



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO AMAZONAS
CAMPUS MANAUS DISTRITO INDUSTRIAL
TECNOLOGIA EM SISTEMAS DE
TELECOMUNICAÇÕES**



PAULO ROGÉRIO VIEIRA GALVÃO

**CONTROLE DE UM SISTEMA DE COMBATE A INCÊNDIO ATRAVÉS
DE COMUNICAÇÃO GSM E PLATAFORMA ARDUINO**

MANAUS

2019

PAULO ROGÉRIO VIEIRA GALVÃO

**CONTROLE DE UM SISTEMA DE COMBATE A INCÊNDIO ATRAVÉS
DE COMUNICAÇÃO GSM E PLATAFORMA ARDUINO**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Sistemas de Telecomunicações, do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, Campus Manaus Distrito Industrial – IFAM/CMDI.

Orientador: Prof. MSc. José Geraldo de Pontes e Souza

MANAUS

2019

Dados Internacionais de Catalogação na Publicação (CIP)

G182c Galvão, Paulo Rogério Vieira.

Controle de um sistema de combate a incêndio através de comunicação GSM e plataforma arduino. / Paulo Rogério Vieira Galvão. – Manaus, 2019. 74 f.: il.

Controle de um sistema de combate a incêndio através de comunicação GSM e plataforma arduino (Tecnologia em Sistema de Telecomunicações) – Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, *Campus* Manaus Distrito Industrial, 2019.

Orientador: Prof. MSc. José Geraldo de Pontes e Souza

1. Combate a incêndio 2. GSM. 3. Arduino. I. Souza, José Geraldo de Pontes e (Orient.) II. Instituto Federal de Educação, Ciência e Tecnologia do Amazonas. III. Título.

CDD 621.382

Elabora por Fc^a. Amélia Frota, registro n.858 (CRB11)



TERMO DE APROVAÇÃO

CONTROLE DE UM SISTEMA DE COMBATE A INCÊNDIO ATRAVÉS DE COMUNICAÇÃO GSM E PLATAFORMA ARDUINO

Por

PAULO ROGÉRIO VIEIRA GALVÃO

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 04 de dezembro de 2019 como requisito parcial para a obtenção do título de Tecnólogo em Sistemas de Telecomunicações. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

José Geraldo de Pontes e Souza
Coordenador de Eletrônica
PORTARIA 386 GDC/CI/DIRFAM 8/11/2018

Prof. MSc. José Geraldo de Pontes e Souza
Prof. Orientador

Prof. MSc. Nivaldo Rodrigues e Silva
Membro titular

Prof. MSc. Ewerton Andrey Godinho Ribeiro
Membro titular

Dedico este projeto a Deus, a minha família, amigos e a todos os meus colegas, professores, leigos e profissionais que tive o privilégio de conhecer ao longo dessa jornada e que trabalham com a nobre missão de conectar pessoas através dos diversos meios de telecomunicações.

AGRADECIMENTOS

Agradeço, em primeiro lugar a Deus, por me proporcionar essa grande oportunidade de galgar mais um degrau na vida profissional.

Ao meu pai Jeová Marciel Galvão, que hoje está em nossos corações e memória, a minha mãe Neuderi Vieira Galvão que me ensinou o valor do trabalho e do estudo.

À minha preciosa esposa Suany Santiago Galvão por todo companheirismo, compreensão e pelas constantes ausências por conta dos estudos.

Ao meu orientador Professor José Geraldo, por me acolher em um momento crucial na minha jornada de conclusão de curso.

Ao Professor Nivaldo Rodrigues, pelas palavras de incentivo, paciência e sabedoria com que me encorajou a chegar até este momento.

A todos os meus colegas, amigos e professores do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas – Campus Manaus Distrito Industrial.

Aos meus colegas de curso que colaboraram com esse trabalho: Paulo Telles, Roberto Lopes, Vasconcelos Rebouças, Jhonathas Moutinho, Thiago Epifânio, Fred Barradas, Richardson Moura, Marllon Silva.

Enfim, obrigado a todos vocês que contribuíram direta e indiretamente para a realização deste trabalho. Deus vos abençoe.

“Com a sabedoria se edifica a casa, e com a inteligência ela se estabelece; e pelo conhecimento se encherão as câmaras de todas as substâncias preciosas e agradáveis”. (Bíblia Sagrada: Pv 24.3-4)

RESUMO

GALVÃO, Paulo Rogério Vieira. **Controle de um Sistema de Combate a incêndio Através de Comunicação GSM e Plataforma Arduino**. 2019. 70 folhas. Trabalho de Conclusão de Curso (Tecnologia em Sistemas de Telecomunicações) – Instituto Federal de Educação, Ciência e Tecnologia do Amazonas – Campus Manaus Distrito Industrial. Manaus, 2019.

Este trabalho tem como finalidade o desenvolvimento de um dispositivo automático de monitoramento remoto por meio de um sistema de combate a incêndio, através da interação entre comunicação GSM e Plataforma Arduino, possibilitando o recebimento de mensagens de texto no telefone celular de um usuário, caso tenha um princípio de incêndio. O sistema também possibilita enviar comando para receber informações dos sensores e restabelecer o sistema após o incêndio controlado.

Desta forma, através do estudo realizado foi possível apresentar uma solução funcional e dinâmica que possibilitou a comunicação do usuário com o mecanismo através do sistema de telecomunicações.

Palavras-chave: Combate a incêndio. GSM. Arduino.

ABSTRACT

GALVÃO, Paulo Rogério Vieira Galvão. **Control of a Fire Fighting System Through GSM Communication and Arduino Platform.** 2019. 70 sheets. Course Completion Work (Technology in Telecommunications Systems) - Federal Institute of Education, Science and Technology of Amazonas - Campus Manaus Industrial District. Manaus, 2019.

This work aims to develop an automatic remote monitoring device through a fire fighting system, through the interaction between GSM communication and Arduino Platform, enabling the receipt of text messages on a user's mobile phone if they have a fire principle. The system also allows the user to send command to receive information from sensors and to restore the system after controlled fire.

Thus, through the study it was possible to present a functional and dynamic solution that enabled the user communication with the mechanism through the telecommunications system.

Keywords: Fire Fighting. GSM. Arduino.

LISTA DE ILUSTRAÇÕES

Figura 1 - Cartão SIM – Fonte: TechTudo.....	20
Figura 2 - Arquitetura de um microcontrolador segundo Von Neuman.	21
Figura 3 - Arduino UNO – Fonte: Filipeflop	22
Figura 4 - Ciclo do processamento do ARDUINO	23
Figura 5 - IDE Arduino, versão 1.8.9	24
Figura 6 - Tela inicial do Arduino Software.....	25
Figura 7 - Shield SIM 800L.....	26
Figura 8 - Diagrama funcional do SIM 800 – Fonte: Datasheet_SIM800L	27
Figura 9 - Consumo do módulo SIM800L em diferentes estados– Fonte: Last Minute Engineers	28
Figura 10 - Fonte De Alimentação Carregador 9v 1a Plug P4 Arduino – Fonte: Filipeflop.....	29
Figura 11 - Regulador de tensão LM2596 – Fonte: Filipeflop	29
Figura 12 - Pinagem do módulo GSM SIM800L – Fonte: Last Minute Engineers	31
Figura 13 - Sensor MQ2 – Fonte: MasterWalker.....	32
Figura 14 - Especificações do sensor de gás MQ2 - Fonte: Last Minute Engineers	32
Figura 15 - Estrutura externa do sensor de gás MQ2	33
Figura 16 - Estrutura interna do sensor de gás MQ2- Fonte: Last Minute Engineers.....	33
Figura 17 - Estrutura interna do elemento de detecção - Fonte: Last Minute Engineers	34
Figura 18 - Ajuste da sensibilidade - Fonte: Last Minute Engineers.....	35
Figura 19 - Pinagem do módulo MQ2 – Fonte: Last Minute Engineers.....	36
Figura 20 - Sensor de umidade e temperatura DHT11– Fonte: Filipeflop	36
Figura 21 - Pinagem do sensor DHT11 – Fonte: Filipeflop	37
Figura 22 - Fluxograma das etapas do Projeto	38
Figura 23 - Diagrama de blocos do sistema.....	40
Figura 24 - Esquema de ligação entre o Arduino e o sensor MQ2.....	42
Figura 25 - Linhas de código para leitura de incide de gás/fumaça	43
Figura 26 - Leitura do nível de gás no monitor serial	43
Figura 27 - Leitura do nível de gás na plotter serial	44
Figura 28 - Esquema de ligação entre o Arduino e o sensor DHT11	44
Figura 29 - Linhas de código para leitura de nível de temperatura e umidade.....	45
Figura 30 - Leitura do nível de temperatura e umidade no monitor serial	46
Figura 31 - Leitura do nível de temperatura e umidade na plotter serial	46
Figura 32 - Esquema de ligação entre o Arduino e o botão de pânico.....	47
Figura 33 - Linhas de código para a função botão de pânico.....	48
Figura 34 - Esquema de ligação entre o Arduino e o módulo GSM	49
Figura 35 - Linhas de código para a comunicação entre o módulo e o Arduino.....	51
Figura 36 - Linhas de código para o início da transmissão SMS.....	51

Figura 37 - Linhas de código para a composição da mensagem SMS	52
Figura 38 - Linhas de código para a composição da mensagem SMS bombeiros	52
Figura 39 - Leitura simultânea dos sensores no monitor serial	52
Figura 40 - Leitura simultânea dos sensores na plotter serial	53
Figura 41 - Esquema geral do projeto	54
Figura 42 - Foto real do mecanismo.....	55
Figura 43 - Tela do celular após a detecção elevada de gás	57
Figura 44 - Tela do celular após o envio do comando “BOMBEIROS”	57
Figura 45 - Tela do celular após a detecção elevada de temperatura.....	58
Figura 46 - Tela do celular após o botão de pânico ser acionado	58
Figura 47 - Tela do celular após o envio do comando “DESATIVAR”	59

LISTA DE TABELAS

Tabela 1 - Lista de comandos AT.....	20
Tabela 2 - Lista de componentes	39
Tabela 3 - Ligação dos pinos entre Arduino e MQ2	42
Tabela 4 - Ligação dos pinos entre Arduino e DHT11	45
Tabela 5 - Ligação dos pinos entre Arduino e botão de pânico.....	47
Tabela 6 - Ligação dos pinos entre Arduino e módulo GSM	49
Tabela 7 - Ligação entre o regulador de tensão e o módulo GSM	50

LISTA DE ABREVIATURAS E SIGLAS

ANATEL	Agência Nacional de Telecomunicações
USB	<i>Universal Serial Bus</i> (Barramento Serial Universal)
IoT	<i>Internet of Things</i> (Internet das Coisas)
GSM	<i>Global System for Mobile Communications</i> (Sistema Global para Comunicações Móveis)
SMS	Short Message Service (Serviço de mensagens curtas)
SMSC	Short Message Service Center (Centro de Serviço de Mensagens Curtas)
GPRS	General Packet Radio Services (Serviços Gerais de Pacote por Rádio)
CPU	Unidade Central de Processamento
RAM	<i>Random Access Memory</i> (memória de Acesso Aleatório)
ROM	<i>Read Only Memory</i> (Memória Somente para Leitura)
GLP	Gás Liquefeito de Petróleo
IDE	Ambiente de Desenvolvimento Integrado
AC	Corrente alternada
DC	Corrente contínua
VCC	Voltagem em Corrente Contínua
GND	Filtro de Densidade Neutra Graduada

SUMÁRIO

1.	INTRODUÇÃO	13
1.1.	TEMA.....	14
1.2.	OBJETIVO GERAL.....	14
1.3.	Objetivos Específicos	14
1.4.	PROBLEMA.....	14
1.5.	JUSTIFICATIVA.....	15
1.6.	ESTRUTURA DO TRABALHO	15
2.	FUNDAMENTAÇÃO TEÓRICA	17
2.2.	TECNOLOGIA GSM	17
2.3.	SMS.....	18
2.4.	CARTÃO SIM	19
2.5.	COMANDOS AT	20
2.6.	MICROCONTROLADOR.....	20
2.7.	ARDUINO	22
2.8.	SOFTWARE ARDUINO (IDE)	24
2.9.	SHIELD GSM	25
2.10.	ANTENA	27
2.11.	FORNECIMENTO DE ENERGIA PARA O MÓDULO SIM 800L	27
2.12.	FONTE DE ALIMENTAÇÃO	28
2.13.	REGULADOR DE TENSÃO LM2596	29
2.14.	PINAGEM DO MÓDULO GSM SIM 800L.....	30
2.15.	SENSOR DE GÁS E FUMAÇA MQ-2.....	31
2.16.	ESTRUDURA EXTERNA E INTERNA DO SENSOR DE GÁS MQ-2.....	33
2.17.	FUNCIONAMENTO DO SENSOR MQ-2.....	34
2.18.	CALIBRE DO MÓDULO SENSOR MQ-2	35
2.19.	PINAGEM DO MÓDULO SENSOR DE GÁS MQ-2.....	35
2.20.	SENSOR DE UMIDADE E TEMPERATURA DHT11	36
2.21.	PINAEGEM DO SENSOR DHT11	37
3.	METODOLOGIA E MATERIAS UTILIZADOS	38
3.1.	MATERIAIS UTIIZADO.....	39
3.2.	METODOLOGIA	39
4.	TESTES E ANÁLISE DE DADOS	56
4.1.	SIMULAÇÃO DETECÇÃO DE GÁS / FUMAÇA	56
4.2.	SIMULAÇÃO DETECÇÃO DE TEMPERATURA.....	57
4.3.	SIMULAÇÃO BOTÃO DE PÂNICO	58
4.4.	TESTE REESTABELECE O SISTEMA.....	59
4.5.	TEMPO DE RESPOSTA DO RECEBIMENTO E ENVIO DE CONMANDOS PARA O MECANISMO.....	59

5.	CONSIDERAÇÕES FINAIS	61
5.1.	SUGESTÕES DE MELHORIAS	61
5.2.	SUGESTÕES DE TRABALHOS FUTUROS	62
	REFERÊNCIAS.....	63
	ANEXOS	64

1 INTRODUÇÃO

A humanidade desde a antiguidade utiliza diversos meios para se comunicar, um exemplo é a fumaça. Sempre utilizaram métodos que foram aprimorando até chegar aos dias atuais. Esses fatos históricos nos mostram que sempre buscamos nos comunicar e evoluir para que a troca de informação seja rápida e sem perdas. Partindo desse ponto tecnologias foram desenvolvidas e aprimoradas para romper essa barreira, como o rádio, a televisão, a internet e a telefonia para resolver as questões que a comunicação à distância impõe entre as pessoas.

Inicialmente a telefonia ficava limitada aos aparelhos de telefonia fixa, e isto restringia a plena comunicação, pois as pessoas não poderiam se comunicar se estivessem longe de um telefone fixo, diante deste cenário surgiu aprimoramentos do sistema de telefonia e desenvolveu-se a Telefonia Móvel. Neste novo sistema as pessoas poderiam se comunicar e se locomover ao mesmo tempo, e esta conveniência encurtou a distância e abriu portas para novas tecnologias e implementações.

A expansão da telefonia móvel proporcionou a elaboração de novos serviços como, por exemplo, o envio de mensagens de texto (SMS) e isto auxiliou na quebra da barreira da distância para realizar comunicação e ainda indicou o caminho para a resolução de outros obstáculos relacionados à distância. O SMS aliado à plataforma de desenvolvimento Arduino, são ferramentas utilizadas neste trabalho como acessórios para enviar comandos liga, desliga e receber informação de equipamentos à distância.

Hoje vivemos na era IoT (Internet das Coisas), IoT é um paradigma que possui a idéia de integrar objetos ao dia a dia das pessoas, utilizando sensores com e sem fio, tecnologias de rastreamento e redes de atuadores, trazendo imersão e praticidade para a vida cotidiana e para o comportamento dos usuários. Dentro desta visão, o conceito de IoT pode permitir acesso a uma quantidade enorme de dados gerada diariamente por tais objetos e, portanto, viabiliza a criação de serviços diferenciados por parte de órgãos públicos e empresas privadas.

Neste trabalho pretende-se utilizar o conceito de integração do GSM e de IoT na construção de dispositivo para monitorar um ambiente que utilizando comunicação GSM, seja capaz de detectar princípio de incêndio. No trabalho

utilizaram-se hardwares de prateleira disponíveis no mercado nacional, para construir um protótipo de tal dispositivo, mostrando a viabilidade técnica do projeto.

