempo Acel. 2a Rampa	0	999	20	20	s	
103	Tempo Desac. 2a Rampa	0	999	20	20	s	
104	Rampa S	0	2	0: Inativa	0: Inativa		
105	Seleção 1a/2a Rampa	0	7	2: D1x	2: D1x		
120	Backup da Ref. Veloc.	0	1	1: Ativa	1: Ativa		
121	Referência pela HMI	0	18000	90	90	rpm	
122	Referência JOG/JOG+	0	18000	150	150	rpm	
123	Referência JOG-	0	18000	150	150	rpm	
124	Ref. 1 Multispeed	0	18000	90	90	rpm	
125	Ref. 2 Multispeed	0	18000	300	300	rpm	
126	Ref. 3 Multispeed	0	18000	600	600	rpm	
127	Ref. 4 Multispeed	0	18000	900	900	rpm	
128	Ref. 5 Multispeed	0	18000	1200	1200	rpm	
129	Ref. 6 Multispeed	0	18000	1500	1500	rpm	
130	Ref. 7 Multispeed	0	18000	1800	1695	rpm	Alterado
131	Ref. 8 Multispeed	0	18000	1650	1650	rpm	
132	Nível Máx. Sobreveloc	0	100	10	10	%	
133	Velocidade Mínima	0	18000	90	90	rpm	
134	Velocidade Máxima	0	18000	1800	1800	rpm	
135	Corrente Máxima Saída	1	11	8	8	A	
136	Boost de Torque Man.	0	9	1	1		
137	Boost de Torque Autom	0	1	0	0		
138	Compensação Escorreg.	-10	10	0	0	%	
139	Filtro Corrente Saída	0	16	0,2	0,2	s	
140	Tempo de Acomodação	0	10	0	0	s	
141	Velocidade Acomodação	0	300	90	90	rpm	
142	Tensão Saída Máxima	0	100	100	100	%	
143	Tensão Saída Intermed	0	100	50	50	%	
144	Tensão Saída em 3Hz	0	100	8	8	%	
145	Vel. Início Enf. Campo	0	18000	1800	1800	rpm	

Offline

Fonte: Acervo do Autor.

Com a parametrização concluída, foi realizada a montagem do inversor com o MIT e os circuitos de comunicação, conforme a Figura 28.

Figura 28 - Montagem para controle de velocidade do MIT por meio da comunicação sem fio



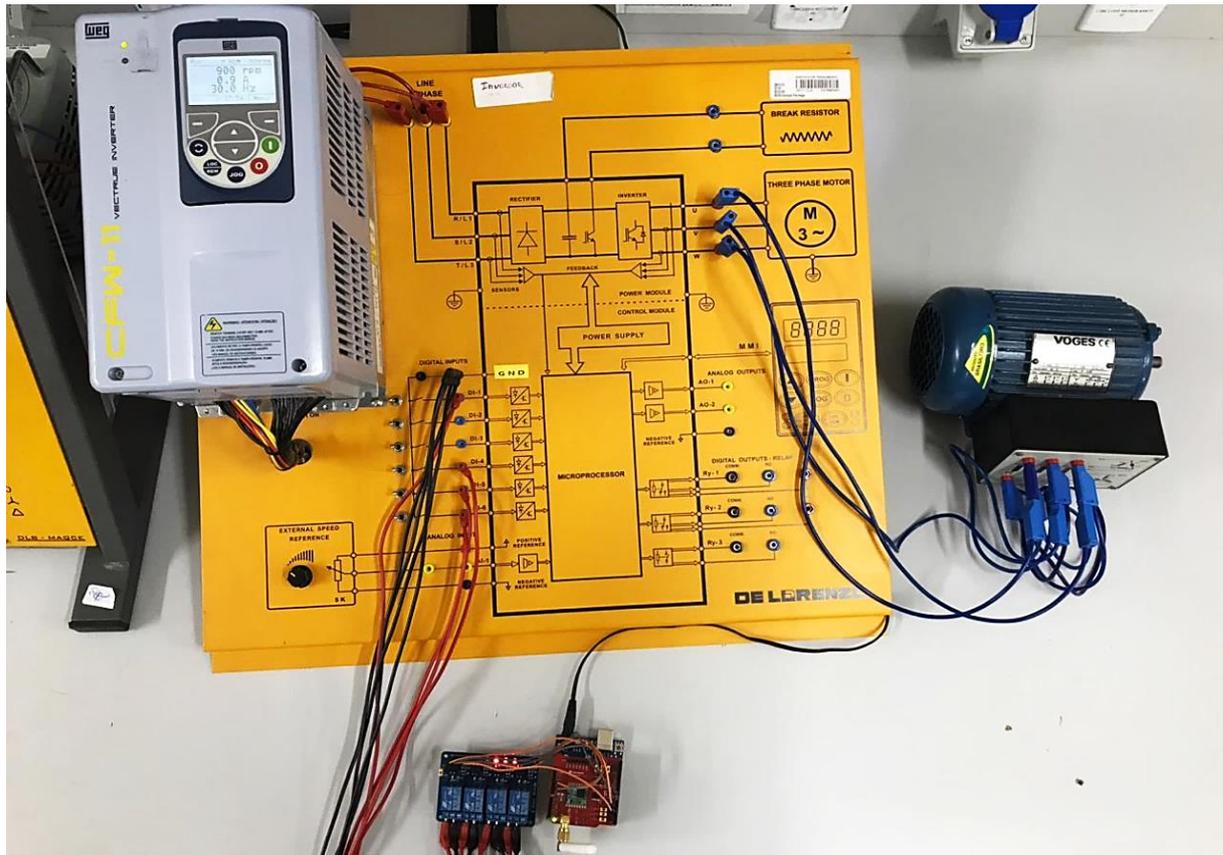
Fonte: Acervo do Autor.

