



Evento	XXI FEIRA DE INICIAÇÃO À INOVAÇÃO E AO DESENVOLVIMENTO TECNOLÓGICO – FINOVA/2012
Ano	2012
Local	Porto Alegre - RS
Título	Desenvolvimento de Software Embarcado de Módulo de Gerenciamento Eletrônico para Motor Rotativo de Combustão Interna
Autor	FABRÍCIO DA SILVA STEIN
Orientador	CARLOS EDUARDO PEREIRA

Desenvolvimento de Software Embarcado de Módulo de Gerenciamento Eletrônico para Motor Rotativo de Combustão Interna

Fabrcio da Silva Stein; MSc. Alexandre Giordani Andreoli; Dr. Carlos Eduardo Pereira

Universidade Federal do Rio Grande do Sul - UFRGS
Av. Osvaldo Aranha, 103 CEP: 90035-190 - Porto Alegre, RS - Brasil
e-mail: fabricio.stein@ufrgs.br

Resumo — Em 2011 a empresa Gyrum⁽¹⁾, Pré-Incubada Externamente na UFRGS, começou o desenvolvimento de um motor rotativo de combustão interna inovador. Surgiu, portanto, a necessidade da criação de um *software* de gerenciamento eletrônico inexistente no mercado e que fosse capaz de controlar sua combustão de forma precisa e capaz de garantir seu correto funcionamento. O objetivo do projeto da BIT - Bolsa de Iniciação Tecnológica foi desenvolver tal *software* embarcado em linguagem C de programação para utilização em microcontroladores capaz de gerenciar todo o sistema de injeção de combustível e ignição da mistura, acionando válvulas injetoras de combustível e bobinas de ignição, realizando a leitura de todos os sensores necessários, como temperaturas, pressões, rotação, entre outros, em tempo real de funcionamento. O complexo programa deve ainda possibilitar ao novo motor rotativo competir com concorrentes grandes e estabelecidos no mercado. Portanto, praticamente todas as funções oferecidas pelo microcontrolador foram usadas no projeto, desde controle de *timers* até rotinas de PWM – *Pulse Width Modulation*, sendo implementadas no microcontrolador PIC18F4550 visando ao domínio da linguagem de programação e do microcontrolador. Após a simulação em *software* dessas rotinas fez-se a simulação em placas de circuito impresso (circuito físico) para teste das mesmas. A união de rotinas e funcionalidades do microcontrolador foi implementada, apesar dos diversos problemas de compatibilidade, através do sequenciamento e manipulação de interrupções. Ao final do primeiro ano, o resultado parcial esperado foi atingido, gerando-se um *software* embarcado capaz de gerenciar um motor de combustão interna. Com o motor rotativo em desenvolvimento, a BIT continuará com testes em conjunto com a parte mecânica do motor, gerando o código do produto final ao término das atividades.

Palavras-chave — *software* embarcado; gerenciamento eletrônico de motor de combustão interna; microcontrolador; linguagem C.

I. INTRODUÇÃO

A linguagem de programação utilizada foi o C. As plataformas utilizadas para escrita e compilação do código foram o programa “Dev-C++” e o “MPLAB IDE”. O MPLAB ainda tem o benefício de oferecer uma forma de rastrear os erros no código e simular possíveis sinais de entrada e saída, o que se faz muito útil na hora de testar o funcionamento das rotinas feitas. Primeiramente usou-se o PIC 18F4685. Mas, após o início da implementação, viu-se que o PIC 18F4550 seria mais dinâmico e responderia melhor às necessidades do projeto.

Para a programação, primeiramente deve-se fazer o reconhecimento das funções importantes que o microcontrolador PIC18F4550 é capaz de manipular. Nesse caso, usou-se o livro “Microcontroladores PIC18 – Aprenda e

programe em linguagem C”. A partir de exemplos e de teoria, aprendeu-se seu funcionamento e suas capacidades.

