Controlo do movimento Pan e Tilt com a visualização de imagem RGB ou Profundidade

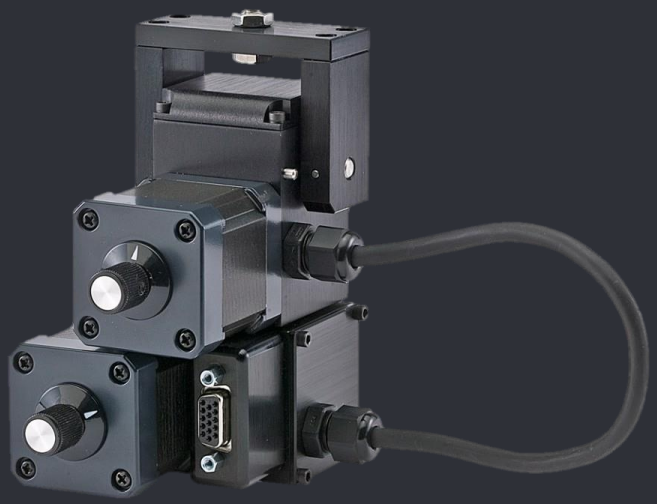
João Santos
76912
MIEM – 2018/19

Sistemas Remotos

Pan and Tilt Unit | Xtion PRO Live

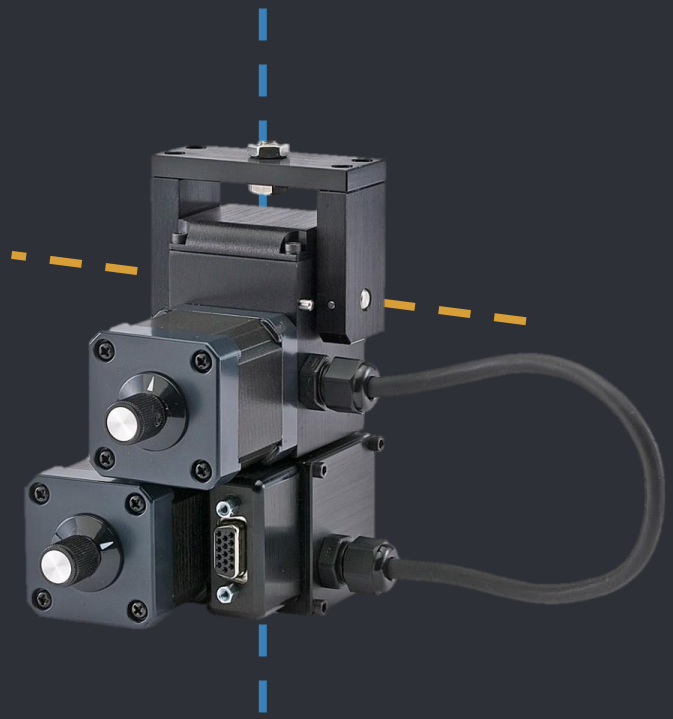
Equipamentos

Pan and Tilt Unit



Xtion PRO Live

Pan and Tilt Unit

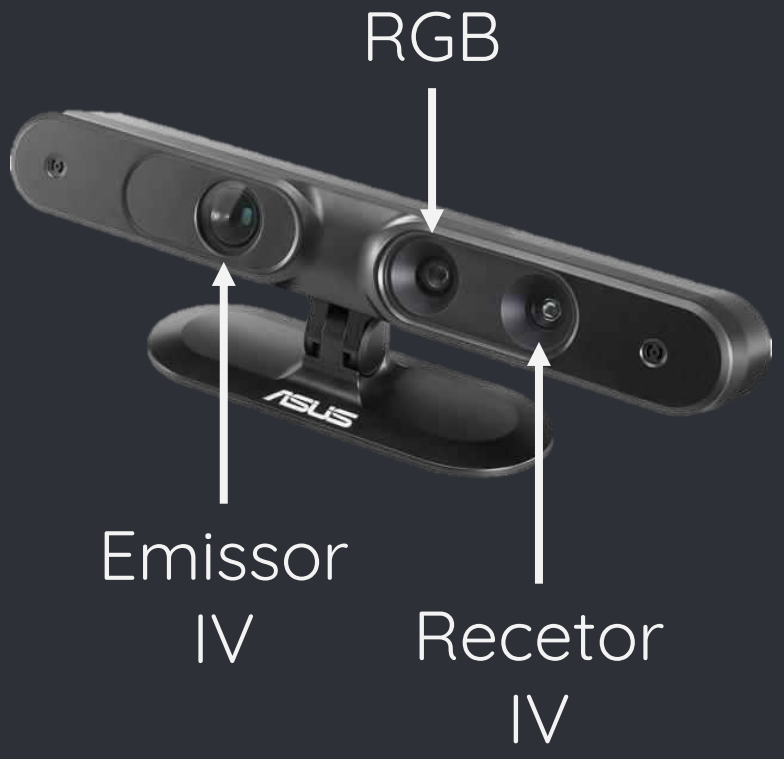


- Directed Perception, Inc.
- PTU-46-17.5
- Pan: ±159°
- Tilt: -47° a +31°
- Resolução:
  - 0,051428°
- Porta série
- RS232

Equipamentos

- Asus
- RGB
- Profundidade
- USB 2.0
- De 0,8m a 3,5m
- FOV:
  - 58° H
  - 45° V
  - 70° D
- 640 x 480: 30fps
- OpenNI2

