

# Toolkit para Linux Embarcado

Autor: Thiago Waltrik  
Orientador: Miguel Alexandre Wisintainer



# Roteiro

- Introdução
  - Objetivos
- Fundamentação Teórica
  - Trabalhos Correlatos
- Desenvolvimento
  - Requisitos
  - Especificação
- Implementação
  - Técnicas e Ferramentas Utilizadas
  - Operacionalidade da Implementação
- Resultados e Discussão
- Conclusão
- Extensões
- Demonstração da Implementação



# Introdução

- **Sistemas Embarcados**
  - Tecnologia Emergente
  - 32 bits e ARM
  - Linux
  - O que é necessário?



# Introdução

- O que é necessário?
  - Compilador cruzado
  - Dispositivo alvo
  - Bibliotecas C
  - Imagem do *kernel* Linux
  - Sistema de arquivos raiz
  - Estação de trabalho de desenvolvimento
  - Documentação
- *Toolkit* para Linux Embarcado



# Introdução

- **Objetivo geral:**
  - desenvolver um *toolkit* para Linux embarcado que possibilite o desenvolvimento de aplicativos em tempo reduzido utilizando o Mini2440 como dispositivo alvo.



# Introdução

- **Objetivos específicos:**

- disponibilizar documentação passo a passo para a preparação de uma estação de trabalho de desenvolvimento de sistemas com Linux embarcado baseada na distribuição Linux Debian;
- customizar, compilar e disponibilizar uma versão do *kernel* Linux para execução no Mini2440;
- disponibilizar mini-aplicativos que explorem os principais recursos de hardware do Mini2440;
- carregar e executar mini-aplicativos no Mini2440.



# Fundamentação Teórica

- Linux
  - Sistema operacional *unix-like*
  - *Kernel*
  - Distribuição
  - Multiplataforma
- Debian
  - Projeto Debian
  - Nível de segurança
  - Estabilidade
  - Popularidade
  - Comprometimento com o *software* livre



# Fundamentação Teórica

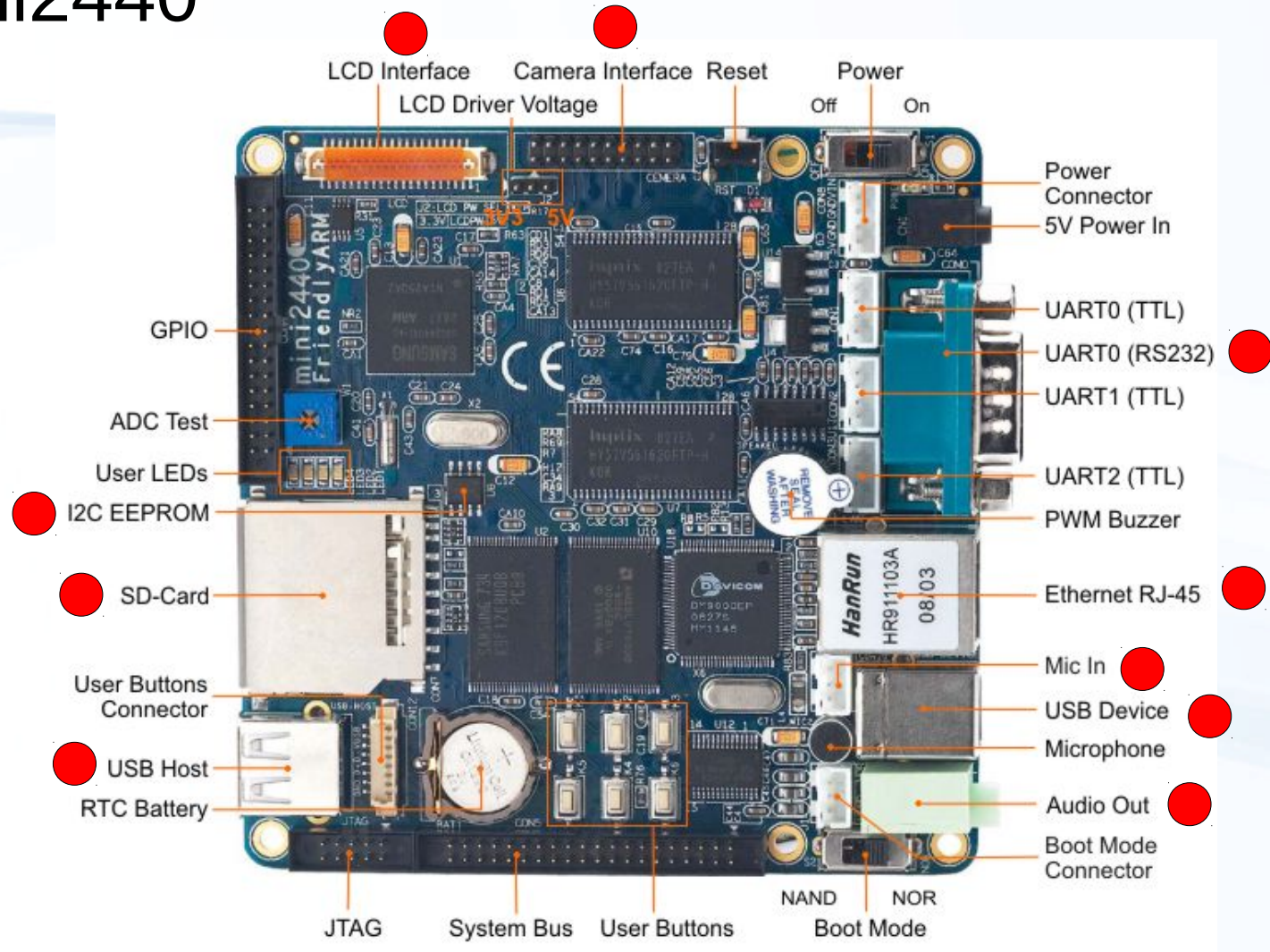
- Arquitetura ARM
  - Derivado da arquitetura RISC
  - Baixo consumo de energia
  - Baixo custo
  - Velocidade de *clock* alta
- Arquitetura RISC
  - Número reduzido de instruções
  - Grande flexibilidade ao *software*





