UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

RELATÓRIO FINAL

PROJETO PROPOSTO DE INICIAÇÃO CIENTIFICA:

AUTOMAÇÃO USANDO SISTEMAS OPERACIONAIS LINUX EMBARCADOS EM MICROCONTOLADORES ARM.

ALUNO:
LEONARDO BRÁS SOARES PASSOS

ORIENTADOR:
PROF. DR. EVANDRO LUIS LINHARI RODRIGUES

São Carlos, Julho de 2011

INTRODUÇÃO

A arquitetura ARM é hoje vastamente aplicada em dispositivos portáteis, desde relógios, celulares até netbooks. Constatada essa flexibilidade, utilizando-se de conceitos básicos [1] e apoiada em trabalhos anteriormente realizados em laboratório [2], a proposta deste projeto objetiva a criação de uma plataforma de automação simples e robusta. Em princípio, para focalizar a dedicação inicial, os trabalhos foram desenvolvidos vislumbrando a automação de processos residenciais, como iluminação inteligente, sistemas e câmeras de segurança, e dotando-a principalmente de operabilidade via redes locais de computadores e internet. Para isso, utilizou-se um sistema operacional compacto, com características de robustez e flexibilidade. O Sistema Operacional Linux além de possuir essas características tem a vantagem de ser *opensource* e possuir vasto suporte.

OBJETIVOS

- Desenvolver habilidades na utilização do Sistema operacional GNU/Linux para plataformas microcontroladas.
- Dominar os procedimentos de minimização de sistemas operacionais Linux para sistemas embarcados.
- Aplicação de conceitos de Redes de Computadores e interface baseado em WEB para a automação de processos.
- Aprendizado de arquitetura ARM.

METODOLOGIA:

Utilizou-se o kit de desenvolvimento SAM-L9260 da OLIMEX (www.olimex.com), o qual conta com um microcontrolador ARM9 de 32 bits, para a instalação do sistema operacional Debian mais recente (atualmente a versão 6.0, codinome “Squeeze”). Foram exploradas ferramentas minimizadas como Busybox e outros aplicativos que visam minimização do sistema operacional, todos cobertos pela licença GPL.

Foi implementado o sistema de desenvolvimento e utilizou-se do GCC para compilar uma nova versão do kernel Linux da arquitetura ARM, que foi montada de maneira a se tornar uma revisão reduzida e otimizada para a aplicação, e esta foi instalada na placa do Kit de desenvolvimento SAM-L9260.

Uma vez instalados os aplicativos básicos para o funcionamento do sistema, foi desenvolvida uma interface web que permitiu desenvolver as experiências de operação via intranet. Testes práticos de implementação foram desenvolvidos para tarefas residenciais automatizadas [3], e ultrapassada essa etapa foi realizado um estudo de aplicabilidade global da plataforma. Um objetivo agregado durante o desenvolvimento do trabalho foi a realização da comunicação com os periféricos através do protocolo SPI e utilizando uma antena que realiza a transferência de dados via radiofrequência. O módulo que contém a antena (MOD-NRF24L) é do mesmo fabricante do Kit.

RESULTADOS :

Inicialmente fez-se, utilizando a interface serial, um estudo prático de utilização do sistema operacional Debian GNU/Linux Codinome "*Etch*", pré instalado no kit de desenvolvimento SAM-L9260, no qual se pesquisaram as funcionalidades do sistema e os métodos de transferência de dados previamente configurados no kit. Dentre essas funcionalidades a que mais se mostrou útil inicialmente foi a ferramenta APT.

APT é a sigla para *Advanced Packaging Tool*, que do Inglês significa Ferramenta de Empacotamento avançado. Ela é utilizada para adquirir aplicativos de software a partir dos repositórios Debian, desde pacotes binários específicos para a arquitetura ARM até códigos fonte que podem ser compilados automaticamente após o download. Entretanto, como se trata de um sistema embarcado, com baixo poder de processamento, a opção utilizada foi a aquisição de pacotes binários.

A partir da interface serial configurou-se a rede *ethernet e* instalou-se, utilizando-se o APT, a ferramenta OpenSSH, que permite o acesso via rede ao Sistema Operacional equipamento. A partir deste passo, todos os acessos foram feitos utilizando-se desta facilidade.

Uma vez estabelecida essa etapa, procedeu-se com a atualização do Debian da versão 4.0 (Codinome "*Etch*") para a versão 5.0 (Codinome "Lenny"). O motivo de tal atualização baseia-se nos fatos de que a versão 4.0, que viera pré instalada no kit de desenvolvimento utilizado, não receberia mais correções de segurança que pudessem vir a afetar o sistema operacional, uma vez que a versão mais nova do Debian encontrava-se disponível há quase dois anos.

Como parte da proposta consiste em desenvolver uma interface web para finalidade de automação, fez-se necessária a instalação de um software servidor web. Com essa finalidade encontrou-se o bem sucedido Apache, que além de ser o mais popular servidor web do mercado, também é licenciado como software livre.