4.4 Distância

Por fim, foram realizados testes para verificar as possíveis distâncias que poderiam ser alcançadas pelo protótipo. Portanto, o circuito receptor, juntamente com o inversor de frequência e o motor de indução trifásico ficaram posicionados no Laboratório de Máquinas do Instituto Federal de Minas Gerais – *Campus* Formiga, conforme a Figura 29. No entanto, foi necessário distanciar o circuito transmissor para testar o alcance da comunicação para a transmissão dos dados.

Os testes foram realizados em três pontos ao longo da cidade de Formiga - MG, cujas diferentes altitudes dos pontos, resultaram em diferentes distâncias de alcance, de aproximadamente 512,49m, 595,79m e o ponto mais distante 806,91m, conforme ilustrado nas Figuras 30, 31 e 32.

Figura 29 - Circuito escravo com inversor de frequência e motor trifásico



Fonte: Acerto do autor.

Figura 30 - Mapa das distâncias de transmissão com 512,49m.



Fonte: GOOGLE MAPS (2019).

Figura 31 - Mapa da transmissão de dados com 595,79m.



Fonte: GOOGLE MAPS (2019).

Figura 32 - Mapa da transmissão de dados com 806,91m.



Fonte: GOOGLE MAPS (2019).

4.5 Custos

Para a construção do protótipo descrito foi realizado um levantamento de custos aproximados dos componentes utilizados por meio de uma pesquisa no mercado *on-line*. O orçamento encontrado é apresentado na Tabela 6.

Tabela 6 - Custos estimado para elaboração do projeto

Componentes	Quantidade	Preço Uni.	Preço Total
Dragino LoRa <i>Shield</i> v1.4	2	R\$ 209,90	R\$ 419,80
Arduino UNO	2	R\$ 42,00	R\$ 84,00
Módulo Relé 4 <i>Channel</i>	1	R\$ 22,50	R\$ 22,50
Fonte 9V	1	R\$ 15,00	R\$ 15,00
Total			R\$ 541,30

Fonte: Acervo do Autor.

4.6 Produção Bibliográfica

O presente trabalho resultou em um artigo aceito para publicação e apresentado na XVII Conferência de Estudos em Engenharia Elétrica em Uberlândia, em novembro de 2019, intitulado como “PROPOSIÇÃO DE SISTEMA DE TRANSMISSÃO DE LONGA DISTÂNCIA PARA SENSORIAMENTO DE TEMPERATURA E UMIDADE”.

5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou resultados provenientes de um estudo sobre a comunicação sem fio que utiliza a tecnologia LoRa, a qual se enquadra nas redes LPWAN. O objetivo do desenvolvimento deste trabalho foi criar um protótipo capaz de realizar o controle de velocidade de um motor de indução trifásico com auxílio de um inversor de tensão por meio de uma comunicação sem fio com uma rede LoRa.

Ao longo do desenvolvimento deste projeto, foi construída uma plataforma de monitoramento, a partir do *software* Processing, que permite a seleção das velocidades do MIT, assim como ligá-lo e desligá-lo.

Para isto, foi utilizada a função *Multispeed*, uma forma de controle remoto e discreto disponível no inversor CFW-11 da WEG, que permite configurar 8 diferentes velocidades e uma partida suave para os motores.

Tanto a parametrização do inversor, realizada por meio do *software* SuperDrive G2, assim como a seleção da velocidade e o monitoramento desta foram realizadas de forma remota. Tal implementação possibilita que equipamentos localizados no “chão de fábrica” sejam controlados de uma sala de controle de forma *on-line*, sem a necessidade do operador estar em contato direto com o equipamento, proporcionando comodidade, flexibilidade e otimização das aplicações.

Além disso, o controle da velocidade do motor foi realizado com a troca de informações por rede sem fio utilizando sistema embarcado, reduzindo a necessidade de cabeamento e, conseqüentemente, os custos da aplicação, como um valor total de R\$ 541,30 para o protótipo proposto, em oposição a um custo de instalação que varia entre R\$ 50,00 e R\$300,00 para uma rede cabeada (MÜLLER, 2016). Além disso, a aplicação mostrou a mobilidade desta rede sem fio, a qual foi testada em diferentes posições da cidade de Formiga - MG, conforme mostram as Figuras 30, 31 e 32.

O Dragino (2017) sugere que o alcance máximo conseguido com o *chip* SX1276 / SX1278 da Semtech é cerca de 1,5km em ambientes ao ar livre e sem obstáculos. Os testes de controle de velocidade resultaram em uma distância máxima de 806,91m em ambiente urbano, tal fato deve-se provavelmente à interferência causada pelas construções urbanas.

Além disso, foi possível caracterizar a aplicação como um sistema de Classe C, pois a parte considerada como circuito transmissor (*network server* e *gateway*) e o circuito receptor (*gateway* e *end-device*) recebem janelas de pacotes durante todo o tempo.

Destaca-se que é possível criar uma rede de comunicação utilizando a tecnologia LoRa para o controle de velocidade de um motor, tornando-a uma alternativa para aplicações industriais, em substituição a tecnologias de curto alcance como o *Bluetooth*, cujo alcance máximo conseguido foi de 2,5m nos resultados obtidos por Neves (2018). Além de poder ser aplicada com outras finalidades, tais como: agricultura, logística, saneamento, entre outros, conforme a própria Alliance (2019) sugere.

5.1 Sugestão para trabalhos futuros

Como forma de desenvolver a técnica abordada e ampliar as possibilidades de aplicação, as sugestões para continuidade da pesquisa são:

- Construir uma estrutura física para a realização de testes em campo, com a finalidade de validar o protótipo em outros ambientes além de laboratórios de ensino;
- Realizar análise de testes de interferências em ambientes industriais;
- Aplicar a comunicação realizada para aplicações como sistemas de irrigação, sistemas de bombeamento de fluidos, sistemas de ventilação e sistemas de movimentação de cargas.