# Fundamentação Teórica

- Mini2440



# Fundamentação Teórica

- Mini2440



# Fundamentação Teórica

- Sistemas Embarcados
  - Sistema embarcado x Computador de uso-geral
- Framework Qt
  - *Framework* gráfico multiplataforma
  - Suporte a C++, Python, Ruby, PHP e .NET
- *Toolkit* para Linux Embarcado
  - Imagem do *kernel* Linux
  - Imagem do sistema de arquivos raiz
  - Bibliotecas
  - *Scripts*
  - *Software* desenvolvido



# Fundamentação Teórica

- Trabalhos correlatos
  - MontaVista Linux
    - Produto
    - *Market Specific Distribution*
    - *Software Development Kit*
    - Suporte
  - OpenEmbedded Project
    - *Framework*
    - Constrói e mantém distribuições
- uClinux
  - Sistema operacional
  - CPU sem MMU



# Desenvolvimento

- Requisitos não-funcionais:

- utilizar sistema operacional Linux Debian para a estação de trabalho de desenvolvimento;
- possuir documentação em língua portuguesa;
- utilizar o dispositivo alvo Mini2440;
- utilizar linguagem de programação C++;
- utilizar linguagem de *script Shell script*;
- criar um sistema de arquivos raiz contendo a distribuição Linux Emdebian;
- configurar um *boot-loader* para carga do *kernel* Linux customizado;
- carregar imagem do *kernel* e sistema de arquivos raiz para o Mini2440;





# Desenvolvimento

- Requisitos funcionais:
  - disponibilizar mini-aplicativos que explorem os seguintes recursos de hardware do Mini2440: relógio de tempo real, entradas e saídas digitais de uso-geral, câmera CMOS, módulo *Wireless*, interface serial RS-232, interface Ethernet, entrada para microfone, saída de áudio estéreo, interface para cartão de memória SD, memória EEPROM I2C e LCD colorido com tela sensível ao toque.



