

UNIVERSIDADE FEDERAL DE ALFENAS

RODRIGO AMORIM MUNHOZ

**AUTOMATIZAÇÃO PARA O EXPERIMENTO DE DIFUSIVIDADE em uma célula
de Arnold**

**POÇOS DE CALDAS/MG
2024**

Comentado [LAH1]: Comentários gerais:

- 1)Formatação errada
- 2)Figura e tabelas devem ser mencionadas em texto
- 3)Toda figura e tabela, depois de apresentadas, devem ser descritas e discutidas
- 4)O texto tem tudo, mas está mal descrito
- 5)Toda medida tem que ter a mesma quantidade de casas decimais que a incerteza
- 6)Toda incerteza tem a quantidade de casas decimais determinada pelos seus algarismos significativos

RODRIGO AMORIM MUNHOZ

AUTOMATIZAÇÃO PARA O EXPERIMENTO DE DIFUSIVIDADE

Produto de Conclusão PIEPEX
apresentada como parte dos requisitos
para obtenção do título de Bacharel em
Interdisciplinar em Ciência e tecnologia
pela Universidade Federal de Alfenas.

Orientador: Prof. Dr. Marlus Pinheiro
Rolemberg
Coorientador: Prof. Dr. Laos Alexandre
Hirano

POÇOS DE CALDAS/MG
2024

RESUMO

O Arduino e o ESP32 são microcontroladores programáveis capazes de executar diversas tarefas de forma automatizada tendo à disposição uma enorme variedade de sensores e módulos garantindo uma maior precisão de medidas se comparado aos métodos convencionais utilizados nos experimentos nas universidades. Experimentos com o da célula de Arnold, o estudante precisa realizar medidas com determinada periodicidade de forma manual. Por se tratar de um experimento demorado, usualmente, é obtida uma medida a cada 12 horas por 5 dias. Este intervalo de tempo entre medidas podem trazer interferências ao resultado do experimento, como variações abruptas de temperatura e pressão sobre a célula ou pela quantidade reduzida de pontos experimentais que podem comprometer parâmetros estatísticos. . Neste trabalho, uma célula de Arnold foi integrada a um protótipo que possui uma placa ESP32 CAM para obter fotos da coluna de ensaio e um sensor BMP280 que mede a temperatura e pressão atmosférica. A aquisição destas imagens e medidas de pressão foram obtidas automaticamente e armazenadas na memória do dispositivo tendo como intervenção do operador apenas sua programação e instrução para iniciar o experimento. Devido à automatização deste sistema de aquisição das medidas experimentais, obteve-se uma quantidade muito maior de pontos foram obtidas reduzindo o valor do χ^2 referente aos ajustes dos pontos a uma reta. Além disso, os dados obtidos foram significativamente precisos em relação às medidas obtidas diretamente com uma régua usada para comparação

Comentado [LAH2]: Cite o valor

Coloque o abstract

Sumário

Lista de Figuras

Lista de Tabelas

INTRODUÇÃO

Para a formação de engenheiros a necessidade da realização de experimentos práticos durante o curso de graduação é indispensável, pois permite o discente ter contato com equipamentos que podem ser utilizados e proporciona a prática do que se aprendeu durante as aulas. Porém alguns experimentos podem durar dias com necessidade de obtenção dos dados regularmente, necessitando que o discente esteja presente na faculdade para a coleta desses dados, onde muitos não tem esse tempo à disposição ou forma de se deslocar.

A título de exemplo, o experimento com a célula de Arnold consiste em monitorar a evaporação de um fluido alojado em um capilar com a tampa aberta para determinar a difusividade. Neste experimento, observa-se o volume evaporado, a temperatura e a pressão em torno da célula em função do tempo. Usualmente, neste experimento, o estudante obtém um ponto experimental a cada 12 horas. Devido ao longo intervalo de tempo entre as medidas, alguns erros que podem ser críticos ao experimento podem ocorrer, tais como, ajuste com poucos pontos experimentais que resultam em ajustes de baixa qualidade, e a impossibilidade em monitorar flutuações de temperatura e pressão nos intervalos entre as medidas. Portanto, o desenvolvimento de dispositivos que possam coletar pontos experimentais de forma automática é desejável pela possibilidade de programas rotinas que permitam diminuir o intervalo de coletas sem a necessidade de intervenções de um operador.