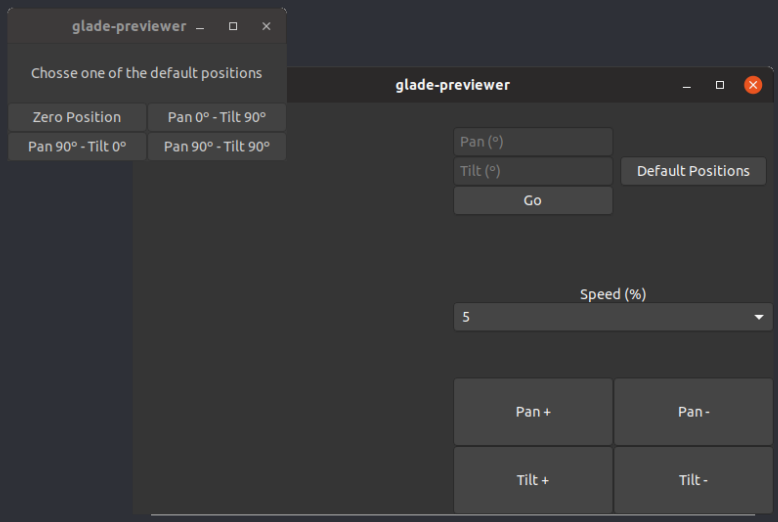
Xtion PRO Live



Objetivos Iniciais

Plano de trabalho

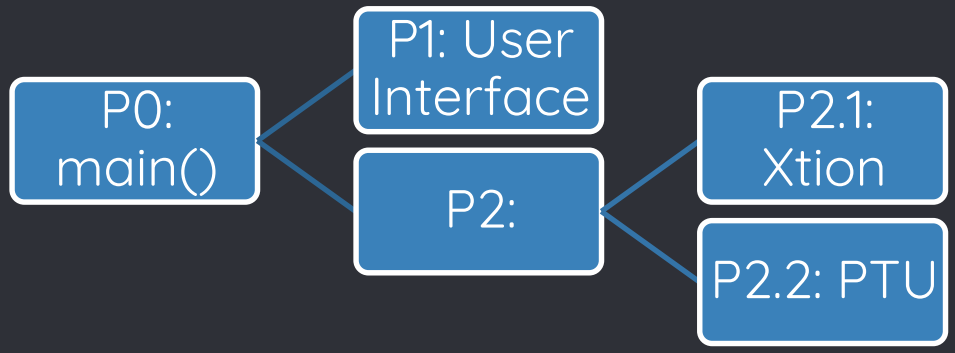
- Visualizar imagem
- Controlar posição absoluta
- Controlar velocidade
- Movimentos incrementais
- Posição pré-definidas