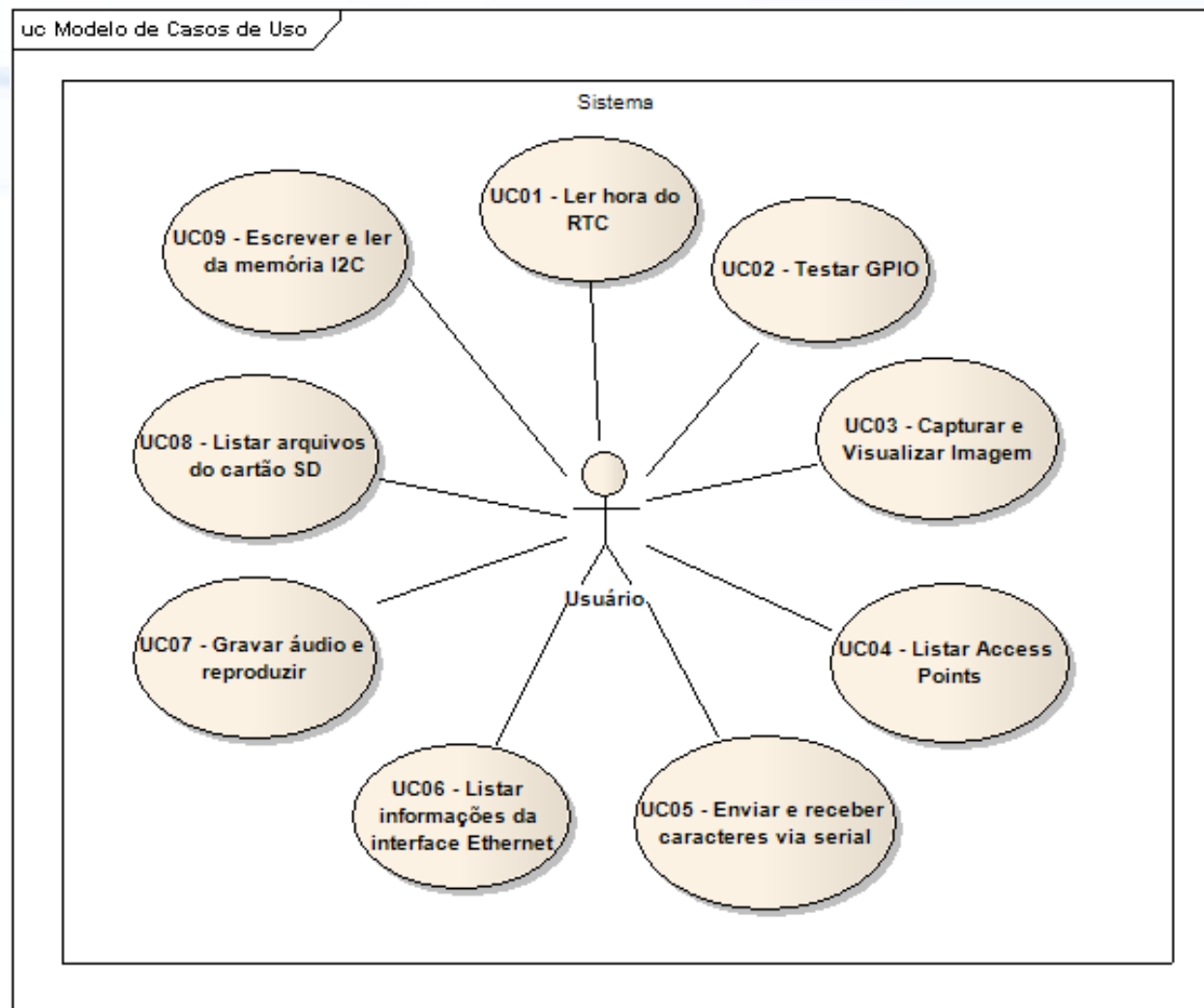
# Desenvolvimento

- Preparação da estação de trabalho de desenvolvimento
  - Configuração básica
  - Instalação do compilador cruzado
- Customização e geração da imagem do *kernel*
  - *Bootup logo*
  - Recursos
- Geração do sistema de arquivos raiz
  - Emdebian