OBJETIVOS

Principal

Instrumentalizar uma célula de Arnold com um dispositivo capaz de medir, dentro de um prazo total e em intervalos programáveis, a altura da coluna do fluido dentro de um capilar, a temperatura e a pressão com a finalidade de determinar a difusividade

Específicos

- Desenvolver uma célula de Arnold para medir a difusividade de xxx, www e yyy
- Integrar e ajustar a câmera equipada no ESP32 CAM para medir o menisco do fluido dentro de um capilar
- Integrar o módulo BMP280 para medir a temperatura e a pressão no momento exato que as fotos forem obtidas pelo ESP32 CAM
- Comparar medidas de difusividade obtidas sem e com o dispositivo com valores descritos na literatura

REVISÃO BIBLIOGRÁFICA

Incluir:

- **O que é difusividade? Inclua uma tabela com valores de difusividade de alguns solventes mais comuns**
- **Quais são as formas para medir difusividade**
- **Descrição da Célula de Arnold**

Difusividade do éter

Utilizando a equação de Fuller [citar referência] podemos calcular a difusividade do éter

$$D_{AB} = \frac{0.00143 T^{1.75}}{P \sqrt{M_A + M_B} (\sum V_A + \sum V_B)^2}$$

onde D_{AB} é a difusividade, P é a pressão T é a temperatura, M_A é a massa molar do éter, M_B é a massa molar do ar, V_A é soma dos volumes atômicos do éter e V_B é a soma dos volumes atômicos do ar.

Para uma temperatura de 303,14 K e uma pressão de 94164,55 Pa o D_{AB} resultante é de $8,71 \times 10^{-8} \text{ m}^2/\text{s}$.

Arduíno

Segundo MCROBERTS (2018) o Arduíno é uma forma fácil e de baixo custo para conectar sensores podendo também ser utilizado para muitos outros propósitos.

É um microcontrolador que pode ser programado a partir de um computador com uma interface de comunicação e utilizando a linguagem C++, possuindo diversas saídas e entradas analógicas e digitais, e podendo servir como fonte de energia com saídas de 3,3 e 5 v.



Foto de um Arduíno UNO

Esp33 CAM

O ESP32 CAM é um dos menores e mais baratos módulos de câmera que possui a flexibilidade de ser controlado por um computador externo (Dietz. et al ,2022)

Possui um processador dual core, modulo WIFI e Bluetooth, entrada para cartão sd, uma câmera de 2MP e pode ser programado pela interface de programação do Arduino. Apesar de não poder ser conectado diretamente a um computador existem diversos módulos para essa função como o ESP32 CAM MB, permitindo a conexão através de um cabo micro USB.



Foto do ESP32 CAM

METODOLOGIA

Comentado [LAH3]: Faltou descrever o BMP280

Materiais

Arduíno UNO
ESP32 CAM
ESP32 CAM MB
Sensor BMP280
Pipetas
Suporte universal
Lâmpada
Computador
Suporte para o ESP32

Comentado [LAH4]: Volume, diâmetro, material da pipeta, graduação, etc

Comentado [LAH5]: Descreva melhor

Tabela X: Materiais

Montagem do experimento baseado no roteiro do ICT

O experimento foi montado com base no Roteiro utilizado pela UNIFAL [cite como referência], onde uma pipeta fechada foi preenchida com éter etílico e colocada no suporte universal, conforme apresentado na Figura 1.

Comentado [LAH6]: Descreva melhor este roteiro

Inclua a descrição da pipeta e o papel da régua que vc mostrou na outra figura. Não esqueça de finalizar a frase com “conforme apresentado na Figura 2”.