1.1. TEMA

Controle de um Sistema de Combate a Incêndio Através de Comunicação GSM e Plataforma Arduino.

1.2. OBJETIVO GERAL

Desenvolver um sistema de segurança, que consiste em uma central de detecção e alarme de combate a incêndio, controlada remotamente via Smartphone, utilizando tecnologia GSM e hardwares de prateleira disponíveis no mercado nacional.

1.3. Objetivos Específicos

- Utilizar a tecnologia GSM e plataforma Arduino para controlar e receber informações dos sensores do sistema;
- Elaborar um programa em linguagem Arduino que permita a interação do usuário remotamente com o sistema via Smartphone;
- Montar um protótipo do sistema de detecção e alarme contra incêndio;
- Verificar, por meio de testes, o funcionamento do sistema.

1.4. PROBLEMA

Atualmente o mundo vive em plena era da modernidade em telecomunicações e sistemas da informação. Os sistemas de telecomunicações têm superado a barreira da simples comunicação a distância e o tempo da difusão do conhecimento entre as pessoas. Pensando nisso, por que também não utilizar a tecnologia de telecomunicações para ir além da comunicação básica entre as pessoas? Por que não utilizar a automatização de equipamentos aliada às

telecomunicações para a otimização da comunicação à distância? A partir do desenvolvimento de tecnologias móveis e do barateamento de equipamentos, o uso de sistemas de monitoramento vem se popularizando e trazendo conveniência para as pessoas com redução de alguns custos, como, implementação de sistemas robustos com equipamentos, cabeamento e flexibilidades de instalações.

Uma situação enfrentada por determinadas pessoas específicas que trabalham com centrais de GLP, é o monitoramento do vazamento de gás, nível de temperatura do ambiente e detecção de princípio de incêndio.

Com base nessas situações, conclui-se ser necessária a elaboração de um dispositivo que permita não somente o monitoramento, mas também controle aos ambientes a qual se encontra altos riscos de incêndio por gás GLP, sendo possível ainda agilizar o acionamento dos órgãos responsáveis de combate a incêndio como o corpo de bombeiros ou brigada contra incêndio.

1.5. JUSTIFICATIVA

Sistemas de combate a incêndio são mais do que prioridade, para alguns tipos de edificações existem determinações regulamentares. Afinal, um projeto preventivo eficiente pode ser fator decisivo para salvar vidas em casos de sinistros. A preocupação com este fato nos inspira a estudar e desenvolver novos equipamentos para o aprimoramento do sistema capaz de monitorar e combater o princípio de incêndio com mais eficiência e dinamismo, como garantir a segurança das pessoas assim como dos bens materiais.

1.6. ESTRUTURA DO TRABALHO

Para facilitar a compreensão do funcionamento do Controle de um sistema de combate a incêndio através de comunicação GSM e plataforma Arduino, este trabalho foi estruturado em 5 capítulos da seguinte forma:

O capítulo 1 traz uma breve introdução sobre o processo histórico da comunicação até chegar no tema proposto nessa monografia. Também foram apresentados os objetivos, o problema a ser resolvido, e a justificativa com os

principais argumentos que levaram o autor a seguir essa linha de pesquisa.

No capítulo 2, é citada a fundamentação teórica, a fim de embasar este trabalho com os principais conceitos e teorias sobre a temática.

No capítulo 3, são apresentadas a metodologia e os materiais utilizados, a fim de alcançar os objetivos propostos, estão descritos os procedimentos técnicos e operacionais de instalação e programação. Na sequência, o capítulo 4, são relatados os testes realizados com o protótipo, juntamente, com a análise dos resultados em relação ao funcionamento e tempo de resposta.

E, finalmente, no capítulo 5, são feitas as considerações finais do trabalho como um todo, avaliando-se os pontos positivos e oportunidades de melhorias. Foram expostas sugestões para trabalhos futuros a serem desenvolvidos como forma de ampliação e inovação no desenvolvimento de novas soluções para sistema de combate a incêndio.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. SISTEMA DE DETECÇÃO E ALARME DE INCÊNDIO

As duas principais razões para a instalação de um sistema de detecção e alarme de incêndio são a proteção a vida e a proteção da propriedade, muitas empresas, instituições e até mesmo residências estão procurando colocar um sistema cada vez mais eficaz para garantir a segurança de seus equipamentos e posses.

Um sistema de detecção e alarme de incêndio, é extremamente importante para garantir a segurança e prevenção contra incêndios em diversos locais, principalmente em lugares com grande circulação de pessoas, escolas, prédios, galpões, indústrias, são só alguns exemplos de onde o sistema de detecção e alarme de incêndio pode ser implantando [1].

Basicamente, um sistema de detecção e alarme de incêndio conta com um conjunto de acessórios que juntos fazem a segurança e prevenção dos incêndios. Existem diferentes tipos de sistema de detecção e alarme de incêndio. Alguns podem ser convencionais, digitais, analógicos ou endereçáveis. Os equipamentos são elaborados de acordo com o local a ser instalado, mas, todos são programados para oferecer total eficiência e qualidade [1].

O sistema de detecção e alarme de incêndio é composto de detectores de fumaça, temperatura e acionadores manuais, instalados estrategicamente no ambiente, detectando a presença irregular de fumaça ou calor no local através de dispositivo automático. Esses equipamentos funcionam integrados a uma central de detecção, que recebe essas alertas acionando os alarmes e outros acessórios do sistema de detecção e alarme de incêndio. O sistema de detecção e alarme de incêndio pode ser convencional, analógico ou programado a partir de endereços configurados, para que os sinais sejam enviados para locais específicos [1].

2.2. TECNOLOGIA GSM

Em telecomunicações uma das tecnologias muito utilizada é a comunicação móvel, sendo o GSM (Global System for Mobile Communications) uma das

tecnologias mais populares do mundo utilizada nos telefones celulares. Considerando toda a popularidade e acessibilidade da tecnologia GSM, cabem inúmeras possibilidades de utiliza-la aliada à sistemas para automatizar procedimentos executados manualmente que necessitam de comunicação à distância.

A tecnologia GSM é um padrão de comunicação para telefonia celular desenvolvido na Europa no início da década de 80, surgiu com o objetivo de padronizar o mercado celular europeu que era caracterizado por um grande número de padrões analógicos incompatíveis, levando a uma situação em que os serviços eram limitados ao próprio país [2].

A tecnologia GSM passou por três fases de evolução, inicialmente foi criada basicamente com serviços de voz, envio de mensagens curtas de texto (SMS) e transmissão de dados síncronos e assíncronos, mas à medida que a demanda por novas aplicações cresceu foram incrementados novos serviços até chegar aos conhecidos pacotes de dados que temos hoje [2].

2.3. SMS

Short Message Service (SMS) consiste na mensagem de texto enviada via componente de serviço de telefone, web, móveis ou sistemas de comunicação que utilizam padronizados protocolos de comunicação a permitir a troca de mensagens curtas de texto entre ligações de dispositivos. SMS até 6 anos atrás era a aplicação de dados mais utilizada no mundo, com 3,6 bilhões de usuários ativos, ou 78% de todos os assinantes de telefonia móvel dentro do globo terrestre, hoje esse percentual caiu devido o avanço de dados móveis e app de troca de mensagens [3].

SMS usado em aparelhos modernos se originaram a partir de radiotelegrafia em pagers que memorizavam o rádio utilizando protocolos padronizados por telefone, definidos no ano de 1985 como parte do Sistema Global. Na mesma ocasião foi definida série de normas ao uso em nível mundial. Adicionar a funcionalidade de mensagens de texto para dispositivos móveis representa ato que ganhou força na comunidade dos serviços de comunicação móvel no início de 1980 [3].

A ideia-chave para SMS era usar o sistema telefônico para o transporte de mensagens sobre os caminhos de sinalização necessários para controlar o tráfego telefônico durante os períodos de tempo em que não existia sinalização. Desse modo, os recursos não usados no sistema poderiam ser utilizados para transportar mensagens a custo mínimo. No entanto, foi necessário limitar o comprimento delas para 128 bytes (posteriormente melhorado a 140 bytes ou 160 caracteres) de modo que pudessem caber nos formatos de sinalização. O conceito permitiu ao SMS ser implantado em cada estação móvel. Este conceito foi fundamental para a implantação em cada estação móvel produzida e na rede desde os primeiros dias. Assim, grande base dos terminais e redes capazes já existia quando os usuários começaram a utilizar o SMS. Um novo elemento de rede necessária era um centro de serviço de mensagens curtas (SMSC). Melhorias foram incentivadas para potencializar a capacidade de rádio e infraestrutura de transporte a acomodar o aumento do tráfego de SMS. No ano de 2010 foram 6,1 trilhões de mensagens de texto SMS enviado ao redor do mundo. Isso se traduz em 193 mil por segundo. O setor cresceu na indústria comercial e ganhou 114.600 milhões dólares em termos globais no mesmo ano. Ainda em 2010 o preço médio global da mensagem de SMS foi de R \$ 0,11, enquanto as redes de comunicações móveis cobraram pelo menos 0,04 dólares nas ligações entre diferentes redes de telefonia ou mesmo com as outras taxas de interconexão [3].

2.4. CARTÃO SIM

O Cartão SIM, ou chip SIM, é um pequeno microcontrolador feito em plástico que pode ser instalado em telefones celulares e tem como funções básicas realizar roaming e, principalmente, autenticar o usuário em uma rede GSM. Como funções complementares um chip SIM pode ser usado para armazenar agenda de contatos, enviar ou receber e-mails e etc. com o avanço da tecnologia e a necessidade de otimização de espaço, os cartões foram diminuindo de tamanho seguindo a tendência dos novos aparelhos celulares, atualmente temos 3 tipos de cartão SIM. A figura 1 é um exemplo de um chip SIM.

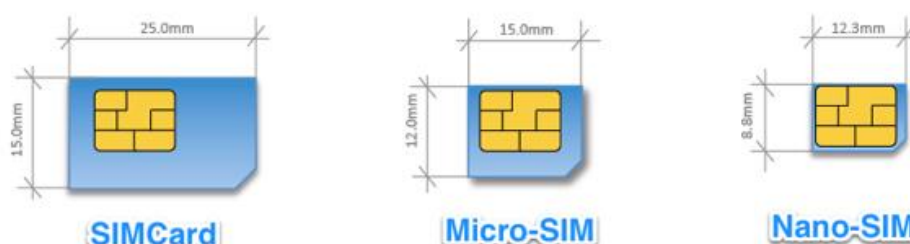


Figura 1 - Cartão SIM – Fonte: TechTudo

2.5. COMANDOS AT

São um conjunto de instruções simples utilizadas para configurar dispositivos que se conectam às redes GSM, também conhecida como Comandos Hayes, essa linguagem foi desenvolvida por Dennis Hayes em 1981 para estabelecer comunicação em modems Hayes Smartmodem. A partir da combinação de simples comandos é possível realizar instruções mais complexas, no Quadro 1 a seguir estão descritas algumas instruções.

COMANDOS	DESCRIÇÃO
AT	Verifica se há resposta do módulo
AT+CMGS	Envio de mensagem SMS
AT+CMGR	Leitura de mensagem SMS
AT+CMGDA	Apaga todas as mensagens SMS
AT+CSCS	Seleciona qual tipo de caractere será utilizado
AT+CMGF	Seleciona o formato da mensagem SMS

Tabela 1 - Lista de comandos AT

2.6. MICROCONTROLADOR

Microcontroladores são circuitos integrados compostos por processador, memória RAM, memória ROM e pinos de entrada e saída de dados, apresentam baixo consumo de energia e menor capacidade de processamento que os conhecidos microprocessadores usados em computadores. Por conta destas características são essencialmente empregados para montagem de sistemas

embarcados que estão ao nosso redor como aparelho de micro-ondas, cafeteiras, televisores, controles remotos e até automóveis.

Os microcontroladores possuem as mesmas partes que um computador precisa ter, tais partes integradas em menor escala são:

- Unidade Central de Processamento (CPU) cuja finalidade é interpretar as instruções de programa.
- Memória Programável Somente de Leitura PROM para armazenamento de instruções e Memória de Acesso Aleatório RAM para manipulação de dados.
- Sistema de clock para dar seqüência às atividades da CPU como dispositivos auxiliar ao funcionamento.
- Entradas para interiorizar na CPU informações do mundo externo recebendo impulsos de sensores, interruptores, etc.
- Saídas para exteriorizar informações processadas pela CPU para o mundo externo e controlar dispositivos externos.
- Programa (firmware) para definir um objetivo ao sistema.

A abaixo representa o diagrama de blocos com a seqüência de funcionalidade das partes que compõem um microcontrolador segundo Von Neuman.

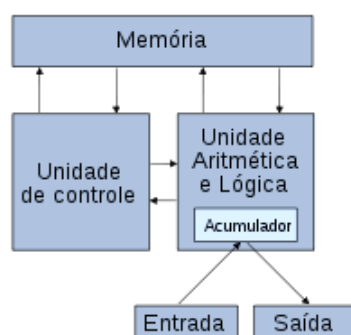


Figura 2 - Arquitetura de um microcontrolador segundo Von Neuman.

Conforme pode ser observado na Figura 2, não existe separação entre dados e programa, uma vez que há uma única área de memória. Dessa forma, o processador deve executar uma única ação por vez: ou acessa os dados ou executa uma instrução [4].

A capacidade de processamento e de armazenamento varia entre os microcontroladores definindo desta forma, famílias de processadores com funções

semelhantes. Existem famílias que são de linhas compactas, isto é, possuem poucas funções, ocupam menos espaço, consomem menos energia. Esses são usados para executar operações mais simples que não necessitem de muitos recursos. Outros são compostos por maior capacidade de armazenamento de dados, palavras de bits maiores, diversas funcionalidades e podem ser usados para a execução de atividades mais complexas, e em algumas ocasiões substituindo computadores [5].

Atualmente um grande número de microcontroladores, integrados em diversos equipamentos, exercem um papel importante no dia a dia das pessoas.

2.7. ARDUINO

O Arduino foi criado em 2005 por um grupo de 5 pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. O objetivo era elaborar um dispositivo que fosse ao mesmo tempo barato, funcional e fácil de programar, sendo dessa forma acessível a estudantes e projetistas amadores. Além disso, foi adotado o conceito de hardware livre, o que significa que qualquer um pode montar, modificar, melhorar e personalizar o Arduino, partindo do mesmo hardware básico. Assim, foi criada uma placa composta por um microcontrolador Atmel, circuitos de entrada/saída e que pode ser facilmente conectada à um computador e programada via IDE (Integrated Development Environment, ou Ambiente de Desenvolvimento Integrado) utilizando uma linguagem baseada em C/C++, sem a necessidade de equipamentos extras além de um cabo USB [6].



Figura 3 - Arduino UNO – Fonte: Filipeflop

Ele pode ser usado para desenvolver soluções eletrônicas stand-alone ou conectados ao computador. A placa Arduino não possui recursos de rede, mas pode

ser combinada com outros Arduinos criando extensões chamadas de shields. Os shields são componentes que podem facilmente ser integrados ao arduino, dando mais funcionalidades a ele, exemplos de shields: SIM800L para prover conexão GSM, sensor de gás e fumaça MQ-2 e outros que serão abordados nos tópicos seguintes deste trabalho.

A criação do Arduino teve a finalidade de ser de rápida compreensão, programação e aplicação, além de ser multiplataforma, podendo ser configurado em ambiente Windows, Linux e Mac OS. Além disso, um grande diferencial deste dispositivo é ser mantido por uma comunidade que trabalha na filosofia open-source, desenvolvendo e divulgando gratuitamente seus projetos. Esta placa de controle é uma plataforma de computação física: são sistemas digitais que realizam a interface ligada a sensores e atuadores, utiliza o microcontrolador Atmega que permitem construir dispositivos que percebam a realidade e respondem com ações físicas. Possui acessos de Entrada/Saída (I/O), sobre a qual foram desenvolvidas bibliotecas com funções que simplificam a sua programação.

De maneira mais prática, o Arduino é uma plataforma com sistema embarcado, ou seja, um sistema dotado de memória, processador e interfaces de entrada e saída que permite a atuação no ambiente por meio da combinação entre hardware e software, como ilustra a figura 4 abaixo.

