



AUTOMAÇÃO DE INSTRUMENTAÇÃO DE LABORATÓRIO

Téo Ferraz Benjamin ¹

Raul Almeida Nunes ²

Dado Sutter ³



¹ Aluno de Engenharia de Controle e Automação da PUC-Rio

² Prof. Dr. do DCMM – Orientador dos projetos

³ Pesquisador Visitante da PUC-Rio – Co-Orientador dos projetos

SUMÁRIO

1. Introdução	3
2. Objetivos	3
3. Metodologia	4
4. Desenvolvimento	4
5. eLua	5
6. Projetos Desenvolvidos	5
5.1 – Manfredo	
5.2 – Césamo	
5.3 – Módulos eLua	
7. Conclusões	8
8. Referências	9

1 - Introdução

O trabalho, desenvolvido desde 2007, visa estudar e desenvolver projetos de instrumentação, controle e automação de equipamentos dedicados. São produzidos projetos baseados em microcontroladores utilizando software e hardware desenvolvidos pelo grupo, resultando em sistemas conhecidos como "Embedded Systems" ou "Sistemas Embarcados".

A ênfase do último ano de trabalho foi o uso do projeto eLua [1] (Embedded Lua), que visa trazer a linguagem Lua [2], criada e desenvolvida na PUC-Rio, para o mundo dos microcontroladores. O projeto, conduzido pelo pesquisador Dado Sutter, está sendo desenvolvido no laboratório (LabMAR) e tem alcançado grandes resultados por suas fortes características de portabilidade, flexibilidade e abstração do hardware, além de levar uma linguagem de alto nível para um ambiente que tradicionalmente dependia de C e Assembly, linguagens de baixo nível. Este último ponto traz uma enorme vantagem para o programador final de aplicações eLua.

Os microcontroladores (MCU – Micro-Controller Unit) são pequenas centrais de processamento equipadas com uma série de componentes periféricos que podem ser usados ou não, dependendo da necessidade. Os microcontroladores usados por nós têm disponíveis Fontes Geradoras de Clock, Conversores A/D e D/A, Timers, Dispositivos de Comunicação Serial, Displays, Teclados e muitos outros periféricos úteis. Programados com o software desenvolvido pelo grupo, permitem o controle de todo o experimento e um processamento de dados em alta velocidade. Os periféricos variam de microcontrolador para microcontrolador e quando é necessário o uso de algum dispositivo inexistente na família de MCU usada, é possível fazer o interfaceamento com componentes externos. Para isto é interessante a criação de bibliotecas que consigam integrar estes dispositivos.

Novos módulos, para diversas plataformas de 32 bits (STM32, AVR32, ARM7, ARM9, ARM Cortex M3, ...) serão desenvolvidos este ano, como o suporte ao protocolo CAN e técnicas de comunicação sem fio como Bluetooth e WiFi. O plano é que os módulos desenvolvidos sejam testados e posteriormente disponibilizados para que outros desenvolvedores utilizá-los em projetos

2- Objetivos

Dominar técnicas de programação de firmware para sistemas embarcados baseados em microcontroladores; técnicas aquisição rápida de dados baseadas em interrupções; técnicas de interfaceamento com dispositivos de comunicação com e sem fio e técnicas de interface com o usuário em dispositivos com poucos recursos. Estudar técnicas de programação para baixo consumo de energia, importantes para sistemas embarcados.

Desenvolver técnicas de programação em linguagem C utilizando ambientes tradicionais de desenvolvimento de aplicações embedded. Dominar a programação em linguagem Lua, bem como sua adaptação para desenvolvimento de aplicações embedded.

Entender a arquitetura de microcontroladores de 16 e 32 bits, sobretudo das famílias ARM [3] e MSP430 [4]. Criação de bibliotecas e módulos para o projeto eLua visando estas e outras plataformas de desenvolvimento.

Projetar e desenvolver hardware dedicado para as aplicações propostas. Projetar e construir uma plataforma de desenvolvimento dedicada à eLua (já em desenvolvimento).

Manter a estrutura de desenvolvimento de Software Livre e de Código Aberto, de forma a suportar uma comunidade crescente de usuários e de desenvolvedores, através da hospedagem e manutenção de ferramentas de desenvolvimento colaborativo.

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, 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.

Para o baixo consumo de energia elétrica, uma preocupação freqüente em nossos trabalhos, os microcontroladores apresentam modos de Low-Power, muito explorados em todos os projetos do grupo. Na família MSP430, por exemplo, são 5 níveis de Low-Power, e quanto maior o nível, mais periféricos são desligados. Além de inutilizar partes que não serão usadas para poupar energia, os modos Low-Power mantém o próprio processador “dormindo”, sem executar operações numéricas do loop principal do programa.

