



PUC
RIO

- 1.
- 2.
- 3.

Marcelo Politzer Couto
Raul Almeida Nunes
Dado Sutter



**Conselho Nacional de Desenvolvimento
Científico e Tecnológico**

1. Aluno de Engenharia de controle e Automação
2. Prof. Dr. do DCMM – Orientador dos projetos
3. Pesquisador Visitante da PUC-Rio – Co-Orientador dos projetos

- 1 - Introdução
- 2 - Objetivos
- 3 - Metodologia
- 4 - Desenvolvimento
- 5 - eLua
- 6 - Projetos Desenvolvidos

1 - Introdução:

O trabalho desenvolvido sob a vigência da bolsa de iniciação científica PIBIC visa o desenvolvimento de projetos usando eletrônica embarcada (embedded). Ao contrario de um computador, estes projetos se destinam geralmente a funções específicas, dedicados a uma ou poucas tarefas. Essa natureza minimalista geralmente não justifica o uso de um computador completo por desperdício de energia, tamanho e custo. Por esses motivos são usados em seu lugar microcontroladores (MCU – Micro-Controller Unit), que são pequenas centrais de processamento com memória e periféricos embutidos em um único chip. Por sua menor escala atende os requerimentos acima. Cabe ao projetista de um sistema embarcado dimensionar qual microcontrolador é mais adequado para as necessidades do seu projeto, dado que existem assim como computadores, microcontroladores mais e menos potentes. Exemplos de tais sistemas são: caixas eletrônicas, forno de micro-ondas, sistema elétrico dos carros, roteadores para redes etc.

Foram estudadas duas famílias de microcontroladores no laboratório GIGA, onde a iniciação ocorreu. Um (AVR[1]) para tarefas de menor processamento: Controle e sensoriamento baseado em MCUs "satélites" distribuído. E um para tarefas mais complexas (ARM[2]), no qual foram desenvolvidos: O refinamento do robô autônomo com navegação por gps, extensões da linguagem Lua para controle (drivers) de displays LCD, desenvolvimento de módulos eLua (Embedded Lua) para automação e controle e Aplicações para o projeto eLua[3][7].

2-Objetivos:

Os AVRs tem cpus de 8bits, são simples e lentos comparados aos ARMs, uma arquitetura largamente usada em MCUs de 32 bits hoje. São indicados a tarefas que exigem pouco processamento, e são utilizados por terem um custo extremamente baixo mesmo comparado aos ARMs. Possuem interfaces de hardware como UART, I2C e SPI, que são protocolos de comunicação entre chips, conversores ADC (Analogic Digital Converter) portas para interfaces com displays, LEDs dentre outras. Normalmente programados em linguagens como Assembly ou C, e exigem um conhecimento grande de eletrônica e da arquitetura da cpu, como operam. Já os ARMs, tem cpus de 32bits, são mais parecidos com computadores e tem uma capacidade de processamento consideravelmente mais alta do que os AVRs. São indicados para tarefas de processamento intermediária, onde um computador ainda teria custos muito altos ou seriam muito grandes e que um AVR não é capaz de realizar a tarefa. Um uso deles no dia a dia são os processadores dos celulares. Possuem ainda mais interfaces de hardware se comparados aos seus concorrentes menores, como DMA e JTAG. Pode ser programado em C, C++, eLua. Mais sobre eLua adiante (por sua abstração da plataforma permite um menor conhecimento de eletrônica e do processador).

Aprender os ambientes de programação em C[5] para os microcontroladores AVR e ARM, e o ambiente eLua apenas para ARM. Estudar as arquiteturas de tais microcontroladores e aprender os protocolos e interfaces de hardware.

Criação de módulos periféricos utilizando AVRs para auxiliar com o pré-processamento de dados e posteriormente enviá-los para os ARMs para o processamento final.

Estender o ambiente eLua com drivers de display, em especial (HD44780, KS0108B e do

Nokia 3310). Estudar e contribuir para a arquitetura do projeto eLua que esta em desenvolvimento no laboratorio citado acima. Refinamento do robo Manfredo e criacao de aplicacoes utilizando o eLua como meio.

3 - Metodologia:

Depois da concepção inicial do projeto, partimos para a fase de prototipação, na qual o instrumento é montado de maneira provisória, de forma a podermos entender e checar todas as necessidades do sistema. Depois de basicamente montado, é realizada uma fase de testes em que toda a montagem é submetida a uma bateria de testes para poder garantir o funcionamento desejado. São utilizadas diversas técnicas de depuração de hardware e software. Todas as possíveis falhas e bugs do sistema são detectados neste processo. Depois de tudo aprovado, começa a fase de montagem final, em que é feita uma placa dedicada ao projeto e enfim vamos obter um produto finalizado.