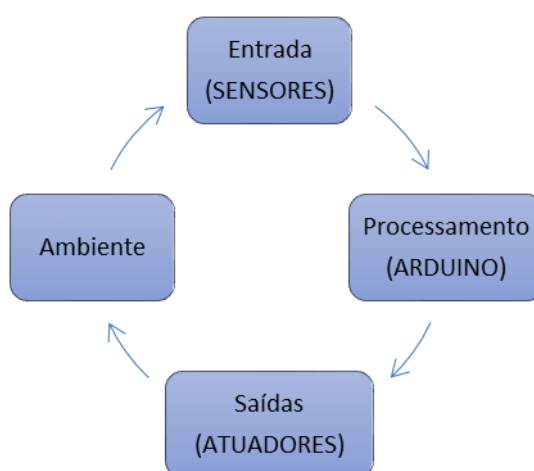


Figura 4 - Ciclo do processamento do ARDUINO

2.8. SOFTWARE ARDUINO (IDE)

Para instruir o Arduino a realizar tarefas, ou seja, construir o código que será embarcado no hardware existe o ambiente de desenvolvimento chamado Arduino Software, esta ferramenta foi desenvolvida na linguagem de programação C, pode ser instalada em vários sistemas. Abaixo é exibida a figura 5 com a tela inicial do Arduino Software:

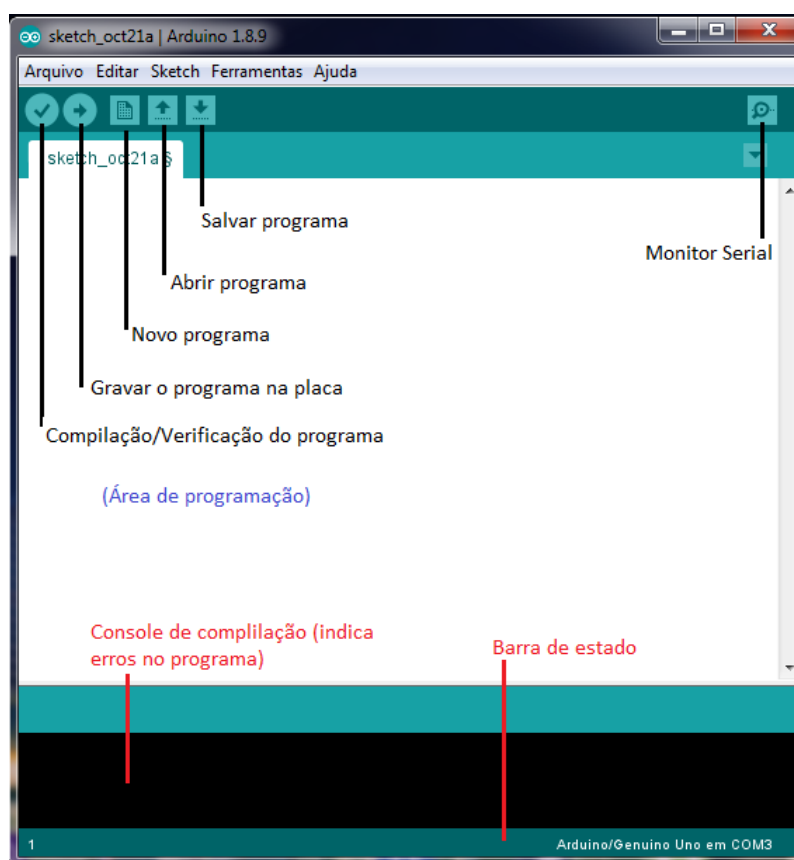


Figura 5 - IDE Arduino, versão 1.8.9

Os códigos para serem inseridos na placa do Arduino feitos neste programa são chamados de sketch e têm como estrutura básica duas funções identificadas por void setup e void loop.

A função void setup é responsável pela configuração inicial do Arduino e é executada somente uma única vez após o dispositivo ser ligado à energia elétrica.

A função void loop é responsável pelo código principal e fica em execução repetidamente até que o Arduino venha a ser desligado da fonte de energia.

Através do Arduino Software é possível também criar ou utilizar bibliotecas que simplificam o trabalho de usuários que queiram reutilizar códigos que já existem, como por exemplo, os códigos que tornam possível o uso de Shields, outra demonstração de simplificação que a IDE trouxe para o uso do Arduino é a ação de um simples apertar de um botão presente na interface do software que compila o código feito pelo usuário e o transfere para a placa Arduino.

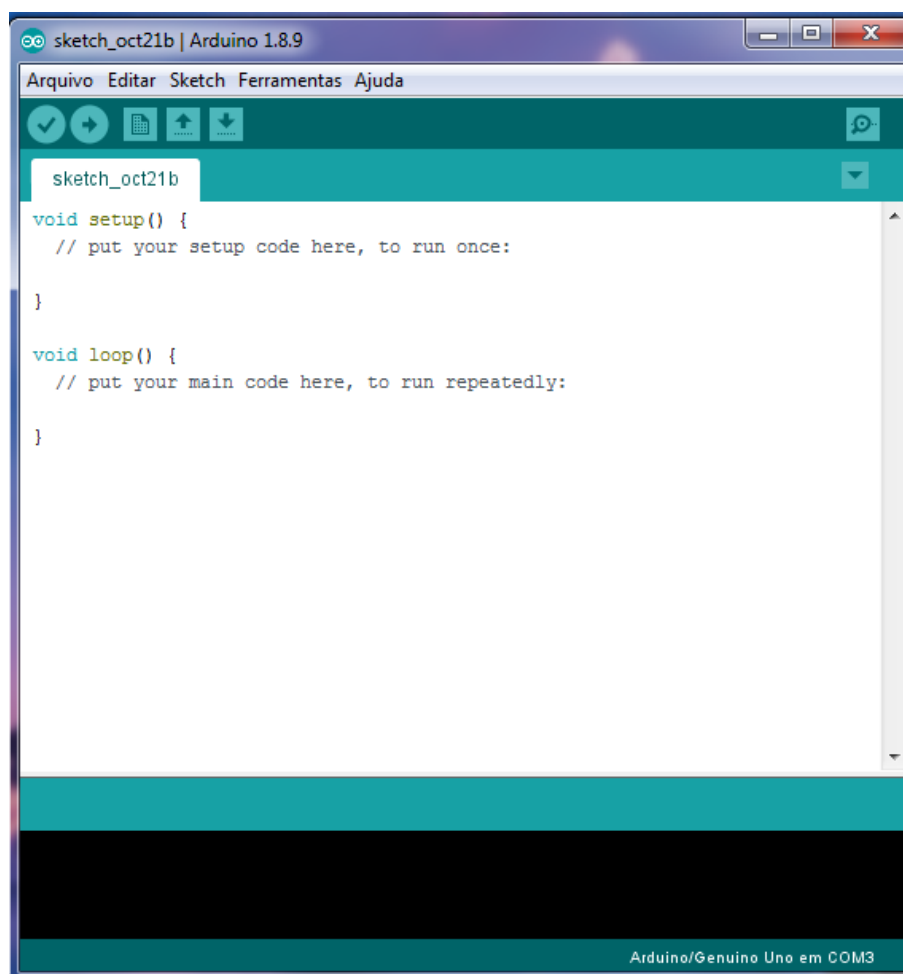


Figura 6 - Tela inicial do Arduino Software

2.9. SHIELD GSM

Segundo ARDUINO, é um dispositivo que funciona através de comandos AT e pode ser acoplado ao Arduino para capacitá-lo a realizar e receber chamadas telefônicas, enviar e receber mensagens curtas de texto (SMS) e até mesmo conectar-se à internet, tudo isso por meio da rede de telefonia GSM. Para integrar o

Shield ao Arduino é necessário ter um cartão SIM igual aos que são usados nos telefones celulares.

O Shield GSM usado neste projeto é como um telefone celular, pertence à empresa SIM Com Company of SIM Tech e é baseado no módulo SIM 800 e além dos recursos de comunicação possui luz indicadora na própria placa, e pinos de conexão tanto para microfone como para alto-falante.

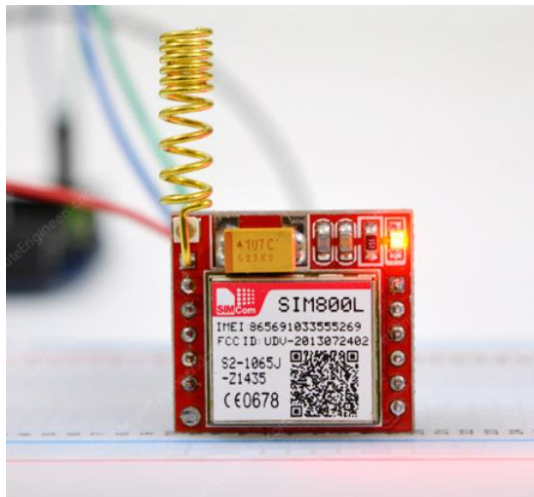


Figura 7 - Shield SIM 800L

De acordo com Company of SIM Tech, outras características deste equipamento são apresentadas abaixo:

- Frequencias: EGSM900, DCS1800, GSM850, PCS1900
- Conector para antena externa U.FL
- GPRS Data Downlink: 85.6 kbps (máximo)
- GPRS Data Uplink: 85.6 kbps (máximo)
- Suporte PAP (password authentication protocol) para conexões PPP
- Protocolo TCP/IP embutido
- Serial: 1200 bps à 115.200 bps
- Slot MicroSIM
- Acompanha antena mola
- Temperatura de operação: -40 a 85°C
- Dimensões: 21 x 15 x 3,2mm

A figura 8 a seguir representa o Shield GSM da Company of SIM Tech com o diagrama funcional do SIM 800L:

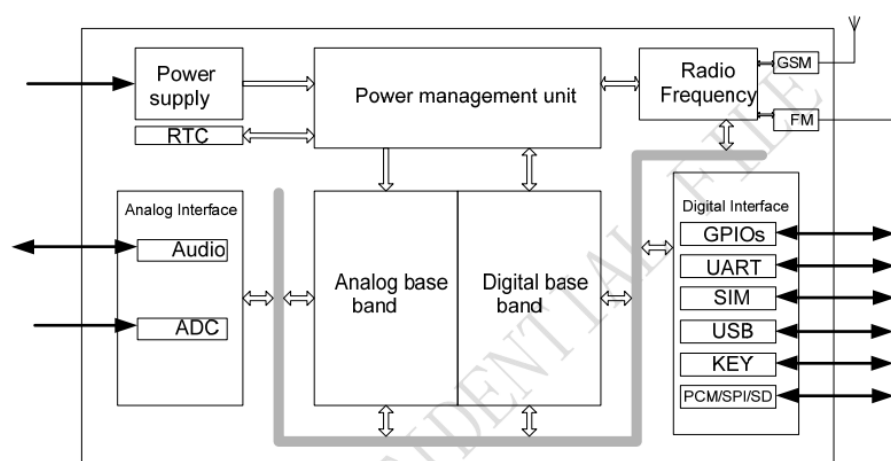


Figura 8 - Diagrama funcional do SIM 800 – Fonte: Datasheet_SIM800L

Este equipamento possui um LED indicador de estado da rede celular, localizado no lado superior direito do módulo celular SIM800L. Ele pisca em várias taxas para mostrar em que estado está:

- Pisca a cada 1s: O módulo está em execução, mas ainda não fez conexão com a rede celular.
- Pisca a cada 2s: A conexão de dados GPRS solicitada está ativa.
- Pisca a cada 3s: O módulo fez contato com a rede celular e pode enviar / receber voz e SMS.

2.10. ANTENA

É necessária uma antena para usar o módulo para qualquer tipo de comunicação de voz ou dados, além de alguns comandos do SIM, portanto, a utilização de uma antena é crucial para o bom funcionamento do sistema.

Esse projeto utiliza uma antena helicoidal GSM, que vem com o módulo e solda diretamente ao pino NET no PCB. Essa antena é muito útil para projetos que precisam economizar espaço, mas se esforçam para obter conectividade.

2.11. FORNECIMENTO DE ENERGIA PARA O MÓDULO SIM 800L

Uma das partes mais importantes do funcionamento do módulo SIM 800L é fornecê-lo com energia suficiente. Dependendo do estado em que está, o SIM 800L

pode ser um dispositivo que consome muita energia. O consumo máximo de corrente do módulo é de cerca de 2A durante a explosão da transmissão. Geralmente não exige muito, mas pode exigir cerca de 216mA durante chamadas telefônicas ou 80mA durante transmissões em rede [7]. A figura 9 da folha de dados abaixo resume a, média de consumo:

Modos	Frequência	Consumo atual
Desligar		60 uA
Modo dormir		1 mA
Espera		18 mA
Ligar	GSM850	199 mA
	EGSM900	216 mA
	DCS1800	146 mA
	PCS1900	131 mA
GPRS		453 mA
Explosão de transmissão		2 A

Figura 9 - Consumo do módulo SIM800L em diferentes estados– Fonte: Last Minute Engineers

Como o módulo SIM800L não é fornecido com o regulador de tensão integrado, é necessária uma fonte de alimentação externa ajustada à tensão entre 3.4V e 4.4V (4.2V ideal). A fonte de alimentação também deve fornecer de 1A a 2A de corrente de pico, caso contrário, o módulo continuará sendo desligado.

2.12. FONTE DE ALIMENTAÇÃO

Fonte DC Chaveada 9V com plug P4 é ideal para alimentar as placas Arduino pois se encaixa perfeitamente e tem corrente de saída igual a 1A. A fonte é bivolt, ou seja, ajusta automaticamente a energia de entrada, podendo ser utilizada em 127V ou 220V. Muito leve, possui plug no padrão universal, o que evita a necessidade de utilização de adaptadores.

Características:

- Tensão de entrada: Bivolt 100~250VAC 47~64Hz
- tensão de saída: 9VDC
- Corrente de saída máxima: 1A
- Plugue: P4



Figura 10 - Fonte De Alimentação Carregador 9v 1a Plug P4 Arduino – Fonte: Filipeflop

2.13. REGULADOR DE TENSÃO LM2596

O módulo regulador de tensão LM2596 trabalha como um conversor DC DC no modo Step Down, sendo capaz de reduzir uma carga de até 3A com ótima eficiência. A tensão de saída pode ser ajustada entre 1,5 a 35v, tendo como entrada 3,2 a 40v. Possui uma velocidade de comutação de 150KHz e pode ser aplicado em circuitos onde a saída de um sensor é superior a 5v, tensão de entrada máxima em um Arduino ou PIC.



Figura 11 - Regulador de tensão LM2596 – Fonte: Filipeflop

Especificações:

- Tensão de entrada: 3,2-40v
- Tensão de saída: 1,5-35v (ajustável, entrada deve ser 1,5v maior que a saída)
- Corrente de saída: 2A corrente nominal, 3A máxima (é recomendado adicionar dissipador de calor se potência maior que 15W)
- Eficiência de conversão: até 92% (quanto maior a tensão de saída, maior a eficiência)
- Tensão suspensa mínima: 1,5v
- Velocidade de comutação: até 150KHz
- Velocidade de resposta dinâmica: 5% 200US.
- Circuito de proteção: SS36
- Temperatura de operação: classe industrial (-40 °C a 85 °C) (potência de saída de 10W ou menos)
- Regulação de carga: $\pm 0,5\%$
- Dimensões aproximadas: 46x22mm

2.14. PINAGEM DO MÓDULO GSM SIM 800L

O módulo SIM800L possui um total de 12 pinos que fazem a interface com o mundo exterior. As conexões são as seguintes:

- **NET** é um pino no qual é possível soldar a antena helicoidal fornecida junto com o módulo.
- O **VCC** fornece energia para o módulo. Isso pode variar de 3,4V a 4,4 volts.
- **RST** (Reset) é um pino de reset rígido. Se você absolutamente colocou o módulo em um espaço ruim, puxe este pino para baixo por 100 ms para executar uma reinicialização total.
- O pino **RxD** (receptor) é usado para comunicação serial.
- O pino **TxD** (transmissor) é usado para comunicação serial.
- **GND** é o pino de aterramento e precisa estar conectado ao pino de GND no Arduino.

- O pino **RING** atua como um indicador de toque. É basicamente o pino de interrupção do módulo. Por padrão, é alto e pulsará baixo por 120 ms quando uma chamada for recebida. Também pode ser configurado para pulsar quando um SMS é recebido.
- O pino **DTR** ativa / desativa o modo de suspensão. Puxá-lo para ALTO colocará o módulo no modo de suspensão, desativando a comunicação serial. Se puxar para baixo, o módulo será ativado.
- **MIC** \pm é uma entrada diferencial de microfone. Os dois pinos do microfone podem ser conectados diretamente a esses pinos.
- **SPK** \pm é uma interface de alto-falante diferencial. Os dois pinos de um alto-falante podem ser ligados diretamente a esses dois pinos.