Para que haja meio de, utilizando a interface web, passar parâmetros para e execução de um aplicativo no servidor, é necessário um módulo no servidor web que interprete uma linguagem que tenha este fim. Nesse caso, o Apache oferece suporte a duas dessas linguagens, o PHP e o CGI.

Como o CGI veio caindo em desuso nos últimos anos por causa de quebras de segurança em sistemas nele baseados, é fortemente recomendado o uso do PHP, que ainda apresenta uma robustez aceitável, um desenvolvimento constante do módulo apache, e uma comunidade de usuários ativa.

Como é objetivo que a automação possa ser realizada remotamente, desde que se tenha acesso à internet, é necessário precaver-se contra possíveis interceptações de conexão que possam acontecer. Para isso, foi instalado um módulo Apache para comunicação criptografada SSL, como também a aplicação OpenSSL, que permitem uma comunicação segura entre usuário e servidor.

Uma vez garantindo uma conexão segura de dados, foi construído em PHP um pequeno rascunho de interface Web, contando com proteção de seção a partir de usuário e senha, e algumas atividades básicas que exemplificavam o uso em automação.

Para facilitar o desenvolvimento da interface Web, foi instalado um Sistema de Gerenciamento de Conteúdo (CMS), que dispensa o desenvolvimento de vários componentes da interface Web planejada. Infelizmente, isso não se mostrou possível, pois o CMS necessita de um Sistema de Gerenciamento de Banco de Dados (SGBD) como o MySQL, e o sistema embarcado utilizado não comporta uma aplicação que exija tantos recursos de hardware.

Uma vez que o andamento da parte de software que controlava a interface Web encontrava-se bem adiantada, foi proposto que se iniciasse a etapa de compilação de

uma nova versão do Kernel Linux. Para isso utilizou-se do código fonte disponível no site oficial do Kernel Linux [5], e arquivos *.patch* disponíveis no site oficial do fabricante do kit [6].

A utilidade dos arquivos *.patch* é carregar pequenas modificações no codigo fonte do Kernel Linux, que foram feitas visando o melhor funcionamento do mesmo no kit utilizado. Para aplicar estes arquivos, é utilizado o comando patch que substitui facilmente as modificações propostas.

Uma vez com o código pronto e otimizado utilizou-se uma ferramenta que compila e organiza o código para a arquitetura ARM, utilizando uma maquina de arquitetura x86 com maior poder de processamento para fazer o serviço pesado. A essa ferramenta atribui-se o nome de toolchain. A toolchain utilizada no caso chama-se Codesourcery G++ lite, que se trata de um conjunto de editor, compilador e *linker, todos opensource, que possibilitam a compilação cruzada (cross-compiling) de x86 para ARM.*

O executável do Kernel resultante da compilação bem sucedida foi transferido ao KIT e foi executado, onde foram observadas algumas falhas. Em busca da correção dessas falhas, consideramos atualizar o sistema operacional Debian, o que desencadeou a próxima parte do projeto.

Não seria possível atualizar o sistema operacional Debian para a versão mais nova pelo fato de que houve mudança no porte da arquitetura ARM padrão do Debian. O porte dos aplicativos e kernel que vieram com o kit é o porte "*little-endian ARM" (arm)* e este porte foi descontinuado pelo projeto Debian. O novo porte utilizando pelo projeto Debian é o ARM-EABI (armel), e portanto seria necessário refazer o kernel e instalar manualmente o sistema operacional, o que fugiria do propósito pretendido.

Felizmente, após alguns e-mails trocados, foi possível que o fabricante fornecesse o sistema operacional Debian já utilizando o porte da *armel.* Procedeu-se então repetindo o trabalho realizado anteriormente, atualizando o sistema fornecido para o então novo lançamento do Debian, versão 6.0 codinome "Squeeze", instalando o SSH e servidor WEB.

Após várias pesquisas, foi descoberto um método de rodar o SGBD e o CMS em apenas 64MB de memória, que é o disponível no kit. Para tal utiliza-se o servidor web

lighttpd, que como o nome sugere tem um consumo de recursos bastante reduzido, o módulo PHP conhecido como FastCGI e o SGBD MySQL, todos com uma configuração minimizada. A mesma pode ser feita editando os respectivos arquivos de configuração.

Aplicou-se então esse método e foi instalado o CMS Drupal 6, que se facilitou a criação de boa parte da interface WEB desejada.

A próxima etapa foi compilar o kernel Linux com suporte a SPI, para que pudesse ser feita a comunicação entre o Kit e os periféricos a serem controlados. Várias tentativas foram feitas, utilizando-se códigos fonte de várias versões do Linux, e por fim decidiu-se por usar o novo código fonte fornecido pelo fabricante juntamente com a versão *armel* do Debian. Como a versão fornecida pela Olimex (fabricante do kit) não suporta comunicação via SPI foi necessária a alteração do código fonte, mudando algumas nomenclaturas.

A partir do código corrigido, foram compilados um kernel modular (onde os módulos são complementos que podem ser carregados ou descarregados quando necessário) e um kernel monolítico (todas as funções são internas ao kernel e são carregadas junto ao *boot*). Por características de gravação do kit, foi necessário criar uma cópia do sistema de arquivos (rootfs) para o sistema modular, no qual os módulos foram adicionados para que possam ser carregados quando necessários.