# Desenvolvimento

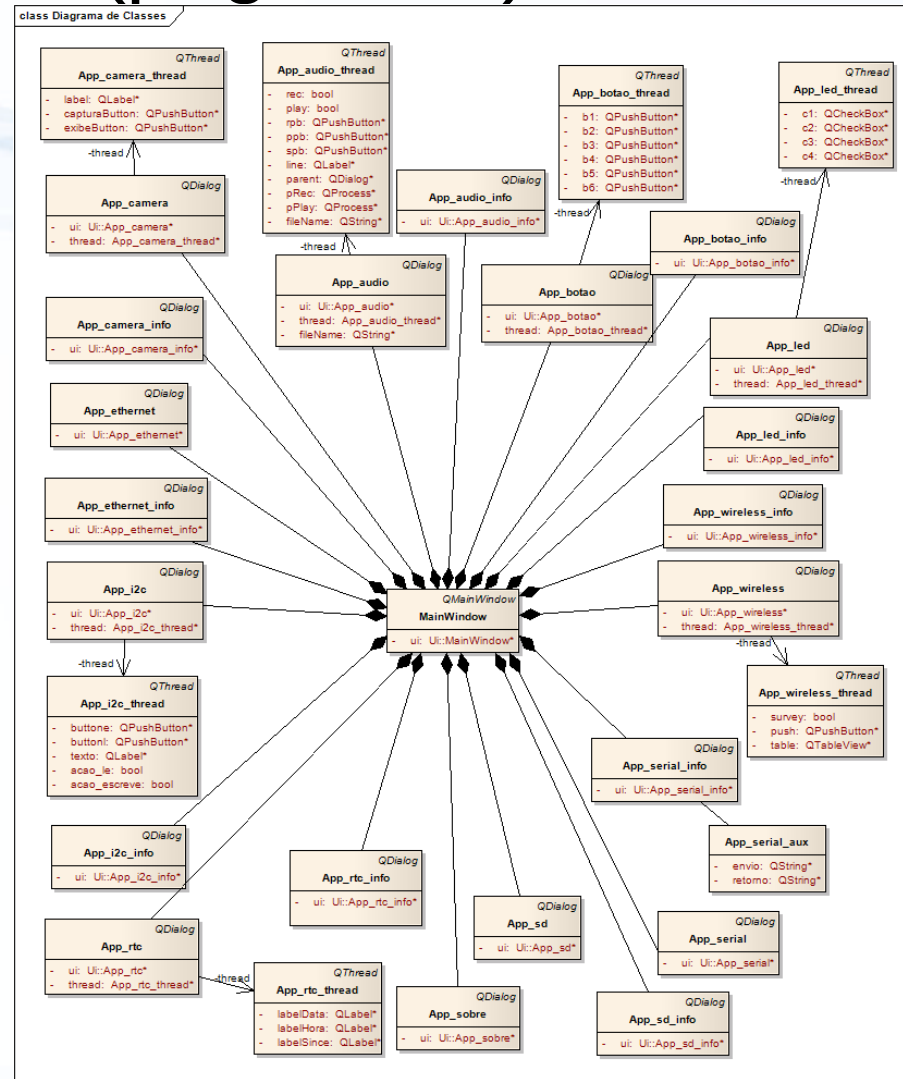
- Especificação



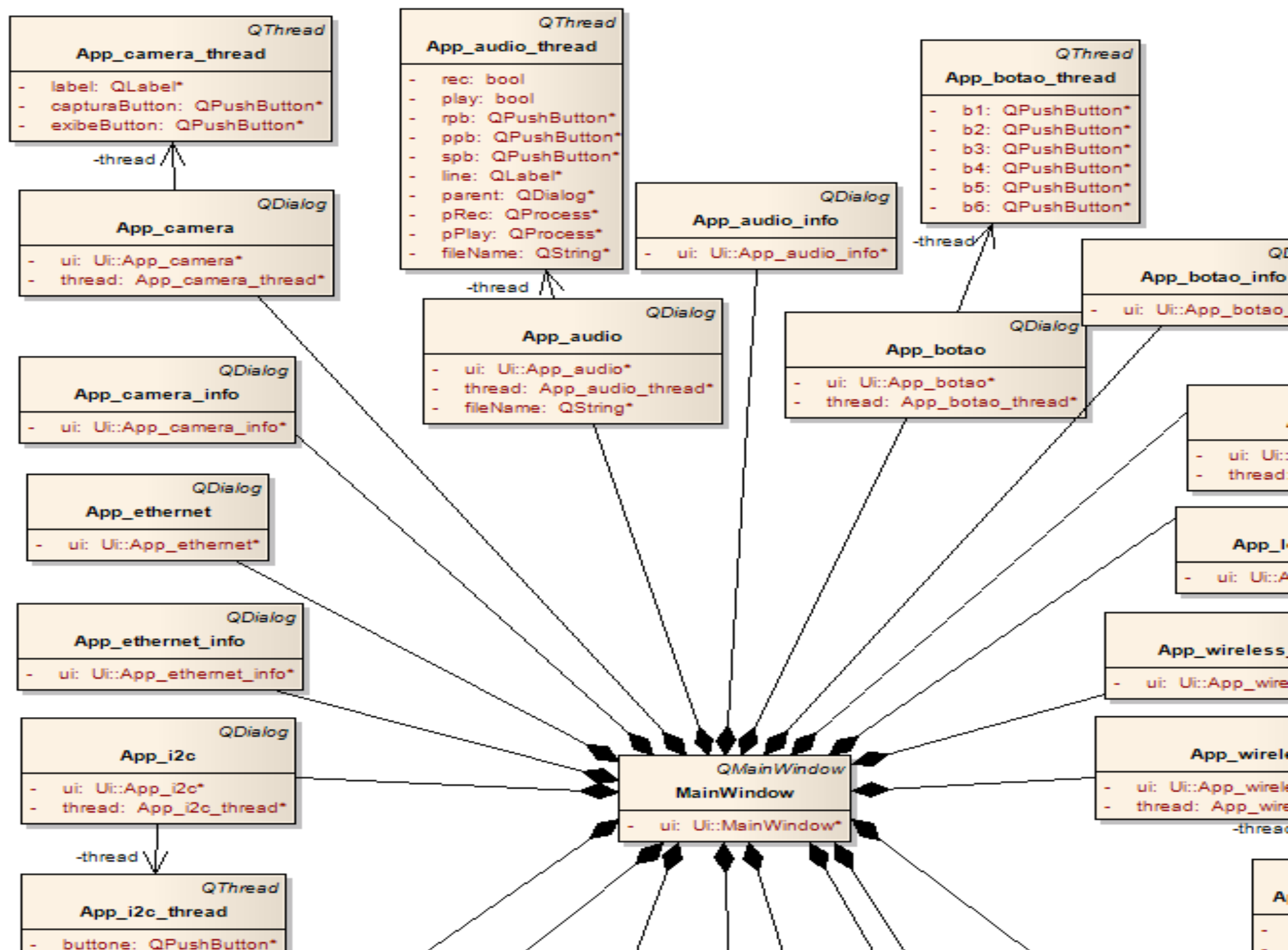


# Desenvolvimento

- Especificação (página 40)



class Diagrama de Classes



