



UNIVERSIDADE FEDERAL RURAL DO SEMI-ARIDO
DEPARTAMENTO DE CIÊNCIAS EXATAS E NATURAIS
BACHARELADO EM CIÊNCIA E TECNOLOGIA

WALKERNANDES FERREIRA DA SILVA FILHO

**PROTÓTIPO DE SISTEMA AUDÍVEL PARA DEFICIENTES VISUAIS NA
IDENTIFICAÇÃO DE RECIPIENTES DE COLETA SELETIVA**

CARAÚBAS - RN

2018

Walkernandes Ferreira da Silva Filho

**PROTOTIPO DE SISTEMA AUDIVEL PARA DEFICIENTES VISUAIS NA
IDENTIFICACAO DE RECIPIENTES DE COLETA SELETIVA**

Trabalho de Conclusão de Curso
apresentado à Universidade Federal Rural
do Semiárido - UFERSA, Câmpus
Caraúbas, para a obtenção do Título de
Bacharel em Ciência e Tecnologia.

Orientador: Me. Juan Rafael Filgueira
Guerra- UFERSA.

CARAÚBAS - RN

2018

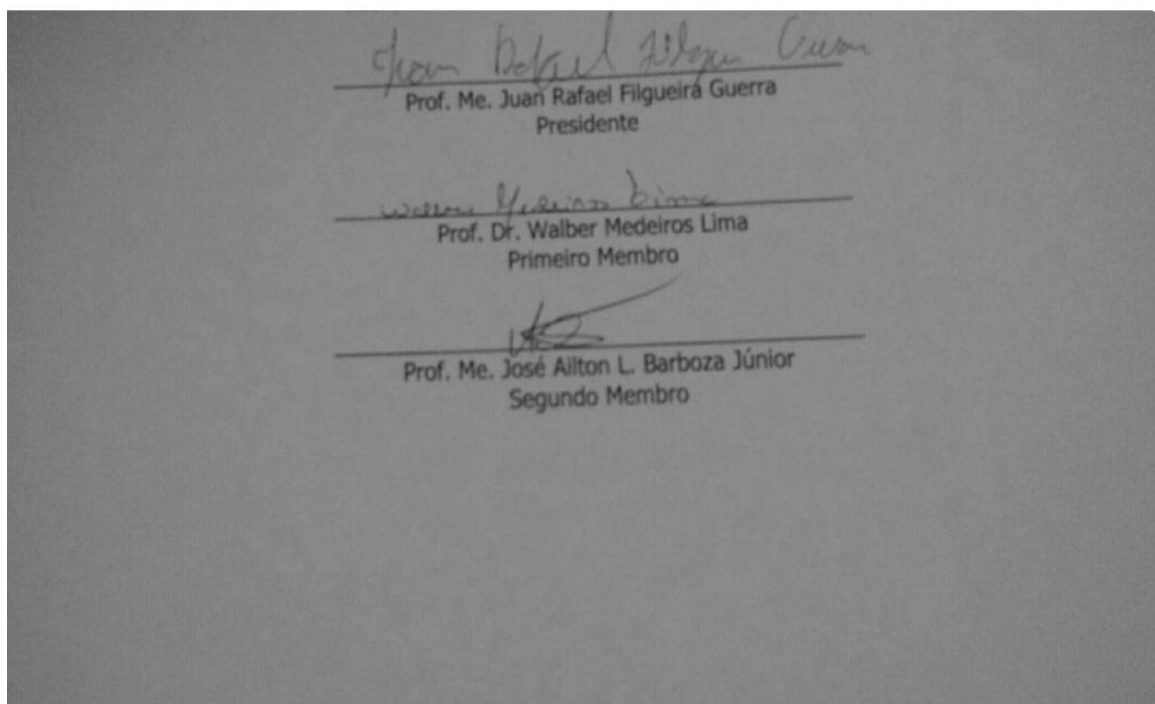
WALKERNANDES FERREIRA DA SILVA FILHO

**PROTÓTIPO DE SISTEMA AUDÍVEL PARA DEFICIENTES VISUAIS NA
IDENTIFICAÇÃO DE RECIPIENTES DE COLETA SELETIVA**

Trabalho de Conclusão do curso de
Ciência e Tecnologia da Universidade
Federal Rural do Semi-Árido – UFERSA,
como exigência final para obtenção do
título de Bacharel em Ciência e
Tecnologia.

APROVADA EM: 19 /04 /2018

BANCA EXAMINADORA



CARAÚBAS - RN

2018

FW179 Filho, Walkernandes Ferreira da Silva.
p PROTÓTIPO DE SISTEMA AUDÍVEL PARA DEFICIENTES
VISUAIS NA IDENTIFICAÇÃO DE RECIPIENTES DE COLETA
SELETIVA / Walkernandes Ferreira da Silva Filho.
2018.
50 f. : il.

Orientador: Juan Rafael Filgueira Guerra.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Ciência e
Tecnologia, 2018.

1. Deficiente visual. 2. Protótipo. 3. Arduino.
I. Guerra, Juan Rafael Filgueira, orient. II.
Título.

DEDICATÓRIA

À meus pais, Walkernandes Ferreira e Vanda Maria, por ter me ajudado durante toda a minha formação acadêmica, possibilitando que eu adquirisse conhecimento e força para concluir este curso e essa etapa da minha vida.

À minha família e amigos que participaram na minha formação pessoal, dando forças e incentivos durante todo o percurso.

À minha amada namorada, Naftally Dantas, por todo amor e incentivo nessa trajetória.

Ao meu amigo e orientador, Juan Rafael Filgueira Guerra, que passou todo o conhecimento e orientação necessária para que fosse possível a realização desse trabalho.

“Tudo aquilo que o homem ignora, não existe para ele. Por isso o universo de cada um, se resume no tamanho de seu saber”.

Carl Sagan.

RESUMO

A fim de diminuir as barreiras existentes que os deficientes visuais enfrentam dia após dia, foi desenvolvido um protótipo com o intuito de facilitar a identificação dos recipientes de coleta seletiva pelos deficientes visuais, através da reprodução de efeitos sonoros que podem facilitar e tornar prático sua compreensão. Foi utilizado um sensor de ultrassom, uma placa microcontrolada chamada Arduino Mega2560, um leitor de cartão micro SD e outros dispositivos eletrônicos. O motivo da escolha do Arduino Mega foi pela sua alta praticidade de trabalho, detém de um bom custo benefício e várias bibliotecas prontas nas quais facilitaram bastante a construção desse protótipo. A linguagem de programação utilizada foi a C, que é utilizada pelo software do Arduino. O sensor de ultrassom foi escolhido por sua alta viabilidade econômica e funcionalidade. A condição programada para que o protótipo funcionasse foi a que: Na medida que um indivíduo entre no raio de atuação do sensor de ultrassom e a distância pré-definida de 200 cm ou menos do sensor, um som seria reproduzido informando qual tipo de recipiente de coleta seletiva o deficiente visual estaria. Na ideia inicial, seria utilizado de um sensor de carga e um módulo amplificador HX711, porém sua construção inicial não foi completada e algumas substituições foram feitas. Todos os testes feito para o protótipo foram atendidos, o sensor ultrassônico captou facilmente a distância e a reprodução de áudio ficou com uma qualidade nítida para seu entendimento.

Palavras-Chave: Deficiente visual. Protótipo. Coleta seletiva. Sensor ultrassom. Arduino.

ABSTRACT

In order to reduce the existing barriers that the visually impaired face day by day, a prototype was developed with the purpose of facilitating the identification of selective collection containers by the visually impaired, through the reproduction of sound effects that can facilitate and make practical their understanding. We used an ultrasound sensor, a microcontroller board called Arduino Mega2560, a micro SD card reader and other electronic devices. The reason for choosing the Arduino Mega was because of its easy workability, it has a great cost benefit and several libraries ready in which they have greatly facilitated the construction of this prototype. The programming language used was C, which is used by Arduino software. The ultrasonic sensor was chosen for its high economic viability and functionality. The programmed condition for the prototype to work was that: As an individual enters the radius of the ultrasound sensor and the preset distance of 200cm or less from the sensor, a sound would be reproduced indicating which type of collection vessel selective visual impairment would be. In the initial idea, it would be used of a load sensor and an amplifier module HX711, however its initial construction was not completed and some substitutions were made. All the tests done for the prototype were attended to, the ultrasonic sensor easily captured the distance and the audio reproduction got a sharp quality for your understanding.

Keywords: Visual impairment. Prototype. Selective collect. Ultrasonic sensor. Arduino.

LISTA DE ILUSTRAÇÕES

Figura 1- Arduino Mega2560.....	14
Figura 2- Ethernet Shield.	15
Figura 3- Arduino Deumilanove	16
Figura 4- Microcontrolador Atmega328p	17
Figura 5- Configurações da placa.....	20
Figura 6 Resistor	21
Figura 7- Capacitores	23
Figura 8- Indutor de ferrite.....	24
Figura 9 Sensor Ultrassônico HC-SR04.....	25
Figura 10- Sensor de carga Strain Gauge.....	27
Figura 11- Módulo HX711.....	28
Figura 12- Software IDE Arduino.....	31
Figura 13- Simulação do sensor ultrassônico.....	36
Figura 14- Virtual Terminal com a distância em centímetros do sensor HC-SR04.....	36
Figura 15- Leitor de Cartão Micro SD	37
Figura 16- Protótipo proposto.	38
Figura 17- Bibliotecas e pinos de entrada	39
Figura 18- Elaboração do Código	40
Figura 19- Elaboração do Código	40
Figura 20- Protótipo proposto	41
Figura 21- Protótipo proposto.	41

LISTA DE TABELAS

Tabela 1- Características do Arduino Mega 2560.....	20
Tabela 2- Código de cores dos resistores	22
Tabela 3- Características do sensor Strain Gauge	28
Tabela 4- Especificações do HX711	29
Tabela 5- Ligação do Cartão Micro SD ao Arduino	38

