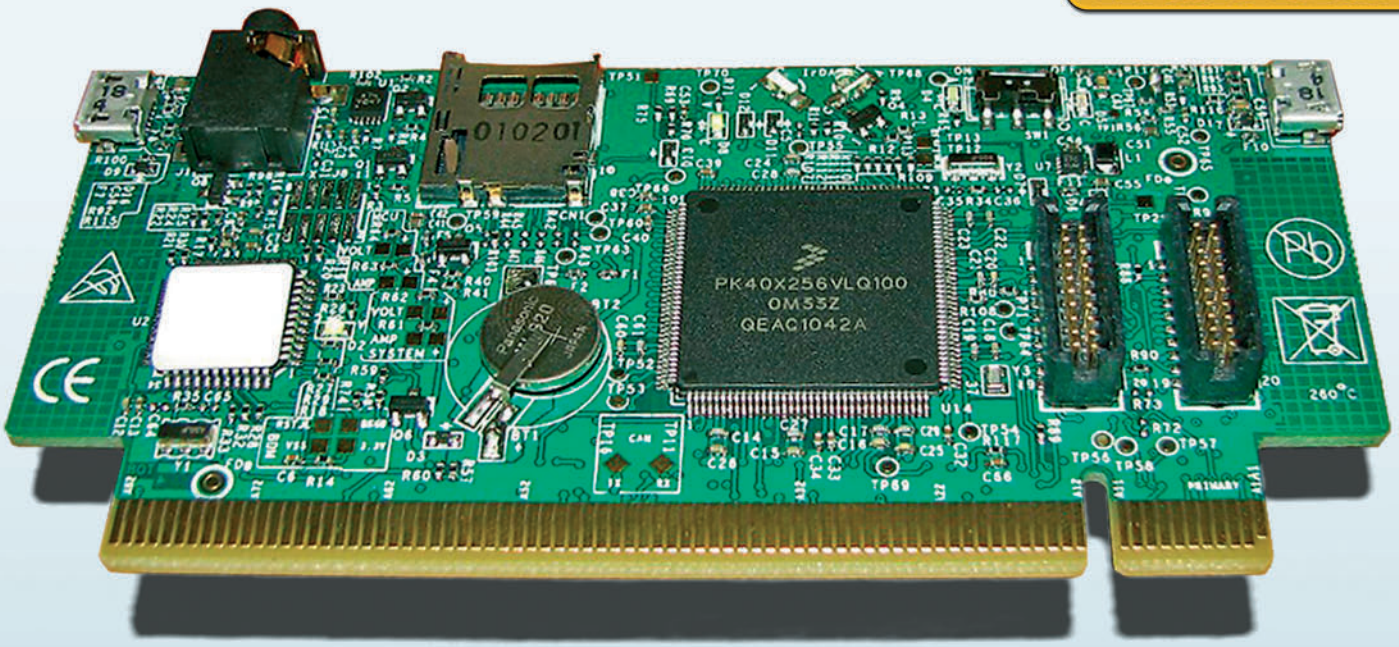
Ano 47 Nº 457 - 2011
Europa € 3,90
Brasil R$14,90

www.sabereletronica.com.br

SABER ELETRÔNICA

http://twitter.com/editora_saber

Industrial



DAC Conversor Digital-Analógico em Kinetis da Freescale

Minicurso LabView
parte 7

Curso de Arduino
parte 3

Medidas em RF Digital

Supervisão e Aquisição de Dados

Introdução a linguagem VHDL

Controle para Automação Industrial

ISSN 0101-6717

Nova Linha de Sensores Angulares GMR Infineon
TLE5010/5012 iGMR Giant Magnetic Resistive Sensor

O TLE5010/5012 é um sensor angular 360 ° que detecta a orientação de um campo magnético utilizando-se de elementos integrados de resistência magnética gigante (IGMR).



Características principais:

- Baseado no princípio GMR (Giant Magnetic Resistive)
- Sensor de campo magnético integrado para detecção e medição de ângulos
- Totalmente calibrado para medições angulares em 0 - 360 ° com contador de voltas e medição de velocidade angular
- Dois conversores separados de alta precisão: single-bit SD-ADCs
- Representação de 15 bits do valor absoluto do ângulo na saída (resolução de 0,01 °)
- Representação de 16 bits dos valores Seno/Cosseno na interface
- Erro angular de no máximo 1,0 ° ao longo da vida util com auto-calibração
- Interface SSC Bi-direcional SSC de até 8 Mbit / s
- Suporta Safety Integrity Level (SIL), com funções de diagnóstico e informações de status
- Interfaces: SSC, PWM, Incremental Interface (IIF), Hall-Switch Mode (HSM)
- 0,25 mM tecnologia CMOS
- Componente qualificado para uso Automotivo: -40 ° C a 150 ° C (temperatura de junção)
- ESD> 4 kV (HBM)
- RoHS-compliant (Encapsulamento Pb-free)

Aplicações típicas:

O sensor GMR TLE5012 foi projetado para o sensoreamento da posição angular em aplicações automotivas e industriais, tais como:

- Comutação de motores elétricos (por exemplo, usado em direção elétrica assistida (EPS))
- Interruptor de rotação
- Medição angular da direção do veículo
- Sensoreamento Angular

A Infineon oferece uma variedade de sensores analógicos e digitais para medição de diversos parâmetros: Pressão, campo magnético, posição, distância, direção e velocidade.

Saiba mais sobre a linha de sensores Infineon para aplicações Automotiva, Industrial e de Consumo. Acesse: www.infineon.com/sensors

Infineon

(11) 3372-9240

vendas.brasil@infineon.com

www.infineon.com

Distribuidores:

ARROW BRASIL S/A.

(11) 3613-9300

vendas@arrowbrasil.com.br

www.arrowbrasil.com.br

AVNET do Brasil

(11) 5079-2150

vendas@avnet.com

www.em.avnet.com

Farnell Newark

(11) 4066-9400

rnetto@farnellnewark.com

www.farnellnewark.com.br

Editora Saber Ltda.
Diretor
Hélio Fittipaldi

SABER ELETRÔNICA
www.sabereletronica.com.br
twitter.com/editora_saber

Editor e Diretor Responsável
Hélio Fittipaldi

Conselho Editorial
João Antonio Zuffo,

Redação
Elizabete Rossi

Revisão Técnica
Eutíquio Lopez

Colaboradores
Alexandre Capelli
Augusto Einsfeldt
César Cassiolato
Edriano Carlos de Araújo
Eutíquio Lopez
Filipe Pereira
Luís Fernando F. Bernabe
Newton C. Braga

Designers
Carlos Tartaglioni,
Diego M. Gomes

Publicidade
Caroline Ferreira

PARA ANUNCIAR: (11) 2095-5339
publicidade@editorasaber.com.br

Capa
Freescale Semicondutores/Divulgação

Impressão
Parma Gráfica e Editora

Distribuição
Brasil: DINAP
Portugal: Logista Portugal tel.: 121-9267 800

ASSINATURAS

www.sabereletronica.com.br
fone: (11) 2095-5335 / fax: (11) 2098-3366
atendimento das 8:30 às 17:30h

Edições anteriores (mediante disponibilidade de estoque), solicite pelo site ou pelo tel. 2095-5330, ao preço da última edição em banca.

Saber Eletrônica é uma publicação bimestral da Editora Saber Ltda, ISSN 0101-6717. Redação, administração, publicidade e correspondência: Rua Jacinto José de Araújo, 315, Tatuapé, CEP 03087-020, São Paulo, SP, tel./fax (11) 2095-5333.

Associada da:

anatec
www.anatec.org.br

Associação Nacional das Editoras de Publicações Técnicas, Dirigidas e Especializadas

Tecnologia e Custos


Hélio Fittipaldi

Nos últimos anos os fabricantes de microcontroladores estão aderindo ao core ARM, tendendo este a se tornar um padrão de mercado. A concorrência obriga os fabricantes a perseguirem a meta de ter microcontroladores cada vez mais rápidos, eficientes e baratos se quiserem sobreviver. Apresentamos nos últimos tempos alguns desenvolvimentos com micros de diversos fabricantes e, agora, descrevemos o Kinetis (da Freescale) em um circuito DAC – Conversor Digital-Analógico elaborado pelo engenheiro e professor Edriano Carlos de Araújo.

O comportamento da indústria elétrica e eletrônica durante o ano de 2011 foi analisado pela ABINEE, a qual apresentou um quadro preocupante e requer urgente intervenção do Congresso Nacional para trabalharem em novas leis que diminuam a altíssima carga de impostos e burocracia do poder público, o que provoca um Custo Brasil que transfere para o exterior os nossos empregos. O desemprego em grande volume no Brasil ainda poderá ser evitado, se os parlamentares e o poder executivo souberem agir coordenadamente e sem o mesquinho jogo político.

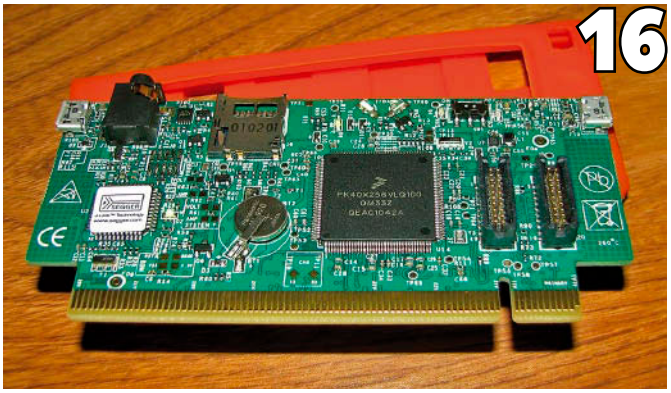
De acordo com os números divulgados, o faturamento da indústria eletroeletrônica, neste ano, crescerá 8% na comparação com 2010, totalizando R$ 134 bilhões, o que frustra as expectativas anteriores que projetavam, inicialmente, um crescimento de 13%, mas ainda é bom se comparado com outros países em crise. Este crescimento não espelha um aumento da produção local mas sim, na maioria dos casos, um aumento de importações de produtos que eram fabricados aqui no Brasil, mas que devido aos altos custos gerados pela burocracia e impostos governamentais transferem empregos para o exterior. Só para ilustrar com um componente que não requer alta tecnologia, temos os para-raios que em 2010 tiveram um volume de importação de US$ 10,5 milhões. Em 2011 houve um aumento de 113%, atingindo US$ 22,5 milhões.

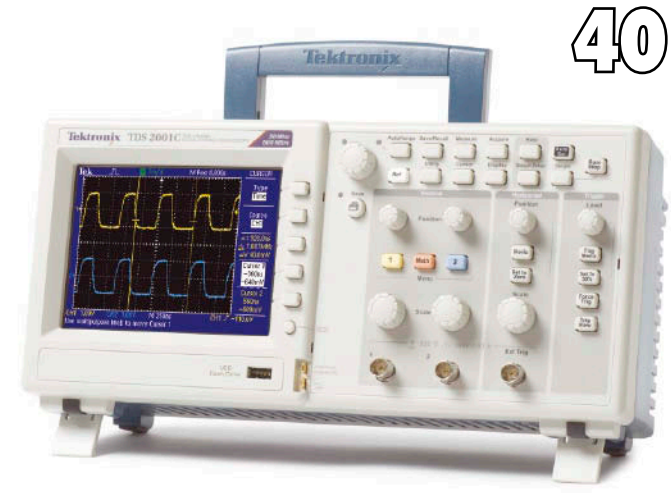
Submissões de Artigos
Artigos de nossos leitores, parceiros e especialistas do setor serão bem-vindos em nossa revista. Vamos analisar cada apresentação e determinar a sua aptidão para a publicação na Revista Saber Eletrônica. Iremos trabalhar com afinco em cada etapa do processo de submissão para assegurar um fluxo de trabalho flexível e a melhor apresentação dos artigos aceitos em versão impressa e online.

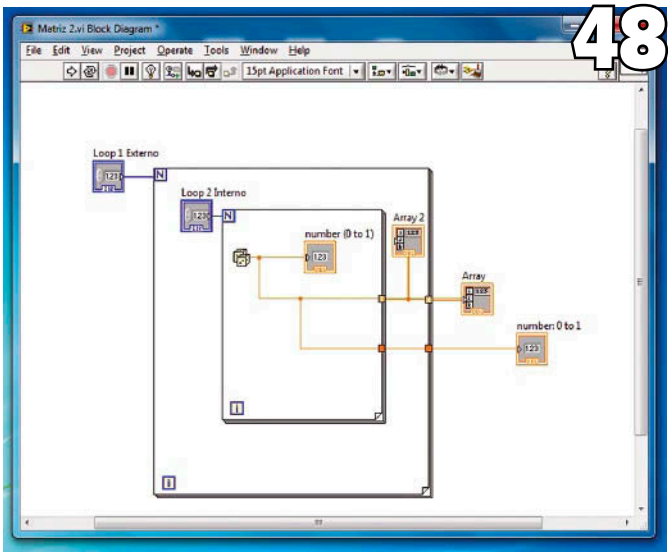
Atendimento ao Leitor: atendimento@sabereletronica.com.br

Os artigos assinados são de exclusiva responsabilidade de seus autores. É vedada a reprodução total ou parcial dos textos e ilustrações desta Revista, bem como a industrialização e/ou comercialização dos aparelhos ou idéias oriundas dos textos mencionados, sob pena de sanções legais. As consultas técnicas referentes aos artigos da Revista deverão ser feitas exclusivamente por cartas, ou e-mail (A/C do Departamento Técnico). São tomados todos os cuidados razoáveis na preparação do conteúdo desta Revista, mas não assumimos a responsabilidade legal por eventuais erros, principalmente nas montagens, pois tratam-se de projetos experimentais. Tampouco assumimos a responsabilidade por danos resultantes de imperícia do montador. Caso haja enganos em texto ou desenho, será publicada errata na primeira oportunidade. Preços e dados publicados em anúncios são por nós aceitos de boa fé, como corretos na data do fechamento da edição. Não assumimos a responsabilidade por alterações nos preços e na disponibilidade dos produtos ocorridas após o fechamento.

índice







Componentes

12 Conheça os Ultracapacitores

Microcontroladores

16 DAC - Conversor Digital-Analítico - no microcontrolador Kinetis, da Freescale

Eletrônica Aplicada

23 Curso sobre Arduino - Parte 3

30 Controle para Automação Industrial PLL, P, D, PID - Uma abordagem prática sobre essas técnicas

38 Reduzindo a EMI com Componentes Magnéticos

Instrumentação

40 Curso sobre Osciloscópio - Parte 2

45 Medidas em RF Digital

48 Minicurso LabView - Parte 7

Projetos

54 Sistemas de Supervisão e Aquisição de Dados

Desenvolvimento

58 Introdução ao VHDL

64 Fonte Fixa de 12 / 12,6 / 13,2 V x 10 A

Editorial 03

Acontece 06

Reportagem 08

Índice de anunciantes

Samtec 5
Tato 9
Patola 9
Globtek 15
Metaltex 21
Honeywell 29
Mosaico 51
Cika 53
NovaSaber 63
Infineon ..... 2ª capa
National Instruments ..... 3ª capa
Texas Instruments ..... 4ª capa

VARIEDADE.
VERSATILIDADE.
VELOCIDADE.

PRODUTO

- Ampla variedade de interconexões de placas com empilhamento
- Passo de 0,4 mm a 5,08 mm
- Perfil baixo (4 mm) a elevado (30 mm)
- Sistemas robustos
- Alimentação para a placa
- IDC, fiação discreta, circuitos flexíveis
- Alta velocidade e micro

CONNECTORS · MICRO · SIGNAL INTEGRITY · CUSTOMS · CABLES · PANEL & I/O · RUGGED · SUDDEN SERVICE®

ATENDIMENTO

WORLD DIRECT

- Remessas diretas para todo o mundo, 3-4 dias úteis até sua porta
- Os menores prazos de entrega da industria
- Liberação na alfândega pela Samtec America Latina
- Preços globais/amostras globais
- Todas as amostras enviadas em 24 horas
- Solutionator on-line de projeto/construção/ pedidos

SUDDEN samtec SAMPLE

SOLUTIONATOR®
DESIGN IN A MINUTE • SHIP IN A WEEK



www.samtec.com

SAMTEC AMERICA LATINA COMERCIO EXTERIOR LTDA
Rua Alagoas Nr 1460 • Sala 805 • Bairro Savassi • 30130-160 Belo Horizonte / MG • Brasil
Tel: +55 31 3786 3227 • Fax: +55 31 3786 3229 • E-mail: brazilsales@samtec.com

Agilent Technologies apresenta a 1ª Aplicação
com Osciloscópio para o Protocolo Ethernet 64b/66b

A empresa norte-americana acaba de aumentar o seu portfólio de aplicações com osciloscópio, apresentando o software de decodificação e "triggering" para o protocolo Ethernet 64b/ 66b. A série Infinium, da Agilent, é a primeira e única família de osciloscópios capaz de suportar a decodificação e o gatilhamento desse protocolo.

Até os dias de hoje, as poucas soluções disponíveis para esse protocolo, utilizando osciloscópio, destinavam-se a barramentos como I²C, SPI, CAN ou esquemas de *decoding* baseados em 8b/ 10b. Os projetistas de 64b/ 66b trabalhavam com dificuldade em seus projetos, obrigados a fazerem a decodificação do protocolo manualmente.

A aplicação N8815A Ethernet 10Gbase-KR faz a decodificação em tempo real, permitindo que as equipes de projeto desenvolvam produtos de qualidade superior em menos tempo.

O protocolo Ethernet 64b/ 66b é utilizado principalmente pelas atuais tecnologias de transmissão e *networking* de dados. FibreChannel, Ethernet e Infiniband consistem nas três tecnologias que se beneficiam do protocolo citado, justamente pelo seu menor custo.

O software N8815A foi desenvolvido para ajudar os projetistas das tecnologias 10Gbase-KR e 40Gbase-KR, definidas pelo padrão IEEE 802.3, as quais operam sobre vias de PCIs backplane.A nova aplicação compartilha a vantagem trazida pela série Infinium, no que se refere à busca e decodificação serial múltipla, obtida recentemente, o que permite aos projetistas observarem 4 dessas vias simultaneamente.

Na maioria das indústrias eletrônicas americanas há equipes de projetistas e testadores que produzem projetos de hardware com uma crescente sofisticação. Isso se vê nas fábricas de computadores, telecomunicações , semicondutores, Espaço Aéreo/ Defesa, automóveis e *wireless*. Os osciloscópios consistem na ferramenta inicial dos engenheiros para testes e "debug" dos seus projetos, portanto, esses instrumentos precisam ter suas capacidade gerais de medida melhoradas continuamente de modo que os profissionais possam entregar produtos mais robustos.

Além de decodificar os sinais 64b/ 66b, o software N8815A também identifica apropriadamente sinais PRBS e seus comprimentos. Quando os projetistas querem ter certeza de que estão passando realmente um padrão PRSB de um determinado comprimento para o seu sinal, o N8815A permite a eles apurarem o padrão desejado e o seu comprimento exato. A aplicação também leva em conta correções de erros precoces.

A aplicação N8815A Ethernet 10GBase-KR é suportada pelos osciloscópios Infinium das séries 90.000 e 90.000X, da Agilent, sendo que esta última família possui os mais rápidos osciloscópios em tempo real da indústria, alcançando uma largura de faixa de 33 GHz. Esses novos equipamentos apresentam o menor ruído de fundo e menor "jitter" entre os osciloscópios industriais, garantindo assim uma precisão de medida superior.

Produtos

Fontes "Green Power" da Globtek cumprem os requisitos de Eficiência – Nível V

A série GT-41135 de Fontes de Alimentação tipo Plug-in, com Lâminas intercambiáveis e entradas IEC, está disponível para venda, tanto no modelo de lâmina única (*wall plug-in*) quanto para comercialização na forma de kits, nas seguintes regiões do mundo: América do Norte, Europa, Reino Unido, Austrália, China, Coreia, África do Sul e Argentina.

As fontes possuem configurações mecânicas com dupla isolação obrigatória, fornecem tensões de saída reguladas desde 5,0 até 48,0 VDC em incrementos de 0,1 VDC: e proveem potência de saída contínua até 12 W. Os modelos fechados são alojados em invólucros de policarbonato sem aberturas de ventilação, resistentes a impactos, com refrigeração térmica, nas dimensões de 43,5 x 74 x 35,3 (mm).

Essa família de fontes caracteriza-se por apresentar: Tensão de Saída regulada com baixíssimo "ripple", Especificações de Proteção internas contra sobrecorrente, sobretensão, sobretemperatura e outras. A série GT-41135 admite Tensão de Entrada Universal de 90 a 264 VAC, e obedece aos requisitos das normas IEC/EN60950, de Laborátorios Certificados (UL e cUL), bem como de outras agências internacionais de segurança de modo a cumprir as Regulações de EMI/RFI e as Diretivas de EMC/CE e FCC Classe B para aplicações ITE.

Todos os modelos trazem os "logos" das agências e a marcação CE. Relatórios , incluindo CB, são produzidos por laboratórios certificados independentes. Projetos customizados e/ou modificados de acordo com o cliente também são possíveis.A Globtek oferece agora uma garantia de 5 anos para os produtos desta série.

1ª Apresentação do Ambiente completo de
Produção de Baterias de Alto Desempenho

A Productronica, principal feira mundial de novos produtos eletrônicos, acaba de apresentar o seu show especial "Battery Manufacturing and power electronics", organizado em estreita colaboração com a Associação Produtrônica na Federação Alemã de Engenharia – VDMA, com a RWTH Aachen University, o Instituto Fraunhofer para Tecnologia da Produção e outras empresas líderes.

O ambiente de produção da bateria completa, desde o revestimento do material do eletrodo até o módulo concluído, foi apresentado pela primeira vez como parte de um evento comercial. Dessa forma , a Digatron Industrie-Elektronik, a ERSA, F&F Devoltec, Automação IBG, KUKA Roboter, KMS Automação, M+W Group, Manz, OMT, Reis Robotics, Siemens, UTL, e Viscom expuseram suas novas soluções para a cadeia do processo de fabricação em uma área de 200 m². O show especial foi realizado no Hall B2 da Productronica.

Caracterizadas pelo alto desempenho, as baterias para os carros elétricos ou energias renováveis diferem muito das baterias para *laptops* ou telefones celulares, por exemplo, em termos de confiabilidade, tempo de vida útil e também nos tamanhos. Isso, por si só, exige enormes avanços na produção, o tema- chave da Productronica. A solução para o maior desafio, ou seja, a redução de custos, requer uma estreita cooperação entre fabricantes, fornecedores, empresas de engenharia mecânica e institutos de pesquisa.

A Productronica 2011 reuniu todos os participantes importantes do mercado e estabeleceu novas normas com respeito à interação entre máquinas, processos, tecnologias de automação, produtos intermediários, etc. e os produtos finais foram demonstrados durante o show especial. Com a ajuda de especialistas em "virtual factory"da RWTH Aachen University, os visitantes da feira foram capazes de construir interativamente uma fábrica de baterias e até mesmo desenvolver alguns cenários de custos. Além disso, empresas e institutos comprometidos com o tema divulgaram as últimas informações sobre mercados, tecnologias e estratégias relativas à produção de baterias no Fórum da Inovação.

A seguir, alguns nomes de Especialistas que participaram da feira Mundial Productronica 2011: Dr. Eric Maiser, Managing Director da VDMA Productronica e Chefe do Grupo Industrial de Produção de Baterias do Fórum VDMA E-Motive; Prof. Dr. Achim Kampker, da RWTH Aachen University; Guido Feind, Chefe de Vendas da Siemens/ Sistemas de Automação Industrial; Dr. Michael Wenzel, Managing Director da Holding Grupo Reis; Rainer Kurtz, Managing Partner da Ersa e Presidente do Comitê Técnico Consultivo da Productronica; Norbert Bargmann, Presidente do Conselho de Administração da Messe München Gmbh.

Com efeito a partir do outono de 2012, a Messe München GmbH ampliará o seu portfólio na Divisão de Novas Tecnologias; a nova feira de negócios orientados para o futuro " Energy Storage" irá abranger a cadeia completa para as tecnologias de armazenamento de energia e novas baterias. A "Energy Storage" será realizada anualmente ao mesmo tempo que a Productronica, no Novo Centro de Feiras de Negócios de Munique, acompanhada por conferências que durarão vários dias. Ela deverá ter início no dia 13 de novembro e estendendo-se até 16 de novembro de 2012.

"PCIM - América do Sul"
será lançada no Brasil

A PCIM – América do Sul será realizada pela primeira vez em São Paulo (Brasil), de 11 a 13 de setembro de 2012.

Após o sucesso da PCIM - Europa e da PCIM - Ásia, a empresa organizadora "Mesago PCIM GmbH" lançará o novo evento no Brasil visando aumentar o alcance internacional da principal Exposição/Conferência sobre produtos de eletrônica de potência, movimento "inteligente", energias renováveis e controle da energia.

O crescente mercado de produtos eletrônicos no Brasil oferece grandes oportunidades, tanto para empresas nacionais como internacionais, uma vez que existe um aumento na procura de informações, troca de ideias e possibilidades de mostrar conceitos e soluções. Com a PCIM - América do Sul, a Mesago oferece às empresas e aos especialistas em tecnologias e ciências aplicadas uma plataforma ideal para atenderem às três demandas citadas simultaneamente, comenta o Sr. Udo Weller – presidente da empresa organizadora do evento.

A fórmula - conceito já comprovada anteriormente, combinando uma exposição de temas dirigidos com conferências orientadas para as aplicações em curso, será replicada na PCIM - América do Sul. O foco principal será voltado para os semicondutores de potência, componentes passivos, gerenciamento térmico, os novos materiais e suas aplicações na área de "movimento", energias renováveis e controle da energia.

O Conselho de Especialistas para a PCIM - América do Sul, formado por 26 membros, é liderado pelo Prof. Marcelo Lobo Heldwein do Instituto de Eletrônica de Potência – INEP, da Universidade de Santa Catarina (Brasil) em conjunto com o Prof. Ivo Barbi, chefe do INEP e com o Prof. Leo Lorenz da Infineon Technologies.

Hélio Fittipaldi

Setor eletroeletrônico cresce em 2011, mas importações preocupam

Humberto Barbato, presidente da ABINEE e os diretores da entidade apresentaram os números do faturamento da indústria eletroeletrônica, em 2011 que crescerá 8% em comparação com 2010, totalizando R$ 134 bilhões.

As expectativas do início do ano previam um crescimento de 13%. Já a balança comercial, do setor, deverá fechar com um déficit de mais de US$ 32 bilhões, o que significa crescimento de 18% em relação ao ano passado. Esse déficit é resultado das importações que deverão atingir US$ 40 bilhões e das exportações que não chegarão aos US$ 8 bilhões. No transcorrer do ano, as perspectivas de se atingir 13% em 2011 foram aos poucos diminuindo como mostra a figura 1.

O diretor da área de Pesquisas e Estatísticas Econômicas da ABINEE o economista Luiz Cezar Elias Rochel em sua apresentação mostrou O quadro do Desempenho da Industria Elétrica e Eletrônica – 2011 (figura 02) e observou que os estoques não tem variado tradicionalmente e um ano para o outro. Isto quer dizer que, o que aumentou, foi mesmo comercializado. Podemos observar que o investimento caiu 14,8% e também houve queda de produção física de produtos.

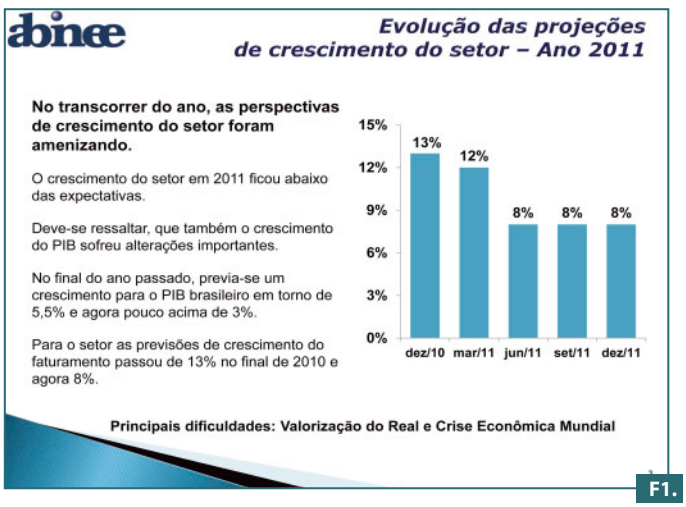
Na figura 3, mostra o Índice de Produção e Importação de Produtos Elétricos e Eletrônicos em 2011 onde podemos notar que a produção (92,6) é menor que a importação (127,4). Estamos perdendo participação pois houve queda de 7,4% de produção física e cresceu 21% a importação.

Em 2008 35% dos produtos eletroeletrônicos eram importados, hoje são 42%.

Segundo Barbato, há alguns anos o faturamento refletia o nível de atividade das empresas, hoje, porém, este índice não está refletindo a situação efetiva pela qual passa o setor eletroeletrônico. "É preciso analisar, o quanto do mercado interno está sendo atendido por produto local ou por importação", disse.

Incentivadas pelo real valorizado, as importações de bens finais representaram 22% do faturamento total do setor, o que evidencia a dificuldade das empresas para competir. Para o presidente da ABINEE, a situação já chegou a tal limite que apenas mexer no câmbio talvez não seja suficiente para reverter esta perda de competitividade. "O que precisamos é criar um programa de incentivos para estimular a compra de produtos de empresas instaladas no Brasil. As estatais e operadoras têm que ter vantagens para comprar daqui", disse. (Figura 05 e 06)

Citando o ex-ministro Delfim Netto que, em reunião do Conselho Consultivo da ABINEE, abordou a necessidade do Brasil criar empregos para uma população



F1.

abinee

Desempenho da Indústria Elétrica e Eletrônica - Ano 2011*

Indicadores	Total	Var %
Faturamento (R$ milhões)	134.904	8%
Faturamento (US$ milhões)	80.883	14,4%
Exportações (US$ milhões)	7.880	3,4%
Importações (US$ milhões)	40.093	14,9%
Saldo (US$ milhões)	- 32.213	18,2%
Emprego (milhares)	183,0	4,8%
Investimentos (R$ milhões)	3.035	- 14,8%
Investimentos (% do faturamento)	2%	-
Produção IBGE	-	- 1,5%
- Elétricos	-	- 2,0%
- Eletrônicos	-	- 1,0%

* projeção

F2.

abinee

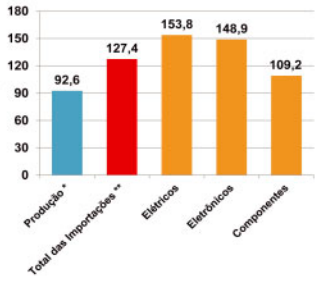
Índice de Produção e Importação de Produtos Elétricos e Eletrônicos em 2011
Base: 2008 = 100

O índice de "quantum" mostra com mais clareza o impacto das importações sobre o mercado interno.

Por meio do índice de "quantum", permite-se identificar efetivamente o crescimento das importações pois elimina os efeitos dos preços e da variação cambial.

Desta forma, considerando, a queda de produção de 7,4% no período acumulado de 2008 a 2011, e o aumento das importações de produtos elétricos e eletrônicos, fica evidente o aumento da participação dos bens importados no mercado brasileiro.

No período de 2008 a 2011, a desvalorização do dólar foi de 9%.



* Produção IBGE – Agregação ABINEE
** Importações – MDIC/SECEX

F3.

economicamente ativa de 130 milhões de habitantes em 2030, Barbato afirmou que com o atual processo de destruição do parque industrial instalado no país, não haverá capacidade para absorver esta mão de obra potencial.

Na ocasião, Barbato voltou a ressaltar que a realização da Copa do Mundo e das Olímpiadas no país deve servir como uma alavanca de oportunidades para as empresas instaladas no país.

Um dos setores mais afetados pelo câmbio é o de Geração, Transmissão e Distribuição de Energia Elétrica (GTD), cujas importações atingiram um crescimento de 54% em 2011. O diretor da área na ABINEE, Newton Duarte, afirmou que os números

abinee

Importações de Produtos da área de Geração, Transmissão e Distribuição de Energia Elétrica - GTD (US$ mil)

Produtos	2010	2011*	2011* X 2010
Transformadores	170.001	247.361	46%
Geradores	42.074	105.548	151%
Seccionadores	53.319	75.230	41%
Disjuntores para GTD	54.313	70.771	30%
Isoladores	30.681	52.014	70%
Cabos para GTD	28.016	41.032	46%
Para-raios	10.559	22.540	113%
Capacitores	8.285	18.876	128%
Medidores de Eletricidade	8.174	8.932	9%
Fusíveis para GTD	1.632	4.389	169%
Painéis e Quadros	1.845	3.350	82%
Outros GTD	122.958	168.253	37%
Total da área de GTD	531.859	818.296	54%

* projeção Fonte: MDIC/SECEX

F4.

abinee

Importações de Produtos Elétricos e Eletrônicos (US$ milhões)

Áreas	2009	2010	2011*	2011* X 2010
Automação Industrial	2.016	2.528	2.886	14%
Componentes	12.922	18.248	20.788	14%
Equipamentos Industriais	2.724	4.023	4.359	8%
GTD	496	532	818	54%
Informática	1.763	2.404	2.694	12%
Material de Instalação	874	1.410	1.602	14%
Telecomunicações	2.332	2.867	3.884	35%
Utilidades Domésticas	1.826	2.869	3.061	7%
Total	24.953	34.882	40.093	15%

* projeção Fonte: MDIC/SECEX

F5.

abinee

Produtos mais importados do Setor (US$ milhões)

Produtos	2010	2011*	2011* X 2010
Comp. p/ Telecomunicações	4.533	5.701	26%
Semicondutores	4.464	5.107	14%
Componentes para Informática	3.351	2.943	-12%
Instrumentos de Medida	1.304	1.604	23%
Eletrônica Embarcada	1.264	1.566	24%
Comp. p/ Equips. Industriais	869	1.084	25%
Telefones Celulares	561	987	76%
Maq. p/ Processamento Dados	761	970	27%
Grupo Motogerador	1.008	865	-14%
Aparelhos Eletromédicos	805	763	-5%

* projeção Fonte: MDIC/SECEX

F6.

abinee

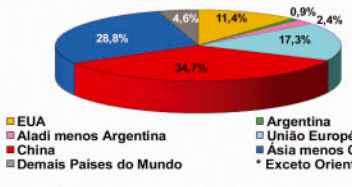
Origem das Importações

Além da China, as importações da União Europeia apresentaram expressivo crescimento.

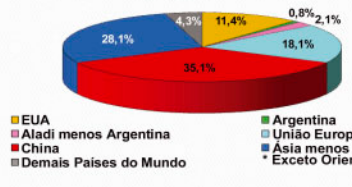
As importações da China continuam ganhando participação na pauta de importações de produtos elétricos e eletrônicos, passando de US$ 12 bilhões, em, 2010, para US$ 14 bilhões, em 2011.

Cabe destacar o crescimento de 20% das importações originárias da União Europeia no ano de 2011 (US$ 7,2 bilhões) em relação às realizadas em 2010 (US$ 6,0 bilhões).

