

**FACULDADE DE TECNOLOGIA DE SÃO BERNARDO DO CAMPO  
“ADIB MOISÉS DIB”**

**LUCAS ARAUJO DE FARIA  
RENAN TELES DELGADO**

**SISTEMA DE IDENTIFICAÇÃO DE LINHAS DE ÔNIBUS PARA DEFICIENTES  
VISUAIS**

São Bernardo do Campo – SP  
Junho/2018

**LUCAS ARAUJO DE FARIA**

**RENAN TELES DELGADO**

**SISTEMA DE IDENTIFICAÇÃO DE LINHAS DE ÔNIBUS PARA DEFICIENTES  
VISUAIS**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Bernardo do Campo “Adib Moises Dib” como requisito parcial para a obtenção do título de Tecnólogo em Automação Industrial.

Orientador: Prof. Me. Pedro Adolfo Galani  
Coorientador: Prof. Dr. Delcínio Ricci

São Bernardo do Campo - SP  
Junho/2018

**LUCAS ARAUJO DE FARIA  
RENAN TELES DELGADO**

**SISTEMA DE IDENTIFICAÇÃO DE LINHAS DE ÔNIBUS PARA DEFICIENTES  
VISUAIS**

Trabalho de Conclusão de Curso  
apresentado à Faculdade de Tecnologia  
de São Bernardo do Campo “Adib Moises  
Dib” como requisito parcial para a  
obtenção do título de Tecnólogo (a) em  
Automação Industrial.

Trabalho de Conclusão de Curso apresentado e aprovado em: \_\_\_\_/\_\_\_\_/ 2018

Banca examinadora:

---

Prof. Me. Pedro Adolfo Galani , FATEC SBC - Orientador

---

Prof. \_\_\_\_\_ , FATEC SBC - Avaliador

---

Prof. \_\_\_\_\_ , FATEC SBC - Avaliador

Dedicamos esse trabalho aos nossos pais, amigos e familiares que de certa forma nos incentivaram, nos compreenderam e estiveram em nosso lado apoiando para realização deste trabalho, assim como aos professores que nos orientaram e direcionaram.

## RESUMO

Pessoas portadoras de algum tipo de deficiência sempre fizeram parte da sociedade em número representativo. As diferentes dificuldades enfrentadas por essas pessoas no dia-a-dia, no entanto, vêm diminuindo consideravelmente com os avanços da tecnologia. A proposta desse trabalho é cooperar com deficientes visuais, auxiliando-os no embarque em ônibus de transporte público sem que seja necessária a ajuda de terceiros. Para realização desse projeto são feitas pesquisas, tanto sobre a realidade do embarque dessas pessoas em linhas de ônibus atualmente, como também acerca das tecnologias que poderiam ser utilizadas para integrar um sistema eficaz, o qual trabalhe em prol dos deficientes, sem que seja algo impraticável para os motoristas dos ônibus. São utilizados nesse trabalho, fundamentalmente, módulos microcontrolados, dispositivos transceptores de rádio-frequência e módulo reproduzidor de áudios, todos programados em linguagem C++.

## **ABSTRACT**

People with disabilities have always been part of society as representative number. However, the different difficulties daily faced by these persons have been decreasing with all the technological advances. The purpose of this study is to cooperate with visually impaired people assisting them on boarding in public buses with no need of help from third parties. For this study to be done, researches concerning the reality of the currently boarding process in buses and also about the technologies which could be used to get an efficient system are required. The intended system must be capable to work in favor of visually impaired besides being practical to the buses drivers. This project uses fundamentally microcontroller modules, transceivers devices and audio reproducer module, all of them are programmed in C++ language.

## LISTA DE FIGURAS

Figura 1.1 – Identificação da linha dos ônibus .....	12
Figura 1.2 – Elevador da estação Paraíso do Metrô de SP .....	14
Figura 1.3 – Plataforma da estação Itapevi da CPTM.....	14
Figura 1.4 – Ônibus com rampa de acessibilidade .....	15
Figura 1.5 – Porcentagem de ônibus acessíveis em algumas capitais .....	16
Figura 1.6 – Comparação entre comunicação serial e comunicação paralela .....	17
Figura 1.7 – Comunicação serial síncrona em diagrama .....	18
Figura 1.8 – Comunicação serial assíncrona em diagrama .....	18
Figura 1.9 – Transmissão de <i>bits</i> em comunicação assíncrona.....	19
Figura 1.10 – Modelo genérico de comunicação.....	19
Figura 1.11 – Modelos de codificação de sinais digitais em ondas analógicas.....	21
Figura 1.12 – Modulação FSK.....	22
Figura 1.13 – Filtro Gaussiano aplicado em um sinal analógico .....	23
Figura 1.14 – Modelo de comunicação mestre-escravo.....	24
Figura 1.15 – Comunicação <i>Half-duplex</i> .....	25
Figura 1.16 – Comunicação <i>Full-duplex</i> .....	25
Figura 1.17 – Diagrama de exemplificação – Interface SPI .....	26
Figura 1.18 – Comparativo entre diferentes padrões de comunicação serial.....	27
Figura 1.19 – Comunicação sem fio via satélite .....	28
Figura 1.20 – Onda Eletromagnética.....	29
Figura 1.21 – Transmissão e recepção de ondas eletromagnéticas por antenas .....	30
Figura 1.22 – Placa dedicada do módulo NRF24L01 .....	31
Figura 1.23 – Circuito eletrônico da placa dedicada NRF24L01 .....	32
Figura 1.24 – Disposição dos terminais externos módulo NRF24L01 .....	33
Figura 1.25 – Diagrama de Blocos Interno Módulo NRF24L01 .....	34

Figura 1.26 – DFPlayer mini.....	35
Figura 1.27 – Pinagem do módulo DFPlayer mini.....	35
Figura 1.28 – Microcontrolador .....	36
Figura 1.29 – Diagrama de blocos de um microcontrolador.....	37
Figura 1.30 – Linguagem C e <i>assembly</i> .....	39
Figura 2.1 – Arquitetura do sistema .....	40
Figura 2.2 – Fluxograma do funcionamento do processo .....	42
Figura 3.1 – Projeto finalizado.....	45
Figura 3.2 – Programa “Blink”.....	46
Figura 3.3 – Teste “blink” funcionando.....	47
Figura 3.4 – Bibliotecas utilizadas na programação.....	48
Figura 3.5 – Ligação entre Arduino e NRF24L01.....	48
Figura 3.6 – Ligação do Arduino com o módulo de áudio.....	50
Figura 3.7– Programação para o DFPlayer.....	51
Figura 3.8 – Teste com módulos integrados.....	53
Figura 3.9 – Furações na miniatura e colocação das interfaces.....	55
Figura 3.10 – Fixação dos componentes na miniatura.....	56
Figura 3.11 – Furações no gabinete e colocação das interfaces.....	57
Figura 3.12– Fixação dos componentes no gabinete.....	57



## SUMÁRIO

<b>INTRODUÇÃO .....</b>	<b>10</b>
<b>1 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>12</b>
1.1 Identificação das linhas de ônibus e acessibilidade .....	12
1.2 Comunicação paralela e serial .....	16
1.3 Protocolo de comunicação .....	19
1.3.1 Codificação e decodificação .....	21
1.3.2 Modulação GFSK .....	22
1.3.3 Acesso ao meio mestre-escravo.....	23
1.3.4 Comunicação <i>half-duplex</i> e <i>full-duplex</i> .....	24
1.3.5 Protocolo de comunicação SPI.....	26
1.4 Comunicação sem fio via radio-freqüência (RF).....	28
1.5 Módulo de radio-freqüência NRF24L01 .....	30
1.6 Módulo de áudio DFPlayer Mini.....	34
1.7 Microcontroladores e linguagem de programação.....	36
<b>2 METODOLOGIA.....</b>	<b>39</b>
2.1 A determinação do tema-problema com justificativa .....	39
2.2 Etapas teóricas e práticas para construção do projeto.....	42
<b>3 DESENVOLVIMENTO DO PROJETO.....</b>	<b>45</b>
3.1 Configurações e testes do Arduino.....	46
3.2 Configuração da programação e testes do módulo RF.....	47
3.3 Configuração do módulo DFPlayer.....	49
3.4 Integração dos módulos Arduino, NRF e DFPlayer.....	51
3.5 Desenvolvimento da parte estrutural e agregação dos módulos.....	54
<b>CONSIDERAÇÕES FINAIS .....</b>	<b>59</b>
<b>REFERÊNCIAS.....</b>	<b>60</b>
<b>APÊNDICES.....</b>	<b>64</b>

## INTRODUÇÃO

Inovações tecnológicas buscam, em sua grande maioria, proporcionar aos usuários maior praticidade na execução de tarefas, por mais variadas que elas sejam. Seu avanço tem ajudado a inclusão social de pessoas portadoras de alguns tipos de deficiência, como a visual, auditiva e física. Muitas atividades, que antes não eram possíveis para pessoas com necessidades especiais, passam a se tornar viáveis.

Dentre essas inovações podem ser citadas: a identificação de pontos de ônibus para deficientes visuais com sinalização tátil no piso, o acesso a ônibus para deficientes físicos com veículos equipados com elevadores hidráulicos, aparelhos com formato similar ao de uma caneta capaz de reconhecer palavras escritas e transformá-las em sons para deficientes visuais.

Também se destacam os *mouses* oculares que permitem a um portador de deficiência motora o pleno uso de um computador, através do uso de eletrodos fixados em regiões próximas aos olhos do deficiente, transformando a movimentação e o piscar dos olhos em comandos para o *mouse*, também o uso de aplicações de *smartphones* para que deficientes visuais possam utilizar o aparelho através de comandos de voz, entre outros.

Por mais peculiar que seja uma deficiência física, o avanço da ciência tem se mostrado capaz de encontrar soluções facilitadoras de tarefas para grande parte desses casos e, inclusive, não apenas quando são citadas as deficiências em seres humanos, mas também em animais. Assim, com o enfoque dos aprimoramentos tecnológicos é possível alcançar melhorias para os deficientes físico-visuais, criando maior facilidade no campo da acessibilidade daqueles que necessitam.

Diante das argumentações explicitadas nasceu a ideia de criar um sistema intitulado Sistema de Identificação de Linhas de Ônibus para Deficientes Visuais, cujo objetivo é tornar o embarque dos deficientes prático e independente. Justifica-se pela fácil acessibilidade, conforto, praticidade do sistema, autonomia do deficiente visual e dispensa de pessoa auxiliadora para identificar o ônibus que se aproxima para o embarque no veículo desejado.

Para a construção do projeto utilizam-se transceptores de rádio frequência, microcontroladores e placas dedicadas.

Este trabalho é dividido em três capítulos:

Capítulo 1 – Fundamentação teórica: destaca-se a contextualização das teorias de autores renomados que dão sustentação a construção do projeto Sistema de Identificação de Linhas de Ônibus para Deficientes Visuais.

Capítulo 2 – Metodologia: descreve o caminho a percorrer para a elaboração do projeto, destacando métodos, técnicas e procedimentos.

Capítulo 3 - Desenvolvimento do projeto: descreve passo a passo a construção e o desenvolvimento do projeto, bem como figuras, tabelas para melhor entendimento.

E finalmente, as Considerações finais: são descritos os objetivos propostos e atingidos, justificativas, pontos fortes e fracos, conquistas alcançadas e possíveis sugestões para futuros trabalhos.

## 1 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo destaca-se a contextualização das teorias de autores renomados que dão sustentação a construção do projeto Sistema de Identificação de Linhas de Ônibus para Deficientes Visuais.

### 1.1 Identificação das linhas de ônibus e acessibilidade

Segundo o site da EBC Agência Brasil (2015), o meio de transporte mais comum nas grandes cidades brasileiras, mesmo nas dotadas de sistema metro-ferroviário, são os ônibus, onde 25% da população brasileira utilizam esse meio para se locomover para o trabalho ou escola.

De acordo com o Guia de Direitos (2017), a identificação das linhas de ônibus na cidade de São Paulo, fica na externa e superior ao painel do motorista, onde estão o número e o destino do ônibus. Normalmente essa identificação é feita através de um mostrador eletrônico. Essa é uma forma simples de identificar a linha que se aproxima. A Figura 1.1 ilustra um exemplo de identificação das linhas de ônibus.

Figura 1.1 – Identificação da linha dos ônibus



Fonte: [www.mobilize.org.br](http://www.mobilize.org.br), 2017

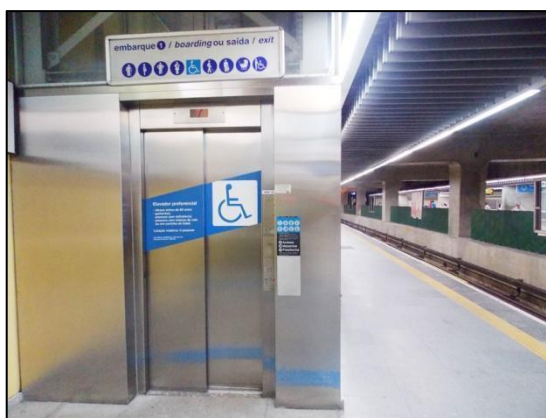
De acordo com Melo (2011), acessibilidade faz referência a existência de condições que facilitem as pessoas se deslocarem de um lugar para outro, usando equipamentos de maneira autônoma e segura, sem que seja necessário o auxílio de terceiros na realização de atividades simples como, por exemplo, o ato de pegar um ônibus.

O Decreto nº 5.296 da Presidência da República (2004) regulamentou as Leis federais nº 10.048 e 10.098 que tratam da acessibilidade para pessoas com deficiência ou mobilidade reduzida no Brasil, inclusive nos meios de transporte, com o intuito de promover a autonomia e segurança do público-alvo.

Segundo o censo do IBGE - Instituto Brasileiro de Geografia e Estatística (2010) dos cerca de 190 milhões de brasileiros, cerca de 24% das pessoas possuem algum tipo de deficiência, sendo que a visual apresenta maior ocorrência. Mais de 3,4% do total de brasileiros alegam ter a dificuldade de forma severa, 3,15% afirmam que têm dificuldade de enxergar e mais de 0,26% informam ser cegas.

De acordo com o Metrô - Companhia do Metropolitano de São Paulo (2017), pessoas com deficiência, com crianças de colo, idosos, obesos, gestantes e com restrição de mobilidade têm direito ao atendimento preferencial, na compra de bilhetes, nos bloqueios, além de usar os elevadores das estações e o assento preferencial nos trens. Existem também pisos táteis, para que deficientes visuais possam se locomover pelas estações. A Figura 1.2 mostra um elevador da estação Paraíso do Metrô de SP.

Figura 1.2 – Elevador da estação Paraíso do Metrô de SP



Fonte: [www.explorasampa.com](http://www.explorasampa.com), 2017

A CPTM - Companhia Paulista de Trens Metropolitanos (2017), informa que a empresa trabalha para adequar suas dependências às normas vigentes de acessibilidade. Das 92 estações, 48 são adaptadas, possuindo passagens em nível, telefones, sanitários e rotas acessíveis, além de comunicação visual e sonora, entre outros meios para facilitar a vida de quem depende de ajuda, conforme ilustra a Figura 1.3.

Figura 1.3 – Plataforma da estação Itapevi da CPTM



Fonte: [www.explorasampa.com](http://www.explorasampa.com), 2017

Segundo a SPTrans - São Paulo Transportes (2017), que é responsável pela gestão do sistema de transporte público de ônibus na cidade de São Paulo, para maior comodidade no embarque e desembarque sem quaisquer obstáculos aos passageiros, principalmente aqueles com mobilidade reduzida, a empresa prevê a utilização de ônibus de entrada baixa ou piso totalmente baixo, nos grandes corredores e em vias exclusivas. Além de se conseguir a eliminação de degraus, o projeto da empresa ainda permite a instalação de rampas para facilitar o acesso de usuários cadeirantes, como mostra a Figura 1.4.