SUMÁRIO

1. INTRODUÇÃO GERAL	11
1.1 DEFICIÊNCIA VISUAL E SEUS PRINCIPAIS DESAFIOS	12
1.2 OBJETIVOS	13
1.2.1 OBJETIVO GERAL:	13
1.2.2 OBJETIVOS ESPECÍFICOS:	13
2. CONSTRUÇÃO DA PLATAFORMA	14
2.1 INTRODUÇÃO	14
Esse modelo de Arduino mostrado na figura acima, foi o utilizado nesse projeto, o Atmega 2560.	15
2.2.1 HARDWARE DO ARDUINO	15
2.3 MICROCONTROLADORES ATMEL	17
2.3.1 COMPONENTES E CARACTERÍSTICAS PRINCIPAIS DOS MICROCONTROLADORES:	18
2.4 PINOUT E PRINCIPAIS FUNÇÕES DO ARDUINO MEGA	19
2.5 COMPONENTES ELETRÔNICOS BÁSICOS	21
2.5.1 RESISTOR	21
2.5.2 CAPACITOR	23
2.5.3 INDUTOR	24
2.6 SENSORES	25
2.6.1 SENSOR DE PRESENÇA	25
2.6.2 SENSOR DE PRESSÃO	26
3 PROGRAMAÇÃO	29
3.1 INTRODUÇÃO	29
3.2 LINGUAGEM C APLICADA A PLATAFORMA ARDUINO	30
3.3 ARDUINO IDE	30
3.3.1 ESTRUTURA DO CÓDIGO	31
3.3.2 BIBLIOTECAS	32
3.3.3 TIPOS DE VARIÁVEIS E PRINCIPAIS FUNÇÕES	33
4- MATERIAL E MÉTODOS	35
4.1 DEFINIÇÃO DO PROTÓTIPO PROPOSTO	35
4.2- ELABORAÇÃO DO CÓDIGO	39
5- RESULTADOS E DISCURSSÕES	43
6- CONSIDERAÇÕES FINAIS	44
REFERÊNCIAS	45

ELETRÔNICA PROGRESSIVA. **Indutores - O que são, como funcionam e para que são usados.** Disponível em: <<http://www.eletronicaprogressiva.net/2013/07/Indutores-o-que-sao-para-que-servem-e-como-funcionam.html>>. Acesso em: 29 Jan. 2018. 45

APENDICE A- ALGORITMO UTILIZANDO UM SENSOR ULTRASSOM E LEITOR DE CARTÃO MICROSD..... 47

1. INTRODUÇÃO GERAL

O canal de relacionamento mais importante de um indivíduo com o mundo exterior é a visão. Tanto como a audição, ela capta registros próximos ou distantes e permite com que se organizem, no nível cerebral, as informações trazidas pelos outros órgãos dos sentidos (GIL, 2000).

A deficiência visual, em qualquer grau, compromete a capacidade da pessoa de se orientar e de se movimentar no espaço com certa segurança e independência. Na infância, uma criança está desenvolvendo sua capacidade de se socializar, isso prejudica (ou até impede) o conhecimento do mundo ao seu redor e seu relacionamento com outras pessoas.

A acessibilidade é uma condição básica para a inclusão social das pessoas com deficiências ou que tenham necessidades especiais. Em uma sociedade que se utiliza de modernas tecnologias de informação e de comunicação, acessibilidade plena passa a ser uma necessidade constante nos centros urbanos. A lei menciona a necessidade de implantação de dispositivos sonoros nos semáforos para a orientação na travessia de ruas e avenidas com trânsito intenso (art. 9º), o que na maioria das vezes e quase nunca ocorrem, e a dificuldade de um deficiente visual em compreender as coisas ao seu redor se torna difícil em uma era em que a tecnologia é alta.

Com os avanços tecnológicos e as diversas mídias digitais, existe atualmente uma demanda educacional e tecnológica maior, que se evidenciam na forma de utilização. Estamos diante de uma revolução digital e diante deste cenário é possível implementar soluções acessíveis que vão ao encontro da inclusão digital.

Com a utilização da tecnologia e dispositivos eletrônicos, é possível auxiliar um portador de deficiência visual a identificar os recipientes de coleta seletiva. Dessa forma será utilizado o microcontrolador Arduino da Atmel e alguns de seus periféricos como sensores e atuadores.

Segundo Moraes (2003), sensores são transdutores, ou seja, conversores de grandezas físicas em sinais elétricos correspondentes. Com isso será utilizado de sensores para relacionar informações sobre grandezas que precisam ser medidas como pressão e velocidade. Assim, os atuadores são dispositivos que modificara a variável controlada, recebendo o sinal do controlador e agindo sobre o sistema controlado. O microcontrolador Arduino será o dispositivo responsável pelo acionamento dos atuadores de acordo com o estado dos sensores e sua lógica inserida em sua memória.

Assim fazendo o uso de microprocessadores que possuem memórias flash "in system programmable" (ISP), onde pode-se usar em aplicações que envolvam processamento de sinais em tempo real. Uma dessas aplicações que envolvem processamento de sinais em tempo real é o gravador de voz. Onde numa aplicação desse tipo, o microprocessador converte sinais analógicos, como o que corresponde aos sons captados por um microfone em sinais digitais que são armazenados na memória flash integrada para que depois, a qualquer instante esses sinais possam ser recuperados por um conversor digital para analógico, amplificados e reproduzidos.

1.1 DEFICIÊNCIA VISUAL E SEUS PRINCIPAIS DESAFIOS

Enxergar não é uma habilidade inata, ou seja, ao nascer ainda não sabemos enxergar e que é preciso aprender a ver. O processo não é consciente. Sem pensar, tenta-se ensinar um bebê a enxergar as coisas mostrando tudo ao seu redor. Assim, a visão é desenvolvida nos primeiros anos de vida. (GIL, 2000)

Ainda de acordo com o autor, a cegueira ou perda total da visão pode ser adquirida ou congênita (desde o nascimento). O indivíduo que nasce com o sentido da visão, perdendo-o mais tarde, guarda memórias visuais, assim conseguindo lembrar das imagens, cores e luzes que chegou a conhecer, sendo assim bastante útil para sua readaptação. Já quem nasce sem a capacidade de "ver", por outro lado, jamais poderá formar uma memória visual ou possuir lembranças visuais.

Da população total brasileira, 23,9% (45,6 milhões de pessoas) declararam ter algum tipo de deficiência. Dentre as deficiências declaradas, a mais comum foi a visual, chegando a atingir 3,5% da população segundo a Organização mundial de saúde (OMS).

As principais causas de cegueira no Brasil segundo a Organização Mundial da Saúde (OMS) são: catarata, glaucoma, retinopatia diabética, cegueira infantil e degeneração macular.

Em dados do IBGE de 2010, no Brasil, das mais de 6.5 milhões de pessoas com alguma deficiência visual: 528.624 são incapazes de enxergar (cegos); 6.056.654 pessoas possuem baixa visão ou visão subnormal (grande e permanente dificuldade de enxergar). Já sobre a deficiência visual no mundo, dados do World Report on Disability 2010 e Vision 2020, a cada 5 segundos 1 pessoa se torna cega no mundo. Além disso, estima-se que até 2020, o número de pessoas com deficiência visual dobre no mundo.

Com os avanços tecnológicos e as diversas mídias digitais, existe atualmente uma demanda educacional e tecnológica maior, que se evidenciam na forma de utilização. Estamos diante de uma revolução digital e diante deste cenário é possível implementar soluções acessíveis que vão ao encontro da inclusão digital. Com isso, um novo mundo de conteúdo universal acessível com imagens, vídeos e áudios as pessoas com deficiência podem ser proporcionadas oportunidades de acessibilidades mais qualificadas.

Com a constante evolução tecnológica, se torna muito mais fácil a prática de dispositivos elétricos para o auxílio na identificação de dispositivos de coleta seletiva para deficientes visuais.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL:

Confecionar um protótipo de um sistema elétrico audível para auxílio de portadores de deficiência visual para a identificação dos recipientes de disposição correta de resíduos sólidos.

1.2.2 OBJETIVOS ESPECÍFICOS:

- Definir sensores necessários;
- Compreender sua utilização junto ao microcontrolador;
- Construção de um protótipo;

2. CONSTRUÇÃO DA PLATAFORMA

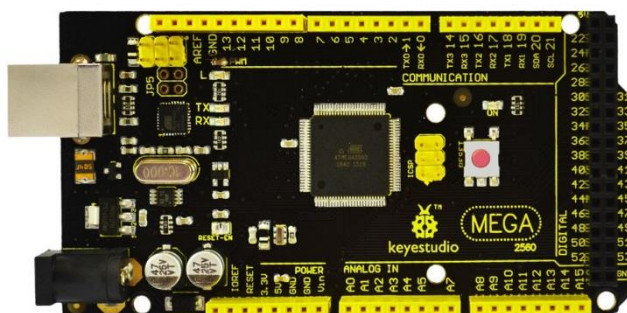
2.1 INTRODUÇÃO

O conceito Arduino surgiu na Itália no ano de 2005, com o objetivo de criar um dispositivo para controlar projetos e protótipos construídos de uma forma mais acessível do que outros sistemas disponíveis no mercado (SILVA, 2014).

O Arduino é uma plataforma de prototipação eletrônica de código aberto (*open-source*), qualquer pessoa pode produzi-la, é prático de se programar e usar, além de possuir preço acessível e ser facilmente encontrado (LIMA; VILLAÇA, 2012). Projetada com um microcontrolador Atmel-AVR de 8 bits com suporte de entrada/saída embutido, utiliza uma linguagem de programação padrão *Wiring* (C/C++).

Com sua boa praticidade, tudo que é preciso para começar a trabalhar com o Arduino é ter o projeto com que quer trabalhar em seu computador ou baixar e conectar diretamente ao Arduino através do cabo *Universal Serial Bus* (USB). Por ser um dispositivo open source qualquer pessoa pode modificar o Arduino de acordo com suas necessidades. Seguinte figura 1 mostra um modelo Arduino Mega.

Figura 1- Arduino Mega2560



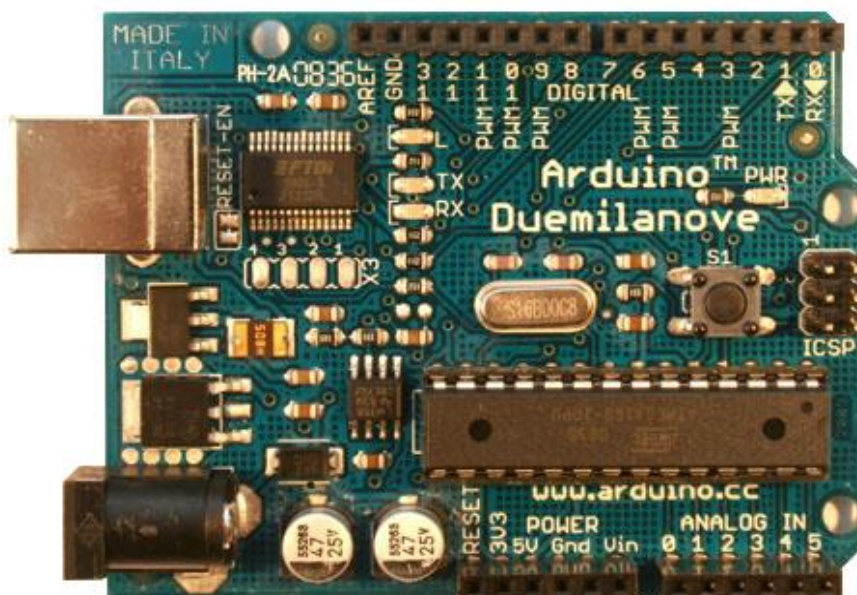
FONTE: Aliexpress, 2017.