O objetivo do projeto é desenvolver um módulo de gerenciamento eletrônico, isto é, um *software* em linguagem C que controla a quantidade de combustível injetado no motor através do acionamento preciso das válvulas injetoras de combustível e o momento exato em que a centelha de ignição deve ser gerada, com base em cálculos efetuados a partir de dados colhidos por diversos sensores espalhados pelo motor. A Figura 1 mostra o motor rotativo Gyrum.



Fig. 1. Protótipo do motor rotativo Gyrum.

II. IMPLEMENTAÇÃO UTILIZANDO LINGUAGEM C

Inicialmente foi feita uma rotina em C envolvendo puramente lógica de programação. Um tempo considerável deve ser reservado para essa atividade, pois desenvolve-se uma rotina capaz de interpolar valores numa matriz de grande ordem - a qual será usada quando existem tabelas (rpm x temperatura, rpm x MAP) com valores de injeção de combustível a serem usados a partir delas.

Na sequência, partiu-se para a prática das rotinas em C. Praticamente todas as funcionalidades que o PIC18F4550 pode oferecer foram exploradas no projeto, entre elas estão: *capture*, *timers*, *compare*, PWM, *display LCD*, interrupções, conversor A/D, portas IO digitais⁽²⁾. Todas essas rotinas foram trabalhadas e testadas separadamente para aprendizado de suas funções e limitações. Uma vez feito isso, juntou-se uma a uma para implementação no projeto com suas contribuições individuais. Tais rotinas também foram implementadas fisicamente (montadas em circuito).

Uma vez testada cada função do microcontrolador, verificou-se a união delas. Geralmente é nessa união que ocorrem os problemas, uma função pode usar um *timer* que estava sendo usado como base de tempo, ou a frequência de interrupções é muito grande e acaba não permitindo ao o microcontrolador executar a sua rotina *main* (principal). São vários problemas que podem surgir, mas a forma mais rápida de fazer tudo funcionar junto é separar as partes que funcionam e tentar juntá-las uma a uma.

O programa MPLAB oferece algumas ferramentas muito úteis na hora da simulação e da procura de erros do

programa. Ele também já tem incluído em suas bibliotecas os padrões de linguagem de cada microcontrolador, as quais devem ser seguidas durante o projeto. Algumas das principais ferramentas usadas foram *Stimulus* (simula um gerador de sinais); *Watch* (supervisiona o valor das variáveis durante a simulação) e *Logic Analyzer* (interface gráfica dos sinais de entrada e saída)⁽³⁾. A partir do domínio do simulador, o processo de programação e de procura de erros acaba se tornando uma supervisão de tempos (tempo de *clock*, tempos de interrupção, etc), tais tempos devem ser calculados e registrados já que um vai influenciar os outros.



Fig. 3. Circuito de testes montado em bancada.

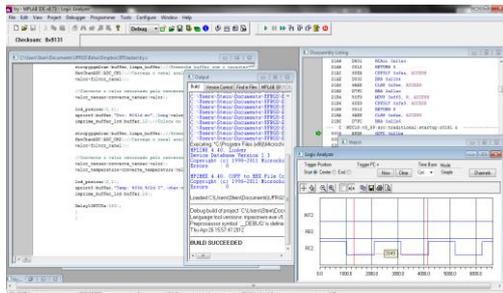


Fig. 2. Ambiente de programação e simulação no MPLAB.

Tem-se, no projeto, basicamente 3 rotinas básicas: leitura de velocidade a partir dos sensores na roda dentada do motor, injeção de combustível através do acionamento das válvulas injetoras de combustível e ignição através do acionamento preciso das velas de ignição. A primeira rotina, já implementada, é a leitura de velocidade através de pulsos provindos da roda dentada do motor. Essa roda dentada é composta, a princípio, de 35 dentes, isto é, 36 dentes com um dente falhado, dessa forma tem-se como saber quando se chega no ponto zero da rotação (dente falhado). Baseado nisso, foi usada a rotina *Capture* para leitura desses pulsos e medição de tempo entre eles. Uma vez que existe um intervalo de tempo n vezes maior que o anterior, este é o ponto zero. A velocidade é medida pela derivada de *clocks* de máquina entre cada dente, já que o tempo para um *clock* de máquina é sempre o mesmo (em condições ideais).