# Implementação

- Técnicas e ferramentas utilizadas:
  - Linguagem de programação C++
  - *Scripts* em Shell Script
  - IDE QtCreator



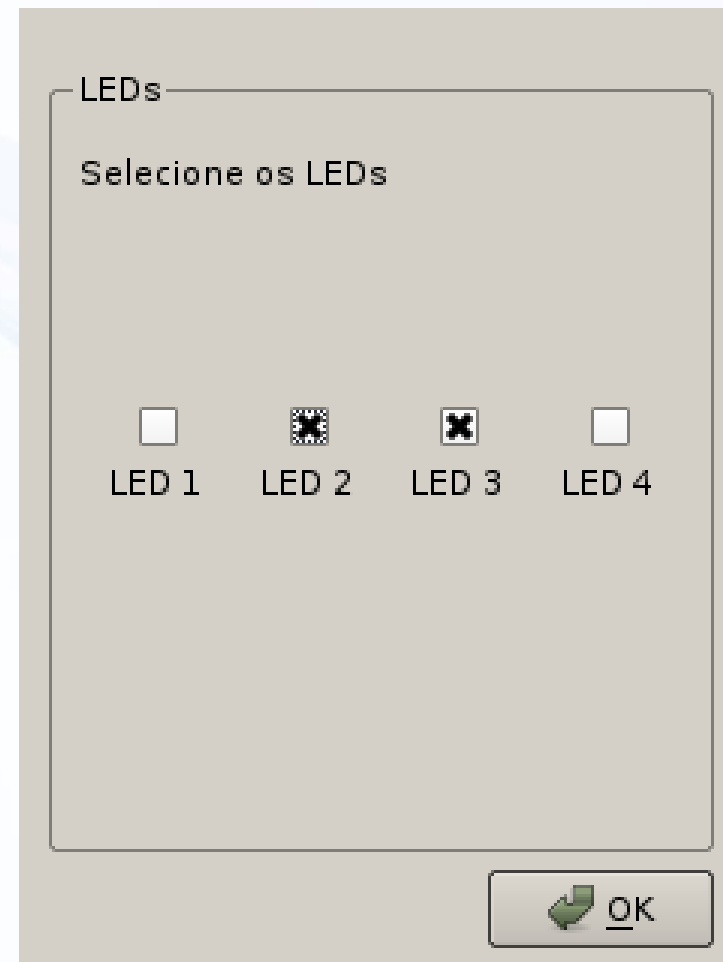
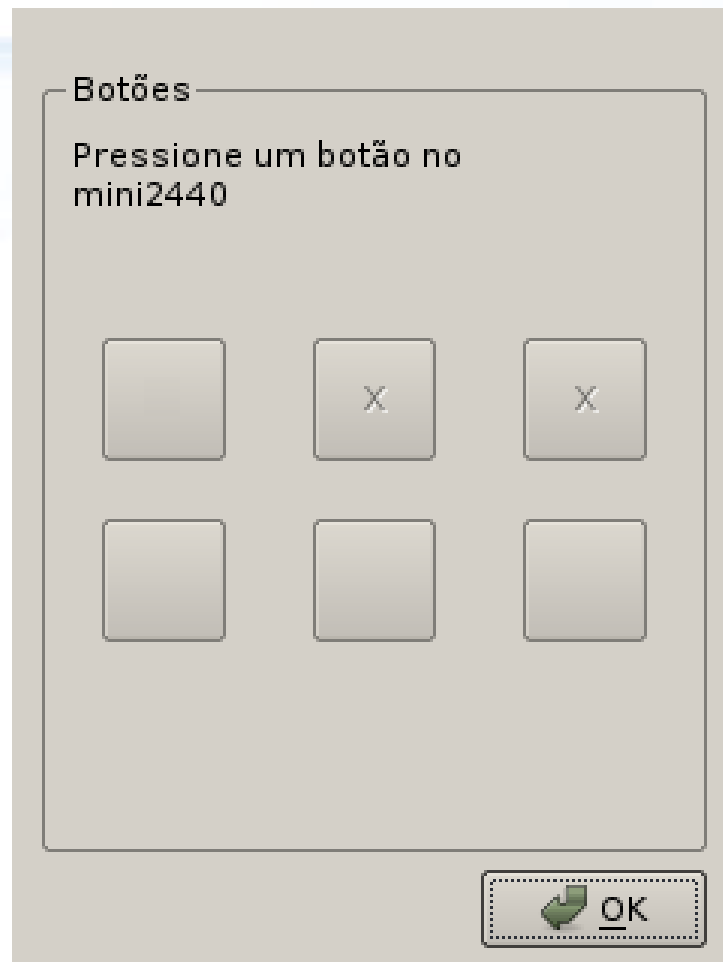
# Implementação