Outra parte bastante importante é a alimentação do Arduino, que pode ser feito de duas formas, seja por uma fonte externa (bateria ou transformador) ou diretamente nos pinos correspondentes. O modo mais prático de energizar a placa Arduino é através do cabo USB, onde há uma linha de alimentação de 5v adequado ao microcontrolador.

2.2.1.1 MODELOS DE ARDUINO

Existem vários tipos de Arduino disponíveis no mercado com suas características próprias, exemplo: Arduino Uno, Deumilanove, Arduino Nano, Arduino Promini, Arduino Pro, Arduino Mega ADK, o utilizado nesse trabalho, Arduino Mega2560 entre outros. A seguir, um exemplo de Arduino.

Figura 3- Arduino Deumilanove



FONTE: Arduino.cc, 2018.

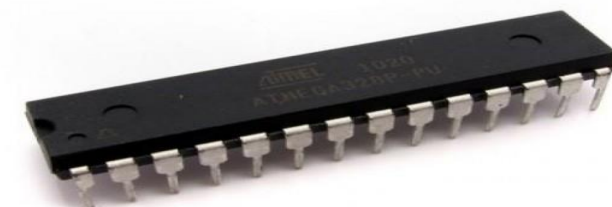
Na imagem acima é apresentada um exemplo de Arduino, o Deumilanove.

2.3 MICROCONTROLADORES ATMEL

O microcontroller unit (MCU) segundo Penido e Trindade (2013), é basicamente um computador em um único *chip*, que contém um processador, memória, periféricos de entrada e saída, temporizadores, dispositivos de comunicação serial, dentre outros. Os microcontroladores, surgiram como uma evolução natural dos circuitos digitais devido ao aumento da complexidade dos mesmos.

Os microcontroladores estão presentes em quase tudo que envolve a eletrônica, pois com a evolução natural dos circuitos digitais e sua complexibilidade, os microcontroladores além de serem pequenos e apresentarem a melhor relação custo/benefício. O microcontrolador detém de uma unidade central de computação (UCP) em um único encapsulamento (circuito integrado), possui memória, oscilador interno de *clock*, entradas e saídas para interação com o utilizador. A figura 4 a seguir representa um microcontrolador Atmega.

Figura 4- Microcontrolador Atmega328p



FONTE: Baudaeletronica, 2018.

A seguir alguns exemplos de microcontroladores em eletrônica embarcada:

- TVs/Monitores;
- Aparelhos de DVD e Rádio;

- Telefone sem fio e celulares;
- Relógios;
- Forno Micro-ondas;
- Geladeiras/Refrigeradores.

2.3.1 COMPONENTES E CARACTERÍSTICAS PRINCIPAIS DOS MICROCONTROLADORES:

Os microcontroladores são formados por diversos componentes e cada um deles têm características próprias, a seguir serão apresentados os principais componentes e suas respectivas características que juntos, formam a maioria dos microcontroladores.

2.3.1.1 PROCESSADOR

Os microcontroladores possuem internamente uma CPU (Unidade Central de Processamento) que está interligado com outros blocos do microcontrolador, tendo como principal função gerenciar os blocos interligados a ela e executar as instruções do programa de acordo com os valores de entrada e atualizando os valores de saída. (SANTANA, 2013)

2.3.1.2 MEMÓRIAS

A memória tem como função principal, armazenar valores de variáveis para serem utilizadas na lógica de programação. As memórias podem ser subdivididas em duas categorias principais: memória de programa e memória de dados.

A categoria memória de programa é o local onde serão carregados os programas em execução, para assim os programas executar e gerenciar várias instruções. Já a segunda categoria memória de dados, será o local de armazenamento de resultados que a unidade de processamento executa. (SANTANA, 2013)

2.3.1.3 PORTAS DE ENTRADAS E SAÍDAS

O microcontrolador precisa interagir com o mundo exterior e para isso ela utiliza-se de portas de entradas e saídas de dados. Tais portas são componentes registradores interligados aos pinos do microcontrolador para que os mesmos sejam programados, essas portas podem ser portas de entradas, saída ou bidirecional (entrada e saída) de dados.

2.3.1.4 TEMPORIZADORES E CONTADORES

Os contadores podem ser chamados também temporizadores e são uma das mais importantes sub-funções de um microcontrolador. Estes permitem processos temporais, gerar sinais e contar os eventos. Geralmente utilizados para funções de retardo ou marcador de pulsos, sendo utilizados e controlados através de registradores internos.

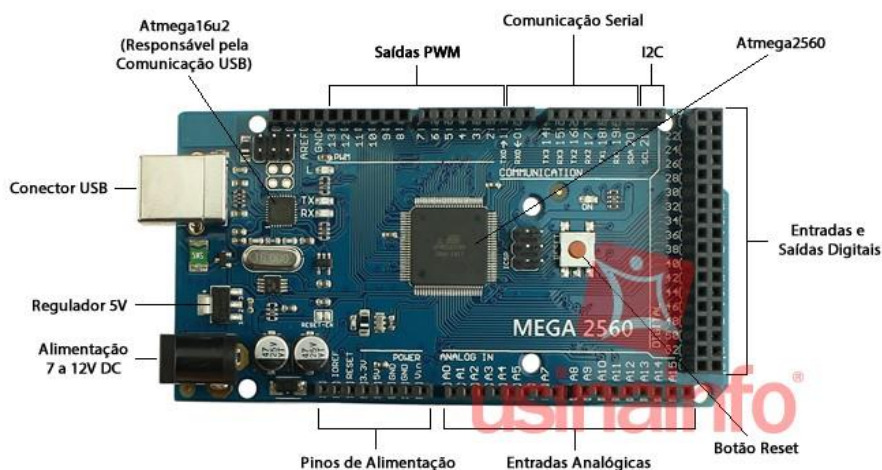
2.3.1.5 UNIDADE LÓGICA ARITMÉTICA

A unidade lógica e aritmética (ULA) é um circuito digital que realiza operações lógicas e aritméticas. A ULA executa as principais operações lógicas e aritméticas do computador, exemplos: somar, subtrair, dividir, determinar se um número é positivo ou negativo ou se é zero.

2.4 PINOUT E PRINCIPAIS FUNÇÕES DO ARDUINO MEGA

A placa Arduino Mega 2560 possui recursos interessantes para prototipagem e projetos mais elaborados. Se baseia no microcontrolador ATmega2560, possui 54 pinos de entradas e saídas digitais onde 15 destes podem ser utilizados como saídas PWM. Possui 16 entradas analógicas, 4 portas de comunicação serial. Além de sua quantidade de pinos, ela conta com uma maior quantidade de memória do que o Arduino UNO, sendo uma ótima opção para projetos que necessitem de muitos pinos de entradas e saídas além da memória de programa com maior capacidade.

Figura 5- Configurações da placa



FONTE: Usinainfo, 2018.

Sua alimentação externa é feita através do conector Jack com positivo no centro, onde o valor da tensão da fonte externa deve estar entre 7 V a 12 V para não correr o risco de ficar com tensão alimenta instável ou ocorrer o risco de sobreaquecer e danificar a placa. A seguinte tabela dará mais alguns detalhes específicos do microcontrolador Mega 2560.

Tabela 1- Características do Arduino Mega 2560

Características	Descrição
Microcontrolador	ATmega2560
Voltagem de alimentação	5V
Voltagem de entrada (recomendada)	7-12V
Voltagem de entrada (limites)	6-20V
Pinos Digitais I/O	54 (dos quais 14 podem ser saídas de PWM)
Pinos de entrada analógica	16
Corrente continua por pino I/O	40mA
Corrente continua para o pino 3.3V	50mA
Memória Flash	256 KB dos quais 4 KB utilizados pelo carregador de inicialização

SRAM	8 KB
EEPROM	4 KB
Velocidade de <i>Clock</i>	16MHZ

FONTE: Autor (adaptado de lojaarduino, 2018).

Cada um dos 54 pinos digitais podem ser usados como entrada ou saída, através das funções *pinMode()*, *digitalRead()* e *digitalWrite()*. Existem dois tipos de pinos diferentes na placa:

- AREF: Tensão de referência para as entradas analógicas.
- Reset: Reseta o microcontrolador. Usado para adicionar um botão de *reset* para *shields* que bloqueiam o que há na placa.

2.5 COMPONENTES ELETRÔNICOS BÁSICOS

2.5.1 RESISTOR

O resistor é um dos componentes mais utilizados na eletrônica, com finalidade primária de limitar o fluxo de corrente elétrica em um determinado circuito. Devido a sua função de resistir a passagem de corrente elétrica, a maior parte dos resistores são feitos com carvão em pasta, devido a ser um bom isolante elétrico (SILVA JÚNIOR, 2017).

A limitação da corrente elétrica feita pelos resistores ocorre pela transformação de energia elétrica em calor. Quando os elétrons em movimento (corrente elétrica) se chocam com os átomos que formam o material condutor, o atrito gera calor, e esse fenômeno é denominado de Efeito Joule (SILVA JÚNIOR, 2017).

Os resistores estão presentes em grande parte dos equipamentos utilizados no dia a dia, exemplo disso são os chuveiros elétricos. A figura 6 mostra um exemplo de resistor.

Figura 6 Resistor



FONTE: Bibliotecadaengenharia, 2013.

Existem duas formas de conhecer o valor de um resistor:

- medir a resistência com um multímetro;
- ler o valor direto do corpo do resistor.

Os valores vêm codificados em cores, e conhecer o código de cores possibilita a leitura desses valores. A seguinte tabela mostra a convenção utilizada para identificação de resistores de uso geral.

Tabela 2- Código de cores dos resistores

Código de Cores

A extremidade com mais faixas deve apontar para a esquerda

Cor	1ª Faixa	2ª Faixa	3ª Faixa	Multiplicador	Tolerância
Preto	0	0	0	x 1 Ω	
Marrom	1	1	1	x 10 Ω	+/- 1%
Vermelho	2	2	2	x 100 Ω	+/- 2%
Laranja	3	3	3	x 1K Ω	
Amarelo	4	4	4	x 10K Ω	
Verde	5	5	5	x 100K Ω	+/- .5%
Azul	6	6	6	x 1M Ω	+/- .25%
Violeta	7	7	7	x 10M Ω	+/- .1%
Cinza	8	8	8		+/- .05%
Branco	9	9	9		
Dourado				x .1 Ω	+/- 5%
Prateado				x .01 Ω	+/- 10%

FONTE: Ebah, 2016.