Comentado [LAH7]: Toda figura e tabela deve ser mencionado no texto

Esta montagem foi acomodada em bancada em temperatura e pressão ambiente e A distância, a temperatura e pressão foram medidas duas vezes ao dia, durante 5 dias. A temperatura ambiente foi medida com um xx modelo e marca xx, a pressão medida com um xx modelo e marca xx e a distância da régua.

Comentado [LAH8]: Qual distância? Por isso a importância do parágrafo anterior de vc explicar o que faz aquela régua

Incluir:

- Este experimento foi repetido? Quantas vezes? Se fez várias vezes, o que fez com os valores medidos? Tirou média?
- Quais incertezas foram consideradas? Como estas foram calculadas?



Figura X: Montagem do experimento 1

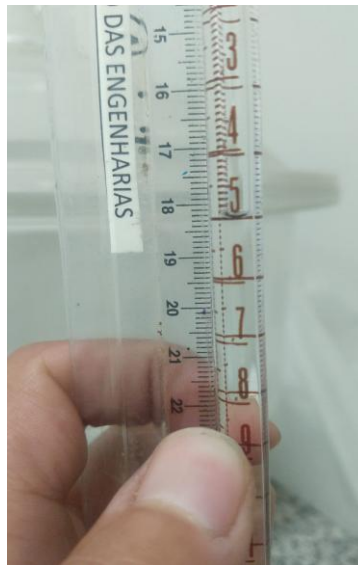


Figura X: Medições do experimento 1

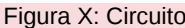
Circuito

O ESP32 CAM foi ligado ao modulo ESP32 CAM MB permitindo assim a comunicação através de uma porta serial USB com o computador.

O Arduíno foi conectado ao sensor BMP280 e então ligado a uma porta serial USB do computador.

A Figura X mostra como a ligação foi realizada.

Comentado [LAH9]: Faltou descrever a programação. Neste caso, explique sucintamente como funciona o código. No final, diga que o código está descrito na íntegra no Anexo A



Para que as informações fossem armazenadas no computador foi feito um código em python que monitorava as portas seriais referentes a cada USB e armazenava as fotos sendo o nome da foto o tempo decorrido, temperatura e pressão respectivamente.

O experimento foi primeiramente montado utilizando o Roteiro utilizado pela UNIFAL onde uma pipeta fechada foi preenchida com éter etílico e colocada no suporte universal.

Foi montado o suporte combinado ao ESP32 de maneira que a foto pudesse focar a pipeta e o papel milimetrado e a placa foi conectada ao computador.

Uma lâmpada foi colocada de maneira a melhorar a luminosidade da foto. Conforme apresentado na Figura 3

Comentado [LAH10]: Mesma coisa. Faltou descrever a programação. Neste caso, explique sucintamente como funciona o código. No final, diga que o código está descrito na íntegra no Anexo B

Comentado [LAH11]: Me parece que isso faz parte do experimento 2, certo? Se sim, coloque na subseção do experimento 2.

Comentado [LAH12]: Isso já foi dito. Ficaria melhor algo como “À montagem do experimento 1, foi fixado a câmera do ESP32 CAM blabla”

O Arduíno foi conectado ao computador e foi executado o script em python, antes da realização do experimento foram tiradas algumas fotos para calibragem da câmera e funcionamento do sensor.

