

MARINHA DO BRASIL
DIRETORIA DE ENSINO DA MARINHA
CENTRO DE INSTRUÇÃO ALMIRANTE WANDENKOLK

**CURSO DE APERFEIÇOAMENTO AVANÇADO EM
SEGURANÇA DA INFORMAÇÃO E COMUNICAÇÕES**

TRABALHO DE CONCLUSÃO DE CURSO

**IMPLEMENTAÇÃO DE UM DETECTOR DE SONS IMPULSIVOS UTILIZANDO O
CONCEITO DE INTERNET DAS COISAS (IOT)**



1ºTen JOSÉ VICTOR DE PINA CORRIÇA

Rio de Janeiro
2021

1ºTen JOSÉ VICTOR DE PINA CORRIÇA

IMPLEMENTAÇÃO DE UM DETECTOR DE SONS IMPULSIVOS UTILIZANDO O
CONCEITO DE INTERNET DAS COISAS (IOT)

Monografia apresentada ao Centro de Instrução Almirante Wandenkolk como requisito parcial à conclusão do Curso de Aperfeiçoamento Avançado de Segurança da Informação e Comunicações.

Orientadores:

Professor PhD. Anderson Oliveira da Silva

Professor Dr. Alan Oliveira de Sá

CIAW
Rio de Janeiro
2021

FOLHA DE APROVAÇÃO

1ºTen JOSÉ VICTOR DE PINA CORRIÇA

IMPLEMENTAÇÃO DE UM DETECTOR DE SONS IMPULSIVOS UTILIZANDO O
CONCEITO DE INTERNET DAS COISAS (IOT)

Monografia apresentada ao Centro de Instrução Almirante Wandenkolk como requisito parcial à conclusão do Curso de Aperfeiçoamento Avançado em Segurança da Informação e Comunicações.

Aprovada em _____

Banca Examinadora:

CMG Gian Karlo Huback Macedo de Almeida – CIAW _____

Professor PhD. Anderson Oliveira da Silva – PUC-Rio _____

Professor Dr. Alan Oliveira de Sá – CIAW _____



Dedico este trabalho a todos os profissionais da Marinha do Brasil, que, por meio de sua dedicação e comprometimento com a pátria, fornecem todo o esforço necessário para garantir os interesses nacionais.

“Cada pessoa deve trabalhar para o seu aperfeiçoamento e, ao mesmo tempo, participar da responsabilidade coletiva por toda a humanidade.”

Marie Curie

IMPLEMENTAÇÃO DE UM DETECTOR DE SONS IMPULSIVOS UTILIZANDO O CONCEITO DE INTERNET DAS COISAS (IOT)

Resumo

A Internet das Coisas surge como uma tecnologia que permite aos dispositivos do cotidiano se conectarem à Internet, possibilitando, assim, que estes se comuniquem entre si. Diante deste conceito, este trabalho apresenta um detector de sons impulsivos capaz de identificar um som impulsivo e transmitir esta informação a um dispositivo através da Internet. A motivação para o desenvolvimento deste detector de sons impulsivos se deu por conta da proposta inicial de se desenvolver um detector de disparo de arma de fogo, o qual poderia servir como um suporte às operações militares de garantia da lei e da ordem, por meio do emprego de detectores de disparo de arma de fogo fixos e conectados à Internet, distribuídos em pontos estratégicos do Estado do Rio de Janeiro, com a finalidade de identificar a localização e o instante exato de um disparo de arma de fogo. Dessa forma, possibilitaria à Polícia Militar e aos militares componentes do Corpo de Fuzileiros Navais empregados nestas operações, obterem, automaticamente, a informação da posição precisa e do momento exato de um determinado disparo, a fim de que as ações iniciais sejam tomadas de forma mais rápida e precisa. Portanto, este trabalho concluiu a proposta em questão, mostrando ser possível a implementação de um detector de sons impulsivos utilizando o conceito de Internet das Coisas, sendo este o passo inicial e inerente ao caminho a ser percorrido até a conclusão de um projeto que possa discernir um som impulsivo de um disparo de arma de fogo.

PALAVRAS-CHAVE: Detector de sons impulsivos. Internet das Coisas. Corpo de Fuzileiros Navais.

LISTA DE FIGURAS

Figura 1: Modelo do sistema de detecção automático de um disparo de arma de fogo.	11
Figura 2: Estrutura básica de um microcontrolador.....	13
Figura 3: Janela inicial do IDE.	15
Figura 4: Principais modelos de placas Arduino.	16
Figura 5: Especificações técnicas dos principais modelos de placa Arduino.	18
Figura 6: Principais modelos de chip da família ESP8266.	19
Figura 7: Especificações técnicas do KY-038 e KY-037.	21
Figura 8: Pinagem do Módulo e Componentes Eletrônicos.	21
Figura 9: Esquema básico dos microfones.	22
Figura 10: Valores analógicos como uma série de tempo no plotter serial Arduino.	22
Figura 11: Exemplos de sons impulsivos.	23
Figura 12: Esquema ADSR.	24
Figura 13: Fluxo de notificações de push.	27
Figura 14: Placa Arduino UNO.	30
Figura 15: Módulo sensor de som KY-038.	31
Figura 16: Diagrama de conexão da placa Arduino UNO com o módulo KY-038.	31
Figura 17: Identificação das abas referentes ao código de programação.	32
Figura 18: Ilustração do serial plotter da IDE Arduino dos valores do RMS (azul) e Desvio Padrão (vermelho) variando no tempo.	33
Figura 19: NODE MCU ESP-12E.	34
Figura 20: Diagrama de conexão do NODE MCU ESP-12 com o módulo KY-038.	35
Figura 21: Imagem do monitor serial da IDE Arduino ao detectar um impulso.	36
Figura 22: Imagem do aplicativo PUSH SAFER ao receber um alerta de impulso.	36
Figura 23: Detector de sons impulsivos finalizado.	37
Figura 24: Ilustração de três sensores S1, S2 e S3 e uma fonte de emissão acústica P.	39

LISTA DE SIGLAS E ABREVIATURAS

ADSR	Attack, Decay, Sustain and Release
API	Application Programming Interface
BLE	Bluetooth Low Energy
CPU	Central Process Unit
GCM	Google Cloud Messaging
GPIO	General Purpose Input/Output
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IOT	Internet of Things
IP	Internet Protocol
LORA	Long Range
PWM	Pulse Width Modulation
RFID	Radio Frequency Identification
RMS	Root Mean Square
SDK	Software Development Kit
TCC	Trabalho de Conclusão de Curso
TCP	Transmission Control Protocol
U-FL	Upper Flammable Limit

SUMÁRIO

1 INTRODUÇÃO	9
1.1 Apresentação do problema	10
1.2 Justificativa e relevância do trabalho	10
1.3 Objetivos do Trabalho de Conclusão de Curso	12
2 REFERENCIAL TEÓRICO	12
2.1 Plataformas de prototipagem eletrônica	12
2.1.1 Microcontrolares	13
2.1.2 Arduino	14
2.1.3 Espressif Systems	18
2.2 Sensores sonoros	20
2.3 Características específicas dos sons impulsivos	23
2.4 Root Mean Square (RMS)	25
2.5 Desvio padrão	25
2.6 Notificação Push	26
2.7 Protocolos de comunicação para IoT	28
2.7.1 Protocolo Wi-Fi (802.11)	28
2.7.2 BLE	28
2.7.3 SigFox	28
2.7.4 LoRa	29
3 DESENVOLVIMENTO DO PROJETO	30
4 CONCLUSÃO	37
4.1 Trabalhos futuros	38
REFERÊNCIAS	41
APÊNDICE A – Aba “TCC_FINAL” da IDE Arduino referente ao código de programação do detector de sons impulsivos	44
APÊNDICE B – Aba “RMS.h” da IDE Arduino referente ao código de programação do detector de sons impulsivos	50

1 INTRODUÇÃO

Internet das Coisas (IoT – *Internet of Things*), como o nome sugere, é a conectividade de dispositivos do dia-a-dia uns com os outros. Com o avanço da tecnologia, vários dispositivos estão usando sensores, atuadores, computação embarcada e computação em nuvem. Isso habilitou a comunicação entre os dispositivos. Para simplificar, a Internet das Coisas permite que os dispositivos interajam e se coordenem uns com os outros, reduzindo assim a intervenção humana nas tarefas básicas do dia-a-dia. Para compreender melhor a IoT, pode-se considerar o cenário de uma casa inteligente. Assim que o alarme soa, é enviado um sinal para a cafeteira e a torradeira, que automaticamente começam a fazer seu trabalho sem qualquer intervenção humana. Assim, economizando tempo e facilitando nossas tarefas diárias, esse tipo de dispositivo de comunicação é a Internet das Coisas (AL-FUQAHA *et al.*, 2015).

A IoT permite que objetos físicos vejam, ouçam, pensem e realizem trabalhos fazendo com que eles operem conjuntamente, de tal forma que seja possível compartilhar informações e coordenar decisões. Uma rede de dispositivos/aplicativos heterogêneos tem seu próprio conjunto de desafios. Além disso, como a comunicação entre esses dispositivos, bem como com os serviços relacionados, deve acontecer a qualquer hora e em qualquer lugar, é frequentemente feita de forma sem fio, autônoma e ad-hoc (AL-FUQAHA *et al.*, 2015).

Além disso, os serviços tornam-se muito mais fluidos, descentralizados e complexos. Conseqüentemente, as barreiras de segurança na Internet das Coisas tornam-se muito mais tênues. A maioria dos dispositivos habilitados para IoT não é muito segura e pode ser acessada por terceiros facilmente. Assim, existe uma grande necessidade de padronizá-lo para garantir que a privacidade do usuário não seja invadida (AL-FUQAHA *et al.*, 2015).

A Internet das Coisas (IoT) representa o próximo passo na direção da digitalização de nossa sociedade e economia, apresentando oportunidades fascinantes em uma incrível variedade de aplicativos, onde objetos e pessoas estão interconectados por meio de redes de comunicação (AL-DOGHMAN; CHACZKO; JIANG, 2017).

Neste Trabalho de Conclusão de Curso (TCC), a IoT se apresenta como uma tecnologia que possibilita a comunicação entre diferentes dispositivos. Dessa forma, ela se torna uma importante ferramenta com a capacidade de sanar diversos problemas atinentes ao cotidiano do ser humano como um todo, seja para acionar um determinado dispositivo eletrônico ou até mesmo receber algum alerta. Com isso, torna-se clara a importância da IoT no mundo atual, no qual cada vez mais se utilizam os dispositivos pessoais para comunicação a fim de tornar a vida mais prática, ágil e simples.

1.1 Apresentação do problema

O trabalho a ser abordado neste TCC é a utilização da Internet das Coisas para detecção de sons impulsivos, através do emprego de um sensor conectado à Internet. Neste cenário, ao passo em que a Internet se tornou a grande via da comunicação moderna, ela veio se tornando cada vez mais presente no cotidiano da população. O uso de dispositivos móveis utilizando a Internet para comunicação se consolidou como um meio prático, simples e popular. Dessa forma, este trabalho visa apresentar um projeto de detecção de sons impulsivos que utiliza o conceito de Internet das Coisas para fazer a notificação dos alertas de detecção gerados pelo sensor.

1.2 Justificativa e relevância do trabalho

Segundo o decreto presidencial nº 3.897, de 24 de agosto de 2001, em seu Art. 2º § 2º fica claro que o Presidente da República, à vista de solicitação de Governador de Estado ou do Distrito Federal, poderá, por iniciativa própria, determinar o emprego das Forças Armadas para a garantia da lei e da ordem.

Desde então, alguns Estados da Federação têm solicitado o apoio das Forças Armadas em diversas ocasiões afetas à Segurança Pública. Diante disso, pode-se citar a operação realizada pelo Corpo de Fuzileiros Navais da Marinha do Brasil em 2017 na comunidade da Rocinha, no Estado do Rio de Janeiro.

O balanço, na avaliação do então Chefe do Estado-Maior Conjunto das Operações em apoio aos Órgãos de Segurança Pública do Rio de Janeiro, Contra-Almirante, Fuzileiro Naval (FN), Roberto Rossatto, foi extremamente positivo. De acordo com o almirante, as ações de cerco das Forças Armadas liberaram os policiais para realizarem tarefas específicas no interior da comunidade, o que contribuiu para a normalização das atividades na Rocinha (MARINHA DO BRASIL, 2017).

O papel das Forças Armadas nessas operações de garantia da lei e da ordem é apoiar as ações de Segurança Pública. Face ao exposto, o uso da IoT pode servir como auxílio nestas operações militares com o emprego de detectores de disparos de arma de fogo conectados à Internet instalados em áreas estratégicas do Estado do Rio de Janeiro a fim de identificar o local e a hora do respectivo disparo de forma que, tanto o Corpo de Fuzileiros Navais quanto a polícia militar, durante as operações de permanência nestas zonas de risco, fiquem cientes sobre onde e quando foi realizado algum disparo na região, possibilitando assim que as ações

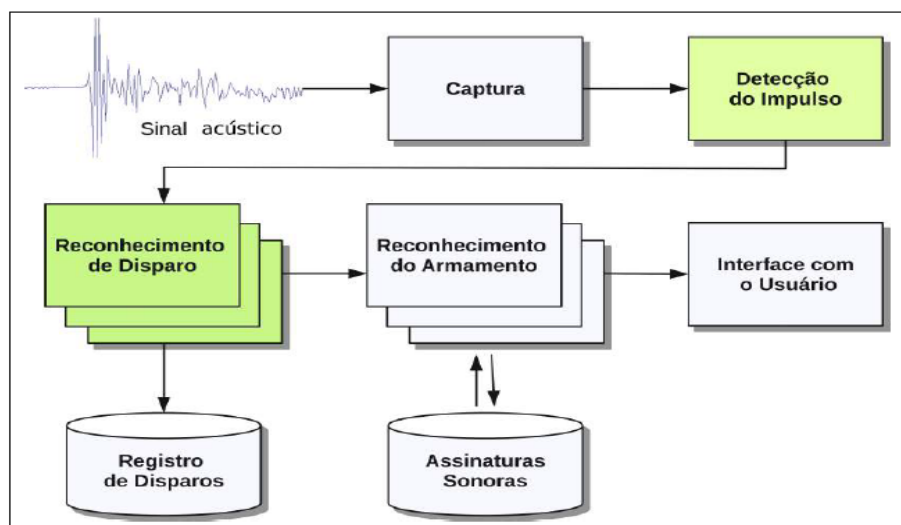
sejam realizadas de forma mais precisa e rápida a fim de interceptar algum provável conflito armado.