Uma excelente forma de explorar este recurso é o que chamamos de software interrupt-driven: o programa que controla o funcionamento da MCU utiliza um modo Low-Power e deixa o processador dormindo; utilizando técnicas de interrupção por hardware, fazemos o microcontrolador realizar uma operação apenas quando uma interrupção for gerada; estas interrupções podem significar operações externas realizadas pelo usuário ou operações internas do funcionamento da MCU, como por exemplo o fim de uma conversão A/D ou um estouro de contagem de um Timer.

4 – Desenvolvimento

Para um equipamento funcionar de forma automatizada é necessário também um circuito de alimentação elétrica e vários tipos de componentes eletrônicos. Para isso, precisamos fazer um projeto completo que possa transformar a idéia e o protótipo em um produto. Utilizando nossas PCBs podemos ligar componentes como por exemplo conectores,

reguladores de tensão, transistores e FETs, conectá-los também às portas corretas do microcontrolador e a um circuito de alimentação elétrica. Com isso podemos fazer o controle de toda a placa utilizando o software já desenvolvido e testado, economizando espaço e criando um produto seguro e confiável.

Para o desenvolvimento de software, utilizamos atualmente as linguagens C [5], Assembly [6] (MSP430 e ARM7 da ST) e Lua. Cada projeto se encaixa melhor em uma determinada característica de uma linguagem, mas no geral utilizamos a linguagem C para programar os microcontroladores. O programa é todo pensado e projetado para as necessidades do projeto e escrito e revisto por nós utilizando diversas técnicas de depuração para a eliminação de bugs e erros do sistema. Durante a etapa de testes o programa deve ser testado diversas vezes de forma que se possa garantir seu bom funcionamento.

5 – eLua

eLua significa Embedded Lua (Lua Embarcado) e é um projeto que visa oferecer todas as funcionalidades da linguagem de programação Lua para o mundo de tecnologia embarcada.

eLuanão é apenas uma parte da linguagem disponibilizada para o programador embedded. Muito pelo contrário. Além de oferecer diferentes implementações da linguagem Lua completa (exemplo: representação numérica em inteiros ou ponto flutuante), 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 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 a 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.

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.