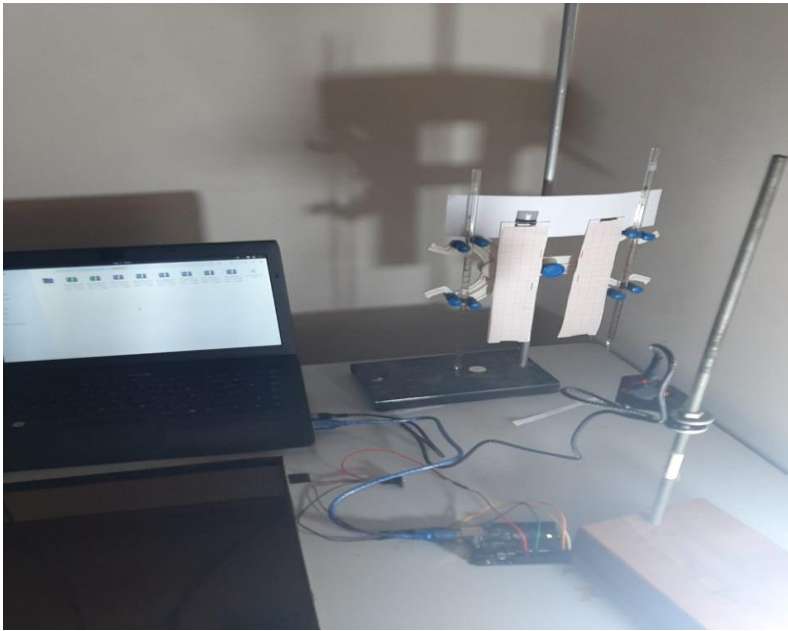


Figura X: Montagem do experimento 2

Procedimento

A câmera foi programada para tirar uma foto a cada duas horas e a cada foto o sensor faz a medição da temperatura e pressão, o script então salva esses dados no computador. A duração do experimento foi de cinco dias sendo 65 fotos no total.

Comentado [LAH13]: Reescreva em português. Tá confuso

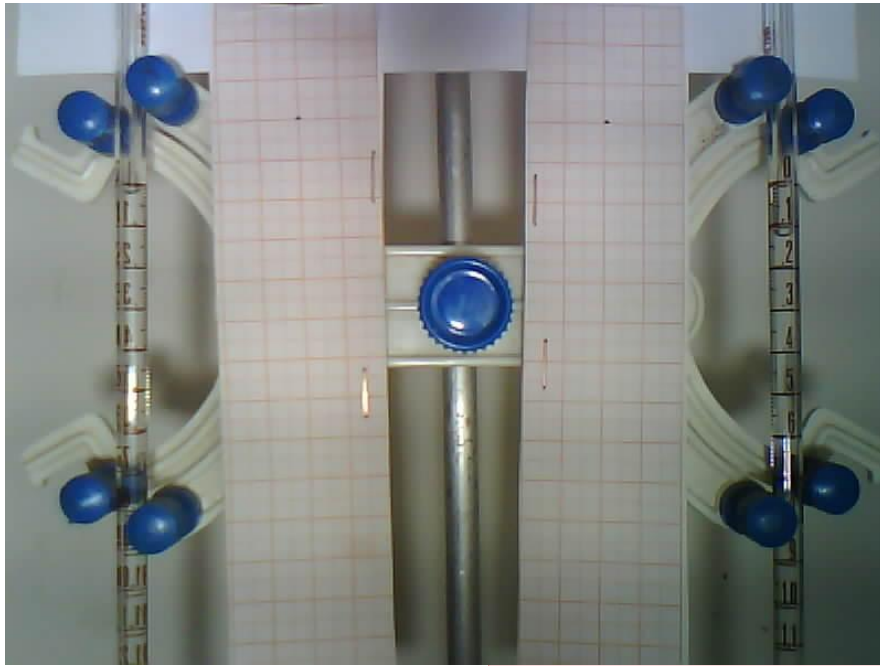


Figura X: Foto capturada pelo ESP32

Comentado [LAH14]: Isso é resultado. Em resultados, apresente isso como ficou a foto obtida para a determinação da posição do menisco

Apos o término dos cinco dias foi medido manualmente a variação de distância de cada foto em relação à referência, e suas respectivas temperaturas e pressões.

Comentado [LAH15]: Não entendi

Os tempos e distâncias foram linearizados de acordo com as escações abaixo:

$$L^2 = L_0^2 + \left(\frac{2D_{AB}PM_A}{\rho_{Aliq}RT} \ln \left[\frac{P}{P - P_{Avap}} \right] \right) t$$

Onde L é a distância da medição, L₀ é primeira medição, D_{ab} é a Difusividade, P é a pressão, P_{Avap} é a pressão de vapor do líquido, t é o tempo decorrido, R é a constante universal dos gases, T é a temperatura, e ρ_{Aliq} é a densidade do líquido.

