



SISTEMAS EMBARCADOS PARA AUTOMAÇÃO DE INSTRUMENTAÇÃO DE LABORATÓRIO

Téo Ferraz Benjamin ¹

Raul Almeida Nunes ²

Dado Sutter ³



**Conselho Nacional de Desenvolvimento
Científico e Tecnológico**

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	3
4. Desenvolvimento	
5. Projetos Desenvolvidos	
5.1 – Bibliotecas de Interfaceamento	
5.2 – Anubisway	
5.3 – Misturador de Longa Duração	
5.4 – Power-Wheel	
6. Conclusões	
7. Referências	

1 - Introdução

O grupo de eletrônica e software embarcado visa atender a demanda de automação e controle de instrumentos do LabMAR (Laboratório de Monitoramento Ambiental Remoto do DCMM) e de outros laboratórios da PUC-Rio, desenvolvendo instrumentos para aquisição de dados, controle e interfaceamento com outros experimentos. Os sistemas desenvolvidos são baseados em microcontroladores dedicados, com software (firmware) e hardware desenvolvidos pelo grupo, resultando em sistemas conhecidos como "Embedded Systems" ou "Sistemas Embarcados".

Sempre focados em produtos com alto grau de inteligência embarcada e performance para processamento numérico, utilizamos microcontroladores dedicados de 16 e 32 bits, principalmente das famílias MSP430 e ARM's (Advanced RISC Machines), para aquisição e processamento dos dados dos instrumentos.



Figura 1:
MSP430

Os microcontroladores (MCUs – Micro Controller Unit) são unidades centrais de processamento equipadas com uma série de componentes periféricos, que podem ser usados ou mantidos inativos, dependendo da necessidade. Os microcontroladores usados por nós disponibilizam periféricos como; Fontes Geradoras de Clock, Conversores A/D e D/A, Timers, Amplificadores Operacionais, Dispositivos de Comunicação Serial e muitos outros igualmente úteis. Programados com o software desenvolvido pelo grupo, permitem o controle de todo o experimento e um processamento de dados em alta velocidade, tipicamente entre 5 e 50 MIPS. Os periféricos disponíveis variam segundo o modelo e quando é necessário o uso de algum dispositivo inexistente na família de MCU usada, é possível fazer o interfaceamento com componentes externos .

Para facilitar nosso trabalho de prototipação e criação dos projetos, utilizamos módulos fabricados pela parceira ScTec [3] (fig 2) que são altamente adaptáveis e preparados diretamente para nossas plataformas de prototipação. Possuímos hoje módulos para MCUs das famílias PIC (Microchip), MSP430 (Texas Instruments), ARM7 (ST Electronics), Z8 Encore (Zilog), HCS08 (Freescale/Motorola), além de outros kits de desenvolvimento para AVR (Atmel), Tibbo (Taiko Industries).

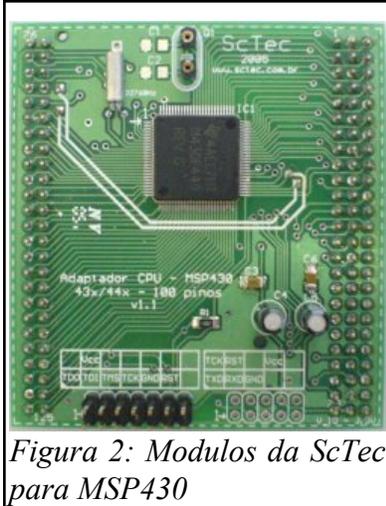


Figura 2: Módulos da ScTec para MSP430

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, técnicas de interface com o usuário em dispositivos com poucos recursos e técnicas de programação em modos de baixo consumo.

Utilizar linguagem C e ambientes de desenvolvimento para aplicações embedded, noções de Assembly de MSP430 e de ARM, noções de Lua, criação de bibliotecas para a reusabilidade do código produzido e explorar técnicas de programação em modos de baixo consumo nas diversas arquiteturas utilizadas.