Ano 2010

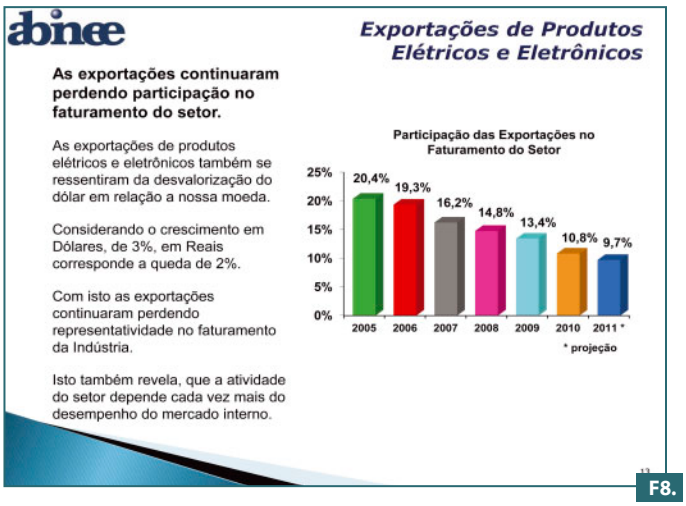


Ano 2011**

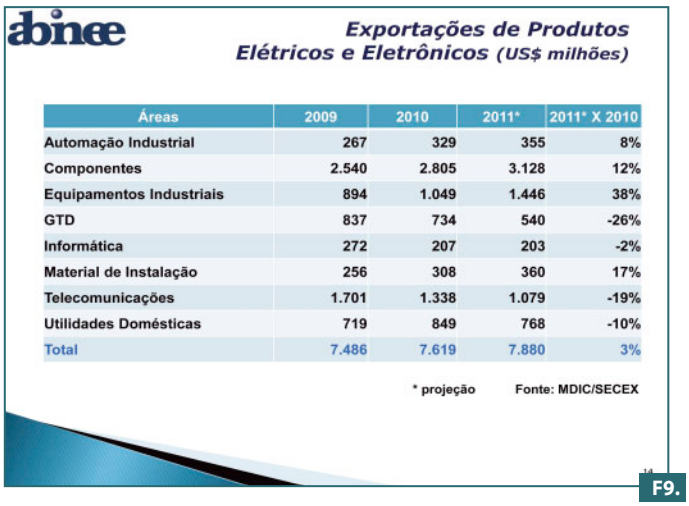


** projeção Fonte : MDIC/SECEX

F7.



F8.



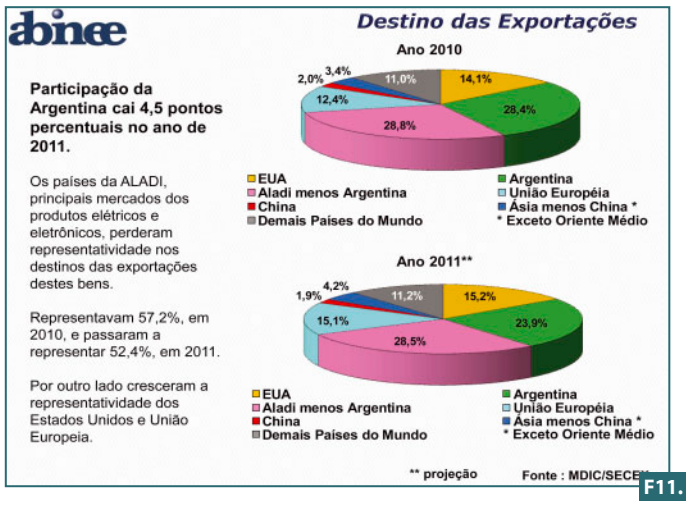
F9.

mostram uma tendência 'horripilante'. "Partimos de um quadro de superávit, no qual importávamos metade do que exportávamos e, agora, pela primeira vez, fomos deficitários. Isto aponta que as empresas não têm mais competitividade para atender os projetos daqui".

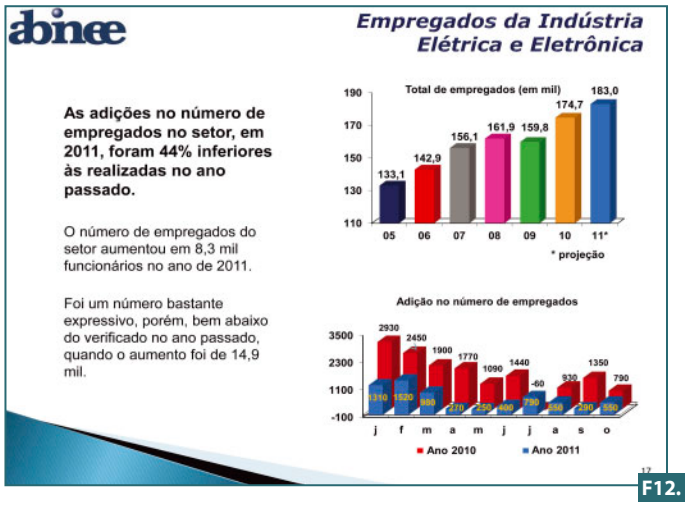
O diretor da área de informática, Hugo Valério, afirmou que o problema de competitividade no país não é recente, mas vem se agravando por conta do Custo Brasil e do câmbio, que expõe ainda mais os gargalos estruturais. "Dessa forma, as empresas passam a buscar maior automação e transferência da produção para outras localidades com menor custo de mão-de-obra, o que pode gerar uma queda de empregos no país", ressaltou. Para ele, é necessário que o governo encare esta situação por um prisma estratégico.



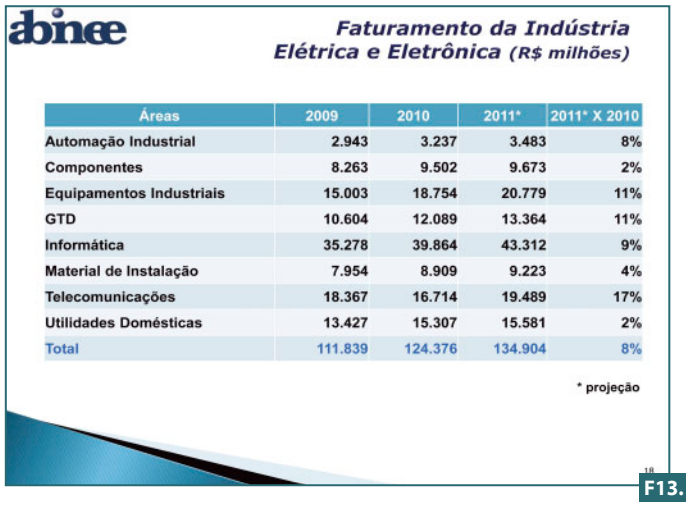
F10.



F11.



F12.



F13.

abinee

Projeções para a Indústria Elétrica e Eletrônica - Ano 2012

Indicadores	2011	2012	Var %
Faturamento (R$ milhões)	134.904	152.534	13%
Faturamento (US$ milhões)	80.883	82.264	2%
Exportações (US$ milhões)	7.880	8.275	5%
Importações (US$ milhões)	40.093	46.100	15%
Saldo (US$ milhões)	-32.213	-37.825	17%
Emprego (milhares)	183	186	2%
Investimentos (R$ milhões)	3.035	3.431	13%
Investimentos (% do faturamento)	2%	2%	-

F14.

abinee

Projeções para o Faturamento da Indústria Elétrica e Eletrônica (R$ milhões)

Áreas	2011	2012	2012 X 2011
Automação Industrial	3.483	3.971	14%
Componentes	9.673	9.867	2%
Equipamentos Industriais	20.779	23.065	11%
GTD	13.364	15.636	17%
Informática	43.312	47.643	10%
Material de Instalação	9.223	9.682	5%
Telecomunicações	19.489	26.310	35%
Utilidades Domésticas	15.581	16.360	5%
Total	134.904	152.534	13%

F15.

Segundo o diretor da área de telecomunicações da entidade, Paulo Castelo Branco, o processo de desindustrialização já está chegando ao segmento. "As importações representaram 33% do faturamento das empresas neste ano", enfatizou.

Ele acrescentou, ainda, que, para atenuar o arrefecimento nas vendas de equipamentos, as empresas estão transferindo seus negócios para a parte de serviços.

Destacando os trabalhos da ABINEE com o objetivo de fortalecer da indústria de componentes, o diretor da entidade, Aluízio Byrro, afirmou que sem o desenvolvimento de uma indústria desta área no Brasil, a agregação de valor dos bens finais continuará reduzido.

O diretor Newton Duarte, disse que precisamos dar isonomia à empresa brasileira. Impostos, investimentos, juros, câmbio semelhantes aos nossos concorrentes estrangeiros que não pagam isto, como os chineses. Almoçar hoje em São Paulo é mais caro do que em diversas cidades do mundo. Os funcionários dos restaurantes ganham mais hoje? Não. Então por que isto? É o custo Brasil.

Precisamos de uma boa estratégia para o Brasil. A estratégia da marolinha, da conversa fiada não dá mais. A presidente Dilma precisa agora por ordem na casa.





Conheça os Ultracapacitores

Quando nos referimos aos veículos autônomos e robôs móveis, a primeira forma de fonte de energia que nos vem à mente para alimentar esses equipamentos é a bateria. No entanto, as baterias têm limitações, e em busca de novas formas de fontes de energia algumas delas têm se revelado muito interessantes em diversas aplicações. Uma delas é o ultracapacitor. Com densidade de armazenamento altíssima e uma capacidade de corrente extremamente elevada, essa nova fonte de energia para alimentar robôs, veículos e outros equipamentos poderá abrir novos campos para o projetista. Veja neste artigo o que são os ultracapacitores.

Newton C. Braga

Até hoje a melhor forma de obtermos energia barata e em boa quantidade para movimentar robôs, veículos e alimentar outros equipamentos elétricos e eletrônicos, é conseguida através de baterias.

Essas células químicas convertem energia química em energia elétrica em um fluxo constante com um bom rendimento, mas possuem algumas limitações.

Uma bateria tem a corrente máxima limitada pela sua resistência interna. Assim, ela não pode fornecer picos de corrente elevados, o que pode ser importante em algumas aplicações.

Se uma bateria tiver que fornecer uma corrente muito intensa por um curto intervalo de tempo, a sua tensão cairá e a maior parte da energia passará a ser dissipada no seu interior, conforme ilustra a **figura 1**.

Isso significa que elas são boas para fornecer energia em pequenas doses por intervalos prolongados de tempo. Entretanto, não é somente em baterias que podemos armazenar energia elétrica. Um dispositivo que também pode armazenar cargas elétricas sob determinada tensão é o capacitor.

O Capacitor

Um capacitor é formado por duas placas de metal separadas por um material isolante denominado dielétrico. O material de que é feito o dielétrico determina seu nome: capacitores de mica, cerâmica, poliéster, policarbonato, eletrolítico, etc.

Na **figura 2** temos a construção básica de um capacitor.

A quantidade de cargas que um capacitor pode armazenar, ou seja, sua capacitância, depende da superfície das placas de metal (armaduras), da espessura do dielétrico e do material de que este é feito através do que denominamos "constante dielétrica".

Desse modo, para armazenar mais cargas os capacitores devem ter grandes superfícies de armadura e a espessura do dielétrico deve ser a menor possível.

Uma técnica muito usada para se conseguir boas capacidades de armazenamento, e portanto grandes capacitâncias, consiste em se enrolar o dielétrico e as armaduras na forma de tubos, obtendo-se assim os chamados capacitores tubulares, conforme mostra a **figura 3.**

Armazenando energia no capacitor

A energia armazenada em um capacitor depende tanto de sua capacitância quanto da tensão em que as cargas se encontram, observe a **figura 4.**

Essa energia pode ser calculada pela fórmula:

$$E = 1/2 \times C \times V^2$$

Onde:

E é a energia, em joules (J);

C é a capacitância do capacitor, em farads (F);

V é a tensão em que as cargas são mantidas, em volts (V).

Ocorre, entretanto, que os capacitores comuns não são capazes de armazenar muita energia. Os maiores capacitores comuns são os eletrolíticos.

Neles, o dielétrico consiste de uma finíssima camada de óxido de alumínio que é formada num eletrodo quando uma substância química denominada eletrólito a ataca, veja a **figura 5**.

Esse óxido, além de ser finíssimo tem uma constante dielétrica algo elevada, o que permite a obtenção de capacitâncias altas.

Os tipos comuns com tamanhos mostrados na **figura 6** podem ter capacitâncias de 1 µF a 100 000 µF, tipicamente.

Maiores densidades de armazenamento podem ser obtidas com materiais de constantes dielétricas maiores tais como o tântalo, dando origem assim à família dos capacitores de tântalo exibida na **figura 7**.

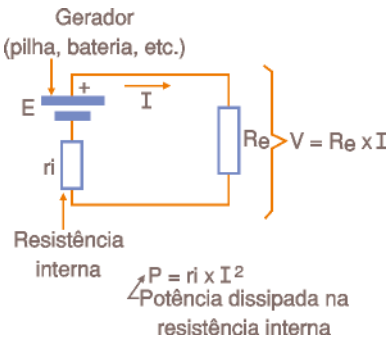
Mas, mesmo assim, a energia armazenada em um capacitor de 100 000 µF (0,1F) com uma tensão de 6 V é irrisória, quando comparada a uma pilha. Um capacitor desse tamanho armazena:

$$E = 1/2 \times 0{,}1 \times 36$$
$$E = 1{,}8 \text{ joules}$$

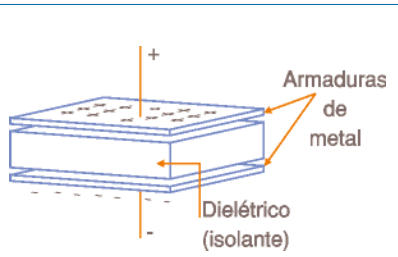
Isso significa que ele poderia alimentar uma lâmpada de 6 V x 100 mA (600 mW) por apenas 3 segundos!

Qual a vantagem de empregar capacitores como fontes de energia?

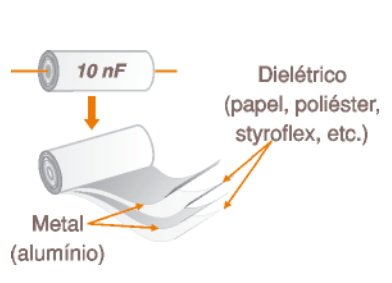
É evidente que com essa capacidade de fornecimento de energia, um capacitor não pode substituir uma pilha comum ou bateria capaz de alimentar a mesma lâmpada por horas seguidas.



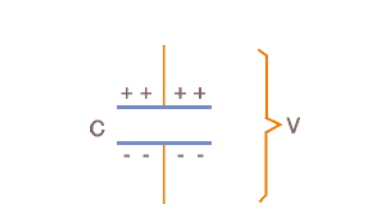
F1. Energia perdida devido à resistência interna do gerador.



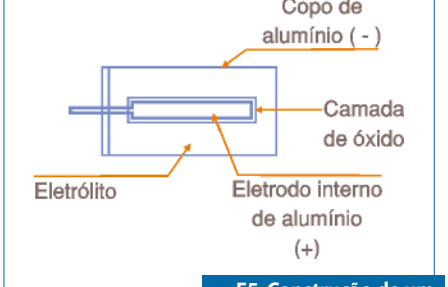
F2. O capacitor básico.



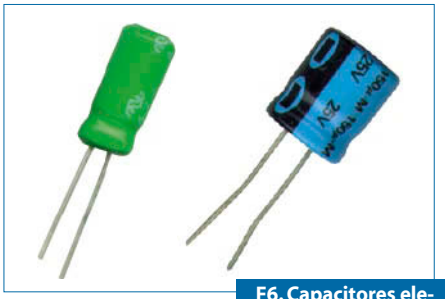
F3. Construção de um capacitor tubular.



F4. Um capacitor armazena energia



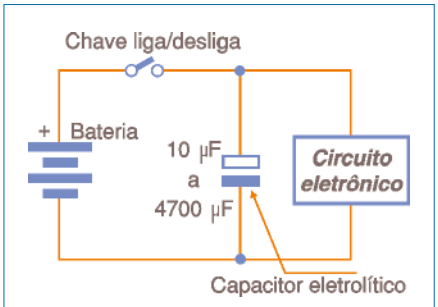
F5. Construção de um capacitor eletrolítico.



F6. Capacitores eletrolíticos comuns



F7. Capacitores de tântalo.



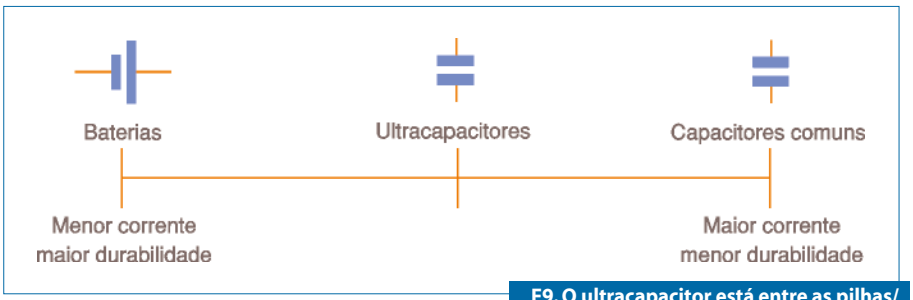
F8. O capacitor funciona como um reservatório de energia "extra"

No entanto, os capacitores possuem uma característica que as baterias não têm: baixa resistência interna.

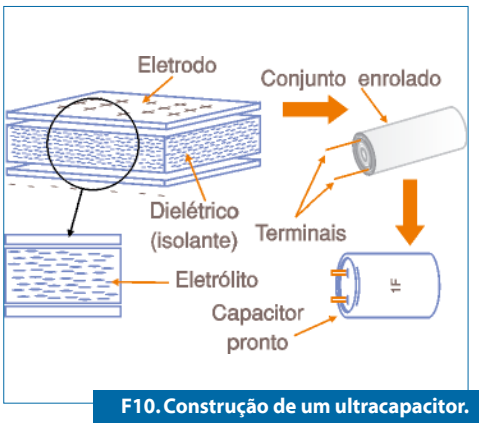
Os capacitores podem fornecer toda a energia armazenada de forma praticamente instantânea. Isso ocorre porque possuem uma resistência interna muito baixa que não limita a corrente como sucede no caso das baterias e pilhas.

Assim, o capacitor poderá ser útil se precisarmos fornecer uma corrente muito alta a um circuito por um curtíssimo intervalo de tempo.

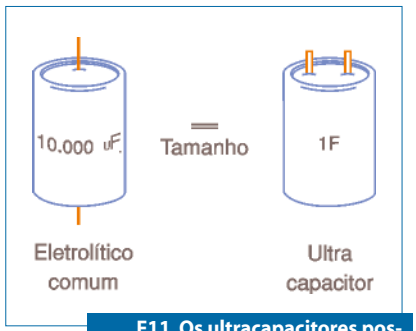
É por este motivo que, na maioria dos circuitos eletrônicos, em paralelo com a bateria ligamos um capacitor de alto valor, que é justamente para suprir os picos de corrente mais intensos em determinados momentos,



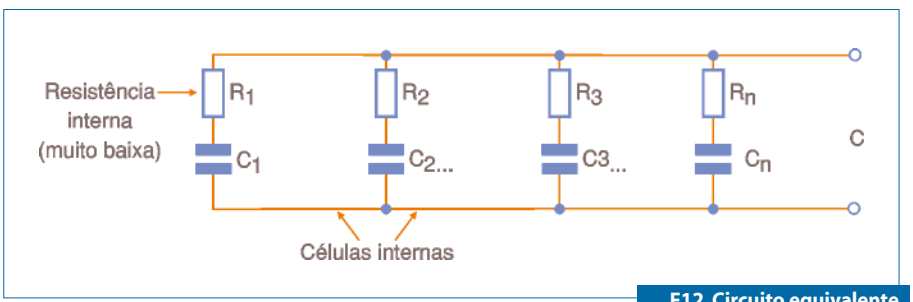
F9. O ultracapacitor está entre as pilhas/ baterias e os capacitores comuns.



F10. Construção de um ultracapacitor.



F11. Os ultracapacitores possuem capacidade de armazenamento 100 vezes maior que os eletrolíticos comuns.



F12. Circuito equivalente a um ultracapcitor.

o que não poderia ser feito com uma bateria comum, conforme mostra a **figura 8.**

Os Ultracapacitores

Se, por um lado, as baterias fornecem correntes pequenas durante intervalos de tempo longos e, por outro, os capacitores podem ser colocados para fornecer correntes intensas por curtíssimos intervalos de tempo, o projetista poderá, em breve, contar com uma solução intermediária: o ultracapacitor.

Empregando tecnologias apropriadas será possível multiplicar a densidade de armazenamento de energia e, com isso, fabricar capacitores de valores tão elevados que permitam o armazenamento de quantidades de energia comparáveis àquelas que uma pilha ou bateria comum é capaz de armazenar. Isso nos permitirá colocar o ultracapacitor como fonte de energia em um ponto intermediário entre o capacitor comum e a bateria, observe a **figura 9.**

Conforme já vimos, em um capacitor a capacitância é tanto maior quando maior é a superfície efetiva das armaduras e menor a espessura do dielétrico.

Para se obter capacidades de armazenamento enormes, o que se faz no ultrapacitor é utilizar um eletrodo poroso (à base de carbono), o qual é imerso numa substância condutora (eletrólito), veja exemplo na **figura 10**.

O eletrólito penetra nos poros do carbono e, ao reagir, forma uma película ultrafina, (da ordem de ângstrons) cobrindo internamente todos os poros e atuando assim como dielétrico.

A superfície total ocupada pelo dielétrico é enorme, uma vez que se espalha internamente por todos os poros. Pode-se, então, obter uma densidade de capacitância até 100 vezes maior do que a que seria possível com as tecnologias de fabricação dos capacitores eletrolíticos convencionais.

Em outras palavras, dois capacitores com mesmo tamanho e mesma tensão de trabalho, sendo um eletrolítico comum e o outro um ultracapacitor, mostram que este último tem uma capacitância 100 vezes maior, atente para a **figura 11.**

Outra característica importante que é obtida nos ultracapacitores é sua baixíssima resistência em série, conforme podemos ver pelo circuito equivalente da **figura 12,** o qual limita a corrente de pico de descarga.

Os ultracapacitores têm resistências da ordem de fração de milésimo de ohm. Assim, para um capacitor de 0,1 mohm com 2,5 V, é possível obter uma corrente de pico de 625 ampères! Um exemplo do que isso significa: um ultracapacitor do tamanho de uma moeda desenvolvido por uma empresa americana é capaz de armazenar energia suficiente para fornecer (por alguns segundos) a corrente necessária para a partida de um motor de caminhão!

Aplicações atuais e possíveis para os ultracapacitores:

- *Nobreak* - Os ultracapacitores podem manter o circuito em funcionamento por alguns segundos durante curtas interrupções no fornecimento de energia. Além disso, eles atuam como excelentes dispositivos para absorver surtos e transientes.
- Partida rápida para carro - Os ultracapacitores podem ser usados para a partida de carros sob quaisquer condições de tempo já que seu desempenho, diferentemente das baterias, não é afetado pela temperatura. Além do mais, o processo da partida contribui para a diminuição da vida útil da bateria pela corrente exigida, sendo o capacitor uma alternativa para seu prolongamento.
- Pré-aquecimento do catalisador - Os catalisadores dos carros precisam de um certo tempo para se aquecerem e entrar em ação, evitando a emissão de gases poluentes. Com o uso do ultracapacitor no pré-aquecimento esses dispositivos entram em ação no momento em que o carro é ligado.

- Sinalização em boias e faróis alimentados por baterias solares durante o dia.
- Aplicações onde carga rápida é importante, tais como brinquedos e veículos mecatrônicos, etc.
- *Backup* de memórias - mantendo alimentadas RAMs, por longos intervalos de tempo.
- Uso automotivo, principalmente em veículos elétricos.

Aplicações em Robótica e Mecatrônica

Existem casos onde se necessita de uma potência muito grande por apenas alguns segundos ou mesmo fração de segundo, o que não pode ser conseguido a partir de baterias ou motores comuns.

Com os ultracapacitores isso é possível. Podemos citar como exemplo o caso de um robô de combate que pode requerer uma alta potência apenas por um instante para acionar uma garra que esmague o adversário, ou ainda para desferir um potente golpe.

Um robô que seja usado em trabalhos de salvamento poderá precisar de uma potência instantânea elevada para arrombar uma porta ou vencer um obstáculo encontrado em seu caminho.

Na indústria, uma potência instantânea muito alta pode ser usada para que uma ferramenta possa furar, cortar ou dobrar uma peça, mesmo que a máquina não disponha de uma fonte de capacidade elevada. A energia ficará armazenada no ultracapacitor, sendo depois aplicada ao ponto em que é necessária apenas pela fração de segundo que resulte nos efeitos desejados. E

EMI - Uma arma?

EMI é a abreviação de **Electro-Magnetic Interference** ou Pulso Eletromagnético. Há alguns anos, os militares soviéticos desenvolveram uma arma bastante simples baseada na sensibilidade dos aparelhos eletrônicos à interferência eletromagnética: em um conflito com os americanos, eles simplesmente detonariam uma bomba atômica entre a ionosfera e a atmosfera da Terra. Como a ionosfera funciona como uma das armaduras de um gigantesco capacitor onde a outra armadura é a própria Terra, contendo uma carga elétrica descomunal, a detonação colocaria em curto esse capacitor descarregando-o com uma faísca de proporções enormes no local da explosão.

Pois bem, essa faísca geraria uma corrente de descarga com a produção de um pulso eletromagnético tão forte que queimaria todos os circuitos eletrônicos sensíveis em um raio de muitos quilômetros. Assim, todos os equipamentos eletrônicos do inimigo, tais como radares, sistemas de guia de mísseis, detectores e sistemas de comunicações ficariam instantaneamente inutilizados!

Mas, e os deles? Os russos desenvolveram na ocasião uma tecnologia eletrônica totalmente baseada em válvulas (o que no Ocidente era visto como atraso!), que não são sensíveis aos pulsos eletromagnéticos. Desse modo, os equipamentos deles não seriam afetados.

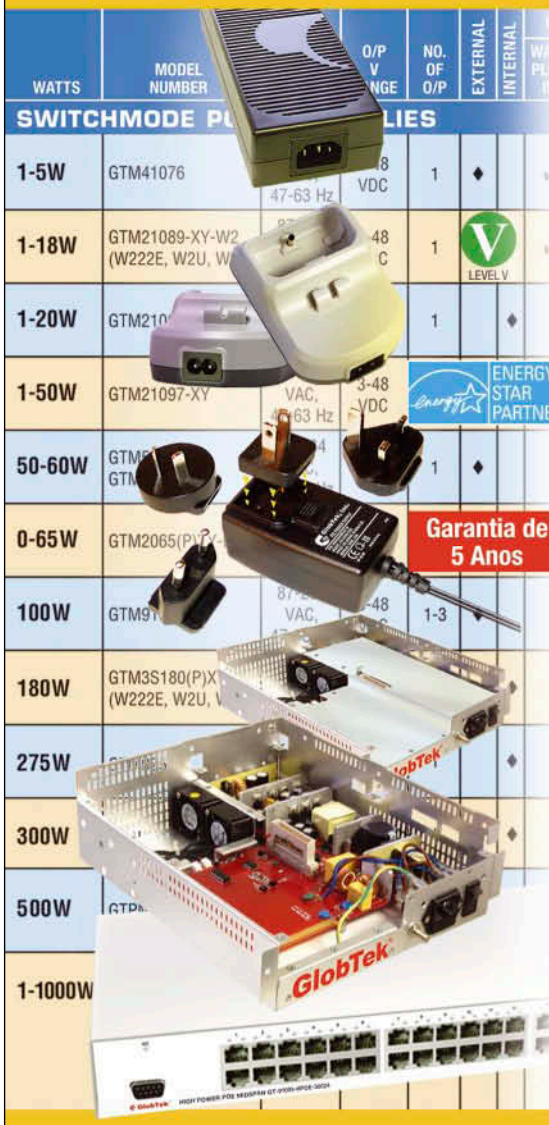
A "guerra fria" acabou e parece que o projeto não tem mais finalidade, mas existe ainda um perigo: os computadores e outros sistemas sensíveis de armazenamento de dados.

Um terrorista que esteja levando um ultracapacitor numa maleta (o qual é pequeno o suficiente para isso) e entre em um local com muitos computadores, poderá (com o simples apertar de um botão) comandar um circuito de descarga de baixa resistência formado por um pedaço de fio ou uma bobina.

O resultado será uma corrente instantânea de várias centenas ou milhares de ampères que gerará um potente pulso eletromagnético capaz de afetar computadores, discos rígidos, memórias, etc., apagando seu conteúdo ou paralisando seu funcionamento! Uma forma de terrorismo que deve ser prevista e analisada!

Soluções em Energia para a Indústria de Equipamentos Médicos e TI

Projetos Personalizados a partir de um modelo Padrão.
Projeto & Desenvolvimento Personalizados.
Time-to-Market reduzido.



GlobTek®

sales@globtek.com
+1.201.784.1000
+1.201.784.0111

RoHS

www.globtek.com

CERTIFICADO ISO 9001:2008

CB EISA CEC

DAC Conversor Digital-Analógico com microcontrolador Kinetis, da Freescale

Aprenda a desenvolver um Conversor Digital-Analógico com um microcontrolador de núcleo ARM

Edriano Carlos de Araújo

É notável o aumento de fabricantes produzindo microcontroladores equipados com núcleos ARM que, por sua vez, estão cada vez mais próximos de se tornarem padrão de mercado. Para os fabricantes, fornecer microcontroladores rápidos, eficientes e baratos se tornou uma questão de sobrevivência o que, naturalmente, faz com que invistam cada vez mais em pesquisa e desenvolvimento.

Os μCs Kinetis

Atualmente podemos encontrar microcontroladores repletos de periféricos, cada vez mais rápidos e baratos, o que não ocorria no passado.

Seguindo esta tendência de mercado, a Freescale Semicondutores lançou a sua mais recente família de microcontroladores Kinetis que, segundo a sua própria definição, é "*the gateway to the most scalable portfólio of ARM® MCUs in the industry*".

O Kinetis possui dois módulos DAC de 12 bits que podem ser alocados para um pino de saída, entrada do módulo comparador, conversor digital-analógico, amplificador operacional ou para outros periféricos.

Conforme descrito anteriormente, os periféricos presentes nesta família de microcontroladores não estão simplesmente isolados no sistema executando funções simples, mas sim fazendo parte de um sofisticado e complexo conjunto de *features* que devolve ao programador a capacidade de inovar e fazer a diferença, pois devido à grande quantidade de possibilidades oferecidas pelo microcontrolador, o sucesso e a qualidade do produto sendo desenvolvido estão diretamente ligados, à capacidade do profissional.

Partindo deste princípio, será proposta uma situação-problema simples que possa ser desenvolvida de maneira fácil e prática, mas que proporcione ao leitor ferramentas suficientes para o entendimento deste periférico. Não serão abordados neste artigo a transferência por DMA e a geração de interrupções, ficando estes para uma segunda parte.

Para o desenvolvimento destas tarefas utilizaremos o Kit de desenvolvimento KiwikStik (**figura 1**).

O módulo DAC

A **figura 2** mostra um diagrama de blocos do módulo DAC. Suas principais *features* são:

- Tensão de saída em passos de 1/4096 Vin;
- Vin "tensão de entrada" selecionável entre duas fontes distintas;
- Operação estática do módulo em modos STOP e WAIT;
- **Buffer** de 16 palavras com níveis máximo e mínimo selecionáveis, capazes de gerar interrupções;
- E suporte a DMA.

Situação-problema

Utilizando o módulo DAC do Kinetis, gerar uma forma de onda senoidal de 5 kHz, analisando o impacto no uso da CPU e dos recursos aplicados no projeto.

Proposta A

Nesta primeira solução, o módulo será configurado de maneira simples sem considerar o *buffer* de 16 palavras, ou qualquer outra característica avançada do DAC, utilizando o periférico como um módulo isolado muito semelhante à maioria dos DACs presentes em microcontroladores mais antigos.

É importante salientar que durante o desenvolvimento deste artigo, o *Reference Manual* do Kinetis K40 já estava em 1731 páginas e aumentando com o tempo, pois, como em toda família nova, ajustes são feitos de tempos em tempos. Portanto, mesmo lendo este artigo, é de suma importância se ter o *Reference Manual* próximo.

Também é importante entender que devido à grande quantidade de registradores contidos nesta família de microcontroladores apenas iremos inicializar os registradores que afetam diretamente o nosso desenvolvimento, deixando os demais em sua condição de RESET.

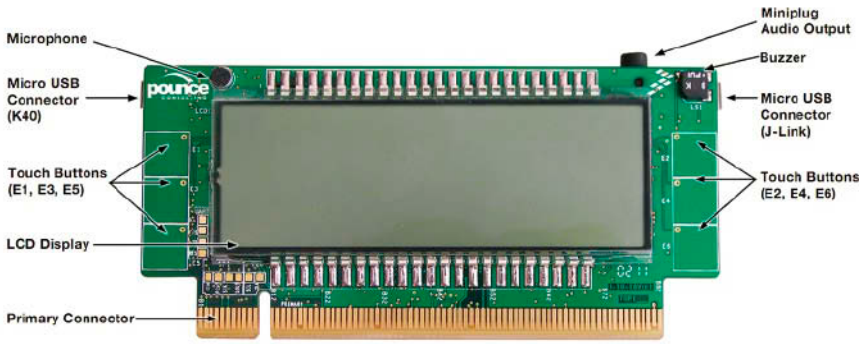
Primeiro passo

Configurar o módulo. Para um melhor entendimento será utilizado como base o exemplo contido no CD que acompanha o livro (**box 1**):

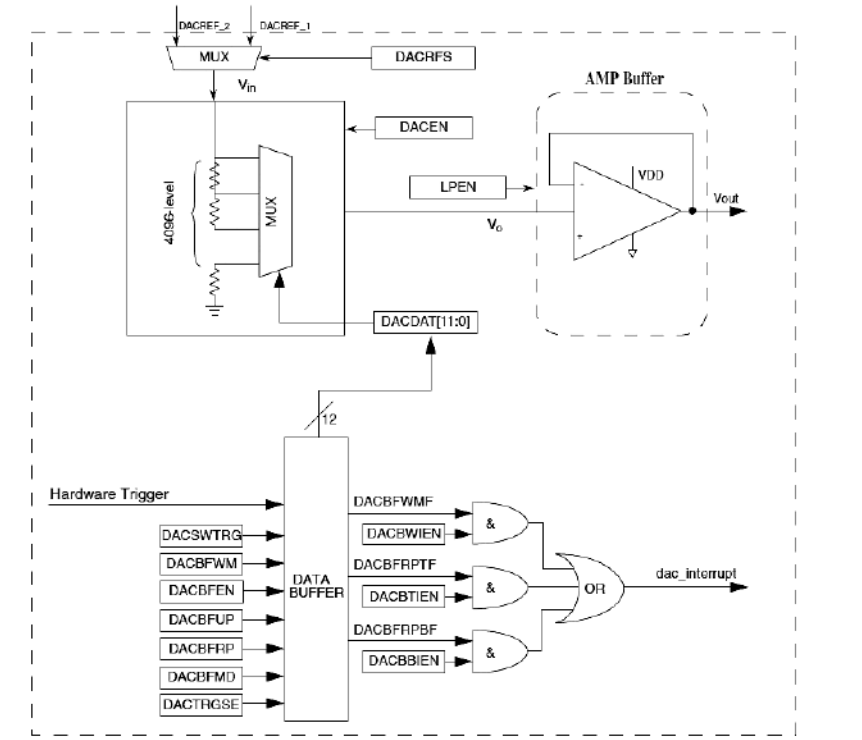
- Configurar o *clock* para o periférico. Neste caso, o *clock* é habilitado *setando* o BIT 12 do registrador SIM_SCGC2;
- Se, por algum motivo, o *clock* do módulo não seja habilitado, uma interrupção por *Hard fault* será gerada, portanto é muito importante que o *clock* do módulo seja habilitado (**figura 3a**);
- Zerar o conteúdo dos registradores de controle. Este passo não é obrigatório, pois, em sua grande maioria os registradores "nascem do Reset" em zero (**figura 3b, 3c e 3d**);
- Configurar os registradores de controle (**tabela 1**).