Arquitetura

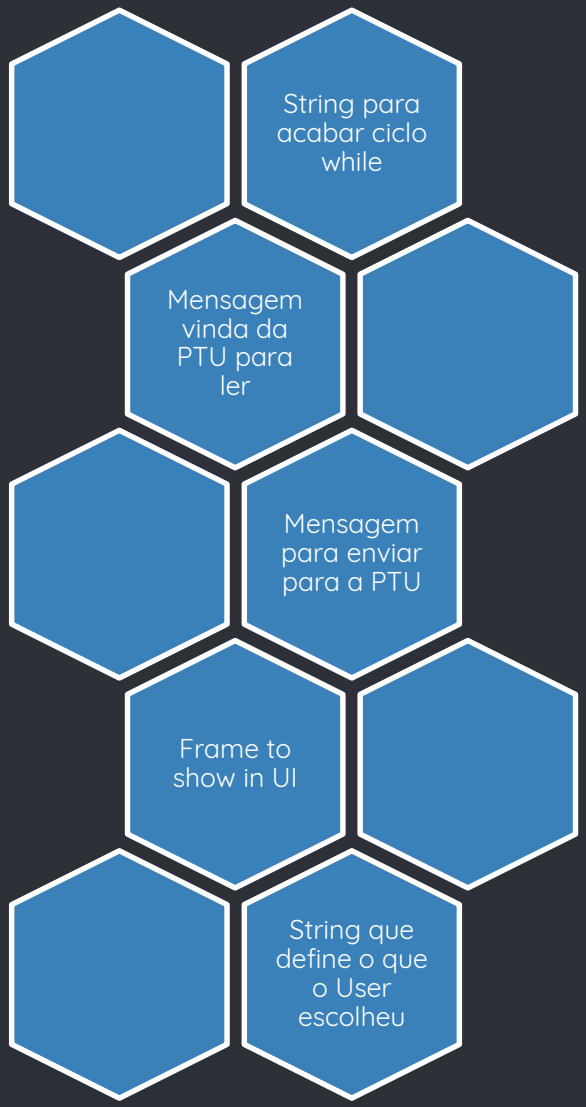
Resumo

- Processos: 3



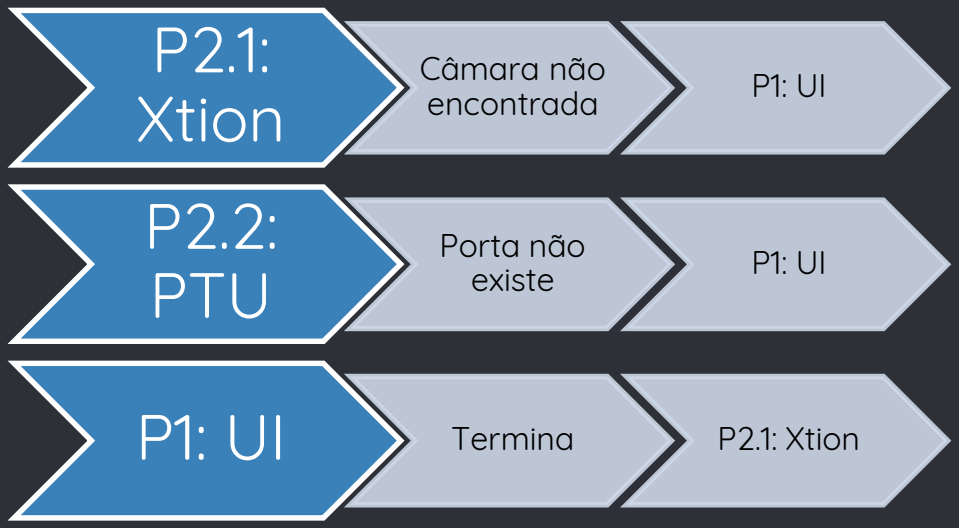
Resumo

- Processos: 3
- Shared Memories: 5



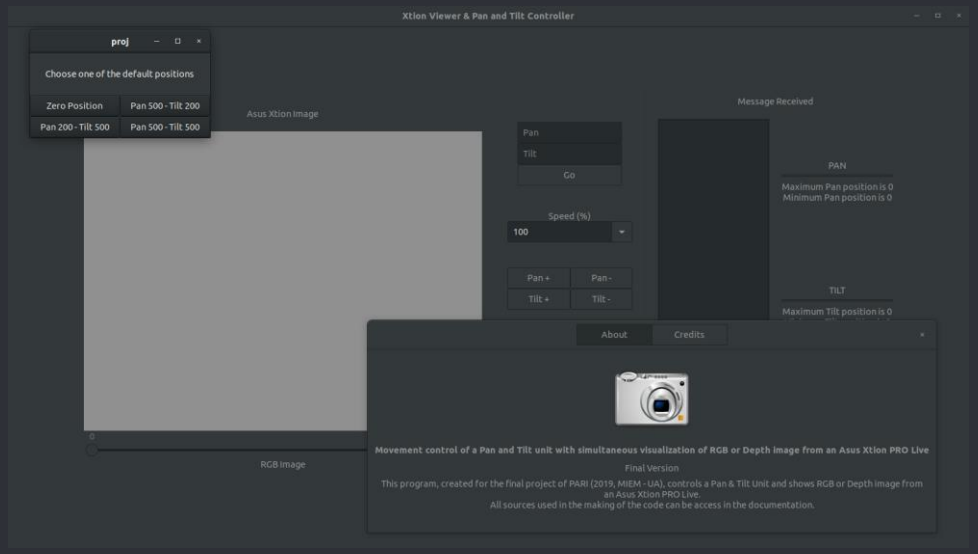
Resumo

- Processos: 3
- Shared Memories: 5
- Signals: 3