A fim de obter uma reta do tipo y = ax + b utilizou-se

$$C = \frac{2D_{AB}PM_A}{\rho_{Aliq}RT} \ln \left[\frac{P}{P - P_{Avap}} \right]$$

Então isolando L² – L₀² :

$$L^2 - L_0^2 = C \cdot t$$

Após a linearização obteve-se C como coeficiente angular, assim se pode calcular D_{ab} através da eq.

$$D_{AB} = \frac{C \cdot \rho_{Aliq}RT}{2PM_A \ln \left[\frac{P}{P - P_{Avap}} \right]}$$

RESULTADOS

Experimento 1

Após os cinco dias realizando duas medidas por dia, uma de manhã e outra à tarde se obteve os seguintes dados.

Distância 0,0005(m)	±	Tempo ± 0,5 (s)	Temperatura ± 0,5 (°C)	Pressão ± (Pa)	1
0,0910		0,0	27,0	101500	
0,1320		71100,0	28,0	101200	
0,1710		159480,0	27,0	101300	
0,1830		177960,0	27,0	101500	
0,2050		246600,0	27,0	101400	
0,2140		264360,0	28,0	101400	

Tabela X: Dados do experimento 1

As médias de temperatura e pressão acompanhado dos dados da literatura utilizados foram:

Temperatura Media (K)	Pressão Media (Pa)
300,3	101383
Pressão de Vapor (Pa)	Densidade (kg/m ²)
61320	713
Constante dos gases (J/mol K)	Massa Molar (kg/mol)
8,314	0,0742

Tabela X: Médias e dados de experimento 1

Após a linearização calculou-se C e D ab:

Comentado [LAH16]: Sugestões:

- 1) Exclua as tabelas. Os gráficos já trazem as informações necessárias
- 2) Depois do gráfico, deve vir a descrição. Neste caso, aproveite 100% do que vc já colocou
- 3) Depois da descrição, faça uma discussão. Enfim, o que devo entender deste resultado?
- 4) Faça o mesmo para o experimento 2

Comentado [LAH17]: Não se usa "seguinte", e sim, conforme tabela/figura xx

C	D ab (m²/s)
1,3853e-7 ± 1,7e-9	1,7655e-5 ± 2,2e-7

Tabela X: C e D ab do experimento 1

O gráfico gerado de $L^2 - L_0^2$ foi:

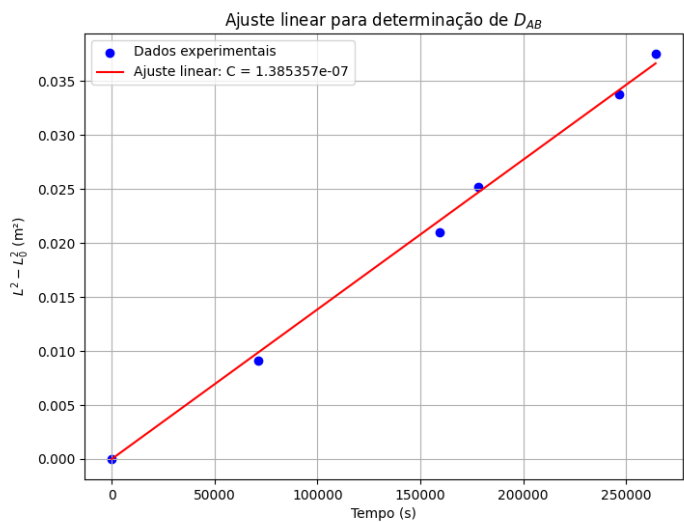


Gráfico X: Experimento 1

Experimento 2

Distancia ± 0,0005 (m)	Temperatura ± 0,01 (K)	Pressão ± 1 (Pa)	Tempo ± 1 (s)
0,0120	302,92	94327	0
0,0140	302,92	94323	7200
0,0180	302,77	94276	14400