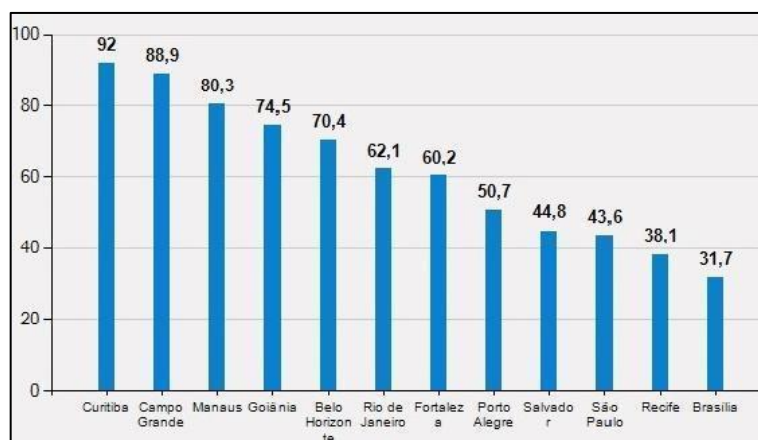
Figura 1.4 – Ônibus com rampa de acessibilidade



Fonte: [www.vidamaislivre.com.br](http://www.vidamaislivre.com.br), 2017

Mobilize (2017) enfatiza que dados fornecidos por prefeituras de alguns estados brasileiros, mostram que os recursos especiais no embarque de coletivos são precários. As principais capitais do país não possuem quantidade de ônibus acessíveis para pessoas com deficiência. Muitos coletivos não são adaptados para total acessibilidade nesse meio de transporte. A Figura 1.5 ilustra a porcentagem de ônibus em algumas capitais que são acessíveis a deficientes.

Figura 1.5 – Porcentagem de ônibus acessíveis em algumas capitais



Fonte: [www.mobilize.org.br](http://www.mobilize.org.br), 2017

## 1.2 Comunicação paralela e serial

Albuquerque e Alexandria (2009) enfatiza que na comunicação paralela os *bits* que compõem uma palavra de dados são transmitidos ao longo de um conjunto de vias, sendo cada uma dessas vias designada para um *bit* específico. Já na transmissão serial, a quantidade de vias pode ser bastante reduzida, uma vez que os dados a serem transferidos podem ser antes convertidos numa sequência serial de *bits*.

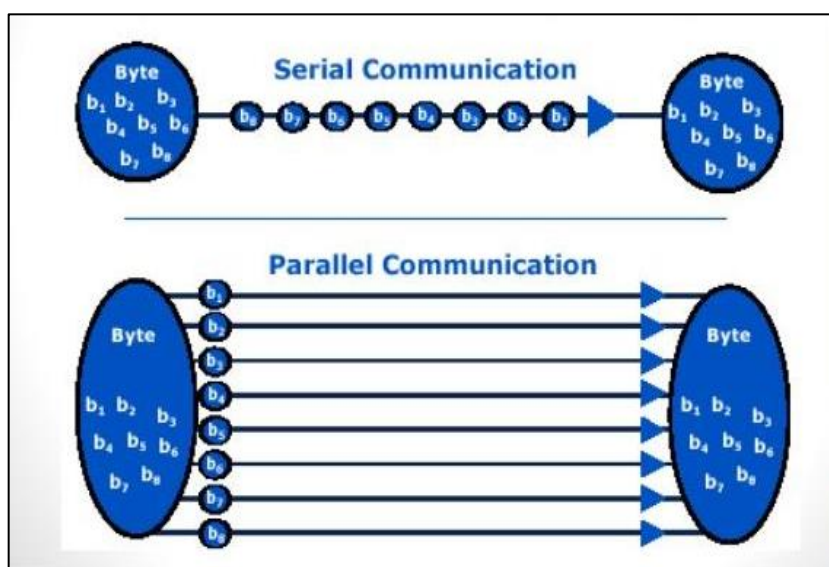
Em casos de necessidade de comunicação entre dois sistemas localizados a curta distância, a transferência de dados através da comunicação paralela se faz uma boa alternativa, pois envia apenas um *bit* por fio, ao invés de uma sequência de *bits* na mesma via. Sendo assim, todos os *bits* de um dado chegam ao receptor ao mesmo tempo, aumentando assim, a eficácia da conexão quanto à velocidade de transferência.

No entanto, à medida que a distância entre os elementos da rede que precisam se comunicar aumentam, em dezenas de metros, por exemplo, o custo de



implementação de uma via para cada *bit* de dado cresce consideravelmente. O que passa a viabilizar o uso da comunicação serial, que apresenta melhores resultados, não apenas com relação ao custo de implementação, mas também acerca da imunidade a ruídos. Um exemplo de comparação entre os meios físicos da comunicação serial (*Serial Communication*) e da comunicação paralela (*Parallel Communication*), bem como o modo como os *bits* trafegam por meio destes meios físicos, em cada tipo de transmissão de dados, conforme ilustra a Figura 1.6.

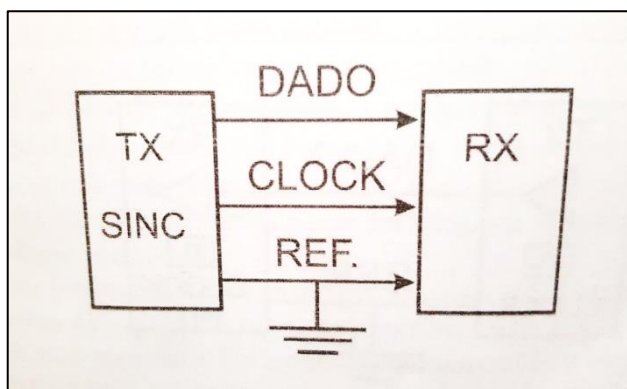
Figura 1.6 – Comparação entre comunicação serial e comunicação paralela



Fonte: [www.pt.slideshare.net](http://www.pt.slideshare.net), 2014

Comunicação serial síncrona: geralmente possui um dos sistemas conectados que gera um *clock* que deve ser seguido pelos demais sistemas. Gerando, dessa forma um sincronismo entre os elementos da comunicação. A Figura 1.7 exemplifica como são transferidos dados nesse tipo de comunicação.

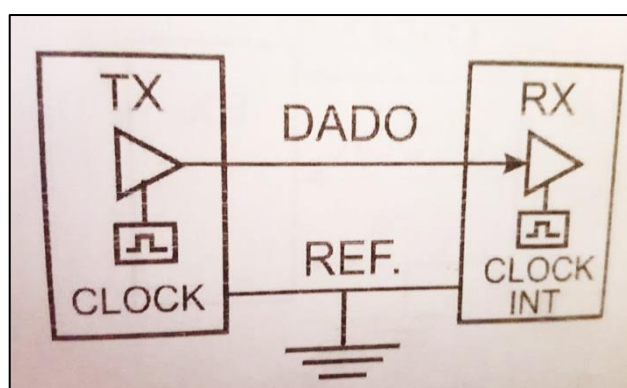
Figura 1.7 – Comunicação serial síncrona em diagrama



Fonte: ALBUQUERQUE E ALEXANDRIA, 2009, p. 49

Comunicação serial assíncrona: necessita de geradores de *clock* internos e esses devem ser programados para estar na mesma taxa de transmissão. Para a transferência de dados não é necessário um sinal de sincronismo externo (*clock*). No entanto, cada um dos sistemas da comunicação assíncrona deve possuir seu próprio *clock*. A Figura 1.8 mostra, através de um diagrama, como funciona esse tipo de transmissão de dados.

Figura 1.8 – Comunicação serial assíncrona em diagrama

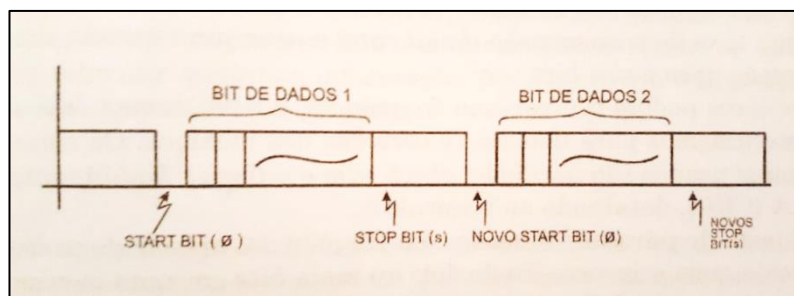


Fonte: ALBUQUERQUE E ALEXANDRIA, 2009, p. 50

Nesse tipo de transmissão, cada conjunto de *bits* enviado (chamado de *byte*) possui em seu início um sinal de *start* e, em seu final, um sinal de *stop*. Esses sinais têm o nome de *start bit* e *stop bit*, respectivamente. Portanto, sempre que o

transmissor possui dados para enviar, ele transmite um *start bit*, depois os *bits* de dados seguidos pelo *stop bit*, conforme ilustra a Figura 1.9.

Figura 1.9 – Transmissão de *bits* em comunicação assíncrona



Fonte: ALBUQUERQUE E ALEXANDRIA, 2009, p .51

### 1.3 Protocolo de comunicação

Dantas (2002) destaca que, para compreender os princípios de um protocolo de comunicação é necessário um entendimento básico acerca de um modelo de transmissão de dados simples, bem como dos itens que o compõem. A Figura 1.10 mostra um diagrama genérico de comunicação.

Figura 1.10 – Modelo genérico de comunicação



Fonte: DANTAS, 2002, p. 8

Pela Figura 1.10 observa-se que a fonte, em um circuito de comunicação, é responsável pela geração da informação a ser transmitida. O transmissor executa a

conversão dos dados emitidos pela fonte. Ou seja, é no transmissor que o sinal enviado pela fonte é codificado/decodificado a fim de ser transformado em uma informação capaz de ser enviada à rede. Pela rede de comunicação o sinal chega até o receptor que, por sua vez, exerce o papel de receber o sinal transmitido através dela e tratar dessa informação a fim de enviá-la ao destinatário, que é o elemento para o qual a informação da fonte é endereçada, assim que estabelecida a comunicação.

Dantas (2002) destaca que em uma rede de comunicação, os dados são transferidos da fonte ao destinatário através de sinais elétricos, os quais podem ser analógicos ou digitais. O sinal analógico se dá por uma variação contínua de onda eletromagnética enquanto o digital se faz, resumidamente, por uma sequência de pulsos de tensão elétrica.

Um protocolo de comunicação pode ser definido como um conjunto de regras na transmissão de um sinal digital entre duas estações. Como principais elementos de um protocolo de comunicação podem ser citados: o conjunto de símbolos (denominados caracteres), o conjunto de regras determinantes da sequência e tempo das mensagens transferidas e também os procedimentos que auxiliam na detecção/tratamento de erros.

Em outras palavras, Dantas (2002) ressalta que um protocolo de comunicação deve conter aspectos como a sintaxe, a semântica e a temporização. A sintaxe orienta como é a codificação dos dados, o formato deles e também quais níveis de sinal devem ou não ser considerados pelo protocolo. A semântica é parte de controle da informação, onde são coordenados e tratados os erros. E a parte do temporizador, adequa os tempos de transferência da informação e as sequências de mensagens permitidas.

Albuquerque e Alexandria (2009) definem protocolo de comunicação como o

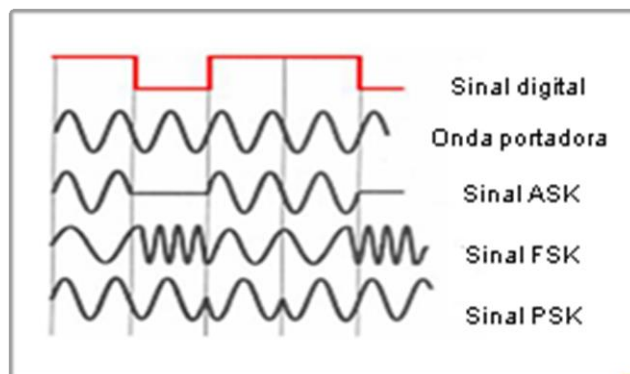
conjunto de convenções e procedimentos que tornam regulamentada a transmissão de dados entre diferentes equipamentos, de forma completa ou em alguns de seus aspectos.

### 1.3.1 Codificação e decodificação

Mendes (2015) afirma que a codificação de sinais é necessária para que sinais analógicos ou digitais emitidos por uma fonte de dados possam se adequar à transmissão em uma rede. O elemento que faz essa codificação tem o nome de codificador. Antes de chegar ao seu destino, o sinal deve passar por um elemento basicamente igual, mas que faz a operação reversa, codificando/decodificando a informação entregando, desta forma, o mesmo sinal de forma restaurada. Em resumo, a codificação e decodificação de um sinal têm o objetivo de permitir o acesso aos dados de maneira mais rápida e confiável.

Machado (2013) ressalta que a codificação e decodificação de um sinal é feita para que seja possível ocupar menor largura de banda nas transmissões, para melhor capacidade do sistema com relação à detecção de erros e também para maior imunidade a ruídos. Os três modelos mais comuns de codificação de sinais digitais em ondas analógicas se apresentam na Figura 1.11.

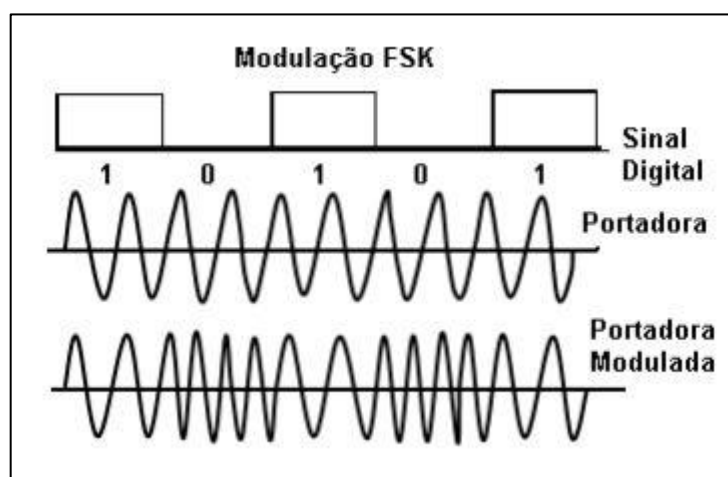
Figura 1.11–Modelos de codificação de sinais digitais em ondas analógicas



### 1.3.2 Modulação GFSK

Conforme Baptista (2008), a modulação GFSK (*Gaussian frequency shift keying*) é, de certa forma, um aprimoramento da modulação FSK (*frequency shift keying*) que, por sua vez, consiste na variação da frequência de uma onda portadora de sinal, em função de um sinal digital modulante. A Figura 1.12 demonstra o funcionamento de uma codificação de sinal FSK.