Em seu trabalho, Clóvis Ferreira dos Reis (2015) apresentou uma ferramenta computacional que possibilita realizar a detecção e reconhecimento automático de disparos de armas de fogo. Para tanto, inicialmente foi apresentado o módulo de detecção de som impulsivo através de algoritmos, o qual realiza a identificação da ocorrência de sons impulsivos, diferenciando-os dos demais tipos de sons. Após isso, foi apresentado o módulo de reconhecimento de disparo (módulo de classificação) utilizando algoritmos que permitam diferenciar um som impulsivo qualquer de um disparo de arma de fogo, utilizando técnicas no domínio do tempo ou transformadas do sinal para o domínio das frequências.

Todo disparo de arma de fogo é um som impulsivo. Porém, este apresenta características exclusivas que o diferenciam dos demais sons impulsivos, pois é composto por uma somatória de sinais sonoros. Face ao exposto, torna-se claro que foi preciso inicialmente identificar uma forma de detectar um som impulsivo para que, a partir daí se pudesse desenvolver um algoritmo para detecção de um disparo de arma de fogo, conforme pode ser observado na figura 1.

Este trabalho se justifica por apresentar um dispositivo de detecção de som impulsivo que, pode ser utilizado como protótipo inicial para a implementação de um detector automático de disparo de arma de fogo, o qual poderia ser futuramente implementado para auxiliar o Corpo de Fuzileiros Navais durante suas operações de garantia da lei e da ordem no apoio à Segurança Pública.

Figura 1: Modelo do sistema de detecção automático de um disparo de arma de fogo.



Fonte: (REIS, 2015)

1.3 Objetivos do Trabalho de Conclusão de Curso

Este Trabalho de Conclusão de Curso tem como objetivo principal o uso da IoT como ferramenta para detecção de sons impulsivos. Este sensor, sendo aprimorado e desenvolvido ao ponto de ter a capacidade de identificar um disparo de arma de fogo, poderia ser empregado no apoio à Segurança Pública.

Tanto os militares que compõem o Corpo de Fuzileiros Navais que estejam sendo empregados durante as operações de garantia da lei e da ordem quanto os policiais militares receberiam, automaticamente, informações como localização e hora de um determinado disparo identificado em algum ponto da cidade do Rio de Janeiro onde pudesse estar ocorrendo algum conflito armado. Possibilitando assim, que as ações policiais pudessem ser implementadas de forma rápida e precisa.

2 REFERENCIAL TEÓRICO

Neste capítulo serão apresentadas algumas plataformas de prototipagem eletrônica e sensores sonoros, abordando suas principais características e funcionalidades específicas. Outros pontos importantes a serem tratados são as características específicas dos sons impulsivos, assim como os cálculos do root mean square (RMS) e do desvio padrão. Além disso, será exposto um tipo específico de comunicação baseado na Internet conhecido como notificação push. Por último, serão expostos alguns protocolos de comunicação para IoT.

2.1 Plataformas de prototipagem eletrônica

Uma plataforma de prototipagem eletrônica é uma placa composta por um microcontrolador, algumas linhas de entrada/saída digital e analógica utilizando uma interface serial ou USB, podendo ser programado através de uma linguagem de programação padrão. Tem como objetivo criar diversas ferramentas de fácil utilização e baixo custo, sendo utilizado para desenvolver objetos interativos independentes ou pode também ser conectado a um computador hospedeiro.

Dentre as plataformas de prototipagem eletrônica utilizadas atualmente, destacam-se aquelas fabricadas pela Arduino e Espressif. Diante disso, serão apresentadas as características dos principais modelos de placas de ambas as empresas.

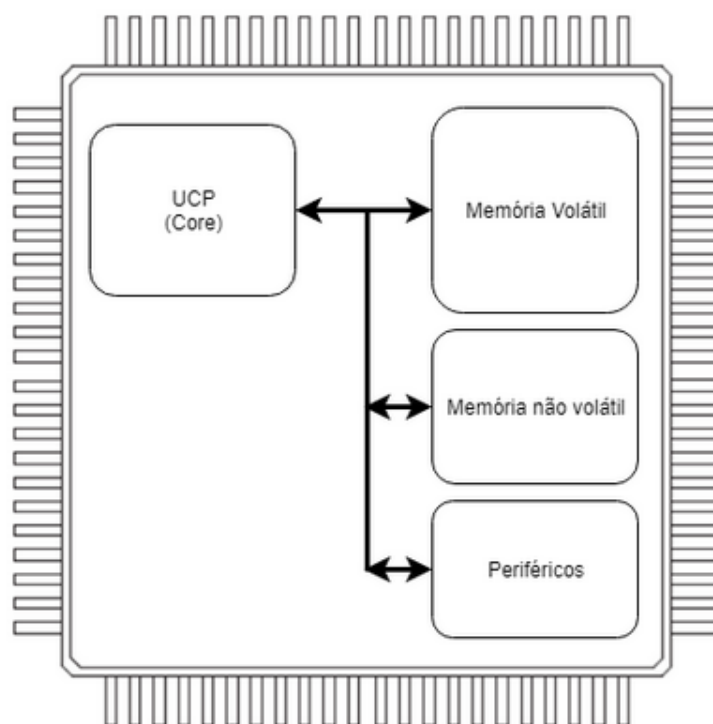
2.1.1 Microcontrolares

Um microcontrolador é um circuito integrado compacto projetado para governar uma operação específica em um sistema embarcado. Um microcontrolador típico inclui um processador, memória e periféricos de entrada / saída em um único chip (LUTKEVICH, 2019).

Microcontroladores são encontrados em diversos dispositivos e são essencialmente simples computadores pessoais em miniatura projetados para controlar pequenos recursos de um componente maior, sem um sistema operacional front-end complexo (LUTKEVICH, 2019).

Um microcontrolador está embutido dentro de um sistema para controlar uma função única em um dispositivo. Ele faz isso interpretando os dados que recebe de seus periféricos de entrada e saída usando seu processador central. A informação temporária que o microcontrolador recebe é armazenada em sua memória de dados, onde o processador a acessa e usa instruções armazenadas em sua memória de programa para decifrar e aplicar os dados recebidos. Em seguida, ele usa seus periféricos de entrada e saída para se comunicar e executar a ação apropriada (LUTKEVICH, 2019).

Figura 2: Estrutura básica de um microcontrolador.



Fonte: (EMBEDDED SYSTEMS, 2018).

Os principais elementos de um microcontrolador são a unidade central de processamento (CPU – *Central Process Unit*), a memória e os periféricos.

A CPU processa e responde a várias instruções que direcionam a função do microcontrolador. Esta ação envolve a execução de operações aritméticas, lógicas e de entrada e saída básicas. Ele também executa operações de transferência de dados, que comunicam comandos a outros componentes no sistema incorporado maior (LUTKEVICH, 2019).

A memória de um microcontrolador é usada para armazenar os dados que o processador recebe e usa para responder às instruções que foi programado para executar. Além disso, possui dois tipos principais de memória, sendo elas a memória volátil e não volátil (LUTKEVICH, 2019).

A memória não volátil mantém informações ao longo do tempo sem a necessidade de uma fonte de alimentação. Na memória volátil, isso ocorre de outra forma, os dados nela contidos são temporários e só são mantidos se o dispositivo estiver conectado a uma fonte de alimentação (LUTKEVICH, 2019).

Os dispositivos de entrada e saída são a interface do processador para o mundo exterior. As portas de entrada recebem informações e as enviam ao processador na forma de dados binários. O processador recebe esses dados e envia as instruções necessárias para dispositivos de saída que executam tarefas externas ao microcontrolador (LUTKEVICH, 2019).

2.1.2 Arduino

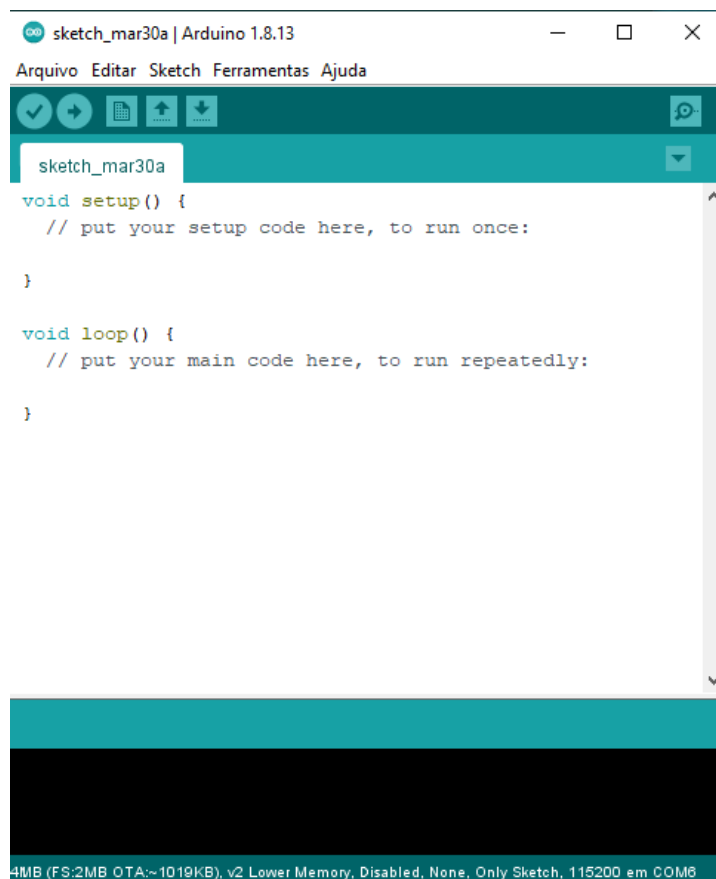
O Arduino é uma placa composta por um microcontrolador, circuitos de entrada/saída e que pode ser facilmente conectada a um computador e programada via *Integrated Development Environment* (IDE) utilizando uma linguagem baseada em C/C++, sem a necessidade de equipamentos extras além de um cabo USB (THOMSEN, 2014a).

O IDE é o software oficial apresentado pelo Arduino.cc, o qual é usado principalmente para editar, compilar e enviar o código no dispositivo Arduino. Todos os módulos do Arduino são compatíveis com este software que é de código aberto e está prontamente disponível para instalar e começar a compilar o código em movimento (FEZARI; AL DAHOUD, 2018).

Diante disso, o IDE torna possível a programação de todos os modelos de placas Arduino e sempre que o projeto apresentar algum problema em sua configuração ou até mesmo em seu código, o mesmo irá gerar uma notificação e elencar onde pode estar o problema (STRAUB, 2019).

O software Arduino conta com um layout bastante completo e de fácil navegação, todas as opções são separadas de acordo com suas funções e possibilitam os mais diversificados procedimentos de forma direta, simples e totalmente em português. Além disso, existe uma quantidade extremamente grande de bibliotecas disponíveis, algumas delas são instaladas junto ao Arduino IDE quando realizamos o download e outras devem ser incluídas de acordo com as necessidades de cada projeto (STRAUB, 2019). Na figura 3 pode-se observar a janela inicial do IDE.

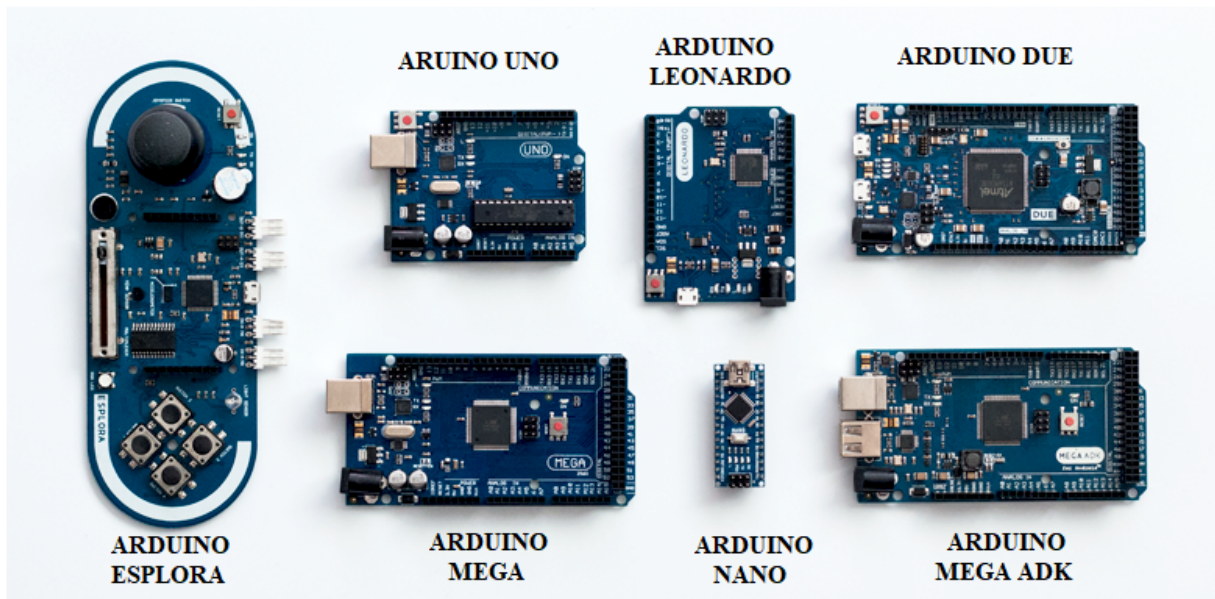
Figura 3: Janela inicial do IDE.



Fonte: Próprio autor.