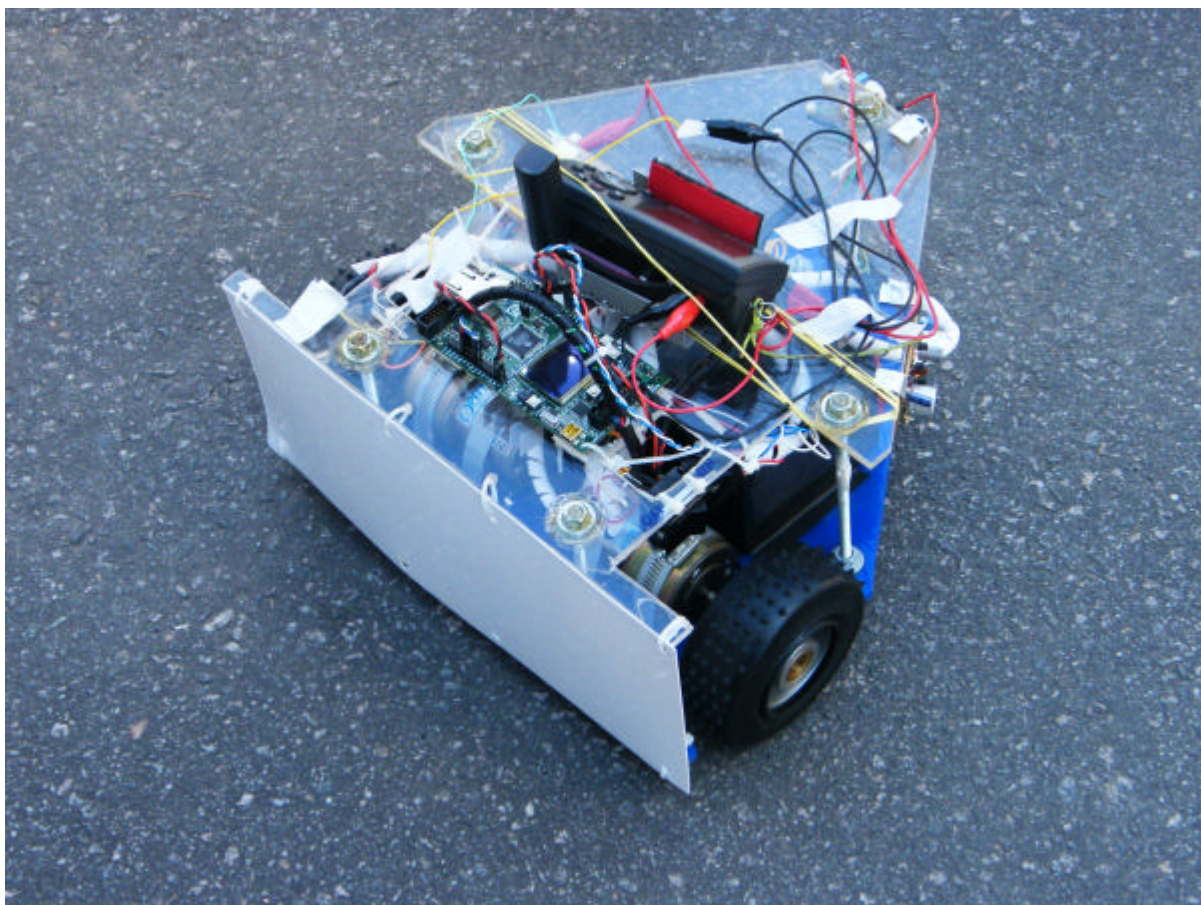


Foto: versão preliminar Robô Manfredo

6.2 – CÉSAMO

O projeto Césamo consiste em um sistema eletrônico de segurança e abertura de portas. Utilizando um microcontrolador da família MSP430, o programa foi desenvolvido para controlar a abertura da porta do laboratório através de uma senha. Tudo foi projetado pensando em algumas facilidades que incluem uma mudança de senha extremamente simples e um histórico de dias e horários em que a porta foi aberta. Posteriormente, podem haver várias senhas individuais para que se saiba quem abriu a porta.

O sistema se baseia em sensores capacitivos, que detectam a variação do campo magnético em torno de uma área metálica. Estes sensores podem detectar até seis áreas diferentes, cada uma sendo um botão. Cada botão pode ser colocado atrás de uma superfície lisa, como vidro ou plástico, sem ter a detecção do toque atrapalhada. Isto significa que uma matriz com seis áreas pode ser colocada atrás de um vidro próximo à porta de acesso e servir como teclado para a senha. O fato de ficar localizada atrás do vidro dificulta qualquer ação depredatória, protege o equipamento de impactos acidentais, sujeira e facilita muito a manutenção.

O software, escrito em linguagem C, foi desenvolvido em conjunto pelos alunos do laboratório. É todo baseado em Low-Power Mode e interrupt-driven. Cada toque em um botão gera uma interrupção no programa que checa se a senha já foi completada. Caso a senha seja digitada corretamente, a porta se abre com um pulso elétrico.

Todo o hardware (placa de circuito impresso, placa com os botões, ligações elétricas,

cabeamento etc) foi projetado e montado no laboratório. Existe um circuito auxiliar que ajuda a modular o pulso de abertura da porta e um interruptor que serve para abri-la direto pelo lado de dentro, sem a necessidade da senha. Tudo foi colocado em uma caixa e grudado junto ao vidro da janela do hall de entrada do laboratório.

O sistema se encontra hoje em fase de testes e continua ganhando novas funcionalidades. No futuro pode haver um sistema que avisa (por e-mail, sms ou qualquer outro meio) se a porta for aberta em um horário considerado anormal ou se a senha for digitada de erradamente algumas vezes consecutivas. A idéia é tornar o produto utilizável e atraente para outros interessados, seja em laboratórios da PUC-Rio ou não.

6.3 – MÓDULOS ELUA

Uma parte importante dos trabalhos nos últimos tempos tem sido ajudar a testar, refinar e criar módulos eLua. Um bom exemplo são os testes os principais módulos como Timer, PI/O, Uart e PWM. Módulos são bibliotecas com as funções básicas para o uso de cada um destes periféricos. Todas as funções precisam ser testadas e corrigidas e esta é uma função fundamental no desenvolvimento do projeto.

Também foram desenvolvidos alguns módulos para aplicações específicas. Além do módulo Nav, escrito em C e já citado anteriormente, foram criados módulos para facilitar o interfaceamento com um Display de OLED e um específico para aplicações com uso de GPS.

Este último tem diversas funções que permitem adquirir dados de um GPS qualquer no padrão NMEA (protocolo de comunicação comum em aparelhos GPS) e calcular informações importantes. Este módulo foi extremamente útil no projeto do robô Manfredo e em outros projetos, como uma demonstração para a Liquigás.

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://www.eluaproject.net> – Data da consulta: 12/8/2009
- 2 - http://en.wikipedia.org/wiki/Lua_language – Data da consulta: 12/8/2009
- 3 - PEREIRA, Fabio **Tecnologia ARM: Microcontroladores de 32 bits**. São Paulo: Editora Érica, 2007.
- 4 - PEREIRA, Fabio **Microcontroladores MSP430 - Teoria e Prática**. São Paulo: Editora Érica, 2006. 416p.
- 5 - http://en.wikipedia.org/wiki/C_language – Data da consulta: 12/8/2009
- 6 - http://en.wikipedia.org/wiki/Assembly_language – Data da consulta: 12/8/2009