

MANUAL - 1.0.0

Pré-requisitos.....	1
Configurando o Arduino.....	2
Configurando o Python.....	3
Executando o programa.....	4
Correção de erros.....	5
Erros relacionados à biblioteca Serial.....	5
Erros relacionados ao código.....	5

O código está disponível em: <https://github.com/daguerpedro/ArduinoCSVManager>

13/08/2024

Pré-requisitos

1. Python (<https://www.python.org/downloads/>) versão $\geq 3.12.5$.
 - a. Quando for instalar o Python, verifique a opção “Adicionar python.exe ao PATH”.
 - b. Após instalar o Python, abra o prompt de comando para instalar as bibliotecas **pyserial** e **requests**.
 - c. Execute o comando: “***pip install pyserial requests***”.
2. Arduino IDE (<https://www.arduino.cc/en/software>)

Configurando o Arduino

É necessário modificar o código que é enviado ao Arduino.

1. Abrir comunicação Serial

a. Dentro da função “**setup**”:

- i. Na primeira linha adicione “**Serial.begin(9600);**”
- ii. Na ultima linha adicione “**Serial.println("Começar123");**”

exemplo de código:

```
void loop() {  
  Serial.begin(9600);  
  // Algum código no meio  
  Serial.println("Começar123");  
}
```

2. Enviar dados

O programa de recepção identifica os dados **separados por vírgulas** (CSV). Dentro da função “loop()” usando o Serial, utilize a função “**print**” para enviar cada variável, e na última variável sempre utilize a função “**println**” para delimitar o fim da sequência de dados.

exemplo de código:

```
void loop() {  
  Serial.print(horario);  
  Serial.print(",");  
  Serial.print(altitude);  
  Serial.print(",");  
  Serial.println(temperatura);  
  
  delay(1000); //Facilitar a leitura  
}
```

Configurando o Python

Abra o arquivo "RecepçãoDados.py" com um editor de texto, ou de código.

1. Altere a variável "*comport*" para a porta Serial conectada ao Arduino.

```
comport = "COM4"
```

Você pode verificar em qual porta o Arduino está diretamente pelo Arduino IDE, ou rodar o script "listarportas.py" que se encontra dentro da pasta "utils". (Abra um prompt de comando na pasta que contenha "RecepçãoDados.py" e execute o comando: "*python .\utils\listarportas.py*")

2. Altere a variável "*baud*" para a velocidade da porta Serial que foi definida no código do Arduino.

```
baud = 9600
```

3. Altere a lista "*variaveis*" de acordo com as variáveis que são enviadas pelo arduino .

```
variaveis = ["HORARIO", "ALTITUDE", "TEMPERATURA"]
```

Sempre mantenha a estrutura de lista ["X", ..., "Y"]

4. Para configurar a tabela (dígitos de alinhamento, separador de células etc) impressa no terminal acesse: <https://github.com/daguerpedro/PythonTable>

Executando o programa

Abra um prompt de comando e navegue até o diretório em que se encontra o arquivo 'RecepçãoDados.py'

Para executar o programa digite o seguinte comando e aperte enter.

(O comando irá depender da versão do python, teste os dois comando!)

```
python .\RecepçãoDados.py  
python3 .\RecepçãoDados.py
```

Após iniciado, o código estabelece uma conexão de dados entre o Arduino e o programa. Após receber a mensagem “Começar123”, o código começa a tabular os valores emitidos pelo Arduino (Valores separados por vírgula). Caso o programa não consiga iniciar a conexão, ou algum outro erro aconteça, uma informação de erro será emitida no terminal.

Para encerrar o programa basta clicar com o botão esquerdo dentro do CMD e apertar Control + C.

Os valores lidos serão salvos em tabelas separadas por semanas do ano. Para verificar isso abra o arquivo “/utils/filesaver.py” e procure pela função: `__nomeArquivoCSV()`¹.

¹ É necessário um conhecimento de Python

Correção de erros

Erros relacionados à biblioteca Serial.

```
AttributeError: module 'serial' has no attribute 'Serial'  
AttributeError: module 'serial' has no attribute 'SerialException'
```

Esse problema ocorre devido ao conflito com outra biblioteca que possui o mesmo nome, mas não os mesmos métodos, fazendo com que o Python não consiga importar a biblioteca desejada corretamente. **Para resolver, basta utilizar um ambiente virtual² do Python** e instalar a biblioteca 'pyserial'.

Erros relacionados ao código

Os erros de código 2, 3 e 5 não ocorreram durante o tempo de uso do código. No entanto, estão previstos de acordo com a documentação do Python e das bibliotecas utilizadas.

Código 1: Erro ao abrir a porta serial

- A. A porta Serial é inválida
 - a. Verifique no código se a variável '**comport**' foi configurada corretamente. Em sistemas Linux, normalmente as portas seriais começam com '**/dev/tty**'. Já em sistemas Windows, começam com a constante '**COM**', e em seguida o número da porta.
 - b. Caso necessário, utilize o arquivo '**utils/listarportas.py**' para verificar em qual porta o Arduino está se comunicando.
- B. O programa não possui permissão para acessar a porta Serial.
 - a. Nesse caso, será necessário buscar ajuda externa.

Código 2: Erro de comunicação:

- A. Ocorre devido a algum byte mal codificado, lixos de memória ou dreno excessivo de tensão. Para evitar esse erro:
 - 1. Não utilize o Arduino diretamente como fonte de alimentação para outros dispositivos, tais como LEDs ou Servomotores.
 - 2. Evite contato físico com os componentes conectados ao Arduino.
 - 3. Evite trepidações ou mexer no cabo de comunicação Serial.

Código 3: Erro ao carregar a tabela que já existia:

- A. O programa não possui permissão para abrir ou criar o arquivo, tente executar o terminal como Administrador. Caso persista, verifique se o disco de armazenamento não está lotado ou confira as permissões de escrita no diretório em que o arquivo **'RecepçãoDados.py'** se encontra.

Código 4: Erro enquanto tentava salvar a tabela:

- A. **Provavelmente**³, ocorre devido a algum byte mal codificado ou lixos de memória. Não há o que fazer.
- B. O programa não possui permissão para abrir ou criar o arquivo. Resolução igual ao **Código 3**.

Código 5: Erro enquanto tentava salvar o cabeçalho na tabela nova:

- A. O programa não possui permissão para abrir ou criar o arquivo. Resolução igual ao **Código 3**.
- A.

³ Não se tem certeza do que ocasionou o erro.