Na tabela acima podemos calcular a resistência do resistor que começa com a cor verde da seguinte maneira:

1º Faixa: A cor verde mostra o primeiro algarismo do valor da resistência;

2º Faixa: A cor azul mostra o segundo algarismo da resistência;

3º Faixa: A cor amarela representa seu multiplicador;

Como as primeiras duas cores representam o 5 e o 6 respectivamente e o multiplicador da cor amarela representa o 10, então juntando o 5 e o 6 e multiplicando por 10, temos a resistência de 560 Ohms.

2.5.2 CAPACITOR

Segundo Graça (2012, p. 89) dois corpos condutores quaisquer, separados por um isolante, formam um capacitor, cuja característica física é sua capacitância, ou propriedade de armazenar carga elétrica eficientemente. Onde esse mesmo material isolante em alguns casos dá nome ao próprio capacitor (capacitores de poliéster, cerâmicos, de vidro, etc). Figura 7, mostra exemplos de capacitores.

Figura 7- Capacitores



FONTE: Eletronicadidatica, 2018.

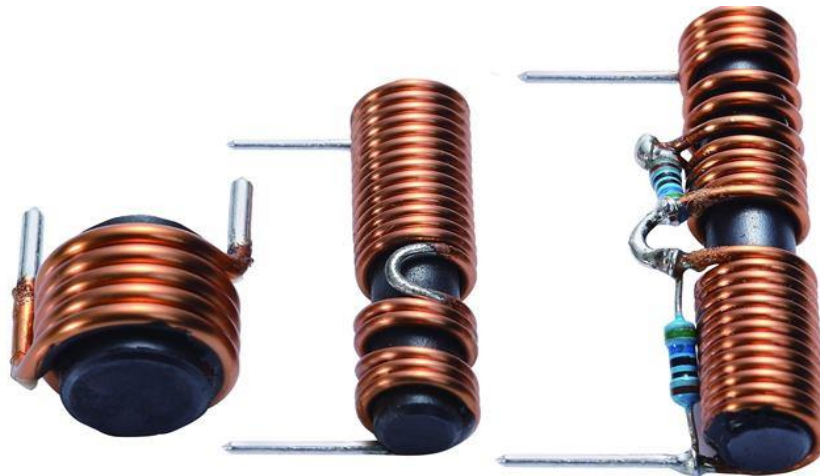
A capacitância determina a quantidade de carga capaz de ser armazenada por um capacitor, cuja unidade é dado em **Farad (F)**. Porém 1 Farad (1 F) é tido como uma capacitância muito alta, por isso normalmente se ver unidades menores como microFarad (uF), nanoFarad (nF), etc.

2.5.3 INDUTOR

Os indutores são basicamente bobinas, enrolamentos de fios colocados em torno de um núcleo, que é usado em circuitos elétricos, eletrônicos e digitais com a determinada função de acumular energia através de um campo magnético. Os indutores são usados também para formar um transformador e para impedir variações na corrente elétrica. Algumas de suas características segundo Gussow (1997, p.309):

- A indutância aumenta com o número de espiras em torno do núcleo;
- A indutância aumenta com a permeabilidade relativa do material com o qual é feito o núcleo;
- À medida que a área abrangida em cada espira aumenta, a indutância aumenta;
- A indutância diminui à medida que o comprimento da bobina aumenta (desde que o número de espiras permaneça constante). A figura 8 mostra um indutor de ferrite.

Figura 8- Indutor de ferrite



FONTE: Mecanicaindustrial, 2016.

Na imagem 8 acima é apresentado o indutor de ferrite, onde devido ser feito por um tipo de cerâmica ferrimagnética não condutora, não apresenta correntes parasitas, além de baixa histerese. São empregas em alta frequência, onde o material irá apresentar um rendimento melhor.

2.6 SENSORES

Sensores são dispositivos capaz de captar ações ou estímulos externos e responder em consequência. Ou seja, podem transformar grandezas físicas ou químicas em grandezas elétricas. Os sensores podem ser classificados como um tipo de transdutor. Segundo Patsko (2006) um transdutor é um componente que transforma um tipo de energia em outro. Exemplo é um motor, que converte energia química ou elétrica em energia mecânica.

2.6.1 SENSOR DE PRESENÇA

O sensor utilizado nesse trabalho, será o sensor ultrassônico HC-SR04. No qual, o sinal emitido, ao colidir com qualquer obstáculo, é refletido de volta na direção do sensor. Durante todo esse processo, o aparelho age como uma espécie de “cronômetro” de alta precisão funcionando. Assim, dá para descobrir quanto tempo o sinal gastou desde sua saída até o seu retorno (ARAÚJO, 2017). Figura a seguir mostra um exemplo do sensor.

Figura 9 Sensor Ultrassônico HC-SR04



FONTE: Filipeflop, 2018.

Como se sabe a velocidade do som no ar, é possível, de posse do tempo que o sinal levou para ir até o obstáculo e voltar, o cálculo entre a distância do sensor e o obstáculo. Afim de resolver esse cálculo, será considerado a velocidade do som no ar (340 m/s) na seguinte equação:

$$d = \frac{(V \times T)}{2}$$

Onde:

- *d = Distância entre o sensor e o obstáculo (no qual se quer descobrir);
- *V= Velocidade do som no ar (340 m/s);
- *t = Tempo necessário para o sinal ir do sensor até o obstáculo e voltar (é o que o módulo sensor ultrassônico mede).

A divisão por dois se dá pelo fato do tempo medido pelo sensor ser para ir e voltar, ou seja, duas vezes a distância que queremos descobrir.

O sensor HC-SR04 é muito prático e fácil de ser utilizado com o Arduino, além de ter boa precisão e ser bastante acessível. A seguir, algumas características do sensor ultrassônico HC-SR04:

- VCC = Alimentação do modulo com +5 V.
- TRIG= Gatilho para disparar o pulso ultrassônico.
- ECHO= Gera um pulso com a duração do tempo necessária para o eco do pulso ser recebido pelo sensor.
- GND= Terra.

2.6.2 SENSOR DE PRESSÃO

Os sensores de pressão segundo Whatley (2011) são utilizados para medir a pressão. Sensores de pressão funcionam como um aviso quando a pressão aplicada é muito grande ou muito pequena. O sensor de pressão é um transdutor que converte a

pressão em um sinal elétrico analógico. A seguinte figura mostra o sensor de carga utilizado nesse projeto.

Figura 10- Sensor de carga Strain Gauge



FONTE: Filipeflop, 2018.

O sensor de carga chamado de Strain Gauge utilizado nesse projeto, é capaz de medir até 50 kg. Esse sensor é um dos vários utilizados na medição de peso. Seu funcionamento se dá através da mudança na resistência elétrica do sensor proporcional a uma determinada pressão ou força aplicada a ele (daí se dá o nome “Strain”). A sensibilidade de cada sensor é bastante diferente e são expressas pelo chamado *Gauge Factor*(GF). Sendo este, definido como a razão entre uma mudança na resistência elétrica em relação a uma mudança no comprimento provocada por uma força externa (VIDAL, 2018).

Essa célula de carga é um sensor que funciona como meia ponte, ou seja, utiliza de uma meia ponte de **Wheatstone** onde existirá uma resistência de referência e um elemento sensor cuja resistência irá variar de acordo com a pressão aplicada. Utilizando de duas células de carga, têm-se uma ponte de Wheatstone completa, sendo que uma célula funcionará para medir compressão e outra para medir tensão (forças aplicadas em diferentes direções).

Uma dificuldade bastante comum desses sensores de pressão, é que as deformações provocadas pela aplicação de uma força ou pressão externa são muito pequenas, causando variações na escala de décimos de ohms (VIDAL, 2018). Devido essas variações de ordem bastante pequenas, é necessário a utilização de um segundo componente, que agirá como um amplificador de sinais. Existem alguns modelos

comerciais, mas nesse projeto será utilizado o amplificador HX711. A seguir, algumas características do sensor Strain Gauge.

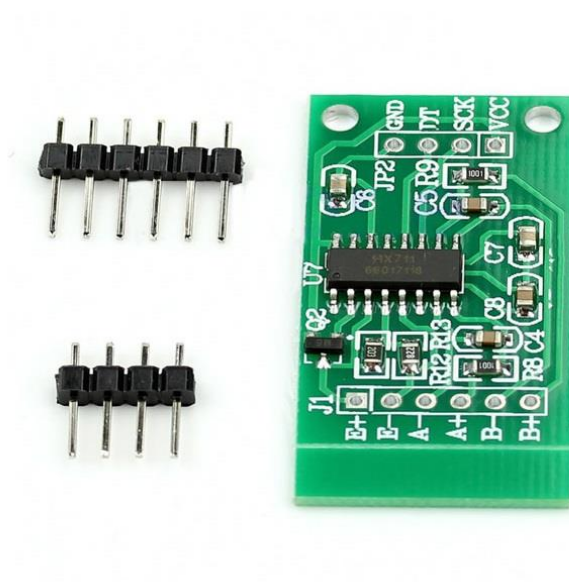
Tabela 3- Características do sensor Strain Gauge

Dimensões (mm)	28*28*8
Capacidade (Kg)	50
Margem de erro	0.2%
Tensão de excitação (V)	5~10

FONTE: Autor (adaptado de VIDAL, 2018).

Se tratando de um sensor ativo, ele é alimentado com um sinal de excitação externo. Sua saída é dada em volts. Devido a pequena variação na resistência do elemento sensor, a variação na tensão de saída também é muito pequena. Sendo esse sinal de tensão de saída que será conectado ao amplificador HX711.

Figura 11- Módulo HX711



FONTE: Filipeflop, 2018.

O módulo conversor e amplificador HX711 de 24bits é utilizado para amplificar o sinal de dispositivos como as células de carga, fazendo a interligação entre essas células e o microcontrolador (VIDAL, 2018). Algumas especificações do HX711:

Tabela 4- Especificações do HX711

Tensão de operação (V)	4,8 a 5,5 DC
Corrente de operação (mA)	1,6
Temperatura de operação	-20 a 85°C
Interface	SPI
Dimensões (mm)	29 x 17 x 4 (sem os pinos)

FONTE: Autor (adaptado de VIDAL, 2018).

A ligação é feita da seguinte maneira, o fio vermelho é o sinal de saída ligado ao pino **A-** do HX711. Para uma melhor identificação, é feita a medição das resistências entre os fios da célula para saber qual é qual. Os fios cujo sua resistência for igual a 2 K, devem ser conectados ao HX711 nos pinos **E+** e **E-** que na maioria das vezes são os fios branco e preto respectivamente. O fio no qual obtiver uma leitura de 1 k, deve-se conectar ao **A+** (geralmente fio vermelho). Assim, o pino **A+** do HX711 será ligado ao ponto comum entre os dois resistores de 1k (VIDAL, 2018).