A criação de bibliotecas, tanto em linguagem C quanto em Lua[4], é um passo fundamental de toda a parte de desenvolvimento de software. É necessário criar um banco de funções testadas e trabalhadas para que possam ser utilizadas posteriormente em diferentes aplicações. Um dos grandes focos de cada projeto é fazer toda a programação de forma que o código possa ter um alto grau de reusabilidade.

Os projetos também envolvem todo o desenvolvimento de hardware necessário. É um trabalho de eletrônica embarcada, que depende da escolha da MCU ideal, levantamento dos componentes necessários, esquematização elétrica, montagem e prototipação. Todo o estudo e criação do hardware são realizados no laboratório em conjunto com o resto da equipe. Também existe um trabalho de produção de “Printed Circuit Boards” (PCB’s ou Placas de Circuito Impresso), que são placas eletrônicas comuns, com trilhas elétricas entre os componentes do projeto. As técnicas utilizadas na produção das PCB’s são perfeitas para produtos sem larga produção, protótipos ou testes.

4 - Desenvolvimento:

O desenvolvimento se dá em 3 etapas principais, a escrita do código (software), a concepção da eletrônica e por fim a confecção das placas de circuito impresso (hardware). A criação do hardware começa com a definição do microcontrolador utilizado. Esta se dá dimensionando os requerimentos do sistema. Para aplicativos satélites, que exigem pouco processamento utilizaremos os AVR’s e para projetos mais complexos utilizaremos os ARM’s. Essa definição influi diretamente na criação de software, pois altera as ferramentas a serem utilizadas, e por isso geralmente é a primeira etapa.

No caso dos projetos satélites fizemos uma interface de rotary switch. Uma pequena rodinha parecida com o botão de rolagem dos mouses que será utilizada para interface dos aparelhos com o usuário final dos produtos. Utilizaremos a linguagem C e o microcontrolador AVR ATtiny25 para tal. Ele será responsável por converter os sinais do rotary em um acumulador e que será posteriormente transmitido por I2C para um outro microcontrolador.

5 - eLua:

eLua significa Embedded Lua (Lua Embarcado) e é um projeto que oferece a linguagem de programação Lua para o mundo da eletrônica embarcada.

eLua não é apenas uma parte da linguagem disponibilizada para o programador embedded, é Lua em versão completa. eLua estende alguns tipos nativos de Lua para o mundo embarcado (exemplo: light-tables, light-functions etc) para permitir um uso mais otimizado das memórias RAM e ROM das MCU's.

Lua é o perfeito exemplo de uma linguagem mínima, porém absolutamente funcional. Embora normalmente anunciada como uma "linguagem de script" (e usada desta forma na indústria de jogos eletrônicos), Lua também é totalmente capaz de rodar aplicações "stand-alone", ou seja, que se sustentam por si só, sem outra aplicação principal escrita em outra linguagem. A limitada lista de requerimentos do sistema faz com que a linguagem seja muito compatível com várias famílias de microcontroladores. Lua é incrivelmente portátil (o código é todo ANSI C e funciona virtualmente em qualquer plataforma conhecida).

O projeto eLua foi arquitetado de forma a suportar as mais populares MCU's do mercado e abre um novo sentido para o termo "portabilidade" no mundo embedded. O mesmo programa Lua, com nenhuma ou pouquíssimas alterações, roda em todas as plataformas suportadas por eLua, mesmo em diferentes arquiteturas.

O objetivo do projeto é ter um ambiente de desenvolvimento Lua totalmente funcional no próprio microcontrolador, sem a necessidade de instalar nenhum tipo de ambiente de desenvolvimento específico no lado do PC.

6 - Projetos Desenvolvidos:

Os trabalhos neste período permitiram o avanço em vários projetos. Aqui serão ressaltados alguns dos principais, analisando os resultados obtidos.

A maior parte do desenvolvimento no laboratório diz respeito ao projeto eLua, com diferentes necessidades envolvidas. Nos últimos tempos a maior ênfase tem sido criar aplicações baseadas em eLua. Estas aplicações tem sido muito variadas e visam sistemas de aquisição de dados, automação de laboratório, rastreamento, navegação robótica, comunicação sem fio (protocolo Zigbee) e outras.