Figura 1.12 - Modulação FSK



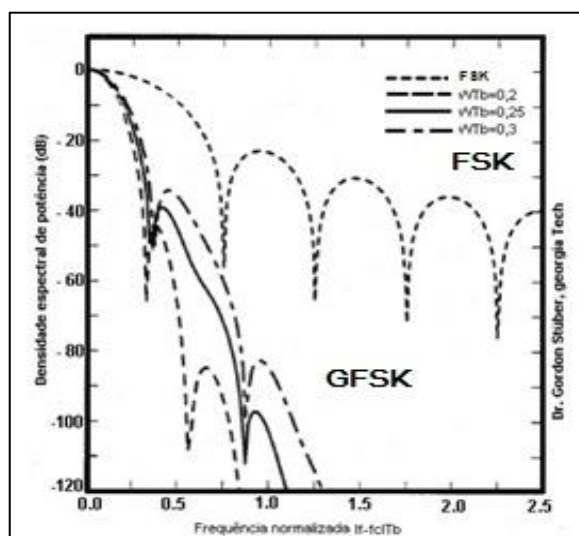
Fonte: BAPTISTA, 2008,p. 33

Em outras palavras, a modulação FSK mantém a amplitude e a frequência da onda do sinal transmitido sempre as mesmas enquanto a informação digital permanece em zero "0". Assim que o sinal digital assume valor um "1", a frequência da onda portadora muda para uma forma pré-definida. A forma dessa onda referente ao sinal lógico "1" será a mesma sempre que o dado digital for desse valor.

Na modulação GFSK, o processo de alteração do sinal se dá exatamente da mesma forma que o FSK, mas acrescenta-se um processo antes de a forma de onda ser alterada. Nesse tipo de modulação, a onda passa primeiramente por um filtro, chamado filtro Gaussiano, onde sua amplitude e frequência são reduzidas,

mas de maneira proporcional, de modo que não se altera a característica do sinal. Trata-se apenas de um modo de compactar taxas de dados dentro de faixas de RF com largura limitada. A Figura 1.13 demonstra a aplicação de um filtro Gaussiano em uma onda portadora de sinal analógico.

Figura 1.13 - Filtro Gaussiano aplicado em um sinal analógico



Fonte: BAPTISTA, p. 19, 2008

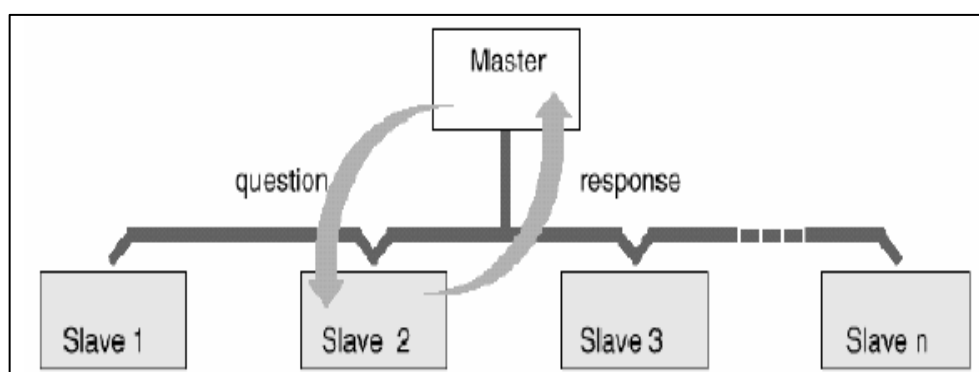
### 1.3.3 Acesso ao meio mestre-escravo

De acordo com Albuquerque e Alexandria (2009), o modelo de comunicação mestre-escravo se trata de um sistema onde o mestre é o dispositivo responsável por controlar a rede e é também por onde todos os dados do sistema passam ou são concentrados. Além disso, o mestre pode funcionar também como interface para sistemas supervisórios. Nesse modelo, os dispositivos escravos são elementos periféricos que têm a função de receber uma requisição do mestre e executá-la. Os escravos não podem trocar dados entre si, mas sim enviar dados ao mestre, que se torna responsável pela interpretação das mensagens e tomada de decisões.

Um escravo pode também efetuar processamento de sinais, realizar

medições e manipular eventos de modo pré-determinado. O mestre pode solicitar dados de um escravo apenas e aguardar por sua resposta, em um modelo chamado de requisição/resposta, ou então enviar uma mensagem comum a todos os elementos escravos da comunicação, em um modo chamado de difusão. A Figura 1.14 ilustra um modelo de comunicação mestre-escravo.

Figura 1.14 - Modelo de comunicação mestre-escravo



Fonte: [www.instrumentacaoecontrole.blogspot.com.br](http://www.instrumentacaoecontrole.blogspot.com.br), 2014

Segundo Zanco (2011), em uma comunicação mestre-escravo o elemento mestre é sempre o responsável pela geração do sinal de sincronismo *clock*, e os dispositivos escravos são elementos periféricos, como a memória RAM/ROM, driver de LCD, conversores A/D (analogico/digital) e D/A(digital/analogico), processador de áudio/vídeo.

#### 1.3.4 Comunicação *half-duplex* e *full-duplex*

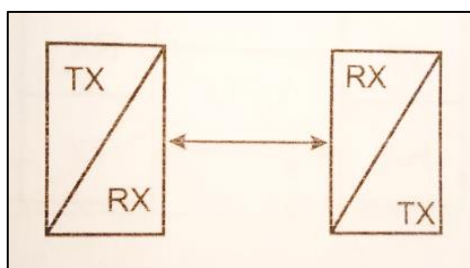
Segundo Albuquerque e Alexandria (2009), tratando-se do sentido de envio de dados e também da ordem de qual elemento transfere informações em cada momento da comunicação, os modos mais usuais de transmissão são o *half-duplex* e o *full-duplex*.

Na comunicação *half-duplex*, também chamada por *semi-duplex*, têm-se elementos que transmitem e recebem dados, no entanto, essas operações



acontecem em momentos distintos. Como exemplo, cita-se o aparelho *walkie-talkie*, onde os dois terminais podem enviar e receber, no entanto, enquanto um fala, o outro fica limitado apenas a receber até que a fala seja concluída e os papéis puderem se inverter. A Figura 1.15 mostra, em um diagrama de blocos, o funcionamento de uma comunicação *half-duplex*.

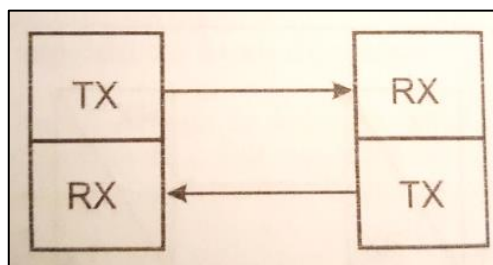
Figura 1.15 – Comunicação *Half-duplex*



Fonte: ALBUQUERQUE e ALEXANDRIA, 2009, p. 51

Ainda segundo Albuquerque e Alexandria (2009), a comunicação *full-duplex*, ou apenas *duplex*, é um modo de transmissão onde os sistemas podem enviar e receber dados simultaneamente. Esse modelo de comunicação permite maior velocidade nas operações em relação à transferência de dados *half-duplex*. A Figura 1.16 demonstra a arquitetura de um sistema *full-duplex*.

Figura 1.16 – Comunicação *full-duplex*

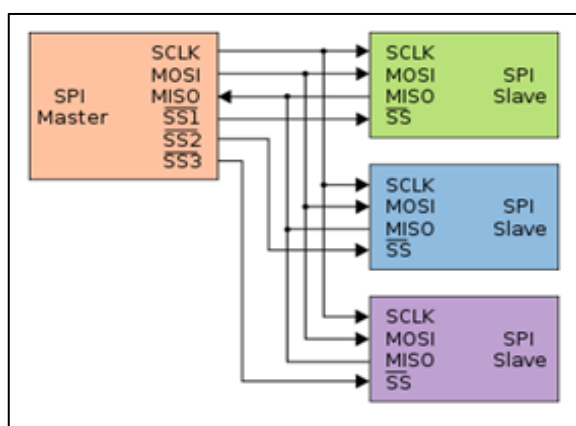


Fonte: ALBUQUERQUE e ALEXANDRIA, 2009, p. 54

### 1.3.5 Protocolo de comunicação SPI

Embarcados (2014), relaciona como principal característica do protocolo SPI a transmissão de dados que é feita em ambos os sentidos da comunicação. Ou seja, se trata de uma interface *full duplex*. Além disso, o SPI é também um protocolo do tipo *master-slave*, onde o gerador do sinal de sincronismo é definido como mestre da comunicação e os dispositivos que utilizam este sinal são chamados de escravos, podendo haver em uma rede um ou mais escravos. A pinagem dos componentes usados em transmissões com o protocolo SPI é dada como mostra a Figura 1.17.

Figura 1.17 – Exemplo de interface SPI



Fonte: [www.embarcados.com.br](http://www.embarcados.com.br), 2014

A Figura 1.17 mostra o terminal SCLK (*serial clock*) que se trata de um gerador de *clock*, por onde é determinado o momento no qual cada escravo executa uma ação, tendo em vista que se trata também de uma interface síncrona, onde todos os escravos estão em sincronismo com o mesmo sinal de *clock*. Os terminais MOSI (*Master Out Slave*) e MISO (*Master In Slave Out*) indicam onde cada mestre e cada escravo envia/recebe as informações. E o SS (*Slave Select*) trata-se da seleção feita pelo mestre de qual escravo recebe/envia dados em determinado momento da transmissão. Cada escravo possui apenas um terminal SS, mas no elemento mestre da comunicação, a quantidade de terminais SS é a mesma que a de escravos na rede, sendo cada um deles diferenciado, normalmente, por um

número indicativo depois da descrição padrão, como SS1, SS2 e SS3.

A tecnologia do protocolo SPI traz consigo a possibilidade de uma transferência de dados de grande velocidade, em comparação a outros protocolos de comunicação serial. Fazendo uso, por exemplo, do controlador AT45BD0100D, a velocidade de transferência do SPI pode alcançar até 66 Mbps, conforme ilustra a Figura 1.18.

Figura 1.18 – Comparativo entre diferentes padrões de comunicação serial

TECNOLOGIA	BARRAMENTO DE COMUNICAÇÃO	TAXA MÁXIMA	FLUXO DE DADOS
UART (RS232)	2	115.200 bps	<i>Half Duplex ou Full Duplex</i>
SPI	3 + nº de escravos	66 Mbps	<i>Full Duplex</i>
I2C	2	1 Mbps	<i>Half Duplex</i>

Fonte: [www.embarcados.com.br](http://www.embarcados.com.br), 2014

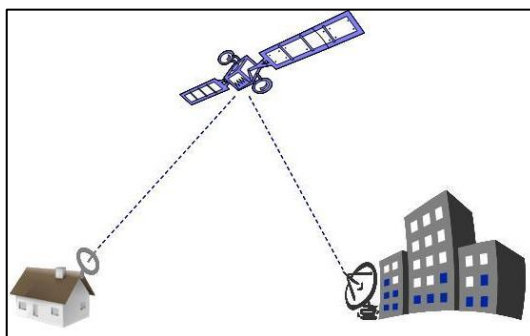
Segundo a Microchip (2005), o SPI é utilizado quando poucas linhas de I/O (entrada/saída) estão disponíveis mas a comunicação entre dois ou mais dispositivos deve ser mais rápida e de fácil implementação.

Uma vez que o SPI faz uso do pulso de *clock* para o mestre iniciar ou encerrar a transferência de dados para a rede, as informações dos escravos são captadas apenas em sincronismo com esse pulso de *clock*, o que torna o protocolo SPI ideal em situações onde o microcontrolador não sabe exatamente em qual momento recebe um sinal de *clock*.

#### 1.4 Comunicação sem fio via radiofrequência (RF)

Conforme Freitas (2011), os meios de transmissão sem fio se justificam quando a distância entre o emissor e receptor é muito grande, como nas comunicações via satélite. Outra vantagem é a redução dos números de fios e cabos nas construções e na praticidade de implementar outros serviços, sem a necessidade de muitas mudanças estruturais. A Figura 1.19 ilustra a comunicação sem fio, via satélite.

Figura 1.19 –Comunicação sem fio via satélite



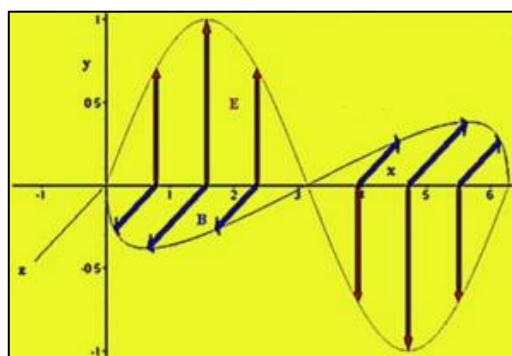
Fonte: [www.teleco.com.br](http://www.teleco.com.br), 2017

Albuquerque e Alexandria (2009) destaca que o meio de transmissão é o caminho que os dados trafegam entre as estações de trabalho em uma rede digital. Existem diversas tecnologias, tais como o infravermelho, transmissão via satélite, microondas, raios laser, ondas de rádio, entre outras.

As ondas de rádio são fáceis de serem geradas, percorrem longas distâncias e atravessam paredes de grandes construções. Devido a essas características elas são amplamente utilizadas na comunicação sem fio, em ambientes fechados ou abertos. As ondas de rádio, dependendo do tipo de antena, também são omnidirecionais, ou seja, percorrem todas as direções a partir da origem. Assim emissor e receptor não precisam ficar alinhados para que haja comunicação.

Queiroz Filho, Cugnasca e Rodrigues (2009) apontam que quando o elemento possui carga elétrica, cria-se em sua volta um campo elétrico. Se este campo se encontrar em movimento surge um campo magnético ao seu redor e vice-versa, assim sendo, o campo elétrico e o campo magnético estão sempre relacionados e sua combinação forma o campo eletromagnético. O sistema de radiofrequência é baseado na propriedade dos campos eletromagnéticos de propagar perturbações sob a forma de ondas eletromagnéticas. O campo elétrico (E) e o campo magnético (B) que oscilam de forma perpendicular, formando a onda eletromagnética, conforme ilustra a Figura 1.20.

Figura 1.20–Onda Eletromagnética

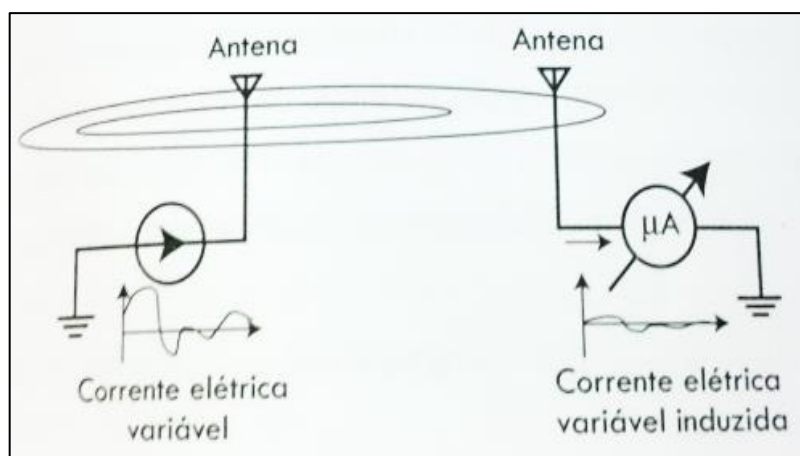


Fonte: [www.uol.com.br](http://www.uol.com.br), 2017

A velocidade com que as ondas eletromagnéticas se propagam não depende de sua frequência de oscilação, mas sim do meio em que elas se propagam. A frequência das ondas de rádio é chamada de radiofrequência (RF), e é usada para comunicação tanto de voz como de dados.

Um elemento importante na transmissão e recepção de voz ou dados é a antena, cujo papel é maximizar a irradiação das ondas eletromagnéticas. Neste dispositivo circula uma corrente elétrica variável forçada com o objetivo de produzir oscilações no campo eletromagnético em sua vizinhança. A Figura 1.21 ilustra a transmissão e recepção de ondas eletromagnéticas por antenas.

Figura 1.21 – Transmissão e recepção de ondas eletromagnéticas por antenas



Fonte: [www.wi2be.com.br](http://www.wi2be.com.br), 2017

A atenuação, ou enfraquecimento do sinal de rádio depende de diversos fatores como a geometria das antenas, o modo de propagação da onda, a existência de obstáculos, reflexões etc. Esses e outros fatores influenciam na distância máxima entre transmissor e receptor.

