

Controle integrado e distribuído de tomadas inteligentes

Luca Fachini Campelli

Marcio Monteiro

Rodrigo Pedro Marques

Florianópolis, 2016

Resumo

Atualmente, há um alto consumo energético desnecessário que pode ser evitado. Tendo este fato em vista, buscou-se propor uma solução que pudesse amenizar este problema. Com o auxílio de placas EPOSMoteII, foi desenvolvido um algoritmo para um consumo energético mais inteligente para tomadas equipadas com a placa.

Palavras-chaves: tomadas inteligentes, sistemas operacionais, eposmote2.

1 Introdução

Este trabalho foi realizado para a disciplina INE5412 - Sistemas Operacionais I, no semestre 2016.1. O objetivo principal deste trabalho é implementar o descobrimento automático de tomadas em uma rede onde elas possam tomar decisão sobre o consumo energético.

Para a realização deste projeto, foram utilizados três placas EPOSMote II ([LISHA, 2016b](#)) que simularam o comportamento de tomadas inteligentes. Para que isto fosse possível, o projeto foi subdividido em partes. São elas: realizar a comunicação entre as placas via *broadcast*, propor o algoritmo para consumo inteligente de energia, implementação deste algoritmo, validação e testes do algoritmo.

Este relatório está organizado da seguinte forma: em [2](#) são apresentadas as etapas em maior detalhes relacionadas ao desenvolvimento deste projeto; em [3](#) são apresentados os testes realizados após o desenvolvimento; em [4](#) são apresentadas as considerações finais e conclusões em relação a este projeto; finalizando este relatório, na seção [4.1](#) apresentamos possíveis temas para serem discutidos posteriormente.

2 Desenvolvimento

Para que o projeto pudesse ser desenvolvido, primeiramente foi necessário descobrir como era realizada a comunicação entre as placas. Para tal, foram realizados exemplos que constam na documentação do EPOSMote II sobre comunicação entre placas via *broadcast*. Em [A](#) é possível contemplar como foi implementado o código para alcançar este objetivo no projeto. Como é possível observar, foi utilizada a função NIC, já presente na no EPOSMoteII onde ela é responsável por prover acesso à redes.

A figura [1](#) apresenta a modelagem em UML do projeto. A classe *Tomada* representa uma tomada física, inteligente ou não. Neste projeto foram consideradas quatro tipos de tomadas: uma tomada simples, que liga e desliga; uma tomada com *dimer*, que possibilita *dimerizar* (*controlar*) a energia que passa por ela; uma tomada com sensor, onde é possível medir o seu

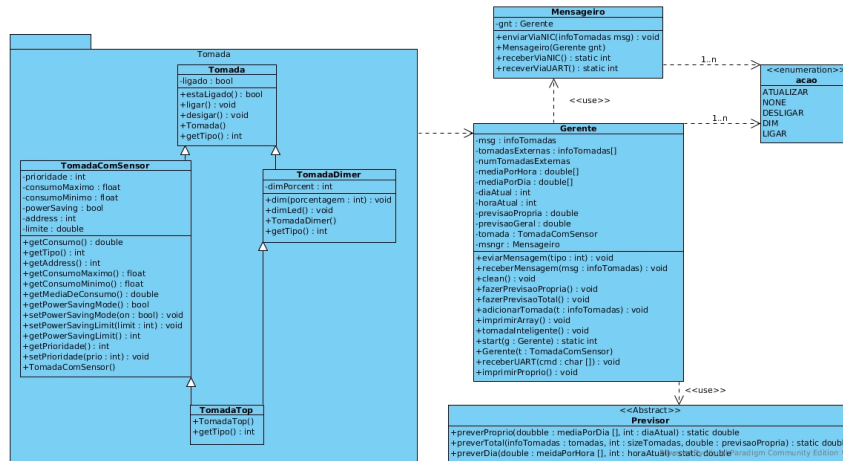


Figura 1 – UML do Projeto.

consumo, colocar em modo de economia de energia, atribuir um limite de consumo, atribuir uma prioridade a ela e verificar o seu consumo mínimo e máximo já registrado; e uma tomada *top* que possui todas as funções das tomadas citadas anteriormente. Cada tomada possui um *Gerente Monitor* responsável por monitorar e atualizar a tomada, além de enviar e receber mensagens de outras tomadas no ambiente. Além disto, ele mantém o endereço de todas as tomadas no ambiente, guarda informações (previsão do consumo de energia, endereço da tomada, prioridade e tipo) da tomada monitorada por ele e também possui um *Mensageiro* e um *Previsor*. A classe *Mensageiro* é responsável por receber mensagens via NIC ou UART. O *Previsor* auxilia o gerente realizando as previsões de consumo da tomada gerenciada por ele. Esta classe pode fazer as previsões de consumo diárias, mensal e de todas as tomadas do ambiente.

3 Validação e Testes

Os testes e validação se deram por algumas alterações no código e observação do comportamento do ambiente. Na figura 2 é possível observar uma tela geral do comportamento das tomadas. Elas ficam conversando entre si atualizando o seu consumo e previsão de consumo geral.

Inicialmente, simulamos um ambiente com duas tomadas onde uma tomada diz que não irá tomar uma ação. Para isto, ela envia a mensagem "NONE" para as outras. Como as outras tomadas tem prioridade maior àquela que enviou a mensagem NONE, elas são obrigadas a tomar uma ação. A imagem 3 mostra este funcionamento. Vale ressaltar que o *print* NONE foi enviado pela tomada "tomada minha".