Figura 12 - Pinagem do módulo GSM SIM800L – Fonte: Last Minute Engineers

2.15. SENSOR DE GÁS E FUMAÇA MQ-2

O MQ2 é um dos sensores de gás comumente usados na série de sensores MQ. É um sensor de gás do tipo Metal Oxide Semicondutor (MOS), também conhecido como quimiresistores, pois a detecção é baseada na alteração da resistência do material sensor quando o gás entra em contato com o material. Usando uma rede simples de divisores de tensão, é possível detectar concentrações de gás.

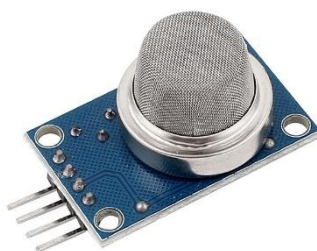


Figura 13 - Sensor MQ2 – Fonte: MasterWalker

O sensor de gás MQ2 funciona em 5V DC e consome cerca de 800mW. Ele pode detectar concentrações de GLP, fumaça, álcool, propano, hidrogênio, metano e monóxido de carbono em qualquer lugar de 200 a 10000ppm.

Ao medir gases como dióxido de carbono, oxigênio ou metano, o termo concentração é usado para descrever a quantidade de gás em volume no ar. As duas unidades de medida mais comuns são partes por milhão e porcentagem de concentração. Partes por milhão (ppm abreviado) é a proporção de um gás para outro. Por exemplo, 1.000 ppm de CO significa que, se você pudesse contar um milhão de moléculas de gás, 1.000 delas seriam de monóxido de carbono e 999.000 moléculas seriam outros gases.

Tensão operacional	5V
Resistência de carga	20 KΩ
Resistência do aquecedor	33Ω ± 5%
Consumo de aquecimento	<800mw
Resistência de detecção	10 KΩ - 60 KΩ
Escopo de concentração	200 - 10000ppm
Tempo de pré-aquecimento	Mais de 24 horas

Figura 14 - Especificações do sensor de gás MQ2 - Fonte: Last Minute Engineers

2.16. ESTRUDURA EXTERNA E INTERNA DO SENSOR DE GÁS MQ-2

Na verdade, o sensor é encerrado em duas camadas de malha fina de aço inoxidável chamada rede ante explosão. Ele garante que o elemento aquecedor dentro do sensor não cause explosão, pois estamos detectando gases inflamáveis.

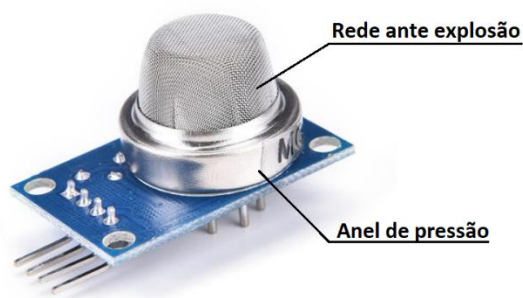


Figura 15 - Estrutura externa do sensor de gás MQ2

Ele também fornece proteção ao sensor e filtra as partículas suspensas, de modo que apenas elementos gasosos possam passar dentro da câmara. A malha é amarrada ao resto do corpo através de um anel de aperto banhado a cobre.

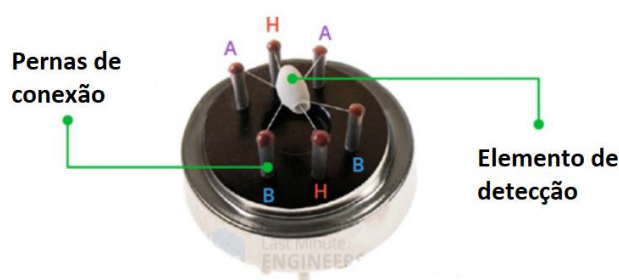


Figura 16 - Estrutura interna do sensor de gás MQ2- Fonte: Last Minute Engineers

O elemento sensor tubular é constituído por cerâmica à base de óxido de alumínio (Al_2O_3) e possui um revestimento de dióxido de estanho (SnO_2). O dióxido de estanho é o material mais importante, sendo sensível aos gases combustíveis. No entanto, o substrato cerâmico apenas aumenta a eficiência do aquecimento e garante que a área do sensor seja aquecida constantemente a uma temperatura de trabalho.

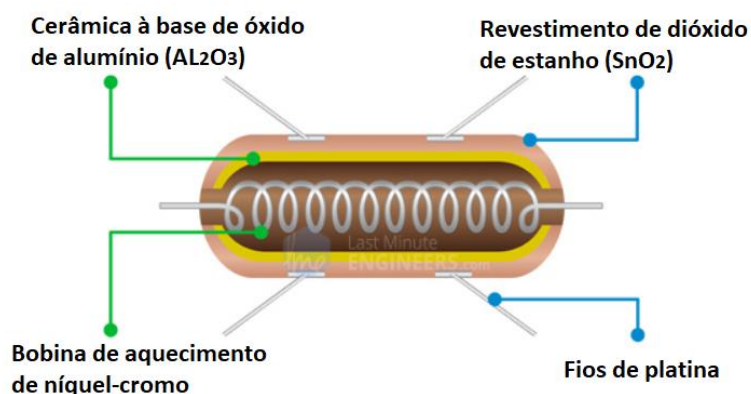


Figura 17 - Estrutura interna do elemento de detecção - Fonte: Last Minute Engineers

Assim, a bobina de níquel-cromo e a cerâmica à base de óxido de alumínio formam um sistema de aquecimento enquanto os fios de platina e o revestimento de dióxido de estanho formam um sistema de detecção.

2.17. FUNCIONAMENTO DO SENSOR MQ-2

Quando o dióxido de estanho (partículas semicondutoras) é aquecido no ar a alta temperatura, o oxigênio é adsorvido na superfície. No ar limpo, os elétrons do doador em dióxido de estanho são atraídos para o oxigênio que é adsorvido na superfície do material sensor. Isso evita o fluxo de corrente elétrica.

Na presença de gases redutores, a densidade superficial do oxigênio adsorvido diminui à medida que reage com os gases redutores. Os elétrons são então liberados no dióxido de estanho, permitindo que a corrente flua livremente através do sensor. A tensão de saída analógica fornecida pelo sensor muda proporcionalmente à concentração de fumaça / gás. Quanto maior a concentração de gás, maior é a tensão de saída; enquanto menor concentração de gás resulta em baixa tensão de saída.

O sinal analógico do sensor de gás MQ2 é posteriormente alimentado ao comparador de alta precisão LM393 (soldado na parte inferior do módulo), para digitalizar o sinal. Junto com o comparador, há um pequeno potenciômetro que é usado para ajustar a sensibilidade do sensor.

2.18. CALIBRE DO MÓDULO SENSOR MQ-2

Para calibrar o sensor de gás, você pode segurar o sensor perto de fumaça / gás que deseja detectar e continuar girando o potenciômetro até que o LED vermelho no módulo comece a brilhar. Gire o parafuso no sentido horário para aumentar a sensibilidade ou no sentido anti-horário para diminuir a sensibilidade.

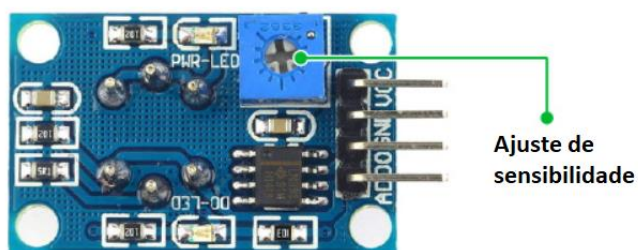


Figura 18 - Ajuste da sensibilidade - Fonte: Last Minute Engineers

O comparador no módulo verifica continuamente se o pino analógico (A0) atingiu o valor limite definido pelo potenciômetro. Quando ultrapassa o limite, o pino digital (D0) fica ALTO e o LED do sinal acende. Essa configuração é muito útil quando você precisa disparar uma ação quando determinado limite é atingido.

2.19. PINAGEM DO MÓDULO SENSOR DE GÁS MQ-2

O módulo MQ2 possui um total de 4 pinos que fazem a interface com o mundo exterior e o Arduino. As conexões são as seguintes:

- O **VCC** fornece energia para o módulo. Pode conectar à saída de 5V do seu Arduino.
- **GND** é o pino de aterramento e precisa estar conectado ao pino de GND no Arduino.
- **DO** fornece uma representação digital da presença de gases combustíveis.
- **A0** fornece tensão de saída analógica proporcional à concentração de fumaça / gás.



Figura 19 - Pinagem do módulo MQ2 – Fonte: Last Minute Engineers

2.20. SENSOR DE UMIDADE E TEMPERATURA DHT11

O sensor digital de temperatura e umidade DHT11 é um sensor composto que contém uma saída de sinal digital da temperatura e umidade. Aplicação de módulos digitais dedicados tecnologia de coleta e tecnologia de detecção de temperatura e umidade.

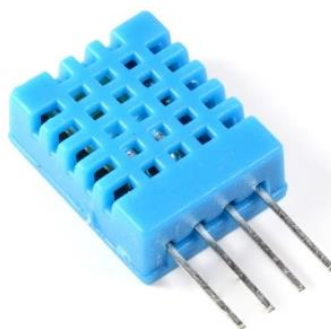


Figura 20 - Sensor de umidade e temperatura DHT11– Fonte: Filipeflop

Especificações:

- Modelo: DHT11 (Datasheet)
- Alimentação: 3,0 a 5,0 VDC (5,5 Vdc máximo)
- Corrente: 200uA a 500mA, em stand by de 100uA a 150 uA
- Faixa de medição de umidade: 20 a 90% UR
- Faixa de medição de temperatura: 0° a 50°C
- Precisão de umidade de medição: $\pm 5,0\%$ UR

- Precisão de medição de temperatura: ± 2.0 °C
- Tempo de resposta: < 5s
- Dimensões: 23mm x 12mm x 5mm (incluindo terminais)

2.21. PINAGEM DO SENSOR DHT11

O sensor DHT11 possui um total de 4 pinos que fazem a interface com o mundo exterior e o Arduino. As conexões são as seguintes:

- **VCC** fornece energia para o módulo.
- **DADOS** Liga na porta digital do Arduino.
- **N.C** não é utilizado.
- **GND** é o pino de aterramento do sensor

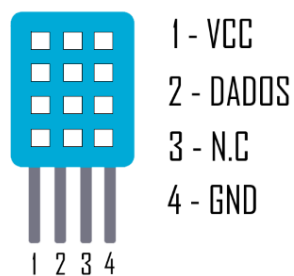


Figura 21 - Pinagem do sensor DHT11 – Fonte: Filipeflop

3. METODOLOGIA E MATERIAS UTILIZADOS

Devido às características e objetivos do projeto, podemos avaliá-lo de duas formas:

- Quanto aos fins: o objetivo do projeto é proporcionar as melhores formas e soluções para resolver um problema concreto, motivado pela necessidade de desenvolver um sistema de segurança e monitoramento para central de GLP ou qualquer outro ambiente em que possa ser aplicado;
- Quanto aos meios: a maior parte do projeto baseia-se em atividades experimentais e pesquisas bibliográficas, recorrendo a análise de aspectos como detecção, coleta de dados e comunicação, assim como o estudo de livros e outras referências a respeito dos dispositivos utilizados.

A figura 22 ilustra as etapas do processo de desenvolvimento do projeto.

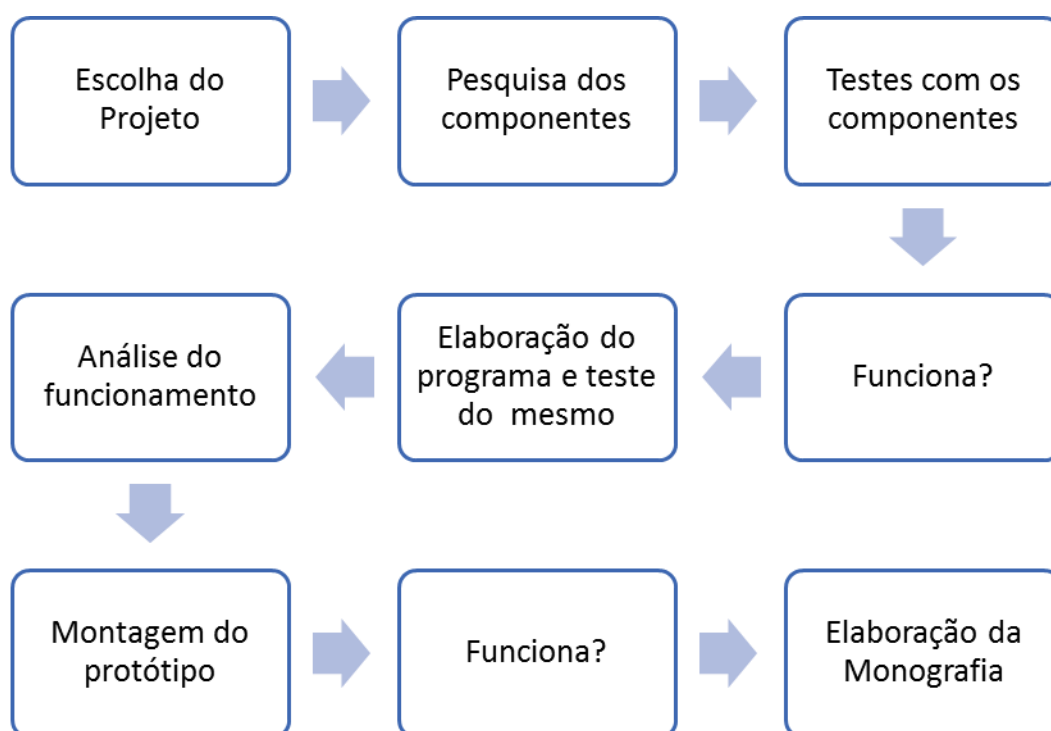


Figura 22 - Fluxograma das etapas do Projeto

3.1. MATERIAIS UTILIZADO

A montagem do mecanismo se deu a partir da utilização de componentes eletrônicos citados na tabela 2 abaixo:

COMPONENTES	QUANTIDADE
Arduino Uno	1
Shield GSM Sim 800L	1
Chip SIM	1
Sensor de gás MQ2	1
Sensor de temperatura DTH11	1
Chave Botão R13-507 2 terminais	1
Regulador de tensão LM2596	1
Fonte de 9V	1
Buzzer	1
Led vermelho	1
Resistor de 1 k Ω	2
Capacitor de 2200 μ F	1
Fios Jumper	20
Placa em acrílico para disposição dos componentes.	2

Tabela 2 - Lista de componentes

3.2. METODOLOGIA

O protótipo de monitoramento proposto neste projeto apresenta a composição mostrada na figura 23. Este sistema deve ser capaz de estimar níveis variáveis de temperatura, umidade e gás de um ambiente com a menor interferência possível sobre os valores lidos, para aplicação de um sistema remoto de detecção e alarme contra incêndio.

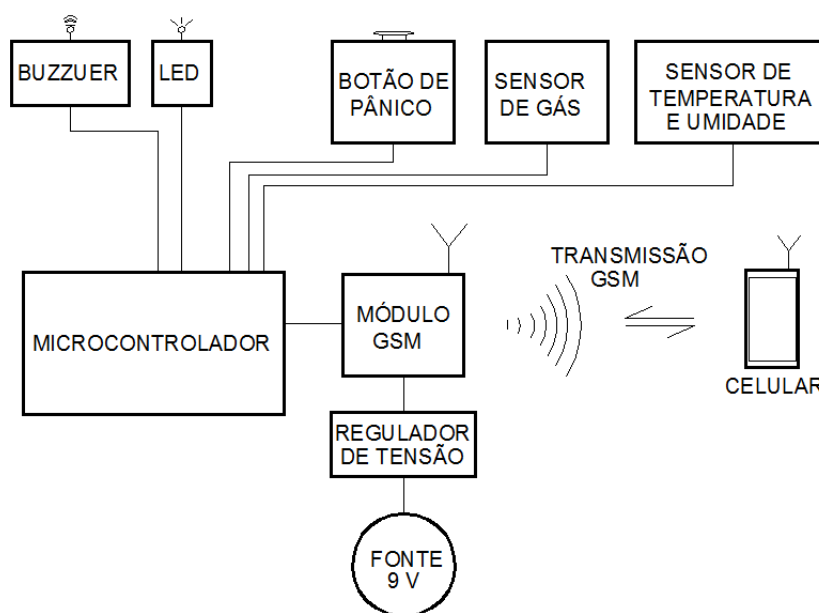


Figura 23 - Diagrama de blocos do sistema

No diagrama estão presentes os seguintes componentes:

- **Fonte chaveada:** A fonte chaveada é um dispositivo capaz de converter corrente alternada em corrente contínua fornecendo tensões elétricas com boa capacidade de corrente, sem necessitar de um transformador. Assim, a fonte chaveada é um componente compacto e muito poderoso, indicado para aparelhos que precisam de corrente contínua para funcionar. Nesse projeto adotamos uma fonte de 9V com plug P4 ideal para alimentar a placa Arduino.
- **Regulador de tensão LM2596:** O regulador trabalha como um conversor DC DC no modo Step Down, ou seja, exerce a função de reduzir e regular a tensão de entrada para o módulo GSM, pois o mesmo trabalha em uma tensão de 4,2V.
- **Módulo GSM:** O módulo SIM 800L é umas das peças principais do projeto, pois através dele faremos a comunicação e envio de dados para um celular remotamente através da tecnologia SMS, tudo por meio da rede de telefonia GSM.
- **Microcontrolador:** O microcontrolador escolhido para o projeto foi o ATmega328P produzido pela ATMEL Corporation, carregado com o Arduino Standard Bootloader e integrado a uma placa de desenvolvimento de forma a facilitar o processo de prototipagem,

através do microcontrolador implementaremos a programação e controle dos componentes de entrada e saída.