0,019	302,18	94339	21600
0,022	302,33	94432	28800
0,025	302,65	94420	36000
0,029	303,3	94318	43200
0,034	303,94	94124	50400
0,036	304,6	93944	57600
0,041	305,27	93881	64800
0,044	305,53	93961	72000
0,05	305,6	93955	79200
0,056	308,42	93943	86400
0,06	304,57	94080	93600
0,066	304	94081	100800
0,069	303,86	94185	108000
0,072	303,56	94287	115200
0,076	303,95	94243	122400
0,078	304,38	94136	129600
0,081	304,67	93990	136800
0,085	304,76	93954	144000
0,089	303,71	93964	151200
0,092	304,44	94203	158400
0,094	304,16	94287	165600
0,097	302,92	94187	172800
0,101	303,21	94089	180000
0,105	302,8	94097	187200
0,108	302,39	94214	194400
0,11	302,18	94231	201600

0,113	301,87	94211	208800
0,115	301,65	94153	216000
0,117	301,77	94047	223200
0,119	301,81	93979	230400
0,121	301,73	94072	237600
0,123	301,46	94270	244800
0,125	301,35	94284	252000
0,127	300,9	94222	259200
0,128	301,01	94193	266400
0,13	300,35	94293	273600
0,132	299,89	94427	280800

Tabela X: Medias e dados de experimento 2

Temperatura Media (K)	Pressão Media (Pa)
303	94165
Pressão de Vapor (Pa)	Densidade (kg/m²)
61320	713
Constante dos gases (j/mol K)	Massa Molar (kg/mol)
8,314	0,0742

Tabela X: Medias e dados de experimento 2

C	D ab (m²/s)
5,7011e-8 ± 1,2688e-9	6,9608e-6 ± 1,5491e-7

Tabela X: C e D ab do experimento 2

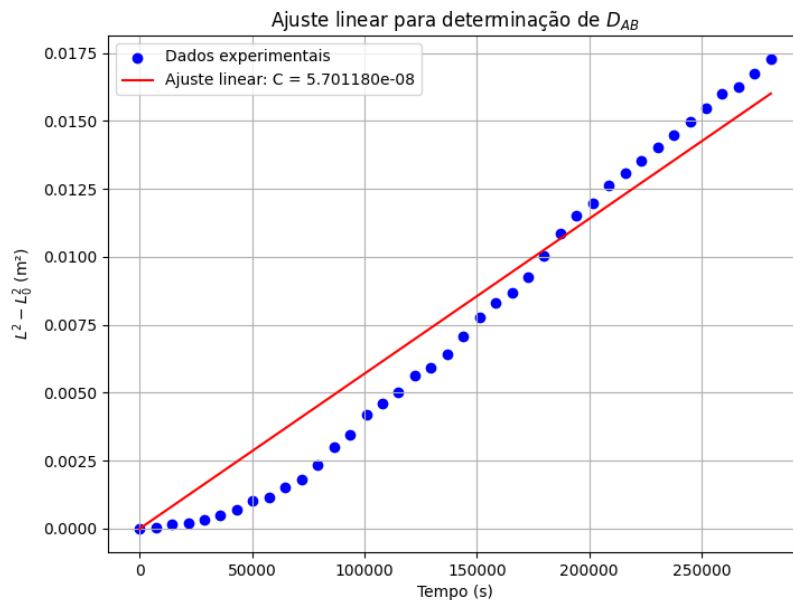


Gráfico X: Experimento 2

CONCLUSÃO

Após a obtenção e análise dos dados, se pode notar que com um maior volume e precisão dos dados o valor de D_{AB} obtido se aproxima melhor dos presentes na literatura.

Os microcontroladores se comportaram da maneira esperada capturando as fotos e medições de acordo com a programação realizada, provando que para a proposta do projeto, o implemento de automação para obter medições funciona de maneira satisfatória.

Ao comparar os gráficos se pode notar uma curva mais próxima a realidade presente no experimento 2 se comparado ao experimento 1, com a difusividade variando com as diferenças de temperatura e pressão.