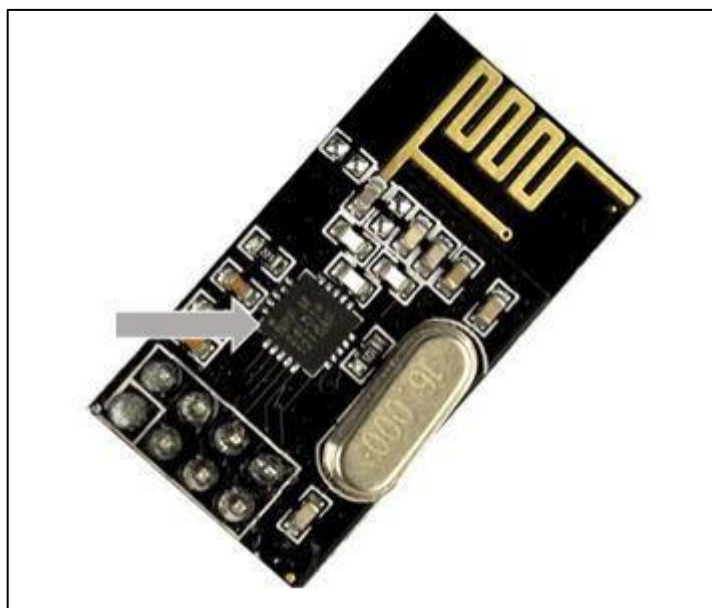
Transmissão de sinais digitais: viaradiofrequência tem boas vantagens, devido à flexibilidade de trabalho com a informação codificada, oferece segurança, privacidade na comunicação, permite o controle mais eficiente da comunicação e a simplicidade de representação da informação. A informação original só é recuperada pelo receptor, caso a codificação empregada pelo transmissor seja conhecida para que a informação original chegue de forma íntegra (QUEIROZ FILHO, CUGNASCA E RODRIGUES, 2009).

## 1.5 Módulo de radiofrequência NRF24L01

De acordo com a NordicSemiconductors(2007), o NRF24L01 se trata de um transceptor de dados. Ou seja, o mesmo componente envia e recebe informações,

não sendo necessário um módulo para envio e um para recebimento de dados em cada ponta da transmissão. Este componente se trata de um CI de 20 pinos, com tamanho físico de 4 x 4 mm, mas que é disponibilizado para venda em uma placa eletrônica dedicada de 33 x 15 mm. A Figura 1.22 ilustra a placa dedicada do NRF2401, com indicação do circuito integrado de 20 pinos.

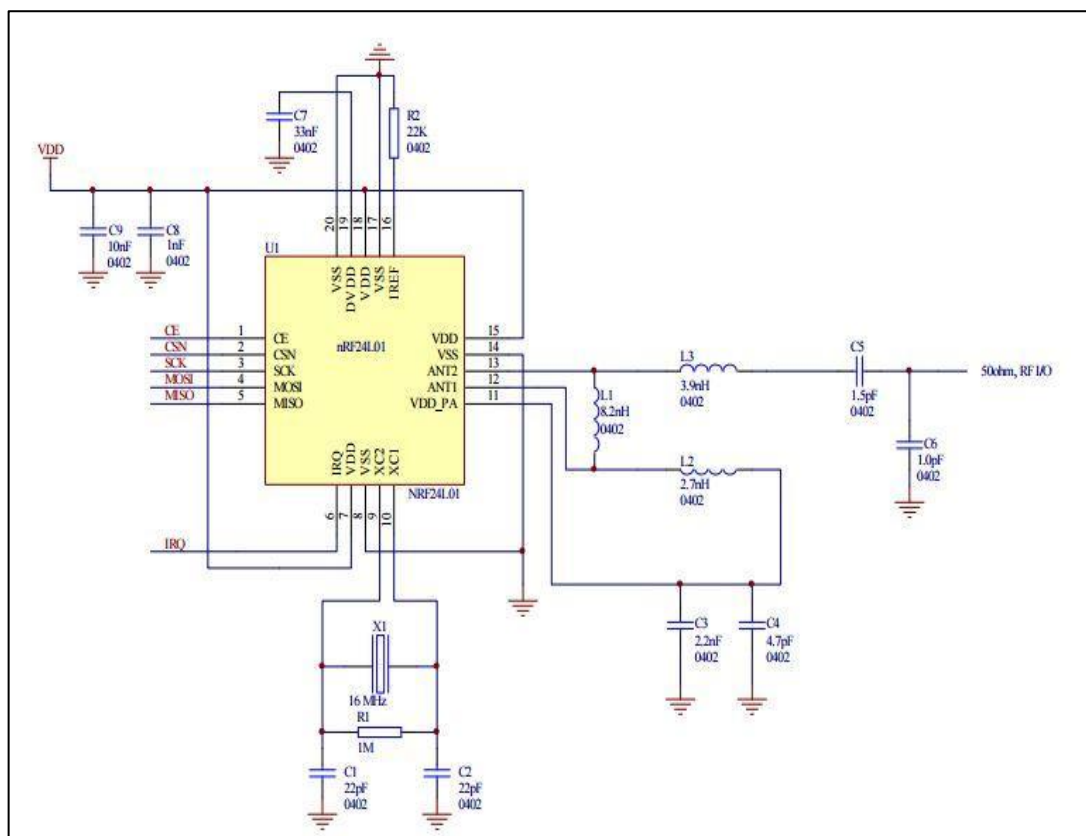
Figura 1.22– Placadedicadado módulo NRF24L01



Fonte: [www.hallard.me](http://www.hallard.me), 2013

A placa possui cristal e antena integrados, além de resistores, capacitores e indutores previamente dimensionados. O circuito eletrônico se ilustra na Figura 1.23.

Figura 1.23 – Circuito eletrônico da placa dedicada NRF24L01



Fonte: [www.nordicsemi.com](http://www.nordicsemi.com), 2017

Segundo o site Arduino (2015), o circuito integrado de 20 pinos do módulo NRF24L1 possui 8 desses terminais abertos para ligação externa, sendo dois deles para alimentação (+VCC), dois para GND, um terminal para *clock* externo, chamado de SCK, um pino CE (*Chip Enable TX/RX*) usado para habilitação do módulo em modo de envio/recebimento de dados, um pino CSN (*Chip Select*) utilizado para seleção do escravo com qual o mestre da conexão deseja se “comunicar” em dado momento, um pino chamado IRQ (*InterruptRequestSelect*) utilizado para entrada de sinal de interrupção durante a transferência de dados, além de dois terminais para entrada/saída de dados, MOSI (*Master Out Slave In*) e MISO (*Master In Slave Out*) por se tratar de um componente que faz uso do protocolo SPI para comunicação. A Figura 1.24 mostra a disposição física dos terminais em aberto contidos na placa do módulo NRF24L01.



Figura 1.24 – Disposição dos terminais externos módulo NRF24L01

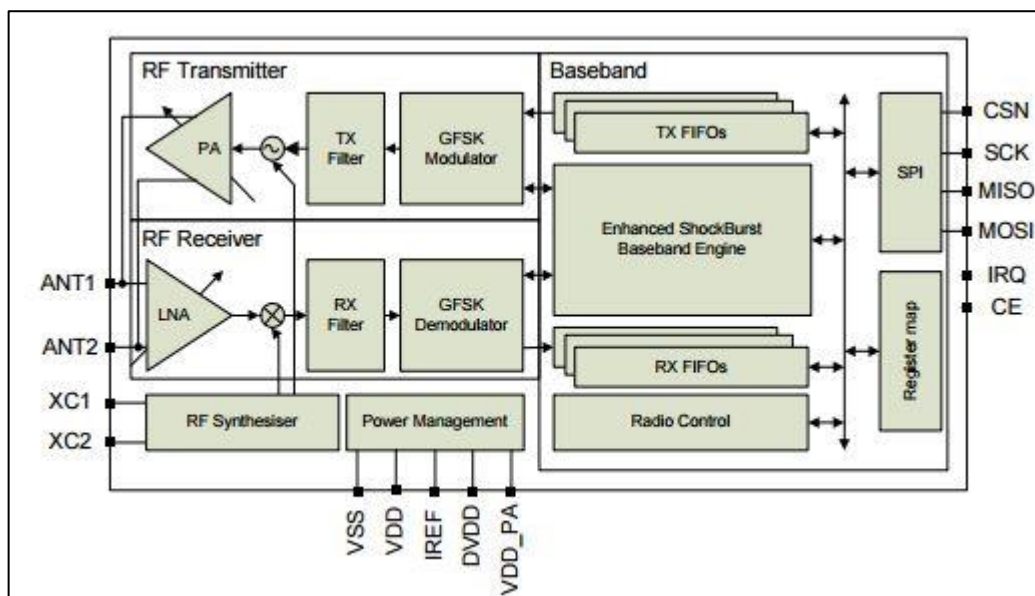


Fonte: [www.arduinoocia.com.br](http://www.arduinoocia.com.br), 2015

A velocidade de transmissão do módulo NRF24L01 é de até 2Mbps e o módulo é capaz de enviar e receber informações fazendo uso de correntes inferiores a 14uA. A frequência usada por este módulo RF é de 2,4GHz e a alimentação dele pode variar entre 1,9 a 3,6 V. O NRF24L01 garante uma autonomia que vai de meses a anos, fazendo uso de pilhas AA/AAA.

Este módulo RF é configurado e operado através da interface SPI. Através desse protocolo o mapa de registradores se faz possível. Esse mapa contém todos os registradores do NRF24L01. Há nele também, internamente, um protocolo embutido chamado *Enhanced ShockBurst™* o qual é baseado na comunicação por pacotes e também utiliza do sistema FIFO (*first in first out*), o que significa que a primeira informação que chega ao módulo é também a primeira a sair, garantindo assim uma ordem lógica e, conseqüentemente, um bom fluxo de dados entre ele e o microcontrolador. A Figura 1.25 exemplifica, em um diagrama de blocos, as partes que compõem o módulo NRF24L01 internamente.

Figura 1.25 – Diagrama de Blocos Interno Módulo NRF24L01



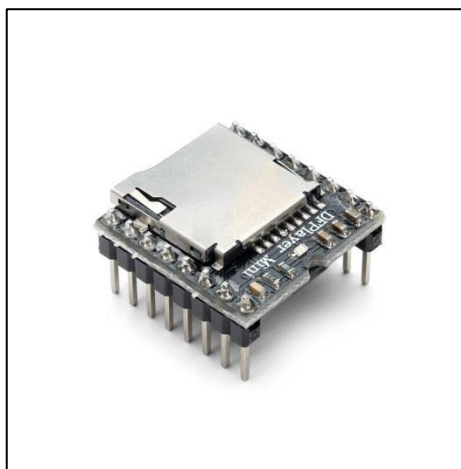
Fonte: [www.nordicsemi.com](http://www.nordicsemi.com), 2017

## 1.6 Sistema de gravação de voz digital –DFPlayer mini

DFRobot (2018) afirma que o DFPlayer mini consiste em um módulo de som de 16 pinos designado a adicionar áudios e de saída simplificada a altos falantes. O componente funciona com alimentação de 3,3 V a 5 V, com cartões microSD de tamanho máximo de 32 GB, sendo capaz de armazenar até 100 pastas, com 255 músicas cada.

Trata-se de um módulo compacto, com dimensões de 21 x 20 x 4,5 mm, de fácil utilização e integração com placas e microcontroladores, podendo ser controlado via interface serial ou através de botões conectados diretamente ao módulo. Suporta os formatos MP3, WAV e WMA e apresenta 30 níveis de volume ajustáveis e 6 níveis de equalização também ajustáveis. A Figura 1.26 mostra o módulo DFPlayer mini.

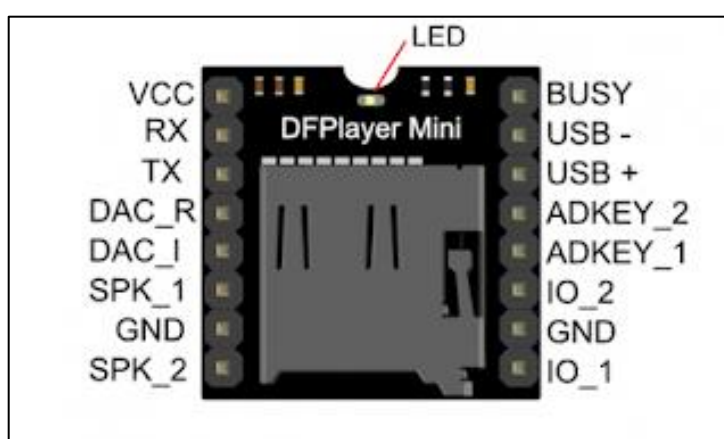
Figura 1.26–DFPlayer mini



Fonte: [www.dfrobot.com](http://www.dfrobot.com), 2018

A pinagem do módulo resalta os pinos RX/TX, que são as conexões para a comunicação serial e controle utilizando um microcontrolador, pinos ADKEY para utilização de um teclado analógico, pinos SPK\_1 e SPK\_2 como saída para alto falante, pinos DAC para ligar o módulo a um amplificador e os pinos VCC e GND que são a alimentação do módulo, conforme ilustra a Figura 1.27.

Figura 1.27 –Pinagem do módulo DFPlayer mini

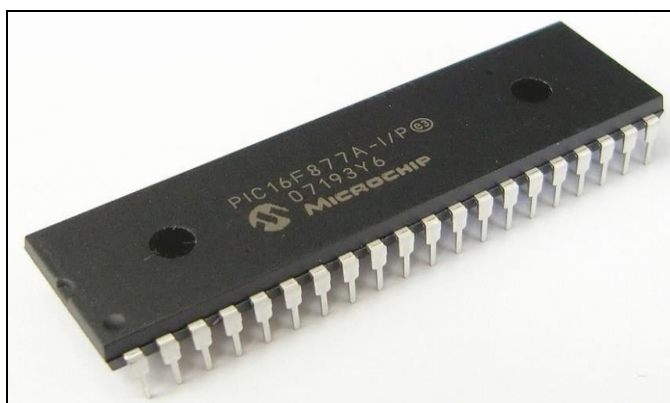


Fonte: [www.dfrobot.com](http://www.dfrobot.com), 2018

## 1.7 Microcontroladores e linguagem de programação

De acordo com Souza (2010), o microcontrolador é um componente eletrônico programável que faz controle de outros componentes periféricos, tais como botões, relés, sensores, entre outros. São chamados de controles lógicos, pois suas ações lógicas se baseiam no estado dos seus periféricos. A Figura 1.28 ilustra um microcontrolador.