6.1 - ROBÔ MANFREDO:

Foi iniciado o desenvolvimento de um robô autônomo com navegação baseada em GPS. O robô, batizado de Manfredo, é uma plataforma terrestre motorizada e controlada para ajudar a criar, estudar e testar diversos módulos de posicionamento e navegação interessantes para diversas aplicações. O projeto é baseado em um chip da família ARM-Cortex M3 da Luminary Micro e programado em linguagem Lua (utilizando o conceito de eLua).

O algoritmo criado permite que o robô receba uma rota pré-estabelecida de pontos em forma de coordenadas geográficas e, a partir do primeiro ponto, siga através dos demais seguindo a trajetória planejada. Estes pontos devem ser definidos anteriormente e carregados no sistema juntamente com o programa de controle.

Foram adicionados também alguns sensores de ultra-som para que Manfredo possa identificar e desviar de obstáculos. Estes sensores são capazes de perceber barreiras, tais como paredes, veículos, pessoas etc, e alertar o programa principal. São quatro sensores instalados: um de cada lado, um na frente e um nas costas do robô. Sabendo de onde

veio o alerta, o programa se encarrega de desviar para o lado correto e seguir o percurso normalmente.

Todo o software foi escrito em linguagem Lua pelos alunos do laboratório. O programa cuida de todo o posicionamento, acionamento dos motores, movimentação e checagem dos sensores. Foi criada também uma biblioteca de navegação, nomeada “nav”, escrita em linguagem C para auxiliar algumas contas matemáticas com as coordenadas recebidas através do GPS.

Foram construídas três principais placas de circuito impresso: para alimentação, comunicação serial e controle dos motores. Todas as placas foram desenvolvidas pela equipe do laboratório e montadas pelos alunos. A alimentação do robô é feita com duas baterias de 12V, são usados dois motores DC e há uma roda “biruta” na parte da frente. Foi colocada recentemente uma haste para a fixação da antena do GPS em um ponto mais alto.

Manfredo é provavelmente o primeiro robô de navegação autônoma montado em uma universidade brasileira e certamente o primeiro controlado em Lua. O projeto segue em desenvolvimento e há planos para que, em um futuro próximo, seja incorporado um módulo de navegação inercial para as áreas que não têm cobertura de sinal GPS. Também está previsto um módulo de comunicação sem fio para que o robô possa enviar e receber informações em tempo real.

