



UNIVERSIDADE PRESBITERIANAMACKENZIE

– Faculdade de Computação e Informática –

## **Projeto: Desenvolvimento de sistema de gestão inteligente de estoque Arduino para indústria automotivo**

**Nome: Angelita de M Hora, Nome: Angelita Oliveira dos Santos, Nome: Any Kelly dos  
Santos Nunes**

Faculdade de Computação e Informática  
Universidade Presbiteriana Mackenzie (UPM) – São Paulo, SP – Brasil  
[10412411@mackenzista.com.br](mailto:10412411@mackenzista.com.br), [10369216@mackenzista.com.br](mailto:10369216@mackenzista.com.br),  
[10369330@mackenzista.com.br](mailto:10369330@mackenzista.com.br)

**Abstract.** *The development of a smoke detector with Arduino is an example of how technology can be used to solve real problems and contribute to a safer world. By combining the versatility of Arduino with the importance of early fire detection, this project represents a significant advance in the field of robotics and technology.*

**Resumo:** *O desenvolvimento de um detector de fumaça com Arduino é um exemplo de como a tecnologia pode ser utilizada para resolver problemas reais e contribuir para um mundo mais seguro. Ao combinar a versatilidade do Arduino com a importância da detecção precoce de incêndios, este projeto representa um avanço significativo na área da robótica e da tecnologia.*

### **Introdução.**

A crescente preocupação com a segurança em ambientes de estocagem automotiva impulsiona a busca por soluções inovadoras e eficientes para a detecção precoce de incêndios. Nestes locais, a presença de materiais inflamáveis como plásticos, borracha e fluidos, juntamente com a grande quantidade de veículos, torna a prevenção de incêndios uma prioridade absoluta.

## 1. Objetivo

O objetivo principal deste projeto é desenvolver um sistema de detecção de incêndio utilizando um Arduino como plataforma principal. Este sistema terá a capacidade de detectar a presença de fogo através de um sensor específico, acionar um alarme e notificar um usuário remoto via internet, utilizando o protocolo MQTT. Realizando a detecção precoce de incêndios, permitindo uma resposta rápida e eficaz, minimizando danos e salvando vidas.

## 2. Estrutura do Documento

- **Resumo:** Apresentação concisa do projeto, destacando os principais objetivos e resultados esperados.
- **Introdução:** Contextualização do problema da detecção de incêndios, justificativa da escolha do Arduino e do protocolo MQTT, e relevância do projeto.
- **Materiais e Métodos:** Descrição detalhada dos componentes eletrônicos, software utilizado, metodologia de montagem e programação.
- **Resultados:** Apresentação dos resultados obtidos durante os testes do sistema.
- **Discussão:** Análise dos resultados, limitações do projeto e sugestões para trabalhos futuros.

## 3. Materiais e Métodos Hardware

- **Arduino Uno:** Placa microcontroladora de código aberto utilizada como cérebro do sistema.
- **Sensor de Chama KY-026:** Detecta a radiação infravermelha emitida pelo fogo, fornecendo um sinal digital.

**Relé:** Aciona dispositivos elétricos, como sirenes ou bombas de água, em caso de detecção de fogo.

- **Buzzer:** Emite um sinal sonoro para alertar sobre a presença de fogo.
- **Módulo Wi-Fi ESP8266:** Permite a conexão do Arduino à internet para comunicação MQTT.
- **Resistores:** Utilizados para limitar a corrente elétrica em circuitos.
- **Fonte de alimentação:** Fornece energia para o circuito.
- **Placa de prototipação:** Facilita a montagem do circuito.
- **Cabos jumper:** Conectam os componentes eletrônicos.
- **Arduino IDE:** Ambiente de desenvolvimento integrado para programar o Arduino.
- **Biblioteca MQTT para Arduino:** Permite a comunicação com um servidor MQTT.
- **Servidor MQTT:** Software responsável por gerenciar a comunicação entre dispositivos MQTT (ex: Mosquitto, HiveMQ).

## 4. Metodologia

1. **Montagem do circuito:** Conectar os componentes de acordo com o diagrama esquemático.
2. **Programação do Arduino:**
3. **Ler o sinal do sensor de chama.**
4. **Acionar o relé e o buzzer em caso de detecção de fogo.**
5. **Publicar uma mensagem no servidor MQTT com informações sobre o incêndio (tempo, local).**
6. **Configuração do servidor MQTT:** Instalar e configurar o servidor MQTT.
7. **Teste do sistema:** Simular um incêndio e verificar o funcionamento do sistema.