REFERÊNCIAS BIBLIOGRÁFICAS

ABDI, Agência Brasileira de Desenvolvimento Industrial. **Agenda brasileira para Indústria 4.0**. 2018. Disponível em: <http://www.industria40.gov.br/>. Acesso em: 15 out. 2019.

ANATEL, Agência Nacional de Telecomunicações (Brasil). **Resolução nº 680, de 27 de junho de 2017**. [S. l.], 29 jun. 2017. Disponível em: <https://www.anatel.gov.br/legislacao/resolucoes/2017/936-resolucao-680>. Acesso em: 15 out. 2019.

ALLIANCE. **What is the LoRaWAN® Specification?**. [S. l.], 2017. Disponível em: <https://lora-alliance.org/about-lorawan>. Acesso em: 15 out. 2019.

AL-KASHOASH, H. A. A.; KEMP, Andrew H. Comparison of 6LoWPAN and LPWAN for the Internet of Things. **Australian Journal of Electrical and Electronics Engineering**, [s. l.], p. 268-274, 1 dez. 2017. DOI 10.1080/1448837X.2017.1409920. Disponível em: <https://www.tandfonline.com/doi/full/10.1080/1448837X.2017.1409920>. Acesso em: 15 out. 2019.

ARDUINO (Org.). **What is Arduino?**. 2018. Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 15 out. 2019.

BEN, **Balanco Energético Nacional 2004 (ANO BASE 2017)**. [On-line] Disponível em: <www.mme.gov.br>. Acesso em: 15 jan. 2019.

CASSIOLATO, César. **Redes Industriais**. Smar Technology Company, [S. l.], p. 1-16, 1 jan. 2019. Disponível em: <http://www.smar.com/brasil/artigo-tecnico/redes-industriais>. Acesso em: 15 out. 2019.

CHAPMAN, Stephen J. **Fundamentos de Máquinas Elétricas**. 5ª Edição. Bookman. 2013.

CROTTI, Yuri. **Avaliação de Desempenho de Protocolos de Transmissão Para Redes de Sensores Sem Fio Aplicadas à Agricultura**. 2017. 146 p. Trabalho de Conclusão de Curso

(Engenharia de Computação) - UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO, Araranguá, 2017. Disponível
em: <https://repositorio.ufsc.br/bitstream/handle/123456789/182214/Yuri-Crotti-TCC.pdf?sequence=1&isAllowed=y>. Acesso em: 15 out. 2019.

DRAGINO. **Lora Shield**. [S. l.], 20 out. 2017. Disponível em:
<http://www.dragino.com/products/module/item/102-lora-shield.html>. Acesso em: 30 set.
2019.

FINNEGAN, Joseph; BROWN, Stephen. **A Comparative Survey of LPWA
Networking**. arXiv, [s. l.], p. 1-10, 12 fev. 2018. Disponível em:
<https://arxiv.org/pdf/1802.04222.pdf>. Acesso em: 15 out. 2019.

FITZGERALD, A. E., et al. **Máquinas elétricas com Introdução à Eletrônica de Potência**.
6ª Edição. Bookman. 2006.

FOUNDATION, Processing. **Overview. A short introduction to the Processing software
and projects from the community**. [S. l.], 2019. Disponível em:
<https://processing.org/overview/>. Acesso em: 11 set. 2019.

GOOGLE MAPS. **Mapa**. [S. l.], 2019. Disponível em: <https://www.google.com.br/maps/@-20.4530603,-45.4386307,508m/data=!3m1!1e3>. Acesso em: 16 out. 2019.

HADDAD, Jamil. **Conservação de energia**: 2ª Edição Procel/Eletróbrás 2006. [pp 36-41.]

HALLER, Stephan. **The Things in the Internet of Things**. Research Gate, [S. l.], p. 1-3, 10
jan. 2010. Disponível em:
https://www.researchgate.net/publication/228488111_The_Things_in_the_Internet_of_Things?enrichId=rgreq-d7b9cca6f9000351e9d86c36eb357a5d-XXX&enrichSource=Y292ZXJQYWdlOzIyODQ4ODExMTtBUzo0MDQ3NDAzMzM2ODY3ODRAMTQ3MzUwOTAyMzQ3MQ%3D%3D&el=1_x_2&_esc=publicationCoverPdf.
Acesso em: 15 out. 2019.

KARIMI, Kaivan; ATKINSON, Gary. **What the Internet of Things (IoT) Needs to Become a Reality**. White Paper, [S. l.], p. 1-16, 6 maio 2014. Disponível em: <https://www.nxp.com/docs/en/white-paper/INTOTHNGSWP.pdf>. Acesso em: 15 out. 2019.

KOSOW, Irving L. **Máquinas Elétricas e Transformadores**. 4º Edição. Editora Globo. 1982.

LAVRIC, Alexandru; POPA, Valentin. **Internet of Things and LoRa™ Low-Power WideArea Networks: A Survey**. 2017 International Symposium on Signals, Circuits and Systems (ISSCS): IEEE, Iasi, Romania, p. 1-5, 13 jul. 2017. DOI 10.1109/ISSCS.2017.8034915. Disponível em: <https://ieeexplore.ieee.org/document/8034915>. Acesso em: 15 out. 2019.

MELLO, Ronaldo Lewicki Cunha. **Eficiência Energética em Motores Elétricos Trifásicos de Indução**. Orientador: Fábio Marinato de Viveiros. 2013. 7 p. Trabalho de Conclusão de Curso (PósGraduação em Engenharia de Sistemas Eletrônicos, Automação e Controle Industrial) - Instituto Nacional de Telecomunicações, [S. l.], 2013. Disponível em: https://www.academia.edu/38346275/TCC_Efici%C3%Aancia_energ%C3%A9tica_em_motores_el%C3%A9tricos_trif%C3%A1sicos_de_indu%C3%A7%C3%A3o. Acesso em: 15 out. 2019.