- **Sensor de temperatura e umidade:** Responsável por detectar alterações fora do padrão definidos de temperatura e umidade do ambiente, esse sensor foi escolhido pela existência de bibliotecas já programadas para Arduino, por meio do comando “#include<DHT.h>” obtém os parâmetros do meio.
- **Sensor de gás:** Este sensor é responsável por detectar a variação de gás e fumaça do ambiente, esse sensor foi escolhido pois tem a capacidade de detectar concentrações de GLP, fumaça, álcool, propano, hidrogênio, metano e monóxido de carbono em qualquer lugar de 200 a 10000ppm, ideal para a implementação do projeto.
- **Botão de pânico:** Dispositivo a ser acionado manualmente para ativar o alarme contra incêndio e desencadear o processo de disparo do sistema automático de combate a incêndio, função principal no projeto. A segunda aplicação do botão é restabelecer os sistemas em caso de alarme falso de incêndio ou após o incêndio controlado.
- **Led / Buzzer:** Tem a função de um avisador audiovisual que emite simultaneamente sinais sonoros e visuais caso uns dos detectores tenham uma variação acima dos padrões definido ou acionado o botão de pânico.
- **Celular:** O celular funciona como a central de detecção e alarme de incêndio, através dele é possível monitorar remotamente o ambiente, enviar e receber informações do sistema através do módulo GSM.

3.2.1. LEITURA DE GÁS E FUMAÇA COM SENSOR MQ-2

3.2.1.1. Diagrama

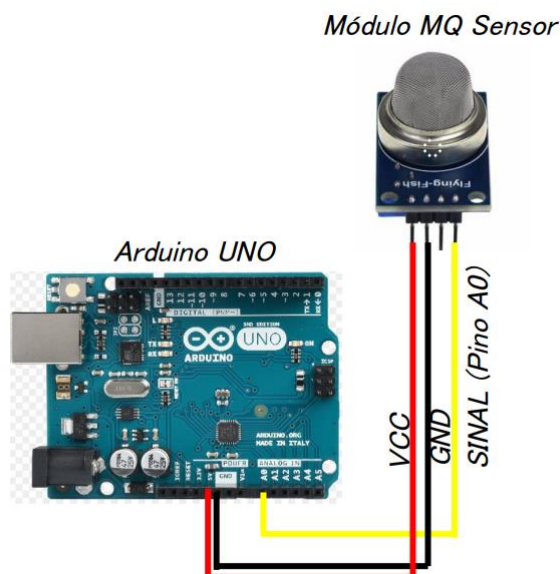


Figura 24 - Esquema de ligação entre o Arduino e o sensor MQ2

A partir do diagrama podemos observar na tabela 3 abaixo a correspondência das ligações entre portas do Arduino e os pinos do sensor, estas ligações são feitas por meio de fios jumper.

ARDUINO	MQ2
Porta analógica A0	A0 - fornece tensão de saída analógica proporcional à concentração de fumaça / gás.
5 V	VCC - fornece energia para o módulo
GND - aterramento	GND - aterramento

Tabela 3 - Ligação dos pinos entre Arduino e MQ2

3.2.1.2. Funcionamento

Para que seja realizado a detecção de níveis de gás e fumaça, a tensão de saída analógica fornecida pelo sensor muda proporcionalmente à concentração de gás / fumaça. Quanto maior a concentração de gás, maior é a tensão de saída, enquanto menor concentração de gás resulta em baixa tensão de saída.

Com o código abaixo é possível observar a saída do módulo MQ2 em tempo real através da comunicação serial presente na placa de desenvolvimento do microcontrolador. Além disso, a IDE fornecida pela plataforma Arduino permite uma visualização do sinal coletado em tempo real através de sua ferramenta Plotter Serial.

```
#include <SoftwareSerial.h>
void setup() {
  //DEFINE OS NÍVEL DE GAS
  float nivelDeGasAtual
  Serial.begin(9600);
}

void lerSensores() {
  //LÊ O SENSOR E SALVA O VALOR LIDO NA VARIÁVEL DE ESTADO
  nivelDeGasAtual = analogRead(A0);

  Serial.print(" NÍVEL DE GÁS: ");
  Serial.println(nivelDeGasAtual);
}
```

Figura 25 - Linhas de código para leitura de incide de gás/fumaça

Utilizando-se as ferramentas do monitor serial e plotter serial presentes na plataforma Arduino a seguinte saída foi observada.

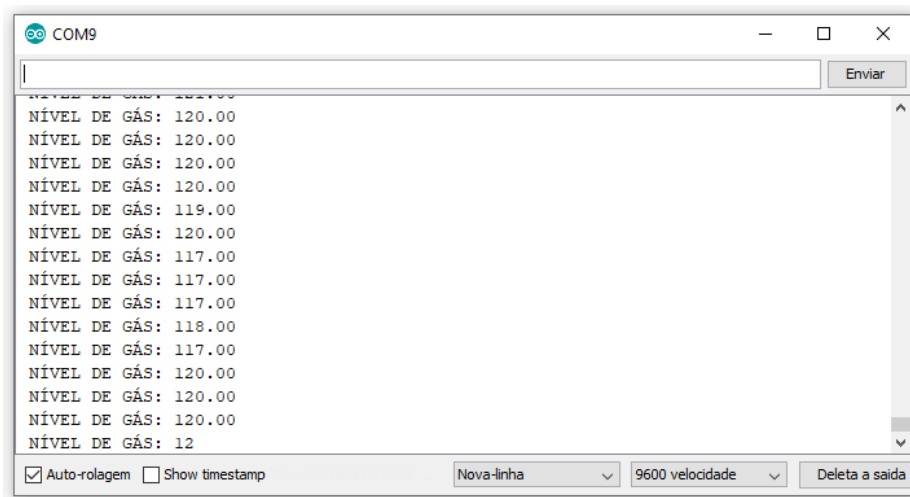


Figura 26 - Leitura do nível de gás no monitor serial

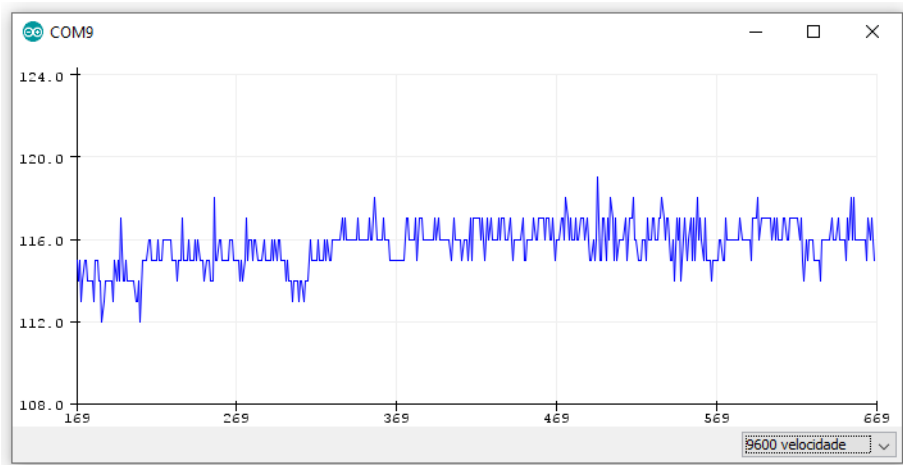


Figura 27 - Leitura do nível de gás na plotter serial

Os gráficos acima, figura 26 e 27 mostra os níveis de gás lido na saída do módulo MQ2, que ao ser interpretada pelo conversor analógico do Arduino assume valores aproximadamente entre 117 e 120 em leitura normal sem excitação de gás e fumaça, dentro de um range correspondente às tensões de 0V e 5V respectivamente.

3.2.2. LEITURA DE TEMPERATURA E UMIDADE COM O SENSOR DHT11

3.2.2.1. Diagrama

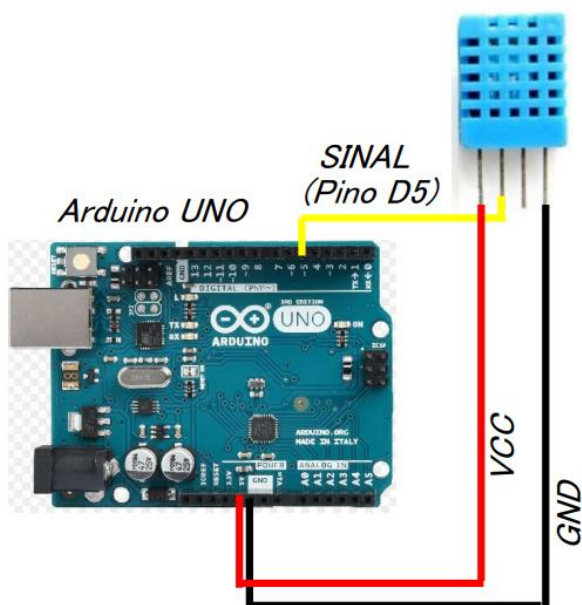


Figura 28 - Esquema de ligação entre o Arduino e o sensor DHT11

A tabela 4 abaixo contém correspondência das ligações entre portas do Arduino e pinos do sensor, estas ligações são feitas por meio de fios jumper.

ARDUINO	DHT11
Porta digital D5	DADOS - Representado pela linha amarela
5 V	VCC - fornece energia para o módulo
GND - aterramento	GND - aterramento

Tabela 4 - Ligação dos pinos entre Arduino e DHT11

3.2.2.2. Funcionamento

O sensor de temperatura e umidade do ar tem precisão de 1°C e 0,5% de umidade, esse sensor foi escolhido pela existência de biblioteca já programada para o Arduino, um simples comando “#include<DHT.h>” obtém os parâmetros do ambiente. Para isso é necessário baixar a biblioteca do sensor, com a IDE berta basta clicar em Sketch, incluir biblioteca e depois gerenciar biblioteca, terminando esse procedimento, basta procurar pela biblioteca digitando o nome do sensor DHT11.

```
#include <DHT.h>
#include <DHT_U.h>
#include <SoftwareSerial.h>

DHT dht(5,DHT11);
//DECLARAÇÃO DE FUNÇÕES
float temperaturaAtual;
float humidadeAtual;

void setup() {
  Serial.begin(9600);
  dht.begin();
  pinMode(5, INPUT);
}

void lerSensores() {
  //LÊ OS SENSORES E SALVA O VALOR LIDO NAS VARIÁVEIS
  temperaturaAtual = dht.readTemperature();
  humidadeAtual = dht.readHumidity();

  Serial.print("NÍVEL DE TEPERATURA: ");
  Serial.print(temperaturaAtual);
  Serial.print(" NÍVEL DE UMIDADE: ");
  Serial.println(humidadeAtual);
}
```

Figura 29 - Linhas de código para leitura de nível de temperatura e umidade

Abaixo temos o gráfico utilizando as ferramentas de terminal serial e plotter serial presentes na plataforma Arduino, a seguinte saída foi observada.

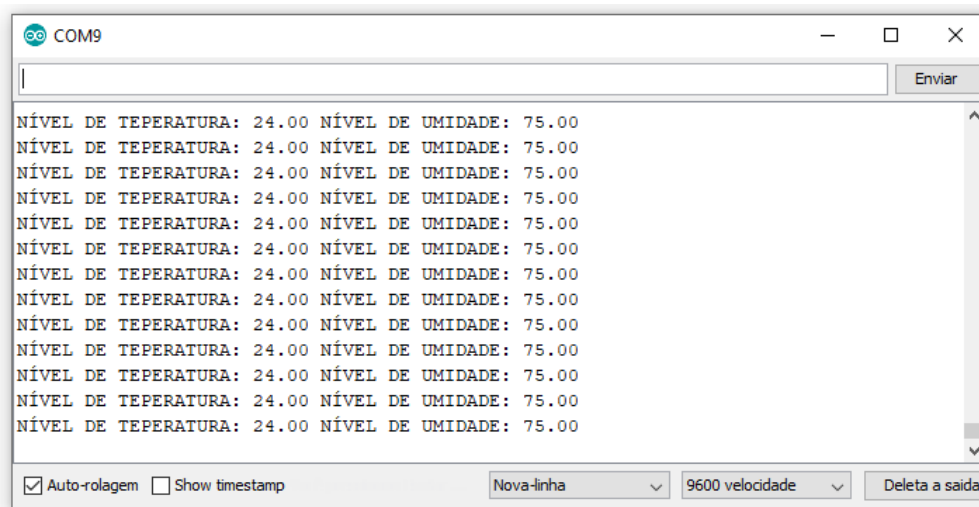


Figura 30 - Leitura do nível de temperatura e umidade no monitor serial

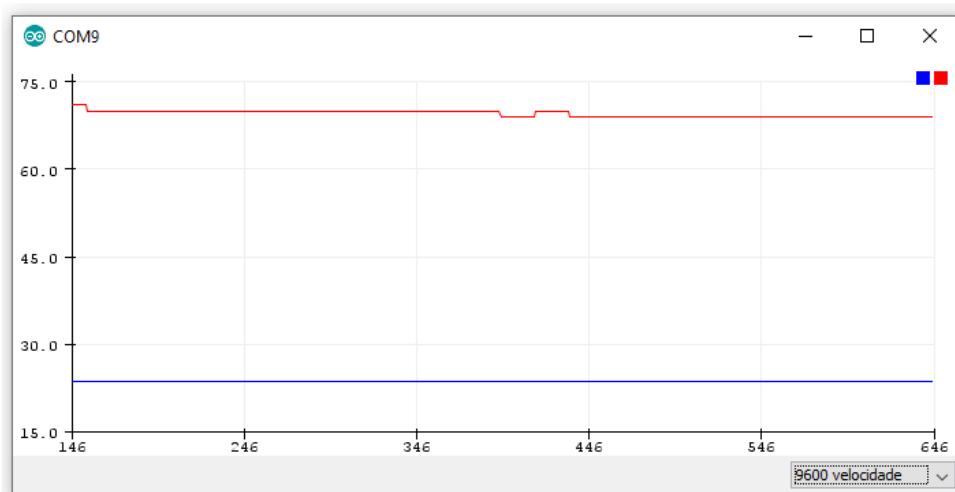


Figura 31 - Leitura do nível de temperatura e umidade na plotter serial

Nos gráficos acima notamos uma boa estabilidade da leitura de temperatura e umidade feito pelo sensor DHT11, tornando dessa forma as medidas mais assertivas e confiáveis para a finalidade de detecção.

3.2.3. INSERÇÃO DO BOTÃO DE PÂNICO

3.2.3.1. Diagrama

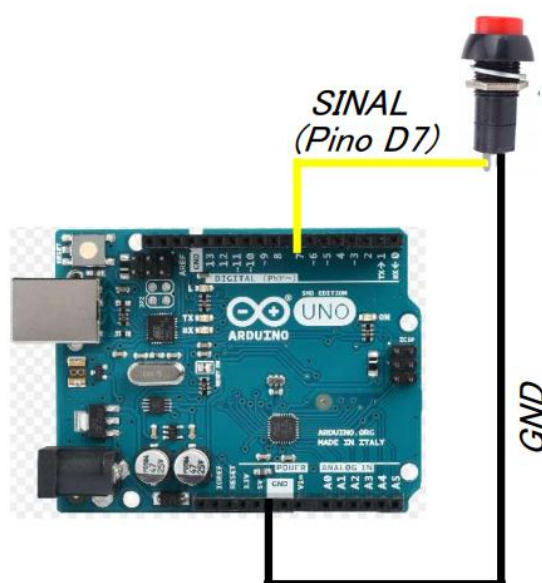


Figura 32 - Esquema de ligação entre o Arduino e o botão de pânico

A tabela 5 abaixo a correspondência de ligações entre portas do Arduino e pinos do botão, estas ligações são feitas por meio de fios jumper.