Segundo passo

Configurar uma interrupção de tempo responsável pela atualização do mesmo e



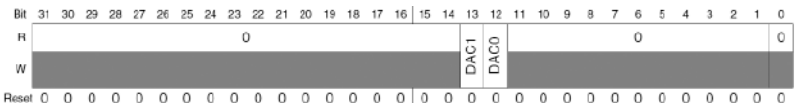
F1. Kit de desenvolvimento KiwikStik.



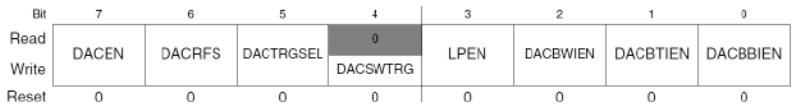
F2. Diagrama de Blocos do Conversor Digital-Analógico (DAC).

BOX1 : Configuração do módulo DAC

```
void inic_dac(void)
{
UInt8 offset;
   SIM_SCGC2 |= SIM_SCGC2_DAC1_MASK; // Habilita o clock do DAC1
   DAC1_C0 = 0; // Limpa registradores de controle do DAC
   DAC1_C1 = 0;
   DAC1_C2 = 0;
   // Habilita DAC1 trigger por software e referencia2
   // OBS: Referencia dois -> VDDAD
   DAC1_C0 |= DAC_C0_DACEN_MASK | DAC_C0_DACTRGSEL_MASK | DAC_C0_DACRFS_MASK;
   // Desabilita Buffer neste caso apenas o primeiro registrador é usado
   DAC1_C1 = 0 << DAC_C1_DACBFEN_SHIFT;
   //Posiciona ponteiro do buffer na posição zero e o limite máximo para 15
   DAC1_C2 |= DAC_C2_DACBFUP(0x0F) | DAC_C2_DACBFRP(0x00);
}
```



a) SIM_SCGC2 |= SIM_SCGC2_DAC1_MASK; // Habilita o clock do DAC1



b) DAC1_C0 = 0; // Limpa registradores de controle do DAC



c) DAC1_C1 = 0;



d) DAC1_C2 = 0;

F3. Habilitação do clock do módulo DAC.

BOX2 : Rotina de atualização da saída do DAC

```
void grava_dac(UInt16 dado)
{
   DAC1_DAT0L = (UInt8)(dado);                    // carrega o byte menos significativo
   DAC1_DAT0H = (UInt8)((dado >> 8) & (0x0F)); // carrega o byte menos significativo
}
```

BOX3 : Rotina de interrupção gerada a cada 6,25 µs

```
unsigned const int Dac_Tab1[]={4095,4050,3918,3704,3418,3071,2680,2262,1833,1415,1024,6
77,391,177,45,0,   45,177,391,677,1024,1415,1833,2262,2680,3071,3418,3705,3918,4050};
void Timer_Isr(void)
{
   PIT_TFLG0 = 0X01;                    // limpa o flag da interrupção de tempo
    ++ Dac_Timer;                       // incrementa o ponteiro utilizado para carregar a tabela
   if (Dac_Timer > 31) Dac_Timer = 0;   // zera o ponteiro caso fim da tabela
   grava_dac(Dac_Tab[Dac_Timer]);       // efetivamente atualiza o valor do dac
   (void) PIT_TCTRL0;                   // prepara para que outra interrupção seja servida
}
```

DAC1_C0 |= DAC_C0_DACEN_MASK | DAC_C0_DACTRGSEL_MASK | DAC_C0_DACRFS_MASK;

DACEN	Nível lógico um habilita todo o sistema.	1
DACRFS	Este módulo pode ser conectado a duas fontes de referencia VDDAD alimentação externa ou VREFs gerada internamente pelo Kinetis. É muito importante observar que, se VREFs for utilizada simultaneamente para o DAC e para o ADC alguma degradação nas medidas feitas através do ADC pode ser observada. Nível lógico 0 VREFs selecionado. Nível lógico 1 VDDAD selecionado.	1
DACTRGSEL	Seleciona a maneira como um novo dado é fornecido na saída do DAC. Nível lógico zero, o trigger é feito por hardware. Nível lógico um, o trigger é feito por software.	1

DAC1_C1 = 0 << DAC_C1_DACBFEN_SHIFT;

DACBFEN	Nível lógico zero desabilita o buffer. Nível lógico um Habilita o Buffer.	0

DAC1_C2 |= DAC_C2_DACBFUP(0x0F) | DAC_C2_DACBFRP(0x00);

DACBFRP	Posição do ponteiro do buffer. Neste exemplo será usado somente o primeiro registro.	0
DACBFUP	Limite máximo do ponteiro do buffer. Neste caso esta configuração não é relevante, pois apenas o primeiro registrador será utilizado.	15

T1. Contribuição dos registradores de controle.

criar uma sub-rotina para atualizar o valor de saída do DAC. Lembrando que, neste caso, apenas o primeiro registrador do buffer será utilizado:

- Rotina responsável por atualizar a saída do DAC. Ao atualizar o registrador DAC1_DAT0H a saída é atualizada. O Kinetis possui um conjunto de 32 registradores intitulados DACx_DATxL E DACxDATxH, ambos de 8 bits, para cada um dos seus módulos DAC. A representação deles neste artigo ocuparia muito espaço, sendo melhor sua verificação no *reference manual* (**box 2**);
- Rotina de interrupção gerada a cada 6,25 µs. Esta rotina é baseada no timer PIT0, para maiores informações sobre a sua inicialização vide capítulo TIMER (**box 3**). Nesta rotina foi implementada uma tabela onde cada valor diz respeito a uma posição da senoide (**figura 4**);
- Rotina Principal inicializa a PLL, configura o TIMER e o DAC. A partir deste ponto uma interrupção é gerada a cada 6,25 µs atualizando o DAC e, consequentemente gerando uma senoide. O código-fonte para a PLL e o Timer está presente no CD que acompanha o livro.

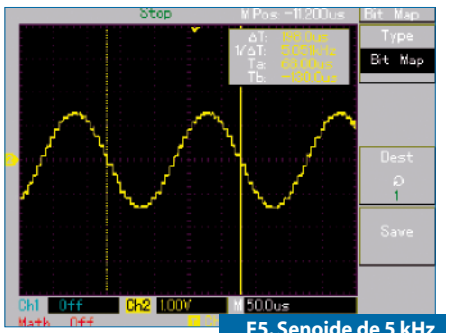
Neste momento o programa será gravado e executado e o resultado comentado a seguir:

```
int main(void)
{
  inicializa_pll2();
  Configura_Timer();
 inic_dac();  // Inicializa DAC com buffers
desabilitados
  for(;;)
  {
}
```

Conforme mostra a **figura 5** é possível observar que o objetivo foi alcançado e uma senoide de 5 kHz foi gerada na saída do DAC. É importante salientar que para que o resultado fosse alcançado uma interrupção foi gerada a cada 6,25 µs e uma posição de memória carregada, utilizando assim muito tempo de processamento da CPU para uma tarefa relativamente simples para o Kinetis.



F4. Uma posição de senoide.



F5. Senoide de 5 kHz na Saída do DAC.

Proposta B

Neste segundo exemplo será utilizado um pouco mais dos features avançados do Kinetis, o objetivo será o mesmo: gerar uma forma de onda senoidal de 5 kHz na saída do DAC, porém diminuindo um pouco mais o consumo em processamento exigido da CPU.

Para que este objetivo seja alcançado o buffer de 16 words do Kinetis será habilitado e o *trigger* por software realmente aproveitado:

- Assim como na primeira proposta, o primeiro passo será configurar o módulo. Algumas mudanças básicas podem ser observadas nesta nova configuração como, por exemplo o buffer foi habilitado assim como o *modo swing*;
- Habilitando-se o modo *swing* (**box 4**) temos a nossa disposição 16 palavras do buffer e não 32 como utilizamos anteriormente, portanto a tabela será carregada de tal forma que a cada 16 interrupções metade da senoide seja gerada, conforme ilustra a **figura 6.**
- Configurar os registradores de controle. A partir deste ponto apenas mudanças ou adaptações no código serão explicadas (**tabela 2**);
- Carregar a tabela no buffer do DAC. Ao contrário da primeira proposta, a tabela será carregada uma única vez no buffer, excluindo a necessidade de acesso a memória pela CPU todas as vezes que um novo dado deva ser carregado (**box 5**);
- Reescrever a rotina de interrupção. Neste caso uma interrupção a cada 6,25 µs ainda se faz necessário, porém o único processamento por parte da CPU será o de acionar o trigger por software e o periférico se encarregará de atualizar a saída do DAC (**tabela 3**);
- Não há mudanças significativas na rotina principal:

BOX4 : Habilitação do modo swing

```
void inic_dac_sw_trg(void)
{
UInt8 offset;
        SIM_SCGC2 |= SIM_SCGC2_DAC1_MASK; // Habilita o clock do DAC1
        DAC1_C0 = 0; // Limpa registradores de controle do DAC
        DAC1_C1 = 0;
        DAC1_C2 = 0;
        // Habilita DAC1 trigger por software e referencia2
        // OBS: Referencia dois -> VDDAD
        DAC1_C0 |= DAC_C0_DACEN_MASK | DAC_C0_DACTRGSEL_MASK | DAC_C0_DACR-
FS_MASK;
        // Habilita Buffer neste caso todos os 16 registradores da fila serão usados no modo
SWING
        DAC1_C1 = DAC_C1_DACBFEN_MASK | DAC_C1_DACBFMD(1);
        //Posiciona ponteiro do buffer na posição zero e o limite máximo para 15
        DAC1_C2 |= DAC_C2_DACBFUP(0x0F) | DAC_C2_DACBFRP(0x00);
        // Carrega a tabela nos 16 registradores do DAC
        for (offset=0; offset != 16; ++ offset)
        {
        DAC_DATL_REG(DAC1_BASE_PTR,offset) = (UInt8)(Dac_Tab[offset]);
        DAC_DATH_REG(DAC1_BASE_PTR,offset) = (UInt8)((Dac_Tab[offset] >> 8) & (0x0F));
        }
}
```

DAC1_C1 = DAC_C1_DACBFEN_MASK | DAC_C1_DACBFMD(1);

DACBFEN	Nível lógico zero desabilita o buffer. Nível lógico um Habilita o Buffer.	1
DACBFMD	Seleciona o modo de trabalho do buffer os quais podem ser definido como. Normal mode: Neste modo o buffer se comporta de forma circular o ponteiro vai de zero a quinze e retornando ao zero reiniciando o ciclo. Modo ativado em 00. Swing mode: neste modo o ponteiro vai de zero a quinze não retornando ao zero e sim decrescendo até zero. Modo ativado em 01. One-time scan: Neste modo o ponteiro incrementa a cada trigger e para na posição quinze. Modo ativado em 10.	1

T2. Configurações dos registradores de controle.

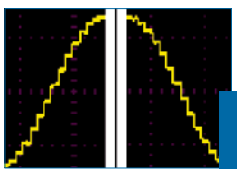
BOX5 : Carregamento único da tabela 2 no buffer do DAC

```
for (offset=0; offset != 16; ++ offset)
{
DAC_DATL_REG(DAC1_BASE_PTR,offset) = (UInt8)(Dac_Tab[offset]);
DAC_DATH_REG(DAC1_BASE_PTR,offset) = (UInt8)((Dac_Tab[offset] >> 8) & (0x0F));
}
```

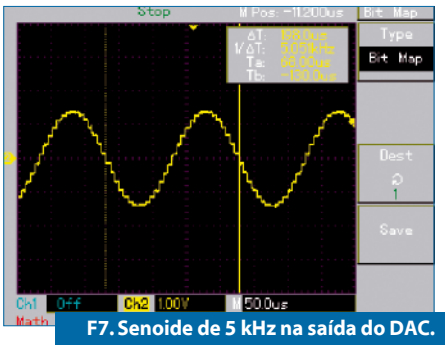
Antes	Depois

void Timer_Isr(void)
{
PIT_TFLG0 = 0X01;
++ Dac_Timer;
if (*Dac_Timer* > 31)
Dac_Timer = 0;
grav a_dac(*Dac_Tab*[*Dac_Timer*]);
(**void**) PIT_TCTRL0;
}
	void Timer_Isr(void)
	{
	PIT_TFLG0 = 0X01;
	DAC1_C0 |= 1 << DAC_C0_DACSWTRG_SHIFT;
	(**void**) PIT_TCTRL0;
	}

T3. Re-escrita da rotina de interrupção.



F6. Uma metade da senoide é gerada a cada 16 interrupções.



F7. Senoide de 5 kHz na saída do DAC.

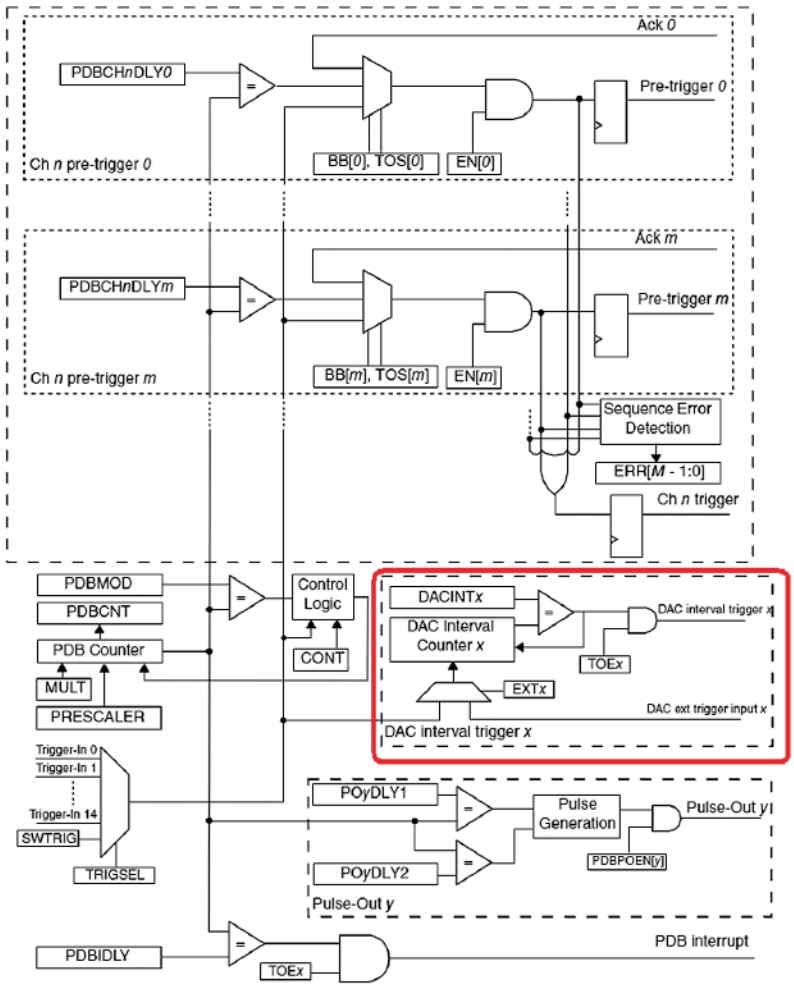
```
int main(void)
{
   inicializa_pll2();
   Configura_Timer();
  inic_dac_sw_trg();// Inicializa DAC com
buffers habilitados e trigger por software
   for(;;)
   {
}
```

- Como observado, a rotina grava_dac não se faz necessária assim como não será mais preciso carregar uma posição da tabela Dac_Tab a cada interrupção.
- Através da **figura 7** é possível notar que o resultado obtido foi o mesmo da proposta anterior, utilizando muito menos processamento da CPU.

Proposta C

Neste terceiro exemplo o objetivo será o mesmo: gerar uma forma de onda senoidal de 5 kHz. O principal diferencial desta proposta será o de que não haverá consumo da CPU para geração da senoide, sendo esta tarefa executada única e exclusivamente pelos periféricos DAC e PDB.

- O módulo PDB provê tempos controlados em função de triggers externos ou internos, ou de pulsos programados para os periféricos ADC e DAC; utilizando estes "TICs" programados, podemos automatizar a geração da senoide, onde cada atualização do DAC será disparada através de um TIC do PDB. Este módulo não será alvo de estudos neste momento, mas é possível verificar um pouco de sua estrutura através do Diagrama de Blocos da **figura 8;**
- Como de costume, o primeiro passo será o de configurar os registradores de controle, e para tal utilizaremos



F8. Diagrama de blocos do módulo PDB.

BOX6 : Configuração dos registradores de controle

```
void inic_dac_sw_trg(void)
{
UInt8 offset;
   SIM_SCGC2 |= SIM_SCGC2_DAC1_MASK; // Habilita o clock do DAC1
   DAC1_C0 = 0; // Limpa registradores de controle do DAC
   DAC1_C1 = 0;
   DAC1_C2 = 0;
   // Habilita DAC1 trigger por hardware e referencia2
   // OBS: Referencia dois -> VDDAD
   DAC1_C0 |= DAC_C0_DACEN_MASK | DACDAC_C0_DACRFS_MASK;
   // Habilita Buffer neste caso todos os 16 registradores da fila serão usados no modo SWING
   DAC1_C1 = DAC_C1_DACBFEN_MASK | DAC_C1_DACBFMD(1);
   //Posiciona ponteiro do buffer na posição zero e o limite máximo para 15
   DAC1_C2 |= DAC_C2_DACBFUP(0x0F) | DAC_C2_DACBFRP(0x00);
   // Carrega a tabela nos 16 registradores do DAC
   for (offset=0; offset != 16; ++ offset)
   {
      DAC_DATL_REG(DAC1_BASE_PTR,offset) = (UInt8)(Dac_Tab[offset]);
      DAC_DATH_REG(DAC1_BASE_PTR,offset) = (UInt8)((Dac_Tab[offset] >> 8) & (0x0F));
      }
}
DAC1_C0 |= DAC_C0_DACEN_MASK | DACDAC_C0_DACRFS_MASK;
```

DACTRGSEL	Seleciona a maneira como um novo dado é fornecido na saída do DAC. Nível lógico zero, o trigger é feito por hardware. Nível lógico um, o trigger é feito por software.	0

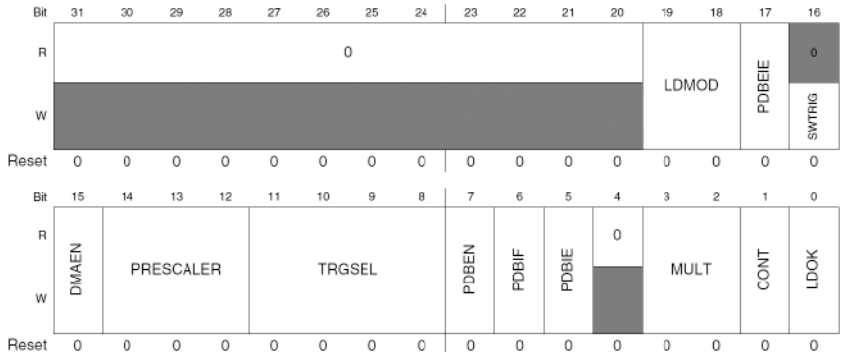
a mesma função da proposta anterior, fazendo apenas uma pequena alteração comentada (**box 6**);

- **Observe** que a máscara DAC_C0_DACTRGSEL_MASK, foi retirada do código, fazendo com que o bit DACTRGSEL do registrador C0 fique em zero. Habilitando assim o *trigger* por hardware;
- Configurar o módulo PDB (*Programable Delay Block*). O PDB é um periférico do Kinetis que torna possível gerar *triggers* periódicos tanto para os módulos ADC quanto DAC. Através deste periférico também é possível gerar *delays* entre conversões dos ADCs e conversões do DAC. Devido a complexidade e versatilidade destes módulos, aqui serão apenas configurados os registradores pertinentes ao DAC, conforme **figura 9** (para maiores esclarecimentos consultar o capitulo PDB no *Reference Manual*).
- Como regra uma das dicas mais importantes é: nunca se esqueça de habilitar o *clock* do módulo, caso contrário uma interrupção por *hard falt* será gerada (**box 7**).
- Veja um exemplo do contador em modo contínuo na **figura 10**;
- **Rotina de interrupção.** Não há mais necessidade de uma rotina de interrupção, o PDB se encarregara de gerar uma atualização do DAC a cada 6,25 µs;
- **Rotina Principal.** Foi retirada a rotina configura timer, a qual não mais se faz necessária e acrescentada a rotina configura PDB que, a partir deste momento, será responsável pela atualização do DAC (**box 8**);
- Conforme a **figura 11,** é possível notar que o resultado obtido foi o mesmo alcançado pelas propostas anteriores, com a vantagem de que a partir do momento em que o PDB foi inicializado Não há consumo em processamento da CPU, ficando esta livre para executar outras tarefas.

BOX7 : Habilitação do clock do módulo PDB

```
void Configura_Pdb(void)
{
        // HABILITA CLOCK DO MÓDULO PDB
         SIM_SCGC6 |= SIM_SCGC6_PDB_MASK;
        // HABILITA O MODO DE CONTAGEM CONTINUA E TRIGGER POR SOFTWARE
         PDB0_SC = PDB_SC_CONT_MASK | PDB_SC_TRGSEL(15);
        // CARREGA O INTERVALO ENTRE TRIGGERS NO COMPARADOR DO PDB
         PDB0_DACINT1 = (uint32_t)0x013DUL;
        // HABILITA HABILITA GERAÇAO DE TRIGGERS PARA O DAC
         PDB0_DACINTC1 |= PDB_INTC_TOE_MASK;
        // HABILITA O MÓDULO PDB
         PDB0_SC |= (uint32_t)0x81UL;
        // INICIA A GERAÇÃO DE TRIGGERS ATRAVÉS DE UM SOFTWARE TRIGGER.
         PDB0_SC |= PDB_SC_SWTRIG_MASK;
}
```



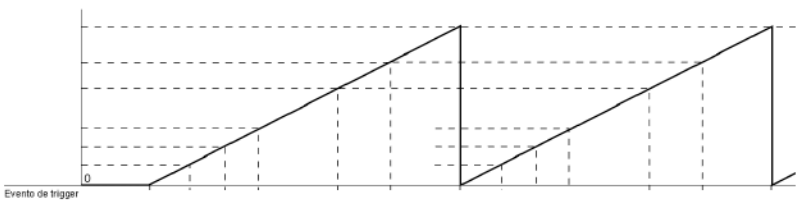

```
PDB0_SC = PDB_SC_CONT_MASK | PDB_SC_TRGSEL(15)
```

CONT	Este Bit controla a maneira como o contador do PDB funciona. Nível lógico zero, o contador do PDB será inicializado no modo one-shot, onde o contador faz apenas uma varredura e aguarda o próximo trigger. Nível lógico um, O contador será inicializado no modo continuo.	1

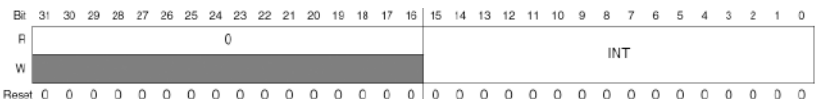
F9. Configuração do módulo PDB.

BOX8 : Configuração da rotina principal

```
int main(void)
{
inicializa_pll2();// Inicializa Inicializa PLL
        inic_dac_sw_trg();// Inicializa DAC com buffers habilitados e trigger por hardware
        Configura_Pdb();// Inicializa PDB
        for(;;)
        {
}
```

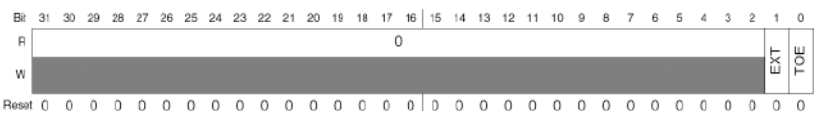


TRGSEL	O trigger do modulo pode ser selecionado entre várias fontes do Kinetis, tornando seu funcionamento muito flexível, neste caso específico o trigger utilizado será por software. Através deste BIT é possível selecionar a fonte do Trigger Quinze seleciona por software.	15




```
PDB0_DACINT1 = (uint32_t)0x013DUL;
```

INT	Específica o intervalo entre triggers aplicado ao DAC. Quando o valor especificado em INT for igual ao valor do contador um trigger é gerado.	




```
PDB0_DACINTC1 |= PDB_INTC_TOE_MASK;
```

TOE	Habilita o gerador de trigger para o DAC.	1
EXT	Nível lógico um, Habilita Trigger externo para o gerador de intervalos do DAC. Nível lógico zero, Ultiliza o mesmo trigger do PDB para o gerador de intervalos.	0

F10. Contador do PDB em modo da rotina principal.

Conclusão

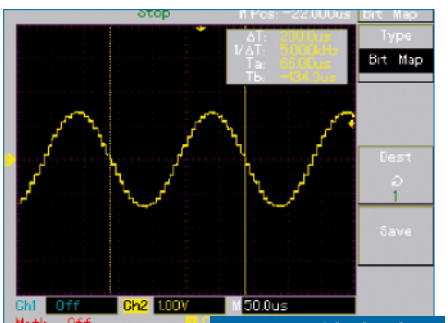
O Conversor digital-analógico do Kinetis pode ser configurado e utilizado de várias formas, desde as mais simples até as mais avançadas, cabendo ao programador identificar a melhor maneira de se implementar a sua solução.

Ainda é possível otimizar o funcionamento do DAC fazendo uso de suas interrupções, ou até mesmo do DMA, quando a quantidade de dados a se transferir se torna uma preocupação.

Em algum momento pode-se questionar a qualidade da senoide gerada, porém para o entendimento do projeto esta foi pensada desta maneira para proporcionar uma melhor compreensão do periférico, e algumas soluções podem ser utilizadas para melhorar a qualidade:

- Colocar um filtro passabaixas na saída do DAC, utilizando-se um resistor e um capacitor para obter uma senoide de 5 kHz perfeita;
- Aumentar a taxa de amostragem de saída, para isto deveríamos ativar as interrupções e alimentar o buffer do Kinetis a cada vez que o buffer esvaziasse. E assim teríamos consumo de processamento da CPU novamente;
- Ativar o DMA e programá-la para sempre que o buffer esvazie, a mesma transfira um novo bloco de dados, desta forma sem consumo de processamento da CPU;

Como, foi visto, cabe ao programador desenvolvedor achar a solução que melhor lhe atenda tanto em tempo de aprendizado quanto em custo. E



F11. Senoide de 5 kHz na saída do DAC.

Curso sobre Arduino

Parte 3

Dando continuiade ao curso apresentamos a descrição e sintaxe dos principais tipos de variações e funções aplicáveis no Arduino e também duas aplicações práticas: o controle de um servomotor de posição e a conversão analógico-digital para aquisição de sinais

Filipe Pereira

Variável do tipo Boolean

Uma variável deste tipo apenas pode tomar dois valores distintos - "true" ou "false". Esta variável reserva 1 Byte de memória para a sua utilização.

Sintaxe:

- boolean variável = "valor";
- Quando é referido acima a "valor", estamos nos referindo a "true" ou "false".

Exemplo:

- (1) int entrada_analogica=3,val,i=0,f[10]; // Declaração de uma variável do tipo integer, com o nome "entrada_analogica" inicializada com o valor "3", de uma variável do mesmo tipo com o nome "val" não inicializada, "i" inicializada com o valor "0" e de um vetor "f[]" com 11 posições de memória do mesmo tipo (integer).
- (2) boolean teste = false; // Definição de uma variável do tipo boolean inicializada com "false"
- (3) void setup() {
- (4) Serial.begin(9600); // Permite a inicialização da comunicação Série
- (5) pinMode(entrada_analogica,INPUT); // Permite definir o pino 3 como "INPUT"
- (6) }
- (7) void loop(){
- (8) do{ // Permite definir o início do ciclo
- (9) val=analogRead(entrada_analogica); // Permite a leitura analógica do valor do pino 3, atribuindo o seu valor à variável "val"
- (10) f[i]=val: //Atribui o valor da variável "val" ao vetor "f[i]", ou seja, permite preencher o mesmo com valores provenientes do pino analógico 3
- (11) i++; // Incremento do valor de "i" inicialmente a zero, instrução semelhante a "i=i+1"
- (12) teste = !teste; //modifica o valor atribuído à variável booleana "teste" ou seu oposto (se "true" fica "false", e vice-versa)
- (13)}
- (14) while(i<=10); // Se a condição se verificar, repete o ciclo; caso não se verifique, o ciclo termina
- (15) }

Variável do tipo float vs double

A variável do tipo float apresenta uma maior resolução, face às variáveis do tipo integer. São reservados em memória 4 Bytes (32 bits), para armazenar o seu conteúdo. Este tipo de variável pode conter valores no intervalo de −3.4028235◎10^38 e 3.4028235◎10^38. Paralelamente, o conceito de double leva-nos a considerar uma variável que possua o dobro da precisão de uma variável do tipo float, apesar de no contexto de estudo a declaração de uma variável do tipo float ou do tipo double serem iguais. Ou seja, não existe um incremento de precisão mantendo o mesmo espaço reservado de memória para ambos os tipos utilizando o Arduino.

Sintaxe:

- float variável = valor;
- double variável = valor;

Exemplos:

float exemplo = 1.589; // Declara um variável do tipo "float" e atribui-lhe o valor de "1.589"

double exemplo_1 = 1.589; // Declara um variável do tipo "double" e atribui-lhe o valor de "1.589"

Variável do tipo array e a Noção de string

É considerado um *array*, um vetor de variáveis do mesmo tipo ao qual se pode aceder através do seu respectivo índice. O conceito de *string* é comumente utilizado para designar um vetor de variáveis do tipo char.

Sintaxe:

tipo_variável nome_variável[índice] = valor;

Ao declarar um índice de valor 10, estamos na verdade reservando 11 espaços na memória para a variável do tipo declarado. Pois, também, temos de contar com o índice zero, este fator torna-se muitas vezes objeto de erro.

Exemplos:

- float exemplo[10]; // Declaração de um vetor com 11 "espaços", do tipo "float"
- float exemplo_2[]={1,2,3,4,5,6,7,8,9,10}; // Declaração de um vetor do tipo" float", que vai ter um índice compreendido entre 0 e 9
- char exemplo_3[11]="Hello World"; // Declaração de uma string
- char exemplo_4[11]={'H','E','L','L','O',' ','W','O','R','L','D'}; // Declaração de uma string com o conteúdo semelhante ao exemplo anterior, mas com outra forma de colocar o mesmo conteúdo

pinMode()

Ao recorrer a esta instrução, é possível configurar o modo de comportamento

de um determinado pino, possibilitando assim defini-lo como input ou output, esta definição normalmente é efetuada recorrendo à função "void setup()".

Sintaxe:

- pinMode(Número do pino, Modo);
- O "Modo" acima descrito pode ser definido da seguinte forma:
  - "INPUT"
  - "OUTPUT"

Exemplo:

- (1) int botao=3,val; // Declaração de uma variável do tipo integer com o nome "botao", inicializada com o valor "3" e de uma variável do mesmo tipo com o nome "val" não inicializada com nenhum valor
- (2) void setup() {
- (3) Serial.begin(9600); // Permite a inicialização da comunicação Série
- (4) pinMode(botao,INPUT); // Permite definir o pino 3 como "INPUT"
- (5) pinMode(13,OUTPUT); // Permite definir o pino 13 como "OUTPUT"
- (6) }
- (7) void loop(){
- (8) val=analogRead(botao); // Permite a leitura analógica do valor do pino 3, atribuindo o seu valor à variável "val"
- (9) if(val>=500) // Ciclo if que testa a condição – "val>=500" (maior ou igual a 500)
- (10) digitalWrite(13,HIGH); // Se a condição for verificada, é atribuído ao pino digital 13 a condição "HIGH" (5 V)
- (11) }
- (12) }

digitalWrite()

Possibilita estabelecer nos pinos configurados como output, através da instrução "pinMode", a saída dos respectivos pinos com o valor lógico 1 (HIGH – 5 V) ou com o valor lógico 0 (LOW – 0V)

Sintaxe:

- digitalWrite(Número do pino, Modo);
- O "Modo" acima descrito, pode ser definido como:
  - "HIGH"
  - "LOW"

Exemplo:

- (1) int botao=3,val; // Declaração de uma variável do tipo integer com o nome "botao", inicializada com o valor "3" e de uma variável do mesmo tipo com o nome "val" não inicializada com nenhum valor
- (2) void setup() {
- (3) Serial.begin(9600); // Permite a inicialização da comunicação Série
- (4) pinMode(botao,INPUT); // Permite definir o pino 3 como "INPUT"
- (5) pinMode(13,OUTPUT); // Permite definir o pino 13 como "OUTPUT"
- (6) }
- (7) void loop(){
- (8) val=analogRead(botao); // Permite a leitura analógica do valor do pino 3, atribuindo o seu valor à variável "val"
- (9) if(val>=500) // Ciclo if que testa a condição – "val>=500" (maior ou igual a 500)
- (10) digitalWrite(13,HIGH); // Se a condição for verificada, é atribuído ao pino digital 13 a condição "HIGH" (5 V)
- (11) }
- (12) }

digitalRead()

Possibilita a leitura de uma entrada digital específica, retornando um valor no formato integer (int). Se obtivermos um valor de retorno de "1", estaremos perante uma leitura do tipo "HIGH" (valor lógico 1). Se tal não se verificar, e tivermos um valor de retorno igual a "0", estaremos perante uma leitura do tipo "LOW" (valor lógico 0).

Sintaxe:

- Variável do tipo integer = digitalRead (Número do pino);

Exemplo:

- (1) int botao=3,val; // Declaração de uma variável do tipo integer com o nome "botao", inicializada com o valor "3" e de uma variável do mesmo tipo com o nome "val" não inicializada com nenhum valor
- (2) void setup() {
- (3) Serial.begin(9600); // Permite a inicialização da comunicação Série
- (4) pinMode(botao,INPUT); // Permite definir o pino 3 como "INPUT"
- (5) pinMode(13,OUTPUT); // Permite definir o pino 13 como "OUTPUT"
- (6) }
- (7) void loop(){
- (8) val=digitalRead(botao); // Permite a leitura do valor do pino 3 digital, atribuindo o seu valor à variável "val"
- (9) if(val==1) // Ciclo if que testa a condição – "val>=500" (maior ou igual a 500)
- (10) digitalWrite(13,HIGH); // Se a condição for verificada, é atribuído ao pino digital 13 a condição "HIGH" (5 V)
- (11) }
- (12) else{ // Caso a condição do ciclo if não se verifique
- (13) digitalWrite(13,LOW); // Se a condição não for verificada, é atribuído ao pino digital 13 a condição "LOW" (0 V)
- (14) }
- (15) }

Input/Output analógico – analogRead()

Possibilita a leitura do valor analógico do pino especificado com um conversor A/D com uma resolução de 10 bits, o que leva a que um valor compreendido entre 0 e 5 V esteja compreendido entre os valores inteiros (int) 0 e 1023.