3 PROGRAMAÇÃO

3.1 INTRODUÇÃO

Os computadores atualmente ainda não são capazes de entender a linguagem natural que utilizamos no dia a dia, então é preciso de outro “idioma” (linguagem) para instruir os computadores a fazerem as tarefas que desejamos. Sendo assim, nós seres humanos precisamos converter nossas ideias para uma forma que os computadores possam processar, ou seja, a linguagem de máquina. Sendo que existem várias linguagens de programação diferentes. (CHAVIER, 2017)

Para toda e qualquer programação, tem como base uma nova linhagem chamada *processing*. Existem várias funções para o ambiente de desenvolvimento que facilitam a progressão de qualquer software, pois possuem já prontas, bibliotecas que facilitam a comunicação com outros hardwares. Como já falado anteriormente, o ambiente de desenvolvimento do Arduino é um compilador gcc (C/C++) que utiliza de uma interface gráfica construída em Java. As funções da IDE do Arduino são basicamente de permitir o desenvolvimento de um software e enviar o código a placa para ser executado. (REBESCHINI, 2012 *apud* TÓFOLI, 2014)

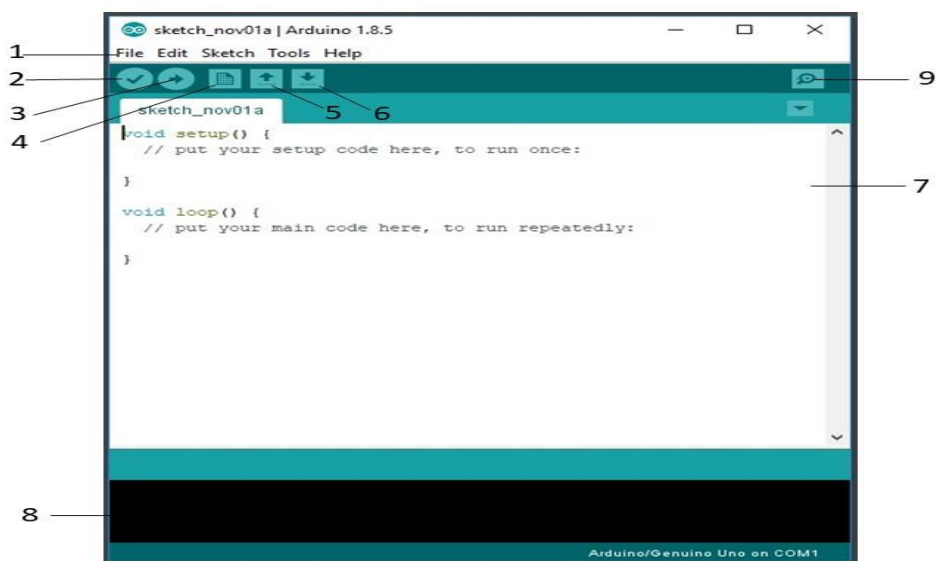
3.2 LINGUAGEM C APLICADA A PLATAFORMA ARDUINO

Como os algoritmos voltados a placa Arduino são feitos em *Wiring* (C++), ela detém de um menor acervo de bibliotecas e funções. Comparando a linguagem *Wiring* com a linguagem C que é naturalmente mais utilizada, pode-se constatar uma restrição no acervo de funções e bibliotecas para esta plataforma, devido sua aplicação em projetos que não são tão complexos.

3.3 ARDUINO IDE

A IDE (ambiente integrado de desenvolvimento), é onde será feito os programas que serão gravados no microcontrolador e sua linguagem de utilizada para programar é a *Wiring* e é baseada na linguagem C/C++, que é utilizado por diversos microcontroladores, ao se fazer upload do *sketch*, nome dado ao código utilizado pelo Arduino, feito na IDE. A figura 12 mostra a IDE do Arduino.

Figura 12- Software IDE Arduino



Fonte: Autor (adaptado do IDE.2018).

As funções do software IDE Arduino mostrada na figura 12 são mostradas a seguir:

- 1: Funções do menu principal;
- 2: Compilar/Verificar;
- 3: Gravar o programa na placa;
- 4: Novo programa;
- 5: Abrir programa;
- 6: Salvar Programa;
- 7: Área de programação;
- 8: Console do compilador;
- 9: Monitor serial.

No canto inferior da IDE existe um console capaz de reportar as mensagens de erros e as respectivas mensagens relacionadas ao código. O microcontrolador utilizado nesse protótipo foi da ATMEL, onde a IDE Arduino possui um compilador que transforma o *sketch* (programa) em um arquivo objeto, sendo esse compilador chamado de *avr-gcc*.

3.3.1 ESTRUTURA DO CÓDIGO

O Arduino detém de uma estrutura bastante simples na linhagem de programação. São basicamente formadas por dois blocos de funções que carregam outros blocos de

funções escritas em linguagem C/C++. O primeiro bloco de funções forma a função void **setup()**, já o segundo, a função void **loop()**. (SILVEIRA, 2012)

Mas o que são funções? Funções em linguagens de programação são como sub-rotinas ou procedimentos, pequenos blocos de programas usados para manter o programa principal. São desenvolvidas pelo próprio programador a fim de realizar tarefas repetitivas, ou podem ainda serem importadas prontas para o programa em forma de bibliotecas. (SILVEIRA, 2012)

Seguindo a partir disso, toda função deve ser declarada antes mesmo de ser chamada, atribuindo-lhe um tipo e um nome seguido de parênteses, onde serão colocados todos os parâmetros da passagem da função. Depois do nome, são definidos os procedimentos que a função irá executar entre as chaves ‘{’ e ‘}’. Dito isso, pode ser chamado a primeira função quando o programa se inicia: **setup()**. Ela é executada no início e apenas uma vez. Ela é uma função de preparação, pois dita o comportamento dos pinos do Arduino e inicializa a porta serial.

Já a segunda função **loop()** é chamada logo a seguir para que todas as funções embarcadas nela sejam executadas repetidamente. Ela ler os pinos de entrada do Arduino e comanda os pinos de saída e a porta serial.

3.3.2 BIBLIOTECAS

Como a plataforma é open source, existem vários meios de facilitar ainda mais o processo de se criar um programa/código, chamado bibliotecas. As bibliotecas (*library*) fornecem a possibilidade de usar um conjunto de funções pré-determinadas por outros programadores que já resolveram determinado problema, sem que precise você reinventar algo no qual já foi resolvido, assim, apenas modificando e adaptando ao programa que você quer criar. Existem vários tipos de bibliotecas associados a vários problemas específicos, exemplos:

- OpenCV: Utiliza de um conjunto de funções para processamento de imagem e visão computacional;
- OpenAL: Conjunto de funções para processamento, síntese e execução de áudio;
- SD: Leitura e escritura de cartões SD;
- Wifi: Conecta o Arduino com a internet Wifi *Shield*;

- Servo: Controle de motores servo;
- OpenCV: Processamento de imagens e vídeos;
- Ultrassom: Biblioteca do sensor ultrassônico;
- Wire: para envio e recebimento de dados acerca de sensores ou aparelhos;
- Ethernet: Conecta o Arduino com a internet ou rede local através do *shield* Ethernet;
- GSM: Conecta a uma rede GSM/GPRS usando o *shield* GSM;
- EEPROM: Ler e grava dados em uma memória EEPROM no Arduino;
- LiquidCrystal: Controla displays de cristal líquido (LCD).

As principais bibliotecas que foram utilizadas nesse protótipo foram as bibliotecas *SD* e *TMRpcm*, onde a biblioteca *SD* fará a leitura do cartão microSD e a biblioteca *TMRpcm* que fará a leitura dos áudios reproduzidos em formato *wav* e padrão *PCM*.

3.3.3 TIPOS DE VARIÁVEIS E PRINCIPAIS FUNÇÕES

Variável é um local onde permite guardar qualquer tipo de dado na memória, ou seja, é uma forma de dar nome e guardar um valor para ser utilizado no decorrer do programa. Para toda variável, precisa-se ter um tipo para que possa ser determinado o que ela será capaz de guardar em sua memória. Os tipos mais comuns de dados utilizados são (CHAVIER, 2016):

- *Boolean*: valor verdadeiro (*true*) ou falso (*false*);
- *Char*: um caractere;
- *String*: sequência de caracteres;
- *Double*: número real de precisão dupla;
- *Float*: número real de precisão simples;
- *Byte*: um byte, ou sequência de 8 bits
- *Int*: número inteiro de 16 bits com sinal (-32768 a 32767)
- *Long*: número inteiro de 16 bits com sinal (-2147483648 a 2147483647).

Uma função é uma sequência de comandos que pode ser reutilizada várias vezes ao longo de um algoritmo. Para se criar uma função, é necessário fazer uma “declaração de função”. Exemplo:

```
void setup() {  
    pinMode(led, OUTPUT);  
}
```

A função declarada foi a *setup()*. Ela executa os comandos de outra função: *pinMode()*. Esta ação de executar comandos de uma função inicialmente declarada é denominada de **chamada de função**. Não é preciso declarar a função *pinMode()*, pois no caso do Arduino ela já é declarada automaticamente (CHAVIER, 2016).

4- MATERIAL E MÉTODOS

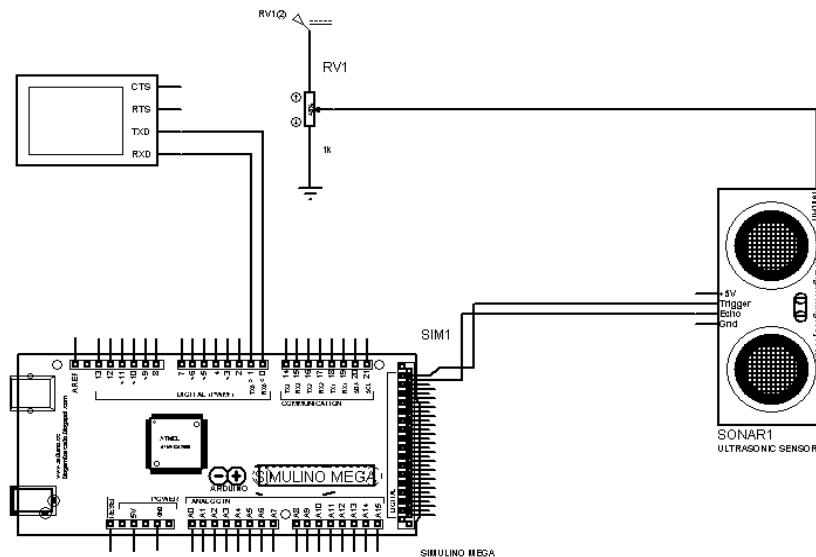
Para a realização deste trabalho foram utilizados os seguintes materiais e equipamentos:

- Software Arduino IDE versão 1.8.5;
- 1 Arduino Mega2560;
- 1 Sensor Ultrassônico;
- Software Proteus 8;
- 1 Leitor de Cartão MicroSD;
- 1 Cartão de Memória MicroSD;
- 4 botões para Arduino;
- 4 Resistores de 240 ohm cada;
- 1 caixa de Som;
- 1 Microcomputador, processador intel core i3 com 4 gigabytes de memória RAM;
- Software Audacity;
- Software Easy Voice Recorder.