MOTTOLA, Luca; PICCO, Gian Pietro. **Programming Wireless Sensor Networks : Fundamental Concepts and State of the Art**. *ACM Computing Surveys*, [s. l.], jan 2011. Disponível em: https://www.researchgate.net/publication/220566260_Programming_Wireless_Sensor_Networks_Fundamental_Concepts_and_State_of_the_Art. Acesso em: 15 out. 2019.

MÜLLER, Ivan. **Redes de Sensores Sem Fio Industriais e o Protocolo WirelessHART**. [S. l.: s. n.], 2016. 132 p. Disponível em: <http://www.inf.ufrgs.br/~caco/ESSE2016/Ivan%20Muller%20IWSN.pdf>. Acesso em: 15 out. 2019.

NERY, Frederico Henrique. **Desenvolvimento de um Protótipo com Utilização de LoRa como Alternativa de Comunicação no Carregamento Ferroviário**. Orientador: Ana Paula Lima Dos Santos. 2018. 62 p. Trabalho de Conclusão de Curso (Bacharelado Engenharia de

Elétrica) - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus Formiga, Formiga, 2018. Disponível em:

https://www.formiga.ifmg.edu.br/documents/2018/Biblioteca/TCCs_e_Artigos/Frederico-Henrique-Nery_EE.pdf. Acesso em: 15 out. 2019.

NEVES, Thuanny Reis. **Sistema Embarcado com Comunicação Sem Fio para Controle de Velocidade de Motor de Indução Trifásico Acionado por Inversor de**. Orientador: Ana

Paula Lima Dos Santos. 2018. 90 p. Trabalho de Conclusão de Curso (Bacharelado Engenharia de Elétrica) - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus Formiga, Formiga, 2018. Disponível em:

https://www.formiga.ifmg.edu.br/documents/2018/Biblioteca/TCCs_e_Artigos/TCC_THUANNYREISNEVES.pdf. Acesso em: 15 out. 2019.

NOLAN, Keith E.; GUIBENE, Wael; KELLY, Mark Y. **An evaluation of low power wide area network technologies for the Internet of Things**. 2016 International Wireless

Communications and Mobile Computing Conference (IWCMC): IEEE, [s. l.], 5 set. 2016.

DOI 10.1109/IWCMC.2016.7577098. Disponível em:

<https://ieeexplore.ieee.org/document/7577098>. Acesso em: 15 out. 2019.

NOREEN, Umber; KEMP, Ahcene; CLAVIER, Laurent. **A Study of LoRa Low Power and Wide Area Network Technology**. 3rd International Conference on Advanced Technologies

for Signal and Image Processing: IEEE, [s. l.], p. 1-6, 22 maio 2017. DOI

10.1109/ATSIP.2017.8075570. Disponível em:

<https://ieeexplore.ieee.org/abstract/document/8075570>. Acesso em: 15 out. 2019.

OLIVEIRA, Rúben Pedrosa. **Redes de sensores com múltiplas tecnologias: curto e longo alcance**. Orientador: Susana Sargento, Lucas Guardalben. 2016. 126 p. Dissertação de

mestrado (Mestrado em Engenharia Eletrônica e Telecomunicações) - Universidade de

Aveiro, Londrina, 2018. Disponível em: <http://hdl.handle.net/10773/22735>. Acesso em: 15

out. 2019.

PETÄJÄJÄRVI, Juha; MIKHAYLOV, Konstantin; HÄMÄLÄINEN, Matti; IINATTI, Jari.

Evaluation of LoRa LPWAN Technology for Remote Health and Wellbeing

Monitoring. IEEE, [s. l.], p. 1-6, 27 jun. 2016. DOI 10.1109/ISMICT.2016.7498898. Disponível em: <https://ieeexplore.ieee.org/document/7498898>. Acesso em: 15 out. 2019.

PILON , Valcir Antonio. **Estudo Para Aplicação de Redes Sem Fio no Ambiente Industrial.** 2009. 52 p. MONOGRAFIA - ESPECIALIZAÇÃO (Especialização em Automação Industrial) - UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ DEPARTAMENTO ACADÊMICO DE ELETRÔNICA ESPECIALIZAÇÃO EM AUTOMAÇÃO INDUSTRIAL, Curitiba, 2009. Disponível em: <http://www.ct.utfpr.edu.br/deptos/ceaut/monografias/EstudoparaAplicacaodeRedesSemFionoAmbienteIndustria.pdf>. Acesso em: 15 out. 2019.

RIZZI, Mattia; FERRARI, Paolo; FLAMMIN, Alessandra; SISINNI, Emiliano. **Evaluation of the IoT LoRaWAN Solution for Distributed Measurement Applications.** IEEE Transactions on Instrumentation and Measurement (Volume: 66, Issue: 12 , Dec. 2017), [s. l.], p. 3340 - 3349, 13 set. 2017. DOI 10.1109/TIM.2017.2746378. Disponível em: <https://ieeexplore.ieee.org/document/8036410>. Acesso em: 15 out. 2019.

ROHDE & SCHWARZ. **Smart Factories and industrial IoT.** 2018. Disponível em: <https://www.rohde-schwarz.com/br/solucoes/test-and-measurement/wireless-communication/iot-m2m/smart-factories/overview/visao-geral-de-fabricas-inteligentes_233886.html?rusprivacypolicy=0>. Acesso em: 16 de jan. de 2019.

SANT'ANA, Jean Michel de Souza. **Redes LoRaWAN: implantação e desenvolvimento de aplicações.** 2017. 67 p. Trabalho de Conclusão de Curso (Bacharelado Engenharia de Telecomunicações) - Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, São José, 2017. Disponível em: https://wiki.sj.ifsc.edu.br/wiki/images/a/ac/TCC290_Jean_Michel_de_Souza_Sant%27Ana.pdf. Acesso em: 15 out. 2019.