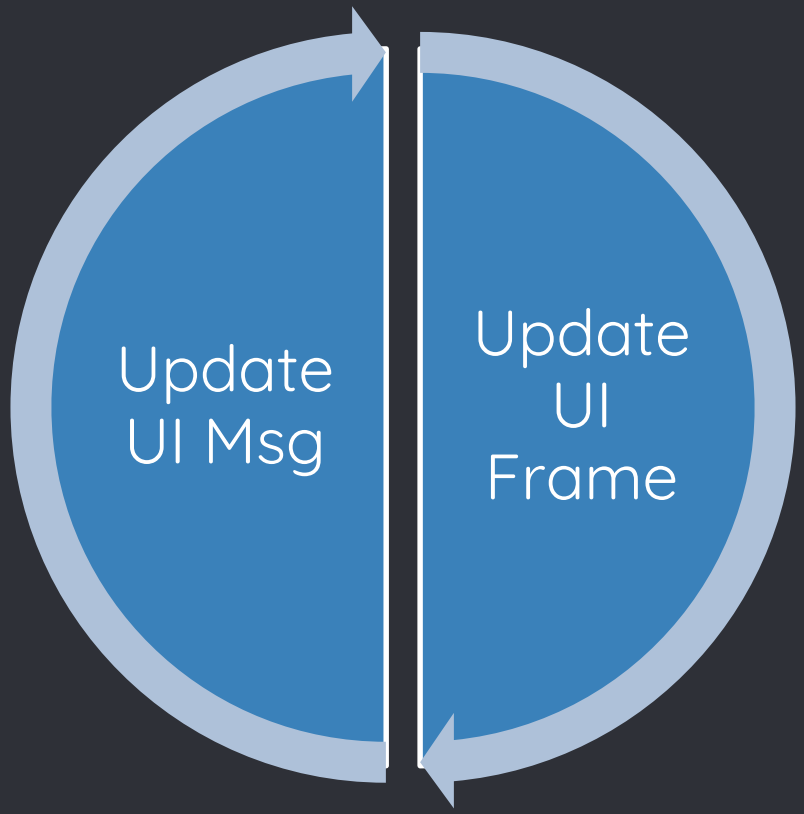
Resumo

- Processos: 3
- Shared Memories: 5
- Signals: 3
- Interface: 1
  - Janelas: 3



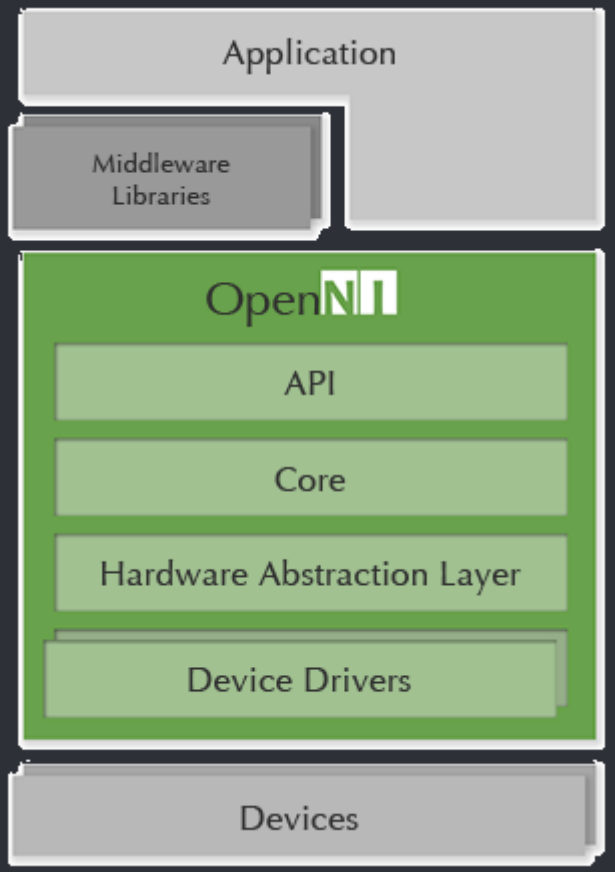
Resumo

- Processos: 3
- Shared Memories: 5
- Signals: 3
- Interface: 1
  - Janelas: 3
- Funções Idle: 2



Resumo