4.1 DEFINIÇÃO DO PROTÓTIPO PROPOSTO

Inicialmente para a obtenção dos primeiros resultados, foi utilizado do software *proteus* para a simulação do sensor ultrassônico HC-SR04 e da IDE Arduino para a simulação do código feito em C. As bibliotecas utilizadas para essa simulação foram: *Ultrasonic.h* e *Arduino 4.0*.

Figura 13- Simulação do sensor ultrassônico



FONTE: Autor, 2018

No canto direito da figura, podemos ver o sensor utilizado HC-SR04, onde possui 4 pinos de conexão sendo eles: a) o *Vcc*, que é o pino onde rebe a alimentação de 5V; b) o *Trigger*, onde será enviado o sinal sonoro; c) o *ECHO*, onde recebe o sinal sonoro; e d) o *GND*, que é o aterramento ou pino negativo. Sendo assim, o *VCC* é alimentado por uma tensão de 5V do próprio Arduino (canto inferior esquerdo da tela), o pino *Trigger* (emissor) e o pino *Echo* (receptor) são conectados respectivamente aos pinos 22 e 23 do Arduino Mega.

Figura 14- Virtual Terminal com a distância em centímetros do sensor HC-SR04

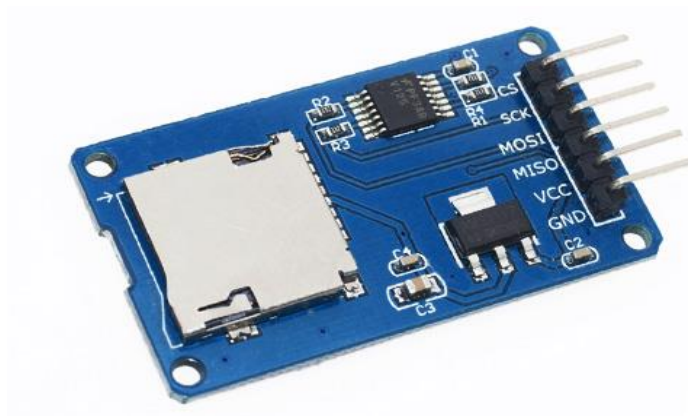


FONTE: Autor, 2018

No Virtual Terminal da imagem acima, mostra o resultado da simulação do sensor HC-SR04 no Proteus, convertendo sua distância em centímetros (cm). Onde se variar a resistência, muda a distância observada no Virtual Terminal.

Após concluir o processo de simulação no Proteus do sensor ultrassônico, foi colocado 4 botões para simular a carga, que de primeira mão, seria o sensor de carga e o módulo HX711 que faria a pesagem do recipiente que o deficiente visual traria para medir e descobrir em qual tipo de coleta seletiva iria se enquadrar, seja papel, plástico, metal ou vidro. Com os botões simulando a carga, faltaria apenas como reproduzir o resultado dessa simulação de carga em áudio. Para isso foi utilizado um Leitor de Cartão Micro SD, para reproduzir de forma audível o que se espera do protótipo.

Figura 15- Leitor de Cartão Micro SD



FONTE: multilogica-shop.com

Primeiramente, foi feita uma formatação no cartão micro SD de memória para *FAT32*. Posteriormente foi gravado áudios no programa ***Easy Voice Recorder*** no qual todos foram gravados no formato ***wave*** utilizando o padrão ***PCM*** que trabalha de forma facilmente com o Arduino. Apesar da facilidade de utilizar esses arquivos de áudio, um problema constatado foi a baixa qualidade de som reproduzido. Todos os áudios foram gravados em uma taxa de amostragem de 8 KHz, 11 KHz, 16 KHz e 22 KHz para teste. Porém, para a teoria da amostragem, têm-se que a qualidade do som seria resultado da multiplicação entre a *amostra de frequência* e a *resolução da amostra*. Porém, ainda sim os áudios apresentavam uma baixa qualidade em todas as taxas de amostragem onde só mudou após converter os áudios para 8 bits de resolução. Nessa conversão, outro

programa foi utilizado, o **Audacity**, onde foi convertido todos os áudios de 8 á 22KHz para 8bits de resolução.

Dado os áudios prontos para reprodução, foi feito a interligação do leitor de Cartão micro SD com o Arduino. Tendo sua ligação feito da seguinte maneira:

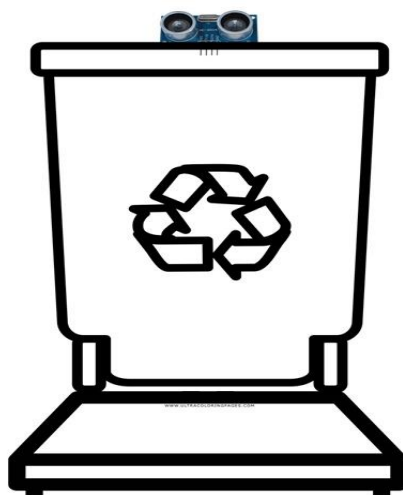
Tabela 5- Ligação do Cartão Micro SD ao Arduino

Arduino Mega	Cartão SD
50	MISO
51	MOSI
52	SCK
53	CS
GND	GND
5 volts	VCC

FONTE: Aduinovacao, 2016.

Como mostrado na tabela acima, o Vcc vai alimentado nos 5V do Arduino, o *GND* no *GND* e dos pinos 50 ao 53 são alimentados respectivamente no *MISO*, *MOSI*, *SCK* e *CS*. A biblioteca utilizada para leitura dos arquivos no formato *wave* e padrão *PCM*, foi a biblioteca *TMRpc*.

Figura 16- Protótipo proposto.



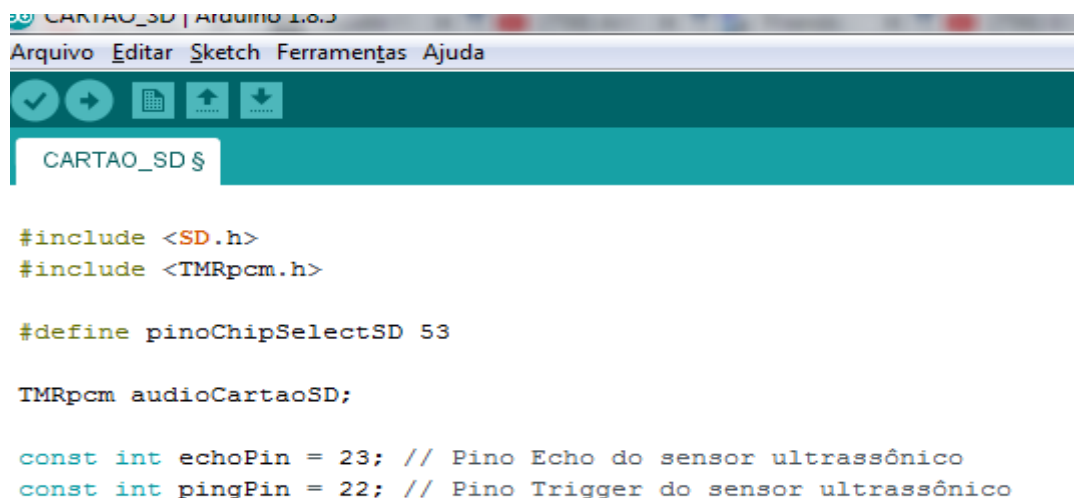
FONTE: Autor, 2018.

O protótipo proposto funcionaria como na imagem acima, o sensor permaneceria acima do recipiente de coleta seletiva na medida que a balança de peso ficaria abaixo do recipiente, só que um pouco a frente.

4.2- ELABORAÇÃO DO CÓDIGO

A elaboração completa do código do protótipo proposto foi feita através do software IDE do Arduino, baseado na linguagem C de programação. Inicialmente foi adicionado as bibliotecas *SD* para o Cartão Micro SD, depois foi adicionado a biblioteca *TMRpcm* na qual faz a leitura dos áudios no formato *WAV* e padrão *PCM*. Logo depois é definido o pino que irá selecionar o cartão micro SD no Arduino que foi o 53. Após isso, foi definido os pinos *ECHO* e *TRIGGER* do sensor ultrassônico ligados ao Arduino, que foram os pinos 23 e 22 respectivamente.

Figura 17- Bibliotecas e pinos de entrada



```

CARTAO_SD | Arduino 1.8.5
Arquivo Editar Sketch Ferramentas Ajuda

CARTAO_SD $

#include <SD.h>
#include <TMRpcm.h>

#define pinoChipSelectSD 53

TMRpcm audioCartaoSD;

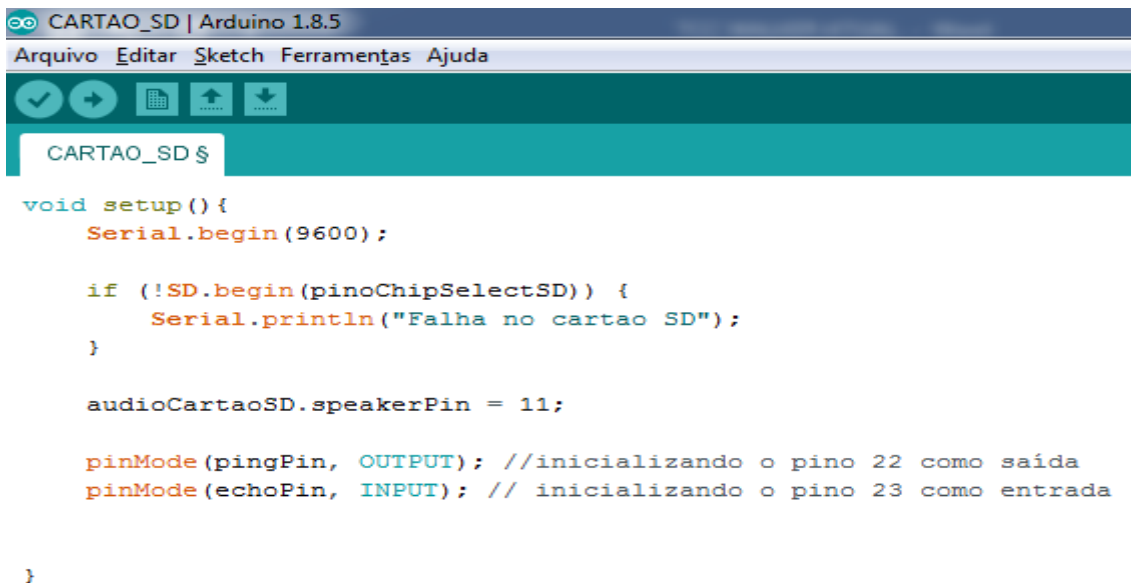
const int echoPin = 23; // Pino Echo do sensor ultrassônico
const int pingPin = 22; // Pino Trigger do sensor ultrassônico

```

FONTE: Autor, 2018.