3 - Metodologia

Nos últimos projetos e instrumentos desenvolvidos, foi dada ênfase aos aspectos de "Interface com o Usuário" e de "Técnicas de Programação para Baixo Consumo". Novas técnicas de interação com o usuário, notadamente baseadas em Rotary Encoders Óticos com botão e Displays de LCD variados, foram desenvolvidas, testadas e refinadas. Os resultados foram finalizados em bibliotecas portáteis para diversas arquiteturas de microcontroladores que foram utilizadas em produtos que serão mostrados neste trabalho.

Para o baixo consumo de energia elétrica, uma preocupação frequente em nossos trabalhos, os microcontroladores apresentam modos Low-Power, muito explorados em todos os projetos do grupo. Na família MSP430, por exemplo, são 5 níveis de Low Power. Quanto maior o nível, mais periféricos são desativados. Além de desabilitar módulos 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", onde o programa que controla o funcionamento da MCU utiliza um modo Low Power e deixa o processador "dormindo", inativo, em standby com apenas algumas funções básicas ativas. 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 Analógico-Digital ou um estouro de contagem de um Timer.

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 montado, tanto o hardware quanto o software são submetidos a uma bateria de testes para poder garantir o funcionamento desejado. Depois de tudo aprovado, começa a fase de montagem final, na qual uma placa de circuito impresso dedicada ao projeto é projetada e construída, para enfim vamos obter um produto finalizado.

4 - Desenvolvimento

Para o desenvolvimento de todo o hardware, utilizamos a plataforma de prototipação Microlab X1 (fig 3), protoboards e simuladores de circuitos. Recentemente iniciamos também o desenvolvimento de Placas de Circuito Impresso (PCB – Printed Circuit Board) próprias totalmente projetadas e produzidas no laboratório.

O uso da plataforma MicroLabX1 se mostrou extremamente produtivo, por apresentar uma arquitetura multi-MCU, ser muito versátil e permitir a realização de montagens de modo rápido e fácil. Uma das grandes facilidades da MLX1 é a presença de diversos tipos de dispositivos instalados, como potenciômetros, relés, displays de 7 segmentos, um pequeno teclado e vários outros, frequentemente utilizados em projetos. Além disso, ela contém barramentos extras que ajudam no uso de Displays LCD, módulos de controle de Ponte-H, displays gráficos e outros dispositivos.

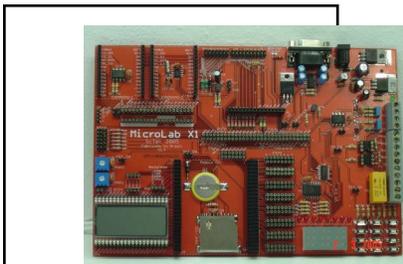


Figura 3: MicroLab X1

Para um equipamento funcionar de forma automatizada é necessário também um circuito de alimentação elétrica e diversos outros componentes eletrônicos, o que exige a elaboração de um projeto completo, que possa transformar a idéia e o protótipo em um produto final. Nossas PCBs (Printed Circuit Boards) garantem então a interconexão destes componentes, como por exemplo, conectores, reguladores de tensão, transistores, diodos, FETs, (...) e também as ligações com as portas corretas do microcontrolador e a um circuito de alimentação elétrica. Assim podemos fazer o controle de toda a placa utilizando o software já desenvolvido e testado, economizando espaço e criando um produto seguro e confiável. O software desenvolvido para operar desta forma, "embarcado" em um controlador dedicado, é conhecido como Software Embedded. O conjunto de programas de um sistema Embarcado como este é também conhecido por Firmware.

Para o desenvolvimento de software, utilizamos atualmente as linguagens C [4], Assembly [5] (MSP430 e ARM7 da ST) e Lua [6]. 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 - Projetos Desenvolvidos

Os trabalhos neste período permitiram finalizar alguns produtos, enquanto outros continuam em desenvolvimento. Entre os principais concluídos, podemos citar as bibliotecas para o uso de Displays LCD e Rotary Optical Encoders, o Misturador de Longa Duração para o DCM, e o interfaceamento da MicrolabX1 com um servidor web embarcado. Entre os que continuam em desenvolvimento, podemos citar o equilíbrio do Anubisway, um antigo robô de combate da RioBotz [7] e protótipos de interfaceamento com redes de telefonia IP da Cisco Systems.