Sintaxe: Variável do tipo inteiro = analogRead(Número do pino);

Exemplo:

- (1) int botao=3,val; // Declaração de uma variável do tipo integer com o nome "botao", inicializada com o valor "3" e de uma variável do mesmo tipo com o nome "val" não inicializada com nenhum valor
- (2) void setup() {

- (3) Serial.begin(9600); // Permite a inicialização da comunicação Série
- (4) pinMode(botao,INPUT); // Permite definir o pino 3 como "INPUT"
- (5) pinMode(13,OUTPUT); // Permite definir o pino 13 como "OUTPUT"
- (6) }
- (7) void loop(){
- (8) val=analogRead(botao); // Permite a leitura analógica do valor do pino 3, atribuindo o seu valor à variável "val"
- (9) if(val>=500) // Ciclo if que testa a condição – "val>=500" (maior ou igual a 500)
- (10) digitalWrite(13,HIGH); // Se a condição for verificada, é atribuído ao pino digital 13 a condição "HIGH" (5 V)
- (11) }
- (12) }

analogWrite()

Possibilita a utilização dos pinos PWM (*Pulse Width Modulation*) da placa Arduino. O sinal PWM mantém-se até ser modificado através de uma outra instrução que afete esse pino, a frequência do sinal PWM criado é de ≅ 490 Hz.

Sintaxe:

- analogWrite(Número do pino, valor);
- O "valor" referido anteriormente varia entre 0 (sempre desligado), até ao valor 255 (que representa um sinal de 5 V constante).

Exemplo:

- (1) int botao=3,val; // Declaração de uma variável do tipo integer com o nome "botao", inicializada com o valor "3" e de uma variável do mesmo tipo com o nome "val" não inicializada com nenhum valor
- (2) void setup() {
- (3) Serial.begin(9600); // Permite a inicialização da comunicação Série
- (4) pinMode(botao,INPUT); // Permite definir o pino 3 como "INPUT"
- (5) pinMode(13,OUTPUT); // Permite definir o pino 13 como "OUTPUT"
- (6) }
- (7) void loop(){
- (8) val=analogRead(botao); // Permite a leitura analógica do valor do pino 3, atribuindo o seu valor à variável "val"
- (9) if(val>=500) // Ciclo if que testa a condição – "val>=500" (maior ou igual a 500)
- (10) analogWrite(9,255); // Instrução com a mesma função que "digitalWrite(9,HIGH)"
- (11) }
- (12) else{
- (13) analogWrite(9,0); // Instrução com a mesma função que "digitalWrite(9,LOW)"
- (14) }
- (15) }

Tempo – millis()

Possibilita o retorno da quantidade de tempo, em milissegundos, na forma de uma variável do tipo "unsigned long". O valor retornado representa o tempo que passou desde que o programa atual começou a ser executado. O *overflow* (voltar ao início, valor zero) do contador ocorre passado um tempo de ≅ 50 dias.

Sintaxe:

- unsigned long tempo = millis();

Exemplo:

- (1) unsigned long tempo; // Declarada uma variável do tipo unsigned long com o nome "tempo"
- (2) void setup() {
- (3) (…..)
- (4) }
- (5) void loop(){
- (6) tempo = millis(); // Atribui à variável "tempo" o valor, em milissegundos, desde que o *sketch* atual começou a ser executado
- (7) }

delay (milissegundos)

Possibilita efetuar uma pausa no programa em execução, por uma quantidade de milissegundos especificada. Útil para manter um estado durante uma certa quantidade de tempo.

Sintaxe:

- *delay* (tempo que deseja efetuar a pausa – ms);

Exemplo:

- (1) int led_pin=13; // Declaração de uma variável do tipo integer com o nome "led_pin", sendo-lhe atribuído o valor "13"
- (2) void setup() {
- (4) pinMode(led_pin,OUTPUT); // Permite definir o pino 13 como "OUTPUT"
- (5) Serial.begin(9600); // Permite a inicialização da comunicação Série
- (6) }
- (7) void loop(){
- (8) digitalWrite(led_pin,HIGH); // É atribuído ao pino digital 13 a condição "HIGH" (5 V)
- (9) delay(200); // É efetuado um delay de 200 ms, antes de efetuar a próxima instrução
- (10) digitalWrite(led_pin,LOW); // É atribuído ao pino digital 13 a condição "LOW" (0 V)
- (11) delay(600); // É efetuado um delay de 600 ms, antes de efetuar a próxima instrução, neste caso a função "void loop()" recomeça
- (12) }

Servomotor

Um servomotor é um pequeno dispositivo, cujo veio pode ser posicionado numa determinada posição angular de acordo com um sinal de entrada. Enquanto esse sinal se mantiver constante e for enviado para o servomotor, este irá manter a sua posição angular. Ao variar o sinal de entrada, possibilita uma variação da posição angular do veio. Os servomotores são muito usados no controle de aviões telecomandados, robôs, barcos telecomandados, helicópteros telecomandados, entre outras possíveis aplicações. Veja **figura 1**.

Um servomotor, como se pode ver na seguinte figura, possui três entradas. Uma entrada é a alimentação (normalmente 5 V para servomotores standard), outra será a massa e a última que falta referir será o sinal de entrada. Este sinal de entrada será um pulso PWM, onde, fazendo-se variar o seu "duty-cicle", podemos variar a posi-

ção angular do servomotor. Isto em servomotores de posição, pois também existem servomotores de rotação contínua. Nestes, a variação do "duty-cicle" fará variar não a posição angular, mas a velocidade e sentido de rotação do servomotor. Existe possibilidade de modificar, sem grande dificuldade, um servomotor de posição para rotação contínua. Esta modificação é permanente, não podendo ser reversível. No entanto, o objetivo deste subcapítulo não se centra nisso, mas sim em como fazer a sua interação com a plataforma de desenvolvimento Arduino. (**Figura 2**)

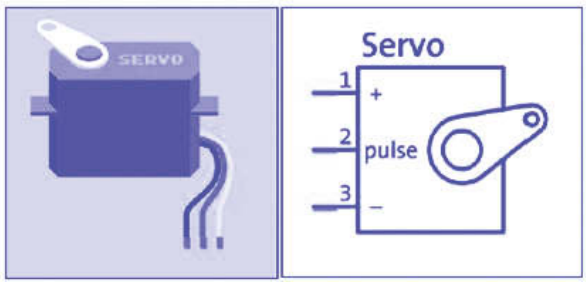
Pela análise da figura anterior, podemos constatar que os componentes mais importantes de um servomotor de posição são os seguintes:

- Circuito de controle (Control Circuit)
- Potenciômetro (Potentiometer)
- Motor DC (Motor)

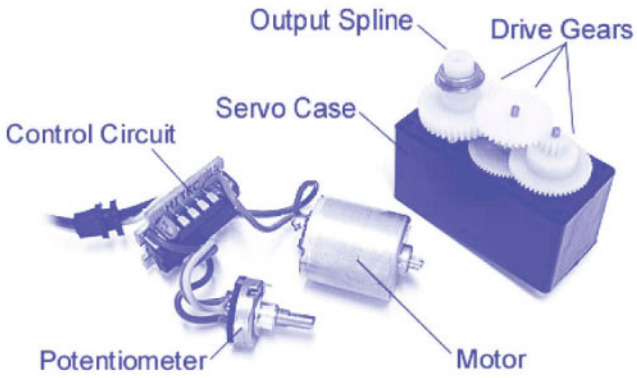
O circuito de controle, como o próprio nome indica, é responsável por controlar e mover o motor DC, com base nas informações obtidas pela leitura do potenciômetro e do sinal de entrada. O potenciômetro, neste contexto, é utilizado acoplado ao eixo de rotação para conseguir obter a posição angular do eixo.

A enorme aplicabilidade dos servomotores leva a que, tendo em conta o universo da eletrônica, seja uma temática obrigatória. Dessa necessidade surge a criação de uma biblioteca específica para facilitar a operação com servomotores, que até já se encontra disponível no Software de desenvolvimento Arduino.

A biblioteca "Servo.h" utiliza os pinos 9 e 10 para fazer o controle dos respectivos servomotores, ao utilizar esta biblioteca (ao associar um servomotor a uma das duas saídas) não poderá utilizar-se os pinos 9 e 10 para outras aplicações. A seguir, vai ser explorada a sintaxe possível para facilitar o uso desta biblioteca.



F1. Servomotor: aspecto e símbolo.



F2. Componentes de um servomotor de posição.

Sintaxe das Instruções da Biblioteca <Servo.h> - attach()

Permite atribuir a um pino (pinos disponíveis – 9 ou 10), uma variável do tipo "Servo". A atribuição de uma variável deste tipo vai ser necessária para controlar o respectivo servomotor.

Sintaxe:

- Servo.attach(Pino);
- Servo.attach(Pino, Valor do ângulo mínimo, Valor do ângulo máximo);
- "Servo" – Variável do tipo "Servo" pode ser declarada da seguinte forma:
- Servo nome_a_atribuir: Podendo atribuir qualquer nome à escolha, o exemplo abaixo é esclarecedor em relação a esta questão:
  - "Valor do ângulo mínimo" – Permite definir qual a largura mínima do pulso a utilizar, em microssegundos, que vai corresponder ao ângulo mínimo (0°).
  - "Valor do ângulo máximo" – Permite definir qual a largura máxima do pulso a utilizar, em microssegundos, que vai corresponder ao ângulo máximo (180°).

Exemplo:

- (1) #include<Servo.h> // Declaração da biblioteca <Servo.h>
- (2) Servo Exemplo; // Criação de uma variável do tipo Servo, com o nome de "Exemplo"
- (3) void setup(){
- (4) Exemplo.attach(10); //Atribuição da variável do tipo Servo "Exemplo", ao pino digital 10
- (5) (…..)
- (6) void loop(){
- (7) (…..)
- (8) }

detach()

Permite eliminar a ligação entre uma variável do "Servo" a um dos pinos possíveis de utilizar pela biblioteca "<Servo.h>" (Pinos 9 e 10). Se nenhum dos pinos possíveis estiver atribuído a uma variável do tipo "Servo", os pinos 9 e 10 podem ser utilizados normalmente.

Sintaxe:

- Servo.detach();
- Servo" – Variável do tipo Servo pode ser declarada da seguinte forma:
- Servo nome_a_atribuir;

Exemplo:

- (1) #include<Servo.h> // Declaração da biblioteca <Servo.h>
- (2) Servo Exemplo; // Criação de uma variável do tipo Servo, com o nome de "Exemplo"
- (3) void setup(){
- (4) Exemplo.attach(10); // Atribuição da variável do tipo Servo "Exemplo", ao pino digital 10
- (5) (…..)
- (6) void loop(){
- (7) Exemplo.detach(); // Elimina a ligação entre a variável do tipo Servo "Exemplo"e pino digital 10
- (8) analogWrite(10, 255); // Como nenhuma variável do tipo Servo se encontra atribuída, podem-se usar as funcionalidades PWM dos pinos 9 e 10
- (9) }

write()

Permite movimentar o eixo de um servomotor de posição para o ângulo pretendido. No caso de um servomotor de rotação contínua, o valor 0 corresponderá à velocidade de rotação máxima num sentido, o valor 180 a velocidade de rotação máxima noutro sentido e o valor 90 será o ponto em que o servomotor se encontrará parado.

Sintaxe:

- Servo.write(Ângulo);
- Servo" – Variável do tipo "Servo" pode ser declarada da seguinte forma:
- Servo nome_a_atribuir;
- "Ângulo" ◎ [0,180]

Exemplo:

- (1) #include<Servo.h> // Declaração da biblioteca <Servo.h>
- (2) Servo Exemplo; // Criação de uma variável do tipo Servo, com o nome de "Exemplo"
- (3) Servo Exemplo_2; // Criação de uma variável do tipo Servo, com o nome de "Exemplo_2"
- (4) void setup(){
- (5) Exemplo.attach(10); // Atribuição da variável do tipo Servo "Exemplo", ao pino digital 10
- (6) Exemplo_2.attach(9); // Atribuição da variável do tipo Servo "Exemplo_2", ao pino digital 9
- (7) (…..)
- (8) void loop(){
- (9) Exemplo.write(0); // Faz com que um servomotor de posição, por exemplo, se movimente para a posição correspondente ao ângulo "0"
- (1) Exemplo_2.write(90); // Faz com que um servomotor de rotação contínua, por exemplo, pare a sua rotação
- (11) }

read()

Instrução que retorna o valor do último ângulo utilizado, recorrendo à instrução "Servo.write(Ângulo)".

Sintaxe:

Ângulo (int) = Servo.read();

Servo" – Variável do tipo Servo pode ser declarada da seguinte forma:

Servo nome_a_atribuir;

Exemplo:

- (1) #include<Servo.h> // Declaração da biblioteca <Servo.h>
- (2) int angulo=0; // Declaração de uma variável do tipo integer de nome "ângulo", atribuindo-lhe o valor de "0"
- (3) Servo Exemplo; // Criação de uma variável do tipo Servo, com o nome de "Exemplo"
- (4) void setup(){
- (5) Exemplo.attach(10); // Atribuição da variável do tipo Servo "Exemplo", ao pino digital 10
- (6) (…..)
- (7) void loop(){
- (8) Exemplo.write(angulo); // Faz com que um servomotor de posição se movimente para a posição correspondente ao valor da variável "ângulo"
- (9) angulo = (Exemplo.read() + 1); // Vai incrementado uma unidade ao último valor de ângulo utilizado, recorrendo à instrução "Exemplo. write(angulo)"
- (10) }

attached()

Instrução que permite verificar se uma variável do tipo "Servo" se encontra atribuída a um pino específico.

Sintaxe:

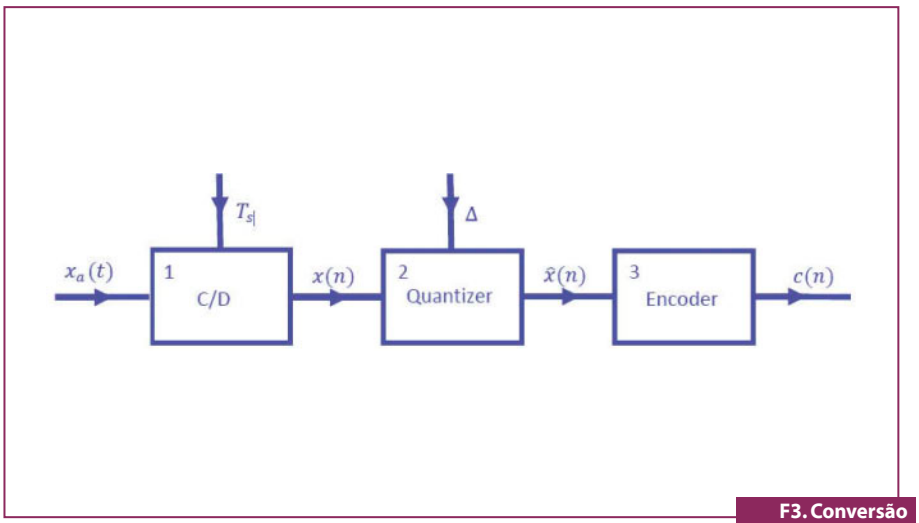
- Estado (int) = Servo.attached();
- Servo" – Variável do tipo "Servo" pode ser declarada da seguinte forma:
- Servo nome_a_atribuir;
- "Estado" 0,1 , ou seja, só pode tomar o valor "true" (valor "1") ou "false" (valor "0").

Exemplo:

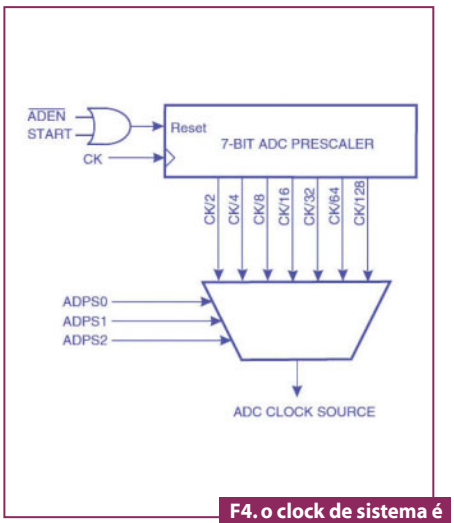
- (1) #include<Servo.h> // Declaração da biblioteca <Servo.h>
- (2) int atribuido=0; // Declaração de uma variável do tipo integer de nome "atribuído", atribuindo-lhe o valor de "0"
- (3) Servo Exemplo; // Criação de uma variável do tipo Servo, com o nome de "Exemplo"
- (4) void setup(){
- (5) Exemplo.attach(10); // Atribuição da variável do tipo Servo "Exemplo", ao pino digital 10
- (6) (…..)
- (7) void loop(){
- (8) atribuido = Exemplo.attached(); // Atribui o valor 1 à variável atribuído, caso a variável do tipo Servo "Exemplo" se encontre associada a algum pino; caso contrário retorna o valor 0
- (9) if (atribuido ==1){ //Se a variável do tipo Servo "Exemplo" estiver atribuída, a condição é verificada
- (10) Exemplo.write(180); // Faz com que um servomotor de posição se movimente para a posição correspondente ao valor "180"
- (11) }
- (12) (…..)
- (13) }

Resumo

O uso desta biblioteca simplifica a operação com servomotores, o que não implica que a movimentação dos servomotores não seja feita com recurso diretamente à instrução "analogWrite()", ou seja, gerar os sinais PWM diretamente. Tal é possível, obviamente, porém mais trabalhoso, existindo esta biblioteca para simplificar esse trabalho de implementação.



F3. Conversão analógico-digital.



F4. o clock de sistema é dividido pelo prescaler.

Aquisição de Sinal: Conversão A/D

Os microcontroladores utilizados até ao momento, nos modelos disponíveis da plataforma de desenvolvimento Arduino, possuem conversores A/D com 10 bits de resolução (informação retirada do *datasheet* dos respectivos microcontroladores), ou seja, uma entrada analógica cujo valor varia entre 0 V e 5 V, terá a sua correspondência em binário a valores entre 0 (0000000000) e 1023 (1111111111) respectivamente. O que nos leva a resoluções na ordem dos 5 mV. **Figura 3**.

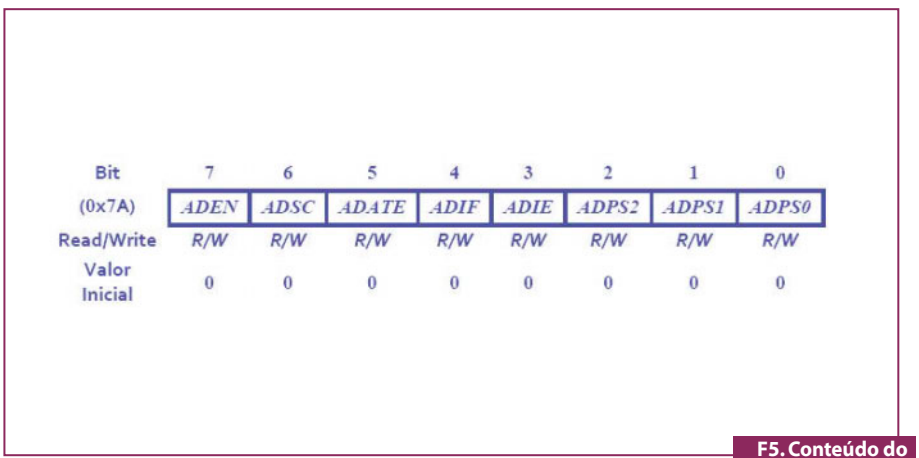
Segundo informação disponível no datasheet dos respectivos microcontroladores, a primeira conversão A/D demora 25 ciclos de Clock enquanto as restantes demoram 13. Esta diferença acontece, pois a primeira conversão A/D precisa de efetuar a inicialização do ADC. Analisando o referido anteriormente, e considerando um Clock de 1 MHz, podemos facilmente concluir o seguinte (ignorando a conversão inicial):

$$N°_{\text{amostras por segundo}} = \frac{106}{13} = 77000$$

Ou seja, estamos perante uma taxa de 77 kHz. Analisando a taxa de amostragem obtida, e tendo em conta o referido no subcapítulo anterior, a frequência máxima do sinal a amostrar deve ser menor que 38,5 kHz, de forma a respeitar o Teorema da Amostragem.

O Arduino possui um "Clock de sistema" de 16 MHz, no entanto, não é este o Clock de entrada que vamos obter no nosso conversor A/D. O Clock de sistema é dividido por um "fator de divisão" (*Prescaler*), sendo este sim o Clock que vamos obter no nosso ADC como podemos constatar pela **figura 4**.

Mas existe uma forma de configurar este "fator de divisão", fazendo variar o conteúdo do bit ADPS0, ADPS1 e ADPS2 (como se pode constatar pela análise da figura anterior). Estes bits são parte integrante do registro ADCSRA, que pode ser visto na **figura 5**.



Bit	7	6	5	4	3	2	1	0
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Valor Inicial	0	0	0	0	0	0	0	0

F5. Conteúdo do registro ADCSRA.

As combinações possíveis, para os bits referidos anteriormente, podem ser resumidas pela **tabela 1**.

Ou seja, o Clock de entrada do conversor A/D é obtido através da seguinte condição:

$$f_{\text{input clock}} = \frac{\text{Clock de Sistema (16 MHz)}}{\text{Fator de divisão (Prescaler)}}$$

ADPS2	ADPS1	ADPS0	Fator de divisão
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

T1. Combinações possíveis para os bits.

Da equação anterior podemos retirar facilmente, e apenas efetuando o cálculo acima referido para cada condição (**Tabela 2**).

Pela análise da tabela anterior podemos afirmar, então, que a maior frequência de amostragem que poderemos obter se situa nos 8 MHz, o que nos leva a concluir: para haver uma reconstrução sem perda de informação de um possível sinal amostrado, o mesmo deverá ter uma frequência máxima de 4 MHz.

Depois do abordado anteriormente, resta referir a forma de fazê-lo, indicando a sintaxe a utilizar para alterar os registros referidos antes. A sintaxe utilizada não serve especificamente para alterar o conteúdo do registro ADCSRA, podendo a mesma sintaxe ser utilizada para alterar o conteúdo de outros registros com as respectivas adaptações, é claro. Os registros disponíveis, e o seu conteúdo e função, encontram-se no datasheet do respectivo microcontrolador. Na sequên-cia serão descritas quais as instruções a utilizar, dando um exemplo de implementação.

Sintaxe utilizada:

- sbi(Nome do registro, Nome do bit); // Define o bit "Nome do bit" situado no registro "Nome do registro" a 1 (set bit)
- cbi(Nome do registro, Nome do bit); // Define o bit "Nome do bit" situado no registro "Nome do registro" a 0 (clear bit)

O código acima apresentado está elaborado de forma a medir, na realidade, qual o tempo gasto para efetuar uma conversão A/D. Os valores foram obtidos por medidas experimentais, fazendo a média de 1000 amostras de tempo gasto na conversão A/D (Valores obtidos por comunicação série efetuando a aquisição e cálculo com o Software MatLab). Os resultados obtidos são apresentados na **tabela 3**.

Fator de divisão	Clock de entrada no ADC
2	8
4	4
8	2
16	1
32	0,5
64	0,25
128	0,125

T2. Valores de Clock de entrada no ADC.

Fator de divisão	Taxa de amostragem (kHz) (Valores aproximados)
128	8
64	16
32	31
16	50
8	63
4	83
2	125

T3. Valores obtidos na conversão A/D.



Controle para Automação Industrial

PLL, P, D, PID – Uma abordagem prática sobre essas técnicas

O assunto "controle" para sistemas automáticos, principalmente com servomecanismos, é um tanto curioso nos meios acadêmico e industrial. Enquanto de um lado vê-se cálculo integral e transformadas de "Laplace", do outro utilizam-se softwares de controle de movimento e supervisórios. Mas, afinal, qual é o ponto comum entre esses dois "mundos"? Quais são as técnicas, a melhor tecnologia e os circuitos práticos que executam esses controles? É possível tratarmos o assunto em uma linha ideal entre a teoria e a prática?

A intenção deste artigo é responder a essas questões e mostrar, no campo, como funcionam e quais são as características das principais filosofias de controle: PLL, Proporcional, Integral, Derivativo, e PID. Como sempre, com propostas de soluções práticas.

Alexandre Capelli

Controle Phase-Locked Loop (PLL)

A primeira filosofia de controle que veremos é o PLL. Esse nome pode ser traduzido de várias formas, mas, "elo travado em fase", no meu entender, é o mais adequado.

Mas o que é um elo travado em fase, afinal?

O PLL é um sistema fechado de controle, cujo funcionamento é baseado na detecção da diferença de fases entre a saída e a entrada de dois sinais, controlados através de um oscilador.

Como mencionado logo na introdução deste artigo (e aproveitando, agora para explicá-lo melhor), você não encontrará muitas fórmulas e teoremas matemáticos nessa matéria. Isso não significa, entretanto, que não os julguemos importantes, apenas desejamos passar ao leitor uma visão mais prática sobre o tema , uma vez que a teórica é relativamente fácil de ser obtida.

Para abordarmos sobre a técnica de PLL, acredito que "vale a pena" estudar um pouco sobre sua história.

Antes de ser utilizada no controle de movimento, a técnica de PLL foi (e até hoje ainda é) muito usada nos equipamentos de RF, principalmente em rádios. Na verdade, no início de 1930, o rádio super-heteródino dominava o mercado. A própria RCA (*Radio Corporation of America*) monopolizava o mercado deste tipo de equipamento que já ultrapassa o ambiente residencial. A técnica do super-heteródino, por exemplo, foi muito útil nos sistemas de micro-ondas dos radares, desenvolvidos na Segunda Guerra Mundial.

Contudo, devido ao grande número de estágios de um receptor super-heteródino (frequências intermediárias), um método mais simples era necessário. Em 1932 um time de engenheiros ingleses desenvolveu um novo receptor, chamado "homódino", mais tarde renomeado "sincródino". Basicamente, era composto de um oscilador local, um *mixer*, e um amplificador de áudio.

Quando o sinal recebido era misturado ao sinal do oscilador local, na mesma fase e frequência (principal diferença com o super-heteródino), a saída era o sinal de áudio pronto para ser amplificado. O único problema com esse método era garantir a precisão da frequência de oscilação do oscilador local. Erros de "drift" nesse módulo comprometiam a recepção.

A solução foi mudar o oscilador tradicional para um oscilador controlado por tensão (VCO). Dessa forma, as frequências do sinal recebido e do mixer eram comparadas via um detector de fase. O "erro" entre elas gerava um sinal DC de correção que atuava sobre o VCO, corrigindo assim o erro de drift.

Resumindo, o PLL é composto por um oscilador controlado por tensão, cuja frequência é oriunda de um sinal DC de controle, gerado a partir da diferença de fase entre o sinal recebido e o próprio oscilador local. Caso ambos os sinais estejam em perfeito sincronismo, a tensão de correção (controle) DC deverá ser nula.

Curiosamente, o primeiro artigo escrito de que se tem registro a respeito dos princípios de PLL foi de autoria do francês H. de Bellescise, em "La Réception Synchrone".

Algum tempo depois, em meados de 1940, o PLL começou a equipar circuitos dos osciladores verticais de TV, chamados de "Synchro-Lock".

A partir daí a tecnologia PLL foi estendida para outras áreas, tais como: satélites, decodificadores FSK, controle de motores, robótica, e receptores AM e FM.

Embora o foco deste artigo seja o controle de motores e servomecanismos, é nos receptores de rádio que podemos contemplar bem esta evolução. Quem mora na cidade de São Paulo, por exemplo, e tem mais de trinta anos de idade, já deve ter notado a diferença de performance dos rádios automotivos atuais quando se transita pela Avenida Paulista.

Considerado um dos locais mais poluídos com interferências eletromagnéticas no mundo, devido a maciça presença de antenas transmissoras (TVs, rádios, serviços, telefonia, etc.), não permitia o funcionamento de nenhum rádio por mais de 10 m de trajeto. Nem mesmo o famoso "road-star"conseguia um bom resultado. O jeito era desligar o rádio (ou colocar uma fita cassete) até sair do perímetro próximo a região. Hoje, porém, um rádio ("normalmente também equipado com CD") opera sem interferências do lado do prédio da Gazeta! Como? Com o PLL.

Como funciona o PLL?

A **figura 1** ilustra o diagrama de um PLL básico, onde temos: um detector de fase, um filtro passabaixa, e um oscilador controlado por tensão. Sendo:

- Ve = sinal de entrada
- Vd = sinal de detector de fase para o filtro
- Vc = tensão de controle DC para o VCO
- Vo = sinal de retorno (saída)

Temos que, na ausência de sinal de entrada Ve, a tensão Vc é zero, e o VCO oscila na frequência central fo . Caso a frequência do sinal de entrada (fe) varie, uma tensão Vd surgirá na saída do detector de fase e, após filtrada, se transformará em uma tensão de controle Vc.

Essa tensão controlará o oscilador VCO, de modo a manter a frequência de entrada igual a de saída (fe = fo), porém, com uma defasagem fixa entre ambas.

A **figura 2** mostra um exemplo típico de defasagem de 90 °, onde, nessa condição o circuito fica estável, sincronizado e travado em fase. Em outras palavras, o PLL mantém as frequências de entrada e saída iguais às custas de uma defasagem fixa.

É bom lembrar, entretanto, que a capacidade de manter a estabilidade é limitada segundo uma faixa de variação de frequência do sinal de entrada. Esse parâmetro é conhecido como "lock range", e a **figura 3** exibe um exemplo de comportamento do PLL, segundo Vc em função de fe.

Comportamento Dinâmico

O circuito PLL tem, por excelência, um comportamento dinâmico. Para cada variação de frequência do sinal de entrada, uma resposta imediata da tensão de controle para o oscilador ocorre. É fato, contudo, que o sinal de entrada pode ter sua frequência invariável (fixa).

Alguns autores dizem que, nessa situação, o comportamento do PLL é estático, e apenas quando variações sucessivas de fe acontecem é que ele pode ser considerado dinâmico.

Particularmente, prefiro dizer que, quando temos fe fixa, o PLL é "menos dinâmico"(se é que isso existe!) do que quando fe varia significativamente.

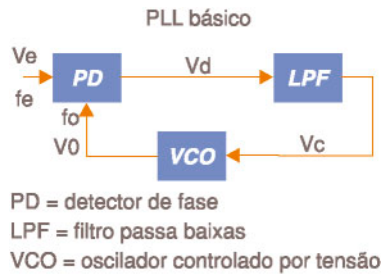
A preocupação com isto não é em vão, porém. Através da **figura 4** podemos notar o comportamento do PLL sob variações severas de fe. Como os circuitos não são perfeitos, atrasos e defasamentos são introduzidos à resposta do circuito em variações muito rápidas. Com certeza, este é um fator a se considerar na utilização desta técnica.

Construindo um PLL

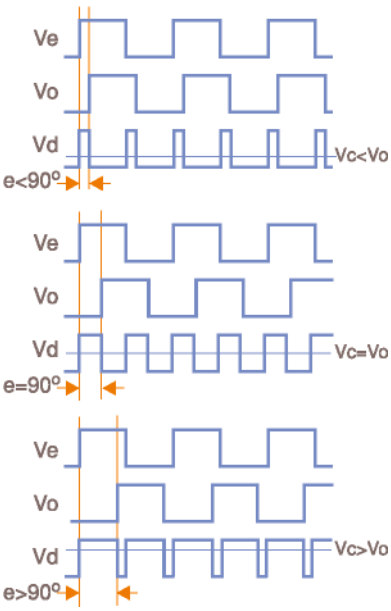
Construindo separadamente cada bloco e, depois, interligando-os.

O bloco detector de fase pode ser construído com uma simples porta "OU"-exclusivo. O filtro pode ser feito com um circuito RC. Os dois blocos já unidos são mostrados na **figura 5.**

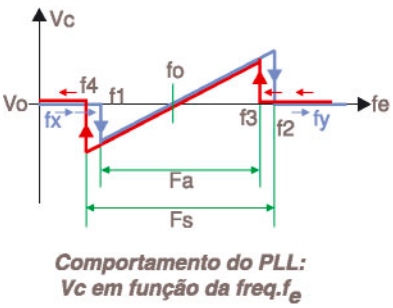
Quanto ao VCO, conforme ilustra a **figura 6**, é montagem com um FET cujo *gate* é excitado por uma rede de deslocamento



F1. Diagrama de um PLL básico.

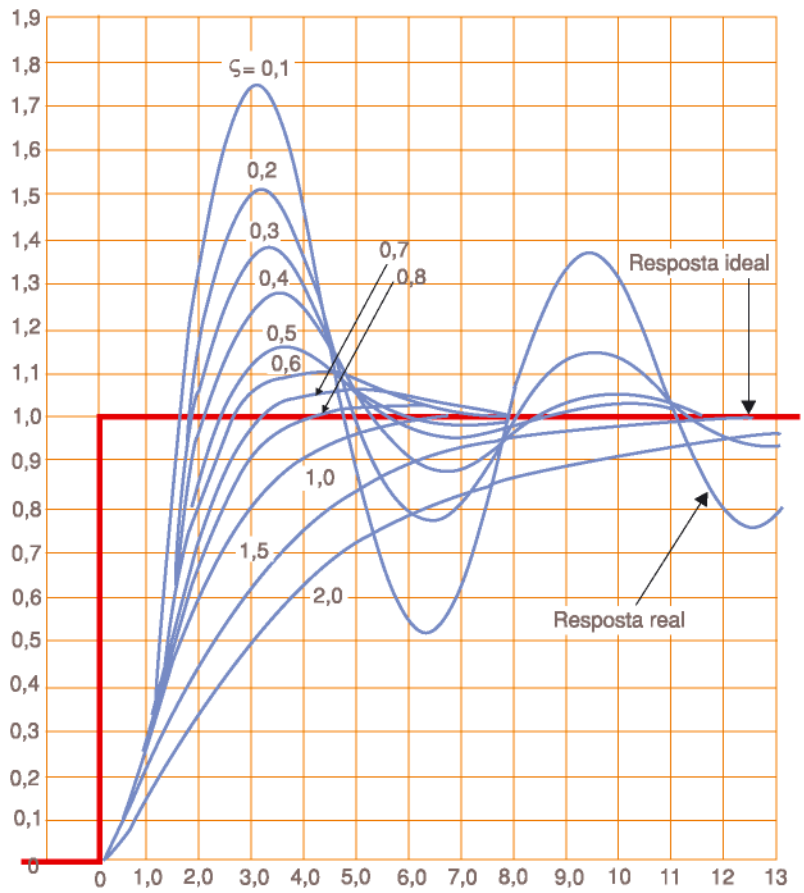


F2. Exemplo típico de defasagem de 90°.



Comportamento do PLL: Vc em função da freq.f_e

F3. Comportamento do PLL.



F4. Comportamento dinâmico.

de fase. Notem que um dos capacitores é um *varicap*, que muda sua capacitância segundo a tensão a qual está submetido. Alterando sua capacitância, automaticamente, a frequência de operação do VCO muda na mesma proporção. Estrategicamente, a tensão de controle é alocada exatamente sobre o varicap.

Não há um modo mais simples?