A injeção de combustível deve ser feita com as rotinas de PWM. Primeiramente havia sido feita uma rotina manualmente para indução da válvula injetora de gasolina, mas depois de pesquisa viu-se que seriam necessárias frequências de indução mais altas que até poderiam ser alcançadas pela rotina feita, mas não haveria necessidade de tantas interrupções quando se tem um PWM do próprio PIC18F4550 que faz a função desejada. O que deve ser feito é setar as condições desejadas para uma frequência de acionamento de, aproximadamente, 17 kHz na saída, o que deve ser suficiente para induzir o bico e permitir a passagem de combustível.

III. RESULTADOS

O aprendizado de funções como *display lcd*, *timers* e algumas interrupções não deve tomar muito tempo. Mas a fusão de várias funções e interrupções pode criar certas barreiras. Aprendeu-se sobre tratamento de interrupções, prioridade para driblar esse problema.

Com o sucesso desses testes, partiu-se para alguns testes com programação própria e a instalação da válvula injetora de combustível acoplado à *protoboard* (circuito), conforme a Figura 3.

Para isso, fez-se necessário o uso de algumas placas desenvolvidas previamente pelo MSc. Alexandre Andreoli. Foram usadas: uma placa fonte de 12V de entrada e 12V e 5V de saída para outros ramos do circuito, uma placa de *leds* para testes e uma placa *driver* de conversão, a qual converte a tensão de saída do PIC18F4550 para uma tensão adequada ao acionamento da válvula injetora de combustível.

Outros testes foram realizados e conseguiu-se a indução da válvula injetora em várias frequências, atingindo o objetivo de abri-la de fato a partir de um sinal PWM. Primeiramente, havia sido feito uma rotina manualmente que induzia o bico injetor com frequências menores. Após pesquisa, viu-se que o bico injetor usa frequências mais altas (aproximadamente 17 kHz) para acionamento, que podem ser produzidas pelo uso do próprio PWM do PIC18F4550, de qualquer forma, notou-se importantes relações de tempo de interrupção usadas na rotina de captura de velocidade rpm.

Após isso, conseguiu-se implementar uma rotina de captura de velocidade (rpm). A partir da roda dentada do motor pode-se retirar medições de rpm. Tais medições podem acontecer com maior ou menor frequência. Como o intervalo de tempo entre cada dente da roda dentada a altas rotações é muito pequeno, não se pode medir o tempo de dente para dente, uma vez que a própria execução da interrupção iria demorar mais que o processo em si. Nesse caso, mede-se o tempo transcorrido de quatro em quatro dentes para altas rotações, o que nos dá um intervalo um pouco maior para dar tempo de executar as rotinas e não atrapalha a precisão das medidas. Para rotações menores, pode-se usar uma resolução de captura maior e faz-se a leitura do tempo de dente em dente, o que resulta em uma boa precisão.

VI. CONCLUSÕES

Percebe-se que o projeto é bastante único e extenso, sendo que o que já foi feito em um ano de trabalho mostrou-se, de fato, expressivo. A partir de testes em laboratório, concluiu-se que o *software* embarcado é capaz de gerenciar um motor de combustão interna.

VII. TRABALHO FUTURO

Será dada continuidade ao projeto utilizando a programação orientada a objeto de modo que seja possível testar o sistema de gerenciamento de um motor rotativo através de ensaios de bancada no motor, com término previsto para o final do primeiro semestre de 2013.

REFERÊNCIAS

1. Disponível em <http://www.gyrum.com.br>.
2. Disponível em <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>.
3. Miyadaira, Alberto Noboru, Microcontroladores Pic18 - Aprenda e Programe em Linguagem C, Erica, Brasil, 2009.