SANTOS, D. R. G.; VOLANTE, C. R. **A Importância da tecnologia sem fio na indústria 4.0.** Revista Interface Tecnológica, v. 15, n. 2, p. 245-254, 30 dez. 2018. Disponível em: <https://revista.fatectq.edu.br/index.php/interfacetecnologica/article/view/487>. Acesso em: 15 out. 2019.

SANTOS, Sandro. **Introdução à Indústria 4.0: Saiba tudo sobre a revolução das máquinas**. São Paulo: Copyright, 2018. Disponível em:

<<https://books.google.com.br/books?id=FSRhDwAAQBAJ&pg=PA69&lpg=PA69&dq=conectividade+sem+fio+dentro+da+industria+4.0&source=bl&ots=DaOCg9VDk1&sig=qImU4erPyJEqfJJeMRRmSS9xse4&hl=en&sa=X&ved=2ahUKEwj7ltjgiejdAhVIEJAKHa4FD5M4FBDoATACegQICBAB#v=onepage&q=conectividade%20sem%20fio%20dentro%20da%20industria%204.0&f=false>>. Acesso em: 15 mar. 2019.

SILVEIRA, Simoni Júlia da. **Manual de normalização para apresentação de trabalhos acadêmicos do IFMG – Campus Formiga. Formiga, 2011**. Disponível em: Acesso em: 04 dez. 2019.

SOUZA, Paulo Mateus de. **LPWAN: Redes de Longo Alcance e Baixa Potência**. In: SOUZA, Paulo Mateus de. *Análise sobre as tecnologias Disponíveis em Redes LPWAN*. Londrina: [s. n.], 2018. cap. 2, p. 12-19. Disponível em: http://www.uel.br/cce/dc/wp-content/uploads/Paulo_Preliminar.pdf. Acesso em: 15 out. 2019.

WANG, Lin; HE, Jinfeng; XU, Songjie. **The Application of Industry 4.0 in Customized Furniture Manufacturing Industry**. In: GLOBAL CONGRESS IN MANUFACTURING AND MANAGEMENT, 2016, China, Zhengzhou. Anais [...]. [S. l.: s. n.], 2017. Acesso em: 16 de jan. de 2019.

WEG. **Inversor de frequência CFW-11: manual de programação e manutenção**.

Documento: 0899.5654 / 04. 08 abr. 2019. Disponível em:

<https://static.weg.net/medias/downloadcenter/h2b/h91/WEG-cfw11-manual-de-programacao-0899.5654-2.0x-manual-portugues-br.pdf> . Acesso em: 15 out. 2019.

WEG, AUTOMAÇÃO. **Guia de Aplicação de Inversores de Frequência**. 3. ed. [S. l.]:

WEG AUTOMAÇÃO, 2005. 267 p. v. 1. Disponível em: <https://sidrasul.com.br/wp-content/uploads/2014/09/Guia-de-Aplica%C3%A7%C3%A3o-de-Inversores-de-Frequ%C3%Aancia-WEG-3%C2%AA-Edi%C3%A7%C3%A3o.pdf>.

Acesso em: 15 out. 2019.

ZACARONI, Rodrigo Menezes Sobral. **Projeto e Desenvolvimento de uma Máquina "CNC Fio Quente"**. Orientador: Gustavo Lobato Campos. 2019. 69 p. Trabalho de Conclusão de Curso (Bacharelado Engenharia de Elétrica) - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus Formiga, Formiga, 2019. Disponível em: https://www.formiga.ifmg.edu.br/documents/2019/Biblioteca/TCCs/TCC_Rodrigo__Sobral_Zacaroni.pdf. Acesso em: 15 out. 2019.

