Dashboard Overlay - Documentação Completa

1 Dashboard Overlay – README

Projeto em Python para simular e/ou ler dados (Raspberry Pi) e sobrepor valores em pontos de uma imagem (ex.: fluxograma da planta). Inclui uma segunda janela com gráficos (barras com min/max e “flash” quando bate novos extremos).

1.1 1) Requisitos

- **Python 3.9+** (recomendado 3.10/3.11)
- **Sistema**: macOS, Windows ou Linux (Raspberry Pi opcional)

1.1.1 Dependências (modo simulação – qualquer computador)

```
pip install opencv-python
```

Se você usa virtualenv:

```
python -m venv .venv
source .venv/bin/activate     # macOS/Linux
# .venv\Scripts\activate      # Windows
pip install opencv-python
```

1.1.2 Dependências extras (modo Raspberry Pi – opcional)

No Raspberry Pi (Bullseye/Bookworm):

```
sudo apt update
sudo apt install python3-opencv python3-pip -y
pip install RPi.GPIO
```

Se for usar termopares (ex.: MAX6675):

```
# Exemplo: instale a lib do seu módulo de termopar
# (use a biblioteca compatível com seu hardware)
pip install max6675  # ou a lib equivalente que você utiliza
```

Habilite interfaces se precisar:

- **SPI/I2C** (para alguns conversores/ADCs): `sudo raspi-config → Interface Options → habilitar`
- Evite conflito de GPIO 2 e 3 (SDA/SCL) se forem usados por sensores

1.2 2) Arquivo principal

O projeto é um único arquivo:

- `dashboard_overlay.py`

Coloque a sua imagem de fundo (jpg/png) em uma pasta do projeto.

1.3 3) Como rodar (simulação – valores aleatórios controlados por teclado)

```
python dashboard_overlay.py --img "/caminho/para/sua_imagem.jpg" --scale 1.0
```

Parâmetros: - `--img` : caminho da imagem de fundo (obrigatório) - `--scale` : escala da janela principal (opcional). Ex.: 0.8, 1.0, 1.2

Na janela **Painel**, você verá os valores sobrepostos nos campos verdes da sua imagem. Na janela **Gráficos**, você vê os cartões com barras, min/max e o efeito visual quando atingir novo mínimo/máximo.

1.4 4) Alternar entre Simulação e Raspberry Pi

No topo do arquivo `dashboard_overlay.py` existe a chave:

```
USE_RPI = False  # False = simulação; True = tenta usar Raspberry Pi
        (GPIO/termopares)
```

- **False**: modo simulação (valores gerados a partir dos “bases” com ruído)
- **True**: tenta inicializar GPIO e leitura de termopares (se a biblioteca estiver instalada)

Se não conseguir, faz fallback para simulação e mostra um aviso.

1.5 5) Controles (teclado)

1.5.1 Selecionar variável: teclas 1–8 (na ordem exibida/na lista abaixo)

1. Temp Forno
2. Velocidade
3. Temp Tanque
4. Temp Saída Gases
5. Pressão Gases
6. Torre Nível 1
7. Torre Nível 2
8. Torre Nível 3

1.5.2 Ajustar o valor base da variável selecionada:

- **Seta ↑**: + passo pequeno
- **Seta ↓**: – passo pequeno
- **Q**: + passo grande
- **A**: – passo grande

1.5.3 Ruído global (intensidade de variação):

- **]** (colchete direito): aumentar ruído
- **[** (colchete esquerdo): diminuir ruído

1.5.4 Min/Max:

- **R**: resetar mínimos/máximos observados

1.5.5 Ajuda:

- **H**: mostrar/ocultar ajuda rápida na tela

1.5.6 Sair:

- **ESC** ou **Ctrl+C**: fecha sem traceback

> ***Observação**: os valores só mudam quando você mexe nos controles (teclado). Não há "tick" automático por tempo no modo simulação (evita variações indesejadas).*