ARDUINO	BOTÃO
Porta digital D7	DADOS - Representado pela linha amarela
GND - aterramento	GND - aterramento

Tabela 5 - Ligação dos pinos entre Arduino e botão de pânico

3.2.3.2. Funcionamento

O botão de pânico nesse projeto tem duas finalidades, a primeira é simular o acionador manual de um sistema de detecção e alarme contra incêndio, a segunda é restabelecer o sistema caso tenha ocorrido um alarme falso. Para acionar o alarme basta pressionar o botão por cerca de 200 milissegundos, e para restabelecer o sistema basta pressionar o botão por 2000 milissegundos, esses parâmetros foram definidos no código, como mostra a figura 33 abaixo.

```

#include <SoftwareSerial.h
//CONSTANTES DO ESTADO DO ALARME
const int ALARM_STATUS_LENDO = 0;
const int ALARM_STATUS_ATIVADO = 9;
int alarmStatus = ALARM_STATUS_LENDO;
//CONSTANTES DOS MOTIVOS DE ACIONAMENTO DO ALARME
const String MOTIVO_SMS = "APARELHO DE CELULAR (SMS)";
const String MOTIVO_BOTAO_DE_PANICO = "BOTAO DE PANICO";
String alarmCause = "";
//VARIÁVEIS DE ESTADO E DECLARAÇÃO DE FUNÇÃO
float startAcumulador = 0;
float acumulador = 0;
int botaoDeEmergencia;
void soarAlarme();
void pararAlarme();

void setup() {
  Serial.begin(9600);
  pinMode(7, INPUT_PULLUP); //UTILIZADO MODO PULLUP PARA O PINO DO BOTÃO
}

void lerSensores() {

  Serial.print(" ESTADO BOTÃO: ");
  Serial.println(botaoDeEmergencia);
  Serial.print(" ");
  digitalWrite(0, LOW);
}

void checarEstado() {

  //CHECA SE O BOTÃO DE EMERGÊNCIA ESTÁ APERTADO E INICIA UM CRONÔMETRO
  //(TEMPO DO CRONÔMETRO É GUARDADO NA VARIÁVEL ACUMULADOR)
  if(botaoDeEmergencia!=1){
    //CONTA O TEMPO ENQUANTO O BOTÃO DE EMERGÊNCIA ESTIVER ACIONADO
    startAcumulador==0?startAcumulador=millis():acumulador=millis()-startAcumulador;
    Serial.println(int(acumulador));
  }
  else {
    //SE O BOTÃO FOR SOLTO O ACUMULADOR VOLTA A ZERO E O CRONÔMETRO É REINICIADO
    startAcumulador = 0;
    acumulador = 0;}

  //CHECA SE O TEMPO DO ACUMULADOR PASSOU DE 200ms E SE O ESTADO DO ALARME É "LENDO"
  //SE ISSO ACONTECER O ESTADO PASSA A SER "ATIVADO"
  //RESUMINDO: CHECA SE O BOTÃO DE EMERGÊNCIA FOI ACIONADO
  ((alarmStatus==ALARM_STATUS_LENDO) && (acumulador>200)) ?
  alarmCause = MOTIVO_BOTAO_DE_PANICO, delay(100), comunicarSMS(),
  alarmStatus=ALARM_STATUS_ATIVADO, startAcumulador = 0, acumulador = 0 :Serial.print("");

  //CHECA SE O TEMPO DO ACUMULADOR PASSOU DE 2000ms E SE O ESTADO DO ALARME É "ATIVADO"
  //SE ISSO ACONTECER O ESTADO PASSA A SER "LENDO"
  //RESUMINDO: CHECA SE O ALARME FOI DESATIVADO USANDO O BOTÃO DE EMERGÊNCIA
  ((alarmStatus==ALARM_STATUS_ATIVADO) && (acumulador>2000)) ?
  pararAlarme(), delay(200), alarmStatus=ALARM_STATUS_LENDO, startAcumulador = 0,
  acumulador = 0 :Serial.print("");
}

```

Figura 33 - Linhas de código para a função botão de pânico

3.2.4. COMUNICAÇÃO GSM

3.2.4.1. Diagrama

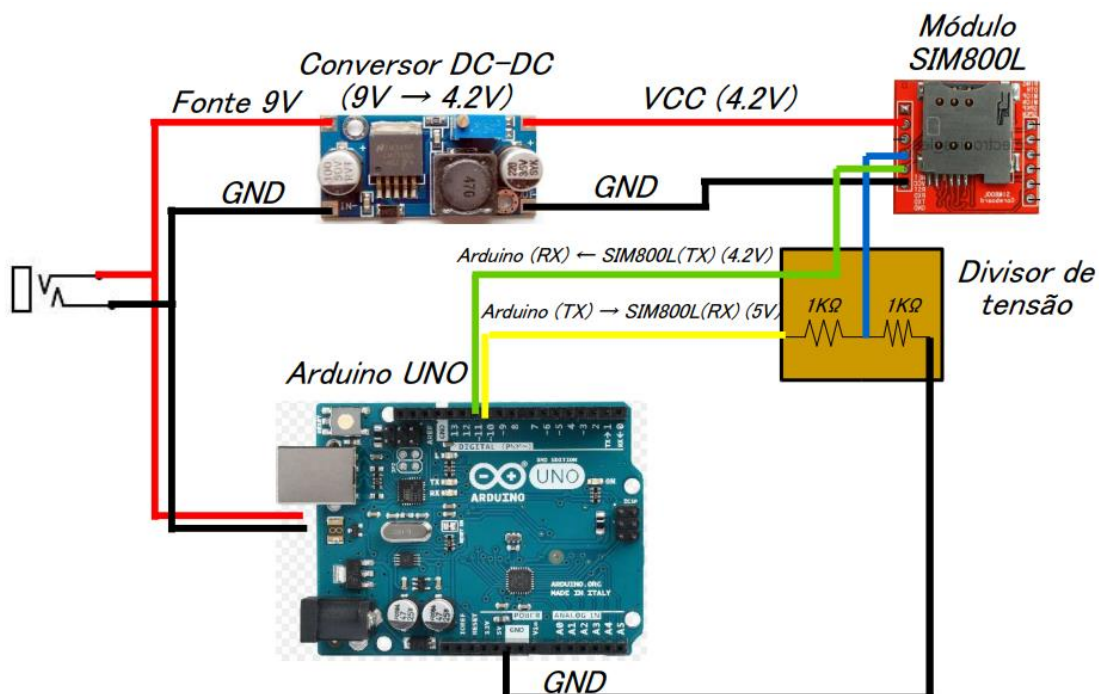


Figura 34 - Esquema de ligação entre o Arduino e o módulo GSM

A partir do diagrama podemos observar na tabela 6 abaixo a correspondência de ligações entre portas do Arduino e pinos do sensor, estas ligações são feitas por meio de fios jumper.

ARDUINO	MÓDULO GSM
Porta digital D10	RX - (receptor) é usado para comunicação serial
Porta digital D11	TX - (transmissor) é usado para comunicação serial.
GND – aterramento (fornecido pela fonte)	GND - aterramento

Tabela 6 - Ligação dos pinos entre Arduino e módulo GSM

3.2.4.2. Alimentação do módulo GSM

O Módulo GSM requer uma boa alimentação para seu funcionamento, para isso foi adotado uma fonte chaveada de 9V e corrente de saída de 1A. No entanto o módulo trabalha em um ranger de tensão entre 3.4V a 4.4V.

Para ajustar a tensão proveniente da fonte de forma a não danificar o componente, foi implementado o regulador de tensão LM2596, a tensão adotada de trabalho foi de 4.2V ideal para o seu funcionamento. Abaixo na tabela 7 temos a ligação dos pinos entre o regulador de tensão e o módulo GSM, conforme diagrama da figura 35.

REGULADOR DE TENSÃO LM2596	MÓDULO GSM
4.2 V (fornecido pela fonte)	VCC - fornece energia para o módulo
GND – aterramento (fornecido pela fonte)	GND - aterramento

Tabela 7 - Ligação entre o regulador de tensão e o módulo GSM

3.2.4.3. Divisor de tensão

Por precaução, a fim de proteger o módulo na ligação entre o pino digital 11 do Arduino e o Módulo GSM, foi implementado dois resistores de 1k que são responsáveis pela divisão da tensão proveniente do Arduino, de forma que esteja dentro dos valores de operação do módulo, neste caso a tensão de trabalho ficou em 2.5V, essa tensão não é usada para a transmissão de dados, ele é responsável pela comunicação serial entre o microcontrolador e o módulo.

Fórmula do divisor de tensão.

$$V_{out} = V_{in} * \frac{R2}{R1 + R2}$$

Cálculo do divisor de tensão do circuito.

$$V_{out} = 5V * \frac{1k}{1k + 1k}$$

$$V_{out} = 2.5v$$

3.2.4.4. Funcionamento

Para estabelecermos a comunicação entre o módulo GSM e o Arduino não foi necessário baixar biblioteca, pois a comunicação foi feita através mensagens AT serial.

Para concluir o processo de comunicação entre Arduino e Shield GSM e ter disponível o uso das funções de envio de mensagens de texto, devem-se incluir os comandos abaixo no código que está sendo feito no Arduino Software:

```
void setup() {

  sim8001.begin(9600);
  Serial.println(F("Inicializando..."));
  //INICIALIZA O MODO SMS PARA O MÓDULO SIM
  sim8001.println("AT+CMGF=1");
  //CONFIGURA O MÓDULO SIM PARA ENVIAR TODOS OS SMS RECEBIDOS PARA A PORTA SERIAL
  sim8001.println("AT+CNMI=1,2,0,0,0");
}
```

Figura 35 - Linhas de código para a comunicação entre o módulo e o Arduino

Para estabelecer a comunicação com o celular do usuário é necessário configurar o módulo SIM para transmissão de SMS, na figura 36 abaixo, estão presentes as linhas de comando para tal finalidade, no código também está presente o número a que destina a mensagem.

```
//CONFIGURA O SIM PARA MODO DE SMS
sim8001.println("AT+CMGF=1");
//CONFIGURA O MÓDULO SIM PARA ENVIAR TODOS OS SMS RECEBIDOS PARA A PORTA SERIAL
sim8001.println("AT+CNMI=1,2,0,0,0");
updateSerial();
//INICIA TRANSMISSÃO DE SMS
sim8001.println("AT+CMGS=\"92992626406\"); //INICIA TRANSMISSÃO DE SMS
updateSerial();
```

Figura 36 - Linhas de código para o início da transmissão SMS

A mensagem enviada, é composta pelo motivo no qual o alarme foi acionado e por duas opções que o usuário pode escolher para retornar como resposta para o mecanismo, “DESATIVAR”, caso queira restabelecer o sistema em caso de alarme falso, ou “BOMBEIROS” caso queira enviar uma mensagem automática para o número cadastrado do código, esse número pode ser da brigada de incêndio por exemplo, o motivo do acionamento do alarme está armazenado na variável “*alarmCause*”, na figura 38 abaixo contém o código para a composição da mensagem a ser enviada.

```
//COMPÕE A MENSAGEM SMS
sim8001.print("O alarme foi acionado pelo "); //text content
sim8001.println(alarmCause);
sim8001.println("Responda para este numero com uma das mensagens em letras maiusculas:");
sim8001.println("1: \"DESATIVAR\" para encerrar o alarme ");
sim8001.println("2: \"BOMBEIROS\" para contactar os bombeiros automaticamente");
updateSerial();
```

Figura 37 - Linhas de código para a composição da mensagem SMS

As linhas de código na figura 38 abaixo demonstra a composição da mensagem a ser enviada para a brigada de incêndio, caso o usuário retorne para o mecanismo com a palavra “BOMBEIROS”.

```
void acionarBombeiros() {

//CONFIGURA O SIM PARA MODO DE SMS
sim8001.println("AT+CMGF=1");
//CONFIGURA O MÓDULO SIM PARA ENVIAR TODOS OS SMS RECEBIDOS PARA A PORTA SERIAL
sim8001.println("AT+CNMI=1,2,0,0,0");
updateSerial();
//INICIA TRANSMISSÃO DE SMS
sim8001.println("AT+CMGS=\"92992626406\"");
updateSerial();
//COMPÕE A MENSAGEM SMS
sim8001.println("Emergência em RUA TOYAMA, N 260, PARQUE 10 DE MARCO ");
sim8001.print(alarmCause);
sim8001.println(" acionado!");
```

Figura 38 - Linhas de código para a composição da mensagem SMS bombeiros

Abaixo temos a imagem gerada pela ferramenta terminal serial e protter serial presentes na plataforma Arduino, nela temos as medias de temperatura, umidade, índice de fumaça no ambiente teste e estado do botão.

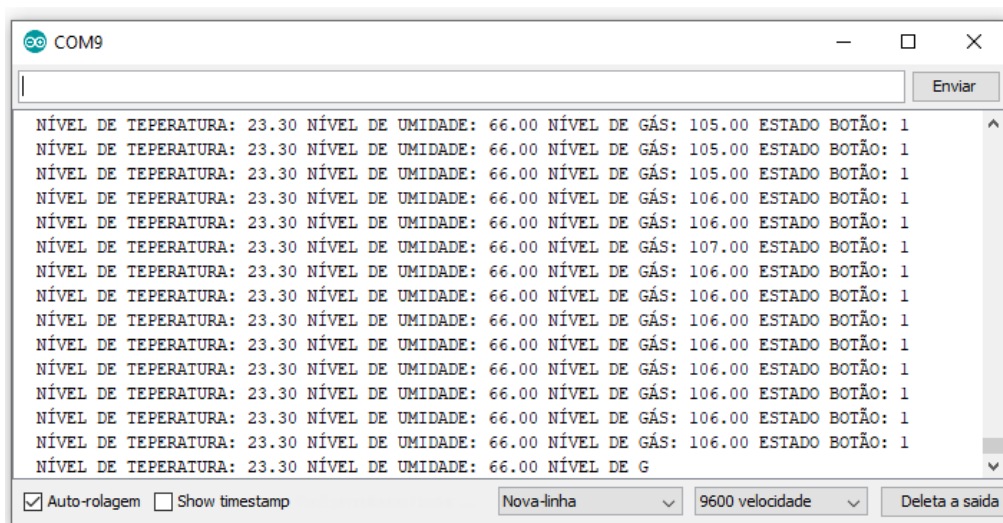


Figura 39 - Leitura simultânea dos sensores no monitor serial

Na figura 40 abaixo temos a leitura feita através da ferramenta plotter serial

do Arduino, de forma que a linha de cor alaranjada, indica o estado do botão em nível 1, não acionado, linha azul indica o nível de temperatura do ambiente, linha vermelha indica nível de umidade e pôr fim a linha verde, indica o nível de gás do ambiente.

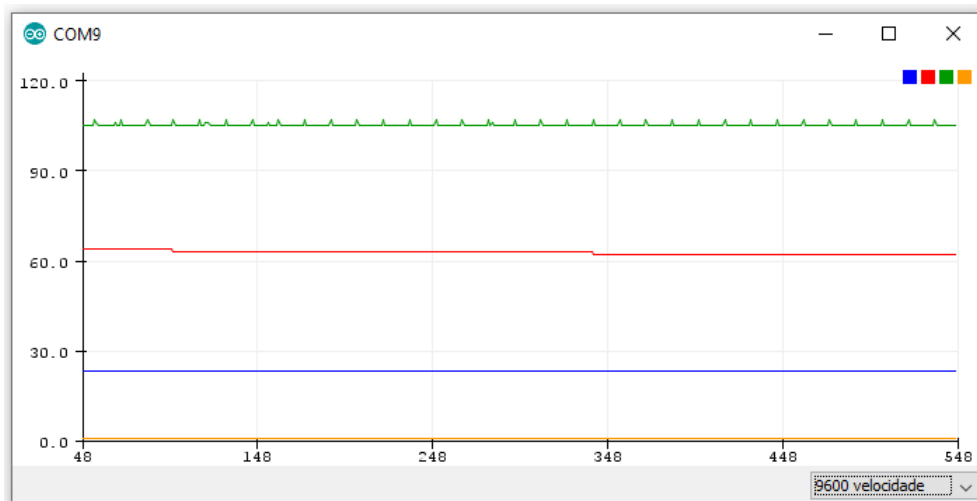


Figura 40 - Leitura simultânea dos sensores na plotter serial

3.2.5. MONTAGEM DO PROTÓTIPO

Foi desenvolvido um protótipo baseado no diagrama de bloco da figura 23. O protótipo possui entrada para comunicação USB serial com um computador e é utilizado para o carregamento do código, também é munido de um plug para ser conectado a fonte de alimentação chaveada, o mecanismo é compatível com tensões de 110Vrms a 220Vrms.