ANEXO I

```
/******
```

```
* IFMG - CAMPUS FORMIGA
* Engenharia Elétrica - 2019
* Bruno Gonçalves Silva
* TCC - "Proposição de sistema de transmissão de longa distância para controle de
velocidade
* de motor de indução trifásico"
*/
```

```
import processing.serial.*; //Importa a biblioteca para abrir uma comunicação Serial
Serial port; //Instância a biblioteca para a comunicação Serial
```

```
//Cria uma instância para cada imagem da interface
PImage fundo; //Background
PImage off; //
PImage on; //
PImage offgeral; //LED Vermelho (ligado)
PImage ongeral;
```

```
int ajuste_y=14; //Variável para ajusta o eixo x na tela (Centralizar)
//Se não precisar de ajuste deixe com valor 0
int v1=0, v2=0, v3=0, v4=0, v5=0, v6=0, v7=0, v8=0, m1_on=0, m1_off=0, m2_on=0,
m2_off=0, iniciar=0;
int aux1=0, aux2=0,aux_v1=0, aux_v2=0, aux_v3=0, aux_v4=0, aux_v5=0, aux_v6=0,
aux_v7=0, aux_v8=0, v_aux_1=0, nopress_vel=0;
```

```
boolean stat_vel1 = false;
boolean stat_vel2 = false;
boolean stat_vel3 = false;
```

```
void setup() {

  size(817, 511);
  printArray(Serial.list());
  port = new Serial(this,Serial.list()[0],9600);
  fundo = loadImage("Fundo1.jpg");
  off = loadImage("off.jpg");
  on = loadImage("on.jpg");
  offgeral = loadImage("offgeral.jpg");
  ongeral = loadImage("ongeral.jpg");
}
```

```
void draw(){
  background(fundo);
```

```
  verifica_on_um();
```

```
verifica_off_um();  
verifica_vel_um();  
verifica_vel_dois();  
verifica_vel_tres();  
verifica_vel_quatro();  
verifica_vel_cinco();  
verifica_vel_seis();  
verifica_vel_sete();
```

```
if(iniciar==0){  
    iniciar=1;  
    off_v_um();  
    off_v_dois();  
    off_v_tres();  
    off_v_quatro();  
    off_v_cinco();  
    off_v_seis();  
    off_v_sete();
```

```
    aux_v1=0;  
}
```

```
if(aux_v1==0){  
    off_v_um();  
}  
if(aux_v1==1){  
    on_v_um();  
}
```

```
if(aux_v2==0){  
    off_v_dois();  
}  
if(aux_v2==1){  
    on_v_dois();  
}
```

```
if(aux_v3==0){  
    off_v_tres();  
}  
if(aux_v3==1){  
    on_v_tres();  
}
```

```
if(aux_v4==0){  
    off_v_quatro();  
}  
if(aux_v4==1){  
    on_v_quatro();  
}
```

```
if(aux_v5==0){
    off_v_cinco();
}
if(aux_v5==1){
    on_v_cinco();
}

if(aux_v6==0){
    off_v_seis();
}
if(aux_v6==1){
    on_v_seis();
}

if(aux_v7==0){
    off_v_sete();
}
if(aux_v7==1){
    on_v_sete();
}

if(aux_v8==0){

}

if(v1==1 && aux1==1){ //Seleciona velocidade 1
    v1=0;
    aux_v1=1;
    aux_v2=0;
    aux_v3=0;
    aux_v4=0;
    aux_v5=0;
    aux_v6=0;
    aux_v7=0;

    port.write('c');
}

if(v2==1 && aux1==1){ //Seleciona velocidade 2
    v2=0;
    aux_v1=0;
    aux_v2=1;
    aux_v3=0;
    aux_v4=0;
    aux_v5=0;
    aux_v6=0;
    aux_v7=0;
    port.write('d');
}
```

```
if(v3==1 && aux1==1){//Selecciona velocidad 3
v3=0;
aux_v1=0;
aux_v2=0;
aux_v3=1;
aux_v4=0;
aux_v5=0;
aux_v6=0;
aux_v7=0;
port.write('e');
}
```

```
if(v4==1 && aux1==1){//Selecciona velocidad 4
v4=0;
aux_v1=0;
aux_v2=0;
aux_v3=0;
aux_v4=1;
aux_v5=0;
aux_v6=0;
aux_v7=0;
port.write('f');
}
```

```
if(v5==1 && aux1==1){//Selecciona velocidad 5
v5=0;
aux_v1=0;
aux_v2=0;
aux_v3=0;
aux_v4=0;
aux_v5=1;
aux_v6=0;
aux_v7=0;
port.write('g');
}
```

```
if(v6==1 && aux1==1){//Selecciona velocidad 6
v6=0;
aux_v1=0;
aux_v2=0;
aux_v3=0;
aux_v4=0;
aux_v5=0;
aux_v6=1;
aux_v7=0;
port.write('h');
}
```

```
if(v7==1 && aux1==1){//Selecciona velocidad 7
```

```

v7=0;
aux_v1=0;
aux_v2=0;
aux_v3=0;
aux_v4=0;
aux_v5=0;
aux_v6=0;
aux_v7=1;
port.write('i');
}

if(aux1==0){ // auxiliar para permitir ligar motor 1
  ongeral_um();
}
if(aux1==1){ // auxiliar para permitir desligar motor 1
  offgeral_um();
}

if(m1_on==1){// Ao pressinar o botao on motor 1, permite desligar motor 1
  m1_on=0;
  aux1=1;
  port.write('a'); //Liga D1 no inversor
}
if(m1_off==1){// Ao pressinar o botao off motor 1, permite ligar motor 1
  m1_off=0;
  aux1=0;
  port.write('b');//Desliga D1 no inversor
  aux_v1=0;
  aux_v2=0;
  aux_v3=0;
  aux_v4=0;
  aux_v5=0;
  aux_v6=0;
  aux_v7=0;

}
}

// DEFINIÇÃO DAS POSIÇÕES DAS IMAGENS, AS QUAIS IDENTIFICARAM
QUAL VELOCIDADE ESTÁ SELECIONADA
// E SE O MOTOR ESTA LIGADO. ALEM DE SELECINAR O MOMENTO PARA
LIGAR.
void off_v_um(){
  image(off, 680, 200, 30, 20);
}
void on_v_um(){
  image(on, 680, 200, 30, 20);
}
void off_v_dois(){
  image(off, 680, 227, 30, 20);
}

```

```

}
void on_v_dois(){
    image(on, 680, 227, 30, 20);
}
void off_v_tres(){
    image(off, 680, 254, 30, 20);
}
void on_v_tres(){
    image(on, 680, 254, 30, 20);
}
void off_v_quatro(){
    image(off, 680, 282, 30, 20);
}
void on_v_quatro(){
    image(on, 680, 282, 30, 20);
}
void off_v_cinco(){
    image(off, 680, 310, 30, 20);
}
void on_v_cinco(){
    image(on, 680, 310, 30, 20);
}
void off_v_seis(){
    image(off, 680, 338, 30, 20);
}
void on_v_seis(){
    image(on, 680, 338, 30, 20);
}
void off_v_sete(){
    image(off, 680, 366, 30, 20);
}
void on_v_sete(){
    image(on, 680, 366, 30, 20);
}

void offgeral_um(){
    image(offgeral,405,460,40,40);
}
void ongeral_um(){
    image(ongeral,345,460,40,40);
}

// DEFINIÇÃO DA ÁREA DE ATUAÇÃO DO MOUSE

void verifica_vel_um(){ // VERIFICA POSIÇÃO VELOCIDADE 1 680, 200, 30, 20
    if(mousePressed){
        delay(100);
        if((mouseX>=680)&&(mouseX<=710)){ // Define a área de atuação dos botões
            if((mouseY>=200)&&(mouseY<=220)){

