

Universidade Tecnológica Federal do Paraná (UTFPR)  
Departamento Acadêmico de Eletrônica (DAELN)

# SISTEMAS EMBARCADOS

## **Laboratório 1**

Prof. André Schneider de Oliveira

[andreoliveira@utfpr.edu.br](mailto:andreoliveira@utfpr.edu.br)

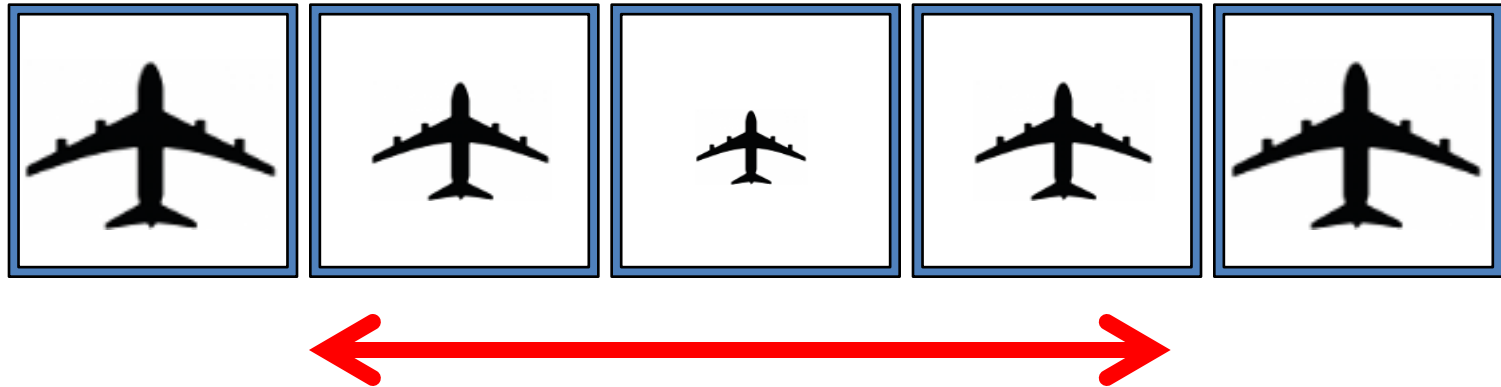
# Objetivo

- Escrever uma função em linguagem Assembly Cortex-Mx que realizará o redimensionamento das imagens armazenada em memória para representação no display Oled ou terminal UART.
- A função de manipulação de imagem ser chamada a partir de um programa em linguagem C (ver especificação do padrão ATPCS).

# Especificações

- O código C++ gerenciará o display Oled ou UART (podendo utilizar a biblioteca exemplo do projeto base) e fará a interface com os dispositivos de entrada (botões).
- Toda a codificação de redimensionamento da imagem deverá ser feita pela função Assembly. A abordagem é livre mas deve respeitar o conteúdo da imagem (ela deve ficar compreensível).

# Especificações



- O tamanho base deve ser aumentado e decrescido **5 vezes**
- Duas imagens devem ser utilizadas

# Especificações

1. A imagem deve ser representada automaticamente no display/UART, com transições a cada 500ms
2. A imagem deve ser redimensionada aumentando 5 vezes e reduzindo 5 vezes (11 transições a cada ciclo)

$$T_{N/32} \leftrightarrow T_{N/16} \leftrightarrow T_{N/8} \leftrightarrow T_{Nx/4} \leftrightarrow T_{N/2} \leftrightarrow \mathbf{T_N} \leftrightarrow T_{Nx2} \leftrightarrow T_{Nx4} \leftrightarrow T_{Nx8} \leftrightarrow T_{Nx16} \leftrightarrow T_{Nx32}$$

1. O botões devem fazer a seleção da imagem e a inversão da cor (inversão de preto e branco)

# Dicas Importantes

- A imagem de entrada possui três valores de 8 bits (R, G e B) para cada pixel e terá  $3 \cdot \text{dim\_x} \cdot \text{dim\_y}$  pixels na sequência R, G, B, R, G, B, ..., na ordem da esquerda para a direita e de cima para baixo.
- No site da existe um banco com imagens de teste.



# Atividades a Serem Realizadas

- Planejar estruturas de dados *antes* de iniciar a implementação
- Usar o código-exemplo como base
- Planejar casos de teste – sugestão: utilizar o formato de arquivo de imagem Portable Pixel Map (PPM) para armazenar as imagens em disco



# Cronograma de Avaliação – Lab 1

- 19/8
  - Apresentação das funções em C funcionando (display/UART e botões) diretamente para o professor (sem apresentação em slides).
  - **A imagem deve ser apresentada no display.** A não execução acarretará na **perda de 1 ponto** na nota do laboratório.
- 26/08
  - Demonstração e teste