Aqui serão destacados alguns de nossos projetos, os que receberam um maior foco e tiveram maior participação do aluno nos últimos tempos.

5.1 - BIBLIOTECAS DE INTERFACEAMENTO

Uma das primeiras dificuldades que detectamos em nossos trabalhos foi a de utilizar os mesmos dispositivos em diferentes projetos e sermos obrigados a reprogramar toda a configuração dos mesmos dispositivos em cada projeto.. Isto motivou a criação de bibliotecas únicas, de fácil uso e rápida configuração, para as tarefas mais usadas nos projetos.

No princípio fizemos um template básico das configurações de clock e periféricos internos de algumas famílias de MSP430 (especialmente 449, 1611 e 2013: as mais usadas por nós). Com isso, em vez de escrever todas as linhas de comando, era necessário apenas retirar ou ativar o que fosse necessário para a programação de software de cada projeto.

Mais tarde vimos a necessidade de criar uma biblioteca para o interfaceamento com Displays de LCD, presentes em vários projetos. Até então nós utilizávamos uma biblioteca que utilizava um protocolo de comunicação com os displays que acabou se mostrando pouco eficiente. Adaptamos então uma biblioteca encontrada em pesquisas pela rede para o tipo de Display que usamos.

Depois de uma série de adaptações para tornar esta biblioteca portátil para diversos tipos, modelos e tamanhos de Displays, algumas mudanças em uma série de funções auxiliares e a inclusão de algumas outras, terminamos por torná-la rápida, eficiente e prática para nossos projetos.

Ainda na parte de “interface com o usuário”, começamos a utilizar bastante os dispositivos Rotary Encoders com botão e a explorar suas funcionalidades. Começamos então a desenvolver técnicas de interfaceamento com os Rotaries, buscando facilidade de uso, praticidade e resposta rápida do software. Entre essas técnicas, tivemos que criar uma boa forma de fazer *debounce* do Encoder utilizando técnicas de interrupção para gerar um delay sem prejudicar a usabilidade do produto. Depois de alguns protótipos criados utilizando este dispositivo e com a filosofia já concebida, partimos para a criação de uma biblioteca única para este tipo de programação, que se mostra hoje extremamente útil para diversos projetos em andamento.

Assim como a biblioteca para o uso dos Displays, esta consiste em um arquivo de configuração separado, todo escrito em linguagem C, que deve ser incluído no projeto desejado. Depois disso, basta mudar alguns parâmetros e variáveis de acordo com o uso desejado no projeto. A variação destes parâmetros se deve a mudanças que podem ocorrer em algumas características de cada programa, como por exemplo: velocidade do clock principal, portas de saída utilizadas no microcontrolador para cada dispositivo e periféricos que são usados ou não.

É importante ressaltar que todas as bibliotecas desenvolvidas e utilizadas permitem que o software funcione em Low Power mode se necessário. Podendo ser utilizadas em programas interrupt-driven.

5.2 - ANUBISWAY

O Anubisway é um projeto que foi iniciado há cerca de um ano em parceria com a equipe de robótica da PUC-Rio, a RioBotz [7]. Depois de um tempo parado o projeto foi retomado e se encontra em fase de testes.

A RioBotz cedeu um antigo robô de combate, do ano de 2004, chamado Anubis (fig 4) para ser usado no projeto. O robô possui um único eixo e duas rodas. A “arma” do Anubis, um martelo de ponta, era uma haste rígida com uma ponta cortante que ficava acoplada no centro da plataforma. Todo o conjunto pesava 55kg e teve que sofrer algumas adaptações para ser usado no projeto: a arma foi substituída por uma haste de apoio, foram retirados alguns pesos extras e componentes de reforço do chassi e foi colocado um material anti-derrapante na superfície na qual se pisa.