```

```

        v1=1;
    }
}
}
}
void verifica_vel_dois(){ // VERIFICA POSIÇÃO VELOCIDADE 2 680, 227, 30, 20
    if(mousePressed){
        delay(100);
        if((mouseX>=680)&&(mouseX<=710)){ // Define a área de atuação dos botões
            if((mouseY>=227)&&(mouseY<=247)){
                v2=1;
            }
        }
    }
}
void verifica_vel_tres(){ // VERIFICA POSIÇÃO VELOCIDADE 3 680, 254, 30, 20
    if(mousePressed){
        delay(100);
        if((mouseX>=680)&&(mouseX<=710)){ // Define a área de atuação dos botões
            if((mouseY>=254)&&(mouseY<=274)){
                v3=1;
            }
        }
    }
}
void verifica_vel_quatro(){ // VERIFICA POSIÇÃO VELOCIDADE 4 680, 282, 30, 20
    if(mousePressed){
        delay(100);
        if((mouseX>=680)&&(mouseX<=710)){ // Define a área de atuação dos botões
            if((mouseY>=282)&&(mouseY<=302)){
                v4=1;
            }
        }
    }
}
void verifica_vel_cinco(){ // VERIFICA POSIÇÃO VELOCIDADE 3 680, 310, 30, 20
    if(mousePressed){
        delay(100);
        if((mouseX>=680)&&(mouseX<=710)){ // Define a área de atuação dos botões
            if((mouseY>=310)&&(mouseY<=330)){
                v5=1;
            }
        }
    }
}
void verifica_vel_seis(){ // VERIFICA POSIÇÃO VELOCIDADE 3 680, 338, 30, 20
    if(mousePressed){
        delay(100);
        if((mouseX>=680)&&(mouseX<=710)){ // Define a área de atuação dos botões
            if((mouseY>=338)&&(mouseY<=358)){

```

```

        v6=1;
    }
}
}
}
void verifica_vel_sete(){ // VERIFICA POSIÇÃO VELOCIDADE 3 680, 366, 30, 20
    if(mousePressed){
        delay(100);
        if((mouseX>=680)&&(mouseX<=710)){ // Define a área de atuação dos botões
            if((mouseY>=366)&&(mouseY<=386)){
                v7=1;
            }
        }
    }
}
}
}
void verifica_on_um(){ // VERIFICA POSIÇÃO 345,460,40,40
    if(mousePressed){
        if((mouseX>=345)&&(mouseX<=385)){ // Define a área de atuação dos botões
            if((mouseY>=460)&&(mouseY<=500)){
                m1_on=1;
            }
        }
    }
}
}
}
void verifica_off_um(){ // VERIFICA POSIÇÃO 405,460,40,40
    if(mousePressed){
        if((mouseX>=405)&&(mouseX<=445)){ // Define a área de atuação dos botões
            if((mouseY>=460)&&(mouseY<=500)){
                m1_off=1;
            }
        }
    }
}
}
}
}

```

ANEXO II

```

/*****
* IFMG - CAMPUS FORMIGA
* Engenharia Elétrica - 2019
* Bruno Gonçalves Silva
* TCC - "Proposição de sistema de transmissão de longa distância para controle de
velocidade
* de motor de indução trifásico"
*/

//Transmissor

#define LORAFREQ (915000000L)
#define SF 11
#include <SPI.h>
#include <LoRa.h>
#include <SoftwareSerial.h>

//#define BTN1 4 // Define botão no pino 4
//#define BTN2 8
//#define BTN3 3
//#define BTN4 0

int von = 0;

String turnOa = "oa";
String turnOb = "ob";
String turnOc = "oc";
String turnOd = "od";
String turnOe = "oe";
String turnOf = "of";
String turnOg = "og";
String turnOh = "oh";
String turnOi = "oi";

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  // pinMode(BTN1, INPUT_PULLUP);
  // pinMode(BTN2, INPUT_PULLUP);
  // pinMode(BTN3, INPUT_PULLUP);
  // pinMode(BTN4, INPUT_PULLUP);
  //Configuração Parâmetros LoRa
  LoRa.begin(LORAFREQ);
  LoRa.setSpreadingFactor(SF);
  LoRa.receive();
}

```

```
void loop() {
  // put your main code here, to run repeatedly:
  while (1) {
    LoRa.setSpreadingFactor(SF);
    LoRa.receive();
    if (Serial.available()) {
      char val = Serial.read();
      if (val == 'a') {
        sendpacketoa();
        delay(50);
      }
      if (val == 'b') {

        sendpacketob();
        delay(50);
      }
      if (val == 'c') {
        sendpacketoc();
        delay(50);
      }
      if (val == 'd') {
        sendpacketod();
        delay(50);
      }
      if (val == 'e') {
        sendpacketoe();
        delay(50);
      }
      if (val == 'f') {
        sendpacketof();
        delay(50);
      }
      if (val == 'g') {
        sendpacketog();
        delay(50);
      }
      if (val == 'h') {
        sendpacketoh();
        delay(50);
      }
      if (val == 'i') {
        sendpacketoi();
        delay(50);
      }
    }

    // if(digitalRead(BTN1) == 0) {
    //   sendpacketoa();
    //   while(digitalRead(BTN1) == 0);
```

```

// delay(50);
// }
// if(digitalRead(BTN2) == 0) {
//   sendpacketob();
//   while(digitalRead(BTN2) == 0);
//   delay(50);
// }
// if(digitalRead(BTN3) == 0) {
//   sendpacketoc();
//   while(digitalRead(BTN3) == 0);
//   delay(50);
// }
// if(digitalRead(BTN4) == 0) {
//   sendpacketod();
//   while(digitalRead(BTN4) == 0);
//   delay(50);
// }
}
}

void sendpacketoa () {
  LoRa.beginPacket();
  LoRa.print(turnOa);
  Serial.println("A");
  LoRa.endPacket(); //Término do envio de pacote
  LoRa.receive(); //Retorna o dispositivo para modo de recepção
}

void sendpacketob () {
  LoRa.beginPacket();
  LoRa.print(turnOb);
  Serial.println("B");
  LoRa.endPacket(); //Término do envio de pacote
  LoRa.receive(); //Retorna o dispositivo para modo de recepção
}

void sendpacketoc () {
  LoRa.beginPacket();
  LoRa.print(turnOc);
  Serial.println("C");
  LoRa.endPacket(); //Término do envio de pacote
  LoRa.receive(); //Retorna o dispositivo para modo de recepção
}

void sendpacketod () {
  LoRa.beginPacket();
  LoRa.print(turnOd);
  Serial.println("D");
  LoRa.endPacket(); //Término do envio de pacote
  LoRa.receive(); //Retorna o dispositivo para modo de recepção
}