Para iniciar a porta serial, foi definida a função `Serial.begin(9600)` e logo após foi feita a condição de que se o pino 53 selecionado para ler o Cartão SD não for lido, irá apresentar a seguinte mensagem: “Falha no cartão SD”.

Figura 18- Elaboração do Código



```

CARTAO_SD | Arduino 1.8.5
Arquivo  Editar  Sketch  Ferramentas  Ajuda

CARTAO_SD $

void setup() {
  Serial.begin(9600);

  if (!SD.begin(pinoChipSelectSD)) {
    Serial.println("Falha no cartao SD");
  }

  audioCartaoSD.speakerPin = 11;

  pinMode(pingPin, OUTPUT); //inicializando o pino 22 como saída
  pinMode(echoPin, INPUT); // inicializando o pino 23 como entrada

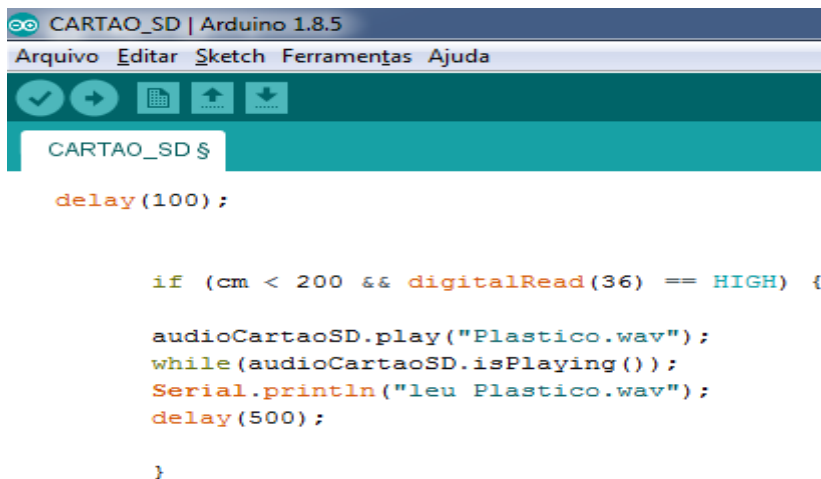
}

```

FONTE: Autor, 2018.

O pino definido no Arduino para a saída de áudio na caixinha de som, foi o 11. Depois foi inicializado o pino 23 como a entrada de sinal do sensor ultrassônico e o 22 como a saída. Os arquivos de áudio do Plástico, Metal, Vidro e papel foram definidos respectivamente pelos pinos 36, 37, 38 e 39 do Arduino. A condição imposta para que um áudio seja reproduzido é que alguém entre no raio de atuação do sensor ultrassônico numa distância pré-definida de 200cm (2 metros). Esse alguém, seria o deficiente visual em questão e enquanto ele estiver no raio de atuação o mesmo áudio ficará sendo repetido em um intervalo de 1segundo até que se saia do raio de atuação do sensor.

Figura 19- Elaboração do Código



```

CARTAO_SD | Arduino 1.8.5
Arquivo  Editar  Sketch  Ferramentas  Ajuda

CARTAO_SD $

delay(100);

if (cm < 200 && digitalRead(36) == HIGH) {

  audioCartaoSD.play("Plastico.wav");
  while(audioCartaoSD.isPlaying());
  Serial.println("leu Plastico.wav");
  delay(500);

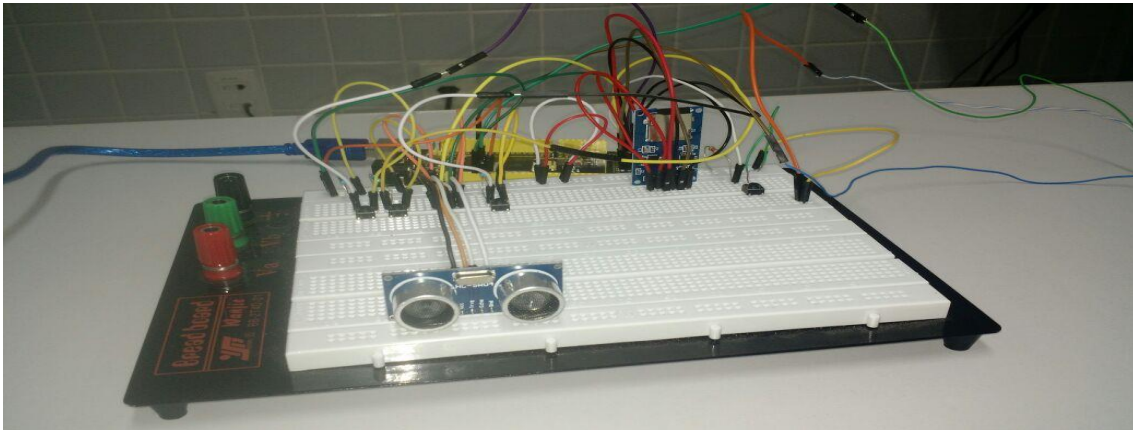
}

```

FONTE: Autor, 2018

Quando a condição for ativada e o áudio for reproduzido, o monitor serial irá escrever “leu Plástico.wav”. Isso se aplica pra cada áudio e pino selecionado. A imagem a seguir demonstra o protótipo proposto.

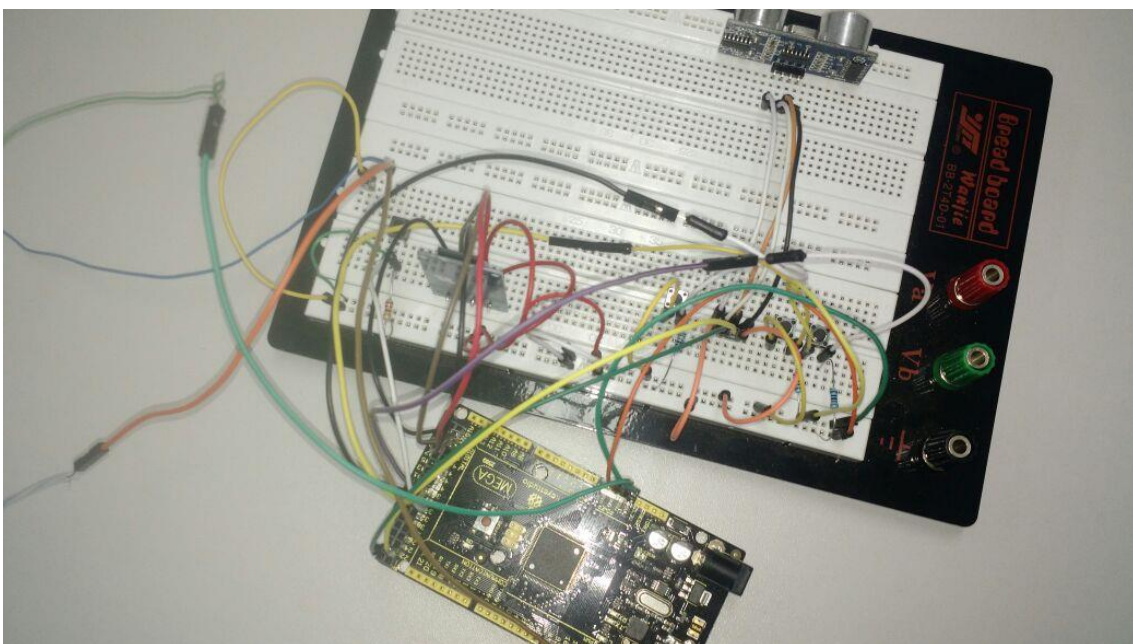
Figura 20- Protótipo proposto



FONTE: Autor, 2018

Protótipo visto de frente, com o sensor ultrassônico na frente, para detectar os sinais que serão emitidos pelo sensor que será de tamanha importância na aplicação e reprodução do sistema.

Figura 21- Protótipo proposto.



FONTE: Autor, 2018.

As imagens acima mostram a montagem final do protótipo proposto. A segunda foto mostra o protótipo visto de cima, onde se tem todas as ligações feitas no projeto para o Arduino e a protoboard. Mostrando de fato como ficou sua montagem final.

5- RESULTADOS E DISCURSSÕES

Inicialmente seria utilizado do sensor de carga e o módulo amplificador HX711 para simular a carga obtida pelo deficiente e posteriormente definir o tipo de recipiente que seria coletado. Porém, na medida que o protótipo ia sendo desenvolvido, foi decidido a substituição da célula de carga e seu módulo amplificador HX711 por 4 botões que posteriormente foi utilizado para simular as cargas em cada recipiente de coleta seletiva.

Não foi encontrado nenhuma dificuldade em trabalhar com o sensor ultrassônico HC-SR04, conseguindo de forma rápida a obtenção da distância simulada pré-definida em 2 metros a partir do recipiente de coleta para a pessoa que entrasse em seu raio de atuação. Algumas dificuldades foram encontradas na parte da obtenção do áudio sem que houvessem ruídos, porém foi constatado de forma limpa, como reproduzir o áudio com um bom entendimento.

Um problema encontrado no trabalho, foi não ter conseguido entrevistar diretamente pessoas com problemas visuais, para uma melhor compreensão e melhor obtenção de resultados. Então tudo foi pensado e construído de forma para que auxiliasse da melhor forma possível um deficiente visual. Desde a distância pré-definida pelo sensor ultrassônico para identificação do deficiente, como um baixo intervalo de reprodução de áudio, visando possível má compreensão do indivíduo para seleção da coleta seletiva que será utilizada por ele posteriormente.

6- CONSIDERAÇÕES FINAIS

Este trabalho se propôs a auxiliar de forma audível quem detém de problemas visuais facilitando seu acesso aos recipientes de coleta seletiva. Com todo o avanço tecnológico existente atualmente, a ideia desse protótipo veio como meio de inclusão social e digital para que fosse utilizado do vasto avanço tecnológico como meio de facilitar um pouco mais a vida dos deficientes visuais.