- Processos: 3
- Shared Memories: 5
- Signals: 3
- Interface: 1
  - Janelas: 3
- Funções Idle: 2
- Biblioteca OpenSource: 3
  - GTK
  - OpenCV
  - OpenNI



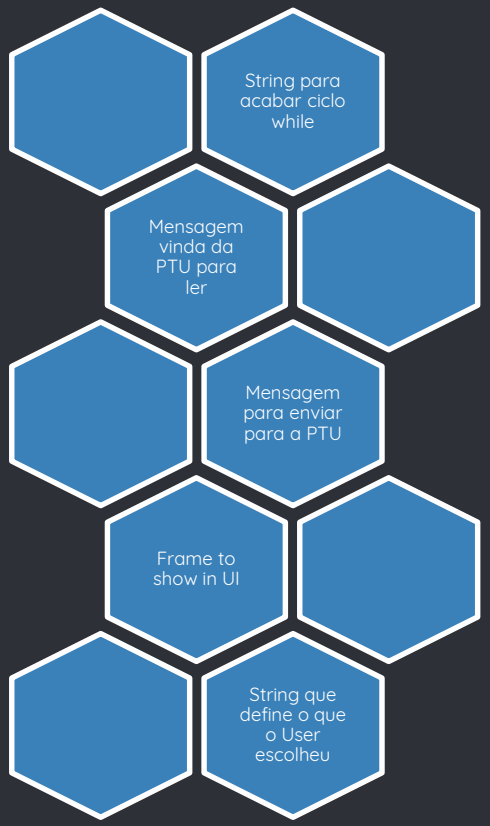
1

Processos

Descrição

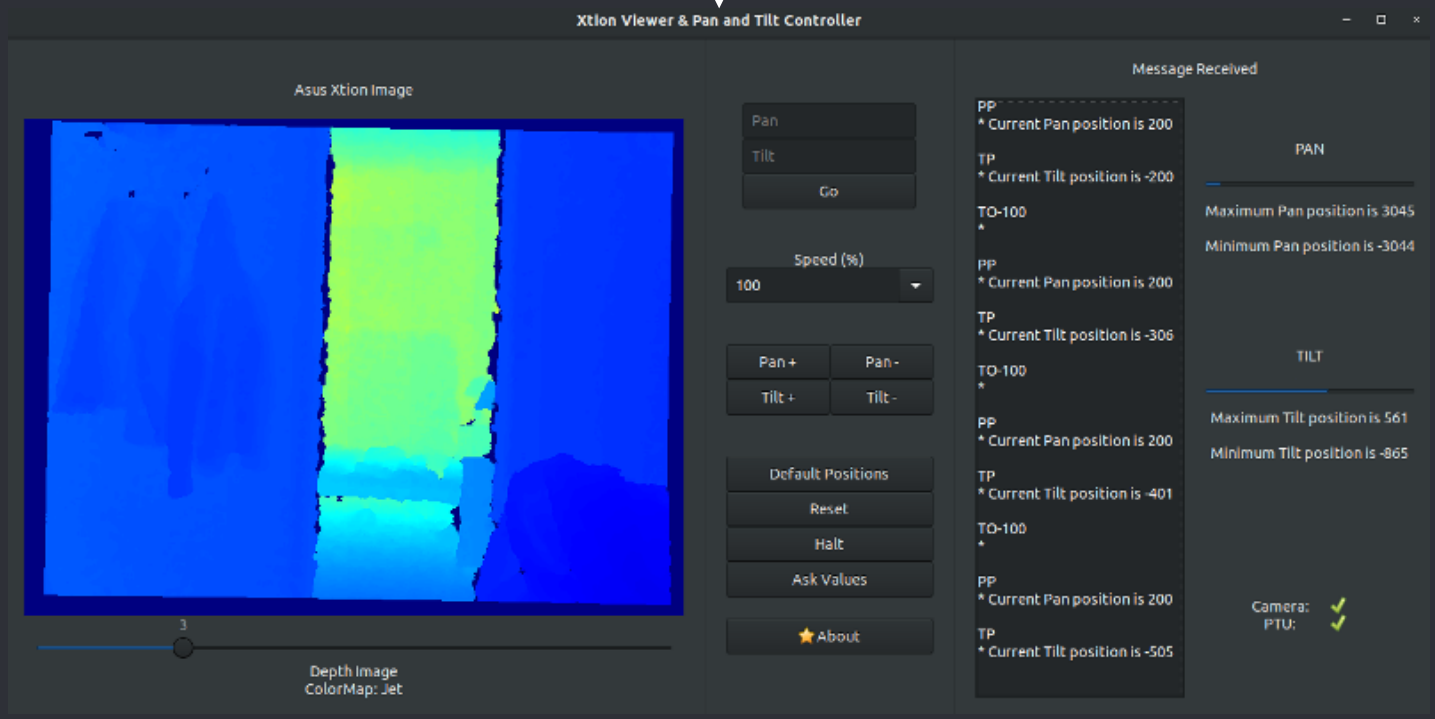
P0 main()