```

```
}  
void sendpacketoe () {  
    LoRa.beginPacket();  
    LoRa.print(turnOe);  
    Serial.println("E");  
    LoRa.endPacket(); //Término do envio de pacote  
    LoRa.receive(); //Retorna o dispositivo para modo de recepção  
}  
void sendpacketof () {  
    LoRa.beginPacket();  
    LoRa.print(turnOf);  
    Serial.println("F");  
    LoRa.endPacket(); //Término do envio de pacote  
    LoRa.receive(); //Retorna o dispositivo para modo de recepção  
}  
void sendpacketog () {  
    LoRa.beginPacket();  
    LoRa.print(turnOg);  
    Serial.println("G");  
    LoRa.endPacket(); //Término do envio de pacote  
    LoRa.receive(); //Retorna o dispositivo para modo de recepção  
}  
void sendpacketoh () {  
    LoRa.beginPacket();  
    LoRa.print(turnOh);  
    Serial.println("H");  
    LoRa.endPacket(); //Término do envio de pacote  
    LoRa.receive(); //Retorna o dispositivo para modo de recepção  
}  
void sendpacketoi () {  
    LoRa.beginPacket();  
    LoRa.print(turnOi);  
    Serial.println("I");  
    LoRa.endPacket(); //Término do envio de pacote  
    LoRa.receive(); //Retorna o dispositivo para modo de recepção  
}
```

ANEXO III

```

/*****
* IFMG - CAMPUS FORMIGA
* Engenharia Elétrica - 2019
* Bruno Gonçalves Silva
* TCC - "Proposição de sistema de transmissão de longa distância para controle de
velocidade
* de motor de indução trifásico"
*/
//Receptor

#define LORAFREQ (915000000L)
#define SF 11
#include <SPI.h>
#include <LoRa.h>
#include <SoftwareSerial.h>

int rele1 = 3;
int rele2 = 4;
int rele3 = 8;
int rele4 = 7;

int von = 0;

void setup() {
  // put your setup code here, to run once:
  pinMode(rele1, OUTPUT);
  pinMode(rele2, OUTPUT);
  pinMode(rele3, OUTPUT);
  pinMode(rele4, OUTPUT);

  digitalWrite(rele1, HIGH);
  digitalWrite(rele2, HIGH);
  digitalWrite(rele3, HIGH);
  digitalWrite(rele4, HIGH);

  Serial.begin(9600);
  LoRa.begin(LORAFREQ);
  LoRa.setSpreadingFactor(SF);
  LoRa.onReceive (onReceive);
  LoRa.receive();
}

void loop() {

}

void onReceive () {
  String input;

```

```

LoRa.receive();
LoRa.setSpreadingFactor(SF);
// put your main code here, to run repeatedly:
// Define para receber sinal
//Serial.println("A");
if (LoRa.available() > 1) {
  LoRa.receive();
  input = LoRa.readString(); // Leitura de sinal
  //Serial.println("1");

  //Serial.println("2");
  Serial.println(input);
  if (input == "oa") {
    digitalWrite(rele1, LOW);
    digitalWrite(rele2, HIGH);
    digitalWrite(rele3, HIGH);
    digitalWrite(rele4, HIGH);
    von = 1;
    Serial.println("A");
  }

  LoRa.receive();
  if (input == "ob") {
    digitalWrite(rele1, HIGH);
    digitalWrite(rele2, HIGH);
    digitalWrite(rele3, HIGH);
    digitalWrite(rele4, HIGH);
    von = 0;
    Serial.println("B");
  }

  LoRa.receive();
  if (input == "oc" && von == 1) {
    digitalWrite(rele3, HIGH);
    digitalWrite(rele4, HIGH);
    digitalWrite(rele2, LOW);

    Serial.println("C");
  }

  LoRa.receive();
  if (input == "od" && von == 1) {
    digitalWrite(rele2, HIGH);
    digitalWrite(rele4, HIGH);
    digitalWrite(rele3, LOW);
    Serial.println("D");
  }

  LoRa.receive();

```

```
if (input == "oe" && von == 1) {  
  
    digitalWrite(rele4, HIGH);  
    digitalWrite(rele2, LOW);  
    digitalWrite(rele3, LOW);  
    Serial.println("E");  
}  
  
LoRa.receive();  
if (input == "of" && von == 1) {  
    digitalWrite(rele2, HIGH);  
    digitalWrite(rele3, HIGH);  
    digitalWrite(rele4, LOW);  
    Serial.println("F");  
}  
  
LoRa.receive();  
if (input == "og" && von == 1) {  
    digitalWrite(rele3, HIGH);  
    digitalWrite(rele2, LOW);  
    digitalWrite(rele4, LOW);  
    Serial.println("G");  
}  
  
LoRa.receive();  
if (input == "oh" && von == 1) {  
    digitalWrite(rele2, HIGH);  
    digitalWrite(rele3, LOW);  
    digitalWrite(rele4, LOW);  
    Serial.println("H");  
}  
  
LoRa.receive();  
if (input == "oi" && von == 1) {  
    digitalWrite(rele2, LOW);  
    digitalWrite(rele3, LOW);  
    digitalWrite(rele4, LOW);  
    Serial.println("I");  
}  
  
}  
delay(50);  
}
```