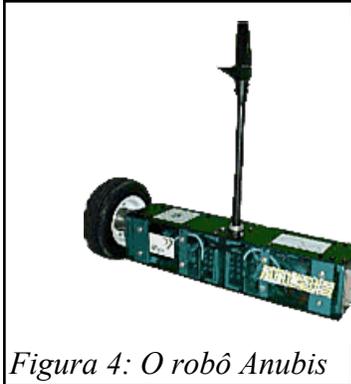


Figura 4: O robô Anubis

O projeto consiste em transformar este antigo robô em uma plataforma auto-equilibrada, modelando e implementando variáveis de controle a fim de se obter um meio de locomoção como um Segway [8](fig 5). Para o equilíbrio são usados acelerômetros e um giroscópio, controlados por nossos microcontroladores e programas.

O Segway é exatamente uma plataforma de duas rodas, auto-equilibrada, pilotável, eletricamente controlada e já disponível comercialmente. Com a inclinação do corpo do piloto para frente ou para trás, o centro de gravidade é deslocado e o dispositivo, ao medir essa diferença, acelera para frente ou para trás para equilibrar de novo o peso todo no eixo correto. Isto significa que com um controle bem modelado e instalado, pode-se fazer a plataforma andar para frente ou para trás apenas inclinando o corpo. Este é exatamente o objetivo do Anubisway. O nome do projeto é uma combinação do nome do antigo robô (Anubis) com o nome deste produto comercial hoje bem conhecido (Segway).



Figura 5: Segway

Em princípio o objetivo é fazer a plataforma se equilibrar sozinha, sem tombar para nenhum lado. Depois começaremos a refinar o controle até conseguir criar um meio de locomoção prático e principalmente seguro.

Para este projeto foram utilizados um acelerômetro de três eixos para medir os ângulos de inclinação e um giroscópio para calcular a aceleração angular em torno do eixo. Uma PCB foi desenvolvida especialmente para o projeto e todos o software está sendo escrito em linguagem C no laboratório e refinado ao longo dos testes. O interfaceamento com estes dispositivos já foi todo feito utilizando um microcontrolador da família MSP430 e o processamento dos dados é bastante eficiente. Porém, o controle refinado para que seja possível a presença de um piloto é mais complicado e exigente, do ponto de vista da precisão e calibração.

5.3 - MISTURADOR DE LONGA DURAÇÃO

O Misturador de Longa Duração (MLD) foi um projeto desenvolvido para ser utilizado em uma pesquisa do Departamento de Ciência dos Materiais e Metalurgia da PUC-Rio. Um modelo de um Misturador em V desenvolvido no LabMAR (Laboratório de Monitoramento Ambiental Remoto do DCMM PUC-Rio), gira em torno de um eixo acoplado a um motor DC de 12V. Para esta experiência o motor deve girar por períodos longos, tipicamente entre 30 e 90 dias, controlado com velocidade variável para realizar misturas que serão posteriormente analisadas.

Utilizando um microcontrolador da família MSP430 foi desenvolvido um software de controle do instrumento e da interface com o usuário, baseado em apenas um Display LCD e um Rotary Encoder ótico, utilizando as nossas bibliotecas previamente mencionadas.

Foi produzida uma PCB para o projeto, acoplando o microcontrolador, o Display LCD, o Rotary Encoder, fontes de alimentação externas e um módulo de controle para o motor via PWM, utilizando um circuito em Ponte-H. Tudo isso foi instalado em uma caixa de alumínio especialmente concebida para o projeto, de forma a apresentar um produto autônomo ao cliente final.

Por fora, o MLD é uma caixa fechada, com um Display e um Rotary Encoder na parte da frente, por onde se dá todo o controle, um conector para uma tomada comum de 110Vac e uma saída feita sob medida para o motor DC. O usuário pode escolher o tempo da experiência com uma resolução de horas e um máximo de 99 dias e 23 horas. Também há uma função para seleção de velocidade, com nove opções disponíveis entre aproximadamente 8 e 72 rpm. Enquanto o motor gira e a mistura é realizada o Display mostra informações sobre o tempo em que a experiência esteve girando e também o tempo em que esteve parada (para eventuais manutenções ou procedimentos de limpeza, por exemplo).

Por dentro todo o circuito de alimentação elétrica de uma fonte chaveada chega a uma PCB projetada e produzida no laboratório que faz toda a ligação eletrônica necessária entre os componentes. O software, desenvolvido em linguagem C, cuida de toda a parte de interface com o usuário e contagem de tempo restante, além de ser o gerador de PWM para o motor.