- Obtém id's das Shared Memories

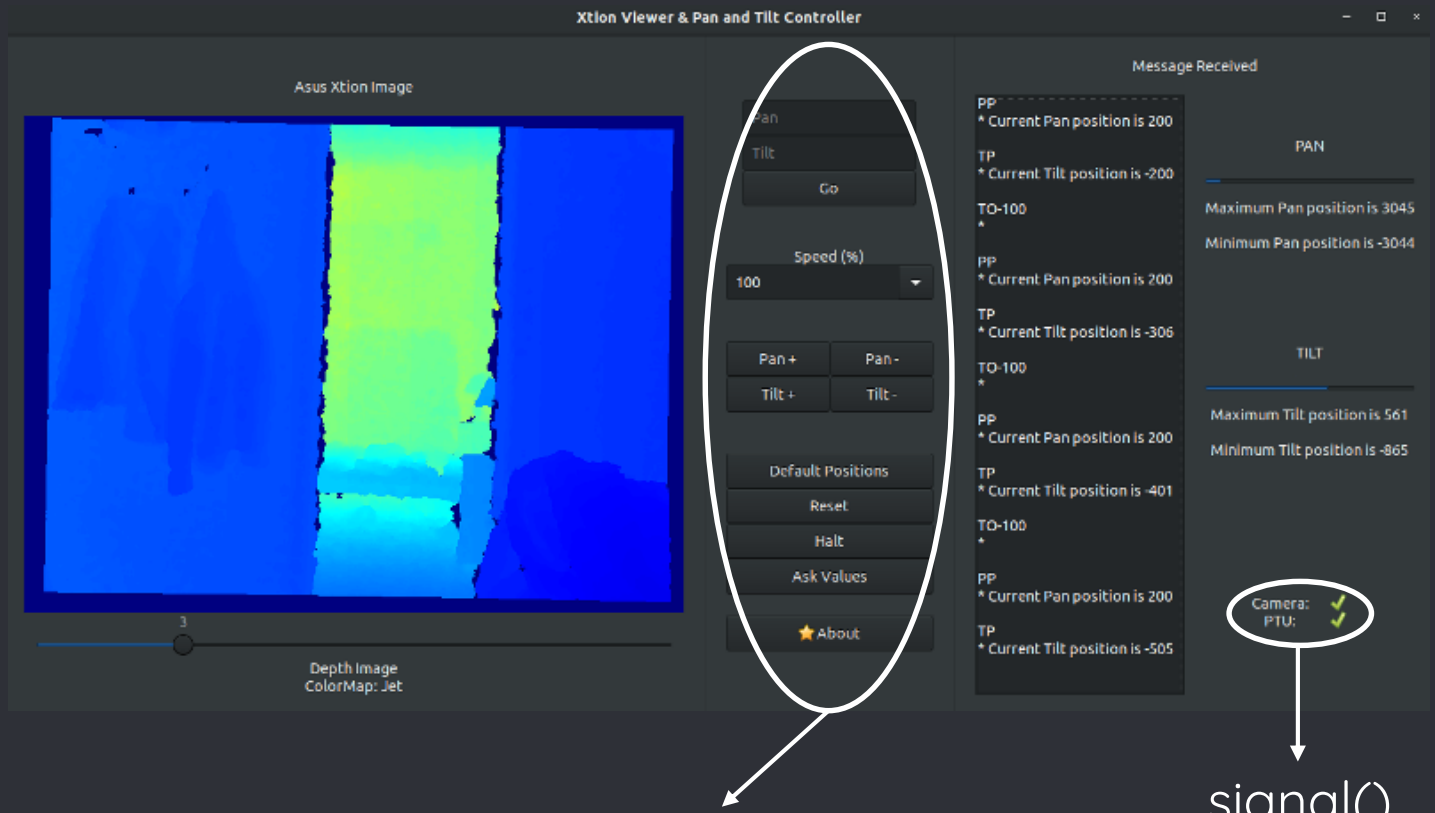


P1 UI

UI.glade



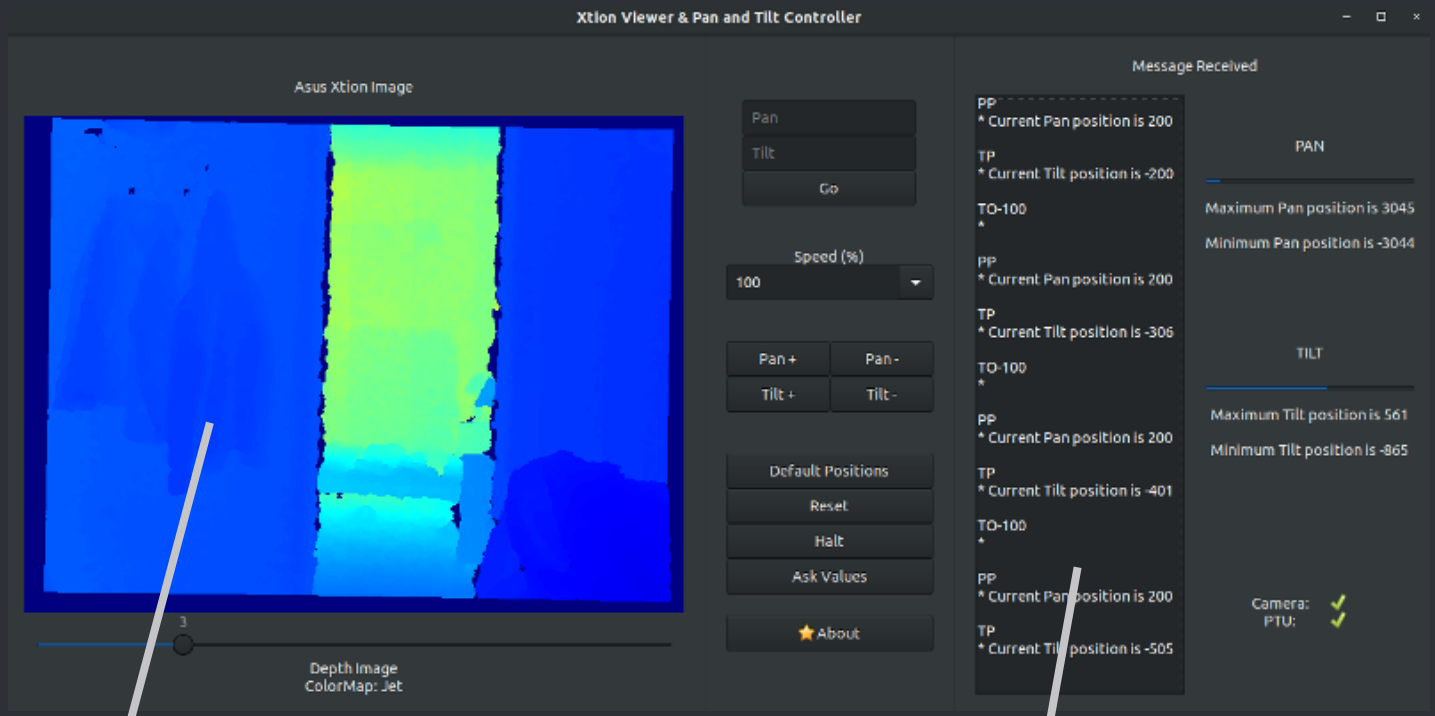
P1 UI



gtk_builder_connect_signals()

signal()