Diante de todos os dados apresentados nesse trabalho, conclui-se, que o protótipo atende a ideia inicial, apesar de necessitar de algumas melhorias, ele apresenta uma boa praticidade, custo benefício, além de ser de fácil aplicação.

Em trabalhos futuros, é possível concluir as devidas melhorias no protótipo, para que de fato as células de carga funcionem, melhorando a praticidade para inclusão social do deficiente visual. Além de que, possa fazer aplicações pessoalmente com quem precise, para melhorar a ideia inicial e deixar mais prático e fácil sua utilização.

REFERÊNCIAS

ARGUS CONTROL. **Sensores de presença de estranhos: tipos e como funcionam.** Disponível em: <<http://arguscontrol.com.br/blog/sensores-de-presenca/>>. Acesso em: 01 Fev. 2018.

ARDUINOVAÇÃO. **Faça Seu Arduino Falar.** Disponível em: <<https://arduinovacao.blogspot.com.br/2016/08/faca-seu-arduino-falar.html>>. Acesso em: 05 Mar. 2018.

BRASIL ESCOLA. **O que são resistores?** Disponível em: <<http://brasilecola.uol.com.br/o-que-e/fisica/o-que-sao-resistores.htm>>. Acesso em: 22 Jan. 2018.

BUILDBOT. **Como utilizar o sensor ultrassônico HC-SR04.** Disponível em: <<http://buildbot.com.br/blog/como-utilizar-o-sensor-ultrasonico-hc-sr04/>>. Acesso em: 25 Jan. 2018.

CONCEITO DE. **Conceito de Sensor.** Disponível em: <<https://conceito.de/sensor>>. Acesso em: 28 Jan. 2018.

ELETRÔNICA PROGRESSIVA. **Indutores - O que são, como funcionam e para que são usados.** Disponível em: <<http://www.eletronicaprogressiva.net/2013/07/Indutores-o-que-sao-para-que-servem-e-como-funcionam.html>>. Acesso em: 29 Jan. 2018.

EMBARCADOS. **Arduino MEGA 2560.** Disponível em: <<https://www.embarcados.com.br/arduino-mega-2560/>>. Acesso em: 01 Fev. 2018.

ELETRÔNICA PARA ARTISTAS. **Arduino 3: Introdução à programação.** Disponível em: <<http://eletronicaparaartistas.com.br/arduino-3-introducao-a-programacao/>>. Acesso em: 02 Fev. 2018.

EMBARCADOS. **Introdução ao Arduino - Primeiros passos na plataforma.** Disponível em: <<https://www.embarcados.com.br/arduino-primeiros-passos/>>. Acesso em: 25 Jan. 2018.

ELETROGATE BLOG. **Balança digital com Arduino e célula Strain Gauge.** Disponível em: <<http://blog.eletrogate.com/balanca-digital-com-arduino-aprenda-a-usar-a-celula-de-carga/>>. Acesso em: 25 Jan. 2018.

ELETROGATE BLOG. **Gravação de cartão micro SD com arduino - Faça seu Datalogger!** Disponível em: <<http://blog.eletrogate.com/gravacao-de-cartao-micro-sd-com-arduino/>>. Acesso em: 05 Mar. 2018.

FAZEDORES. **Conheça os shields e incremente seu Arduino com eles.** Disponível em: <<http://blog.fazedores.com/conheca-os-shields-e-incremente-seu-arduino-com-eles/>>. Acesso em: 26 Jan. 2018.

FILIPEFLOP. **Como conectar o Sensor Ultrassônico HC-SR04 ao Arduino.** Disponível em: <<https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>>. Acesso em: 25 Jan. 2018.

GRAÇA, Cláudio. **Eletromagnetismo**. Série Didática, Física 3. Santa Maria – RS: Universidade Federal de Santa Maria, 2012. Disponível em: <<http://coral.ufsm.br/cograca/eletro12.pdf>> Acesso em: 22 Jan. 2018.

GRUPO DE ENGENHARIA DE COMPUTADORES. **Microcontroladores PIC**. Disponível em: <<http://gec.di.uminho.pt/lecom/li2/material/PicBook-PT.pdf>>. Acesso em: 25 Jan. 2018.

HOEPERS, Rodrigo. **Veículo autônomo usando arduído**. Trabalho de Conclusão de Curso (Curso de Ciência da Computação), Centro de ciências Tecnológicas da Terra e do Mar, Universidade do Vale do Itajaí. Itajaí, 2012.

MUNDO DA ELÉTRICA. **O que é um resistor?** Disponível em: <<https://www.mundodaeletrica.com.br/o-que-e-um-resistor/>>. Acesso em: 22 Jan. 2018.

MUNDO EDUCAÇÃO. **Resistores**. Disponível em: <<http://mundoeducacao.bol.uol.com.br/fisica/resistores.htm/>>. Acesso em: 22 Jan. 2018.

MECÂNICA INDUSTRIAL. **O que é um indutor de ferrite**. Disponível em: <<http://www.mecanicaindustrial.com.br/274-o-que-e-um-indutor-de-ferrite/>>. Acesso em: 29 Jan. 2018.

MULTILÓGICA SHOP. **Arduino Mega2560 R3**. Disponível em: <<https://multilogica-shop.com/arduino-mega2560-r3>>. Acesso em: 25 Jan. 2018.

MACHADO, Robson. **Energizando o Arduino**. Disponível em: <<http://www.eaduino.com.br/2011/08/energizando-o-arduino/#sthash.L02fEC7F.dpbs/>>. Acesso em: 26 Jan. 2018.

PATSKO, Luís Fernando. **Tutorial Aplicações, Funcionamento e Utilização de Sensores**. Disponível em: <http://www.maxwellbohr.com.br/downloads/robotica/mec1000_kdr5000/tutorial_eletronica_-_aplicacoes_e_funcionamento_de_sensores.pdf>. Acesso em: 28 Jan. 2018.

SISTEMA DE DESPACHO DE BAGAGEM. **Arduíno com célula de carga**. Disponível em: <<http://carlosrafaelgn.com.br/blog/sdb/2016/09/15/arduino-com-celula-de-carga/>>. Acesso em: 25 Jan. 2018.

VIDA DE SILÍCIO. **HC-SR04 – Sensor Ultrassônico de distância com Arduino**. Disponível em: <<https://portal.vidadesilicio.com.br/hc-sr04-sensor-ultrassonico/>>. Acesso em: 25 Jan. 2018.

APENDICE A- ALGORITMO UTILIZANDO UM SENSOR ULTRASSOM E LEITOR DE CARTÃO MICROSD

```

#include <SD.h>// Biblioteca do leitor cartão SD incluída.
#include <TMRpcm.h>// Biblioteca do reprodutor de audio incluída.

#define pinoChipSelectSD 53// pino leitor do cartão SD

TMRpcm audioCartaoSD;

const int echoPin = 23; // Pino Echo do sensor ultrassônico
const int pingPin = 22; // Pino Trigger do sensor ultrassônico

void setup(){ //executado apenas uma vez
    Serial.begin(9600);// Habilita Comunicação Serial a uma taxa de 9600 bauds.

    if (!SD.begin(pinoChipSelectSD)) { // se o pino 53 na qual é responsável pela leitura
do cartão não for lido, a seguinte mensagem será lida: "Falha no cartão SD".
        Serial.println("Falha no cartao SD");
    }

    audioCartaoSD.speakerPin = 11;// pino 11 do arduino escolhido para saída de
audio na caixinha de som.

    pinMode(pingPin, OUTPUT); // inicializando pino 22 como saída
    pinMode(echoPin, INPUT); // inicializando pino 23 como entrada

}

void loop(){ //laço principal executado indefinidamente

    long duration, cm;
    digitalWrite(pingPin, LOW);//deixa pino em low

```

delayMicroseconds(2);

digitalWrite(pingPin, HIGH);//dispara pulso ultrassônico

delayMicroseconds(10);

digitalWrite(pingPin, LOW);// deixa pino em low

duration = pulseIn(echoPin, HIGH); // usando a função **pulsIn** para determinar o tempo total de ida e volta

cm = microsecondsToCentimeters(duration); // método de chamada

delay(100);// intervalo de repetição de 1 segundo para repetição do audio.

if (cm < 200 && digitalRead(36) == HIGH) { // Se a distância detectada pelo sensor for de 200cm ou menos o pino 36 é lido.

audioCartaoSD.play("Plastico.wav"); //o CartãoSD reproduz "Plastico.wav".

while(audioCartaoSD.isPlaying());// Enquanto o CartãoSD está reproduzindo.

Serial.println("leu Plastico.wav");// O monitor serial apresenta a mensagem "leu Plastico.wav".

delay(500);// intervalo de 5 segundos.

}

if (cm < 200 && digitalRead(37) == HIGH){ // Se a distância detectada pelo sensor for de 200cm ou menos o pino 37 é lido.

audioCartaoSD.play("Metal.wav");// o CartãoSD reproduz "Metal.wav".

while(audioCartaoSD.isPlaying());// Enquanto **audioCartãoSD** está reproduzindo.

Serial.println("leu Metal.wav");// O monitor serial apresenta a mensagem "leu Metal.wav".

delay(500);// intervalo de 5 segundos.

```
}

```

if (cm <200 && digitalRead(38) == HIGH) { // Se a distância detectada pelo sensor for de 200cm ou menos o pino 38 é lido.

```
    audioCartaoSD.play("Vidro.wav");// o CartãoSD reproduz "Vidro.wav".
```

while(audioCartaoSD.isPlaying()); // Enquanto audioCartãoSD está reproduzindo.

Serial.println("leu Vidro.wav");//O monitor serial apresenta a mensagem "leu Vidro.wav".

```
    delay(500); // Intervalo de 5 segundos
```

```
    .
```

```
}

```

```
}

```

if (cm <200 && digitalRead(39) == HIGH) { // Se a distância detectada pelo sensor for de 200cm ou menos o pino 39 é lido.

```
    audioCartaoSD.play("Papel.wav");// o CartãoSD reproduz "Papel.wav".
```

while(audioCartaoSD.isPlaying()); // Enquanto audioCartãoSD está reproduzindo.

Serial.println("Leu Papel.wav");// O monitor serial apresenta a mensagem "Leu Papel.wav".

```
    delay(500); // Intervalo de 5 segundos.
```

```
}

```

```
    delay(50);
```

```
}

```

long microsecondsToCentimeters(long microseconds) // método para converter microsegundos em centímetros.

```
{

```

```
    return microseconds / 29 / 2;  
}
```