## 5. Plataforma de Prototipagem

A plataforma de prototipagem escolhida para este projeto é o Arduino Uno. O Arduino Uno é uma placa microcontroladora de fácil utilização, com uma grande comunidade de desenvolvedores e uma vasta quantidade de recursos disponíveis.

### Comunicação MQTT

O protocolo MQTT será utilizado para enviar mensagens do Arduino para um servidor remoto. As mensagens conterão informações sobre a detecção de fogo, como o horário e o local do evento. Um aplicativo móvel ou web pode ser desenvolvido para receber essas mensagens e notificar o usuário.

### Benefícios do MQTT:

- Leve e eficiente: Ideal para dispositivos com recursos limitados, como o Arduino.
- Publicação e assinatura: Permite que dispositivos publiquem mensagens em tópicos específicos e se inscrevam para receber mensagens de outros tópicos.
- Qualidade de serviço: Oferece diferentes níveis de qualidade de serviço para garantir a entrega das mensagens.

### Próximos Passos

- Desenvolvimento do diagrama esquemático: Visualizar a conexão entre os componentes.
- Criação do código Arduino: Escrever o código para controlar os componentes e comunicar com o servidor MQTT.
- Configuração do servidor MQTT: Instalar e configurar um servidor MQTT em um computador ou na nuvem.
- Teste do sistema completo: Verificar o funcionamento de todo o sistema em um ambiente controlado.

### Possíveis extensões para o projeto:

- Integração com sistemas de segurança: Conectar o sistema a câmeras de segurança ou sistemas de alarme existentes.
- Visualização de dados: Criar um painel para visualizar os dados coletados pelo sensor e o histórico de eventos.
- Notificações personalizadas: Enviar notificações para diferentes usuários com base em suas preferências.

### Funcionamento do Protótipo Sensor de Chama:

- Princípio de Funcionamento: Detecta a radiação infravermelha emitida pelas chamas.
- Acionamento: Quando a intensidade do calor atinge altas temperaturas, o sensor gera um sinal digital alto.
- Tipos: Utilizamos o Sensor digital, são mais simples de utilizar fornecendo uma saída lógica (0 ou 1).
- Atuador: Lâmpada pisca ou acende para indicar a presença de fogo.
- Função: Gerar alerta quando o sensor de chama detecta um incêndio.

### Funcionamento do Sistema:

- Sensor de Chama: Monitora continuamente o ambiente.
- Detecção de Chama: Quando uma chama é detectada, o sensor envia um sinal digital alto para o Arduino.
- Arduino: Recebe o sinal e aciona o atuador (lâmpada).

## 6. Código

```

#include <ESP8266WiFi.h> #include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>

#define sensorIncendio D1
#define buzzerPin D7 // Definir o pino do buzzer

const char* redeWiFi = "LIVE TIM_OCA4_2G";const char* senhaWiFi = "2fmjpgtbxn";
#define SERVIDOR_AIO "io.adafruit.com" #define PORTA_AIO 1883
#define USUARIO_AIO "angelita_87"
#define CHAVE_AIO "aio_fFNh2533k5lK3xNMulMgsuOnXHL2"

WiFiClient clienteWiFi;
Adafruit_MQTT_Client clienteMQTT(&clienteWiFi, SERVIDOR_AIO, PORTA_AIO, USUA-
RIO_AIO, CHAVE_AIO);
Adafruit_MQTT_Publish publicacaoStatusIncendio = Adafruit_MQTT_Publish(&clienteMQTT,
USUARIO_AIO "/feeds/alerta");

unsigned long tempoUltimaPublicacao = 0; unsigned long tempoUltimoAlarme = 0;

const unsigned long intervaloPublicacaoNormal = 5000;const unsigned long
intervaloPublicacaoAlarme = 2000;

void conectarMQTT() {int8_t estado;

while ((estado = clienteMQTT.connect()) != 0) { clienteMQTT.disconnect();
delay(5000);
}
}

void verificarSensor() {
int statusIncendio = digitalRead(sensorIncendio);

if (statusIncendio == LOW) { // Incêndio detectadodigitalWrite(buzzerPin,
HIGH); // Ligar o buzzer
if (millis() - tempoUltimoAlarme >= intervaloPublicacaoAlarme) { Serial.println("Incêndio detec-
tado!"); publicacaoStatusIncendio.publish("Incêndio detectado!"); tempoUltimoAlarme = millis();
}
} else { // Nenhum incêndio digitalWrite(buzzerPin,
LOW); // Desligar o buzzer
if (millis() - tempoUltimaPublicacao >= intervaloPublicacaoNormal) { Serial.println("Nenhum
incêndio identificado."); publicacaoStatusIncendio.publish("Nenhum incêndio identificado.");
tempoUltimaPublicacao = millis();
tempoUltimoAlarme = 0;
}
}
}

void conectarWiFi() { Serial.print("Tentando conectar ao Wi-Fi");WiFi.begin(redeWiFi, se-
nhaWiFi);