Figura 1.28 – Microcontrolador

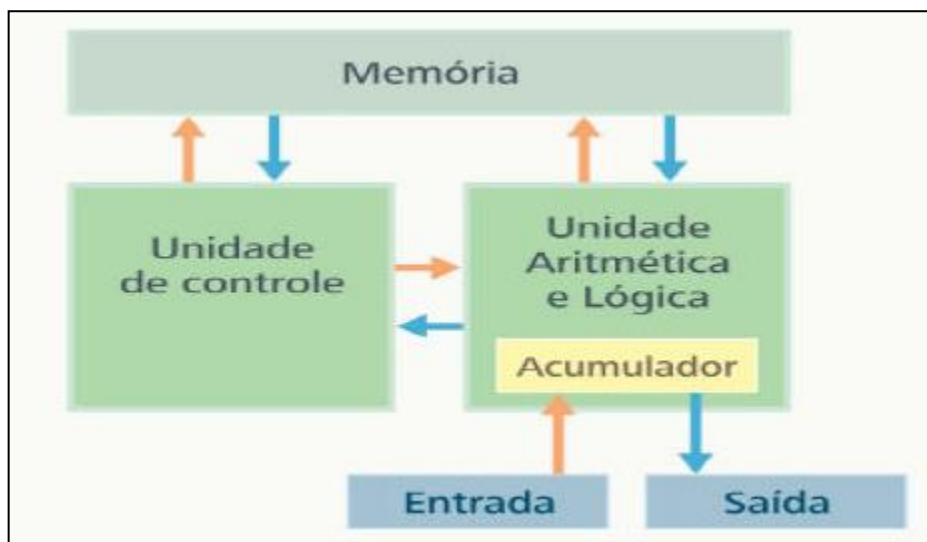


Fonte: [www.solucoesindustriais.com.br](http://www.solucoesindustriais.com.br), 2017

Seu programa é elaborado externamente e depois gravado internamente no *chip*. A inteligência do microcontrolador está associado à ULA (Unidade Lógica Aritmética), pois nessa unidade são realizadas todas as operações matemáticas e lógicas.

Presente em uma única pastilha de silício, o microcontrolador tem todos os componentes para obter o controle de um processo, como memória de programa, memória de dados, portas de entrada/saída, *timers*, contadores e comunicação serial. Essas características são os diferenciais quando comparado com um microprocessador, pois este não possui todos os recursos em uma só pastilha. A Figura 1.29 ilustra o diagrama de blocos de um microcontrolador

Figura 1.29 – Diagrama de blocos de um microcontrolador



Fonte: [www.ifmg.com](http://www.ifmg.com), 2017

Haddad (2016) argumenta que Arduino é uma plataforma de eletrônica de código aberto baseada em *softwares* e *hardwares* fáceis de usar. Essas placas são capazes de ler entradas, como uma luz em um sensor, e transformá-la em uma saída, como ligar um *led* ou um motor. É o operador que diz ao Arduino o que fazer através de um conjunto de instruções para o microcontrolador da placa. Uma maneira de ampliar e expandir o *hardware* do Arduino principal, são os *shields*, que são placas eletrônicas que adicionam funções à placa principal. Sendo assim, o Arduino se torna uma ferramenta fácil para a prototipagem rápida, voltada para estudantes em eletrônica e programação.

Pereira (2010) relata que os primeiros componentes programáveis, como os microcontroladores, eram programados com códigos chamados de códigos de máquina, ou seja, dígitos binários inseridos por meio de um dispositivo de entrada para serem executados pela máquina.

Devido a necessidade de programação de sistemas, surgiu a linguagem em *Assembly*, que consiste em uma forma alternativa de representação dos códigos de máquina, usando abreviações de termos que descrevem a operação realizada. Possui alguns aspectos negativos como a complexidade e a falta de compatibilidade entre diferentes sistemas.

Com o intuito de facilitar para o programador desenvolver um projeto, surgiram outras linguagens de programação, como a linguagem C. Uma linguagem genérica desenvolvida para ser eficiente, rápida, estruturada e lógica.

Bolton (2010) argumenta que a linguagem de programação C é uma linguagem padronizada e também é muito utilizada na programação de microcontroladores. Ela é mais simples e fácil de trabalhar comparando-se com a linguagem de programação *assembly*. A Figura 1.30 ilustra as linguagens C e *assembly*.

Figura 1.30– Linguagem C e *assembly*

<b>C vs Assembly</b>	
<pre>main() {     int val1=10000h;     int val2=40000h;     int val3=20000h;     int finalVal;      finalVal = val1               + val2 - val3; }</pre>	<pre>.data val1 DWORD 10000h val2 DWORD 40000h val3 DWORD 20000h finalVal DWORD ? .code main PROC     mov eax,val1     add eax,val2     sub eax,val3     mov finalVal,eax     call DumpRegs     exit main ENDP</pre>

Para passar as linguagens de programação C ou *assembly* (cuja linguagem se denomina de alto nível), para a linguagem de máquina (cuja linguagem se denomina de baixo nível), há a necessidade de um compilador que transforma os códigos de cada linguagem para números binários, os quais a máquina processa (entende).

## 2 METODOLOGIA

Neste capítulo encontram-se a os procedimentos de toda a trajetória para a execução do projeto intitulado Sistema de Identificação de Linhas de Ônibus para Deficientes Visuais. Trata-se de uma pesquisa aplicada, que é desenvolvida nas dependências da FATEC SB Campo e nas residências dos integrantes do grupo.

Dentre vários autores que versam sobre metodologia, Severino (2011) aponta que ela é fundamental para despertar nos pesquisadores a necessidade de compreensão dos fatos. A preparação metodológica se baseia em métodos e técnicas desde sua elaboração até o resultado final, que compreende em: determinação do tema-problema e justificativa, levantamento da bibliografia referente ao tema, leitura dessa bibliografia após a seleção, construção lógica do trabalho e redação do texto.

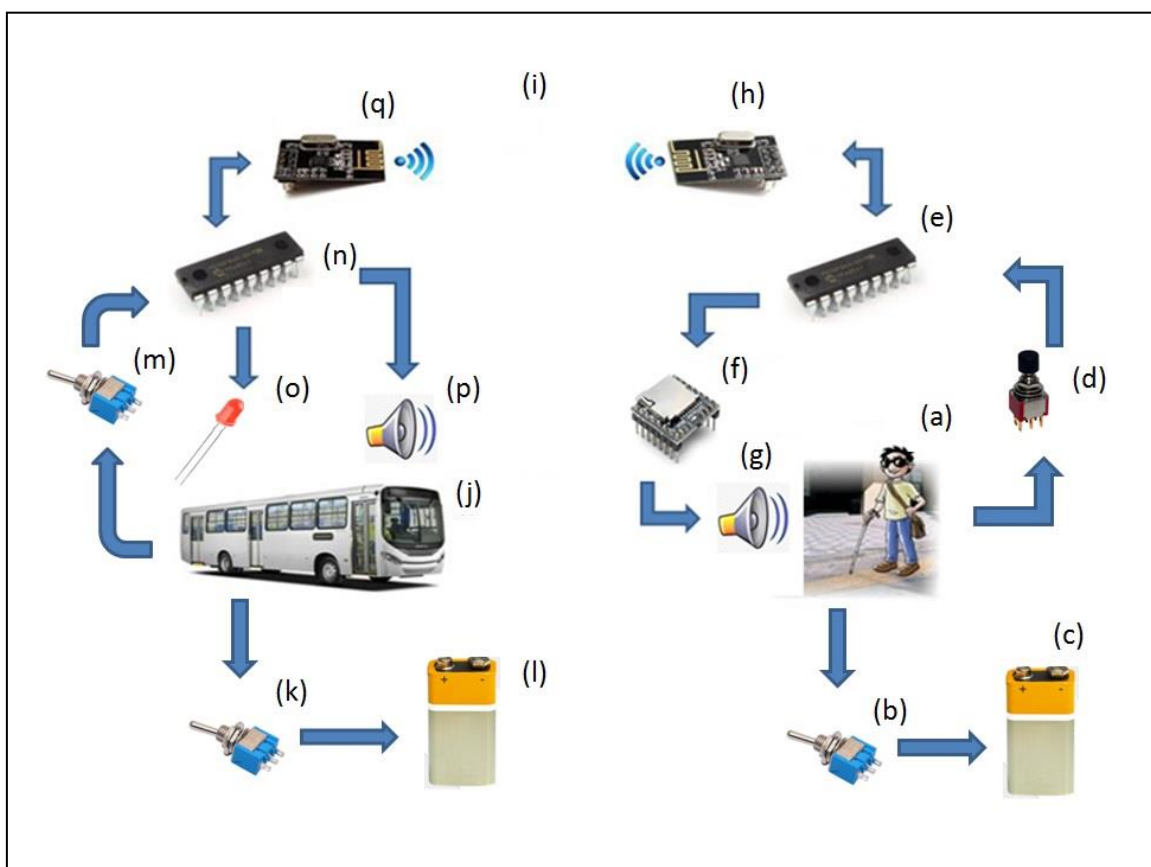
A redação do trabalho tem como base o Manual de Normalização de Projeto de Trabalho de Graduação da FatecSBCampo (2017), que se encontra ancorado nas normas da ABNT. É escrito em uma linguagem simples e concisa

### 2.1 Tema-problema com justificativa e descrição do seu funcionamento

O trabalho que se intitula Sistema de Identificação de Linhas de Ônibus para Deficientes Visuais tem por objetivo a acessibilidade de deficientes visuais ao meio de transporte público mais comum nas grandes cidades brasileiras. Justifica-se pelo grande número de deficientes visuais no Brasil e suas dificuldades na hora de

identificar os ônibus com seus destinos, para que possam sinalizar ao motorista o embarque no veículo, sem que haja a necessidade de ajuda de terceiros. A Figura 2.1 apresenta a arquitetura e um breve descritivo de funcionamento do que se pretende implementar.

Figura 2.1 – Arquitetura do sistema



Fonte: Foto de arquivo pessoal, 2018

Sendo:

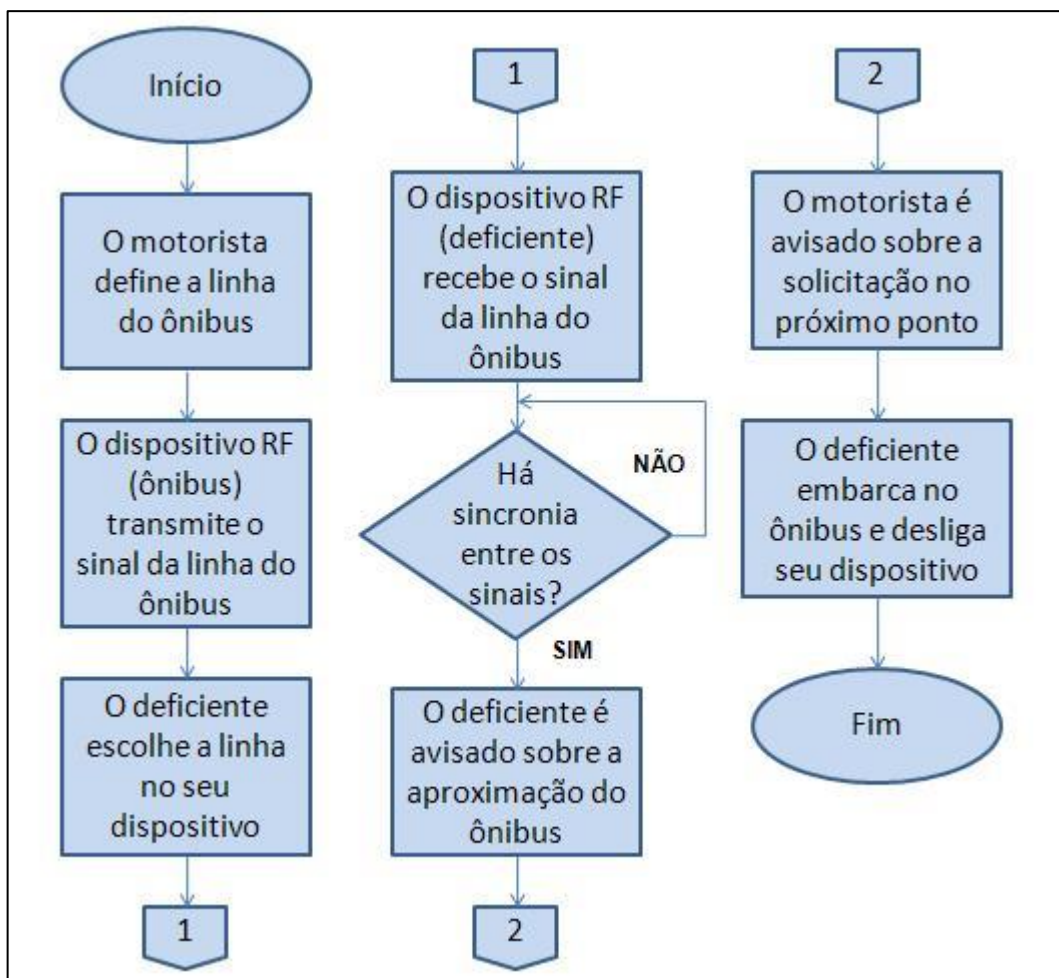
- (a) portador de deficiência visual: pessoa que está no ponto de ônibus e seleciona pelo seu dispositivo a linha de ônibus desejada para seu embarque
- (b) chave duas posições: responsável por ligar e desligar o circuito
- (c) bateria de 9 V: responsável pela alimentação elétrica do circuito



- (d) chave *push button*: responsável pela seleção e confirmação das linhas de ônibus
- (e) microcontrolador (lado portador): componente que vai gerenciar a seleção da linha de ônibus que o portador escolheu e se comunicar com os periféricos para sinalizar ao motorista do coletivo escolhido a presença de um deficiente no ponto de ônibus, o qual deseja embarcar no veículo;
- (f) reproduzidor de voz DFPlayer mini: componente que reproduz as mensagens gravadas no aparelho, informando as linhas de ônibus no alto falante ou fone de ouvido para o portador;
- (g) alto falante ou fone de ouvido: componente que está em porte do deficiente, o qual emitirá o som dizendo as linhas de ônibus disponíveis para ele;
- (h) e (q) módulo de radio-frequência (RF) NRF24L01: responsável pela comunicação (enviar e receber informações) via RF, entre o motorista do ônibus e o portador de deficiência;
- (i) meio de comunicação: é o meio onde há a troca de informações entre as duas partes, o solicitante e o solicitado;
- (j) ônibus: veículo que é solicitado pelo deficiente visual;
- (k) chave duas posições: responsável por ligar e desligar o circuito;
- (l) bateria de 9 V: responsável pela alimentação elétrica do circuito;
- (m) chave duas posições: responsável por definir o sentido do ônibus;
- (n) microcontrolador (lado ônibus): componente responsável em gerenciar e comunicar o motorista e seus periféricos;
- (o) *led* vermelho: responsável em sinalizar ao motorista a solicitação de parada no próximo ponto, devido a presença de um deficiente visual para embarque;
- (p) *buzzer*: responsável por avisar de modo sonoro ao deficiente que o ônibus está no local.

Para facilitar o entendimento lógico, a Figura 2.2 ilustra o fluxograma do funcionamento do processo.

Figura 2.2 – Fluxograma do funcionamento do processo



Fonte: Autoria própria, 2017

## 2.2 Etapas teóricas e práticas para construção do projeto

Após a delimitação do tema a ser construído com justificativa pautada na utilização da ferramenta de análise SWOT (*Strengths-Weaknesses-Opportunities-Threats* – Forças-Fraquezas-Oportunidades-Ameaças), desmembramos o objetivo geral em objetivos específicos, para então consagrarmos a ideia final do nosso

projeto. Com isso desenvolvemos a arquitetura do sistema e sua descrição, seguida de um fluxograma.

Primeira etapa: os integrantes do grupo reuniram-se com o orientador para traçar as diretrizes de como efetuar as pesquisas. O orientador fez uma breve explanação sobre o tema, indicou livros e sites especializados para pesquisa e marcou, obrigatoriamente, um dia por semana para lhe apresentar o andamento do trabalho.

Segunda etapa: realiza-se o levantamento bibliográfico na biblioteca da Fatec SBCampo, pesquisas em *sites* especializados, provenientes de pdf, manuais e catálogos de empresas especializadas.

Terceira etapa: discussão entre o grupo para definição do melhor tipo de linguagem a ser utilizada para programação do microcontrolador, bem como sua plataforma. Discussão levada também ao orientador do projeto que analisa os prós e contras.