Abaixo na figura 41 temos o esquema geral do projeto:

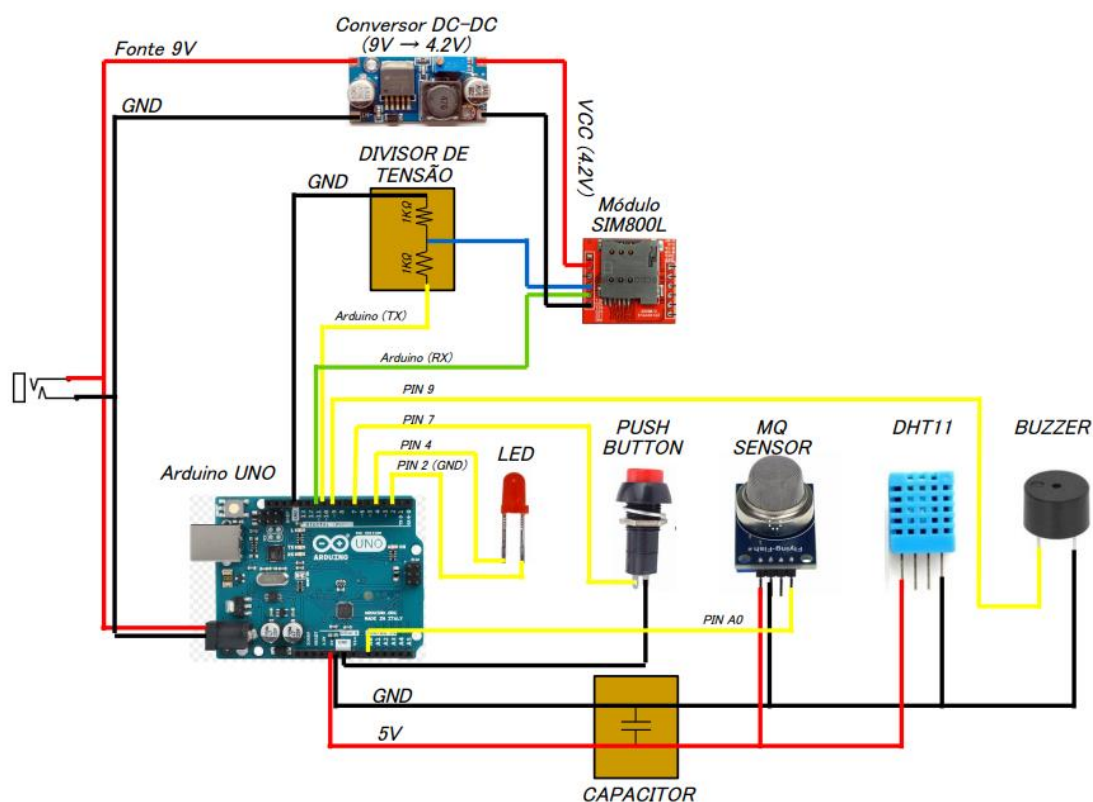


Figura 41 - Esquema geral do projeto

Utilizou-se para o encapsulamento do projeto duas placas de acrílico de 29x21x5cm na qual todos os componentes foram inseridos e fixados com poucas folgas.

Os componentes foram dispostos em 3 seções, a primeira seção foi disposto o sensor de fumaça, temperatura mais o led e o buzzer, que tem a função da sirene áudio visual, na segunda seção foi posicionado o módulo Arduino, o regulador de tensão e os demais componentes, na terceira seção estão locados o módulo GSM e o botão de pânico como mostrado na figura 42 abaixo.

Entre o sensor de gás e temperatura, foi adicionado um capacitor de 2200µF, esse capacitor tem a função de armazenar carga elétrica para ser descarregada no circuito, tornando o circuito mais estável, pois o mesmo sem o capacitor em alguns testes apresentou algumas falhas no funcionamento, a medida foi adotada de forma empírica.

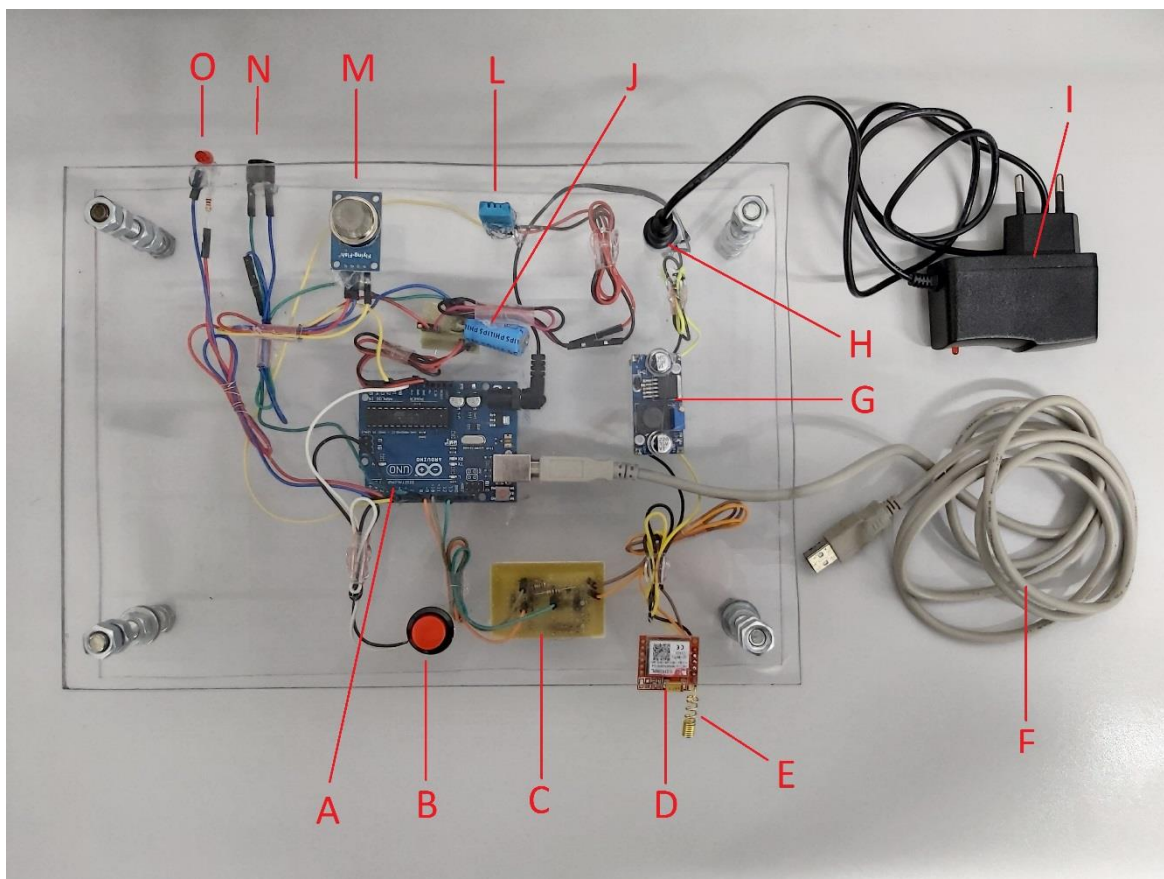


Figura 42 - Foto real do mecanismo

- A- Arduino Uno
- B- Botão de pânico
- C- Divisor de tensão
- D- Módulo GSM SIM800L
- E- Antena
- F- Cabo USB A/B
- G- Regulador de tensão LM2596
- H- Plug fêmea
- I- Fonte de 9V
- J- Capacitor de 2200µF
- L- Sensor de temperatura e umidade DHT11
- M- Sensor de Gás e Fumaça MQ2
- N- Buzzer
- O- Led vermelho

4. TESTES E ANÁLISE DE DADOS

Após concluir a montagem dos componentes, as medidas dos parâmetros de gás, temperatura e umidade, foram feitas em um ambiente de teste, no qual foram usados alguns parâmetros para a excitação dos níveis elevados de gás e temperatura.

O primeiro passo realizado foi a ligar o mecanismo à rede elétrica, para a alimentação do protótipo. Após alguns segundos o mecanismo entrou em operação. Por meio dos LED's indicadores de estado da rede celular de sinalização do módulo GSM, podemos saber quando o equipamento estará pronto para a comunicação, para isso o LED deve piscar a cada 3s.

4.1. SIMULAÇÃO DETECÇÃO DE GÁS / FUMAÇA

Os testes de detecção de gás / fumaça foram feitos por meio da liberação proposital de gás butano presente no isqueiro. Quando o índice predeterminado no código ultrapassar 450 ppm, o buzzer e o LED que são os dispositivos responsáveis pela sinalização audiovisual começaram a soar e piscar simultaneamente, automaticamente será enviado um SMS para um número cadastrado na programação, informando que o alarme foi acionado pelo sensor de gás.

Após o recebimento da mensagem o usuário terá duas opções, desativar o alarme enviando o comando "DESATIVAR" ou acionar os bombeiros enviando o comando "BOMBEIROS", caso seja enviado o comando bombeiros, o dispositivo enviará um SMS automático informando o local do princípio de incêndio.

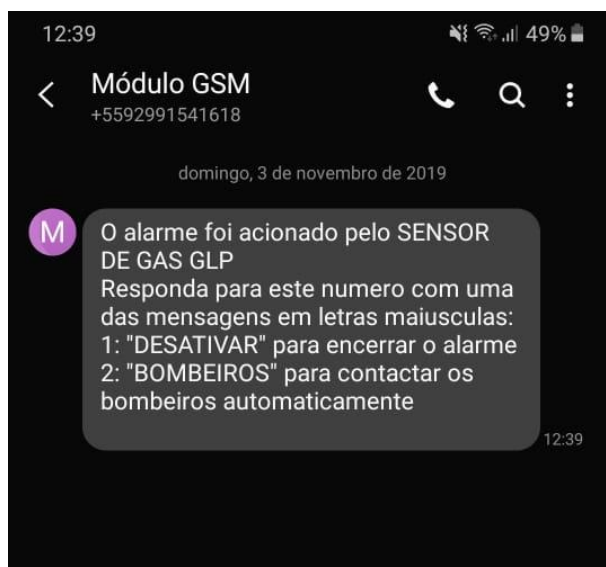


Figura 43 - Tela do celular após a detecção elevada de gás

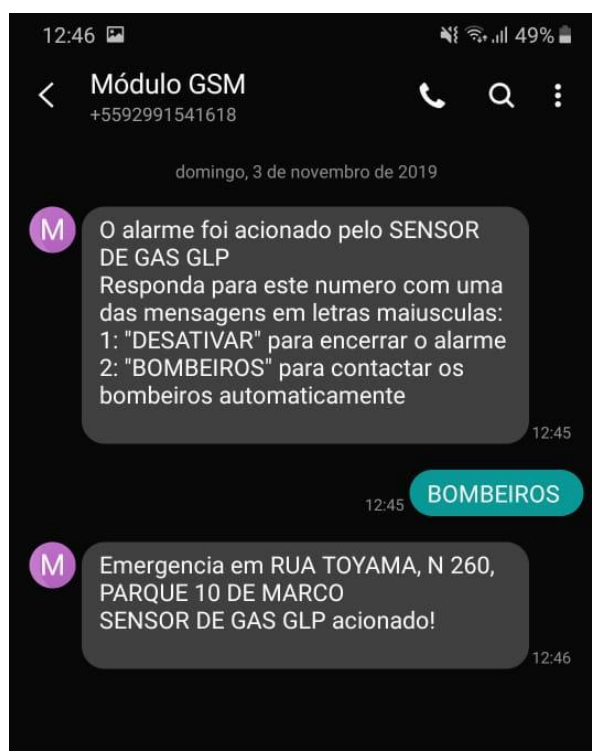


Figura 44 - Tela do celular após o envio do comando "BOMBEIROS"

4.2. SIMULAÇÃO DETECÇÃO DE TEMPERATURA

Os testes de detecção de temperatura foram feitos por meio de aplicação de ar quente proveniente de um secador de cabelo. Quando o índice predeterminado no código ultrapassar 45°C os componentes áudio visual serão acionados, a mensagem de nível elevado de temperatura será enviada para o usuário. Após o

recebimento da mensagem o procedimento será o mesmo descrito no item 4.1 acima.



Figura 45 - Tela do celular após a detecção elevada de temperatura

4.3. SIMULAÇÃO BOTÃO DE PÂNICO

Para o acionamento do sistema de alarme através do botão de pânico, basta pressionar o botão por certa de 200 milissegundos, após essa etapa o sistema audiovisual começara a atuar e o módulo GSM enviará a mensagem para o usuário. Após o recebimento da mensagem o procedimento será o mesmo descrito no item 4.1 acima.

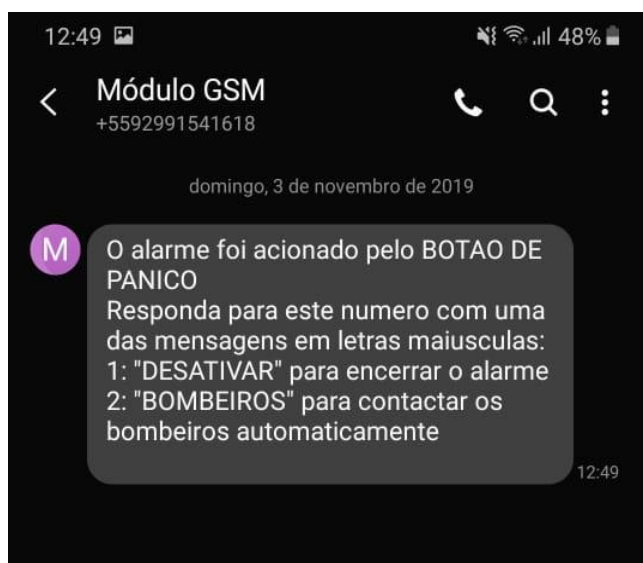


Figura 46 - Tela do celular após o botão de pânico ser acionado

4.4. TESTE REESTABELEECER O SISTEMA

Por algum o usuário queira restabelecer o sistema caso tenha ocorrido um alarme falso ou após o incêndio controlado, basta enviar o comando com a palavra “DESATIVAR”, o módulo GSM receberá a mensagem e através da comunicação serial com o Arduino o mecanismo voltará ao seu estado de monitoramento. Outra forma de restabelecer o sistema é manter pressionado o botão de pânico por certa de 2000 milissegundos, dessa forma o mecanismo entenderá que o alarme deve ser desligado.

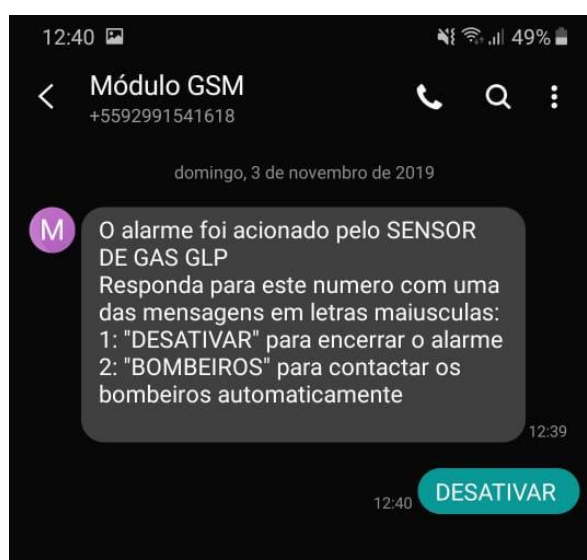


Figura 47 - Tela do celular após o envio do comando “DESATIVAR”

4.5. TEMPO DE RESPOSTA DO RECEBIMENTO E ENVIO DE CONMANDOS PARA O MECANISMO

Para o cálculo do tempo de resposta da comunicação entre o usuário e o mecanismo através das mensagens SMS, foram feitos dez testes cronometrados, tanto para o recebimento do alerta de incêndio como para o envio dos comandos, ao final foi feito uma média aritmética simples para a obtenção do resultado, como podemos observar nos cálculos abaixo.