- Operacionalidade da Implementação



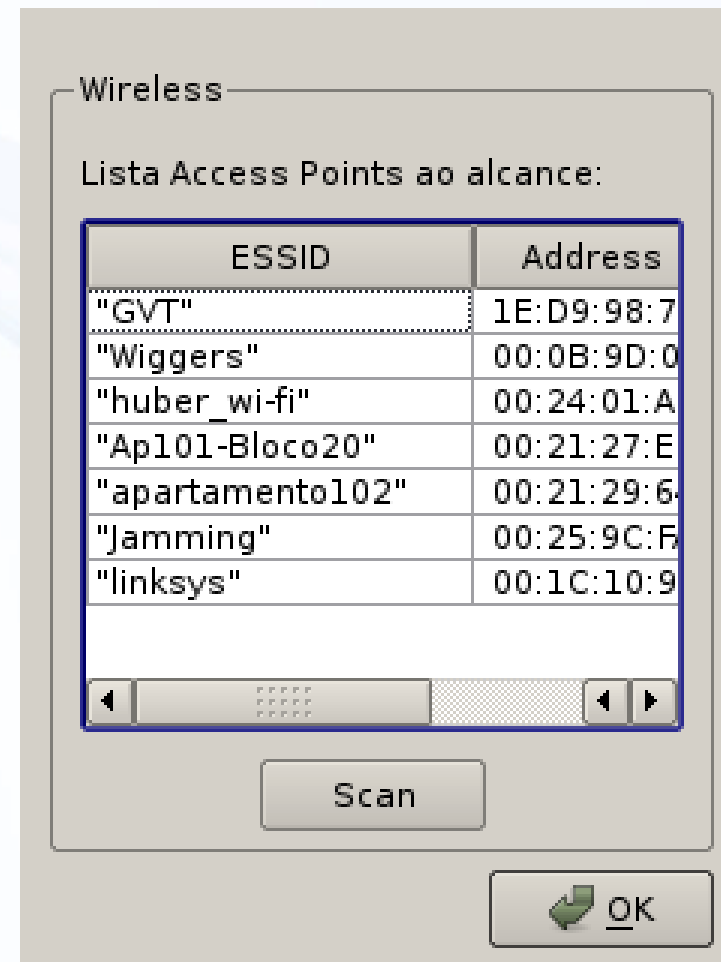
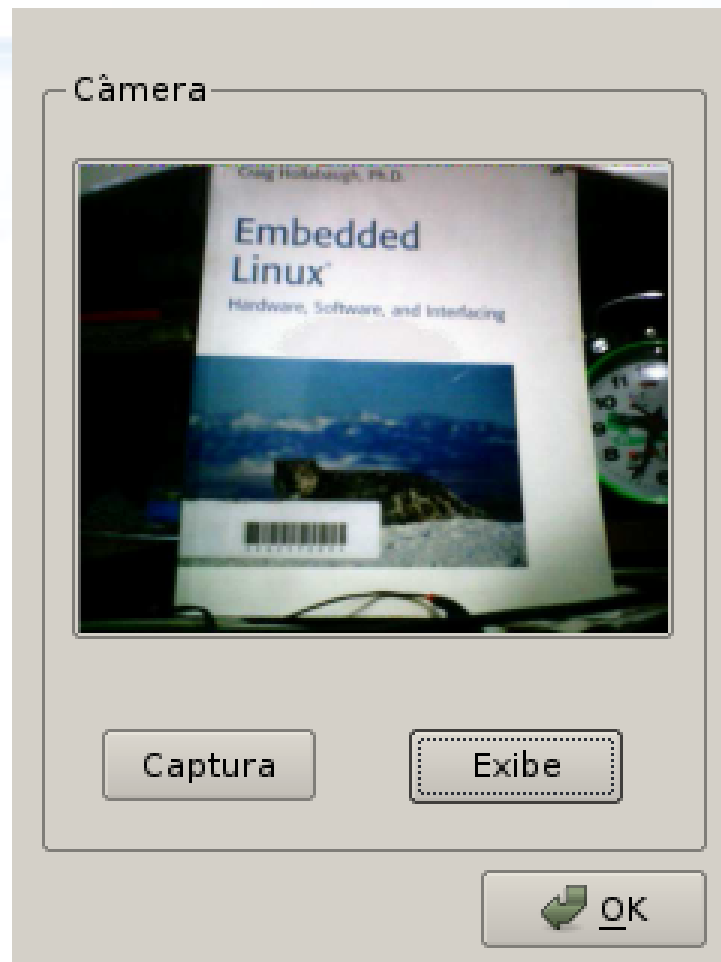
# Implementação