P1 UI

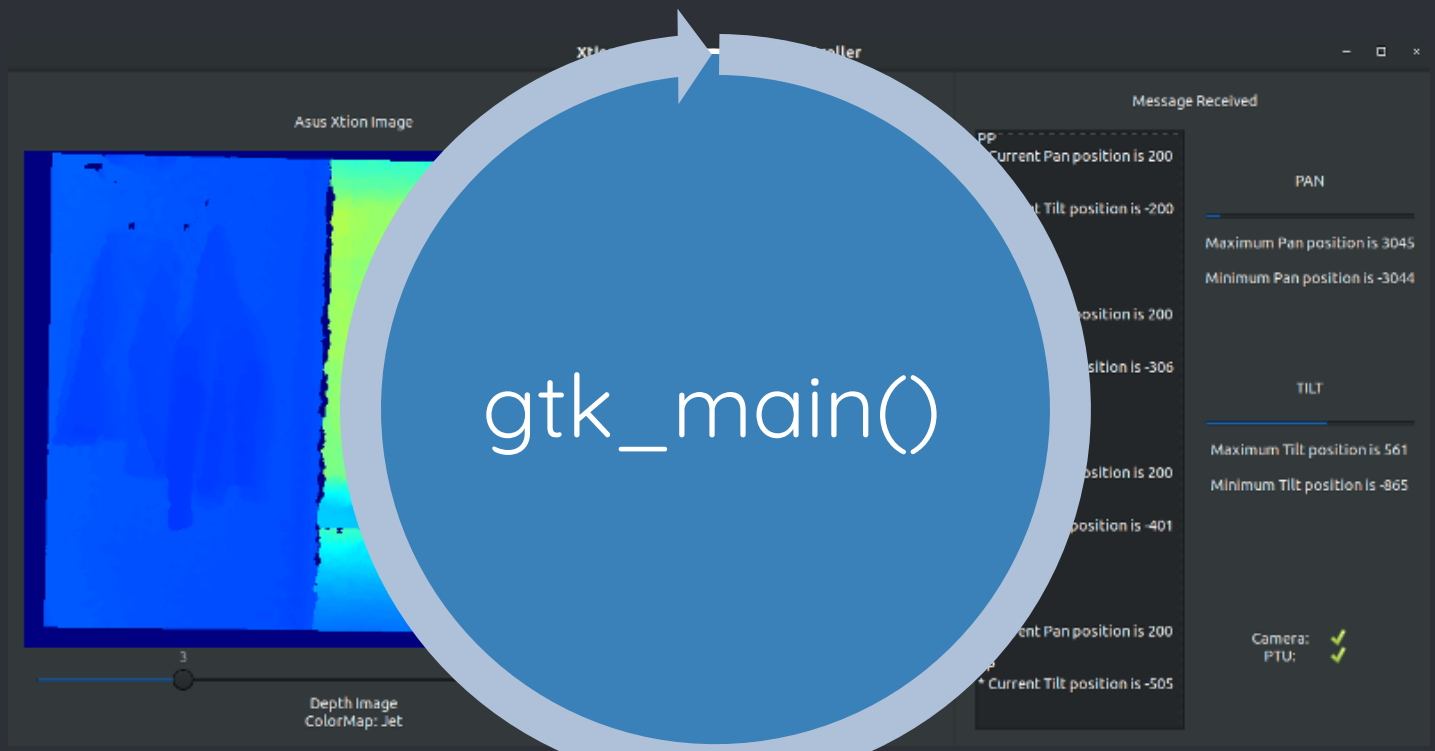


Widget GtkImage *

Widget GtkTextView *

idle

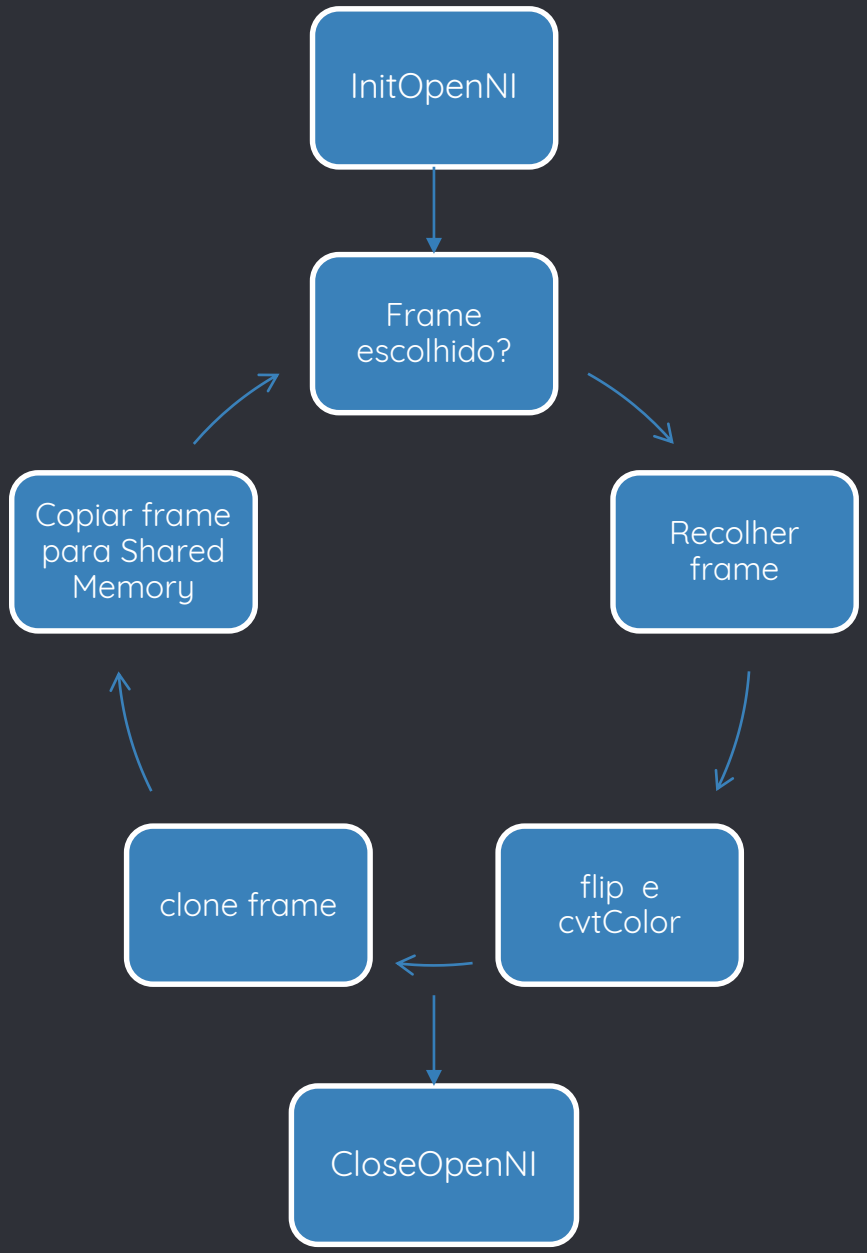
P1 UI


Asus Xtion Image
Depth Image
ColorMap: Jet
Message Received
Current Pan position is 200
Tilt position is -200
PAN
Maximum Pan position is 3045
Minimum Pan position is -3044
TILT
Maximum Tilt position is 561
Minimum Tilt position is -865
Camera:
PTU:
Current Tilt position is -505
gtk_main()