Outro teste que foi realizado foi na questão de ligar e desligar quando possível ou passa da previsão de consumo permitida. Por exemplo, na imagem 3 é possível observar a "tomada 0" previu que a previsão geral ia ser acima 47760, então ela mandou uma mensagem "DESLIGAR", informando assim que ela irá desligar no próximo ciclo. Enquanto a tomada está desligada e a previsão geral mais o consumo máximo da tomada desligada é menor que a permitida, então ela liga.

4 Considerações Finais e Trabalhos Futuros

Neste projeto foi desenvolvido um algoritmo de consumo energético inteligente direcionado ao EPOSMoteII. Foram realizados diversos testes a fim de deixá-lo o mais aprimorado possível. Na seção 3 é possível observar alguns exemplos de funcionamento do mesmo.

```

previsao: 18990
prioridade: 0
previsao geral: 75321
#####
tomada minha
endereço: 1
consumo: 572
previsao: 24250
prioridade: 10
PWSL: 45000
#####
tomada 0
endereço: 2
previsao: 26510
prioridade: 0
#####
tomada 1
endereço: 4
previsao: 18990
prioridade: 0
previsao geral: 76210
#####
tomada minha
endereço: 1
consumo: 672
previsao: 23764
prioridade: 10
PWSL: 45000

```

Figura 2 – Tela Geral das Tomadas Funcionando.

Durante o desenvolvimento deste projeto, percebemos que a área da computação pode ir muito além do que apenas um computador pessoal, de mesa ou *mainframes*. É possível aplicá-la em diversas áreas desde empresas, até itens mais comuns como tomadas ou lâmpadas possíveis de encontrar em casa. Este projeto nos ajudou a refletir sobre o impacto que ele pode causar em questão de consumo energético. Por exemplo, dentro de uma casa, seria possível desligar todas as tomadas do ambiente, deixando apenas as essenciais ligadas (a da geladeira por exemplo), economizando energia e evitando possíveis sobrecargas elétrica. Isto aplicado em uma empresa grande, o impacto disto poderá ser ainda maior.

Este trabalho é apenas um estudo inicial de um tópico atualmente na área da computação que ainda não existe um foco muito grande. Considerando que o gasto energético ainda é um problema global, este trabalho visa sugerir possíveis soluções para diminuição do gasto energético não somente a nível residencial, mas também em nível de indústria que é umas das principais responsáveis pelo grande consumo energético atualmente.

4.1 Trabalhos Futuros

Como trabalhos futuros pode-se apontar:

- Desenvolvimento de um aplicativo para *smartphones* que possibilite a configuração das tomadas.
- Através do aplicativo citado anteriormente, fazer com que a tomada reconheça o usuário

```

consumo: 1160
previsao: 23460
prioridade: 10
PWSL: 45000
#####
tomada 0
endereco: 4
previsao: 25350
prioridade: 0
previsao geral: 47760
#####
tomada 0
endereco: 4
previsao: 27240
prioridade: 0
previsao geral: 48810
DESLIGAR
#####
tomada minha
endereco: 1
consumo: 0
previsao: 17595
prioridade: 10
PWSL: 45000
#####
tomada 0
endereco: 4
previsao: 26910
prioridade: 0
previsao geral: 44835
NONE
#####
tomada minha
endereco: 1
consumo: 0
previsao: 14076
prioridade: 10
PWSL: 45000
#####
tomada 0
endereco: 4
previsao: 28225
prioridade: 0
previsao geral: 40986
LIGAR
#####
tomada 0

```

Figura 3 – Tela de Demonstração de Ação Liga/Desliga.

e se autoconfigure para as configurações pré definidas pelo usuário.

- Integrar o algoritmo de consumo inteligente em luminárias e outros itens inteligentes.
- Através do aplicativo citado no primeiro item, fazer um reconhecimento de proximidade, onde os itens que se encontram na mesma sala que o usuário permaneçam ligados de acordo com o perfil configurado, e os itens fora dessa se desliguem (de acordo com a prioridade) a fim de economizar energia.
- Análise de um ambiente sem tomadas inteligentes e outro apenas com tomadas inteligentes com este algoritmo aplicado.

Referências

LISHA. *LISHA*. 2016. Acesso em 20 de jun de 2016. Disponível em: <<https://lisha.ufsc.br/HomePage>>. Nenhuma citação no texto.

LISHA. *Welcome to the EPOS Project*. 2016. Acesso em 20 de jun de 2016. Disponível em: <<https://epos.lisha.ufsc.br/HomePage>>. Citado na página 1.

APÊNDICE A – Implementacao do Recebimento de Mensagens

```
int Mensageiro::receberViaNIC(){
    NIC nic;
    NIC::Address src;
    unsigned char prot;
    infoTomadas meg;

    while(1){
        while(!nic.receive(&src, &prot,&meg,sizeof(meg)) >0);
        if(meg.address != 0){
            gerente->receberMensagem(meg);
        }
        meg.address = 0;
    }
    return 0;
}

Mensageiro::Mensageiro(Gerente * gnt){
    gerente = gnt;
    Thread *thread;
    thread = new Thread( \&receberViaNIC );
}
```