As principais placas Arduino são o Arduino UNO, Arduino LEONARDO, Arduino DUE, Arduino MEGA ADK, Arduino NANO, Arduino MEGA 2560 e Arduino ESPLORA.

Figura 4: Principais modelos de placas Arduino.



Fonte: (THOMSEN, 2014a).

O Arduino UNO costuma ser a primeira opção quando se procura por uma placa Arduino, pois possui um bom número de portas disponíveis, e grande compatibilidade com os shields disponíveis no mercado. Possui processador ATMEGA328, 14 portas digitais, sendo que 6 delas podem ser usadas como saídas PWM, e 6 portas analógicas. A alimentação pode vir da conexão USB ou do conector para alimentação externa (THOMSEN, 2014b).

O Arduino MEGA 2560 é a maior versão da placa Arduino, com microcontrolador ATmega2560 e 54 portas digitais, das quais 15 podem ser usadas como PWM, além de 15 portas analógicas. Possui clock de 16 Mhz, conexão USB e conector para alimentação externa. É considerado ideal para projetos mais elaborados que exijam grande número de entradas e saídas (THOMSEN, 2014b).

O Arduino Leonardo é uma placa que à primeira vista é muito parecida com o Arduino Uno, porém com um microcontrolador Atmega32u4, possuindo 20 portas digitais, das quais 7 podem ser usadas como PWM, e 12 como portas analógicas. Além disso, esta placa também possui clock de 16 Mhz e conexão para alimentação externa. Diferentemente do Arduino Uno, possui conector micro-usb para ligação ao computador. Outra característica dessa placa é o chip de conexão USB integrado ao microcontrolador, o que elimina a necessidade de um chip adicional de comunicação na placa, e permite que o Arduino Leonardo seja reconhecido pelo computador como se fosse um mouse ou um teclado, e não necessariamente como uma porta serial (THOMSEN, 2014b).

De todos os tipos de Arduino, o DUE é a placa com maior capacidade de processamento, baseada em um microcontrolador ARM de 32 bits e 512 Kb de memória totalmente disponível para programas/aplicações. O bootloader já vem gravado de fábrica em uma memória ROM dedicada. Além disso, possui 54 portas digitais, das quais 12 podem ser usadas como PWM, e 12 portas analógicas. Possui também 4 chips controladores de portas seriais, conexão USB e conector para alimentação externa. Suas ligações exigem especial atenção pois as portas trabalham à 3.3V, o que pode comprometer o uso dos shields disponíveis no mercado, que geralmente trabalham com 5v (THOMSEN, 2014b).

Também baseado no ATmega2560, o Arduino MEGA ADK possui uma conexão USB dedicada à ligação com dispositivos baseados em Android, como telefones celulares. Possui 54 portas digitais, das quais 15 podem ser usadas como PWM, 16 portas analógicas, 4 chips dedicados à comunicação serial, clock de 16 Mhz e conexão ao computador via USB (THOMSEN, 2014b).

O Arduino NANO é uma Placa compacta baseada no microcontrolador ATmega328 (para placas Arduino Nano versão 3.x), ou ATmega168 (versão 2.x). Ao contrário das outras placas, não possui conector para alimentação externa, sendo alimentada por um conector USB Mini-B. Possui 32 Kb de memória (Nano versão 3.0), ou 16 Kb (Nano versão 2.0). Além disso, seu tamanho reduzido faz dessa placa uma boa opção para projetos compactos que exijam atualização constante de software (THOMSEN, 2014b).

O Arduino ESPLORA é uma placa diferente de todas as outras da família Arduino, principalmente por possuir diversos sensores na sua construção. Nessa placa vêm embutidos um buzzer, um joystick, um potenciômetro deslizante, um sensor de temperatura, um acelerômetro, um sensor de luz (LDR), 4 push-buttons e um microfone. Além de tudo isso, ainda possui um soquete para tela LCD. Esta é uma placa indicada para quem deseja aprender Arduino sem se preocupar muito com eletrônica, uma vez que os componentes já estão embutidos na placa. Assim, o usuário pode se concentrar na parte de programação. Baseado no Arduino Leonardo, utiliza o microcontrolador Atmega32U4, com clock de 16 Mhz e 32 KB de memória. Possui 2 conectores de 3 pinos para conexão de módulos adicionais. Por fim, sua conexão com o microcomputador utiliza cabo micro-usb (THOMSEN, 2014b).

Figura 5: Especificações técnicas dos principais modelos de placa Arduino.

	UNO	MEGA 2560	LEONARDO	DUE	ADK	NANO	ESPLORA
Microcontrolador	ATmega328	ATmega2560	ATmega32u4	AT91SAM3X8E	ATmega2560	ATmega168 (versão 2.x) ou ATmega328 (versão 3.x)	ATmega32u4
Portas digitais	14	54	20	54	54	14	-
Portas PWM	6	15	7	12	15	6	-
Portas analógicas	6	16	12	12	16	8	-
Memória	32K (0,5K usado pelo bootloader)	256K (8K usado pelo bootloader)	32K (4K usado pelo bootloader)	512K disponível para aplicações	256K (8K usado pelo bootloader)	16K (ATmega168) ou 32K (ATmega328) (bootloader: 2K)	32K (4K usado pelo bootloader)
Clock	16Mhz	16Mhz	16Mhz	84Mhz	16Mhz	16Mhz	16Mhz
Conexão	USB	USB	Micro USB	Micro USB	USB	USB Mini-B	Micro USB
Conector para alimentação externa	Sim	Sim	Sim	Sim	Sim	Não	Não
Tensão de operação	5V	5V	5V	3.3V	5V	5V	5V
Corrente máxima portas E/S	40mA	40mA	40mA	130mA	40mA	40mA	-
Alimentação	7-12Vdc	7-12Vdc	7-12Vdc	7-12Vdc	7-12Vdc	7-12Vdc	5V

Fonte: (THOMSEN, 2014b).

As ferramentas criadas utilizando placas Arduino possuem diversos tipos de aplicações diferentes. Pode-se citar como exemplos projetos de robôs automatizados, sensores de detecção de fumaça, sistemas de alarme, acesso de segurança utilizando o leitor RFID, dentre outras infinitudes de ideias das quais podem surgir tendo como base estas plataformas de prototipagem eletrônica.

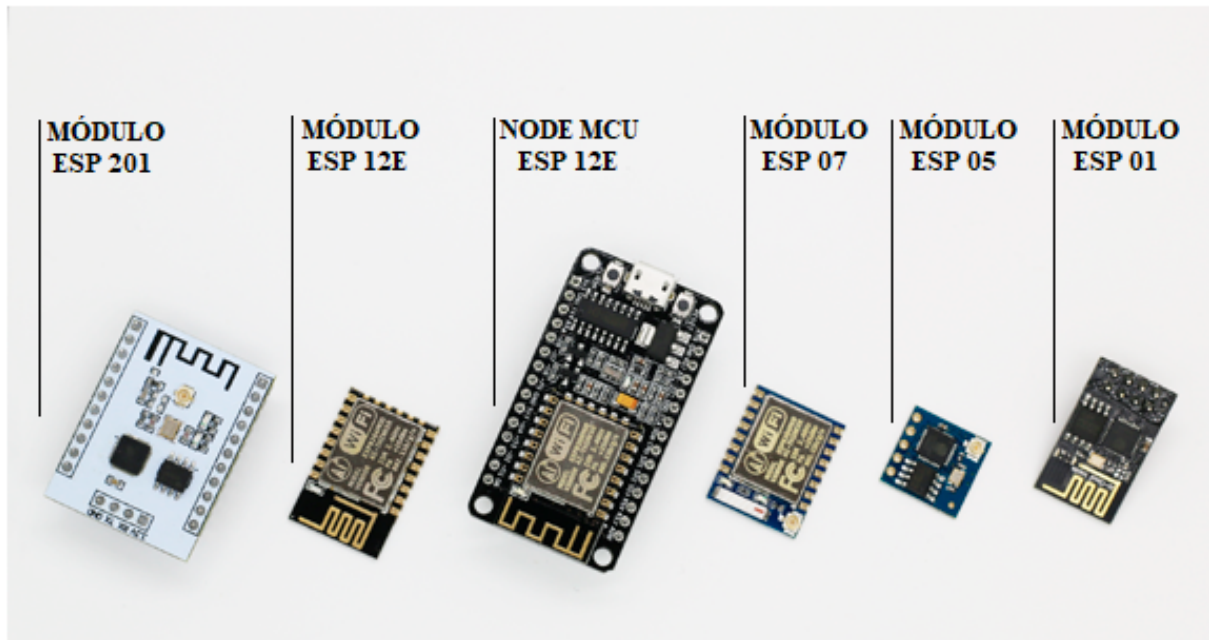
2.1.3 Espressif Systems

A Espressif Systems é uma fabricante chinesa de plataformas de prototipagem eletrônica, dentre as quais se destacam aqueles da família ESP8266.

Os módulos com chip ESP8266 são microcontroladores os quais possuem todo o necessário para se conectarem à Internet. Por este motivo, são uma ótima alternativa para o desenvolvimento de um projeto de IoT, devido tanto ao seu tamanho quanto pela praticidade.

Os principais modelos da família ESP8266 são o módulo ESP 201, módulo ESP 12E, NODE MCU ESP 12E, módulo ESP 07, módulo ESP 05 e módulo ESP 01.

Figura 6: Principais modelos de chip da família ESP8266.



Fonte: (THOMSEN, 2016)

O módulo ESP8266 ESP-201 é um módulo um pouco mais fácil de usar em termos de prototipação, pois pode ser montado em uma protoboard. Os 4 pinos laterais, que são responsáveis pela comunicação serial, atrapalham um pouco esse tipo de montagem. Contudo, pode-se soldar esses pinos no lado oposto da placa, ou utilizar algum tipo de adaptador. O ESP 201 possui 11 portas GPIO, antena embutida e conector U-FL para antena externa (THOMSEN, 2016).

O módulo ESP8266 ESP-12E é muito semelhante ao ESP-07, porém possui apenas antena interna (PCB). Tem 11 pinos GPIO e é muito utilizado como base para outros módulos ESP8266, como o NodeMCU (THOMSEN, 2016).

O Node MCU é uma placa criada para facilitar o desenvolvimento de aplicações para o módulo ESP 12. Há um módulo ESP 12 soldado nesta placa. Sua grande vantagem é que ela funciona como se fosse um Arduino, com o adendo de já possuir integração à rede Wi-Fi. Quando comparado ao Arduino, o Node MCU ESP8266 ESP 12E possui um poder de processamento muito superior.

O ESP8266 Node MCU ESP-12E é uma placa de desenvolvimento completa, que além do chip ESP8266 conta com um regulador de tensão de 3.3V. É um módulo que pode ser

encaixado diretamente na protoboard e dispensa o uso de um microcontrolador externo para operar, já que pode ser facilmente programado utilizando o software IDE Arduino. Possui 11 pinos de GPIO, conector micro-usb para programação/alimentação e botões para reset e flash do módulo (THOMSEN, 2016).

O módulo ESP8266 ESP-07 também é um módulo compacto, porém com um layout diferente, sem os pinos de ligação. O módulo conta com uma antena cerâmica embutida e um conector U-FL para antena externa. Esse módulo tem 9 GPIOs e seu layout permite que ele seja integrado facilmente à uma placa de circuito impresso, muito utilizada em projetos de automação residencial (THOMSEN, 2016).

O módulo Wi-Fi ESP8266 ESP-05 é um módulo um pouco diferente das outras placas da linha ESP8266, pois não possui portas que se pode usar para acionar dispositivos ou ler dados de sensores. Por outro lado, é uma alternativa interessante para projetos de IoT quando se precisa de uma boa conexão de rede/Internet por um baixo custo. Pode ser utilizado, por exemplo, para montar uma web server com Arduino ou efetuar uma comunicação de longa distância entre placas como Arduino/Arduino, Arduino/Raspberry, etc.

O módulo ESP8266 ESP-01 é o módulo mais comum da linha ESP8266. Ele é compacto, e possui dois pinos GPIO que podem ser controlados conforme a programação. Além disso, pode ter o firmware regravado e/ou atualizado, conforme mostramos no post Upgrade de Firmware do Módulo ESP8266, utilizando interface serial. Uma pequena desvantagem desse tipo de módulo é a disposição dos pinos, que dificultam a utilização em uma protoboard, contudo, pode-se facilmente utilizar um adaptador para módulo Wi-Fi ESP8266 ESP-01 ou jumpers macho/fêmea, por exemplo.

2.2 Sensores sonoros

Existem dois sensores de som de microfone diferentes, chamados KY-038 e KY-037. A única diferença entre eles é que o KY-037 possui um microfone embutido com uma sensibilidade mais alta. Portanto, podemos usar os dois módulos sensores de som completamente iguais, pois possuem as mesmas especificações e os mesmos componentes elétricos.

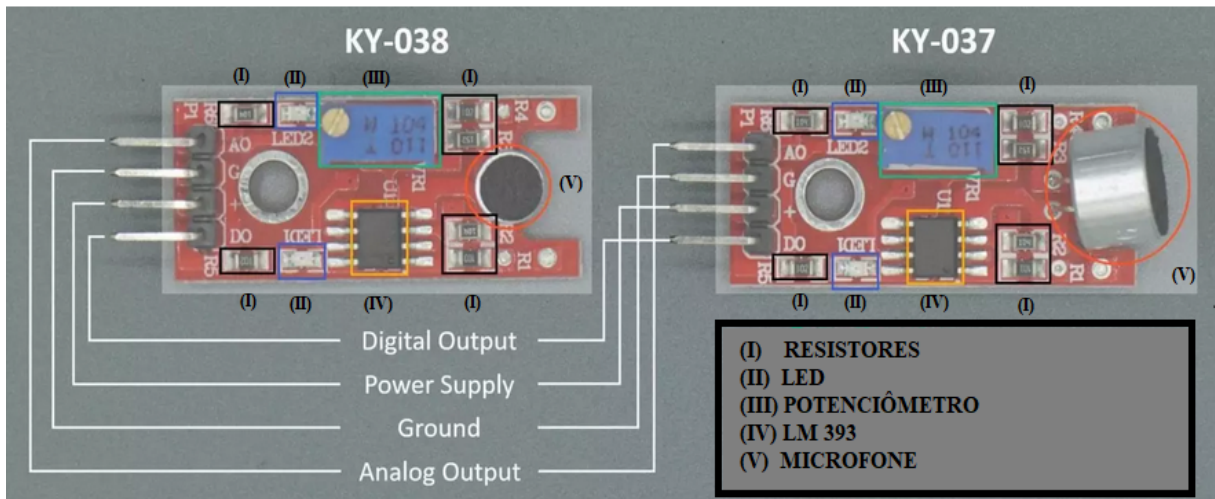
Figura 7: Especificações técnicas do KY-038 e KY-037.