Como já foi dito, boa parte da filosofia dos projetos se baseia em técnicas de baixo consumo de energia. Seguindo esta idéia, todo o programa do MLD funciona em modo de low-

power, gerando um baixíssimo consumo por parte do microcontrolador e sendo totalmente interrupt driven.

5.4 - WIRELESS POWER-WHEEL

Também foram exploradas redes de comunicação sem fio, como links de RF em 900 Mhz e 2,4Mhz. Utilizando um modelo de MSP430 que utiliza este tipo de tecnologia, o ez430RF2500, resolvemos fazer alguns testes e criar protótipos que pudessem nos familiarizar com esse tipo de projeto.

Inspirados em um produto comercial já existente, resolvemos criar a nossa própria versão do dispositivo. Tudo se baseia em um aparelho que, através de uma espécie de Rotary Encoder, é capaz de controlar o cursor e executar macro-comandos no computador em uma configuração específica. Por exemplo: pode-se configurar o giro do Rotary em um sentido para realizar uma função (rolagem de página ou zoom, por exemplo), o outro sentido para outra função, o clique para outra e assim por diante.

Porém, para melhorar um pouco, resolvemos fazer o nosso produto sem fio, diferentemente do que é hoje encontrado comercialmente. A idéia seria fazer o dispositivo sem fio sendo utilizado pelo usuário e mandando as informações via link de RF para um outro dispositivo conectado a uma entrada USB do computador.

Este projeto rendeu bons frutos e bons estudos. Aprendemos a lidar melhor com a tecnologia sem fio da Texas Instruments utilizando a biblioteca de comunicação já distribuída pela empresa (SimpliciTI). Aos poucos elaboramos também a nossa própria biblioteca, mais simples e prática para os nossos projetos de sensoriamento remoto.

6 - Conclusões

Os trabalhos neste período permitiram finalizar alguns projetos, enquanto outros continuam em desenvolvimento. Além dos principais, já citados e explicados neste trabalho, podemos citar também:

- Projeto Eletroquímica, coordenado pelo Prof. Dr. Raul Nunes. É um projeto que visa obtenção de substância húmica através de processos de eletrólise e o estudo da melhor forma de se conseguir isto.
- Interfaceamento da MicrolabX1 com um servidor web embarcado realizado para uma demonstração junto à Petrobras para controlar servomecanismos e outros dispositivos a distância, via internet.
- Protótipos de interfaceamento com redes de telefonia IP da Cisco Systems, com realização de consultas de notícias online via RSS, gerador dinâmico de gráficos para IP Phones e concepção do uso de IP Phones para aplicações de Home Security
- Projeto Sigma: uma parceria entre a PUC-Rio e a Petrobras, na qual o sistema de gerenciamento de dados chamado Sigma, vem sendo melhorado e ajustado para uma realidade mais atual.

Há ainda a expectativa de projetos futuros na área de segurança e algo relacionado ao monitoramento da performance 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.

Todo este trabalho, que envolveu o uso de instrumentos e técnicas de bancada e análise de sinais, contribuiu para solidificar a formação em Instrumentação, Automação e Controle. 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 - PEREIRA, Fabio **Microcontroladores MSP430 - Teoria e Prática**. São Paulo: Editora Érica, 2006. 416p.
- 2 - PEREIRA, Fabio **Tecnologia ARM: Microcontroladores de 32 bits**. São Paulo: Editora Érica, 2007.
- 3 - <http://www.sctec.com.br/> – Data da consulta: 3/8/2008
- 4 - http://en.wikipedia.org/wiki/C_language – Data da consulta: 1/8/2008
- 5 - http://en.wikipedia.org/wiki/Assembly_language – Data da consulta: 1/8/2008
- 6 - http://en.wikipedia.org/wiki/Lua_language – Data da consulta: 1/8/2008
- 7 - <http://www.riobotz.com.br/> – Data da consulta: 3/8/2008
- 8 - <http://www.segway.com> – Data da consulta: 3/8/2008