Média aritmética para o envio do alerta de incêndio.

$$M = \frac{9s + 10s + 9s + 11s + 8s + 8s + 9s + 10s + 8s + 8s}{10}$$
$$M = 9 \text{ segundos}$$

Média aritmética para o recebimento dos comandos.

$$M = \frac{8s + 8s + 7s + 9s + 8s + 8s + 9s + 7s + 8s + 8s}{10}$$
$$M = 8 \text{ segundos}$$

O tempo de resposta obtido foi bastante satisfatório, pois como podemos observar nos cálculos acima a média entra a comunicação foi de 9 segundos o envio do alerta de incêndio e 8 segundos para o recebimento do comando, os testes foram feitos entre a operadora TIM e VIVO.

5. CONSIDERAÇÕES FINAIS

Neste trabalho de conclusão de curso, desenvolveu-se um protótipo com proposta de implementação de um mecanismo de sistema de detecção e alarme e combate a incêndio, com a utilização de comunicação GSM e Plataforma Arduino, este trabalho proporcionou buscar abordagens e informações diferenciadas que vieram agregar mais conhecimentos a aprendizagem no curso de Tecnologia em Sistemas de Telecomunicações, tais como linguagem de programação no desenvolvimento do código, eletrônica básica na montagem do circuito de interface do sensores e módulos, e pôr fim a integração ao sistema de telefonia GSM para realização da comunicação do mecanismo com o usuário via celular.

O protótipo é capaz de detectar níveis elevado de temperatura, umidade e gás de um ambiente, e transmitir via comunicação GSM informações do sistema. O protótipo criado durante o trabalho mostra que, apesar das características aparentemente desafiadoras das propostas iniciais do projeto, foi possível atingir os objetivos com soluções relativamente simples.

Os testes realizados e os resultados obtidos mostram que o projeto desenvolvido, tanto a nível de hardware quanto software pode detectar os níveis elevado predefinidos de forma satisfatória e enviar remotamente para o usuário com o objetivo de informar princípio de incêndio, a interação entre usuário e mecanismo se apresentou de forma fácil e dinâmica através do sistema de comunicação via SMS, tornando o projeto viável a aplicação de detecção.

5.1. SUGESTÕES DE MELHORIAS

Implementar no código uma lógica capaz de informar ao usuário responsável pelo monitoramento do sistema, uma função que informe caso por algum motivo o módulo GSM esteja fora de operação por falta de sinal.

O recurso de incluir telefones celulares de novos usuários com permissão para trabalhar com o mecanismo é uma melhoria que pode ser acrescentada.

5.2. SUGESTÕES DE TRABALHOS FUTUROS

Neste trabalho pode ser incluído um módulo relé para controlar solenoides que são chaves de fluxo responsáveis por cortar a alimentação de gás e dampers corta fogo, tornando assim o projeto mais robusto e completo concernentes ao sistema de detecção e alarme contra incêndio.

REFERÊNCIAS

- [1] **Sistema de detecção e alarme de incêndio**. Disponível em:
< <https://www.sipec.com.br/sistema-deteccao-e-alarme-incendio>>
Acesso em: 28 de agosto de 2019.
- [2] GAWLIK, L. C. Comunicação entre Equipamentos Utilizando Tecnologia 3G. Monografia (Especialização) – Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. 37 p. 2011. Disponível em:
<http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/418/1/CT_GESER_1_2011_16.pdf>
Acesso em: 13 junho de 2019.
- [3] **Tecnologia SMS**. Disponível em:
< <https://tecnologia.culturamix.com/dicas/o-que-e-sms-e-como-funciona>>
Acesso em: 21 de agosto de 2019.
- [4] **Microcontroladores**. Disponível em:
< <http://www.ifmg.edu.br/ceadop3/apostilas/microcontroladores>>
Acesso em: 30 de agosto de 2019.
- [5] ZILLER, R. M. **Microprocessadores**: Conceitos importantes. 2000.
- [6] **Arduino**. Disponível em:
< <https://www.filipeflop.com/blog/o-que-e-arduino/>>
Acesso em: 3 de setembro de 2019.
- [7] **Módulo GSM SIM 800L**. Disponível em:
< <https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/>>
Acesso em: 5 de setembro de 2019.

ANEXOS

1	#include <DHT.h>
2	#include <DHT_U.h>
3	#include <SoftwareSerial.h>
4	//-----
5	DHT dht(5,DHT11);
6	//DEFINE OS NÍVEIS DE TEMP, HUM E GAS
7	const int THRESHOLD_TEMPERATURA_MAXIMA = 45;
8	const int THRESHOLD_HUMIDADE_MINIMA = 20;
9	const int THRESHOLD_GAS_MAXIMO = 450;
10	//-----
11	//CONSTANTES DO ESTADO DO ALARME
12	const int ALARM_STATUS_LENDO = 0;
13	const int ALARM_STATUS_ATIVADO = 9;
14	int alarmStatus = ALARM_STATUS_LENDO;
15	//-----
16	//CONSTANTES DOS MOTIVOS DE ACIONAMENTO DO ALARME
17	const String MOTIVO_SMS = "APARELHO DE CELULAR (SMS)";
18	const String MOTIVO_SENSOR_DE_GAS = "SENSOR DE GAS GLP";
19	const String MOTIVO_SENSOR_DE_TEMPERATURA = "SENSOR DE TEMPERATURA";
20	const String MOTIVO_SENSOR_DE_HUMIDADE = "SENSOR DE HUMIDADE";
21	const String MOTIVO_BOTAO_DE_PANICO = "BOTAO DE PANICO";
22	String alarmCause = "";
23	//-----
24	char simResponse;
25	int count = 0;
26	//-----
27	//VARIÁVEIS DE ESTADO E DECLARAÇÃO DE FUNÇÕES
28	float temperaturaAtual;
29	float humidadeAtual;
30	float nivelDeGasAtual;
31	float startAcumulador = 0;

32	float acumulador = 0;
33	int botaoDeEmergencia;
34	void lerSensores();
35	void soarAlarme();
36	void pararAlarme();
37	void comunicarSMS();
38	String getSimMessage();
39	SoftwareSerial sim800l(10,11);
40	//-----
41	void setup() {
42	Serial.begin(9600);
43	dht.begin();
44	sim800l.begin(9600);
45	Serial.println(F("Inicializando..."));
46	pinMode(2,OUTPUT); // PINO DO LED
47	pinMode(4,OUTPUT); // PINO DO LED
49	pinMode(9,OUTPUT); // BUZZER
50	pinMode(5,INPUT); // GÁS
51	pinMode(7,INPUT_PULLUP); //UTILIZADO MODO PULLUP PARA O PINO DO BOTÃO
52	sim800l.println("AT+CMGF=1"); //INICIALIZA O MODO SMS PARA O MÓDULO SIM
53	sim800l.println("AT+CNMI=1,2,0,0,0"); //CONFIGURA O MÓDULO SIM PARA ENVIAR TODOS OS SMS RECEBIDOS PARA A PORTA SERIAL
54	}
55	//-----
56	void loop() {
57	//DENTRO DO LOOP É UTILIZADA UMA MÁQUINA DE ESTADOS COM 2 ESTADOS SIMPLES
58	digitalWrite(2, LOW); // DESLIGA O LED (APENAS PRECAUÇÃO)
59	switch (alarmStatus){
60	case ALARM_STATUS_LENDO:
61	lerSensores();
62	checarEstado();
63	updateSerial();
64	break;

65	case ALARM_STATUS_ATIVADO:
66	soarAlarme();
67	lerSensores();
68	checarEstado();
69	updateSerial();
70	//-----
71	// comunicarSMS();
72	//alterarEstado();
73	break;
74	default:
75	lerSensores();
76	}
77	}
78	//-----
79	void lerSensores() {
80	//LÊ TODOS OS SENSORES E SALVA O VALOR LIDO NAS VARIÁVEIS DE ESTADO
81	botaoDeEmergencia = digitalRead(7);
82	temperaturaAtual = dht.readTemperature();
83	humidadeAtual = dht.readHumidity();
84	nivelDeGasAtual = analogRead(A0);
85	//-----
86	Serial.print("NÍVEL DE TEPERATURA: ");
87	Serial.print(temperaturaAtual);
88	Serial.print(" NÍVEL DE UMIDADE: ");
89	Serial.print(humidadeAtual);
90	Serial.print(" NÍVEL DE GÁS: ");
91	Serial.print(nivelDeGasAtual);
92	Serial.print(" ESTADO BOTÃO: ");
93	Serial.println(botaoDeEmergencia);
94	//-----
95	}
96	void checarEstado() {
97	if(botaoDeEmergencia!=1){ //CHECA SE O BOTÃO DE EMERGÊNCIA ESTÁ APERTADO E

	INICIA UM CRONÔMETRO (TEMPO DO CRONÔMETRO É GUARDADO NA VARIÁVEL ACUMULADOR)
98	startAcumulador==0?startAcumulador=millis():acumulador=millis()-startAcumulador; // CONTA O TEMPO ENQUANTO O BOTÃO DE EMERGÊNCIA ESTIVER ACIONADO
99	Serial.println(int(acumulador));
100	} else {startAcumulador = 0; //SE O BOTÃO FOR SOLTO O ACUMULADOR VOLTA A ZERO E O CRONÔMETRO É REINICIADO
101	acumulador = 0;}
102	//-----
103	//CHECA SE O TEMPO DO ACUMULADOR PASSOU DE 200ms E SE O ESTADO DO ALARME É "LENDO", SE ISSO ACONTECER O ESTADO PASSA A SER "ATIVADO"
104	//RESUMINDO: CHECA SE O BOTÃO DE EMERGÊNCIA FOI ACIONADO
105	((alarmStatus==ALARM_STATUS_LENDO) && (acumulador>200)) ? alarmCause = MOTIVO_BOTAO_DE_PANICO, delay(100), comunicarSMS(), alarmStatus=ALARM_STATUS_ATIVADO, startAcumulador = 0, acumulador = 0 : Serial.print("");
106	//-----
107	//CHECA SE O TEMPO DO ACUMULADOR PASSOU DE 2000ms E SE O ESTADO DO ALARME É "ATIVADO", SE ISSO ACONTECER O ESTADO PASSA A SER "LENDO"
108	//RESUMINDO: CHECA SE O ALARME FOI DESATIVADO USANDO O BOTÃO DE EMERGÊNCIA
109	((alarmStatus==ALARM_STATUS_ATIVADO) && (acumulador>2000))? pararAlarme(), delay(200), alarmStatus=ALARM_STATUS_LENDO, startAcumulador = 0, acumulador = 0 : Serial.print("");
110	//-----
111	//CHECA SE O SENSOR DE TEMPERATURA FOI ATIVADO
112	temperaturaAtual > THRESHOLD_TEMPERATURA_MAXIMA ? alarmCause = MOTIVO_SENSOR_DE_TEMPERATURA ,delay(100),comunicarSMS(), alarmStatus=ALARM_STATUS_ATIVADO : Serial.print("");
113	//-----
114	//CHECA SE O SENSOR DE HUMIDADE FOI ATIVADO
115	humidadeAtual < THRESHOLD_HUMIDADE_MINIMA ? alarmCause = MOTIVO_SENSOR_DE_HUMIDADE ,delay(100), comunicarSMS(), alarmStatus=ALARM_STATUS_ATIVADO : Serial.print("");
116	//-----

117	//CHECA SE O SENSOR DE GAS FOI ATIVADO
118	nivelDeGasAtual > THRESHOLD_GAS_MAXIMO ? alarmCause = MOTIVO_SENSOR_DE_GAS,delay(100), comunicarSMS(), alarmStatus=ALARM_STATUS_ATIVADO : Serial.print("");
119	}
120	//-----
121	void soarAlarme() {
122	float seno;
123	int frequencia;
124	for(int x=0;x<180;x++){
125	//Converte graus para radiando e depois obtém o valor do seno
126	seno=(sin(x*3.1416/180));
127	seno>0.8?digitalWrite(4,HIGH):digitalWrite(4,LOW);
128	//Gera uma frequência a partir do valor do seno
129	frequencia = 2000+(int(seno*1000));
130	tone(9,frequencia);
131	delay(1);
132	}
133	}
134	//-----
135	void comunicarSMS() {
136	sim800l.println("AT+CMGF=1");//CONFIGURA O SIM PARA MODO DE SMS
137	sim800l.println("AT+CNMI=1,2,0,0,0");//CONFIGURA O MÓDULO SIM PARA ENVIAR TODOS OS SMS RECEBIDOS PARA A PORTA SERIAL
138	updateSerial();
139	sim800l.println("AT+CMGS=\"92981016480\");//INICIA TRANSMISSÃO DE SMS
140	updateSerial();
141	//-----
142	//COMPÕE A MENSAGEM SMS
143	sim800l.print("O alarme foi acionado pelo ");
144	sim800l.println(alarmCause);
145	sim800l.println("Responda para este numero com uma das mensagens em letras maiusculas:");

	sim800l.println("1: \"DESATIVAR\" para encerrar o alarme ");
146	sim800l.println("2: \"BOMBEIROS\" para contactar os bombeiros automaticamente");
147	updateSerial();
148	//-----
149	//FINALIZA O ENVIO DE SMS
150	sim800l.write(26);
151	sim800l.println("AT+CFUN=0");
152	delay(200);
153	sim800l.println("AT+CFUN=1");
154	sim800l.println("AT+CMGF=0");
155	updateSerial();
156	}
157	void acionarBombeiros() {
158	sim800l.println("AT+CMGF=1"); //CONFIGURA O SIM PARA MODO DE SMS
159	sim800l.println("AT+CNMI=1,2,0,0,0"); //CONFIGURA O MÓDULO SIM PARA ENVIAR TODOS OS SMS RECEBIDOS PARA A PORTA SERIAL
160	updateSerial();
161	sim800l.println("AT+CMGS=\"92981016480\"); //INICIA TRANSMISSÃO DE SMS
162	updateSerial();
163	//-----
164	//COMPÕE A MENSAGEM SMS
165	sim800l.println("Emergencia em RUA TOYAMA, N 260, PARQUE 10 DE MARCO ");
166	sim800l.print(alarmCause);
167	sim800l.println(" acionado!");
168	//-----
169	//FINALIZA O ENVIO DE SMS
170	updateSerial();
171	sim800l.write(26);
172	updateSerial();
173	}
174	void updateSerial() {
175	String stringReceived = "";
176	delay(100);

177	//TRANSMITE PARA O SERIAL DO MÓDULO SIM (SMS) TODAS AS MENSAGENS RECEBIDAS PELO ARDUINO (CABO USB)
178	while (Serial.available())
179	{
180	sim800l.write(Serial.read());//Forward what Serial received to Software Serial Port
181	}
182	//TRANSMITE PARA O SERIAL DO ARDUINO (CABO USB) TODAS AS MENSAGENS RECEBIDAS PELO MÓDULO SIM (SMS)
183	while(sim800l.available())
184	{
185	//SALVA TODAS AS MENSAGENS RECEBIDAS POR SMS NA STRING "stringReceived"
186	stringReceived = sim800l.readString();
187	Serial.println(stringReceived);//Forward what Software Serial received to Serial Port
188	}
189	//CHECA SE A "stringReceived" POSSUI AS PALAVRAS CHAVE E NÃO!!!! POSSUI A PALAVRA "para" (ISSO CORRIGE O BUG DA LETRA MAIÚSCULA)
190	if((stringReceived.indexOf("BOMBE") > 0) && !(stringReceived.indexOf("para") > 0)){
191	acionarBombeiros();
192	Serial.println("AJUDA A CAMINHO");
193	}
194	if(stringReceived.indexOf("ATIVAR ALARME") > 0){
195	alarmCause = MOTIVO_SMS;
196	delay(100);
197	comunicarSMS();
198	alarmStatus = ALARM_STATUS_ATIVADO;
199	}
200	if((stringReceived.indexOf("DESAT") > 0) && !(stringReceived.indexOf("para") > 0)){
201	pararAlarme();
202	alarmStatus = ALARM_STATUS_LENDO;
203	}
204	stringReceived = "";
205	}
206	void pararAlarme() {

207	tone(9,2300);
208	delay(400);
209	noTone(9);
210	delay(200);
211	tone(9,2600);
212	delay(300);
213	noTone(9);
214	}