```

```

while (WiFi.status() != WL_CONNECTED) { delay(500);
Serial.print(".");
}
Serial.println();
Serial.println("Conexão Wi-Fi estabelecida!"); Serial.print("Endereço IP: "); Serial.println(WiFi.localIP());
}

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

conectarWiFi(); conectarMQTT();

pinMode(sensorIncendio, INPUT);
pinMode(buzzerPin, OUTPUT); // Configurar o pino do buzzer como saída digital
Write(buzzerPin, LOW); // Inicialmente, o buzzer está desligado

verificarSensor();
}

void loop() {
if (!clienteMQTT.connected()) { conectarMQTT();
}

clienteMQTT.ping(); verificarSensor();
}

```

- Comunicação MQTT para Arduino enviar a mensagem para um servidor MQTT, notificando um sistema de monitoramento remoto sobre o incêndio.

## 7. Materiais

- Plataforma: Arduino Uno
- Sensor: Sensor de chama digital (KY-026)
- Atuador: Sirene, lâmpada LED
- Resistores: Para limitar a corrente
- Jumpers: Para conectar os componentes
- Placa de Prototipagem : Para facilitar a montagem
- Software: Arduino
- Ferramentas: Multímetro, alicate, soldador

## 8. Método

- Montagem do Circuito: Conexão dos componentes de acordo com o diagrama esquemático.
- Programação: Utilizamos o <http://io.adafruit.com> para escrever o código que lê o sensor de chama e aciona o atuador.
- Teste: Verificamos se o sistema funciona corretamente, simulando um incêndio com uma fonte de calor.
- Detecção precoce e precisa: A capacidade do Arduino de processar dados em tempo real e

acionar alarmes rapidamente permite a detecção precoce de incêndios, minimizando danos aos veículos e à infraestrutura do local.

- **Monitoramento contínuo:** O sistema pode operar 24 horas por dia, 7 dias por semana, fornecendo monitoramento constante do ambiente e garantindo uma resposta rápida em caso de emergência.
- **Flexibilidade e personalização:** O Arduino permite a criação de sistemas personalizados, adaptando-se às necessidades específicas de cada ambiente de armazenamento. É possível configurar diferentes níveis de sensibilidade, adicionar sensores para monitorar outros parâmetros (como temperatura e umidade) e integrar o sistema a outros equipamentos de segurança.
- **Custo-benefício:** A utilização de componentes eletrônicos de baixo custo e software livre torna a implementação de um sistema de detecção de incêndio com Arduino uma solução econômica.
- **Escalabilidade:** É possível expandir o sistema para monitorar áreas maiores, adicionando mais sensores e módulos Arduino.
- **Integração com outros sistemas:** O sistema pode ser integrado a sistemas de segurança existentes, como câmeras de vigilância e sistemas de alarme, proporcionando uma solução completa para a proteção do ambiente de armazenam
- **Considerações Específicas para Ambientes de Estocagem Automotiva:**
  - **Materiais inflamáveis:** Os veículos armazenados contêm diversos materiais inflamáveis, como plásticos, borracha e fluidos, o que torna a detecção de incêndios ainda mais crucial.
  - **Espaços confinados:** Os ambientes de armazenamento automotivo podem ser grandes e com espaços confinados, exigindo uma cobertura adequada dos sensores.
  - **Interferências eletromagnéticas:** A presença de equipamentos elétricos e eletrônicos pode gerar interferências eletromagnéticas, que podem afetar o funcionamento dos sensores.