	KY-038	KY-037
VOLTAGEM DE OPERAÇÃO	3.3V...5V	3.3V...5V
RESPOSTA EM FREQUÊNCIA	50 Hz...20 kHz	50 Hz...20 kHz
IMPEDÂNCIA	2.2 kΩ	2.2 kΩ

Fonte: (GOODHEW, 2021).

Estes módulos sensores possuem vários componentes que fazem parte de um circuito. A figura 8 apresenta o KY-038 e o módulo KY-037 com todas as suas partes eletrônicas.

Figura 8: Pinagem do Módulo e Componentes Eletrônicos.

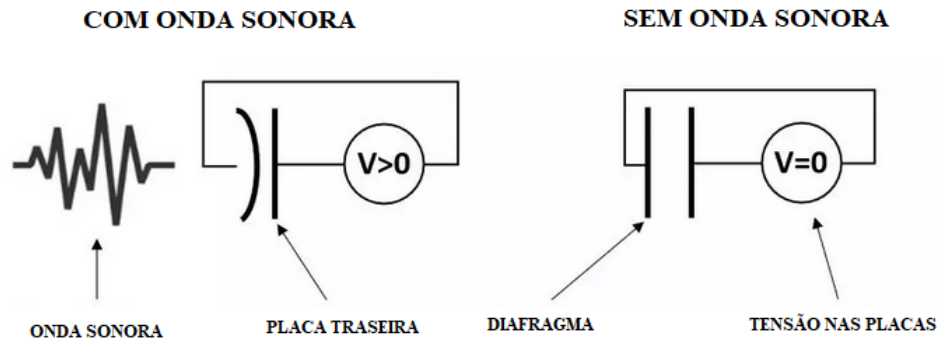


Fonte: (GOODHEW, 2021).

O comparador LM393 tem a função de comparar o sinal gerado pelo microfone com o valor pré-definido através do potenciômetro e também de controlar o status do LED que indica o status da saída digital. Os dois LEDs possuem a função de indicar que o módulo está operando e também de indicar o status do pino digital. O Microfone gera o sinal de som e o potenciômetro define um limite para o pino de saída digital.

A funcionalidade básica do microfone é a mesma para ambos os módulos. O microfone gera a entrada para o módulo sensor de som e consiste em um diafragma fino que é uma placa de um capacitor. A segunda placa do capacitor é a placa traseira e em paralelo ao diafragma em uma distância muito próxima. A figura 9 expõem um esquema básico destes microfones.

Figura 9: Esquema básico dos microfones.

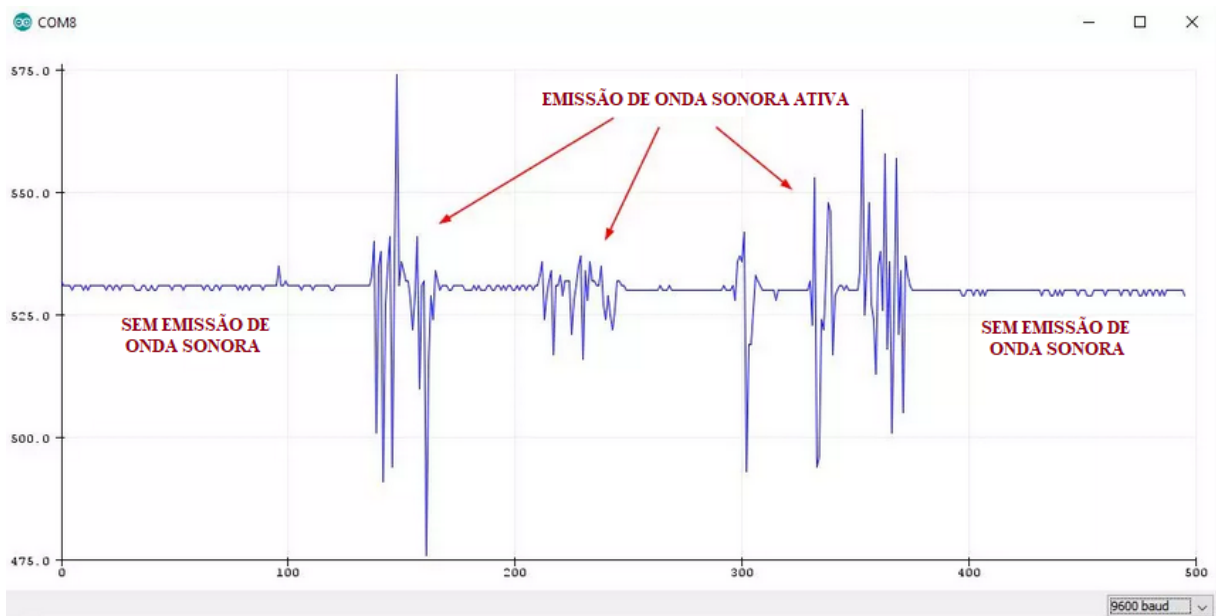


Fonte: (GOODHEW, 2021).

Se alguém fala no microfone, as ondas sonoras criadas pela voz atingem o diafragma. Devido a esses golpes, o diafragma vibra e, portanto, a distância entre as duas placas do capacitor fica menor ou maior.

Como a capacitância é diretamente proporcional à distância entre as placas, as ondas sonoras da voz alteram a tensão nas placas que tem influência direta do circuito do módulo sensor de som.

Figura 10: Valores analógicos como uma série de tempo no plotter serial Arduino.



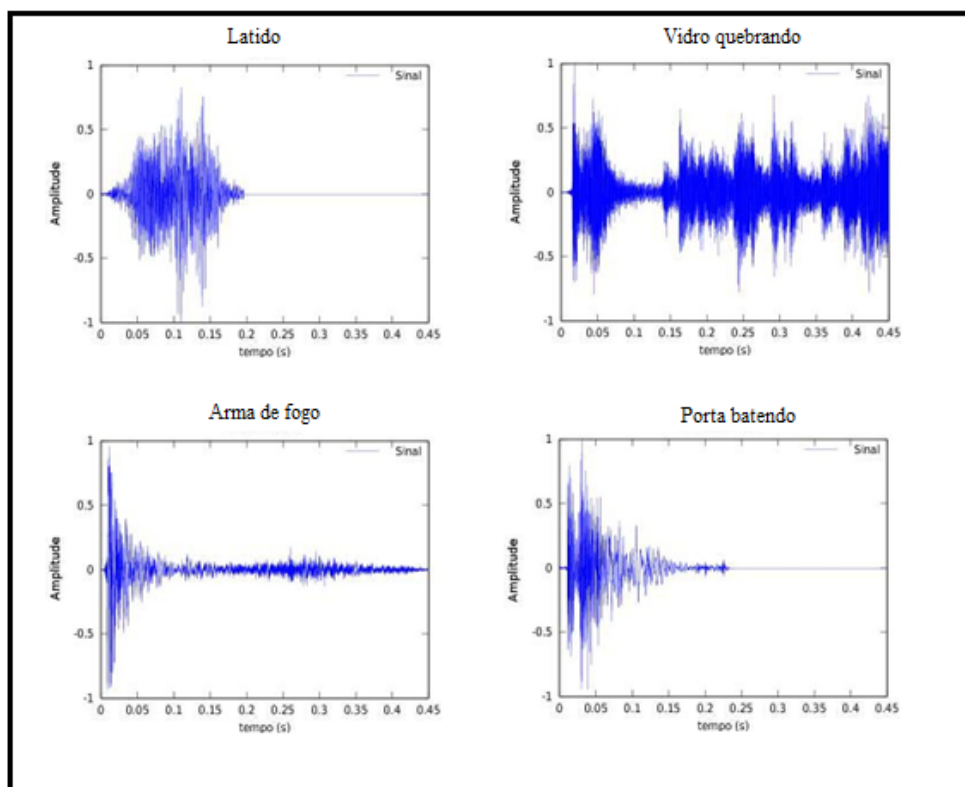
Fonte: (GOODHEW, 2021).

A imagem mostra que o valor analógico é estável em torno de 530 quando não há ruído externo. Se o microfone do módulo KY-038 ou KY-037 for afetado pelo som, a distância das duas placas que formam um capacitor dentro do microfone muda e, portanto, a tensão cai no microfone. Essa mudança de voltagem afeta diretamente a saída analógica do sensor de som. Portanto, se houver um ruído ativo, o valor do sensor analógico oscila em torno do valor estável.

2.3 Características específicas dos sons impulsivos

Sons impulsivos são, por definição, sons de curta duração, com amplitude e duração aleatórias, ou mais especificamente, são aqueles gerados por uma mudança na pressão do ar, produzindo uma grande amplitude com uma curta duração. Via de regra possuirá uma ascensão na casa dos milissegundos e duração na casa dos décimos de segundo (DUFAUX, 2001; VASEGHI, 2000). Eles são decorrentes de eventos que liberam muita energia sonora em um curto espaço de tempo, como é o caso dos instrumentos musicais de percussão, batidas de portas, vidros quebrando, explosões e disparos de armas de fogo (REIS, 2015). A figura 11 apresenta o perfil de alguns sons impulsivos em função do tempo.

Figura 11: Exemplos de sons impulsivos.



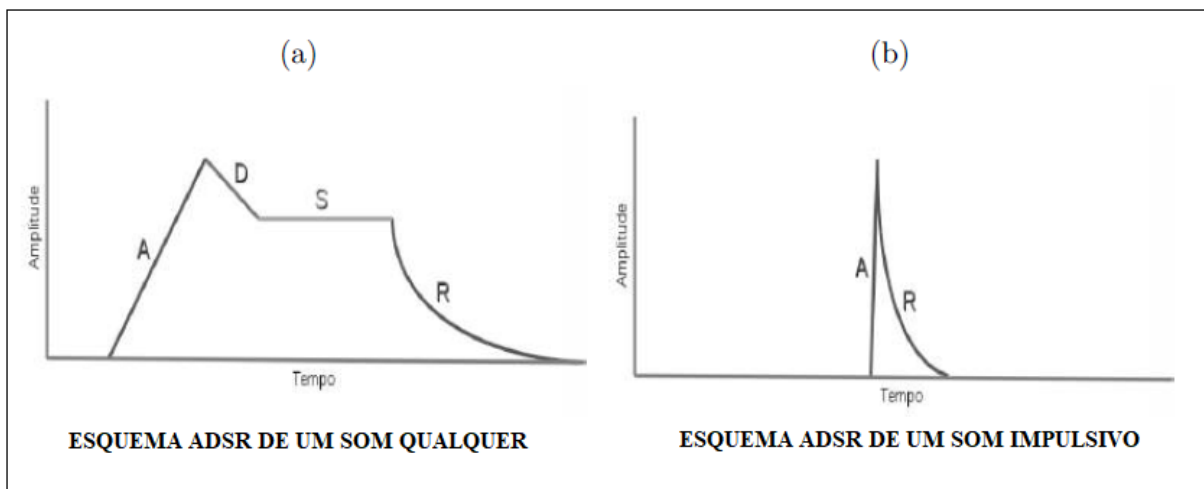
Fonte: (REIS, 2015).

Para um melhor entendimento do que seria um som impulsivo, toma-se por base o conceito ADSR (*Attack, Decay, Sustain, Release*).

Conforme observa-se na figura 12a, a fase do Ataque (*Attack*) demarca o crescimento e liberaçãõ da potênciã sonora. O decaimento (*Decay*) ´e representado por uma pequena perda da potênciã sonora inicial. A seguir a onda sonora passa por um processo de Sustentaçãõ (*Sustain*) pelo meio material que rodeia a fonte de emissãõ da onda sonora, como, por exemplo, uma caixa de ressonânciã. Finalmente a vibraçãõ sonora perde energia at´e extinguir-se, caracterizando a fase de Relaxamento (*Release*) (REIS, 2015).

Desta forma, caracteriza-se um som impulsivo como sendo aquele onde as fases do Decaimento e Sustentaçãõ praticamente nãõ existem, conforme ilustrado na figura 12b, e as fases do Ataque e Relaxamento ocorrem muito rapidamente, possuindo tempos da ordem de d´ecimos de segundos ou menos (REIS, 2015).

Figura 12: Esquema ADSR.



Fonte: (REIS, 2015).

Assim, um som impulsivo ideal possui´a um tempo de ataque muito curto, atingir´a o seu ma´ximo de energia muito rapidamente e nãõ se mantera´ neste estado de alta energia por muito tempo, iniciando o seu decaimento quase que instantaneamente, esvaindo-se r´apida e exponencialmente (REIS, 2015).

2.4 Root Mean Square (RMS)

O *Root Mean Square* (RMS), Valor Quadrático Médio ou Valor Eficaz é uma medida estatística de magnitude do sinal. Ela pode ser obtida pela equação 2.1 (KAUPPINEN, 2002).

$$r_n = \sqrt{\frac{1}{N} \sum_{n=1}^N x^2(n)} \quad (2.1)$$

onde N é o número de amostras consideradas para o cálculo do RMS, x é o valor de cada amostra do sinal e r_n é o valor do RMS. Em um som impulsivo a potência do sinal sobe abruptamente e isto reflete-se também no RMS que aumenta proporcionalmente.

2.5 Desvio padrão

Desvio padrão de uma amostra, de tipo quantitativo, é uma medida de dispersão dos dados relativamente à média, que se obtém tomando a raiz quadrada da variância amostral (MARTINS, 2013).