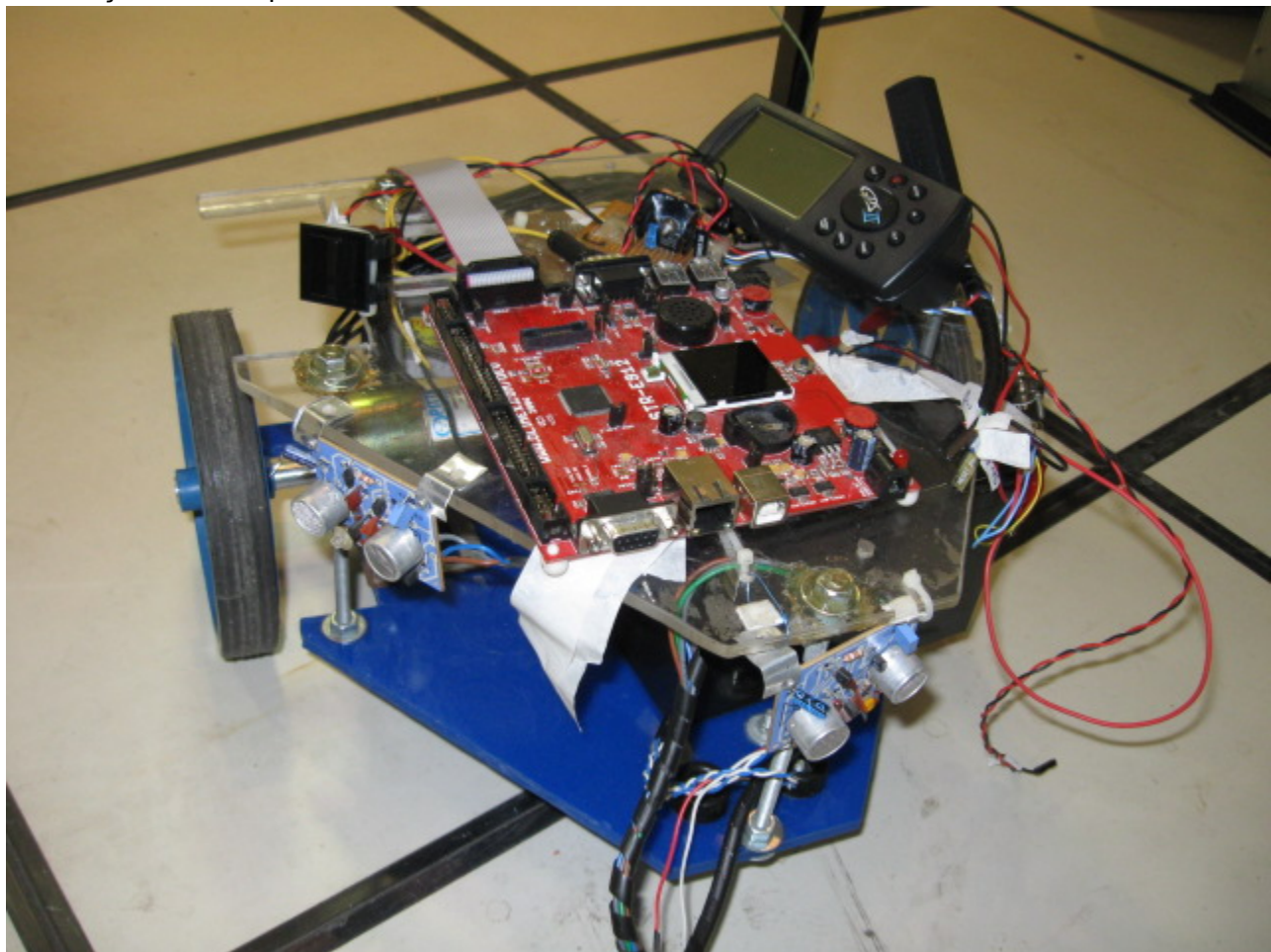


Foto: Visão frontal do manfredo.



Foto: Visão superior do manfredo

6.2 - Driver do display KS0108B:

Este modulo utiliza os recursos PIO de eLua para interfacia-lo com o display da imagem. Para tal foi necessario estudar a arquitetura de eLua e como fazer a adicao de tais modulos. Foi também preciso estudar o display para saber como enviar comandos e dados a ele.

O desenvolvimento demorou em torno de três meses pela dificuldade que é o processo de debug em sistemas embarcados. Mas como podem ver abaixo deu bons resultados. O modulo esta sendo integrado com a versão oficial de eLua e a partir desse, foi criada uma base solida de conhecimento para desenvolvimento dos outros drivers.



7 - Conclusões:

Os trabalhos neste período permitiram finalizar alguns projetos, enquanto outros continuam em desenvolvimento. Além dos principais, já citados e explicados neste trabalho, houve um contato com outros projetos na área de aplicações web e telefonia IP.

Há ainda a expectativa de projetos futuros na área de segurança e algo relacionado ao monitoramento de resultados de atletas brasileiros. Para entrar nestas duas novas áreas de atuação, pretendemos explorar cada vez mais tecnologias de comunicação sem fio para sensoriamento remoto dos instrumentos. Com isso, planejamos no futuro passar a utilizar a promissora e desafiadora arquitetura ZigBee, de forma a expandir as capacidades de comunicação sem fio dos projetos.

A tendência é cada vez mais consolidar o desenvolvimento do projeto eLua, que vem crescendo e conquistando novos resultados. Já existe uma parceria com a Liquigás para um projeto a respeito de aquisição de dados em caminhões da empresa utilizando a tecnologia desenvolvida no laboratório.

Todo este trabalho, que envolveu o uso de instrumentos e técnicas de bancada e análise de sinais, contribuiu para solidificar a formação sobre o assunto. Há hoje uma grande demanda por instrumentação e controle de dispositivos embarcados e queremos entrar neste ramo da engenharia com o melhor preparo possível.

7 - Referências

- 1 - http://en.wikipedia.org/wiki/Atmel_AVR - Data da consulta: 31/7/2010
- 2 - PEREIRA, Fabio Tecnologia ARM: Microcontroladores de 32 bits. São Paulo: Editora Érica, 2007.

- 3 - <http://www.eluaproject.net> – Data da consulta: 31/7/2010
- 4 - http://en.wikipedia.org/wiki/Lua_language – Data da consulta: 31/7/2010
- 5 - Kerningham and Richie The C Programming Language - Prentice Hall PTR - - Fevereiro de 2010
- 6 - http://en.wikipedia.org/wiki/Assembly_language – Data da consulta: 31/7/2010
- 7 - <http://wiki.eluaproject.net> - Data da consulta: 31/7/2010