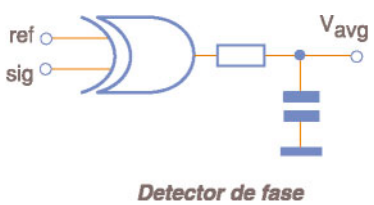
Sim, utilizando circuitos integrados projetados para essa função. A **figura 7** nos traz dois exemplos da National, o LM 565 em encapsulamentos "metal can"e "dual-in-line". Outros tipos famosos são: NE560 a NE567 (Signetics), MC 4046 (Motorola) e NTE 989 (NTE Electronics).

Acionamento CA com PLL

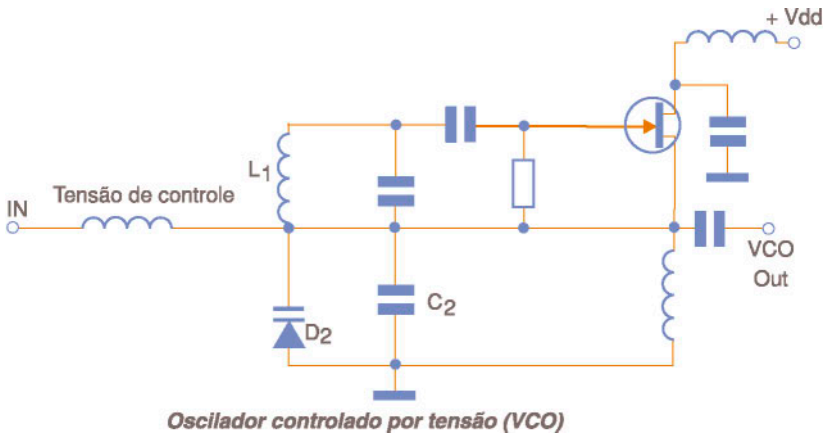
A **figura 8** exibe a estrutura básica de um acionamento do motor elétrico CA trifásico, cujo princípio de funcionamento baseia-se em PLL. A vantagem deste tipo de equipamento sobre um acionamento convencional é uma maior estabilidade do torque mediante uma variação abrupta na carga.

A etapa de potência do dispositivo é feita por um conversor tipo Buck, excitado por PWM. A corrente do motor é monitorada por TCs (transformadores de corrente) e configuram "uma"das entradas do módulo PLL. A outra entrada é o sinal do *encoder* que, nesse sistema, informa a velocidade de rotação.

Podemos analisar o comportamento dinâmico desta técnica através da **figura 9.**



F5. Detector de fase.



F6. Oscilador controlado por tensão (VCO).

Observem que a posição rotórica tende a permanecer constante segundo a variação de carga. Tendo como ponto de partida a situação "a", onde o motor está sob regime constante de carga, notamos a curva "L"ao centro do gráfico. Quando reduzimos a carga repentinamente (situação "b"), a corrente de fase também diminui.

Isso acontece devido a resposta do circuito PLL, que atua sob o circuito PWM. Este último controla a potência sobre o conversor Buck.

Por outro lado, quando a carga é acrescida em um intervalo de tempo pequeno, a corrente aumenta, causando o efeito contrário.

É fácil visualizar que o deslocamento para a direita da posição rotórica (situação b), e para a esquerda (situação c) não foram significativos. Em outras palavras, mesmo após variação da carga, o torque tendeu a se manter constante.

Quanto mais rápido for o circuito PLL, tanto menor será a variação do torque.

Esse tipo de acionamento é muito utilizado em servomotores que atuam no movimento do eixo (máquinas-ferramenta, por exemplo), ou em sistemas onde a precisão de velocidade de rotação é necessária sem, contudo, exigir um sistema vetorial.

Mais adiante mostraremos um quadro comparativo entre essa filosofia de controle e as demais.

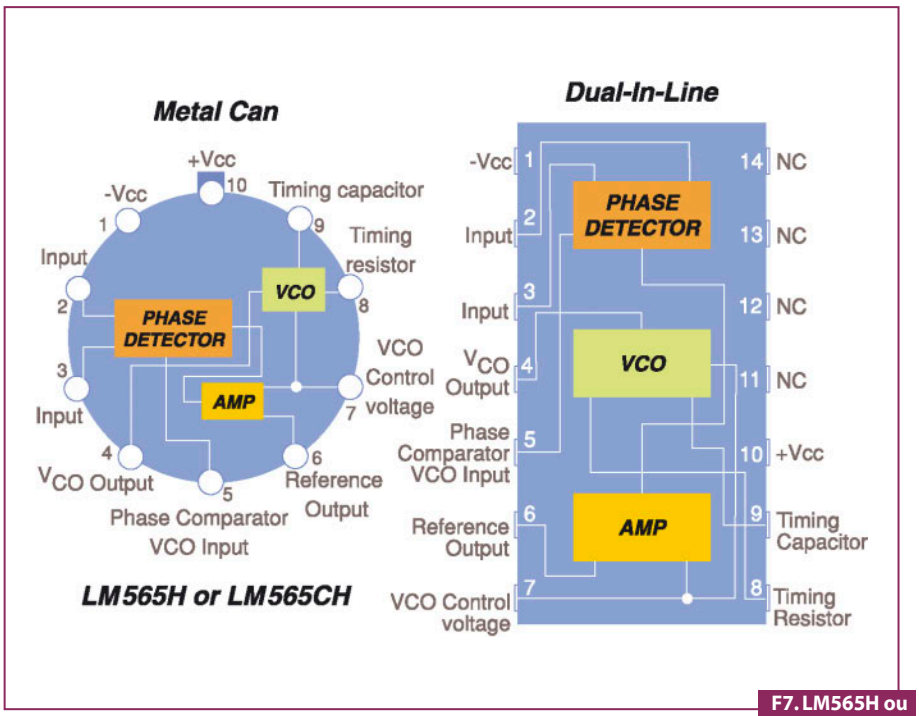
Controle ON-OFF

Outro método de controle, mais simples que o PLL, é o tipo "on-off". Devido a sua simplicidade as variações são mais bruscas, e a ação menos precisa. Porém, há casos em que ele é suficiente para um bom resultado.

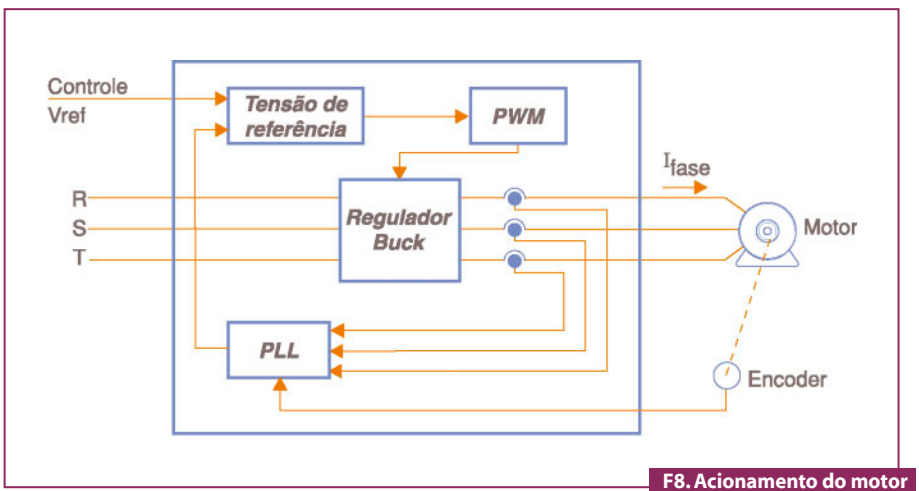
O que o leitor deve ter em mente é a estrutura básica de um sistema de controle. Não importa qual tipo de ação esse controle exerça, essa estrutura é sempre a mesma. A **figura 10** ilustra o diagrama de blocos básico. Três são os parâmetros que comandam o processo:

- Valor desejado – *set point (SP)*
- Valor real (PV)
- Algoritmo de controle.

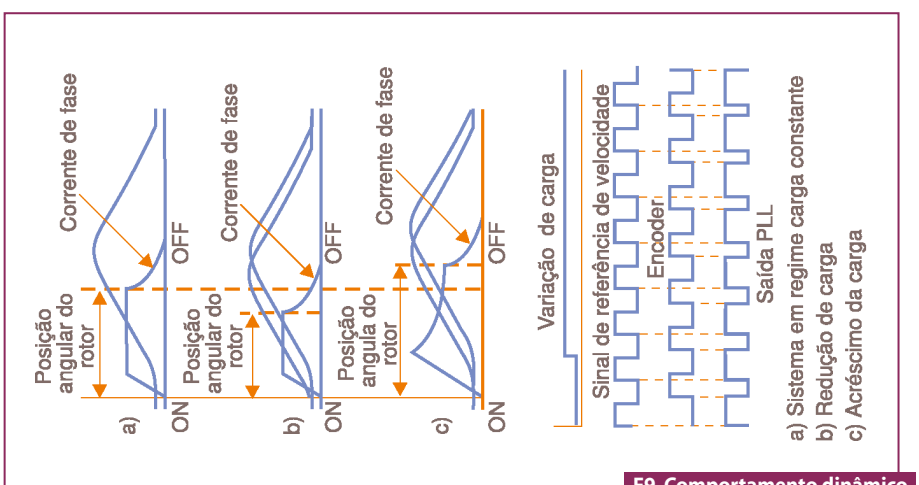
A ideia básica é simples, o controlador compara o valor desejado (SP) com o valor do processo (PV) e, através do algoritmo de



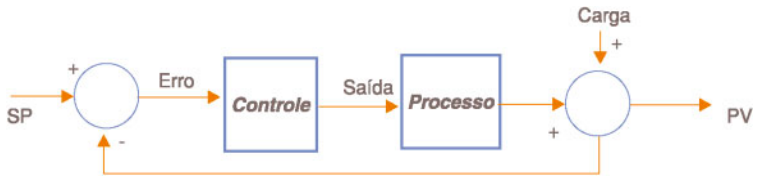
F7. LM565H ou LM565CH.



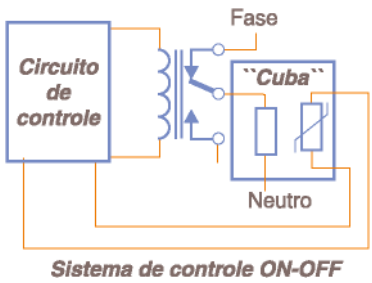
F8. Acionamento do motor elétrico CA trifásico.



F9. Comportamento dinâmico do acionamento CA com PLL.



F10. Diagrama de blocos do controle ON - OFF.



Sistema de controle ON-OFF

F11. Sistema de controle ON - OFF.

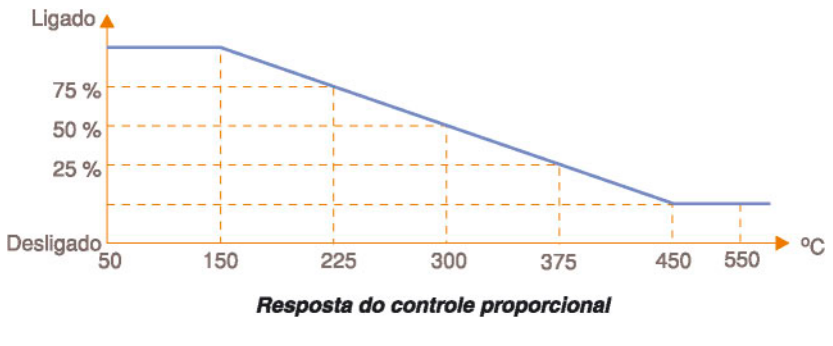
controle, faz as devidas correções a fim de "PV" aproximar-se ao máximo de "SP".

O PLL, on-off, ação proporcional, entre outros, portanto, são os algoritmos de controle. Cada qual com sua filosofia de executar a correção.

A **figura 11** apresenta um esquema básico de controle on-off. Uma "cuba" tem sua temperatura controlada por meio de um resistor de aquecimento e um termistor. Programado determinado valor, ou melhor faixa de valores, assim que a temperatura atinge o set-point, o relé é desligado.

Uma vez que ela decaia, o relé ativa o resistor novamente. Esse controle é muito comum nos equipamentos de refrigeração residenciais e industriais.

A principal vantagem do "on-off" é sua simplicidade, e sua desvantagem é a falta de precisão e linearidade.



Resposta do controle proporcional

F12. Resposta do controle proporcional.

Controle Proporcional

Nem sempre é possível controlar um sistema através do on-off, entretanto. Muitas vezes, necessitamos de uma ação mais suave. Uma filosofia do controle deste tipo é a proporcional.

Sua ação pode ser vista na **figura 12.** Assim como no on-off, também há um momento de desligamento total, ou acionamento total, mas, nesse caso, há uma curva entre os extremos.

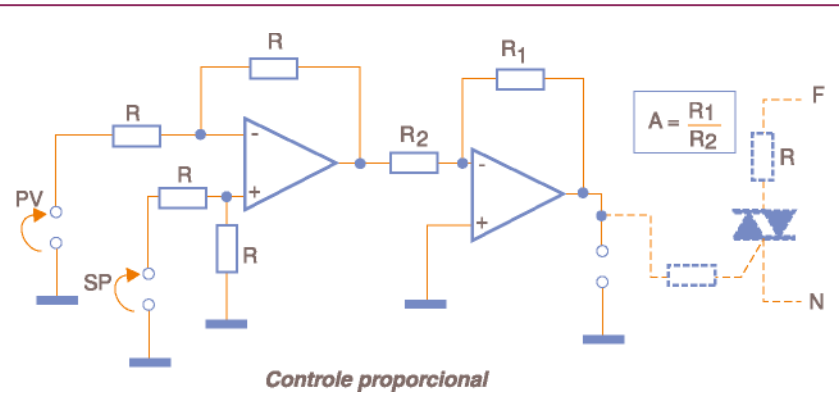
Se fôssemos utilizar o mesmo exemplo da "cuba" teríamos de substituir o relé por um TRIAC e, ao invés de simplesmente ligá-lo (ON) ou desligá-lo (OFF), controlaríamos seu ângulo de disparo.

Ainda com base na curva, podemos ver no exemplo que a temperatura de operação varia de 50 °C a 550 °C. Já o raio de ação do controlador limita-se à faixa de 150 °C a 450 °C, então, entre os extremos ligado e desligado existe uma região sem controle proporcional.

O ponto central é 300 °C, com uma faixa de atuação de ± 150 °C.

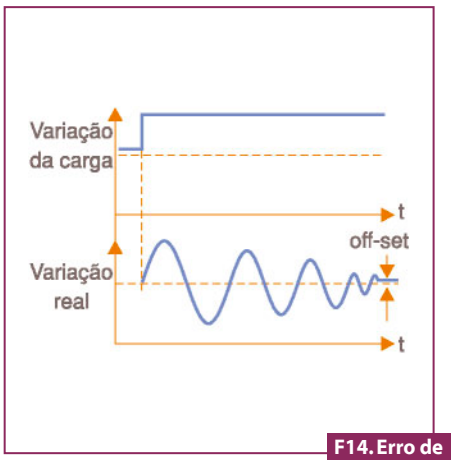
Caso a temperatura caia abaixo dos 150 °C, o "TRIAC" entrará em plena condução (ângulo de disparo próximo a 0°). Por outro lado, no intervalo, sua ação é linear (suave) até atingir a temperatura de 450 °C. Neste ponto, o controle desligará o TRIAC.

A porcentagem de "ação ativa" neste caso pode ser determinada através dos valores limites. Ou seja, o "range" de controle proporcional é 300 °C (450 °C –°C). Dividindo esse valor pela variação de temperatura total do processo 500 °C (550 °C –°C), teremos 300/500 = 60 %.

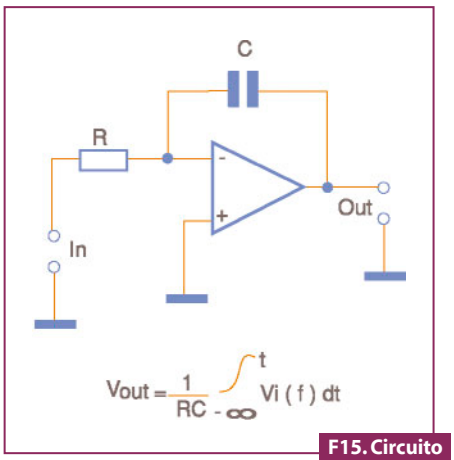


Controle proporcional

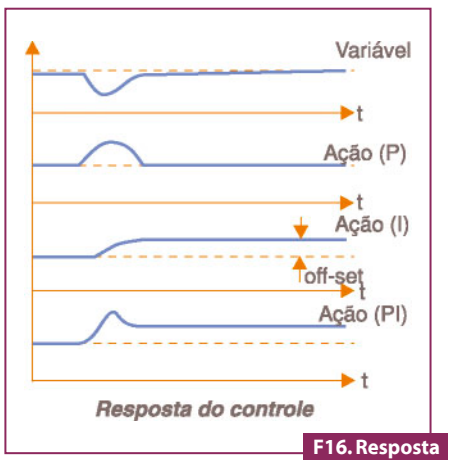
F13. Controle proporcional.



F14. Erro de off-set.



F15. Circuito integrador.



F16. Resposta do controle.

Outra maneira de explicarmos o comportamento desse controlador é através do seu ganho (A), que é a relação entre a porcentagem de variação do elemento controlador pela variação proporcional da grandeza. Assim temos:

$$\text{Faixa proporcional} = \frac{100}{\text{ganho}}$$

No exemplo, o ganho é:

$$\frac{\text{100\% (total do processo)}}{\text{60\% (range de controlador)}}$$

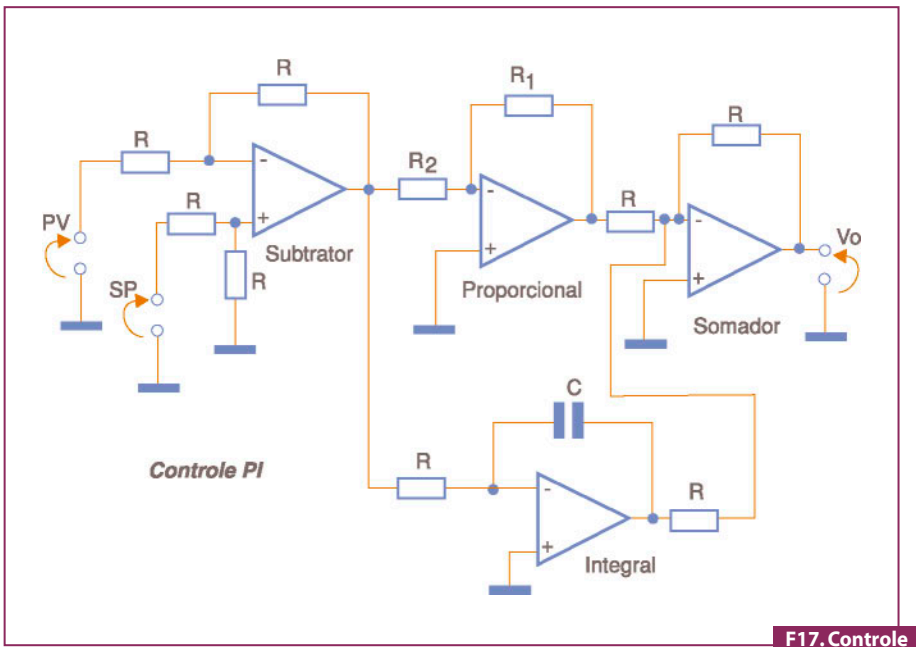
E como construir o circuito de controle proporcional?

Através de um circuito de diferença e um amplificador linear. A **figura 13** mostra o modo clássico utilizando dois amplificadores operacionais. O primeiro deles faz a diferença entre o valor real e o desejado, que, então, é amplificada pelo segundo A.O. Embora não esteja na figura, o sistema é fechado, isto é, o sinal do termistor informa o valor real (PV) através da entrada inversora do primeiro operacional.

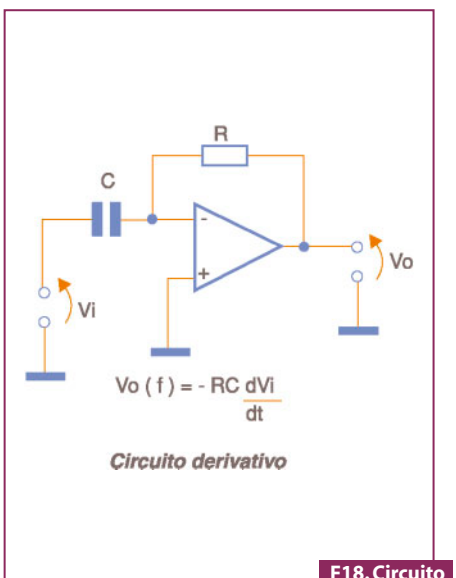
Os valores dos resistores serão analisados mais adiante, através da tabela de ajustes e ganho.

Controle Integral

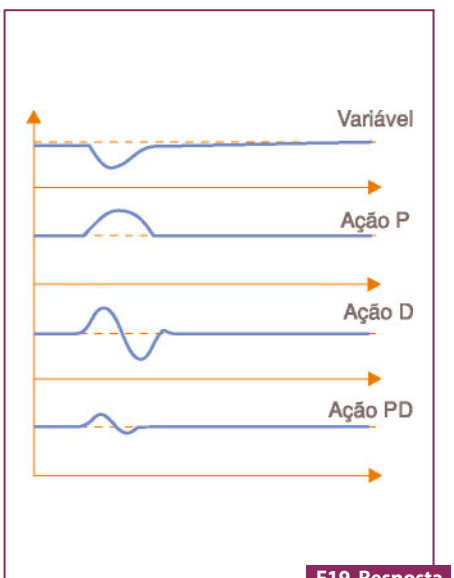
Outro conceito importante quando estudamos controle é o erro de "off-set". O erro de "off-set"é uma característica intrínseca a qualquer ação de controle de variáveis.



F17. Controle PI.

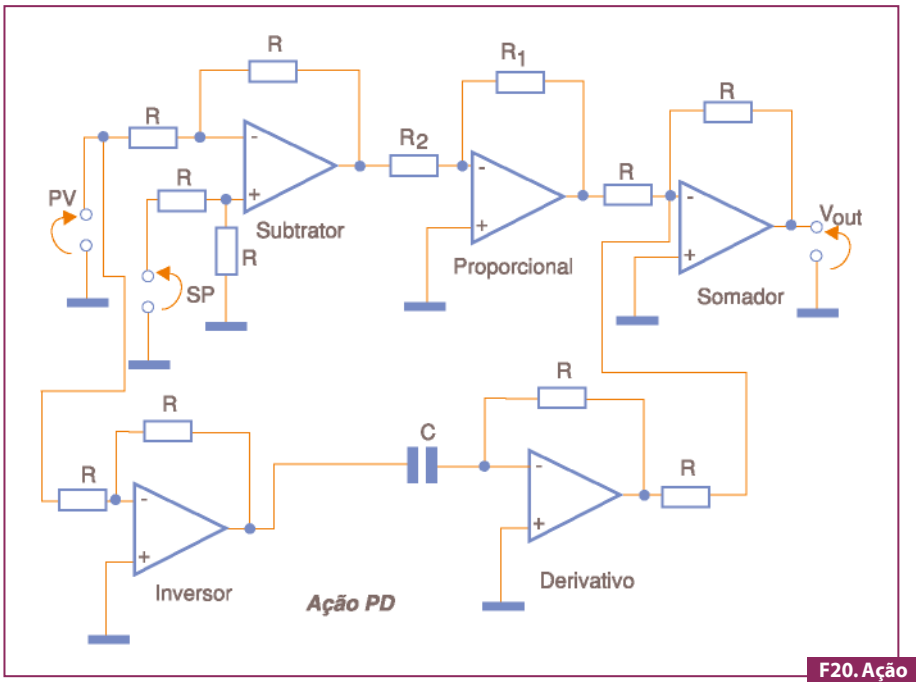


F18. Circuito derivativo.

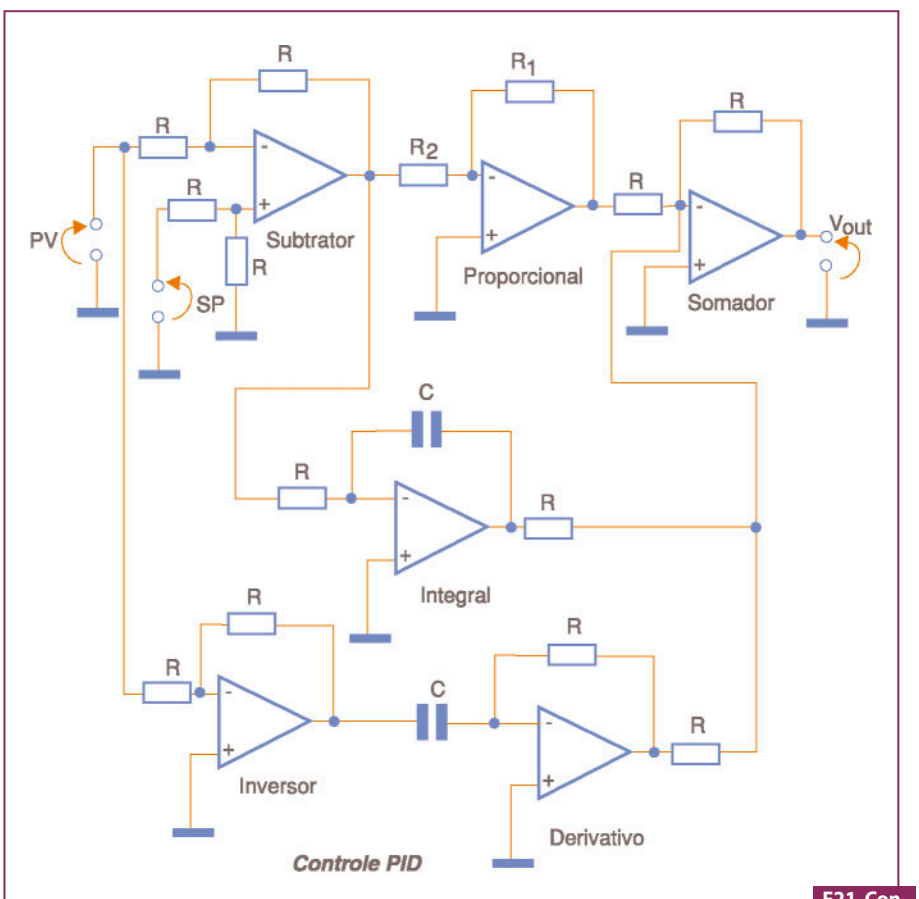


F19. Resposta do controle.

Em termos práticos quando uma grandeza (temperatura, rotação de um motor, etc) varia, a resposta do circuito não se estabiliza no exato valor desejado do set point (SP), mas sim ligeiramente fora dele. Isto ocorre porque a malha de controle (amplificadores operacionais, por exemplo) possui um ganho. O erro de off-set é inversamente proporcional ao ganho, pois os sinais são sempre tratados pelas entradas inversoras. Desta forma, quanto maior o ganho, menor o erro e vice-versa. **Figura 14.**



F20. Ação PD.



F21. Controle PID.

Então, basta um circuito com ganho quase infinito, para o erro ser quase zero, certo?

Teoricamente sim, mas na prática, quando aumentamos o ganho, o sistema começa a oscilar e perde estabilidade. Esse parâmetro, portanto, deve ser ajustado de modo a prover o menor erro segundo a melhor estabilidade.

No controle proporcional isto é feito manualmente, mas há sistemas que exigem uma correção de off-set automática e, mais do que isso, dinâmica. Para esses casos, utilizamos uma ação conjunta entre o controle proporcional e a ação integral, o chamado controle PI. Eletronicamente, o circuito de ação integral é o próprio integrador (**figura 15**).

Através da **figura 16** podemos contemplar as diferentes respostas das ações proporcional, integral e proporcional/integral segundo a variação de uma grandeza.

O circuito completo de um controlador PI pode ser visto na **figura 17**, onde notamos que apenas um somador une as ações proporcional e integral.

Controle Derivativo

A principal característica do controle derivativo é a correção proporcional à velocidade com que o desvio aumenta. Caso haja uma ação conjunta entre o proporcional e o derivativo, teremos uma correção antecipada a um desvio que ainda não aconteceu. Isto também é chamado de "supercorreção".

Eletronicamente, a ação derivativa pode ser implementada através de um simples circuito derivativo (**figura 18**).

Na **figura 19** observamos as diferentes respostas entre as ações P, D, e PD segundo a alteração de uma grandeza. Notem que há uma reposta mais rápida em relação a PI.

Um controlador PD pode ser construído conforme o circuito da **figura 20.** Neste caso também um somador articula as ações proporcional e derivativa.

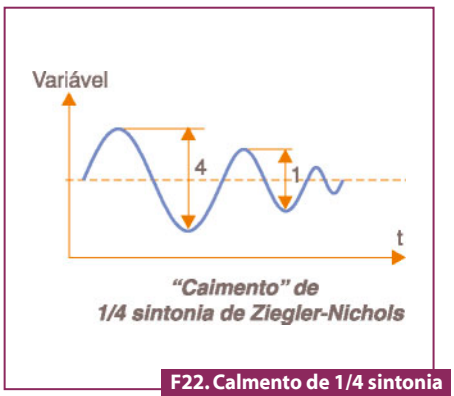
Controle PID

Há ainda uma outra necessidade de controle, onde processos com "tempo morto"(da ordem de dois minutos) são

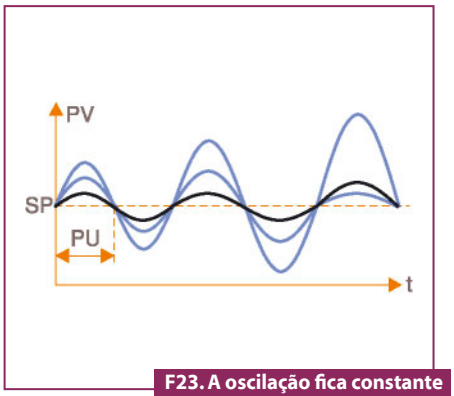
elevados. "Tempo morto"significa que a faixa proporcional a ser estabilizada é muito ampla e torna-se necessário um tempo de ajuste grande para evitar-se oscilações.

Os controles PI e PD não apresentam bons resultados nesses casos. A solução, portanto, é o circuito PID (proporcional/integral/derivativo) e que pode ser visto na **figura 21.** Assim como nos demais, uma etapa somadora articula as ações.

O melhor ponto de operação (otimização) para o controle PID ocorre quando temos um caimento de ¼ no tempo de ajuste segundo a variação da grandeza (**figura 22**).



F22. Calmento de 1/4 sintonia de Ziegler-Nichols.



F23. A oscilação fica constante em amplitude e período.

Mas como posso determinar os ganhos, e, consequentemente os resitores dessas malhas de controle?

O método de ajuste de controle em malhas fechadas mais conhecido e utilizado até hoje, foi desenvolvido em 1942 por J. G. Ziegler e N. B. Nichols, quando trabalhava na empresa americana "Taylor Instrument Company", na cidade de Rochester, Nova Iorque. Esse método, apesar de antigo, é o mais empregado por instrumentistas e na área de controle de processo, tanto nos discretos (manufatura), como nos contínuos.

O método consiste em determinar um ganho, chamado de ganho crítico (Gu), e um período designado como período crítico (Pu). Para isso, deve-se seguir alguns passos.

- Tira-se a ação integral e a derivativa do controlador, deixando apenas a proporcional.
- Mantém-se o controlador em modo automático em malha fechada.
- Ajusta-se o ganho do controlador em um nível baixo a fim de se evitar oscilações no sistema.
- Aumenta-se o ganho, passo a passo, até que a oscilação fique constante em amplitude e período, como é ilustrado na **figura 23**.

Com base no ganho crítico e no período crítico, os ajustes do controlador são calculados para um ideal do controlador.

Ziegler e Nichols observaram que em um controlador proporcional o ganho ideal é a metade do ganho crítico, isto é: Ganho (A) = Gu/2, com esse ganho obtemos, aproximadamente, uma razão de caimento de ¼.

Tenho que, efetivamente, fazer todos esses passos para otimizar meu controle? Não há um modo mais simples e objetivo?

Ziegler e Nichols descobriram que as equações mostradas na **tabela 1** fornecem bons resultados para controle em malha fechada. Basta, então, cumprir os respectivos ganhos.

Supondo que eu queira controlar motores através dos sistemas em malha fechada, como posso saber qual é a melhor filosofia segundo minha necessidade?

A **tabela 2** abaixo deve proporcionar uma comparação entre vantagens e desvantagens de cada uma, servindo como guia ao técnico ou engenheiro de aplicação.

Conclusão

A teoria, regras práticas e demais conceitos tratados nesta matéria, são válidos tanto para processos da manufatura quanto contínuos. O circuito de controle PID, por exemplo, é o mesmo para acionar um servomotor de um eixo robótico, ou um TRIAC para "resistência"de aquecimento de um forno.

Em breve, analisaremos outros sistemas, tais como V –I e EMF.

Esperamos, porém, ter desmistificado as famosas "funções de transferência"abordadas com tanta ênfase em teoria de controle, e mostrar um pouco sobre a aplicação real em campo. E

	Ganho proporcional	Tempo Integral	Tempo Derivativo
Controle (P)	Gu/2		
Controle (PI)	Gu/2,2	1,2 Pu	
Controle (PID)	Gu/1,7	2/Pu	Pu/8

T1. Equações para o controle

		Controle			
		P	PI	PID	PLL
Precisão	Alta				O
	Média		O	O	
	Baixa	O			
Velocidade	Alta	O		O	
	Baixa		O		
Boa performance em baixa rotação		O		O	
Custo	Alto				O
	Médio			O	
	Baixo	O	O		

T2. Comparativo para diversos controles.

Reduzindo a EMI
com componentes magnéticos

Newton C. Braga

Técnicas de layout e de distribuição de componentes em uma placa podem reduzir em muita a EMI gerada por circuitos. No entanto, quando mesmo essas técnicas não reduzem a EMI aos níveis desejados, componentes apropriados deverão ser usados. Uma família importante de componentes empregados na redução da EMI é aquela representada pelos tipos que se baseiam no magnetismo, ou seja, pelos componentes indutivos ou magnéticos.Neste artigo trataremos de alguns desses componentes, baseados em ampla documentação do catálogo de Indutores e Produtos Magnéticos da Vishay

Em artigo anterior, nesta mesma Revista, mostramos como os procedimentos apropriados de disposição de trilhas, componentes e circuitos numa placa de circuito impresso podem ajudar a reduzir de forma acentuada os problemas de EMI (Interferência Eletromagnética).

No entanto, nem sempre é possível reduzir os problemas aos níveis aceitáveis somente com esses procedimentos, caso em que componentes adicionais deverão ser empregados. Em especial destacam-se os componentes indutivos, ou seja, aqueles baseados em bobinas e núcleos de ferrite que, pelas suas propriedades, podem cortar ou desviar sinais de frequências que não devam ser irradiadas ou não devam passar para uma linha de alimentação.

Os componentes indutivos usados com a finalidade de se reduzir a EMI podem ser classificados em quatro grandes grupos de que trataremos a seguir.

Inutores

Dos componentes indutivos, os choques ou indutores são os mais utilizados. Esses componentes podem ser usados tanto como elementos de filtros quanto no armazenamento de energia, conforme mostra a **figura 1**.