Uma vez que a variância amostral se exprime nas unidades dos dados elevados ao quadrado, considera-se como medida de dispersão, não a variância, mas a sua raiz quadrada. Se representarmos os dados por x_1, x_2, \dots, x_n , e por \bar{x} a sua média, o desvio padrão obtém-se a partir da equação 2.2 (MARTINS, 2013)

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (2.2)$$

O desvio padrão é uma medida que só pode assumir valores não negativos e quanto maior for o seu valor, maior será a dispersão dos dados (MARTINS, 2013).

Além da expressão anterior, por vezes também se utiliza a equação 2.3 (MARTINS, 2013)

$$s' = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}} \quad (2.3)$$

quando a dimensão da amostra n é suficientemente grande (é usual considerar um valor de n superior a 30).

2.6 Notificação Push

Os serviços de notificação por push ou push notifications, como são conhecidos, permitem o envio e recebimento de mensagens de forma confiável, eficiente e eficaz. Este método de comunicação é frequentemente utilizado em dispositivos móveis que usufruem dos serviços de notificação integrados aos sistemas operacionais. O Google Cloud Messaging® (GCM), por exemplo, é incorporado ao sistema Android®, podendo também ser incorporado em aplicativos para iOS® (KOLLMANN; BERESFORD, 2017).

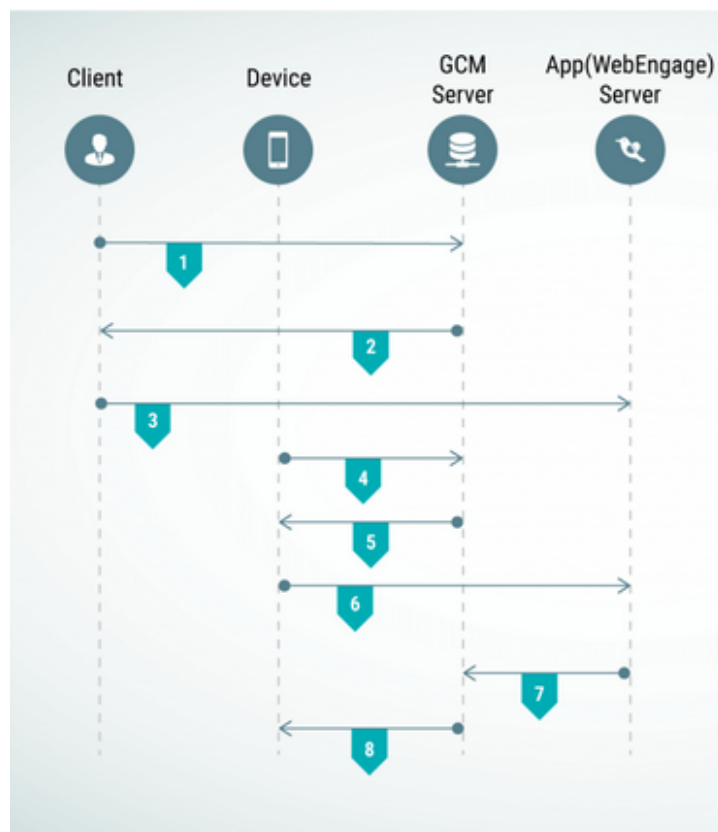
As notificações por push habilitam que os desenvolvedores e administradores de aplicativos se comuniquem com os usuários mesmo que seu aparelho esteja desligado, pois o serviço é capaz de armazenar a mensagem e entregá-la quando o dispositivo acordar. As interações entre cliente e servidor são todas realizadas sobre uma única conexão TCP de longa duração, amenizando problemas de rede e de sobrecarga do aparelho. Outra vantagem é que o algoritmo que gerencia as push notifications pode agrupar diversas notificações, destinadas a diferentes aplicações de um mesmo smartphone, em uma única mensagem, otimizando o desempenho, reduzindo processamento e aumentando a vida útil da bateria (KOLLMANN; BERESFORD, 2017).

Um sistema de notificação push fornece informações úteis e oportunas aos usuários por meio de dispositivos móveis ou dispositivos IoT. Vários tipos de informações podem ser entregues aos usuários conectando dispositivos a diferentes fontes de informação. Com o rápido desenvolvimento de dispositivos inteligentes, tecnologias de comunicação e aplicativos móveis, a notificação push está se tornando cada vez mais popular (PAN et al., 2015).

Na figura 13, pode-se observar o passo a passo de como funciona o envio de uma notificação push (TUCUNDUVA, 2021):

1. A empresa (cliente) solicita uma chave de API do servidor GCM (Google Cloud Messaging, o serviço de notificação por push do Google, que envia push tanto para dispositivos Android quanto iOS) para WebEngage do aplicativo;
2. O servidor GCM emite a chave de API;
3. A empresa insere essa chave de API no painel WebEngage do aplicativo;
4. O usuário faz o download do aplicativo no seu dispositivo que, por sua vez, envia um sinal ao servidor GCM para obter o ID de registro;
5. O GCM emite um ID de registro para o aplicativo no dispositivo;
6. O WebEngage SDK do aplicativo no celular do usuário envia o ID de registro para o servidor WebEngage geral do aplicativo;
7. O servidor WebEngage do app envia a mensagem para o servidor GCM; e
8. O GCM envia a mensagem para o dispositivo do usuário e o WebEngage SDK processa as informações e as mostra na tela do celular.

Figura 13: Fluxo de notificações de push.



Fonte: (TUCUNDUVA, 2021).

2.7 Protocolos de comunicação para IoT

Alguns protocolos de comunicação podem ser empregados na área de IoT. Existem diversas possibilidades do emprego de alguns protocolos *wireless*, que possibilitam o emprego na área de IoT, tais como: Wi-Fi, (*Bluetooth Low Energy*) BLE, SigFox e LoRa.

2.7.1 Protocolo Wi-Fi (802.11)

Redes Wi-Fi são redes sem fio que estão em conformidade com os padrões 802.11 do *Institute of Electrical and Electronics Engineers* (IEEE), que é o grupo que lidera a padronização de redes locais e Metropolitanas. De acordo com as normas deste padrão, as redes sem fio podem operar em dois modos: Ad-Hoc e Infra-Estruturado. (PASQUALINI; MARCONDES, 2012).

Os protocolos Wi-Fi operam nas faixas de 2,4 GHz e 5 GHz. A topologia empregada é a tipo estrela, possuindo alta taxa de transmissão de dados. Contudo, possui baixo raio de cobertura. Diante disso, pode-se afirmar que este protocolo se apresenta como uma opção para aplicações que não exijam grandes áreas de cobertura.

2.7.2 BLE

O BLE foi projetado para operação com energia muito baixa. Para permitir uma operação confiável na banda de frequência de 2,4 GHz, ele aproveita uma abordagem de espectro espalhado de salto de frequência robusto que transmite dados em 40 canais. Além disso, oferece aos desenvolvedores uma enorme flexibilidade, incluindo várias opções de PHY que suportam taxas de dados de 125 Kb / s a 2 Mb / s, vários níveis de potência, de 1mW a 100 mW, bem como várias opções de segurança de nível governamental (BLUETOOTH, 2021).

O BLE também oferece suporte a várias topologias de rede, incluindo ponto a ponto, transmissão e rede em malha (BLUETOOTH, 2021).

2.7.3 SigFox

Uma das tecnologias de comunicação mais importantes atualmente no setor de IoT é a Sigfox. Esta é uma tecnologia proprietária da empresa que leva seu próprio nome e é baseada

em uma modulação diferencial DBPSK para upload para a plataforma e uma modulação GFSK para download de dados. Em ambos os sentidos trabalha com a tecnologia de comunicação “Ultra Narrow Band” (UNB), transmitindo nas bandas de frequência sub-GHz livres (PARDAL GARCÉS, 2017).

Os pacotes transmitidos são de pequeno tamanho, constituídos por uma parte fixa de 12 bytes que inclui um preâmbulo, um identificador de dispositivo e outros metadados, e a parte variável formada pela informação, podendo ser até 12 bytes. No total, cada pacote pode variar de 12 a 24 bytes mais alguns bits extras usados para a autenticação dos parâmetros. Ao longo de um dia, o Sigfox permite que até 140 mensagens sejam enviadas, cada uma levando aproximadamente 2 segundos para chegar à estação base (PARDAL GARCÉS, 2017).

Um recurso a ser observado é que o Sigfox é eficaz para comunicações de terminais para as estações base, com uma taxa de 100 bps, mas não é particularmente eficiente nas transmissões das estações base para os pontos finais. Os downloads são mais lentos do que os uploads. O escopo da tecnologia varia dependendo do ambiente do aplicativo, de 30 a 50 km em ambientes rurais a 3 a 10 km em ambientes urbanos (PARDAL GARCÉS, 2017).

2.7.4 LoRa

A arquitetura de comunicação LoRa é baseada em uma conexão através de gateways, que funcionam como um gateway para a plataforma IoT (PARDAL GARCÉS, 2017).

Essa tecnologia usa uma estratégia de espalhamento espectral para transmitir em uma variedade de frequências e taxas de dados. Isso permite que o gateway se adapte às condições de mudança e otimize a maneira como ele troca dados com cada dispositivo. Pode transmitir em frequências pertencentes às bandas de 109 MHz, 433 MHz, 866 MHz e 915 MHz, que oferecem menos interferências do que outras superiores, como a banda de 2,4 GHz. Nessas frequências os sinais penetram em obstáculos e alcançam longas distâncias com baixo consumo de energia (PARDAL GARCÉS, 2017).

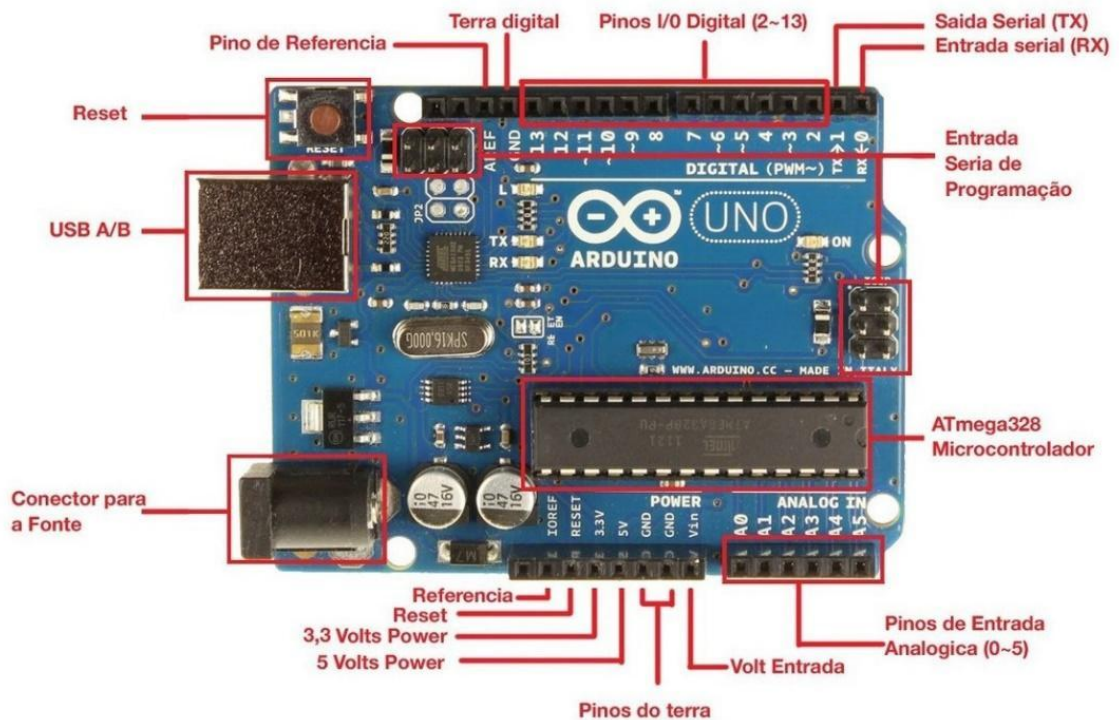
Além disso, o LoRa oferece funcionalidade bidirecional eficaz, tornando-o bom no envio de mensagens de dispositivos para estações base e de estações base para terminais. Tudo isso com uma taxa de transferência entre 300 e 5000 bits por segundo (PARDAL GARCÉS, 2017).

3 DESENVOLVIMENTO DO PROJETO

Como passo inicial, foi realizada a instalação do *software* Arduino. Para isso, é necessário acessar o endereço eletrônico <https://www.arduino.cc/en/Main/Software>. Após acesso ao endereço, deve-se clicar na imagem Arduino referente ao sistema operacional do usuário e iniciar o download.

No início deste trabalho, a plataforma de prototipagem eletrônica escolhida para desenvolver o sensor de detecção de som impulsivo foi o módulo Arduino UNO. Esta placa é amplamente conhecida e possui uma aplicação muito simples e prática. Além disso, existem diversos projetos disponíveis na Internet com seu uso. Diante disso, fica claro o motivo de tê-lo escolhido inicialmente para este projeto em questão.

Figura 14: Placa Arduino UNO.



Fonte: (TECH SUL ELETRÔNICOS, 2021).

Em complemento a esta placa Arduino UNO, foi utilizado o módulo sensor de som KY-038 para que fosse possível captar o sinal sonoro.