1.6 6) Posicionamento dos valores na imagem

As posições dos 8 campos são proporcionais à imagem (0.0–1.0) e ficam em:

```
POSITIONS_NORM = {
    "Temp Forno":        (0.44, 0.42),
    "Velocidade":        (0.44, 0.57),
    "Temp Tanque":       (0.44, 0.77),
    "Temp Saída Gases":  (0.217, 0.53),
    "Pressão Gases":     (0.217, 0.59),
    "Torre Nível 1":     (0.78, 0.81),
    "Torre Nível 2":     (0.77, 0.56),
```

```
    "Torre Nível 3":     (0.75, 0.38),
}
```

Para ajustar: 1. Ative o modo de ajuda do mouse (já vem ativo) 2. Mova o cursor sobre a imagem; no canto superior esquerdo aparecem as coordenadas normalizadas (x, y) 3. Copie esse par para o campo correspondente no dicionário `POSITIONS_NORM`

1.7 7) Gráficos e efeitos visuais

- A janela **Gráficos** mostra cartões com barra horizontal para cada variável
- Quando um valor atinge um novo mínimo ou novo máximo, o cartão "pisca":
  - **Novo Máximo**: flash azulado + marcador no fim da barra
  - **Novo Mínimo**: flash avermelhado + marcador no início da barra

Os ranges (escalas) podem ser ajustados em:

```
RANGES = {
    "Temp Forno":       (0.0, 600.0, "°C"),
    "Temp Tanque":      (0.0, 400.0, "°C"),
    "Temp Saída Gases": (0.0, 600.0, "°C"),
    "Torre Nível 1":    (0.0, 400.0, "°C"),
    "Torre Nível 2":    (0.0, 400.0, "°C"),
    "Torre Nível 3":    (0.0, 400.0, "°C"),
    "Pressão Gases":    (0.0, 10.0,  "bar"),
    "Velocidade":       (0.0, 2000.0,"rpm"),
}
```

1.8 8) Pinagem e notas (para uso futuro no RPi)

Constantes definidas no topo do arquivo:

```
# Termopares (CLK, CS, DO) — ajuste conforme seu módulo/conexão
THERMO_TORRE_1 = (25, 24, 18)
THERMO_TORRE_2 = (7,  8,  23)
THERMO_TORRE_3 = (21, 20, 16)
THERMO_TANQUE  = (4,  3,  2)   # ATENÇÃO: 2/3 são SDA/SCL (I2C)
THERMO_GASES   = (22, 27, 17)
THERMO_FORNO   = (11, 9,  10)

# Sensores de pressão (ex.: via ADC I2C/SPI; placeholders)
PRESSAO_1_PIN = 2
```

```
PRESSAO_2_PIN = 3

# Atuadores
PIN_VENTILADOR = 14
PIN_RESISTENCIA = 26
PIN_MOTOR_ROSCA = 12
PIN_TAMBOR_DIR  = 13
PIN_TAMBOR_PUL  = 19
```

> ***Importante**: Se você usar I2C (GPIO 2/3) para ADC/sensores, não reutilize-os para outra função ao mesmo tempo. Ajuste a pinagem conforme o seu hardware.*

1.9 9) Problemas comuns & soluções

1.9.1 Janela “Gráficos”/“Painel” não abre

- Verifique se `opencv-python` está instalado e se há suporte a GUI (em servidores, use uma máquina com desktop)

1.9.2 Imagem não carrega

- Confira o caminho passado em `--img` e a extensão (png/jpg)

1.9.3 Nada muda ao mexer no teclado

- Clique uma vez na janela Painel para garantir foco
- Use as teclas listadas na seção Controles

1.9.4 Raspberry Pi não lê sensores

- Confirme `USE_RPI=True`
- Instale a/lib correta do seu termopar (MAX6675 ou equivalente)
- Habilite SPI/I2C se necessário e ajuste a pinagem

1.10 10) Roadmap rápido (se quiser evoluir)

- [ ] Conectar leituras reais (termopares, ADC de pressão)
- [ ] Adicionar lógica de controle (ex.: ligar ventilador/resistência por setpoint com histerese)
- [ ] Persistir logs (CSV) dos valores e min/max

- [ ] Página web local (Flask/FastAPI) para monitoramento remoto

---

> ***Qualquer ajuste fino*** *(ranges, passos de tecla, layout dos cartões ou posições), me diga que já deixo pronto no arquivo.*