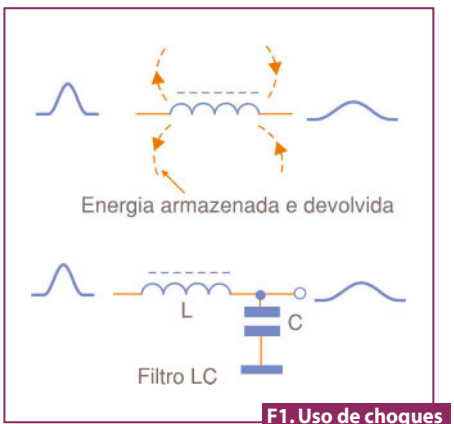
Quando o problema é interferência irradiada, os indutores mais eficientes são os do tipo toroidal, veja a **figura 2**.

Uma característica importante deste tipo de componente é que, pela sua construção ele é menos sensível à interferência que pode ser induzida pela presença de componentes próximos. De fato, se houver indução, o toroide faz com que as tensões induzidas tenham a mesma intensidade, mas fases opostas, cancelando-se, portanto.

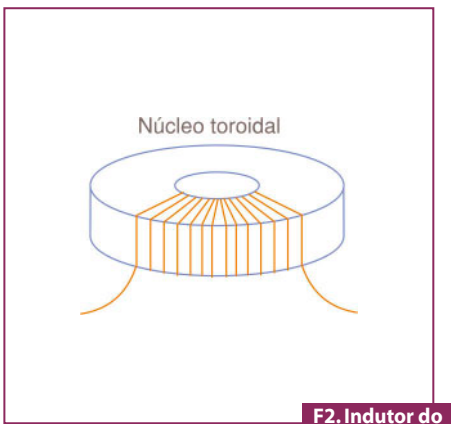
Choques em modo comum e diferencial

Os choques em modo comum e diferencias são empregados para eliminar ruídos que se propagam através de um par de condutores. Ruído em modo comum, conforme já vimos em artigo publicado nesta revista, é aquele que aparece nos dois condutores de uma linha, conforme ilustra a **figura 3**.

Os choques para rejeição de sinais em modo comum devem ser colocados



F1. Uso de choques ou indutores.



F2. Indutor do tipo toroidal.

os mais próximos quanto seja possível tanto do receptor quanto do transmissor. Também podem ser colocados na entrada de sinal na própria placa de circuito impresso.

Escolhendo um choque apropriado, é possível, inclusive, fazer o casamento de impedâncias entre a linha de sinal e o circuito.

Trasnformadores

Os transformadores têm como principal vantagem nos circuitos onde são usados o fato de proporcionarem isolamento entre a linha de sinal e o circuito de processamento desse sinal, conforme sugere a **figura 4**.

Outra vantagem está na eliminação das diferenças entre os potenciais de terra que poderão existir entre o circuito transmissor e o circuito receptor do sinal.

Um ponto importante em que a imunidade ao ruído se faz necessária, são os circuitos de excitação de tiristores.

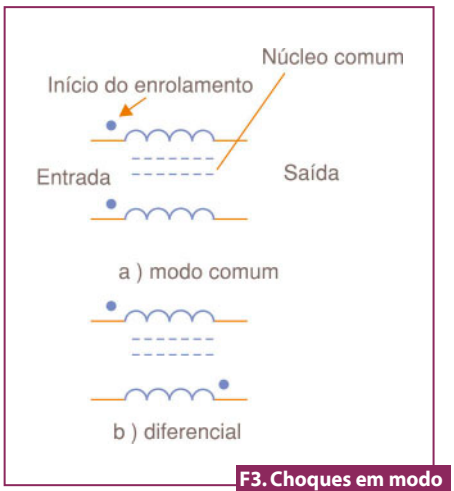
Além de proporcionar um isolamento entre o circuito de potência e o circuito de controle, normalmente baseado em lógica com microcontroladores ou microprocessadores, quando operando com pulsos, ele proverá uma imunidade muito maior para componentes como IGBTs, dada suas capacitâncias algo elevadas de entrada.

Ferrite beads

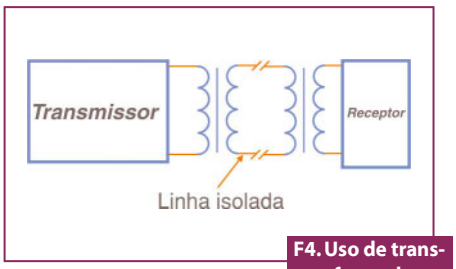
Os anéis de ferrite para montagem em superfície (SMD) ou *chips* de ferrite, como também são chamados, são usados para eliminar a RF que pode estar presente em linhas de sinais ou alimentação de placas de circuito impresso, observe a **figura 5**.

Os anéis de ferrite na forma de chips SMD se comportam como dispositivos de alta impedância para os sinais de RF, mas apresentam uma baixíssima resistência para a corrente contínua e sinais de baixas frequências.

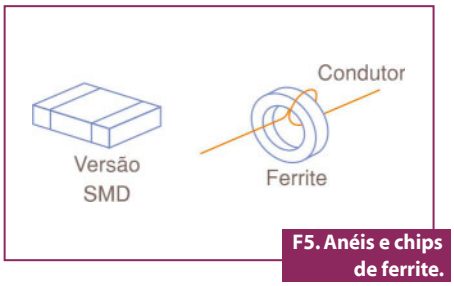
Esses componentes são especificados normalmente para a impedância que apresentam em uma frequência de 100 MHz. Assim, dependendo da aplicação, será necessário que o projetista analise os gráficos de comportamento dos componentes visados para verificar como eles se comportam na frequência dos sinais que devem passar ou que devem ser rejeitados.



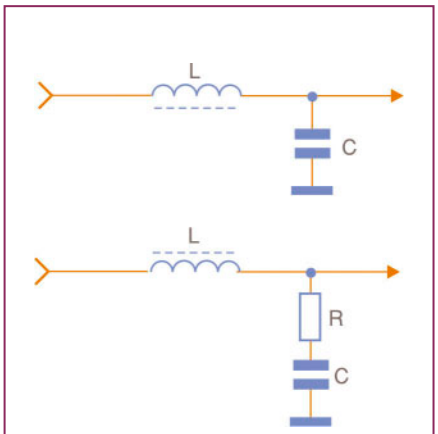
F3. Choques em modo comum e diferencial.



F4. Uso de transformadores.



F5. Anéis e chips de ferrite.



F6. Filtros LC e LCR.

Como selecionar componentes para evitar EMI

Antes de escolher quais componentes usar e de que forma, é preciso identificar os locais do circuito onde o ruído deve ser eliminado. Existem áreas dos circuitos que podem atuar como antenas irradiando os ruídos, assim como outras áreas que podem atuar como antenas recebendo os ruídos.

Então, o projetista deve fazer um estudo do melhor local para posicionar o componente que irá eliminar um possível sinal que esteja sendo transmitido ou captado indevidamente. Nos circuitos onde uma pequena atenuação é desejada, os componentes indutivos apenas já podem proporcionar os resultados desejados.

No entanto, nos circuitos mais complexos de alta velocidade, poderá ser necessário combinar componente indutivos com capacitivos, formando assim filtros. Filtros LC e LCR, veja a **figura 6**, devem ser considerados.

Um ponto importante a ser considerado na escolha dos componentes para o filtro é a estabilidade do terra do circuito. Para os sistemas estáveis, os componentes capacitivos já podem levar aos resultados esperados. Todavia, para terras instáveis, componentes indutivos de alta impedância são os recomendados. Alguns cuidados são necessários para o projeto:

- Os componentes que reduzem EMI devem ser colocados o mais próximo possível das fontes de ruído.
- Os componentes devem ser escolhidos de tal forma a casar sua impedância com a linha de condução em que eles forem colocados.
- O componente selecionado deve ser dimensionado apenas para reduzir nos níveis desejados os ruídos.

Conclusão

Os componentes indutivos ou magnéticos podem ajudar muito na redução dos problemas com a EMI. Porém, como existem diversos tipos com características que se adaptam à diferentes aplicações, é preciso saber escolher o tipo ideal

Lembramos que a inclusão de novos componentes em um circuito implica não só no aumento de custos como também na necessidade de um espaço maior, o que nem sempre é algo que se permite num projeto. Os limites para a inclusão de novos projetos e o aumento de custos são sempre muito estreitos.

Curso sobre o Osciloscópio

Parte 2

Dando continuidade ao estudo do Osciloscópio, abordamos nesta segunda parte do curso o "osciloscópio digital"

Eng. Filipe Pereira

Osciloscópio Digital

Um osciloscópio digital tem etapas parecidas com as de um osciloscópio analógico, embora funcione de forma totalmente diferente (como mostra a **figura 1**).

Em primeiro lugar, e à semelhança do osciloscópio analógico, tem-se uma etapa **atenuadora** e um **pré-amplificador** vertical que amplifica o nível de sinal ao valor adequado para as etapas seguintes.

Do pré-amplificador vertical, o sinal deriva em direção a dois circuitos: por um lado, aplica-se ao comparador que faz parte do **circuito de disparo**, mediante o qual selecionaremos o instante do sinal que dará lugar à varredura horizontal do *spot* na tela. Por outro, aplica-se a um **conversor analógico / digital** (A/D), no qual cada amplitude instantânea do sinal que se visualiza é convertida num código binário.

Para esta conversão é necessária uma amostra do sinal; isso quer dizer que se mede a amplitude do sinal em tempos iguais. Para levar a cabo esta amostra, sem erros de medida, é necessário que os pulsos de amostra tenham uma frequência extremamente estável. Consegue-se isso com um **oscilador a cristal**, que é disparado quando recebe o pulso do início de exploração do circuito de disparo.

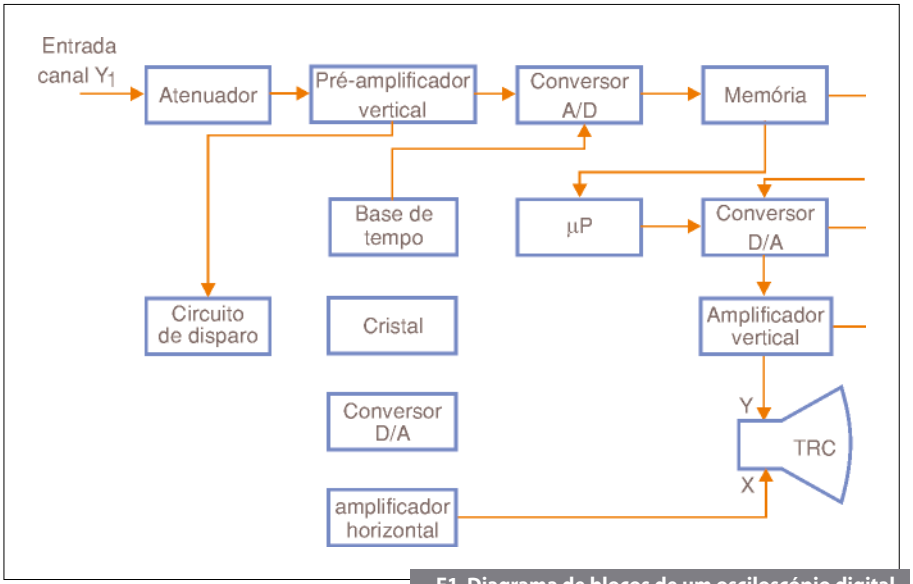
Observe-se no diagrama de blocos que o oscilador a cristal aplica também os seus pulsos retangulares a um **conversor digital/analógico** (D/A). Este conversor é necessário, já que para a varredura horizontal do spot necessitamos de uma tensão em dente de serra, e o oscilador a cristal gera apenas pulsos retangulares. Com o conversor D/A obtemos uma tensão em dente de serra muito estável e linear, válida para aplicar às placas de deflexão horizontal do tubo de raios catódicos previamente amplificadas ao valor adequado pelo **amplificador horizontal**.

Prossigamos o percurso do sinal em direção às placas de deflexão vertical do tubo.

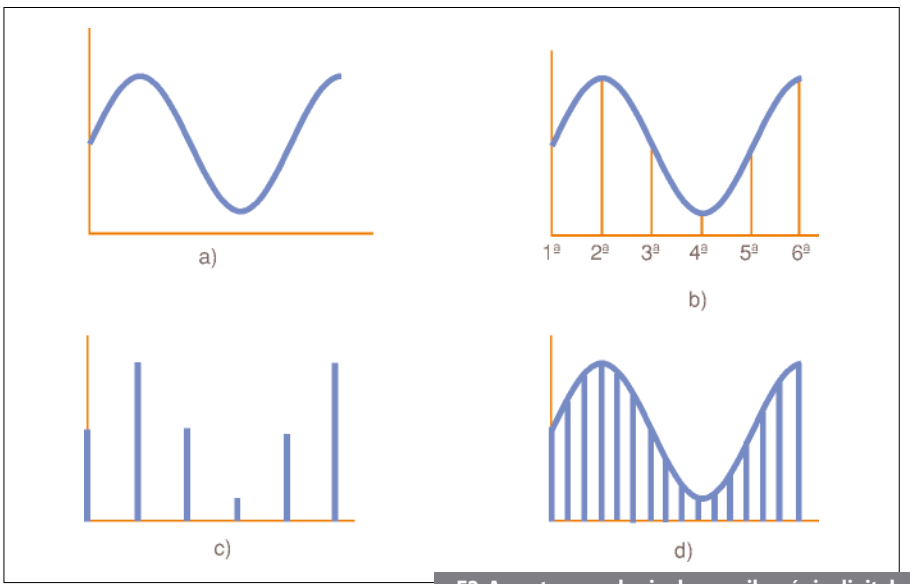
Frisamos que o sinal analógico foi convertido em pulsos binários à saída do **conversor A/D**. Assim pois, a partir deste ponto, o sinal pode ser tratado como qualquer sinal digital e guardá-lo num circuito de **memória**. Do circuito de memória o sinal pode tomar dois caminhos: um para o **conversor D/A** onde se converte de novo numa onda analógica de forma igual à original.

O outro caminho para o sinal digital é em direção a um microprocessador (μP).

O μP permite efetuar medidas automáticas dos parâmetros do sinal, tais como intervalos de tempo, tempos de subida e de descida, amplitude, frequência e visualizar estes dados na própria tela mediante caracteres alfanuméricos (osciloscópios Read - Out) ou enviá-los para uma impressora.



F1. Diagrama de blocos de um osciloscópio digital.



F2. Amostragem do sinal no osciloscópio digital.

Amostragem do sinal nos osciloscópios digitais

O conversor analógico - digital é a etapa chave do funcionamento dos osciloscópios digitais e tem por função transformar o sinal analógico numa série de pulsos elétricos.

Para que a amostra do sinal seja correta e se apresente na tela um sinal que corresponde ao real, a amostra deve realizar-se, pelo menos, a uma frequência dupla da maior harmônica do sinal (**figura 2**).

a) Onda senoidal que se pretende amostrar (**figura 2a**);

b) Pontos de amostragem. Na amostra da onda (**figura 2b**) observa-se que, desde que se inicia o ciclo até que volte a passar pelo mesmo ponto, efetuaram-se quatro amostras, e com elas obteremos um total de quatro valores ou amplitudes para codificar;

c) Pulsos obtidos como consequência dessa amostragem. Estes pulsos de diferentes amplitudes, serão os que se codificam de imediato em binário (**figura 2c**).

Se levarmos a cabo uma amostragem a uma frequência maior, por exemplo triplicando a frequência do oscilador de cristal, obteremos uma série de valores cujo conjunto se aproxima mais da forma de onda original.

Disto se deduz que, quanto maior for a frequência da amostragem, melhor será a informação que obtemos e mais próximo estaremos da forma de onda do sinal, pelo que poderemos reproduzi-la com a máxima fidelidade (**figura 2d**).

Como a frequência da amostragem está limitada tecnologicamente, para aumentar o número de amostras por período deve reduzir-se o limite da largura de banda do aparelho, com o fim de, pelo menos, poderem efetuar-se duas amostras por período.

O problema complica-se perante a presença de harmônicas do sinal de entrada. Um caso típico é o de uma onda retangular ou quadrada.

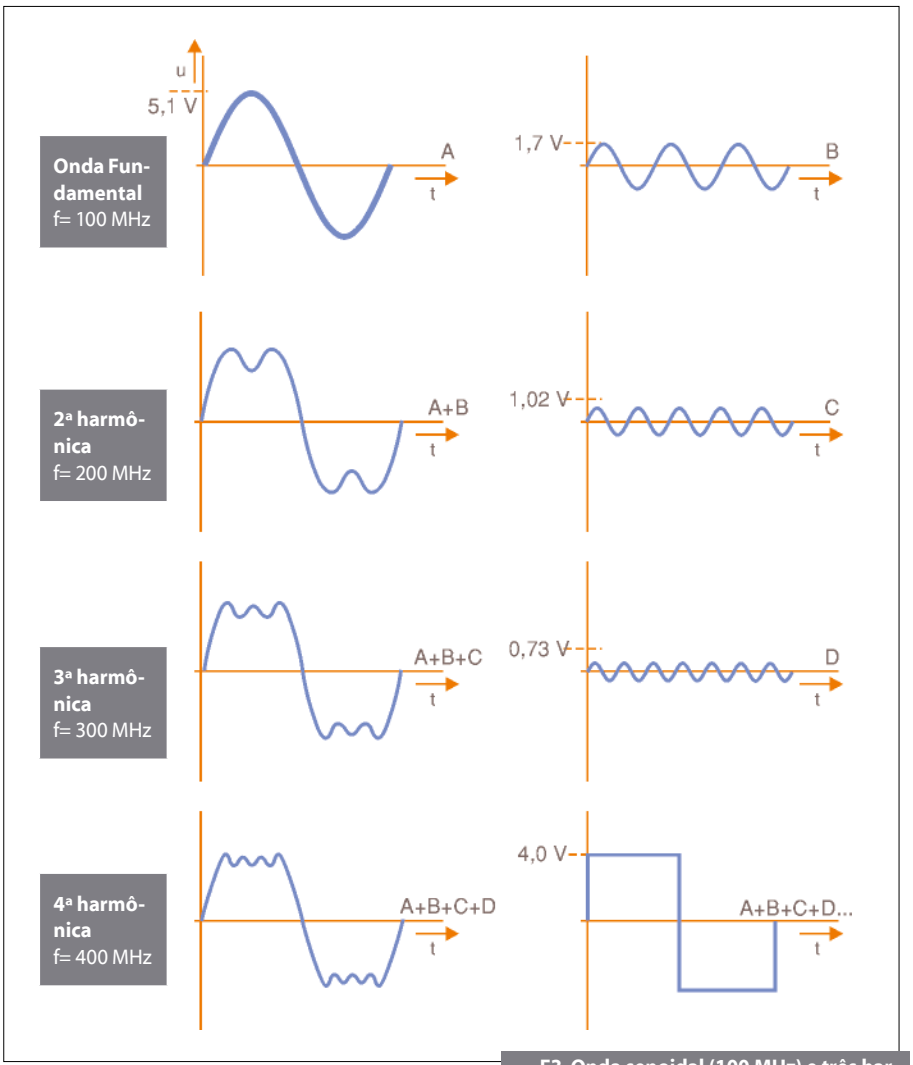
Toda a onda retangular é formada por uma onda senoidal de igual frequência e um número praticamente infinito de harmônicas desta onda fundamental, tal como se mostra graficamente na **figura 3**.

A velocidade de representação nos osciloscópios digitais é lenta, já que é a soma de vários tempos: de quantificação, de voltar à memória, de processamento do sinal e de conversão do digital a analógico.

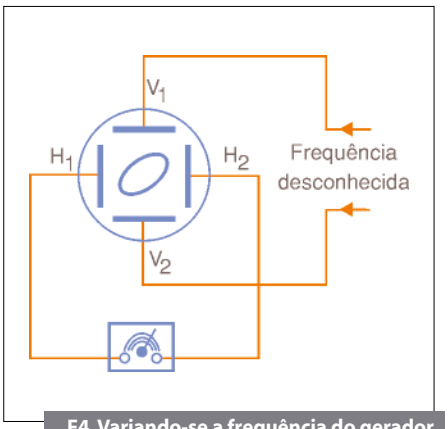
Medida de Frequências

Para medir a frequência de um sinal com o osciloscópio pode-se proceder das seguintes maneiras:

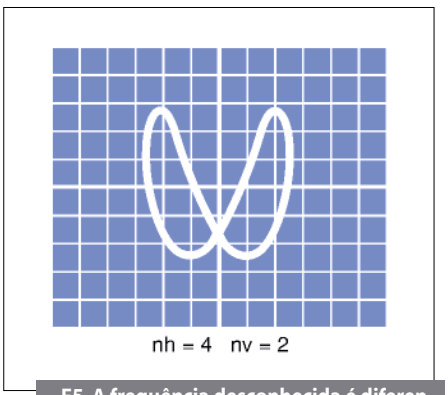
- Com a base de tempo calibrada;
- Por meio das figuras de Lissajous.



F3. Onda senoidal (100 MHz) e três harmônicas em 200; 300 e 400 MHz.



F4. Variando-se a frequência do gerador para obter uma circunferência ou elipse na tela.



F5. A frequência desconhecida é diferente das proporcionadas pelo gerador.

Com a base de tempo calibrada

Este método é o mais simples. Consiste simplesmente em aplicar o sinal a uma entrada vertical e ajustar o varredura até se obter na tela um ou mais ciclos do sinal.

Obs.: O número de ciclos que aparece na tela é o que corresponde ao número de vezes que a frequência do sinal é superior à frequência de varredura.o. Se ambas as frequências são iguais, obtém-se um só ciclo. Se não se consegue obter um só ciclo na tela é porque a frequência de varredura não alcança o alto valor da frequência do sinal.

Contando o número de divisões de um ciclo do sinal e multiplicando-o pela velocidade de varredura da base de tempo (s/div., ms/ div., s/ div.) obtemos o Período (T), fazendo o seu inverso obtemos a frequência do sinal (f = 1/T).

Por meio das figuras de Lissajous

Pressionar o botão XY (que desliga a base de tempo) e aplicar o sinal de frequência desconhecida às placas de deflexão vertical do osciloscópio. A seguir, aplica-se o sinal de um gerador de sinais às placas de deflexão horizontal do osciloscópio.

Efetuadas estas operações, faz-se variar a frequência do gerador de sinais até se obter uma circunferência ou uma elipse na tela.

Quando isso se consegue indica-nos que a frequência desconhecida do sinal é a mesma que a do gerador de sinais, se os dois sinais aplicados são senoidais.

Se, com a gama de frequências do gerador de sinais não se conseguir a citada circunferência ou elipse é porque a frequência desconhecida é distinta das proporcionadas pelo gerador de sinais (**figura 5**).

Observando a imagem na tela, é possível deduzir a relação de frequências entre os dois sinais aplicados.

Na **figura 6** mostram-se exemplos.

Para obter o valor da frequência desconhecida, basta contar o número de vezes que o traço cruza o eixo de simetria horizontal da tela, e dividi-lo pelo número de vezes que cruza o eixo de simetria vertical dela. O quociente é multiplicado pela frequência conhecida proporcionada pelo gerador de sinais, obtendo-se o valor da frequência desconhecida.

$$f_x = f_g \, n_h / n_v$$

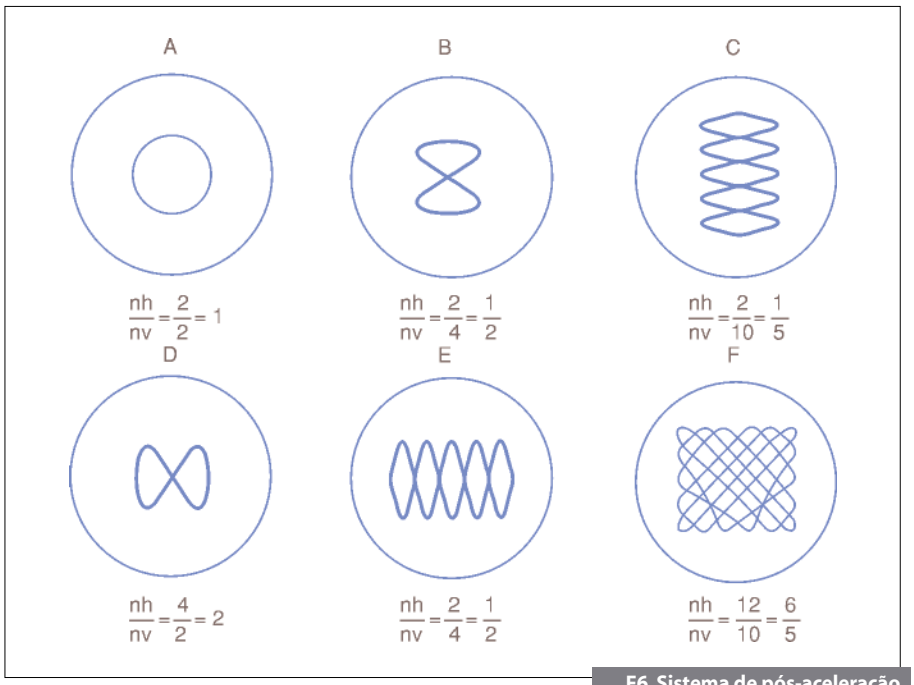
onde:

f_x é a frequência desconhecida;

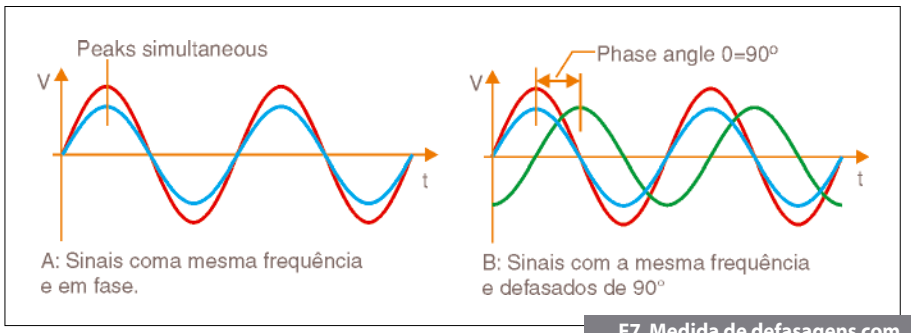
f_g é a frequência proporcionada pelo gerador de sinais;

n_h é o número de vezes que o traço cruza o eixo de simetria horizontal da tela;

n_v é o número de vezes que o traço cruza o eixo de simetria vertical da tela.



F6. Sistema de pós-aceleração com tensão muito elevada.



F7. Medida de defasagens com a base de tempo calibrada.

Por meio das figuras de Lissajous

Tendo em conta que os sinais aplicados ao osciloscópio são da mesma frequência, na tela aparecerão círculos ou elipses com diferentes graus de excentricidade. A excentricidade da elipse na tela determina a defasagem entre os dois sinais analisados.

Na **figura 8**, mostram-se algumas imagens, nas quais se indicam os seus defasamentos. O osciloscópio no modo XY, permite-nos determinar rapidamente o desfasamento existente entre dois sinais.

Aplicação simultânea de duas tensões alternadas senoidais da mesma frequência e em fase

A imagem que surge na tela é um traço oblíquo da direita para a esquerda e cuja

Oscilos-cópios digitais	em **tempo real.**	*Fazem-no à frequência necessária para representar corretamente num único ciclo de amostragem o sinal de entrada.*
	em **tempo equi-valente.**	*Usam amostras do sinal em diferentes pontos, em cada ciclo de amostragem. Posteriormente, reconstituem em memória um ciclo com todas as amostras. Este tipo de amostragem tem o inconveniente de que só é utilizável com sinais periódicos, mas oferece a vantagem de aumentar a largura de banda do aparelho.*

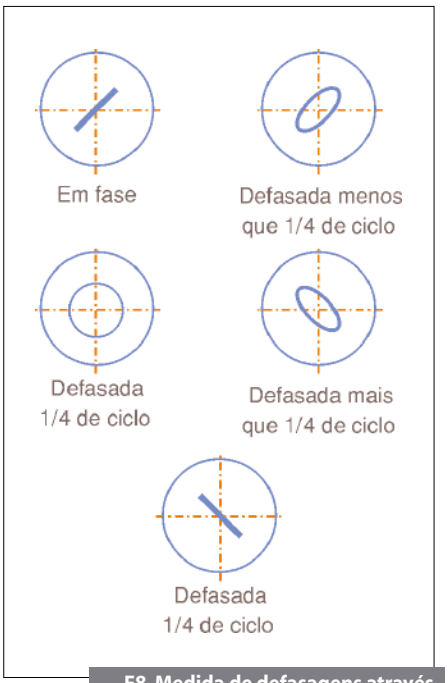
Procurando um público qualificado para o seu produto?



Anuncie na

MECATRÔNICA Atual
Automação Industrial de Processos e Manufatura

A número 1 em Automação Industrial
Tel.: (11) 2095-5339
publicidade@editorasaber.com.br
www.mecatronicaatual.com.br



F8. Medida de defasagens através das figuras de Lissajous.

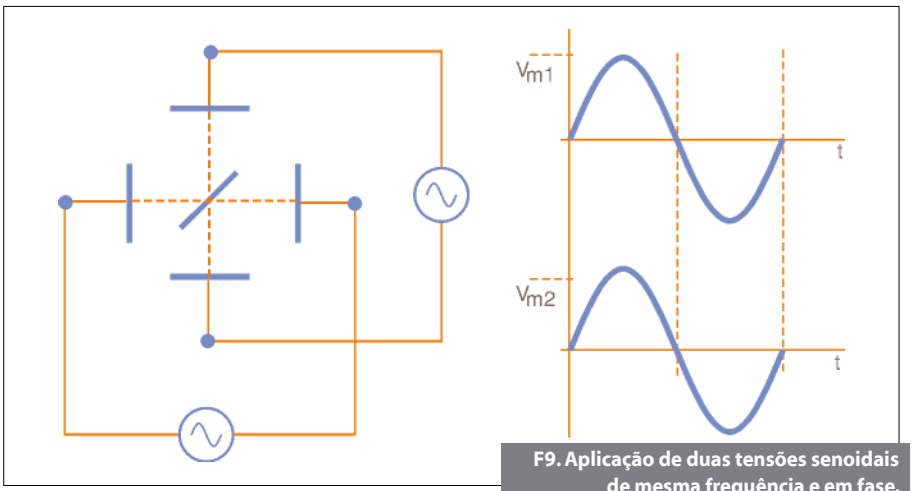
inclinação depende da relação entre as amplitudes das tensões.

Aplicação simultânea de duas tensões alternadas senoidais da mesma frequência defasadas de 1/4 de ciclo

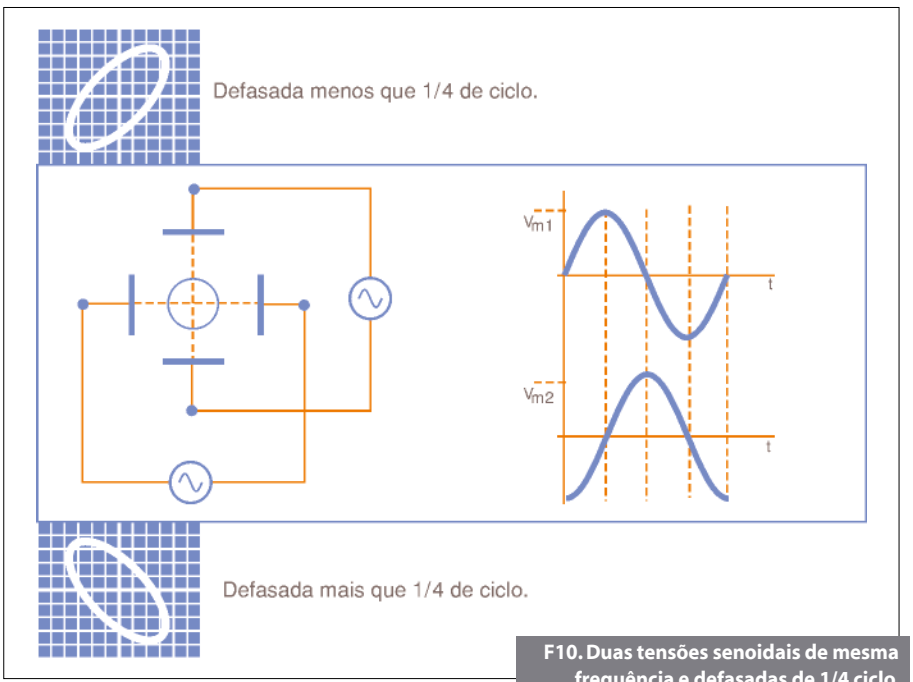
A imagem resultante é uma circunferência se as tensões forem da mesma amplitude. Sendo de amplitudes diferentes, a figura é uma elipse de eixo maior vertical ou horizontal conforme a tensão de maior amplitude for aplicada às placas verticais ou horizontais (**figura 10**).

Aplicação simultânea de duas tensões alternadas senoidais da mesma frequência defasadas de 1/2 ciclo

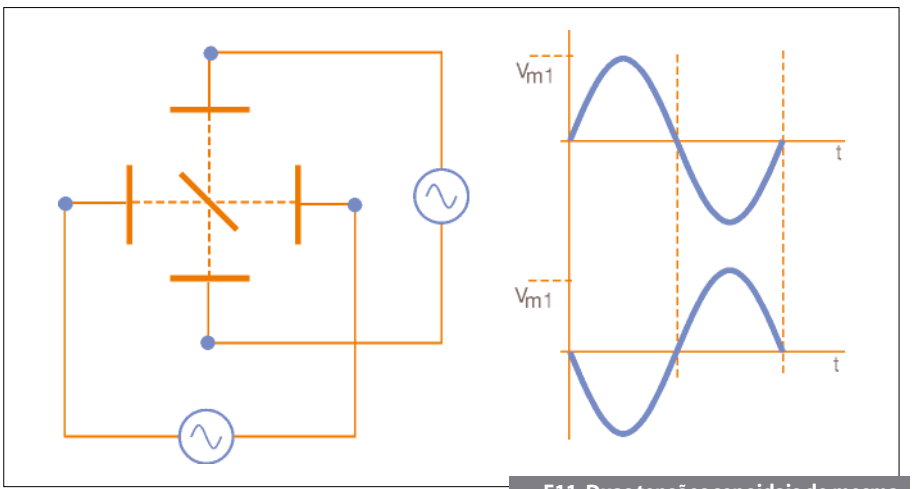
Neste caso a figura que surge na tela é um traço oblíquo da esquerda para a direita, com uma inclinação de 45° se as tensões forem da mesma amplitude. Se forem de amplitudes diferentes, a inclinação poderá tomar valores entre 0° e 90° (**figura 11**).



F9. Aplicação de duas tensões senoidais de mesma frequência e em fase.