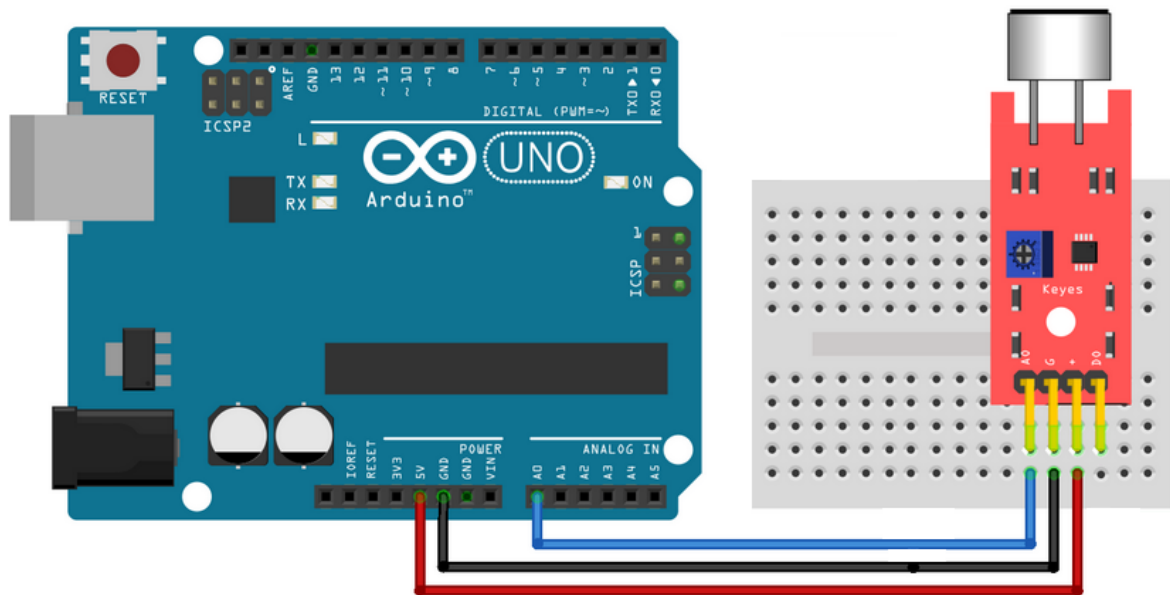
Figura 15: Módulo sensor de som KY-038.



Fonte: (ARDUINO E CIA, 2015)

Para implementar o detector de sons impulsivos, foi necessário conectar o módulo ky-038 à placa Arduino UNO conforme ilustrado na figura 16.

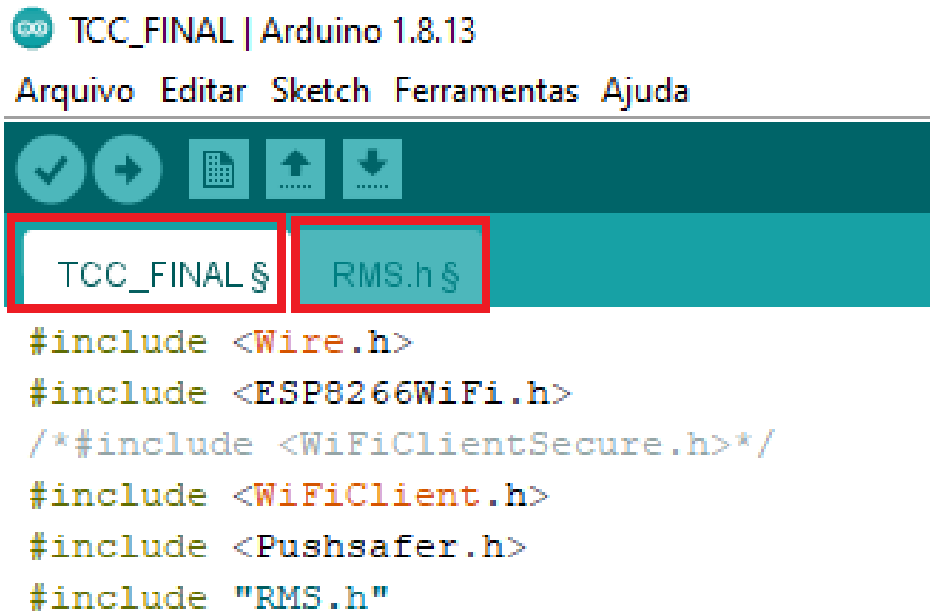
Figura 16: Diagrama de conexão da placa Arduino UNO com o módulo KY-038.



Fonte: Próprio Autor.

Para o desenvolvimento do código de programação na IDE Arduino, foi realizada sua divisão em duas abas diferentes “TCC_FINAL” e “RMS.h”, as quais são apresentadas nos Apêndices A e B, respectivamente.

Figura 17: Identificação das abas referentes ao código de programação.



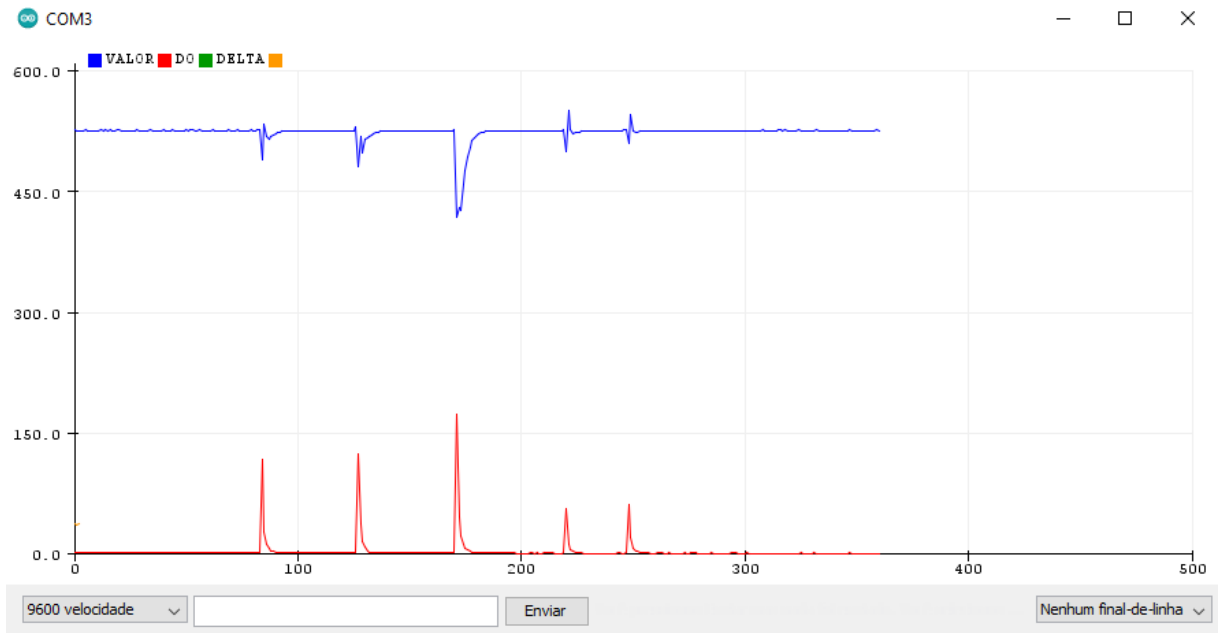
Fonte: Próprio autor.

Inicialmente, utilizou-se o cálculo do RMS para realizar a análise dos sinais sonoros captados, conforme a equação 2.1. Para isso, os valores de cada amostra do sinal inseridos nesta fórmula foram os valores analógicos obtidos pelo módulo KY-038. Contudo, pode-se observar no serial plotter do IDE Arduino, conforme a figura 18, que os valores do RMS se mantinham constantes em 530 quando não havia nenhuma emissão sonora e, ao emitir qualquer sinal sonoro, oscilava para valores acima e abaixo de 530, não podendo, assim, fornecer uma análise muito precisa do comportamento dos sinais sonoros.

Diante disso, optou-se pela implementação do cálculo do desvio padrão (método amplamente utilizado em laboratórios) para analisar os sinais sonoros, conforme a equação 2.3. Para isso, os valores dos dados inseridos nesta fórmula foram os valores analógicos obtidos pelo módulo KY-038.

Analisando o plotter do IDE Arduino, conforme a figura 18, pôde-se observar que os valores do desvio padrão se mantiveram constantes em zero quando não havia nenhuma emissão sonora e, ao emitir qualquer sinal sonoro, oscilava, assumindo apenas valores positivos, o que permitiu, assim, analisar de forma mais precisa o comportamento dos sinais sonoros.

Figura 18: Ilustração do serial plotter da IDE Arduino dos valores do RMS (azul) e Desvio Padrão (vermelho) variando no tempo.



Fonte: Próprio autor.

A partir daí, foi implementado o código de forma que o detector pudesse identificar um som impulsivo utilizando o cálculo do desvio padrão como base.

Conforme abordado anteriormente no subitem 2.3 deste trabalho, sons impulsivos são gerados por uma súbta mudança na pressão do ar, produzindo uma grande amplitude com uma curta duração e, com isso, possuirão uma ascensão na casa dos milissegundos. Tomando esse conceito como base e realizando diferentes experimentos a partir da emissão de sons impulsivos e não impulsivos, foi possível obter alguns resultados.

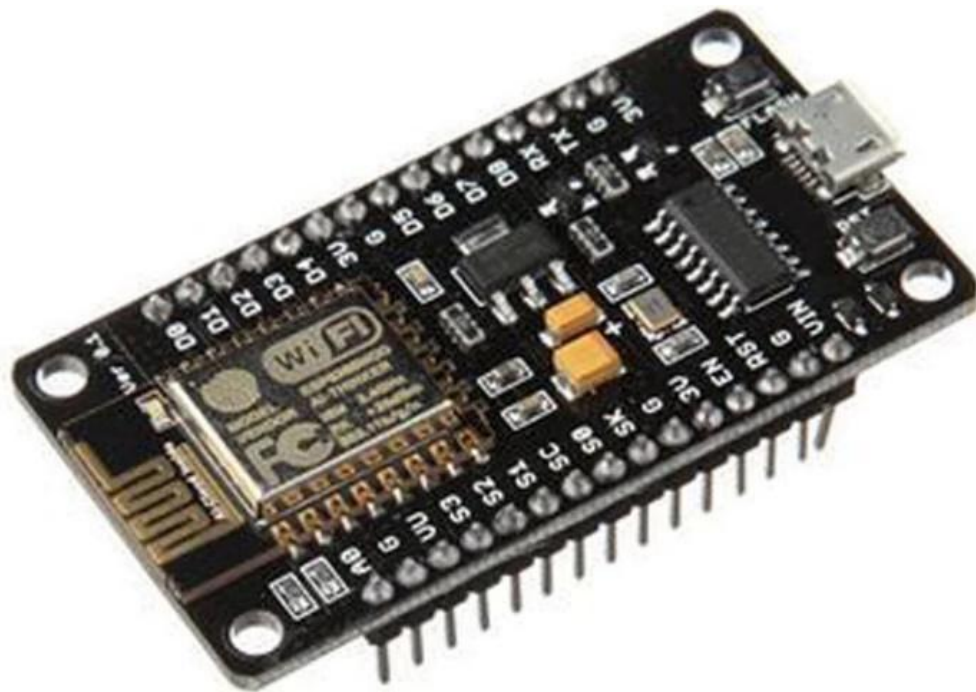
Os sons impulsivos captados, oriundos dos impactos de um objeto sobre uma mesa sempre indicaram uma duração dentro de uma faixa de 20 a 40 milissegundos e com valores de amplitude acima de 100. Além disso, os experimentos feitos com sons não impulsivos, como a pronúncia de alguma letra do alfabeto ou de alguma palavra sempre indicavam uma duração acima de 200 milissegundos e amplitude aleatória.

A partir daí, foi implementado, no código de programação, um limiar de amplitude do impulso no valor de 100 e um limiar de tempo do impulso no valor de 50 milissegundos. Dessa forma, todas as vezes em que o microfone captava sons que satisfizessem essas condições de limiaridade, o texto DETECTEI UM IMPULSO era impresso no monitor serial da IDE Arduino.

Ao finalizar o código, necessitava-se ainda encaminhar a informação de detecção do som impulsivo à algum dispositivo de comunicação para implementar ao trabalho o conceito de Internet das Coisas. O protocolo de comunicação para IoT utilizado foi o Wi-Fi e, para isso, foi necessário adquirir no mercado outra placa que pudesse atender à essa necessidade.

Diante disso, adquiriu-se uma placa de prototipagem eletrônica da família do ESP 8266, o NODE MCU ESP-12E. Esta placa é uma solução à altura da tarefa proposta, visto que é um circuito totalmente integrado com interfaces de entrada/saída digitais e analógicas e, ainda, interface Wi-Fi e possui um microprocessador Tensilica Xtensa LX106 RISC de 32 bits que opera em uma frequência de clock de 160 MHz. Além disso, também possui 128 KB de RAM e 4 MB de memória FLASH e seu chip ESP-12E opera com a rede Wi-Fi na frequência de 2,4 GHz.

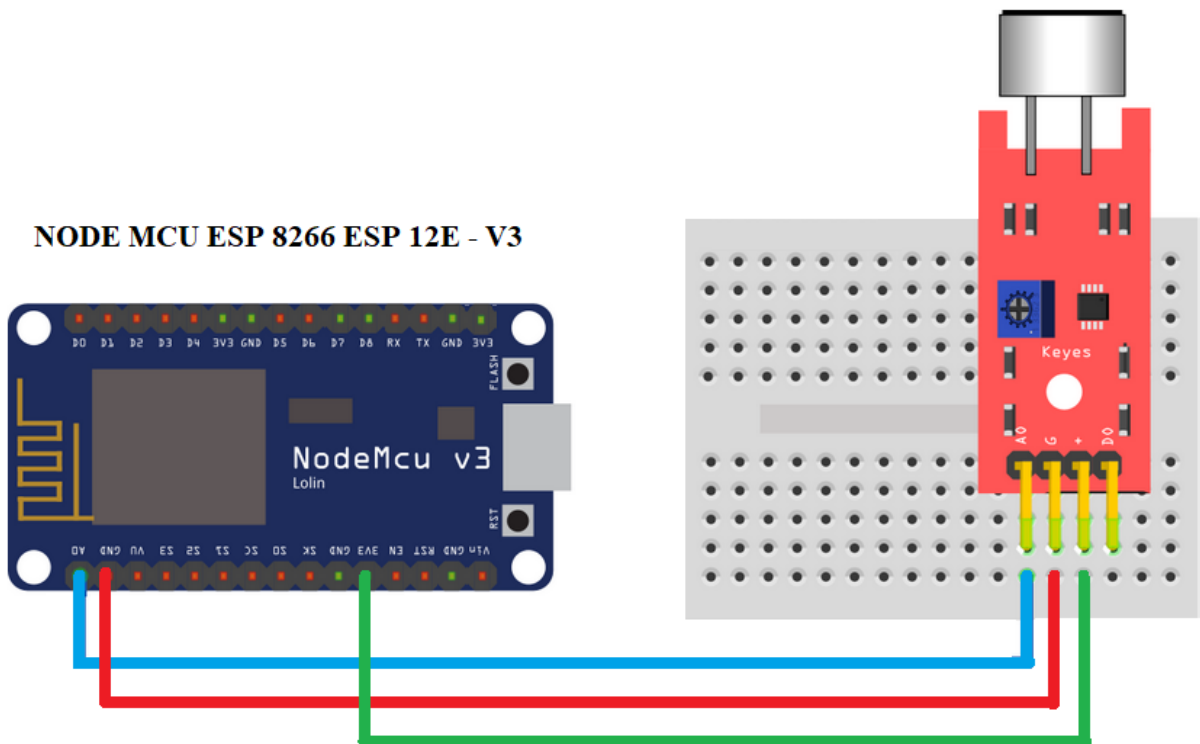
Figura 19: NODE MCU ESP-12E.