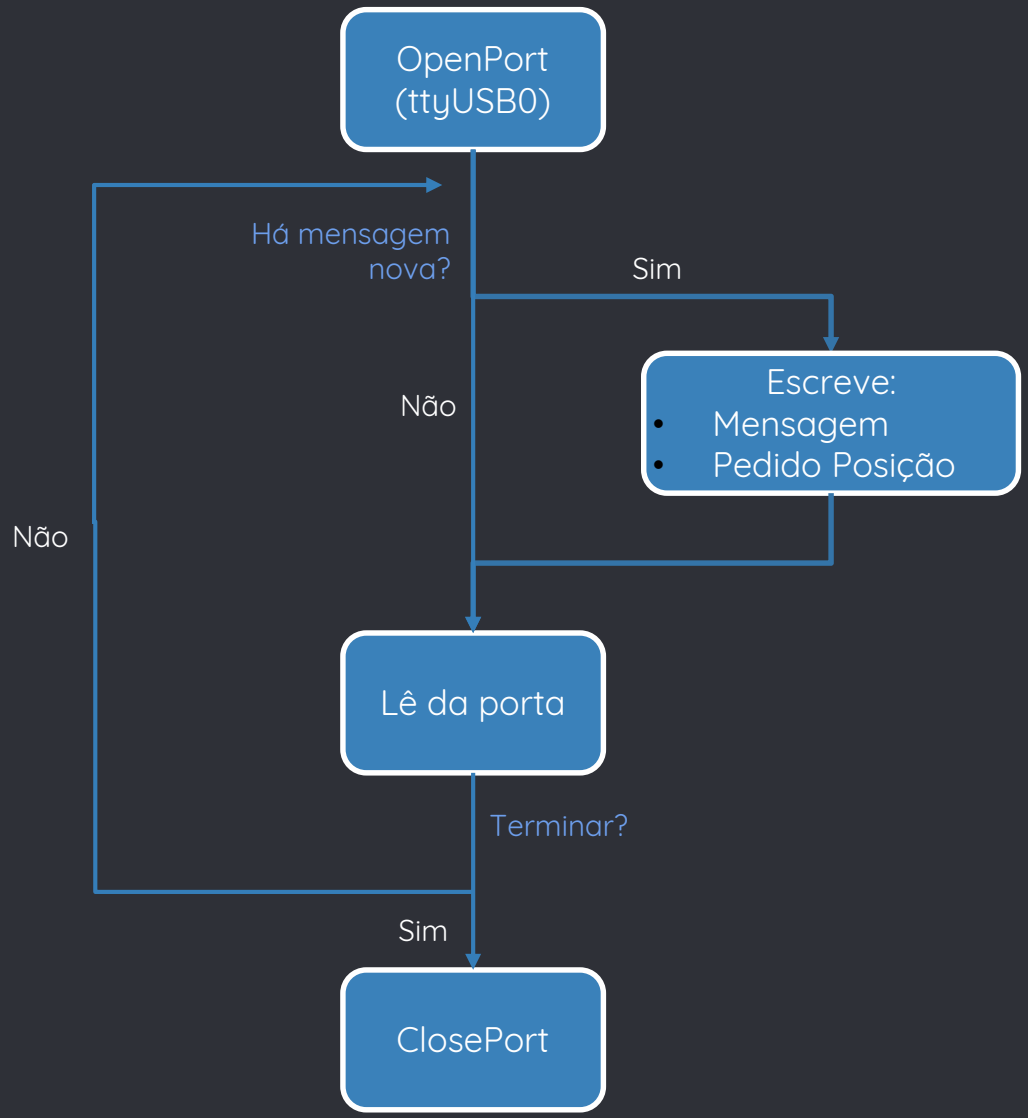
P1 UI

Muda variável / envia sinal
gtk_main_quit()
Remove Shared Memories

P2.1 Xtion


InitOpenNI
Frame escolhido?
Recolher frame
flip e cvtColor
clone frame
Copiar frame para Shared Memory
CloseOpenNI

P2.2 PTU


OpenPort
(ttyUSB0)
Há mensagem
nova?
Sim
Não
Escreve:
Mensagem
Pedido Posição
Não
Lê da porta
Terminar?
Sim
ClosePort

2

Funções

Destaques

- ReadPort
  - lê da porta série e escreve na Shared Memory
- KillTheChild
  - escreve string na Shared Memory

- UpdateUIMsg
  - sscanf() para procurar strings que atualizam valores
- UpdateUIFrame_FromSharedMemory
  - Lê frame (cv::Mat) da Shared Memory
  - Faz uma cópia do pixbuf
  - Memory Leak! g_object_unref()
- UpdateFrameValues
  - Escreve na Shared Memory string: “CM_CF”

- Baseado no openni2_recorder
  - GitHub
- Dividi em funções
- Não apresenta ambos os frames em simultâneo
- Dei a opção de mudar o ColorMap

Demonstração

Conclusões

Objetivos atingidos

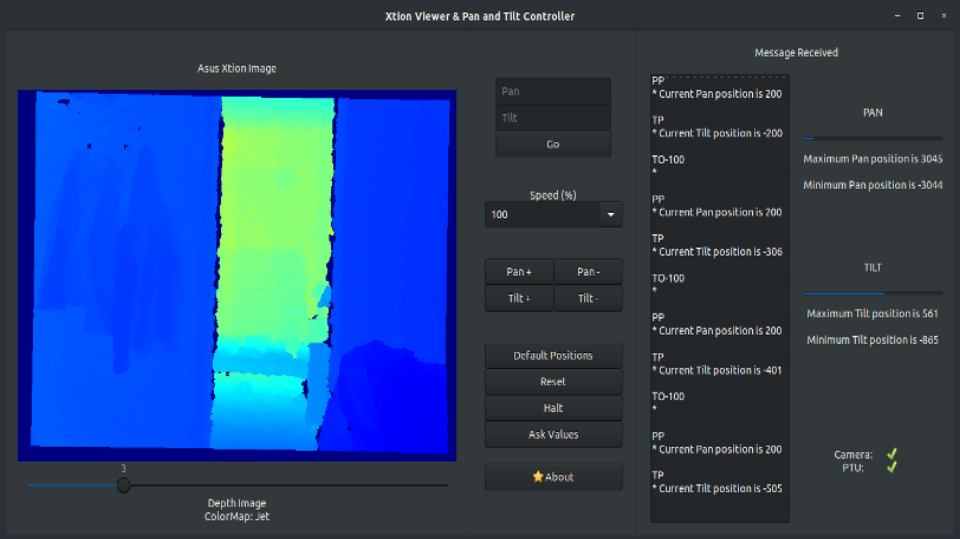
- Visualizar imagem (RGB e Profundidade!)
- Controlar posição absoluta
- Controlar velocidade
- Movimentos incrementais
- Posição pré-definidas

Estes são os iniciais...

Mas há mais!

Objetivos atingidos extra

- Controlar entre RGB ou ColorMap
- Reset/Halt/AskValues à PTU
- Ver mensagens recebidas
- Ver os valores limite da PTU
- Ver o progresso até ao limite
- Informação se os equipamentos forem ou não conectados



Para o futuro

- O User guardar as suas posições
- Remover/conectar USB's durante a execução do programa
- Escolher a porta série
- Guardar frame
- Gravar vídeo

Obrigado!

Questões?