Quarta etapa: após a seleção das bibliografias faz-se uma releitura dos tópicos apropriados ao desenvolvimento e construção do projeto e constrói-se o Capítulo 1 – Fundamentação Teórica e Referências.

Quinta etapa: levantamento dos materiais a serem utilizados para a construção do dispositivo. Consultas em *sitese* lojas especializadas. Viabilidade econômica e aquisição dos materiais conforme a Tabela 2.1.

Tabela 2.1 – Materiais para confecção do projeto

MATERIAL UTILIZADO	QUANTIDADE	PREÇO (R\$)
Placas Arduino UNO	3	131,70
Módulo Wireless NRF24L01	2	70,00
Módulo DFPlayer mini	1	25,90
Protoboard 400 pontos EIC- 65-40-8010 - E.I.C.	2	36,00
Fonte de Alimentação Bivolt 9 V 3A Plug P4 - D1203	2	52,00
Led Pisca 5mm Vermelho Difuso L-517HD	2	2,00
Chave Duas Posições	4	2,00
Chave Push Button - Normal Aberto	2	12,52
Resistor 82R $\Omega$ (ohm)	2	0,60
Cabos Jumper Macho/Macho	40	11,90
Cabos Jumper Macho/Fêmea	40	11,90
Adaptador Bateria 9 V	3	11,60
Bateria 9 V	3	11,60
Ônibus de brinquedo	2	60,00
Gabinete Patola PB-108	3	37,50
TOTAL		457,72

Fonte: Autoria própria, 2017

Sexta etapa: testes em bancada de cada componente eletrônico para verificar o comportamento desses elementos de forma isolada.

Sétima etapa: programação no software Arduino IDE 1.8.1. Testes de comunicação do microcontrolador com o módulo NRF24L01 e o DFPlayer Mini.

Oitava etapa: confecção mecânica das estruturas para o circuito elétrico

Nona etapa: integração das partes: eletrônica, programação e mecânica. Testes de funcionamento. Obstáculos e soluções.

Décima etapa: concluído o projeto faz-se as considerações finais e o resumo final.

### 3 DESENVOLVIMENTO DO PROJETO

Neste capítulo encontra-se o passo a passo da construção lógica e de todo o desenvolvimento do trabalho, além de figuras para um melhor entendimento do projeto que se intitula Sistema de Identificação de Linhas de Ônibus para Deficientes Visuais. Para melhor visualização a Figura 3.1 ilustra o projeto finalizado.

Figura 3.1 – Projeto finalizado



Fonte: Autoria própria, 2017

O sistema consiste em dois modelos de módulos microcontrolados, um que fica em posse do deficiente visual e outro, inserido dentro dos ônibus, com acesso aos motoristas.

O desenvolvimento e a construção estrutural do projeto são fundamentados nos seguintes tópicos:

- configurações e testes do Arduino;
- configuração da programação e testes do módulo de rádio frequência (NRF);

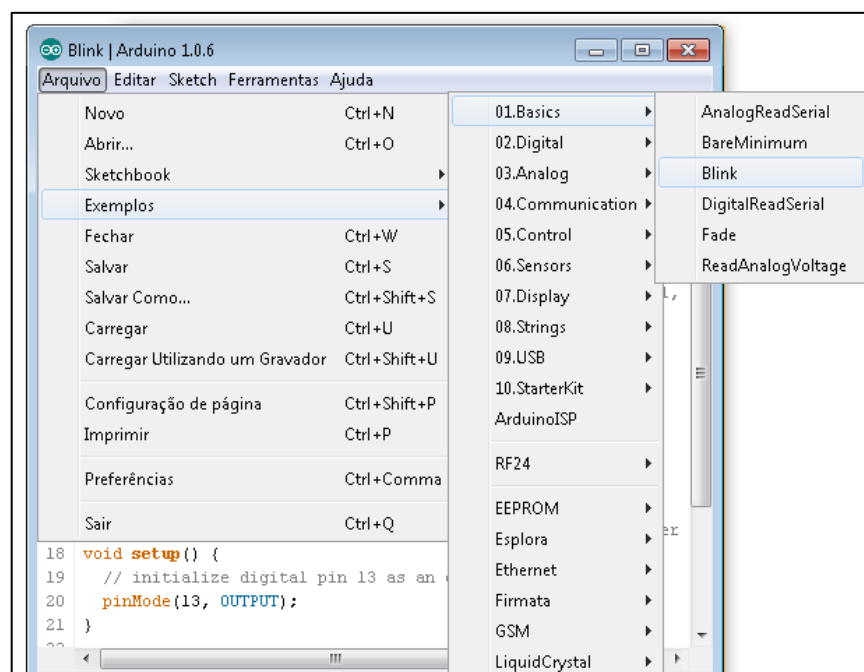
- configuração do módulo de áudio (DFPlayer);
- integração dos módulos Arduino, NRF e DFPlayer;
- desenvolvimento da parte estrutural e agregação dos módulos.

### 3.1 Configurações e testes do Arduino

Inicialmente utiliza-se um módulo Arduino com cabo USB padrão para conectá-lo à placa com o computador. Faz-se o *download* do ambiente de desenvolvimento, no site do Arduino.

Assim que os arquivos são baixados, conecta-se o Arduino ao computador, um led verde, sinaliza que a placa está energizada e acende. Instalam-se os *drivers* no computador, dá um duplo clique no ícone do aplicativo Arduino para executá-lo. Nesse momento abre-se o ambiente de desenvolvimento, e para verificar a funcionalidade da placa abre-se também um programa chamado “Blink”, conforme mostra a Figura 3.2.

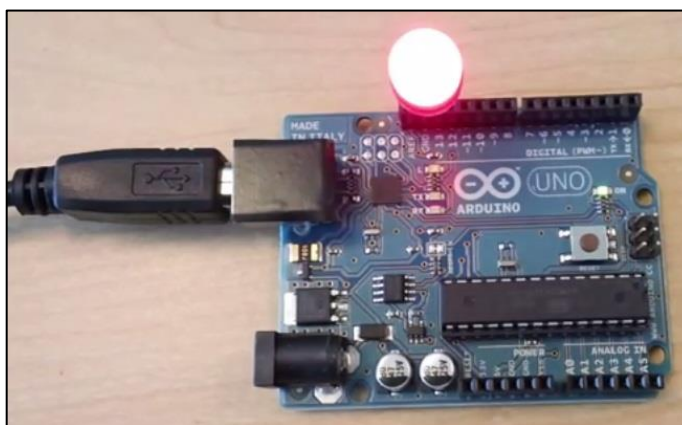
Figura 3.2 – Programa “Blink”



Após selecionar o “Blink”, escolhe-se a placa de Arduino a ser utilizada, neste caso a Uno

A seguir, seleciona-se o dispositivo serial correspondente à placa do Arduino, neste caso a COM3, no menu Ferramentas > Porta Serial. Clica-se no botão “carregar” no ambiente de desenvolvimento. Este procedimento carrega o programa teste “blink”, inicialmente citado. Aguarda-se alguns segundos para que os led’s RX e TX da placa estejam piscando. Se o *upload* for bem sucedido, a mensagem “transferência concluída” vai aparecer na barra de status. Após o *upload* terminar, coloca-se um led no pino 13 e GND e, se ele começar a piscar, determina-se que as configurações foram bem sucedidas e que a placa Arduino está pronta para ser utilizada, conforme ilustra a Figura 3.3.

Figura 3.3 – Teste “blink” funcionando



Fonte: Autoria própria, 2018

### 3.2 Configuração da programação e testes do módulo de rádio frequência

Verifica-se que, para o desenvolvimento das linhas de programação da comunicação do Arduino com o módulo NRF24L01 na linguagem C++, são necessárias algumas bibliotecas específicas. Realizam-se os *downloads* dessas bibliotecas através do *site* github.com. Os nomes das bibliotecas são citados na

Figura 3.4.

Figura 3.4 – Bibliotecas utilizadas na programação

```

1 //Inclusão das bibliotecas necessárias do módulo NRF2401
2 #include <SPI.h>
3 #include <nRF24L01.h>
4 #include <RF24.h>

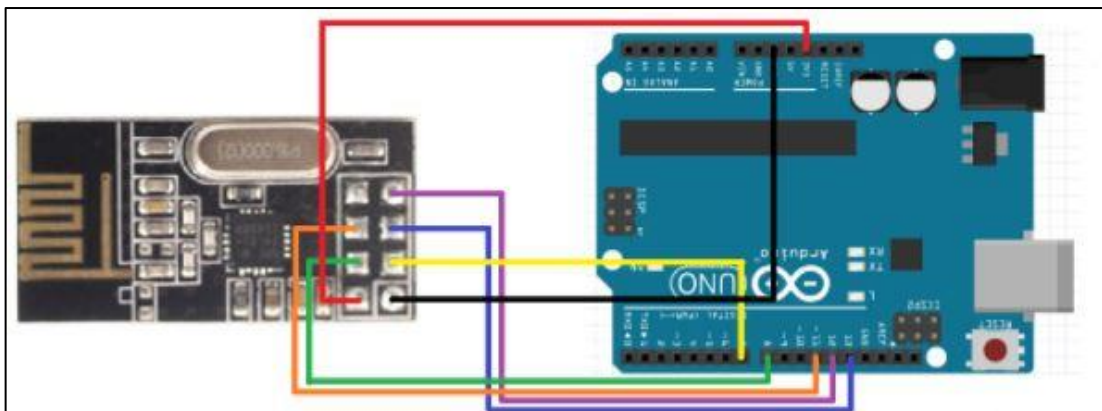
```

Fonte: Autoria própria, 2018

Dando sequência, é definido como deve funcionar o primeiro sistema simulador do projeto, para iniciar a programação do código. Para esse teste são usados dois módulos Arduino, um simulando o circuito do ônibus e o outro simulando o dispositivo do deficiente visual, cada um contendo seu código com linhas de programação dedicadas aos respectivos circuitos.

A energização dos Arduinos se dá por fontes de alimentação de 9 V. A alimentação dos módulos NRF24L01, por sua vez, é feita através de saídas do próprio Arduino. A ligação elétrica entre Arduino e NRF24L01 se dá conforme mostra a Figura 3.5.

Figura 3.5 – Ligação entre Arduino e NRF24L01



Fonte: Autoria própria, 2018



São adicionados aos circuitos dois *proto-boards*, cada um contendo um led e um resistor limitador de tensão e corrente. O led do circuito simulador do dispositivo do deficiente visual é responsável pela identificação do sinal do circuito que simula o ônibus, quando este entra no raio de alcance do NRF24L01. O led do sistema de simulação do ônibus, por sua vez, simula a aceitação do sinal que é feita pelo deficiente. Essa aceitação, no modelo original do projeto é mostrada no painel do ônibus.

O teste consiste em simular o envio do sinal pelo dispositivo do deficiente e seu recebimento no circuito do ônibus. Como a distância entre eles é de, aproximadamente, 10 metros (bem inferior aos 100 metros de alcance do NRF24L01), o sistema identifica a presença de uma linha de ônibus dentro do raio de alcance.

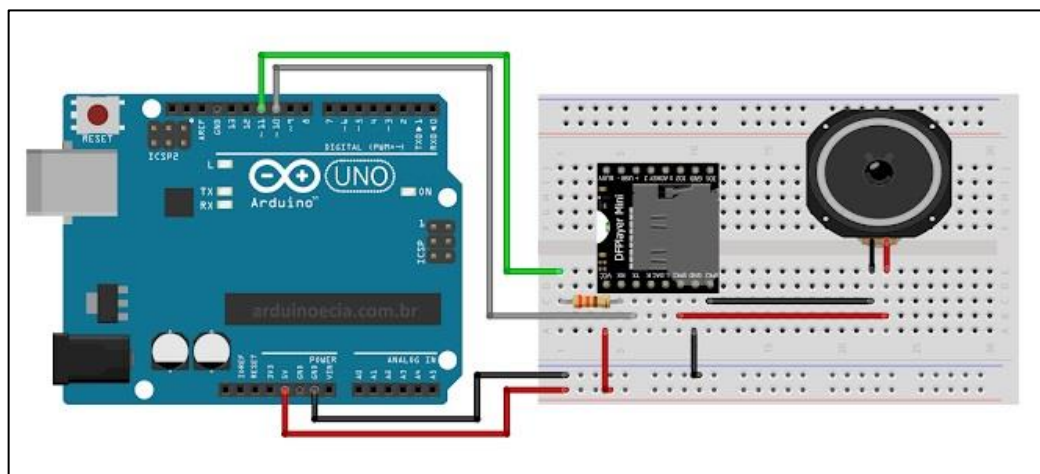
Por mais que esse teste não mostra a realidade exata imaginada para o projeto, ele é eficaz. Ambos os transceptores são capazes de enviar e receber sinais elétricos entre si. Esse teste é considerado como uma etapa inicial e é usado para implementação de melhorias no desenvolvimento do protótipo. Para incrementar o primeiro teste é incluído ao circuito simulador do dispositivo do deficiente visual o módulo de áudio DFPlayer.

### **3.3 Configuração do módulo DFPlayer**

Dando sequência, faz-se a configuração do módulo DFPlayer e, para isso, é necessário instalar a biblioteca DFPlayer Mini no código da programação. Através do ambiente de desenvolvimento do *software* do Arduino clica-se em Sketch > Incluir Biblioteca > Gerenciar Biblioteca e encontra-se a biblioteca DFPlayer Mini, da DFRobot.

O DFPlayer possui em sua constituição física um slot para inserção de cartão de memória microSD, onde são armazenados os áudios de informes necessários ao usuário do dispositivo. Esses áudios são gravados por nós mesmos, através do gravador de voz do sistema Android de celulares. Nas mensagens de áudio constam as seguintes informações: número da linha do ônibus, nome do destino do veículo, além de demais mensagens de orientação ao deficiente visual. Faz-se necessário também um alto-falante de 5 W / 4  $\Omega$ , que é conectado ao circuito para soar as mensagens gravadas. A Figura 3.6 mostra a ligação do Arduino com o módulo de áudio.

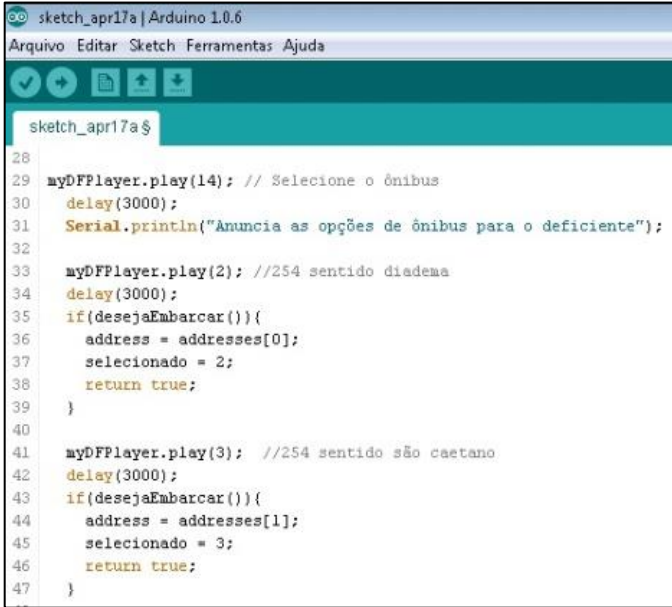
Figura 3.6 – Ligação do Arduino com o módulo de áudio



Fonte: Autoria própria, 2018

Após as configurações de inicialização do DFPlayer, os áudios são reproduzidos com sucesso pelo alto falante conforme programação, tornando o teste com o módulo de áudio satisfatório e eficaz para o projeto. A base da configuração do DFPlayer é ilustrada na Figura 3.7.