Fonte: Próprio autor.

A partir daí, o detector passou a ser implementado pelo NODE MCU ESP-12E em conexão com o módulo KY-038, conforme apresentado na figura 20.

Figura 20: Diagrama de conexão do NODE MCU ESP-12 com o módulo KY-038.



Fonte: Próprio autor.

Com isso, houve a necessidade de desenvolver o projeto em questão de forma que fosse possível realizar a conexão da placa com algum dispositivo, utilizando protocolo Wi-Fi.

Para isso, utilizou-se a tecnologia push através do aplicativo PUSH SAFER. Inicialmente, realizou-se o cadastro no site <https://www.pushsafer.com/en/pushapi>, gerando login e senha e, a partir daí, ao acessar a conta, foi possível identificar a chave privada.

O dispositivo utilizado para realizar a comunicação com a placa foi um celular Iphone, no qual foi preciso realizar o download do aplicativo PUSH SAFER, acessando a conta em questão.

O próximo passo foi desenvolver e incrementar o código de programação para que fosse possível realizar esta comunicação. Além disso, foi necessário realizar o download da biblioteca que permitisse implementar a tecnologia push e, através do ícone sketch da IDE Arduino, incluí-la na lista de bibliotecas.

As informações foram acessadas pelo IP 192.168.0.42 (referente ao sensor Wi-Fi NODE MCU ESP-12E). Após o carregamento do código, o endereço IP da placa foi encontrado no *monitor serial* da IDE Arduino, o qual passou a exibir os valores que estavam sendo captados

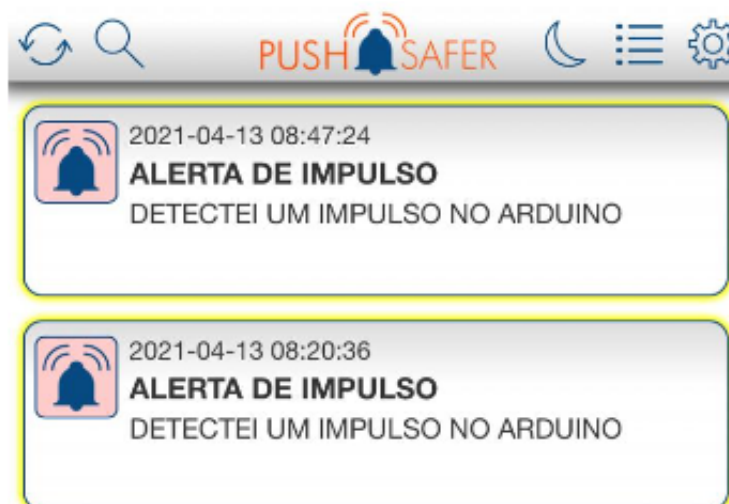
pelo microfone do módulo KY-038. Assim que a condição limite é satisfeita, detecta-se um som impulsivo e o *monitor serial* da IDE Arduino imprime o texto VALOR DO DELTA, que indica a duração deste sinal sonoro em milissegundos, assim como o texto DETECTEI UM IMPULSO, conforme se pode observar na figura 21. Além disso, através da tecnologia push, é encaminhada uma informação com os textos ALERTA DE IMPULSO e DETECTEI UM IMPULSO NO ARDUINO para o celular Iphone utilizado para o experimento, informando a data e o horário exato da detecção sonora pelo microfone do módulo KY-038, conforme se pode observar na figura 22.

Figura 21: Imagem do monitor serial da IDE Arduino ao detectar um impulso.



Fonte: Próprio autor.

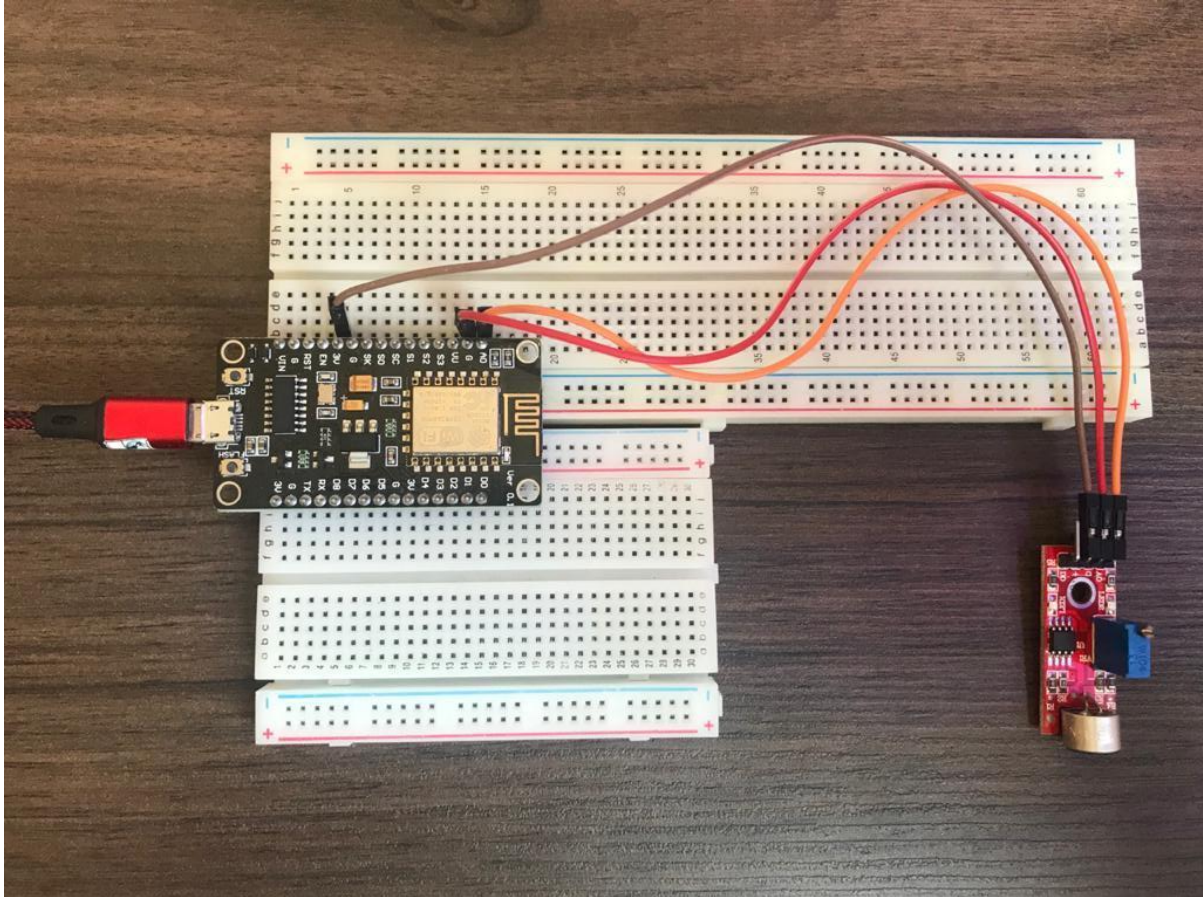
Figura 22: Imagem do aplicativo PUSH SAFER ao receber um alerta de impulso.



Fonte: Próprio autor.

Após todo o estudo levantado e desenvolvimento detalhado, chegou-se ao resultado planejado, concluindo o projeto de um detector de sons impulsivos. A figura 23 apresenta o detector montado, representando o projeto final.

Figura 23: Detector de sons impulsivos finalizado.



Fonte: Próprio autor.

4 CONCLUSÃO

Este Trabalho de Conclusão de Curso apresentou uma breve revisão sistemática sobre aplicação de Internet das Coisas como suporte em um projeto de detecção de sons impulsivos. Diante da problemática em questão, o desenvolvimento do detector em si exigiu a busca de tecnologias de IoT para notificação de alertas do sensor para o smartphone (push messages) e de técnicas para análise de sons impulsivos com base em RMS e desvio padrão.

Neste cenário, à luz da evolução da tecnologia em curso, desenvolveu-se um detector de sons impulsivos, com notificação de alertas via Wi-Fi, sendo essa tecnologia factível, de baixo custo e com potencial para ser desenvolvida a fim de se concluir um detector de disparo de

arma de fogo, visto que a detecção de sons impulsivos é um passo essencial para se concluir um projeto que possa discernir um som impulsivo de um disparo de arma de fogo.

Dessa forma, pode-se inferir que o uso da IoT pode ser útil no apoio às operações militares de garantia da lei e da ordem, através do emprego de sensores de detecção automática de disparo de armas de fogo conectados à Internet e fixados em pontos estratégicos do Estado do Rio de Janeiro. Essa abordagem viabiliza identificar a localização e o instante exato de um determinado disparo de forma que, tanto a Polícia Militar quanto o Corpo de Fuzileiros Navais empregados nestas operações, possam obter, automaticamente, a informação do local e do momento em que foi realizado algum disparo. Dessa forma, as ações em resposta a esses eventos podem ser realizadas de forma mais rápida e precisa.

Dentre as possibilidades existentes no mercado de placas de prototipagem eletrônica, pôde-se observar que o NODE MCU ESP-12 da família ESP8266 foi aquele que se encaixou dentro do propósito deste trabalho, devido à facilidade de possuir integração à rede Wi-Fi. Além disso, dentre as opções existentes de módulos de detecção de onda sonora, observou-se que o módulo KY-038 possui as características necessárias para atender a problemática em questão. Diante disso, pode-se afirmar que ambos atenderam a proposta deste trabalho.

Foram realizados alguns experimentos com o detector e, assim que satisfeitas as condições limites, foi possível detectar os sons impulsivos gerados pelo impacto de um objeto sobre uma mesa. Através da tecnologia *push*, a informação de detecção do som impulsivo foi encaminhada ao celular Iphone utilizado no experimento. Diante disso, chegou-se ao resultado desejado, obtendo-se, assim, o funcionamento do detector de sons impulsivos utilizando o conceito de Internet das Coisas para notificação de alertas.

Face ao exposto, em resposta à problemática sugerida por este TCC, fica claro que é possível a criação de um detector de sons impulsivos utilizando o conceito de Internet das coisas, podendo este protótipo inicial ser futuramente desenvolvido a fim de se concluir um projeto de um sensor capaz de detectar um disparo de arma de fogo.

4.1 Trabalhos futuros

Dente as oportunidades de trabalhos futuros, destacam-se alguns pontos, como o desenvolvimento, a partir deste detector de sons impulsivos, de um módulo de reconhecimento de disparo (módulo de classificação) utilizando algoritmos que permitam diferenciar um som impulsivo qualquer de um disparo de arma de fogo, utilizando técnicas no domínio do tempo ou transformadas do sinal para o domínio das frequências.

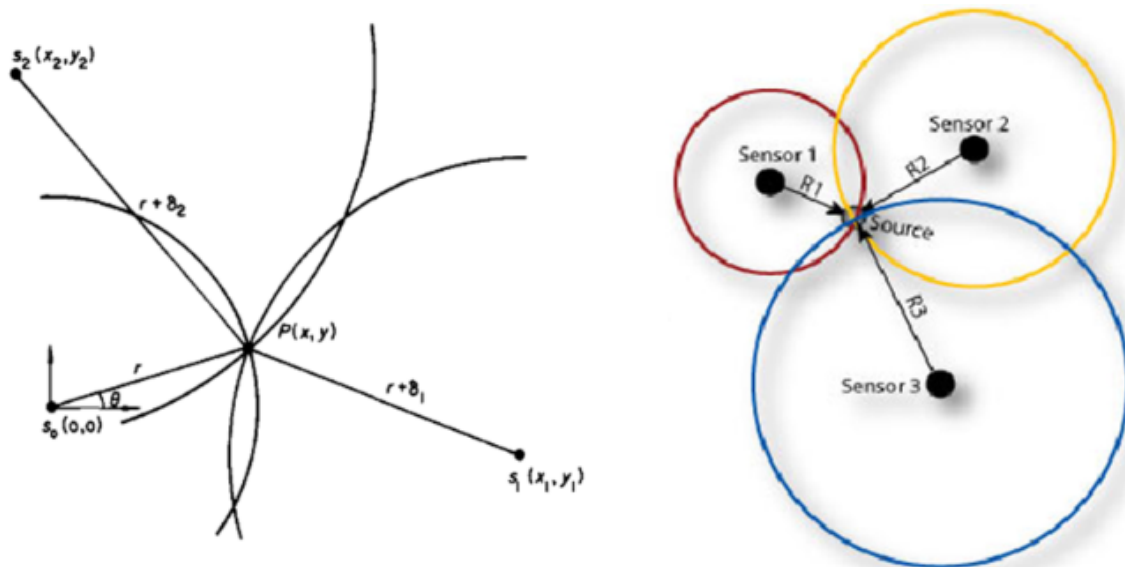
Outro ponto a ser inserido em trabalhos futuros é a realização da triangulação de forma que se possa identificar a localização precisa e o instante exato da emissão da fonte sonora.