- Operacionalidade da Implementação



# Implementação

- Operacionalidade da Implementação



# Implementação

- Operacionalidade da Implementação

Serial

Porta:

Configurações da porta:  
Velocidade: **115200 bps**  
Paridade: **nao**, Flowcontrol: **nao**,  
Stop bit: **nao**, Databits: **8**  
Buffer de envio:

Buffer recebimento:

Ethernet

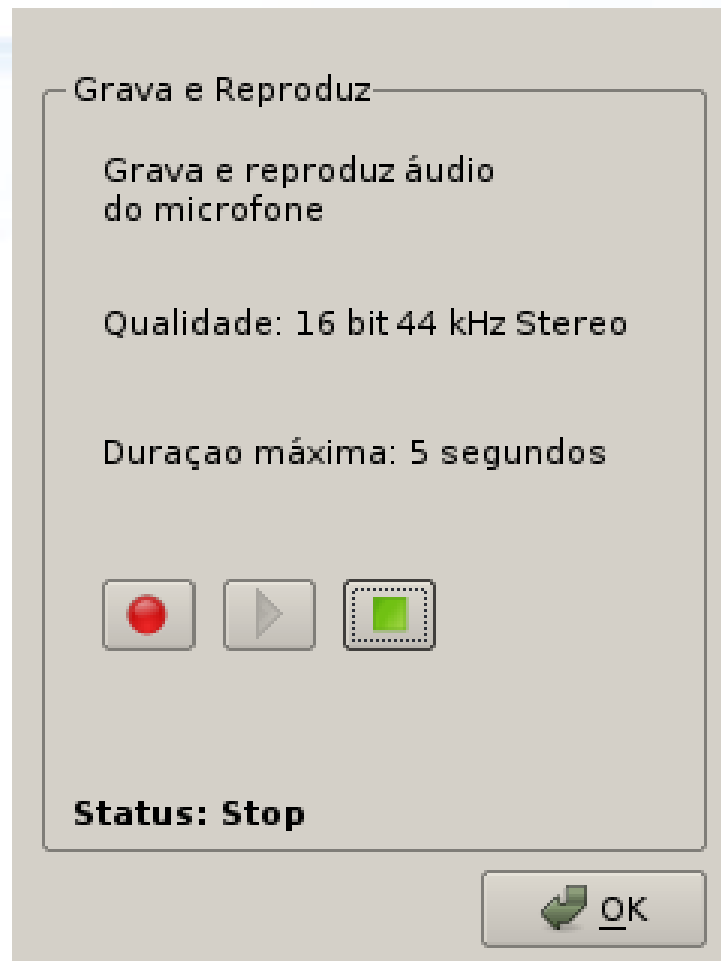
Informações da interface de rede

Interface: eth0  
Modelo: dm9000e  
IP: 10.0.0.2  
MAC: 08:90:90:90:90:90  
Pacotes TX: 160  
Pacotes RX: 244