Alguns pontos a levar em consideração para projetos futuros são a necessidade do projeto precisar de energia constante oque pode ser necessário um no-break em caso de falta de energia, a iluminação para que as fotos fiquem com uma boa

qualidade, tomando cuidado com sombras e reflexos que podem dificultar a análise posterior dos dados, e o foco da câmera.

REFERENCIAS BIBLIOGRÁFICAS

<https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

MCRBERTS, Michael. Arduino básico. Novatec Editora, 2018. Disponível em: https://edisciplinas.usp.br/pluginfile.php/4287597/mod_resource/content/2/Ardu%C3%ADno%20B%C3%A1sico%20-%20Michael%20McRoberts.pdf. Acesso em:11 out 2024.

DIETZ, Henry et al. ESP32-CAM as a programmable camera research platform, 2022. Disponível em: <http://aggregate.org/DIT/ei2022esp.pdf>. Acesso em:11 out 2024.

APÊNDICE

Apêndice A, programa para tirar as fotos

```
#include "esp_camera.h"
```

```
// Configuração dos pinos da câmera
```

```
#define PWDN_GPIO_NUM 32
```

```
#define RESET_GPIO_NUM -1
```

```
#define XCLK_GPIO_NUM 0
```

```
#define SIOD_GPIO_NUM 26
```

```
#define SIOC_GPIO_NUM 27
```

```
#define Y9_GPIO_NUM 35
```

```
#define Y8_GPIO_NUM 34
```

```
#define Y7_GPIO_NUM 39
```

```
#define Y6_GPIO_NUM 36
```

```
#define Y5_GPIO_NUM    21
#define Y4_GPIO_NUM    19
#define Y3_GPIO_NUM    18
#define Y2_GPIO_NUM     5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM   23
#define PCLK_GPIO_NUM   22

void setup() {
    Serial.begin(115200); // Inicializa a comunicação serial

    // Configuração da câmera
    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;
    config.pin_sscb_sda = SIOD_GPIO_NUM;
    config.pin_sscb_scl = SIOC_GPIO_NUM;
    config.pin_pwdn = PWDN_GPIO_NUM;
    config.pin_reset = RESET_GPIO_NUM;
    config.xclk_freq_hz = 10000000; // Ajusta o clock para 10 MHz
    config.pixel_format = PIXFORMAT_JPEG;
    config.frame_size = FRAMESIZE_SVGA; // Resolução da imagem (800x600)
```

```
config.jpeg_quality = 10;      // Qualidade da imagem (0 a 63)
config.fb_count = 2;          // Número de frames buffers

// Inicializa a câmera
if (esp_camera_init(&config) != ESP_OK) {
    Serial.println("Erro ao inicializar a câmera. Verifique as conexões.");
    while (true);
}
Serial.println("Câmera inicializada com sucesso!");
}

void loop() {
    // Captura a imagem
    camera_fb_t *fb = esp_camera_fb_get();
    if (!fb) {
        Serial.println("Erro ao capturar a imagem.");
        delay(5000); // Espera 5 segundos antes de tentar novamente
        return;
    }

    // Informações da imagem capturada
    Serial.println("Imagem capturada com sucesso!");
    Serial.print("Tamanho da imagem: ");
    Serial.println(fb->len);

    // Envia o cabeçalho indicando o início da imagem
    Serial.println("START IMAGE");

    // Envia os dados da imagem
    for (size_t i = 0; i < fb->len; i++) {
        Serial.write(fb->buf[i]);
    }

    // Envia o final da imagem
```

```

Serial.println("\nEND IMAGE");
Serial.println("Imagem enviada!");

// Libera o buffer da câmera
esp_camera_fb_return(fb);

// Aguarda 2 horas antes de capturar a próxima imagem
unsigned long delay_time = 7200000; // 2 horas em milissegundos
unsigned long start_time = millis();
while (millis() - start_time < delay_time) {
    delay(1000); // Evita que o loop fique inativo
}
}