Foi compilada a funcionalidade SPI utilizando-se a opção SPIDEV, e esta disponibiliza o SPI para uso de usuário através do diretório de dispositivo (/dev), o que facilita em muito sua utilização. Usando o dispositivo /dev/spidev1.0, foi testado então o SPI utilizando-se um osciloscópio e um programa em C que envia dados para um periférico (antena), comprovando-se assim seu funcionamento.

Pelo fato de os códigos modificados, arquivos e binários obtidos nesse trabalho se mostrarem necessários para futuros trabalhos na plataforma, foi criado um servidor FTP no qual os mesmos foram disponibilizados. O objetivo desse servidor é também servir de suporte para que os futuros trabalhos realizados nesse mesmo escopo sejam reunidos mais facilmente.

O endereço atual deste servidor é 143.107.235.37 e pode ser acessado livremente

via internet.

ANÁLISE

A proposta de instalação de um Sistema de Gerenciamento de Conteúdo facilita boa parte do desenvolvimento do projeto, entretanto, pelo fato da necessidade de grande quantidade de memória, o SGBD MySQL apresenta instabilidade, sendo descarregado em algumas situações e impedindo o uso do CMS. O mesmo deve acontecer por falta de memória disponível e pelo fato do processo apresentar baixo uso.

Isso foi útil para compreender tanto que as plataformas ARM embarcadas geralmente trabalham com recursos limitados, quanto que essa proposta pode se tornar muito interessante quando houver recursos mais abundantes. Além disso, foi útil tanto para aprender o funcionamento do escalonador de tarefas do Linux quanto para aprender como ajustar os arquivos de configurações dos aplicativos para que os mesmos se ajustem às necessidades do projeto.

Antes de se obter o código fonte atualizado pelo fabricante, para obtenção de um executável do Kernel, foram feitas várias tentativas com várias versões de Kernel e vários arquivos .patch, e mesmo com todos estes processos, o melhor que se obteve foi um Kernel que não possuía suporte a rede ethernet. Apenas conseguiu-se executáveis do Kernel com todas as funções desejadas a partir do código fornecido pelo fabricante. Isso provavelmente se deve ao fato de que o fabricante do kit não utilizava versões estáveis de aplicativos e Kernel para criar os arquivos *.patch,* mas sim versões *snapshot e release candidate (rc).*

O problema de não se utilizar versões estáveis de software para criar arquivos *.patch* é que o código fonte de versões não estáveis é volátil, ou seja, não fica armazenado no repositórios do projeto. Dessa maneira, houve alterações dos códigos entre as versões não estáveis, às quais os aquivos *.patch* foram fornecidos, e as versões estáveis, as quais se tem acesso ao código fonte. Ressalta-se que este problema só foi sanado porque o fabricante do Kit disponibilizou o código fonte das versões para as quais os arquivos *.patch* foram criados.

CONCLUSÕES

Plataformas microcontroladas rodando o sistema operacional Debian possuem pleno potencial de utilização para automação de processos e um custo de projeto bastante reduzido.

Caso seja necessária a utilização de um SGBD, como no caso do CMS é altamente recomendável a aquisição de sistemas microcontrolados com maior disponibilidade de memória, pois apesar da solução encontrada ser funcional, o funcionamento apresentou comportamento razoavelmente instável.

O kit de desenvolvimento facilita em muito o desenvolvimento do projeto, portanto recomenda-se ainda que durante a escolha do kit seja considerado o uso de software “estável”, pois isso pode representar uma drástica redução no tempo ou custo do projeto.

Utilizando-se de algum tempo disponível para aprendizado, foi possível contornar os problemas encontrados na compilação do Kernel e gerar duas versões de executável do Kernel Linux que possuísse todas as funcionalidades necessárias para o cumprimento do projeto.

Uma vez contornadas as dificuldades encontradas, a montagem da interface entre o Kit e os equipamentos a serem automatizados é simples e exige apenas algum tempo e conhecimentos de programação em C.

REFERÊNCIAS

[1] COOPERSTEIN, Jerry. Introduction to Embedded Linux. [online] Disponível na Internet via WWW. URL: http://training.linuxfoundation.org. 20/04/2010.

[2] AMÂNCIO, I. F.; RODRIGUES, E. L. L. Implementação de Linux embarcado como alternativa para sistemas operacionais de tempo real proprietários.

[3] MARIN, P. S. Automação Residencial: visão geral e aplicações. [online] Disponível na Internet via WWW. URL: http://www.paulomarin.com/Files/home_automation_article.pdf. 20/04/2010.

[4] MAZIOLI, Gleydson. Guia Foca GNU/Linux. [online] Disponível na Internet via WWW.

URL: http://www.guiafoca.org. 23/01/2011.

[5] Linux Kernel Repository. [online] Disponível na Internet via WWW. URL: http://www.kernel.org. 24/01/2011.

[6] Olimex SAM9-L9260 Development Board Website. [online] Disponível na Internet via WWW. URL: http://www.olimex.com/dev/sam9-L9260.html. 22/01/2011.