# Implementação

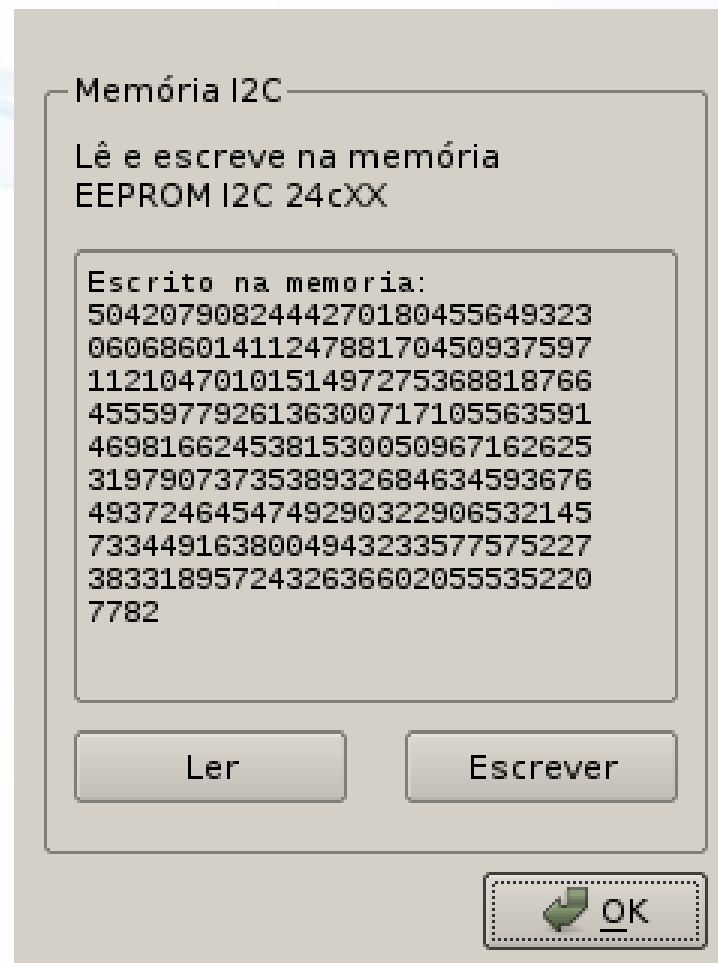
- Operacionalidade da Implementação





# Implementação

- Operacionalidade da Implementação



# Resultados e Discussão

- Disponibilizado:
  - uma imagem do *kernel* Linux customizada;
  - uma imagem do sistema de arquivos raiz contendo a distribuição Linux Emdebian combinada com *scripts* e bibliotecas que permitem a inicialização do Lançador de Aplicativos;
    - mini-aplicativos que exploram os principais recursos de hardware do Mini2440.
- Tempo de inicialização completa: 45 segundos
- Principal limitação do hardware: 64 MB RAM



# Resultados e Discussão

## Comparativo:

Função	Toolkit	MonstaVista	OpenEmbedded	uClinux
Documentação em Língua Portuguesa	X			
<i>Kernel</i> otimizado para dispositivo alvo	X	X		
Aplicativos que exploram recursos de hardware	X			
Suporte a arquitetura ARM	X	X	X	X
IDE		X		
Suporte a processadores sem MMU				X

# Conclusão

- *Framework* Qt: adequado
- Limitações:
  - Latência da interface gráfica
  - Ausência de aplicativos para PWM e USB-Device
  - Geração manual da imagem do *kernel* Linux e do sistema de arquivos raiz



# Conclusão

- Vantagens:
  - Uso de *softwares* livres;
  - Documentação em língua portuguesa;
  - *Framework* multiplataforma;
  - Uso de arquitetura ARM;
  - *Hardware* com mostrador do tipo LCD com *touch-screen*;
  - Código fonte disponibilizado sob a licença *General Public License* (GNU) v3.0;



# Extensões

- Desenvolvimento de um *wizard* capaz de gerar imagens do kernel e do sistema de arquivos raiz;
- Otimização do processo de boot;
- Integração com módulos GPRS/GSM, tal como o TC65i;
- Desenvolvimento de protótipos tais como roteadores, interfaces homem-maquina, *thin clients*, *tablet PCs*, *netbooks* e telefones IP.



# Demonstração da Implementação