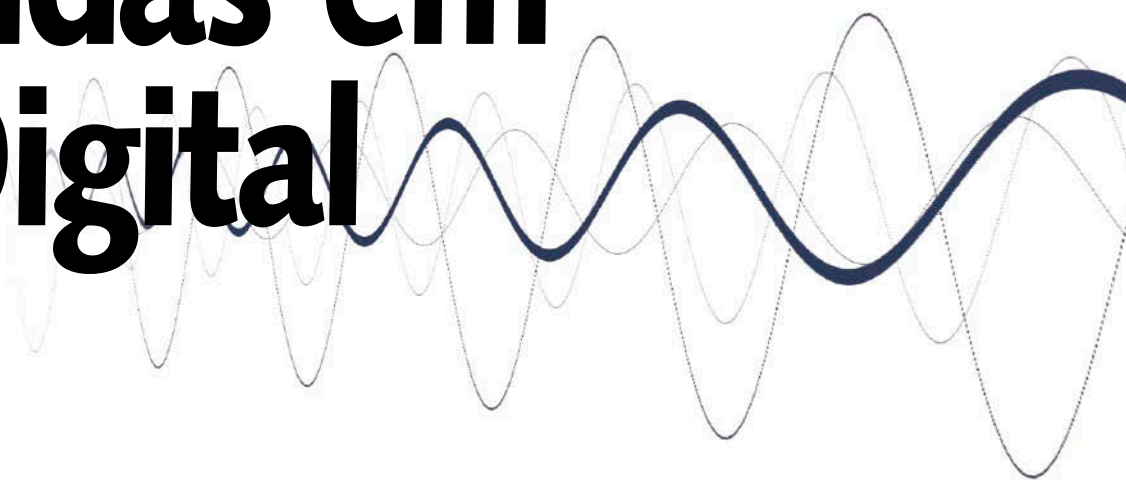


F10. Duas tensões senoidais de mesma frequência e defasadas de 1/4 ciclo.



F11. Duas tensões senoidais de mesma frequência e defasadas de 1/2 ciclo.

Medidas em RF Digital



Neste artigo, baseado em material da Agilent Technologies, trataremos um pouco das medidas feitas nos circuitos de RF digital.

Newton C. Braga

Para se garantir que um sistema de telecomunicações funcione perfeitamente, livre de interferências, ruídos, desvios de fase, temporização e muito mais, é necessário realizar a medida precisa de diversos parâmetros. Essas medidas incluem a análise do circuito modulador e do circuito demodulador, caracterização do sinal transmitido, etc.

O uso do osciloscópio nas medidas de RFdigital não é tão simples como acontece na maioria dos circuitos analógicos, tanto de alta quanto de baixa frequência. O modo como os sinais são gerados e a própria modulação exigem técnicas especiais para a medida, e a interpretação dos parâmetros deve ser feita com pleno conhecimento de seu significado.

Assim, as medidas em RF digital normalmente são divididas em quatro categorias: potência, frequência, temporização e precisão de modulação, as quais passamos a analisar a seguir.

Potência

As medidas de potência realmente estão relacionadas com a potência da portadora, o ganho dos circuitos amplificadores e as perdas introduzidas nos circuitos atenuadores.

Os circuitos usados nos sistemas modulados digitalmente têm as mesmas características dos ruídos. Eles se espalham por uma certa banda de frequências, tendo que ser feita uma integração numa certa faixa, ou ainda devem ser tomadas medidas de densidade espectral (PSD).

As medidas de potência têm geralmente a faixa normatizada em uma largura de 1 Hz. Na **figura 1** damos um exemplo de como a energia se distribui numa medida deste tipo.

Uma outra medida realizada nos sistemas digitais é a da potência no canal adjacente. Ela permite verificar o nível de interferência que afeta os usuários de canais adjacentes. Além disso, testa a quantidade de energia de um sinal de RF modulado digitalmente que se espalha pelos canais próximos.

O resultado desta medida é a relação (em dB) da medida no canal adjacente com a potência total transmitida, conforme ilustra a **figura 2**.

Para sistemas pulsantes como o TDMA, as medidas de potência possuem também uma componente de tempo que deve ser levada em conta. Os tempos de "burst" devem ser considerados quando o circuito liga e desliga. Neste caso, pode ser feita uma média da média do tempo em que os ciclos de sinal estão presentes.

Frequência

As medidas de frequência são também complexas nos sistemas digitais de comunicações. Nestes sistemas, a faixa ocupada é uma medida importante a ser feita, pois ela permite verificar se a faixa destinada à comunicação está sendo usada corretamente.

A medida da faixa ocupada (BW = *bandwidth*, ou largura de faixa) mede como o espectro de frequências é ocupado pelo sinal, observe a **figura 3**.

Esta faixa é medida em Hz e a potência envolvida nela é especificada por uma porcentagem. Por exemplo, um sinal coloca: 99%

da potência na faixa analisada. Uma forma de expressar o resultado desta medida seria "99% da potência do sinal está contida numa faixa cuja largura é 30 kHz".

As larguras de faixa encontradas nos sistemas variam bastante, dependendo de diversos fatores. Temos, por exemplo, 30 kHz para os sinais NADC/DQPSK e 350 kHz para os sinais GSM, e GSMK.. Para vídeo digital a faixa é tipicamente de 6 a 8 MHz.

Observe que a simples utilização de um frequencímetro nessas medidas não fornece precisão, dada a distribuição de um sinal por um espectro relativamente amplo.

Temporização

As medidas de temporização são feitas principalmente nos sistemas pulsados, ou por "burst". Elas incluem a determinação de parâmetros como os intervalos de repetição, ciclos ativos, tempos entre erros de bits, tempo on, tempo off, etc.

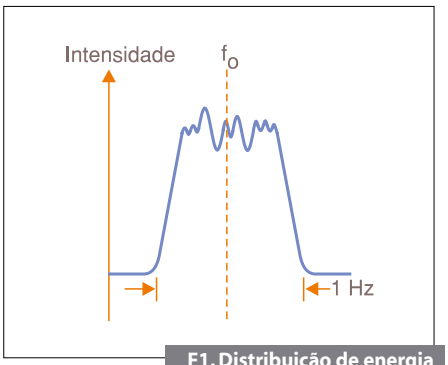
Uma medida importante é a da precisão de modulação. Ela envolve a medida de quão próximos dos estados da constelação, ou da trajetória do sinal, está o sinal real em relação ao sinal teórico. Neste caso, o sinal analisado é demodulado e comparado com uma referência. Os dois sinais são combinados, restando, então, a diferença residual que pode ser medida.

Para estas medidas, é necessário ter recursos para gerar um sinal de referência preciso. Este é muito importante para se avaliar a qualidade de um sinal, encontrando problemas no sistema. Uma vez que o sinal residual seja obtido, torna-se muito fácil visualizar pequenas discrepâncias, que podem ter sido obscurecidas pelo processo de modulação e que não seriam visualizadas de outra forma.

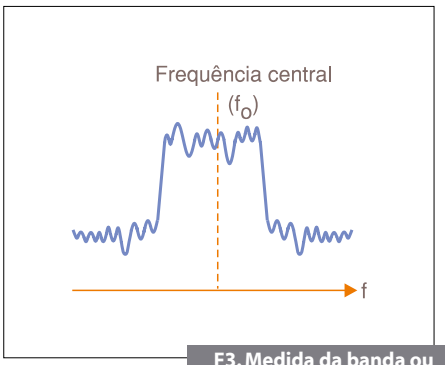
Para estas medidas, também é fundamental saber interpretar os erros de magnitude de vetor. Para entender como realizá-las, convém recordar alguns princípios da modulação vetorial.

Os bits de informação são colocados numa portadora de RF variando a intensidade e a fase do sinal portador. A cada transição do *clock*, a portadora ocupa uma única de certas posições no plano I x Q, conforme mostra a **figura 4**.

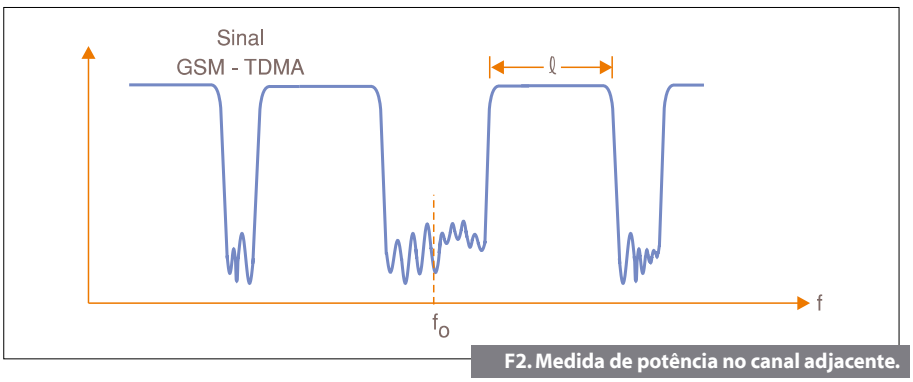
Cada locação corresponde a um conjunto específico de dados, formando assim um diagrama de constelação, conforme ilustra a **figura 5**.



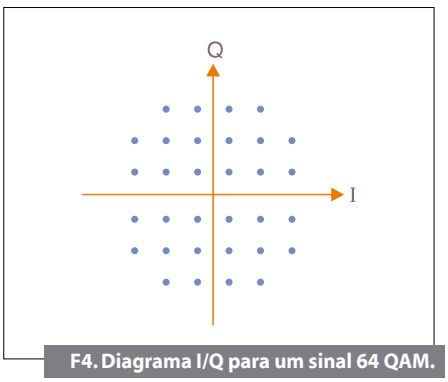
F1. Distribuição de energia em medidas de potência.



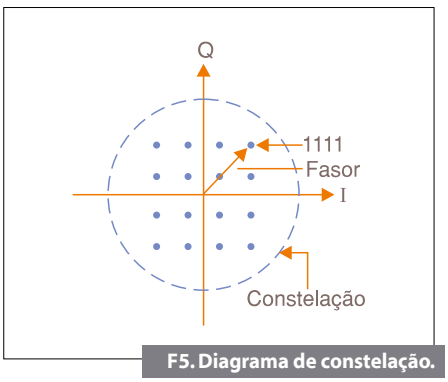
F3. Medida da banda ou largura de faixa BW.



F2. Medida de potência no canal adjacente.



F4. Diagrama I/Q para um sinal 64 QAM.



F5. Diagrama de constelação.

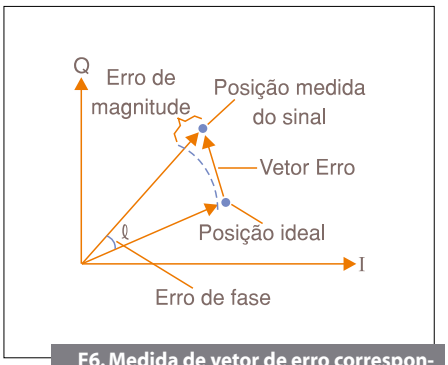
Para demodular este sinal, é preciso que o circuito demodulador consiga identificar exatamente a posição na constelação (fase e intensidade) do sinal transmitido, fornecendo assim o conjunto de bits correspondentes.

O *layout* do diagrama de constelação e as localizações ideais dos sinais dependem do formato de modulação escolhido, tais como BPSK, 16QAM, DQPSK, etc. A trajetória tomada pelo sinal ao passar de uma locação para outras depende da implementação do sistema. Nessa medida, o que se faz é determinar o fasor do sinal real correspondente a cada posição da constelação e compará-lo com o fasor do sinal ideal, observe a **figura 6**.

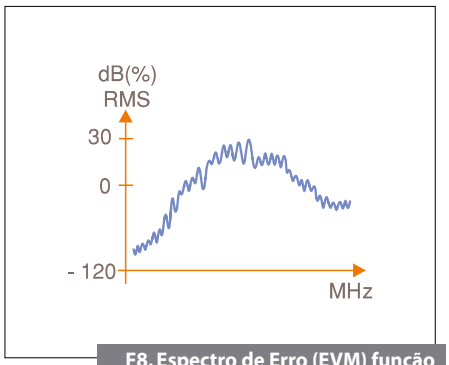
A diferença entre os dois fornece um "vetor de erro" (exemplo na mesma figura).

A medida deste vetor de erro consiste numa ferramenta poderosa de análise de problemas em sistemas de comunicações digitais. Podemos, pela análise deste vetor, determinar diversos desvios que ocorrem na transmissão de um sinal. Assim, a análise deste vetor consiste em um recurso fundamental de service que o profissional de telecomunicações deve conhecer. Por este vetor pode-se determinar que tipo de degradação está ocorrendo em um sinal no seu percurso do transmissor ao receptor.

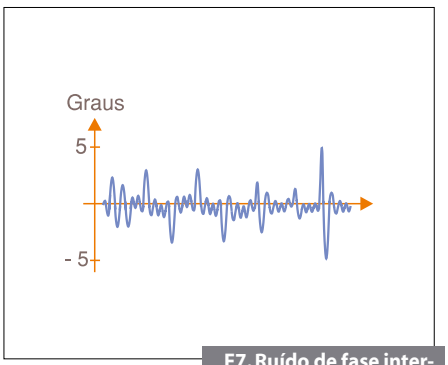
A Agilent Technologies possui um documento que ensina a usar as medidas



F6. Medida de vetor de erro corresponde a cada posição da constelação.



F8. Espectro de Erro (EVM) função da frequência (MHz).



F7. Ruído de fase interferindo no sinal.

da magnitude de erro na análise do sinal, ajudando a identificar suas origens. Trata-se do *Product Note 89400-14*.

Há diferentes mecanismos de erro que afetam os sinais de diversas formas. Esses mecanismos podem alterar o sinal apenas em intensidade, em fase ou em ambos ao mesmo tempo. Conhecendo o modo como cada erro se manifesta, pode-se chegar a determinados tipos de problemas.

Quando o erro de fase (em graus) é substancialmente maior que o erro de intensidade (em porcentagem), algum tipo de modulação de fase indesejável está acontecendo. Isso pode ser causado por ruído, problemas de espúrios ou acoplamentos cruzados, ou ainda frequências indevidas geradas em etapas do circuito.

Os Erros de fase I/Q são medidas referenciadas no tempo. Logo, quando analisados em função do tempo, eles mostram a forma de onda de qualquer sinal residual ou interferente.

A presença de sinais senoidais ou outras formas regulares de onda indicam um sinal interferente. Ruído uniforme é sinal de alguma forma de ruído de fase (*random*, *jitter*, PM/FM residual, etc).

Na **figura 7** mostramos como um ruído de fase pode aparecer no sinal.

Um sinal perfeito tem uma constelação que é simétrica em relação à origem. Um sinal deslocado desta posição indica que a constelação não é "quadrada", havendo um erro de quadratura.

Devemos, inclusive, analisar a Magnitude do Vetor de Erro em relação ao tempo.

EVM (Magnitude do Vetor de Erro) é a diferença entre o sinal de entrada e o sinal de referência interno.

Quando analisados como função do tempo, os erros podem ser relacionados com pontos específicos da forma de onda de entrada como, por exemplo, picos, passagem por zero, etc. O EVM é uma magnitude escalar e os picos indicam que compressão ou corte do sinal.

Um caso típico de não linearidade no cruzamento por zero pode ser dado tomando como exemplo um amplificador *push-pull*, quando as duas metades do sinal são manuseadas por transistores diferentes.

O Espectro de Erro (EVM) em função da frequência é calculado a partir da *Fast Fourier Transformer* (FFT, ou Transformada Rápida de Fourier) e resulta na apresentação de uma imagem no domínio da frequência com detalhes que não são visíveis no domínio do tempo, veja exemplo na **figura 8**.

Conclusão

Não basta ter a instrumentação apropriada para se analisar os sinais existentes nos circuitos de telecomunicações digitais. É preciso saber interpretá-las. Em função deste conhecimento, a realização de medidas específicas em um sistema pode revelar problemas importantes de funcionamento que, de outra forma, seriam muito mais difíceis de localizar.



Minicurso LabVIEW
Aprenda Fazendo

Parte 7

Veremos neste artigo como proceder para a coleta dos dados, análise e aplicação de algumas variáveis estatísticas

Luis Fernando F. Bernabe

Estamos chegando à reta final. A nossa proposta está parcialmente concluída. Fazendo uma breve revisão: começamos com algumas possíveis linguagens, iniciamos a operação do equipamento com a sua descrição simplificada e algumas aplicações "stand alone" como apresentação, fizemos um comparativo entre as linguagens abordadas, apresentamos o processo completo de conexão USB ao PC com os programas de interfaceamento utilizados, desenvolvemos algumas aplicações funcionais e caminhamos para uma aplicação mais profissional e amigável com o usuário final.

Depois de todas estas informações, falta pouco para chegarmos ao término desta série. Mas, como o final não é agora, nesta parte continuamos, seguindo nossos objetivos iniciais.

Agora vamos analisar como podemos proceder para coletarmos os dados da melhor forma possível, se é possível e interessante fazermos uma análise dinâmica ou estática, e ainda aprender ou rever algumas das variáveis estatísticas aplicadas a coleta de dados digitalizados.

Coletando dados

Vimos que os processos para coleta de dados foram bastante simplificados com o avanço da tecnologia e a aplicação da informática no processo e na intstrumentação, com a utilização de computadores, programas mais amigáveis, interfaces "Plug & Play" e rede. Há um grande volume de casos de sucesso neste ramo da Eletrônica. Muitos equipamentos foram criados e/ou se desenvolveram devido a estas aplicações inclusive.

A eletrônica neste campo se iniciou com experimentos científicos e militares aplicados em desenvolvimentos de armamentos, sistemas de defesa, RADAR e SONAR desde antes da 2ª Guerra Mundial, e foi mantida durante a "Guerra Fria" e nas missões espaciais a partir dos projetos Apolo e Saturno, até os ônibus espaciais e os atuais satélites para GPS e missões para Marte e outros planetas.

Se hoje temos mais facilidade de coleta de dados por conta destes sistemas automatizados, devemos nos lembrar sempre de qual variável física estamos medindo, de seus problemas específicos inerentes à

medição e termos um cuidado especial com o volume de dados coletados desnecessariamente, que terão de ser descartados depois, em outro processo, consumindo mais tempo na análise destes dados. Antes tínhamos um sistema dedicado a cada tipo de sensor, hoje todos podem ser conectados num mesmo sistema. Temos que ter cuidado para não nos enganarmos quanto aos sensores, seus tipos, faixa de trabalho, deriva térmica, precisão, reprodutibilidade, etc.

Este tema é muito vasto além de interessante, para limitarmos a quantidade de desvios no mesmo, vamos retomar a nossa meta experimental: "coletar dados suficientes para a calibração de um sensor termopar e comparando-o a um segundo, de referência e previamente calibrado".

A filosofia de calibração

Para executarmos este projeto vimos o equipamento, a interface e os programas; agora vamos aos detalhes externos ao conjunto e que são fundamentais para a sua realização prática.

Quando desejamos efetuar uma medida comparativa entre sensores de temperatura, ambos os sensores devem estar nas mesmas condições exatas de variação térmica, ter os mesmos volumes de sensor e estarem no mesmo ambiente, com tempo suficiente para que se estabilizem as suas leituras. Idealmente teríamos que ter um ambiente com temperatura controlada, com o menor volume possível, para termos condições de variação rapidamente estabilizadas mesmo depois de alterarmos a sua temperatura.

Para a calibração de sensores de alta precisão (faixa especial de 1,1 °C) e que trabalham com grandes variações de temperatura, são necessários ensaios repetidos em condições dinâmicas de trabalho, gerando algumas rampas de aquecimento/resfriamento, medindo características importantes de inércia térmica, tempo de resposta, comparação ponto a ponto, precisão, reprodutibilidade, estabilidade de medição, etc. Considerando, claro, a sua aplicação, só para citar extremos em aplicações comerciais: forno industrial de fundição, câmara térmica de laboratório e aviônica.

Como estamos em uma apresentação didática, a filosofia de medição é "um pouco" diferente. Podemos fazer as medições em degraus estáticos de temperatura, ao invés de um ensaio dinâmico; o que simplifica bastante. Com dois pontos nas extremidades da faixa podemos obter uma precisão razoável o suficiente, sem precisarmos de uma câmara térmica especial ou um banho seco de calibração de termopares.

Para esta tarefa temos que pensar em quais pontos de temperatura seriam interessantes para a calibração dos sensores. É desnecessário calibrar um sensor numa faixa de utilização na qual o mesmo nunca seja utilizado. Para que serve a calibração de um termopar tipo J até o seu limite de 780 °C sendo que este trabalha numa cuba de pré-aquecimento de vapor, e que a temperatura máxima deste equipamento é de 150 °C?

Dependendo do seu fornecedor, talvez o processo de calibração fosse mais simplificado para calibrar este termopar na faixa de 0 °C a 200 °C. Será que sendo mais simples, também seria mais barato e rápido com tempo de entrega menor? Dependendo do caso, sim. Se o processo for altamente automatizado, com banho seco controlado por um programa que também fizesse a aquisição de temperatura, a resposta poderia ser não.

Aqui, como estamos avaliando os procedimentos de automação didaticamente, não temos a rigidez de calibrarmos os termopares em um processo homologado. Com esta liberdade, podemos então colocar os sensores imersos em água com gelo (0 °C) e em água fervente (100 °C), como alguns de nós fizemos nas aulas de ciências. Além do que, temos que considerar um sistema que possa ser simplificado ao máximo.

Obs.: Lembramos que teremos que efetuar uma correção de valores obtidos de temperatura com a pressão atmosférica em função da altitude em que estamos.

Desta forma, para executarmos esta calibração como apresentamos aqui nesta série, faremos a aquisição de um baixo volume de dados, o suficiente para realizarmos uma média, melhorando a precisão das medições realizadas.

Há alguns modos interessantes de efetuarmos as medidas deste ensaio. O primeiro e mais antigo é o processo estático. Com este método definimos as temperaturas de interesse e com uma estufa/geladeira controlada por um termostato (por ex.) estabelecemos a temperatura mais baixa e a medimos com os dois sensores, comparando as duas leituras.

Se o termostato da câmara térmica for do tipo mecânico/eletrônico (liga/desliga), há a necessidade de termos que esperar a temperatura se estabilizar por um tempo mais longo devido a histerese. A histerese ocorre sempre que um dado circuito ou componente muda de estado ativo para inativo, e vice-versa.

As câmaras térmicas mais precisas possuem controladores duplos simultâneos, de resfriamento e aquecimento e um sistema microcontrolado embarcado, com controle PID (Proporcional, Integral e Derivativo) que coloca neste ambiente a dose certa de "frio" e calor, para manter a temperatura dentro da precisão especificada pelo fabricante.

A histerese é a diferença máxima de valor de medida apresentada por um instrumento qualquer, dentro da sua faixa de trabalho, quando efetuada a medida em duas condições diferentes, crescente e decrescente.

Box 1

Vale a pena procurar algumas bibliografias e outras referências sobre o assunto Matrizes:

James Joseph Sylvester
http://pt.wikipedia.org/wiki/James_Joseph_Sylvester

Arthur Cayley
www.ime.usp.br/~leo/imatica/historia/cayley.html

www.portalsaofrancisco.com.br/alfa/arthur-cayley/arthur-cayley.php

Joseph-Luis Lagrange
http://pt.wikipedia.org/wiki/Joseph-Louis_Lagrange

www.iem.unifei.edu.br/juliano/Download/EME33/Historia/Lagrange/texto_Lagrange.pdf

Teoria das Matrizes
www.geometriadinamica.kit.net/Microsoft%20Word%20-%20AlgebraLinear.pdf

www.lia.ufc.br/~mcampelo/notas-matrizes.pdf

Resumindo, vamos considerar um voltímetro qualquer, ajustamos o fundo de escala para 200 Vdc, conectamos o voltímetro a uma fonte de até 300 Vdc. Com este arranjo, variamos a tensão de 0 Vdc até 200 Vdc em passos de 10 Vdc, de modo **crescente**. Observamos a leitura do voltímetro e comparamos com a da fonte. Quando medirmos 200 Vdc, refazemos as medidas novamente em variações **decrescentes**, iniciando em 200 Vdc indo até 0 Vdc. A histerese é a diferença destas medidas de tensão (no caso) nestas duas situações possíveis. A histerese varia com a grandeza, normalmente é menor no fim da escala, por isso é montado um gráfico de histerese dos instrumentos e dos sensores.

Sobre a histerese em instrumentação o "Curso de Formação de Operadores de Refinaria", na área de "Instrumentação Básica" está disponível gratuitamente na internet pela Petrobras. Quem quiser aprender sobre instrumentação, este é um material excelente. Se o leitor tiver mais curiosidade, procure "Curso de Formação de Operadores de Refinaria" (com aspas mesmo) no Google. **www.dequi.eel.usp.br/~felix/Instcont.pdf**

Faltou um "detalhe"

Ao escrever artigos em sequências longas, mesmo que tenham a justificativa de serem mais didáticos ou de abordarem temas mais complexos, há grande possibilidade do escritor se perder, de "fugir" do tema central e/ou serem esquecidos pontos importantes que complementam os objetivos. Por isso, atenção e planejamento são fundamentais.

Cometi um deslize, e me esqueci de abordar um tópico fundamental dos nossos artigos, o assunto é **Matriz,** tanto na breve revisão matemática quanto no próprio LabVIEW**.** Peço que me desculpem pela falta deste tópico nos artigos. Faço um parêntese na sequência abaixo para depois retornar ao tema como deveria ser abordado anteriormente.

As matrizes

As matrizes sempre foram assuntos "chatos" tanto de serem ensinados como de serem aprendidos em nossa matemática de segundo e de terceiro graus. Acredito que seja por falta de aplicações práticas que envolvam e despertem o interesse dos alunos. O mais curioso é que atualmente todas as ciências exatas têm uma forte utilização desta "organização matemática" para a solução de problemas, isso se deve principalmente a informatização. Isso exige que os profissionais destas áreas devam ter necessariamente uma boa habilidade com as matrizes.

Comecemos por uma revisão simples do básico: Como surgiram as matrizes?

Bem, no seu início as matrizes foram utilizadas apenas como um modo de organizar os dados numéricos para serem melhor vistos em conjunto, era o que conhecemos hoje por tabela ou planilha.

Muito recentemente, considerando a sua história, há pouco mais de 150 anos, as matrizes foram reconhecidas como tema importante da Matemática. Por volta desta época se tornou um **método** para a organização de **números de uma equação que encontrariam a sua solução** através de seu **determinante** !!!

O primeiro matemático a utilizar este método de solução parece ter sido Augustin-Louis Cauchy, 1826. Mas, o nome matriz como conhecido hoje, só veio com James Joseph Sylvester, 1850. Seu amigo Arthur Cayley, com o seu livro "Memoir on the Theory of Matrices" (1858), divulgou esse nome e demonstrou algumas de suas utilidade práticas.

James Joseph Sylvester deu o nome de matriz pelo seu significado coloquial: local onde algo se gera ou cria. Antes, o arranjo numérico da matriz era denominado "...um bloco retangular de termos..." de onde se originavam vários determinantes.

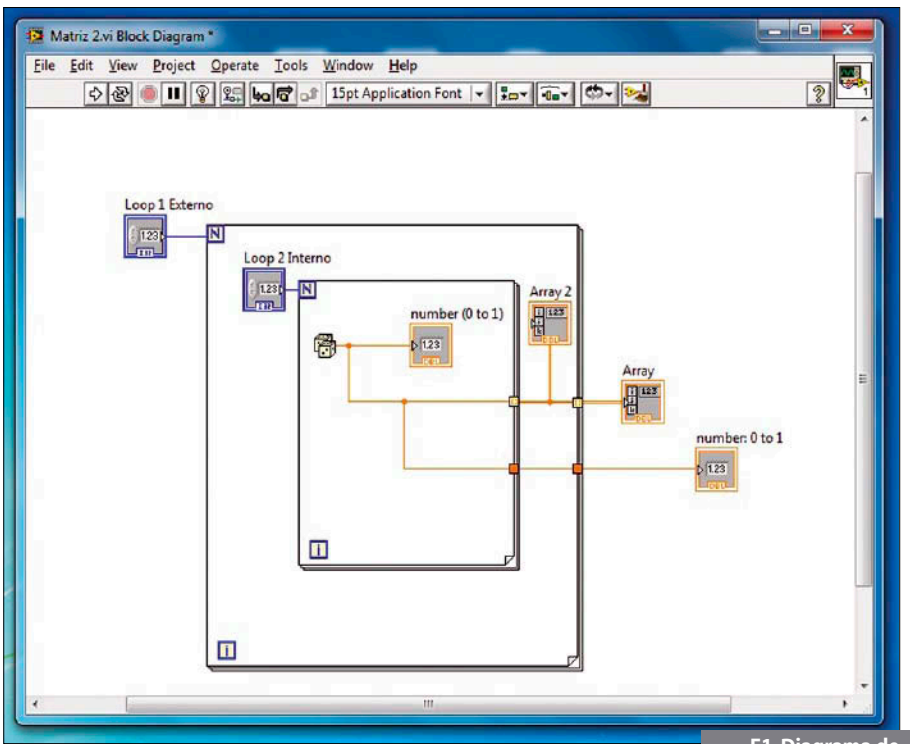
Apesar de ter criado o nome matriz, Sylvester a enxergava apenas como parte dos determinantes e, somente com seu amigo Cayley, as matrizes passaram a ter vida própria.

As matrizes nasceram da necessidade dos matemáticos dos séculos XVIII e XIX, quando começaram a investigar a Teoria das Formas Quadráticas. Hoje, consideramos imprescindível estudar essas formas através da notação e metodologia matricial, como vetores (x,y,z), mas naquela época elas eram tratadas numericamente, como escalares.

Na **equação 1**, a seguir, apresentamos uma equação de duas variáveis representada na sua **forma quadrática**, tanto via **notação escalar** como com a **notação matricial**:

$$q(x,y) = ax^2 + 2bxy + by^2 = [\,xy\,] * \begin{bmatrix} ab \\ bc \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix}$$

A primeira utilização prática do conceito de matriz ocorreu quando Lagrange c. 1790 reduziu a caracterização dos máximos e mínimos de uma função real de várias variáveis, ao estudo do sinal da forma



F1. Diagrama de Blocos completo.

quadrática associada à matriz das segundas derivadas dessa função.

Assim, podemos dizer que a **Teoria das Matrizes** teve como mãe a **Teoria das Formas Quadráticas**, pois seus métodos e resultados básicos foram gerados lá. Hoje, contudo, o estudo das formas quadráticas é um mero capítulo da Teoria das Matrizes. Fonte: **www.mat.ufrgs.br/~portosil/passa3b.html**

Construindo matrizes em LabVIEW

Por que vamos utilizar matrizes? Na verdade, vamos utilizar as matrizes na sua função mais simples, a tabela, uma lista de temperaturas relacionada com valores de tempos da medida. Com se tratam de **duas grandezas físicas diferentes,** não existe a possibilidade de trabalharmos com determinantes (o resultado da matriz) e termos um resultado válido. A não ser que fossem linearizados os dados das medições e que criássemos duas equações de cada gráfico resultante, para compararmos a medição padrão com a do outro sensor em calibração.

Box 2

"*Internet e Formação de Professores de Matemática: Desafios e Possibilidades*"

www.athena.biblioteca.unesp.br/exlibris/bd/brc/33004137031P7/2005/garcia_tmr_me_rcla.pdf **(underscore)**

"*A História da Matemática, o Porquê e o Como***"**

www.uff.br/historiadamatematica/siteantigo/Arquivos/Historia_da_Matematica_por_que_e_como_Andre_Weil.pdf **(underscore)**

A História da Matemática :

www.ime.usp.br/~leo/imatica/historia/

http://educar.sc.usp.br/licenciatura/2003/hm/page01.htm

http://pt.wikipedia.org/wiki/História_da_matemática

www.mat.uc.pt/~jaimecs/indexhm.html

www.infoescola.com/matematica/historia-da-matematica/

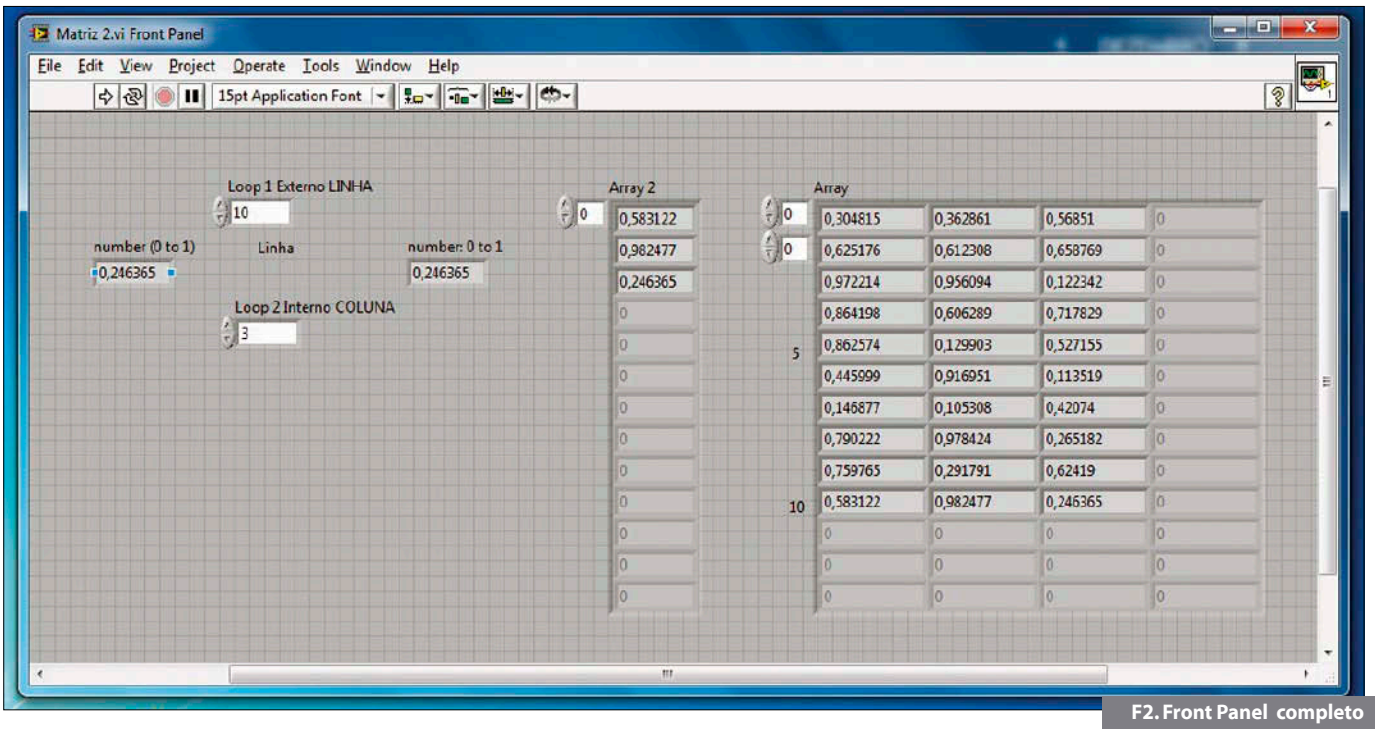
http://usuarios.upf.br/~pasqualotti/hiperdoc/matematica.htm

http://pt.wikipedia.org/wiki/Cronologia_da_história_da_matemática **(underscore)**

http://pt.wikipedia.org/wiki/História_dos_números_longos **(underscore)**

Todo o **Conhecimento Humano** perde muito do seu estímulo de conquistar os alunos e evoluir também, pelo fato de que seus professores se esquecem da **Sua História** e os seus mais importantes colaboradores.





F2. Front Panel completo

Aqui vamos utilizar a matriz como tabela, isso significa que construiremos a matriz, mas o seu tratamento matemático é de tabelar os números numa sequência de interesse. No **LabVIEW** as matrizes e as tabelas têm tratamentos diferentes. Os dados de uma matriz podem ser numéricos, em **string**, hexadecimal, binário, etc. Já na tabela, o foco é apresentação de dados, todo o conteúdo tem que estar exclusivamente em **strings,** no LabVIEW há uma biblioteca de conversão de formatos de dados e utilizamos apenas uma VI para resolver este caso. Há aplicações em que temos que montar uma matriz e numa próxima etapa, convertê-la para tabela e incluí-la no gerador de relatórios, por exemplo.