**10. Manutenção:** É fundamental realizar a manutenção periódica do sistema para garantir seu bom funcionamento. Em resumo, a utilização de um detector de fogo Arduino em um ambiente de estocagem automotivo oferece uma solução eficaz e econômica para a prevenção de incêndios, contribuindo para a segurança dos veículos, dos funcionários e das instalações.

Outras Possibilidades:

**11. Integração com sistemas de sprinklers:** Ao detectar um incêndio, o sistema pode acionar automaticamente os sprinklers, controlando a propagação das chamas.

**12. Notificações em tempo real:** O sistema pode enviar notificações em tempo real para os responsáveis pela segurança, permitindo uma resposta rápida em caso de emergência.

**13. Análise de dados:** Os dados coletados pelos sensores podem ser utilizados para realizar análises e identificar padrões que possam indicar um risco de incêndio.

**14. Definição da arquitetura do sistema:** Determinar a quantidade de sensores, a localização ideal dos sensores, a forma de comunicação entre os sensores e o Arduino, e a interface com o usuário.

**15. Escolha dos componentes:** Selecionar os componentes eletrônicos mais adequados para o projeto, considerando fatores como custo, desempenho e disponibilidade.

**16. Desenvolvimento do software:** Criar o código do Arduino para controlar os sensores, acionar os alarmes e enviar as notificações.

**17. Testes:** Realizar testes em um ambiente controlado para verificar o funcionamento do sistema e ajustar os parâmetros, se necessário implementar um sistema de detecção de incêndio com Arduino em um ambiente de estocagem automotivo, é possível garantir um maior nível de segurança e reduzir significativamente os riscos de perda material e pessoal.

### **18. Conclusão.**

- Detecção precoce e precisa: A capacidade do Arduino de processar dados em tempo real e acionar alarmes rapidamente permite a detecção precoce de incêndios, minimizando danos aos veículos e à infraestrutura do local.
- Monitoramento contínuo: O sistema pode operar 24 horas por dia, 7 dias por semana, fornecendo monitoramento constante do ambiente e garantindo uma resposta rápida em caso de emergência.
- Incêndios, contribuindo para a segurança dos veículos, dos funcionários e das instalações

### **19. Fonte.**

- Desenvolvendo Arduino: Tutorial e projetos <https://create.arduino.cc/> HYPERLINK "https://create.arduino.cc/".
- Repositório: <https://github.com/AngelitaMH/sensor-fogo>
- Fluxograma: <https://lucid.app/>
- Dashboard: Arduino: [https://adafruit.com/users/sign\\_in](https://adafruit.com/users/sign_in)
- Vídeo Áudio: <https://youtu.be/X-ddbYqmiFA>

### **20. Fotos:**

[https://www.canva.com/design/DAGYD9Awd5I/fiqiqegepvG3CwfBDWVCQA/edit?utm\\_content=DAGYD9Awd5I&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebut-  
ton](https://www.canva.com/design/DAGYD9Awd5I/fiqiqegepvG3CwfBDWVCQA/edit?utm_content=DAGYD9Awd5I&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

### **21. Referências: Artigo de revista:**

SILVA, A. B. Da; SANTOS, C. D. Dos. Desenvolvimento de um sistema de detecção de incêndio utilizando a plataforma Arduino. Revista Brasileira de Sistemas Embarcados, São Paulo, v. 15, n. 2, p. 123-135, maio/ago. 2023

### **22. Livro:**

KARNOFSKY, D. Arduino: projetos práticos com a plataforma open-sourcemaís popular do mundo. São Paulo: Novatec, 2020.

### **23. Norma técnica:**

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 6023: informação e documentação: referências: elaboração. Rio de Janeiro, 2002

### **24 . Página da web:**

MakerHero: Tutorial Sensor de Chama com Arduino - [https://www.ma-  
kerhero.com/blog/sensor-de-chama-com-arduino/](https://www.makerhero.com/blog/sensor-de-chama-com-arduino/)

Blog Eletrogate: Guia de Uso do Sensor de Chama com Arduino - [https://blog.eleto-  
gate.com/guia-de-uso-do-sensor-de-chama-com-arduino/](https://blog.eleto-<br/>gate.com/guia-de-uso-do-sensor-de-chama-com-arduino/)