```

Apêndice B, programa para medir temperatura e pressão

```

#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>

Adafruit_BMP280 bmp; // Cria um objeto BMP280

void setup() {
    Serial.begin(115200);
    Wire.begin();

    if (!bmp.begin(0x76)) {
        Serial.println("Erro ao inicializar o BMP280!");
        while (1);
    }
}

```

```

void loop() {
  // Aguardar comando do Python
  if (Serial.available()) {
    String command = Serial.readString();
    if (command == "GET_BMP280") {
      float temperature = bmp.readTemperature(); // Temperatura em °C
      float pressure = bmp.readPressure() / 100.0F; // Pressão em hPa
      Serial.print(temperature);
      Serial.print(",");
      Serial.println(pressure);
    }
  }
}

```

Apêndice C, programa em python para monitorar as portas

```

import serial
import time

# Configuração das portas seriais
esp32_port = "COM7" # Substitua pela sua porta (Windows: "COMX", Linux:
"/dev/ttyUSBX")
arduino_port = "COM5" # Porta serial do Arduino (substitua com a correta)
baud_rate = 115200
output_folder = "./" # Pasta para salvar as imagens

# Conecte-se ao ESP32
try:
    esp32 = serial.Serial(esp32_port, baud_rate, timeout=1)
    esp32.dtr = False # Desativa DTR
    esp32.rts = False # Desativa RTS
    print(f"Conectado ao ESP32 na porta {esp32_port}.")
except serial.SerialException as e:

```

```
print(f"Erro ao conectar na porta {esp32_port}: {e}")
exit()

# Conecte-se ao Arduino para ler os dados do BMP280
try:
    arduino = serial.Serial(arduino_port, baud_rate, timeout=1)
    print(f"Conectado ao Arduino na porta {arduino_port}.")
except serial.SerialException as e:
    print(f"Erro ao conectar na porta {arduino_port}: {e}")
    exit()

# Variável para controle de tempo decorrido (em horas)
elapsed_time = 0

# Função para salvar as imagens
def save_image(image_data, elapsed_time, temperature, pressure):
    filename = f"{elapsed_time}h_temp{temperature}°C_press{pressure}hPa.jpg"
    filepath = f"{output_folder}/{filename}"
    with open(filepath, "wb") as img_file:
        img_file.write(image_data)
    print(f"Imagem salva como {filepath}")

# Função para obter dados de temperatura e pressão do Arduino
def get_bmp280_data():
    arduino.write(b'GET_BMP280') # Envia um comando para o Arduino ler o sensor
    time.sleep(0.1) # Aguarda um pouco para o Arduino responder
    if arduino.in_waiting > 0:
        line = arduino.readline().decode('utf-8', errors='ignore').strip()
        try:
            temperature, pressure = map(float, line.split(","))
            return temperature, pressure
        except ValueError:
            print("Erro ao processar os dados do BMP280.")
            return None, None
```

```
return None, None
```

```
while True:
```

```
    # Recebe dados de imagem do ESP32
```

```
    if esp32.in_waiting > 0:
```

```
        line = esp32.readline().decode('utf-8', errors='ignore').strip()
```

```
    if line == "START IMAGE":
```

```
        print("Recebendo imagem...")
```

```
        image_data = bytearray()
```

```
        while True:
```

```
            chunk = esp32.read()
```

```
            if not chunk:
```

```
                continue
```

```
            image_data.extend(chunk)
```

```
        # Detecta o final da imagem
```

```
        if b"END IMAGE" in image_data:
```

```
            image_data = image_data.split(b"END IMAGE")[0]
```

```
            print(f"Imagem recebida com {len(image_data)} bytes.")
```

```
            break
```

```
    # Obtém os dados de temperatura e pressão do Arduino
```

```
    temperature, pressure = get_bmp280_data()
```

```
    if temperature is not None and pressure is not None:
```

```
        # Salva a imagem com os dados no nome
```

```
        save_image(image_data, elapsed_time, temperature, pressure)
```

```
        # Incrementa o tempo decorrido em 2 horas
```

```
        elapsed_time += 2
```

```
time.sleep(1)
```