Figura 3.7– Programação para o DFPlayer



```

sketch_apr17a | Arduino 1.0.6
Arquivo Editar Sketch Ferramentas Ajuda
sketch_apr17a $
28
29 myDFPlayer.play(14); // Seleccione o ônibus
30 delay(3000);
31 Serial.println("Anuncia as opções de ônibus para o deficiente");
32
33 myDFPlayer.play(2); //254 sentido diadema
34 delay(3000);
35 if(desejaEmbarcar()){
36     address = addresses[0];
37     selecionado = 2;
38     return true;
39 }
40
41 myDFPlayer.play(3); //254 sentido são caetano
42 delay(3000);
43 if(desejaEmbarcar()){
44     address = addresses[1];
45     selecionado = 3;
46     return true;
47 }

```

Fonte: Autoria própria, 2018

### 3.4 Integração dos módulos Arduino, NRF e DFPlayer

Para implementação do módulo de áudio DFPlayer usa-se a base do circuito usado no primeiro teste de integração do módulo NRF24L01 com o Arduino. Nessa nova etapa, simulam-se dois ônibus, ao invés de apenas um. O sistema simulador de cada ônibus possui seu módulo Arduino alimentado por meio de uma fonte de 9V, um módulo transceptor NRF24L01, uma chave seletora da linha/sentido do ônibus, uma chave do tipo *push button* e um *led* para sinalização.

O sistema do deficiente também possui o Arduino alimentado por fonte de 9V. Além disso, possui um outro módulo transceptor NRF24L01, que se comunica com o módulo de radiofrequência do circuito do ônibus e, por fim, o novo módulo DFPlayer, responsável pela reprodução das mensagens de áudio nele gravadas. É usada também uma chave interruptora do tipo *push button*, para interface do sistema com o deficiente visual.

O usuário, ao ligar seu dispositivo, tem reproduzidas em seu dispositivo, em mensagens de áudio previamente gravadas, todas as linhas disponíveis no sistema, as quais possuem veículos aptos a receberem sua solicitação de embarque. A cada opção de linha o sistema dá ao usuário quatro segundos para confirmar, através da chave *push button*, a solicitação daquele ônibus ou, em caso de não ser a linha desejada, após esse tempo é reproduzida em sequência a mensagem referente à outra linha, até que haja a confirmação de requisição de alguma delas.

São colocadas quatro opções de linhas de ônibus no código do programa do dispositivo do deficiente visual: dois destinos para cada um dos ônibus simulados no teste. Essa aplicação simula de maneira ideal a realidade, sendo que cada veículo deve estar apto a enviar duas codificações diferentes, de acordo com o sentido que o ônibus se direciona em dado momento. As mensagens de áudio gravadas são: “Ônibus 254 – Sentido Diadema”, “254 - Sentido São Caetano”, “288 – Sentido Ferrazópolis” e “288 – Sentido Jabaquara”.

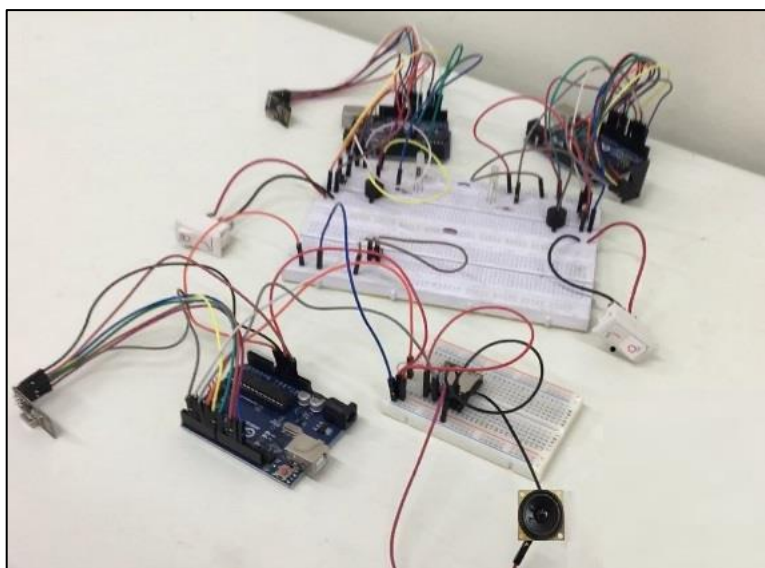
Além das mensagens de informe das linhas disponíveis, são gravadas também outros áudios para interface completa entre usuário e sistema. A mensagem: “Pressione o botão ao ouvir a linha desejada” para ser reproduzida no momento em que o usuário liga o seu dispositivo. A mensagem: “Sua escolha foi salva. Aguarde no ponto a notificação do seu ônibus”, para o momento em que o usuário seleciona alguma das linhas disponíveis e também a mensagem: “Ônibus ‘X’. Para embarcar, pressione o botão”, para o momento em que o veículo escolhido entra no raio mínimo para comunicação do sistema, onde ‘X’ é substituído pelo áudio com o nome do ônibus desejado pelo usuário.

No dispositivo simulador do equipamento de cada um dos ônibus, existem dois destinos disponíveis para seleção. Destino esse que é alterado mudando a posição da chave de duas posições que, no código do programa, faz o sistema trocar a codificação a ser enviada pela rede de rádio-frequência. O código referente a determinado sentido da linha de ônibus (condizente com a posição da chave) é

enviado, de maneira ininterrupta, pela rede *wireless* com o veículo em movimento. Ao estabelecer o raio mínimo para comunicação com o dispositivo de um usuário no ponto de ônibus, ele é notificado. Sendo esse veículo o desejo de embarque do deficiente visual, o motorista recebe uma notificação de confirmação, o que indica que deve parar no próximo ponto para embarque do usuário.

Ao chegar no ponto de embarque, o motorista do ônibus da linha solicitada pelo usuário pressiona o botão do seu dispositivo, que é o responsável por soar um alarme através do *buzzer*. Esse indicativo sonoro torna perfeita a identificação do veículo correto para embarque e, desse modo, o deficiente pode adentrar ao ônibus de maneira adequada. A montagem executada nessa simulação é ilustrada na Figura 3.8.

Figura 3.8 – Teste com módulos integrados



Fonte: Autoria própria, 2018

Essa configuração de teste mostra-se perfeita de acordo com o necessário para viabilidade do projeto. O códigos do programas completos e definitivo encontra-se nos Apêndices.

Sem inconsistências do sistema e com o programa funcionando exatamente como o planejado, parte-se então para o desenvolvimento das estruturas capazes de armazenar os hardwares englobados nesse teste.

### **3.5 Desenvolvimento da parte estrutural e agregação dos módulos.**

A criação da parte física do protótipo se dá a partir da necessidade de simulação de uma situação próxima a que o sistema encontra nas ruas, quando em funcionamento. Para que essa simulação se concretiza de forma semelhante aos usuários deficientes e companhias de ônibus, são necessários três elementos estruturais: um *hardware* para porte do deficiente visual e outros dois, que se encontram nos ônibus.

São utilizados dois ônibus em miniatura, os quais possuem em seus interiores circuitos dedicados para interface com os motoristas. Os ônibus são fixados em um dispositivo mecânico de ferro. Utilizando uma furadeira elétrica, paquímetro e brocas de 5,5 e 6 mm, efetua-se as furações nas miniaturas para que sejam colocados as interfaces com o motorista tais como: botões de aviso sonoro ao deficiente, chaves liga / desliga circuito, chaves de sentido dos ônibus e led vermelho indicando a solicitação do ônibus pelo usuário, conforme ilustra a Figura 3.9.

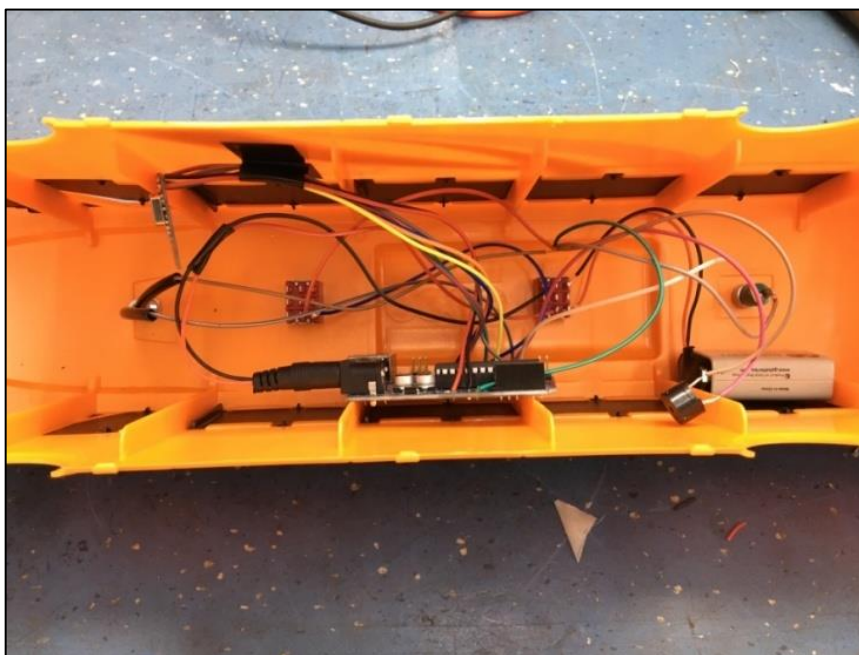
Figura 3.9 – Furações na miniatura e colocação das interfaces



Fonte: Autoria própria, 2018

Retira-se a parte de baixo dos ônibus, para que se tenha acesso a parte interna, onde são alocadas a bateria, a placa com o microcontrolador com seus periféricos, módulo de radiofrequência e o circuito do buzzer. Fixam-se todos os componentes dentro da miniatura, por meio de cola silicone e fitas adesivas, e os conectam por meio de solda com estanho, conforme ilustra a Figura 3.10.

Figura 3.10 – Fixação dos componentes na miniatura



Fonte: Autoria própria, 2018

Todos os processos são realizados de maneira igual nos dois ônibus. Agora através de uma máquina etiquetadora, imprime-se adesivos e identifica-se cada ônibus, com suas linhas, funções de seus botões, chaves e leds.

Após realizar-se todos os procedimentos citados, realiza-se a confecção do dispositivo do deficiente visual. Para isto adquire-se um gabinete com tampa e com as medidas de 190 x 112 x 36 mm, local onde são fixados os componentes do circuito. Com o auxílio de um dispositivo mecânico e utilizando uma furadeira elétrica, paquímetro e brocas de 5, 6 e 7 mm, efetua-se as furações no gabinete para que sejam colocados as interfaces com o usuário tais como: botão de seleção das linhas / de confirmação, chave liga / desliga circuito, e saída para áudio, conforme ilustra a Figura 3.11.



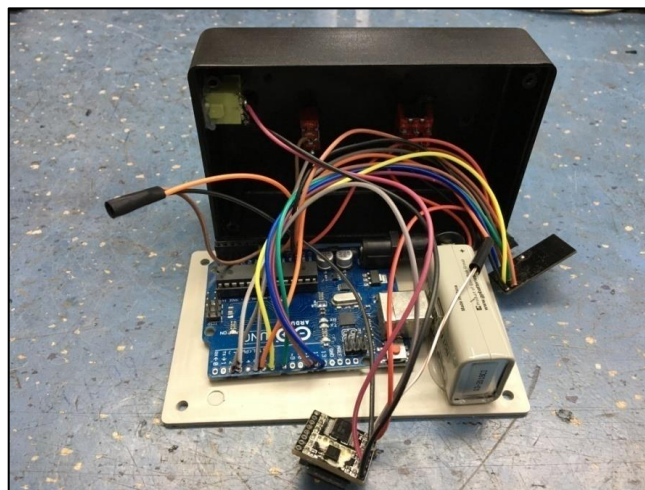
Figura 3.11 – Furações no gabinete e colocação das interfaces



Fonte: Autoria própria, 2018

Abre-se a tampa do gabinete e aloca-se a placa com o microcontrolador, o módulo radio-freqüência, o módulo de áudio e a bateria, sendo que todos são fixos por meio de cola silicone e fita adesiva, conforme ilustra a Figura 3.12.

Figura 3.12– Fixação dos componentes no gabinete



Fonte: Autoria própria, 2018

Agora através de uma máquina etiquetadora, imprime-se adesivos e identifica-se o botão de seleção / confirmação, a chave liga / desliga e, a saída de áudio para alto-falantes ou fones de ouvido no gabinete portátil do deficiente visual.

## CONSIDERAÇÕES FINAIS

O objetivo do trabalho que se intitula Sistema de Identificação de Linhas de Ônibus para Deficientes Visuais é desenvolver um sistema que seja capaz de tornar o embarque dos deficientes prático e independente. Justifica-se pela fácil acessibilidade, conforto, praticidade do sistema, autonomia do deficiente visual e dispensa de pessoa auxiliadora para identificar o ônibus que se aproxima para o embarque no veículo desejado.

O desenvolvimento do projeto se dá, fundamentalmente, por meio de transceptores de rádio-frequência NRF24L01, módulos Arduino Uno e módulo armazenador e reproduzidor de áudio DFPlayer, alocados dentro de miniaturas de ônibus e caixa para desenvolvimento de protótipos.

As teorias pesquisadas são de suma importância para dar sustentação e concretização no desenvolvimento do projeto, principalmente conhecimentos de programação dos módulos Arduino, NRF24L01 e DFPlayer.

A direção dada pela metodologia científica torna a organização do desenvolvimento do projeto mais fácil, principalmente pelas etapas dos métodos e técnicas estudadas. Os métodos e técnicas ampliam o raciocínio facilitando o emprego dos instrumentos adequados.

A principal vantagem desse projeto é a facilitação de uma atividade usual de muitos deficientes visuais, que é o embarque em ônibus, sem que haja a necessidade de auxílio de terceiros.

Os pontos de melhoria do projeto são a redução da disposição física do dispositivo do deficiente visual e uma reprodução mais breve e dinâmica da apresentação das linhas disponíveis no sistema do deficiente, em sua inicialização.

Durante o desenvolvimento do projeto foram encontrados alguns obstáculos com relação, principalmente, à ordem de funcionamento do sistema. O programa chegou a ser dado como concluído com a seguinte configuração: o deficiente visual recebia em seu dispositivo mensagens de notificação referentes a todos os ônibus, portadores do sistema, que se aproximavam do local onde ele se encontrava. Concluiu-se que, para uma aplicação em larga escala, a possibilidade de haver vários veículos emitindo sinais que poderiam confundir o usuário deveria ser evitada. Com isso alterou-se a ordem de funcionamento do sistema, que passou a dar ao deficiente visual, antes de tudo, a opção de seleção da linha de ônibus desejada. Dessa forma ele recebe, portanto, unicamente o sinal de aproximação do veículo no qual ele deseja embarcar.

O projeto tem como sustentação uma contribuição social muito significativa, que é tornar independente uma ação frequente de pessoas com dificuldades visuais: o embarque em ônibus. O engajamento com o tema abordado nesse trabalho unido à integração de tantos outros projetos de inovações tecnológicas, irão sempre colaborar com uma sociedade cada vez mais igualitária, em todos os seus aspectos.

## REFERÊNCIAS

AGENCIA BRASIL. **Um em cada quatro brasileiros usa o ônibus como principal meio de transporte.** Disponível em: <<http://agenciabrasil.ebc.com.br/geral/noticia/2015-10/um-em-cada-quatro-brasileiros-usa-o-onibus-como-principal-meio-de-transporte>>. Acesso em: 18 abr. 2017.

ALBUQUERQUE, P. U. B. de; ALEXANDRIA, A. R. de. **Redes Industriais: Aplicações em sistemas digitais de controle distribuído.** 2. ed. São Paulo: Profissional, 2009.