A triangulação é o processo de determinar a localização de um ponto, medindo apenas ângulos para ele a partir de pontos conhecidos em cada extremidade de uma linha de base fixa, em vez de medir distâncias para o ponto diretamente como na trilateração. O ponto pode então ser fixado como o terceiro ponto de um triângulo com um lado conhecido e dois ângulos conhecidos. Ele se inspira na técnica de triangulação em topografia, onde o alcance de um ponto distante pode ser encontrado usando medições de ângulos múltiplos feitas a partir de uma linha de base bem definida (TEKDAS; ISLER, 2010).

Tobias (1976) desenvolveu um método geral para calcular a localização em duas dimensões a partir dos tempos de chegada da onda sonora nos sensores.

A intersecção de círculos indica a localização da fonte de emissão acústica (EA). Para isso, são utilizados os valores das distâncias entre os sensores, diferença de tempo entre os mesmos, velocidade de propagação da onda no material e posição dos microfones (OLIVEIRA, 2014).

Figura 24: Ilustração de três sensores S1, S2 e S3 e uma fonte de emissão acústica P.



Fonte: (TOBIAS, 1976).

O método se vale de três círculos (Figura 24), ou seja, são utilizados três sensores para o desenvolvimento. A fonte de emissão acústica é localizada no ponto de intersecção dos círculos relativos aos centros que o autor nomeia de S0, S1 e S2. Estes centros são os próprios

microfones. Os centros possuem raios r , $r+\delta_1$ e $r+\delta_2$, os quais $\delta_1=t_1*V$, $\delta_2=t_2*V$ e “ r ” é calculado pela equação a seguir (OLIVEIRA, 2014):

$$R = \frac{1}{2} \frac{D_1^2 - \Delta t_1^2 V^2}{\Delta t_1 V + D_1 \cos(\theta - \theta_1)}$$

Onde:

R = distância do sensor 1 até a fonte EA (m)

D_1 = distância entre sensores 1 e 2 (m)

Δt_1 = diferença de tempo de chegada da onda entre os sensores 1 e 2 (s)

V = velocidade da onda-P no material (m/s)

Θ = ângulo (radianos) entre um plano horizontal e a fonte EA

Θ_1 = ângulo (radianos) entre o sensor 1 e o sensor 2

Considerando um arranjo arbitrário de três sensores localizado em pontos chamados de $S_0(0,0)$, $S_1(x_1,y_1)$ e $S_2(x_2,y_2)$, conforme a Figura 23 (OLIVEIRA, 2014).

Uma fonte de Emissão Acústica $P(x,y)$, uma distância r de S_0 , irá gerar as diferenças δ_1 e δ_2 definidas por $\delta_1 = PS_1-PS_0 = t_1*v$ e $\delta_2 = PS_2-PS_0 = t_2*v$, aonde v é a velocidade de propagação da onda no material e t_1 e t_2 são as diferenças nos tempos de chegada do ruído medidas a partir dos sensores S_1-S_0 e S_2-S_0 , respectivamente. A fonte de emissão acústica no ponto $P(x,y)$ é localizada no ponto de intersecção dos círculos formados por S_0 , S_1 e S_2 , com raios r , $r+\delta_1$ e $r+\delta_2$ respectivamente. As equações dos três círculos são descritas a seguir (OLIVEIRA, 2014):

$$(I) \quad x^2 + y^2 = r^2$$

$$(II) \quad (x - x_1)^2 + (y - y_1)^2 = (r + \delta_1)^2$$

$$(III) \quad (x - x_2)^2 + (y - y_2)^2 = (r + \delta_2)^2$$

Com isso, pode-se afirmar que é possível inserir o conceito de triangulação na proposta deste TCC. Portanto, sugere-se que seja inserido em trabalhos futuros.

REFERÊNCIAS

AL-DOGHMAN, F.; CHACZKO, Z.; JIANG, J. **A review of aggregation algorithms for the Internet of things**. 2017 25th International Conference on Systems Engineering (ICSEng). **Anais...IEEE**, 2017

AL-FUQAHA, A. et al. Internet of things: A survey on enabling technologies, protocols, and applications. **IEEE communications surveys & tutorials**, v. 17, n. 4, p. 2347–2376, 2015.

ARDUINO E CIA. **Medindo a intensidade do som com o sensor KY-038**. Disponível em: <<https://www.arduinoecia.com.br/sensor-de-som-ky-038-microfone-arduino/>>. Acesso em: 6 abr. 2021.

BLUETOOTH. **Bluetooth Radio Versions**. Disponível em: <<https://www.bluetooth.com/learn-about-bluetooth/radio-versions/>>. Acesso em: 5 abr. 2021.

DUFAUX, A. Detection and recognition of impulsive sound signals [Ph. D. thesis]. **University of Neuchatel, Neuchatel, Switzerland**, 2001.

EMBEDDED SYSTEMS. **Estrutura básica de um microcontrolador genérico**. Disponível em: <<https://bit-by-bit.gitbook.io/embedded-systems/sistemas-microcontrolados/estrutura-basica-d-e-um-microcontrolador-generico>>. Acesso em: 2 abr. 2021.

FEZARI, M.; AL DAHOUD, A. Integrated Development Environment “IDE” For Arduino. **WSN applications**, p. 1–12, 2018.

GOODHEW, A. **Technical Datasheet of the KY-038 / KY-037 Sound Sensor**. Disponível em: <<https://diyi0t.com/sound-sensor-arduino-esp8266-esp32/>>. Acesso em: 3 abr. 2021.

KAUPPINEN, I. **Methods for detecting impulsive noise in speech and audio signals**. 2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No. 02TH8628). **Anais...IEEE**, 2002

KOLLMANN, S. A.; BERESFORD, A. R. **The Cost of Push Notifications for Smartphones using Tor Hidden Services**. 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). **Anais...IEEE**, 2017

LUTKEVICH, B. **Microcontroller (MCU)**. Disponível em: <<https://Internetofthingsagenda.techtarget.com/definition/microcontroller>>. Acesso em: 2 abr. 2021.

MARINHA DO BRASIL. **Fuzileiros Navais contribuem para a normalização da rotina na Rocinha**. Disponível em: <<https://www.marinha.mil.br/cgcfm/node/377>>. Acesso em: 28 mar. 2021.

MARTINS, M. E. G. Desvio padrão amostral. **Revista de ciência elementar**, v. 1, n. 1, 2013.

OLIVEIRA, A. M. DE. Estudo da localização e caracterização de defeitos utilizando a técnica de emissão acústica. 2014.

PAN, Z. et al. **Intelligent push notification for converged mobile computing and Internet of things**. 2015 IEEE International Conference on Web Services. **Anais...IEEE**, 2015

PARDAL GARCÉS, P. Redes de Área Extensa para aplicaciones de IoT: modelado de comunicaciones Sigfox. 2017.

PASQUALINI, A. L.; MARCONDES, C. A. C. Estudo do Crescimento das Redes Wireless 802.11–2.4 GHz em Ambiente Urbano–Caso Rio Claro-SP. **Revista TIS**, v. 1, n. 2, 2012.

REIS, C. F. DOS. Sistema Modular para Detecção e Reconhecimento de Disparos de Armas de Fogo. 2015.

STRAUB, M. G. **Arduino IDE – O Software para Gravação de Códigos no Arduino**.

Disponível em:

<<https://www.usinainfo.com.br/blog/arduino-ide-o-software-para-gravacao-de-codigos-no-arduino/>>. Acesso em: 29 mar. 2021.

TECH SUL ELETRÔNICOS. **Arduino UNO**. Disponível em:

<<https://techsuleletronicos.com.br/product/arduino-uno-com-atmega328-r3/>>. Acesso em: 6 abr. 2021.

TEKDAS, O.; ISLER, V. Sensor placement for triangulation-based localization. **IEEE transactions on Automation Science and Engineering**, v. 7, n. 3, p. 681–685, 2010.

THOMSEN, A. **O que é Arduino?** Disponível em:

<<https://www.filipeflop.com/blog/o-que-e-arduino/>>. Acesso em: 2 abr. 2021a.

THOMSEN, A. **Tipos de Arduino**. Disponível em:

<<https://www.filipeflop.com/blog/tipos-de-arduino-qual-comprar/>>. Acesso em: 2 abr. 2021b.

THOMSEN, A. **Qual módulo ESP8266 comprar?** Disponível em:

<<https://www.filipeflop.com/blog/qual-modulo-esp8266-comprar/>>. Acesso em: 2 abr. 2021.

TOBIAS, A. Acoustic-emission source location in two dimensions by an array of three sensors. **Non-destructive testing**, v. 9, n. 1, p. 9–12, 1976.

TUCUNDUVA, R. **O que é push notification?** Disponível em:

<<https://blog.lahar.com.br/midias-sociais/o-que-e-push-notification/>>. Acesso em: 4 abr. 2021.

VASEGHI, S. V. **Advanced digital signal processing and noise reduction**. [s.l.] John Wiley, 2000.

APÊNDICE A – Aba “TCC_FINAL” da IDE Arduino referente ao código de programação do detector de sons impulsivos

```

#include <Wire.h>
#include <ESP8266WiFi.h>
/*#include <WiFiClientSecure.h>*/
#include <WiFiClient.h>
#include <Pushsafer.h>
#include "RMS.h"

// Inicialize a conexão Wifi ao roteador
const char* ssid = "NET_2GD12DE2";
const char* password = "4FD12DE2";

// Pushsafer private or alias key
#define PushsaferKey "XfeNsW5CKmIaKgYw5lKk"

/*WiFiClientSecure client;*/
WiFiClient client;
Pushsafer pushsafer(PushsaferKey, client);

int valorAnalogico;

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

```

```
// Set WiFi to station mode and disconnect from an AP if it was Previously
// connected
WiFi.mode(WIFI_STA);
WiFi.disconnect();
delay(100);

// Attempt to connect to Wifi network:
Serial.print("Connecting Wifi: ");
Serial.println(ssid);
WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
}

Serial.println("");
Serial.println("WiFi connected");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());

pushsafer.debug = true;

pinMode(A0, INPUT);

//Faz uma pausa de 5 segundos antes de começar a exibir os resultados
delay(5000);

}

float calculaMedia(int vetorAmostra[], int amostras)
{
  float media = 0.0;
```

```
int i = 0;
while (i < amostras) {

media = media + vetorAmostra[i];
i++;
}
media = media / amostras;
return media;

}

float calculaDesvioPadrao(int vetorAmostra[], int amostras)
{
float valorMedio = 0.0;
float somatorio = 0.0;
int i = 0;

valorMedio = calculaMedia(vetorAmostra, amostras);

while (i < amostras)
{

somatorio = somatorio + pow((vetorAmostra[i] - valorMedio),2.0);
i++;

}

return sqrt(somatorio / amostras);

}

void enviandoNotificacao() {

struct PushSaferInput input;
```

```

input.message = "DETECTEI UM IMPULSO NO ARDUINO";
input.title = "ALERTA DE IMPULSO";
input.sound = "8";
input.vibration = "1";
input.icon = "1";
input.iconcolor = "#FFCCCC";
input.priority = "1";
input.device = "a";
input.retry = "";
input.expire = "";
input.answer = "";

Serial.println(pushsafer.sendEvent(input));
Serial.println("Sent");
}

void loop()
{

int i = 0; //variável para iteração
float leitura = 0.0; //armazena as leituras
Vrms = 0.0; //reinicia a variável Vrms
float desvioPadrao = 0.0;

while (i < amostras) { //inicia um ciclo de amostragem até que i alcance o número de amostras
leitura = analogRead(leituraTensao); //lê a porta analógica

//Vrms = Vrms + pow(leitura, 2.0); //calcula a soma dos quadrados das tensões lidas
Vrms = Vrms + pow(leitura, 2.0);

vetorAmostra[i]=leitura;

i++; //incrementa o iterador
}

```



```
Vrms = (sqrt(Vrms / amostras));
```

```
desvioPadrao = calculaDesvioPadrao(vetorAmostra, amostras);
```

```
if (cronometro == PARADO && desvioPadrao > limiarAmplitudeImpulso)
```

```
{
```

```
t1 = millis();
```

```
cronometro = DISPARADO;
```

```
}
```

```
if (cronometro == DISPARADO && desvioPadrao < limiarAmplitudeImpulso)
```

```
{
```

```
t2 = millis();
```

```
cronometro = PARADO;
```

```
deltaT = t2 - t1;
```

```
Serial.print("VALOR DO DELTA: ");
```

```
Serial.println(deltaT);
```

```
if (deltaT < limiarTempoImpulso)
```

```
{
```

```
Serial.println ("DETECTEI UM IMPULSO...");
```

```
enviandoNotificacao();
```

```
}
```

```
}
```

```
//detecta se é um valor é máximo
```

```
if (Vrms > Vmax) {
```

```
Vmax = Vrms;
```

```
}
```

```
//detecta se é um valor mínimo
if (Vrms < Vmin) {
  Vmin = Vrms;
}

//calcula a média dos valores máximo e mínimo atuais
Vmed = (Vmax + Vmin) / 2.0;

//saída formatada para plotter serial IDE Arduino
Serial.print(Vrms);
Serial.print(" , ");
Serial.println(desvioPadrao);

}
```

APÊNDICE B – Aba “RMS.h” da IDE Arduino referente ao código de programação do detector de sons impulsivos

```
#define leituraTensao A0

const int amostras = 200;

float Vrms = 0.0; //armazena o valor rms da tensao
float Vmax = 0.0; //armazena o valor máximo detectado
float Vmin = 10000.0; //armazena o valor mínimo detectado
float Vmed = 0.0; //armazena o valor medio entre Vmax e Vmin
float desvioPadrao = 0.0; //armazena o valor desvio padrao
float limiarAmplitudeImpulso = 100.0; // limiar de amplitude do impulso
int vetorAmostra[amostras];

enum estado_cronometro
{
  PARADO, DISPARADO
} cronometro = PARADO;

unsigned long t1=0, t2=0, deltaT=0;
unsigned long limiarTempoImpulso = 50; // limiar de tempo do impulso
```