APLUS. **AP8942A.** Disponível em: <<http://www.aplusinc.com.tw/exec/product.php?mod=show&cid=13&pid=aP8942A&lg=E.>>. Acesso em: 28 abr. 2017.

ARDUINO E CIA. **Comunicação wireless com Arduino e módulo NRF24L01 2.4GHz.** Disponível em: <http://www.arduinoecia.com.br/2015/02/comunicacao-wireless-arduino-nrf24l01.html>. Acesso em: 28 abr. 2017.

BAPTISTA, R. de S. **Identificação de sistemas híbridos e controle robusto.** 2011. Dissertação de Mestrado – Universidade de Brasília, Brasília. Disponível em: <[http://www.repositorio.unb.br/bitstream/10482/10356/1/2011\\_RobertoSouzaBaptista.pdf](http://www.repositorio.unb.br/bitstream/10482/10356/1/2011_RobertoSouzaBaptista.pdf)>. Acesso em 02 mai. 2017

BRASIL. **Decreto nº 5.296, de 02 de dezembro de 2004.** Regulamenta a Lei nº 10.048, de 08 de novembro de 2000, que dá prioridade de atendimento às pessoas que especifica, e a Lei nº 10.098, de 19 de dezembro de 2000, que estabelece normas gerais e critérios básicos para a promoção da acessibilidade das pessoas portadoras de deficiência ou com mobilidade reduzida, e dá outras providências. Diário Oficial da República Federativa do Brasil, Brasília, DF, 3 dez. 2004. Disponível em: <[http://www.planalto.gov.br/ccivil/\\_ato20042006/2004/Decreto/D5296.htm](http://www.planalto.gov.br/ccivil/_ato20042006/2004/Decreto/D5296.htm)>. Acesso em: 4 abr. 2017

DANTAS, M. **Tecnologias de redes de comunicação e computadores.** 1. ed. Rio de Janeiro: Axcel Books, 2002.

DFROBOT. **DFPlayer Mini.** Disponível em: <[https://www.dfrobot.com/wiki/index.php/DFPlayer\\_Mini\\_SKU:DFR0299](https://www.dfrobot.com/wiki/index.php/DFPlayer_Mini_SKU:DFR0299)>. Acesso em: 12. mar. 2018

EMBARCADOS. **Comunicação SPI - Parte 3 - Microcontrolador AT89S8253 + EEPROM25LC256**. Disponível em: <<https://www.embarcados.com.br/comunicacao-spi-parte-3-at89s8253-25lc256/>>. Acesso em: 12 mai. 2017.

FREITAS, A. E. de. **Predição de cobertura em enlaces radioelétricos sobre terrenos irregulares através de equações integrais**. 2001. 87f. Dissertação de Mestrado em Engenharia Elétrica – Universidade Federal de Minas Gerais. Belo Horizonte: UFMG, 2001.

GUIA DE DIREITOS. **Identificação da linha**. Disponível em: <[http://www.guiadedireitos.org/index.php?option=com\\_content&view=article&id=1251&Itemid=299](http://www.guiadedireitos.org/index.php?option=com_content&view=article&id=1251&Itemid=299)>. Acesso em: 18 abr. 2017.

HADDAD, L. C. **Introdução ao Arduino**. 2016. Disponível em: <<http://hardwarelivreusp.org/2016/11/20/arduino-6intro/>>. Acesso em: 12 mar. 2018.

IBGE – INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA, **Estatísticas de Gênero**. Disponível em: <<http://www.ibge.gov.br/apps/snig/v1/?loc=0&cat=-1,-2,-3,128&ind=4642>>. Acesso em: 08 jun. 2017.

MACHADO, R. **Curso de comunicação de dados**. Disponível em: <[http://coral.ufsm.br/gpscom/professores/Renato%20Machado/ComunicacaoDeDados/ComDados08\\_Renato.pdf](http://coral.ufsm.br/gpscom/professores/Renato%20Machado/ComunicacaoDeDados/ComDados08_Renato.pdf)>. Acesso em: 02 mai. 2017.

MANUAL DE NORMALIZAÇÃO DE PROJETO DE TRABALHO DE GRADUAÇÃO – FATEC SBCAMPO. **Material didático para utilização nos projetos de trabalho de graduação dos cursos de tecnologia em automação industrial e informática**. São Bernardo do Campo: Fatec, 2017.

MELO, M. do. S. R. de. **Transporte coletivo urbano e acessibilidade na área central de Teresina**: um instrumento na (re)organização do espaço. Revista brasileira de geociências. Paraíba: UFPB, 2011.

MENDES, D. R. **Redes de computadores: Teoria e prática**. 2. ed. São Paulo: Novatec, 2016.

MOBILIZE, **Estatísticas**. Disponível em: <<http://www.mobilize.org.br/estatisticas/>> .

Acesso em: 10 abr. 2017.

NORDICSEMICONDUCTORS. **NRF24L01**. Disponível em: <<http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01>>. Acesso em: 19 abr. 2017.

PEREIRA, F. **Microcontroladores PIC: Programação em C**. 7. ed. São Paulo: Érica, 2010.

RODRIGUES, M.; CUGNASCA, C. E.; QUEIROZ FILHO, A. P. de. **Rastreamento de veículos**. 1. ed. São Paulo: Oficina de Textos, 2009.

SÃO PAULO. COMPANHIA DO METROPOLITANO DE SÃO PAULO – METRÔ (2017). **Serviços para pessoas com deficiência**. Disponível em: <<http://www.metro.sp.gov.br/pdf/ acessibilidade/atendimento-preferencial.pdf>>. Acesso em: 24 abr. 2017.

\_\_\_\_\_. COMPANHIA PAULISTA DE TRENS METROPOLITANOS – CPTM (2017). **Acessibilidade**. Disponível em: <<http://www.cptm.sp.gov.br/a-companhia/Pages/acessibilidade.aspx>>. Acesso em: 24 abr. 2017.

SEVERINO, A.J. **Metodologia do trabalho científico**. 23. ed. São Paulo:Cortez, 2011.

SOUZA, D. J. de. **Desbravando o PIC: Ampliado e atualizado para PIC 16F628A**.12. ed. São Paulo: Érica, 2010.

ZANCO, W. da. S. **Microcontroladores PIC: Técnicas de software e hardware para projetos de circuitos eletrônicos com base no PIC16F877A**. 2. ed. São Paulo: Érica, 2011.

## APÊNDICES

### APÊNDICE A – PROGRAMA DO DISPOSITIVO DO ÔNIBUS

```
//Inclusão das bibliotecas necessárias do módulo nRF2401
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
//Definição dos GPIO do arduino
#define IRQ 2
#define BOTAO 3
#define INDICADOR 4
#define BUZZER 5
#define SELECIONADOR 9
//Instancia da classe RF24
RF24 radio(7, 8);
//variaveis globais utilizadas no programa
uint64_t addresses[] = {0xB3B4B5B6CDLL, 0xB3B4B5B6A3LL};
uint64_t address;
bool modoTransmissorON;
bool alguemEspera;
void setup() {
  Serial.begin(115200);
  pinMode(BOTAO, INPUT_PULLUP);
  pinMode(INDICADOR, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  pinMode(SELECIONADOR, INPUT_PULLUP);
  delay(1000);
  if(digitalRead(SELECIONADOR)){
    address = addresses[0];
```



```
}  
else{  
    address = addresses[1];  
}  
digitalWrite(INDICADOR, LOW);  
configuraRadio();  
modoTransmissorON = false;  
alguemEspera = false;  
}  
void loop() {  
    if(!modoTransmissorON) modoTransmissor();  
    enviaSinalRF();  
    modoReceptor();  
    if(alguemEspera){  
        if(!digitalRead(BOTAO)){  
            desligaIndicador();  
        }  
    }  
}  
void configuraRadio(){  
    radio.begin();  
    radio.setPALevel(RF24_PA_MIN);  
    radio.setChannel(80);  
}  
void modoTransmissor(){  
    radio.stopListening();  
    radio.openWritingPipe(address);  
    modoTransmissorON = true;  
}  
void enviaSinalRF(){
```

```

const char ida[] = "Linha 0xB3B4B5B6F1LL";
const char volta[] = "Linha 0xB3B4B5B6CDLL";
if(address == addresses[0])
    radio.write(&ida, sizeof(ida));
else
    radio.write(&volta, sizeof(volta));
delay(1000);
}

void modoReceptor(){
    radio.openReadingPipe(0, address);
    radio.startListening();
    modoTransmissorON = false;
    attachInterrupt(digitalPinToInterrupt(IRQ), acionalIndicador, FALLING);
    delay(1500);
    detachInterrupt(digitalPinToInterrupt(IRQ));
}

void acionalIndicador(){
    if(radio.available()){
        digitalWrite(INDICADOR, HIGH);
        alguemEspera = true;
        char texto[32] = "";
        radio.read(&texto, sizeof(texto));
        Serial.print("Recebido: ");
        Serial.println(texto);
    }
}

void desligaIndicador(){
    digitalWrite(INDICADOR, LOW);
    int contador = 0;
    while(contador < 5){

```

```
tone(BUZZER, 2000, 500);  
delay(1000);  
contador++;  
}  
alguemEspera = false;  
}
```

## APÊNDICE B – PROGRAMA DO DISPOSITIVO DO DEFICIENTE VISUAL

```
//Biblioteca módulo MP3 DFPlayer Mini
#include "DFRobotDFPlayerMini.h"

//Biblioteca para comunicação serial do arduino
#include "SoftwareSerial.h"

//Bibliotecas para a utilização do rádio
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

//Constantes
#define IRQ 2 //Define o pino de interrupção do arduino
#define EMBARQUE 3 //Define o botão de seleção de embarque

//Variáveis globais
uint64_t addresses[] = {0x7878787878LL, 0xB3B4B5B6F1LL, 0xB3B4B5B6CDLL, 0xB3B4B5B6A3LL,
0xB3B4B5B60FLL, 0xB3B4B5B605LL};

uint64_t address;

bool modoEscutaON;

int tentativas;

int selecionado;

//Inicia a serial por software nos pinos 5 e 6
SoftwareSerial mySoftwareSerial(5, 6);

//Cria o objeto Player
DFRobotDFPlayerMini myDFPlayer;

//Inicia o rádio nos pinos 7 e 8
RF24 radio(7, 8);

//Configurações
void setup() {
    Serial.begin(115200);
    pinMode(EMBARQUE, INPUT_PULLUP);
```

```

modoEscutaON = false;

configuraAudio();

while(!selecionaOnibus());

myDFPlayer.play(13); //Escolha salva

delay(5000);

configuraRadio();

}

//Função loop

void loop() {

  if(!modoEscutaON) modoEscuta();

  //Verifica se possui sinal de ônibus

  byte pipeNum = -1;

  if(radio.available(&pipeNum)){

    radio.stopListening();

    //-----

    char texto[32] = "";

    radio.read(&texto, sizeof(texto));

    Serial.print("Recebido: ");

    Serial.print(texto);

    Serial.println(" Deseja embarcar?");

    Serial.print(" -- Pipe: ");

    Serial.println(pipeNum);

    //-----

    /*

    * Informa através do módulo wtv20 o ônibus que está próximo

    * e pergunta se deseja embarcar

    */

    informaOnibus(pipeNum);

    if(desejaEmbarcar()){

      tentativas = 0;

```

```
modoTransmissor(pipeNum);
enviaComandoEmbarque();
if(tentativas < 25){
    mensagemConfirmacao();
    while(1);
}
else mensagemErroComunicacao();
}
else{
    mensagemNaoConfirmacao();
}
radio.startListening();
}
}

bool selecionaOnibus(){
    myDFPlayer.play(14); // Selecione o ônibus
    delay(3000);
    Serial.println("Anuncia as opções de ônibus para o deficiente");
    myDFPlayer.play(2); //254 sentido diadema
    delay(3000);
    if(desejaEmbarcar()){
        address = addresses[0];
        selecionado = 2;
        return true;
    }
    myDFPlayer.play(3); //254 sentido são caetano
    delay(3000);
    if(desejaEmbarcar()){
        address = addresses[1];
        selecionado = 3;
```

```

    return true;
}

myDFPlayer.play(4); //288 sentido ferrazopolis
delay(3000);
if(desejaEmbarcar()){
    address = addresses[2];
    selecionado = 4;
    return true;
}

myDFPlayer.play(5); //288 sentido jabaquara
delay(3000);
if(desejaEmbarcar()){
    address = addresses[3];
    selecionado = 5;
    return true;
}

return false;
}

//Configurações e inicialização do módulo MP3 DFPlayer Mini
void configuraAudio(){
    mySoftwareSerial.begin(9600);
    Serial.println();
    Serial.println(F("DFRobot DFPlayer Mini"));
    Serial.println(F("Inicializando modulo DFPlayer... (3~5 segundos)"));
    if (!myDFPlayer.begin(mySoftwareSerial))
    {
        Serial.println(F("Nao inicializado:"));
        Serial.println(F("1.Cheque as conexoes do DFPlayer Mini"));
        Serial.println(F("2.Insira um cartao SD"));
        while (true);
    }
}

```

```

}

Serial.println();

Serial.println(F("Modulo DFPlayer Mini inicializado!"));

//Definicoes iniciais

myDFPlayer.setTimeout(500); //Timeout serial 500ms

myDFPlayer.volume(28); //Volume 19

myDFPlayer.EQ(0); //Equalizacao normal
}

//Configurações do rádio

void configuraRadio(){

  radio.begin();

  radio.setPALevel(RF24_PA_LOW);

  radio.setChannel(80);

  //radio.setAutoAck(false);

}

void modoEscuta(){

  radio.openReadingPipe(0, address);

  /*radio.openReadingPipe(1, addresses[1]);

  radio.openReadingPipe(2, addresses[2]);

  radio.openReadingPipe(3, addresses[3]);

  radio.openReadingPipe(4, addresses[4]);

  radio.openReadingPipe(5, addresses[5]);*/

  radio.startListening();

  modoEscutaON = true;

}

void informaOnibus(byte pipeNum){

  //Informa por meio do módulo DFPlayer Mini o ônibus

  myDFPlayer.play(8); //Onibus

  delay(1000);

  myDFPlayer.play(selecionado); //Linha do onibus

```



```
    delay(3000);

    myDFPlayer.play(9); //Para embarcar

    delay(3000);
}

bool desejaEmbarcar(){

    int i = 0;

    while(i < 10){

        if(!digitalRead(EMBARQUE)){

            return true;

        }

        delay(500);

        if(i % 2 == 0) tocaContador();

        i++;

    }

    return false;

}

void tocaContador(){

    //Emite um som para simular contagem

    myDFPlayer.play(12);

    Serial.print(".");

}

void modoTransmissor(byte pipeNum){

    radio.stopListening();

    radio.openWritingPipe(address);

    modoEscutaON = false;

}

void enviaComandoEmbarque(){

    char texto[] = "Desejo embarcar";

    while(!radio.write(&texto, sizeof(texto)) && tentativas < 25){

        //if(tentativas > 100) break;
```

```
Serial.print("Enviando comando... Tentativa ");  
Serial.println(tentativas);  
tentativas++;  
delay(200);  
}  
}  
void mensagemConfirmacao(){  
Serial.println("Mensagem recebida pelo motorista!");  
myDFPlayer.play(10);  
delay(5000);  
myDFPlayer.play(1);  
delay(4000);  
}  
void mensagemNaoConfirmacao(){  
Serial.println("Mensagem nao foi enviada!");  
}  
void mensagemErroComunicacao(){  
Serial.println("Erro na comunicacao");  
myDFPlayer.play(11);  
delay(6000);  
}
```