Como todo bom programador sabe, para se construir uma matriz precisamos dos dados a serem inseridos em sequência de formação em matriz. Para fazermos isso utilizamos os **Loop For** tantas vezes quantas forem as dimensões da matriz.

Comecemos, então. Abra o **LabVIEW** com uma VI em branco. Com o botão direito do mouse, clique na área branca do **Block Diagram**, com a paleta de **Functions** procure nas estruturas de loop, o **Loop For**, **(Functions > Programming > Structures),** desenhe o loop no diagrama de blocos em tamanho médio, em seguida, repita o processo e desenhe o segundo **Loop For** dentro do primeiro, dê um espaço suficiente para reproduzir a **figura 1**. Aproxime o cursor do mouse do lado da variável **N,** do loop. Com o botão direito, clique neste ícone e dentro da janela, escolha o botão **Create Control** aparecendo um botão de controle nos dois campos, **Block Diagram** e **Front Panel**.

Coloque dentro dos dois loops uma VI **Random Number (Functions > Programming > Numeric).** Esta VI tem o formato de dois dados juntos. Com a VI dentro dos loops, aproxime o cursor dela e automaticamente aparecerá um carretel de "linha" de conexão. Clique com o botão direito, um **Indicator**. Este é o **number (0 to 1)**. Clique novamente neste terminal da VI, arraste o fio de ligação até a borda do loop e clique nesta borda novamente. A função inicial deste **Tunnel** (saída do loop) é de **Auto-Indexed**, dentro do ícone aparecem os colchetes **[ ]** indicando que a saída está automaticamente indexada no formato matriz. Aproxime o cursor, clique no botão esquerdo do mouse, e arraste o fio de ligação até a borda do outro loop e clique nela, como feito na anterior.

Para construir as matrizes, o modo mais simples é aproximar o cursor do ícone do **Tunnel** e clicar com o botão direito e selecionar o botão **Indicator**. Neste ponto foi construído a **Array**. Selecione o **Label** para que fique visível, se preferir.

A matriz 2 (**Array 2**) é construída pelo mesmo processo. Esta matriz é de uma coluna, pois é a saída do **FOR LOOP** interno.

Para vermos os dados em sequência, no caso conseguiremos visualizar somente o último, refaça as mesmas ligações dos fios que foram feitas para as matrizes, até o último loop. Selecione o tipo de **Tunnel** permitindo ou não a indexação, neste caso, a função de indexação está desativada.

No último loop, o externo, aproxime o cursor e crie um **Indicator** novamente, do mesmo modo que o anterior.

Veja, na **figura 2**, o Front Panel completo.

Observe os controles de execução dos loops, Loop 1 Externo LINHA e Loop 2 Interno LINHA, estes botões criam as dimensões das matriz na saída. Selecione outros valores de Linha e Coluna.

O indicador **number (0 to 1)** apresenta o último valor do sorteio dos dados. Os dados são iguais aos do indicador **number: 0 to 1** que é a saída dos dois loop, mas sem a indexação.

Verifique que os controles **number (0 to 1)**, **number: 0 to 1, Array 2 e Array** apresentam os mesmos valores no display e na última posição, no caso das matrizes.

Assim, caros leitores, ficamos por aqui, e na próxima edição continuaremos passo a passo até atingirmos nossa meta.

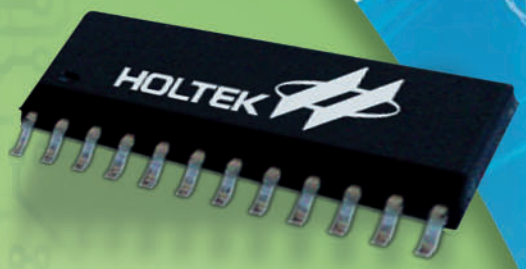
microcontroladores
FLASH
de Baixo Custo

HOLTEK

Holtek lança nova série

HT66Fxx
HT68Fxx

Microcontroladores Enhanced I/O e A/D Flash 8-Bit



Características:

- Arquitetura RISC – Ótimo Desempenho;
- Memória de Programa Flash de 1 Kbyte a 12 Kbytes;
- Memória de Dados RAM de 64 bytes a 576 bytes;
- EEPROM de 32 bytes a 256 bytes;
- Disponibilidade de 18 a 50 portas I/Os;
- Encapsulamentos disponíveis: 16DIP/NSOP/SSOP, 20DIP/SOP/SSOP, 24SKDIP/SOP/SSOP, 28SKDIP/SOP/SSOP, 44QFP, 40/48QFN;
- Interface SPI/I^2C, UART, USB
- A/D de 12 bit para HT66Fxx
- Excelente para aplicações em áreas de eletroeletrônicos, segurança, automação industrial, automotiva, produtos de consumo, etc.

HOLTEK
ANYTEK conectores
FORDATA FOR DATA DISPLAY
módulos RF WENSHING
GlobalSat Wired by Wireless
APLUS CI reprodutor de sons
FAGOR
SIMCom A company of SIM Tech

Visite o site da Holtek e conheça toda linha:
www.holtek.com

Cika Eletrônica
Seu Distribuidor de Componentes Eletrônicos

Tel.: 55 11 2693-6428 / Fax: 55 11 2693-8805
vendas@cikaeletronica.com
www.cikaeletronica.com

Sistemas de Supervisão e Aquisição de Dados

Sistemas de aquisição de dados e controle de dispositivos vêm sendo desenvolvidos para diferentes áreas de atuação, tanto industriais como científicas. O seu objetivo é apresentar ao observador, os valores das variáveis ou parâmetros que estão sendo medidos.

César Cassiolato

Nos últimos anos tem-se visto um crescente desenvolvimento de sistemas de aquisição e tratamento digital de sinais, sendo que vários fatores têm contribuído para a evolução nesta área:

- O avanço da microeletrônica que tem possibilitado o aumento das capacidades e velocidades dos DSPs, elemento central de uma aplicação de tratamento digital de sinal;
- A crescente performance dos computadores pessoais, tal como a sua relação qualidade/preço e a sua confiabilidade;
- A existência de cada vez mais e melhores ferramentas de desenvolvimento de software, que permitem criar aplicações de alto nível com avançadas interfaces gráficas;
- O desenvolvimento de novas tecnologias de comunicação que permitem o controle remoto de instrumentos, usando a internet e o *wireless* como veículos de transmissão de dados.

Por que uma empresa precisa de um sistema de aquisição e supervisão? Simples…para proporcionar níveis melhores de qualidade, redução dos custos operacionais, maior desempenho de produção e, fundamentalmente, para facilitar a excelência operacional.Vale citar um pouco da história dos avanços tecnológicos e que culminaram com "n" benefícios em desenvolvimento de sistemas de aquisição, controle e automação.

Um pouco de história

Os primeiros sistemas de automação foram desenvolvidos no final do século XIX durante a revolução industrial.O trabalho que era manual passou a ser realizado por máquinas dedicadas e customizadas a uma determinada tarefa, visando cada vez mais o aumento da produtividade e eficiência. As funções de controle eram implementadas através de dispositivos mecânicos que automatizavam algumas tarefas críticas e repetitivas. Estes dispositivos eram desenvolvidos para cada tarefa e devido à natureza mecânica dos mesmos, tinham vida útil reduzida e alta manutenção.

Posteriormente, com o advento dos relés e contatores, estes dispositivos foram substituídos e apareceram dispositivos automáticos em linhas de montagens, dando um grande passo na época. A lógica a relés viabilizou o desenvolvimento de funções de controle mais complexas e sofisticadas.

Após a Segunda Guerra Mundial, houve um avanço tecnológico e apareceram as máquinas por comando numérico e os sistemas de controle na indústria de processo, assim como o conceito de referência de tensão para instrumentação analógica. Surgem os primeiros circuitos integrados, os CIs, que proporcionaram o desenvolvimento de uma nova geração de sistemas de automação. Vale lembrar que em 1947, Willian Shockley, John Barden e Walter Brattain inventaram o transistor, que é um componente eletrônico amplamente utilizado nos processadores modernos, de forma integrada.

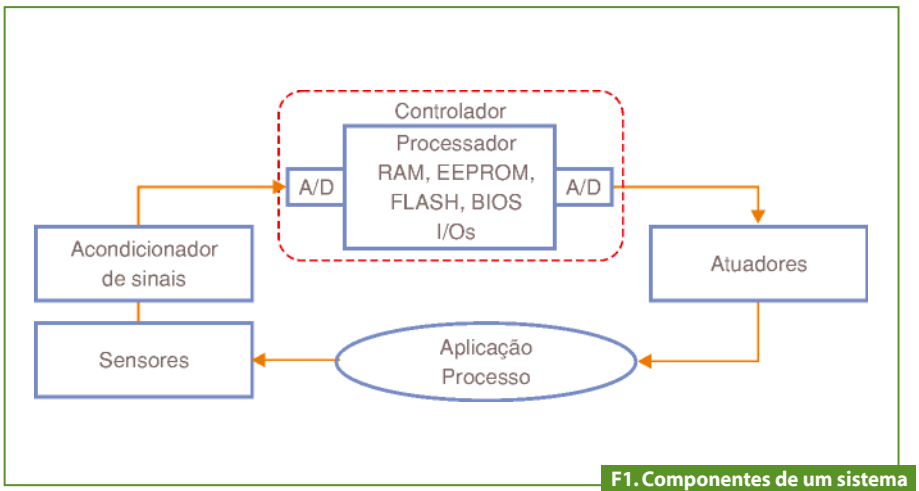
No início dos anos 70, os primeiros computadores comerciais começaram a ser utilizados como controladores em sistemas de automação de grande porte, porém estes computadores eram grandes, ocupando muito espaço, de alto custo, difíceis de programar e muito sensíveis ao ambiente industrial. Mas, tinham a vantagem de manipular a aquisição e controle de várias variáveis.

Ainda na década de 70 tivemos um grande avanço em termos de automação. A partir de uma demanda existente na indústria automobilística norte-americana, foi desenvolvido o *Programmable Logic Controller* (PLC), ou Controlador Lógico Programável (CLP). O CLP é um computador dedicado e projetado para trabalhar no ambiente industrial, onde sensores e atuadores são conectados a cartões de entradas e saídas. Os primeiros CLPs tinham um conjunto de instruções reduzido: normalmente somente condições lógicas, e não possuíam entradas analógicas, podendo manipular apenas aplicações de controle discreto. Os CLPs substituíram os painéis de controle com relés, diminuindo assim o alto consumo de energia, a difícil manutenção e modificação de comandos e também as onerosas alterações de fiação.

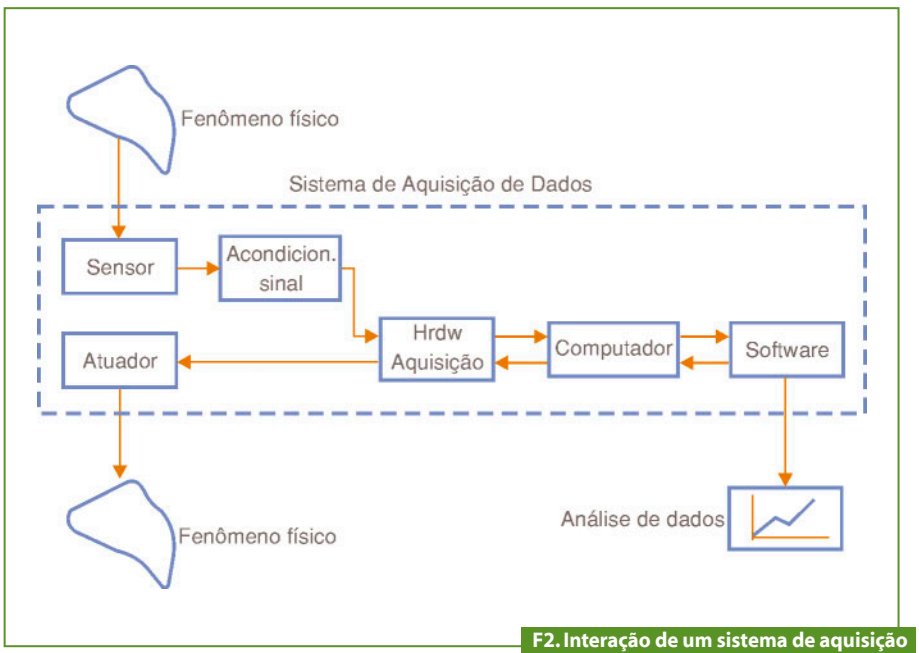
Atualmente, devido à demanda das plantas industriais, os CLPs manipulam tanto controle discreto quanto malhas analógicas. Estes sistemas são usualmente chamados de Controladores Programáveis, por não serem limitados a operações com condições lógicas. As atuais funções de controle existentes em uma planta industrial são em geral distribuídas entre um número de controladores programáveis, os quais são montados próximos aos equipamentos a serem controlados. Os diferentes controladores são normalmente conectados via rede local a um computador supervisório central, o qual gerencia os alarmes, receitas e relatórios.

Entramos em uma fase onde a tecnologia e conectividade industrial eram proprietárias e um "casamento" entre cliente e fornecedor acontecia. No mercado apareceram os SDCSs (Sistemas Digitais de Controle Distribuídos).

Na década de 90, o mundo começou a presenciar enormes avanços na área tecnológica em que os circuitos eletrônicos passaram a proporcionar maior eficiência, maiores velocidades, mais funcionalidades, maiores MTBFs (*Mean Time Between Failures*, maior confiabilidade), consumos menores, espaços físicos menores e ainda, com reduções de custos. Ao mesmo tempo em que impulsionou o desenvolvimento de computadores, interfaces e periféricos mais poderosos, com alta capacidade de processamento e memória e o mais interessante, dando vazão a alta escala de produção com custos reduzidos, foi uma vantagem de forma geral, pois aumentou a oferta de microcontroladores, CIs e ASCIs para toda a indústria.



F1. Componentes de um sistema de aquisição de dados.



F2. Interação de um sistema de aquisição de dados com os fenômenos físicos.

E se não bastasse esta revolução eletrônica, os sistemas mecânicos também passaram e vêm passando por inovações e modificações conceituais com a incorporação da capacidade de processamento, tornando-os mais rápidos, eficientes e confiáveis, com custos de implementação cada vez menores. Ao longo dos últimos anos é cada vez mais frequente a utilização de componentes eletrônicos para acionamento e controle de sistemas mecânicos.

Um grande desenvolvimento verificou-se na interface gráfica com o advento de ferramentas baseadas no Windows. A simplicidade da operação juntou-se à crescente capacidade de processamento, estabelecendo a união entre os computadores e a instrumentação. Tornou-se habitual os instrumentos serem embutidos em computadores de aplicação geral, permitindo medidas diversas e manipulações complexas, e ainda grandes capacidades de armazenamento em memória ou disco, monitorização inteligente, apresentação gráfica de fácil compreensão e controle dos processos avançados.

Não resta dúvida que, hoje, não é somente a condição de controle que importa.

A gestão da informação, a inteligência da instrumentação, a tecnologia verdadeiramente aberta e não proprietária, e os benefícios da tecnologia digital são os fatores que agregam valores ao usuário.

Objetivo de um sistema de aquisição de dados

Os sistemas de aquisição têm por finalidade obter dados digitais através de medições de grandezas físicas (analógicas), tais como temperatura, pressão,densidade, pH, umidade, posição etc, e vêm sendo incorporados por diversas vertentes, tanto industriais quanto científicas.

Estas grandezas são "sentidas" por sensores ou transdutores, e são convertidas em quantidades elétricas por acondicionadores de sinais que levam os sinais até hardwares dedicados e os transformam em valores digitais. Estes valores são processados por controladores e segundo lógicas de controles devolvem a resposta processada aos atuadores (elementos finais de controle), como se vê na **figura 1**. Em sua grande parte, os valores medidos não são iguais ao da variável real, ou seja, o sistema de aquisição tem como entrada o valor real da variável e sua saída é o valor medido.

Os sensores transformam um fenômeno físico em sinais elétricos que, uma vez detectados/produzidos, variam de acordo com os parâmetros físicos que estão sendo monitorados, e devem ser acondicionados para fornecer sinais apropriados ao hardware de aquisição de dados. Os circuitos e ou elementos de acondicionamento de sinais amplificam, isolam, filtram e excitam sinais para que estes sejam apropriados aos hardwares de aquisição. Uma vez acondicionados e trabalhados na forma desejada, os sinais podem ser lidos em controladores ou mesmo computadores, com placas específicas instaladas, e armazenados em diferentes formas, como arquivos de texto, excel, banco de dados etc. **Figura 2**.

Tipo de endereçamento em Sistemas de Aquisição de Dados

Os dispositivos responsáveis por manipular os dados utilizam endereços que podem ser de memória ou de registradores, dependendo do tipo de dispositivo usado.

Assim, tem-se entrada e saída mapeada em memória (*Memory-Mapped IO*) ou entrada e saída mapeada em espaço de entrada e saída (*IO Mapped IO*).

Sistemas de Supervisão e Aquisição de Dados

Abreviadamente SCADA (proveniente do seu nome em inglês *Supervisory Control and Data Acquisition*), são sistemas que utilizam software para monitorar e supervisionar as variáveis e os dispositivos de sistemas de controle. São conectados através de *drivers* específicos. Estes sistemas podem assumir topologia simples, cliente-servidor ou múltiplos servidores-clientes.

De forma genérica, um sistema de supervisão é uma ferramenta de software que permite monitorar e controlar partes ou todo um processo industrial. Têm dois módulos básicos: o desenvolvedor e o executável ("run-time").

A maioria dos passos de programação é automatizada, suprindo a maior parte das necessidades de um projeto sem a real necessidade de conhecimento de linguagem de programação. Em casos mais complexos e específicos, onde os passos não estão automatizados, algumas ferramentas incorporam módulos de programação em VBA (*Visual Basic For Applications*) ou VBS (*Visual Basic Script*).

Data Mining

Prospecção de dados ou mineração de dados (também conhecida como *data mining*) é o processo de explorar grandes quantidades de dados à procura de padrões consistentes, com regras de associação ou sequências temporais, para detectar relacionamentos sistemáticos entre variáveis, detectando assim novos subconjuntos de dados.

Esse é um tópico recente em ciência da computação mas que utiliza várias técnicas da estatística, recuperação de informação, inteligência artificial, sistemas de aquisição de dados e reconhecimento de padrões. A mineração de dados é formada por um conjunto de ferramentas e técnicas que através do uso de algoritmos de aprendizagem ou classificação baseados em redes neurais e estatística, são capazes de explorar um conjunto de dados, extraindo ou ajudando a evidenciar padrões nestes dados e auxiliando na descoberta de conhecimento. Esse conhecimento pode ser apresentado por essas ferramentas de diversas formas: agrupamentos, hipóteses, regras, árvores de decisão, grafos, ou dendrogramas.

Um passo fundamental para um processo de data mining bem sucedido é ter um sistema de aquisição de dados que possa gerar um banco de dados, relatórios, logs, etc. Disto nascem os repositórios organizados (*Data Marts* e *Data Warehouses*).

A ferramenta de aquisição de dados

Este tipo de ferramenta exige a transferência rápida de dados entre dispositivos, software de processamento e de aplicação com interface gráfica avançada, sensores e controladores de elevada precisão. Com a velocidade dos avanços tecnológicos, cada vez mais equipamentos e instrumentação vêm com hardwares poderosos. Na vertente do software, as tecnologias vão-se tornando cada vez mais normalizadas, permitindo um desenvolvimento mais rápido e eficaz dos produtos.

O campo de estudos da instrumentação virtual ainda está a dar os primeiros passos. Nos próximos anos serão desenvolvidos vários equipamentos avançados em recursividades, os chamados instrumentos virtuais. Serão os blocos para a construção da nova geração de instrumentação e medida. As ferramentas devem ser simples e de fácil uso pelo usuário. Devem permitir o acesso a toda a funcionalidade proporcionada pela aplicação, maximizando a funcionalidade e mantendo a simplicidade da interface.

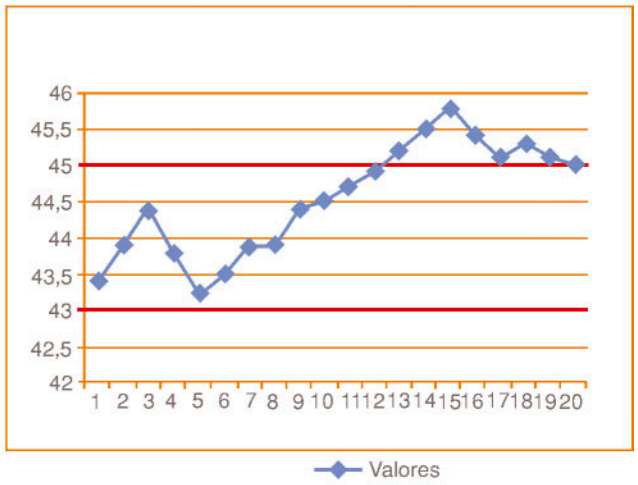
Conclusão

A aplicação da computação para automatização de diversos processos industriais tem crescido de forma significativa, e mais certo são os benefícios advindos da introdução da computação no processo produtivo que tem incrementado a produtividade e reduzido as possibilidades de falhas durante o processo.

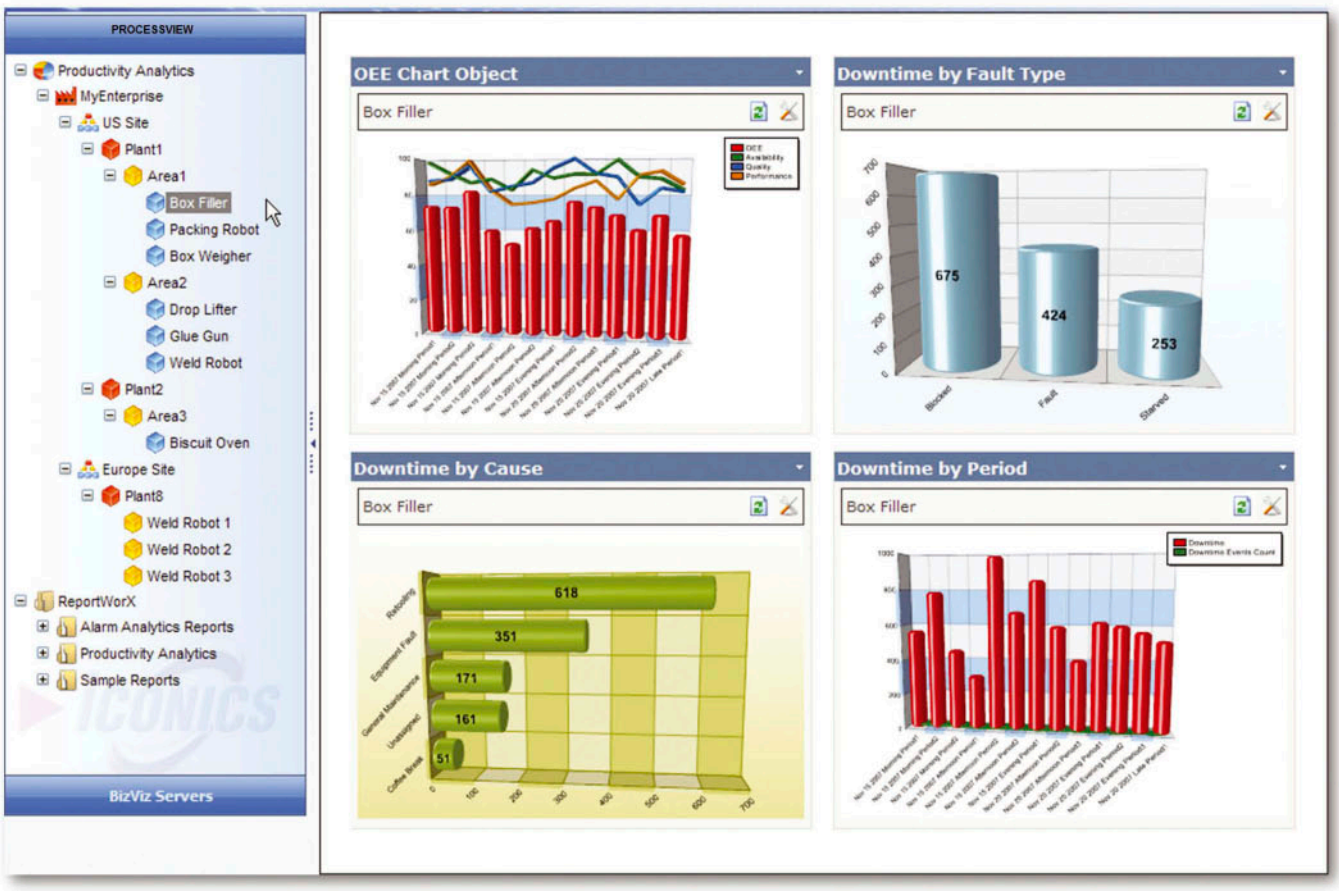
Vimos neste artigo, de forma breve que uma das aplicações que se destacam é a utilização de sistemas de aquisição de dados e supervisão para monitorar os mais diversos sensores que integram um determinado processo e a partir das informações obtidas enviar sinais de controle para outros dispositivos, ou gerar relatórios técnicos para posterior análise.

Usuários vem colhendo benefícios com sistemas avançados de aquisição de dados e supervisão. Essa mudança é encarada como um processo natural demandado pelos novos requisitos de qualidade, confiabilidade e segurança do mercado. A sua utilização traz uma vantagem competitiva, no sentido que essa nova tecnologia traz aumentos de produtividade pela redução das variabilidades dos processos e redução dos tempos de indisponibilidade das malhas de controle.

Amostras	Valores
1	43,4
2	43,9
3	44,4
4	43,8
5	43,2
6	43,5
7	43,9
8	43,9
9	44,4
10	44,5
11	44,7
12	44,9
13	45,2
14	45,5
15	45,8
16	45,4
17	45,1
18	45,3
19	45,1
20	45



F3. Exemplo de uma variável de processo.



F4. Tela típica de representação de processo.

Introdução ao VHDL

Este artigo apresenta o que são exatamente os circuitos integrados de lógica programável e como eles funcionam, com uma abordagem técnica e prática desses componentes.

Augusto Einsfeldt

Todo projetista está acostumado a utilizar desenhos esquemáticos para representar circuitos eletrônicos. Naturalmente este é um modo fácil de também representar circuitos lógicos. Em poucos instantes, analisando um esquema é possível entender o funcionamento do circuito e, desde que o esquema seja bem feito, qualquer outro projetista poderá trabalhar sobre o mesmo e não apenas o projetista original.

Contudo, quando o projeto é complexo, o número de folhas de esquemas aumenta, o tamanho das páginas aumenta e o funcionamento já não fica tão claro para qualquer um. Alterações de projeto exigem redesenhar uma ou mais páginas e se a alteração for significativa pode exigir a subdivisão de um circuito em páginas adicionais.

Esquemas são uma boa forma de representar circuitos mas, em muitos casos, um outro meio pode ser mais eficaz e poupar algum tempo no processo de desenvolvimento.

Este outro meio é o uso de uma linguagem textual para descrição do circuito, HDL (*Hardware Description Language*), onde o comportamento do circuito é o que mais importa e não exatamente onde cada sinal é conectado.

Uma descrição comportamental facilita a vida do projetista pois ele não precisa se preocupar com detalhes puramente matemáticos como os mapas de Karnaugh para redução de lógica ou mesmo a construção de decodificadores para selecionar cada passo de uma máquina de estados. Por exemplo, um contador síncrono pode ser descrito como: `if rising_edge(clock) then saida<=saida+1; end if;`. Note-se que o uso de lógica síncrona é natural em HDL (de fato, para fazer um contador assíncrono tipo *ripple-carry* dá um bocado de trabalho. Leia mais no quadro Lógica Síncrona).

Se alguém pensar em fazer o mesmo usando esquemas, precisa lembrar-se que cada estágio do contador possui um porta AND com tantas entradas quanto o número de estágios anteriores. E quantos estágios são necessários? Se for um contador de 8 bits serão 8 estágios. Mas, e se o projeto exigir 2 bits a mais porque ocorreu uma alteração nas especificações? No esquema será preciso desenhar mais dois estágios (se couberem na folha) ou criar uma nova folha para desenhar o contador completo e depois desenhar as referências de interconexão das páginas, etc.

Como isso é feito em HDL? Como pode ser visto no exemplo citado, a largura em

Saiba mais

Sugestões para acesso a mais informações na Internet:

www.gmvhdl.com/ VHDL.html

www.erc.msstate.edu/ ~reese/ vhdl_synthesis/

www.vhdl-online.de/

www.eej.ulst.ac.uk/tutor.html

bits do contador não entra na equação. Ela é definida em uma parte anterior onde o sinal é descrito como: `signal saida : std_logic_vector (7 downto 0);`. Esta linha diz que o sinal saida tem 8 bits de largura. Se o projetista quiser mudar para 10 bits basta alterar esta linha para `signal saida : std_logic_vector (9 downto 0);`. Simples, não?

Linguagens HDLs são aplicáveis no projeto de circuitos integrados (ASICs – *Application Specific Integrated Circuits*) e no uso em circuitos de lógica programável (FPGAs e CPLDs). Naturalmente, existem muitas linguagens de descrição de hardware além do VHDL, como VERILOG, ABEL e AHDL. Como VHDL tem sido mais divulgada e aplicada em empresas e universidades, esta foi escolhida para esta série de artigos que são, de fato, um mini-curso na revista Saber Eletrônica.

A primeira versão disponibilizada da linguagem VHDL (*VHSIC Hardware Description Language*) surgiu em 1987 e foi originalmente encomendada pelas Forças Armadas americanas. Sistemas complexos e produzidos por múltiplas empresas exigiam uma linguagem padronizada para modelagem e simulação. Por isso muitas das instruções não são sintetizáveis, ou seja, não podem ser convertidas em um equivalente em circuitos eletrônicos e servem, por exemplo, para permitir a simulação de circuitos. Em 1993 surgiu uma nova revisão que é empregada até hoje, o VHDL-93.

Fundamentos

Antes de tudo é importante saber como um texto que descreve o comportamento de um circuito pode ser convertido no circuito eletrônico propriamente dito. Existem duas etapas principais neste processo: síntese e implementação.

A síntese é a tradução da linguagem HDL em um outro texto descritivo que emprega referências de interconexão entre elementos básicos de lógica digital como portas lógicas, registradores e pinos de entrada e saída.

A implementação emprega o resultado da síntese e outras referências específicas do dispositivo a ser utilizado (conforme um determinado modelo de FPGA ou de CPLD) para calcular como este dispositivo deverá ser programado para obter-se o circuito esperado.

A síntese é equivalente à compilação em *software*: traduz-se de uma linguagem como C para código *assembly*. A implementação é equivalente a traduzir a descrição *assembly* em código objeto para um processador.

Apesar do uso da linguagem VHDL estar relacionado, de forma direta, apenas com a síntese, durante o desenvolvimento de um projeto deve-se levar em conta os efeitos na implementação conforme o dispositivo utilizado. Uma função lógica com muitas entradas poderá ser implementada mais eficientemente em um CPLD do que em um FPGA. Contudo, se esta mesma função for descrita de maneira a ser um agrupamento de pequenas funções o resultado poderá ser mais adequado para um FPGA do que para um CPLD. Isso ocorre porque os CPLDs tem poucos geradores de função (até poucas centenas), porém cada um com muitas entradas (alguns com mais de 40), ao passo que o FPGA tem muitos geradores de funções (até muitos milhares) com poucas entradas (tipicamente entre 4 e 6).

Outro aspecto é que a descrição em VHDL pode ser ou muito específica ou, por outro lado, mais genérica. Ela é específica quando indica claramente como o circuito deverá ser montado e utilizando recursos particulares do dispositivo. Por exemplo, indicando o uso de um componente chamado CB8CE, que é natural dos componentes Xilinx, o projetista estaria empregando um contador binário de 8 bits. Isso também é chamado de instanciação (ou ocorrência forçada). Já uma descrição genérica deixaria a ferramenta de síntese calcular qual circuito representaria uma função declarada. Neste caso a declaração `valor <= valor + 1;` produz o mesmo resultado final, ou seja, um contador binário. Este cálculo feito durante a síntese é chamado de inferência (ou dedução).

Naturalmente a inferência é mais simples de usar e de se entender quando o projeto de outra pessoa é lido. Pode existir, contudo, uma grande diferença no resultado final: o componente CB8CE emprega recursos otimizados dos dispositivos Xilinx e garantidamente oferecerá o melhor desempenho em espaço ocupado e velocidade, enquanto a inferência poderá usar recursos genéricos dos dispositivos e ser menos rápida ou ocupar mais espaço. Todavia, ao empregar o CB8CE o projetista está determinando que o contador terá 8 bits de largura. O que ocorre se ele precisar de 11 bits? Seguindo o raciocínio do uso de componentes ele deveria incluir mais um destes componentes e possivelmente ocupar mais espaço com os 5 bits restantes e desnecessários. Usando a inferência somente o número de bits usado será implementado.

Lógica Síncrona

Um projeto é síncrono quando todos os registradores empregados são acionados por um único sinal de *clock*, ou por uma derivação síncrona deste clock (como alguma saída de um divisor). Este procedimento tem a vantagem de tornar conhecido o momento em que cada sinal será registrado.

Em um projeto assíncrono o acionamento dos registradores é feito pelo resultado da lógica combinacional de um ou mais sinais e está sujeito a variações de atrasos e escorregamentos inerentes a estes circuitos. Neste caso, o projetista não pode prever com precisão o momento em que vão ocorrer e isso pode ocasionar falhas funcionais. Um caso típico destes acontece quando um determinado projeto só funciona bem quando utiliza-se um FPGA ou CPLD fabricados em uma data específica e ainda trabalhando sob temperatura controlada. Qualquer mudança nos atrasos internos devido à variações térmicas ou do processo de fabricação pode comprometer o funcionamento.

Um contador síncrono muda todas as suas saídas quase no mesmo instante, independentemente do número de bits, enquanto um contador tipo *ripple-carry* terá seu último bit mudando após o tempo somado de todos os atrasos de cada estágio.

Todo projeto pode ser síncrono. Apenas em alguns casos poderá ser necessário o uso de um clock de referência mais alto do que o exigido no projeto para garantir a inclusão de mudanças rápidas em alguns sinais.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity main is
    Port ( LED : out std_logic;
    CLOCK : in std_logic);
end main;

architecture RTL of main is
signal div : std_logic_vector (7 downto 0);
begin
    process (clock)
    begin
        if rising_edge(clock) then
        div <= div + 1;
        end if;
        end process;

        led <= div(7);
end RTL;
```

Quadro 1: Um projeto VHDL que pisca um LED ao ritmo de CLOCK/256.

As ferramentas de síntese são capazes de um bom nível de otimização que resulta no emprego dos recursos específicos dos fabricantes de lógica programável. Por isso, preferencialmente, o projetista deve descrever o hardware usando inferência, tornando o projeto mais claro e mais fácil de ser desenvolvido. Mais a frente veremos este assunto novamente.

Importante: os sinais e comandos (palavras reservadas) em VHDL podem ser escritos tanto em minúsculas como em maiúsculas ao longo de todo o texto.

VHDL

VHDL é uma linguagem típica: simples em sua funcionalidade básica e complexa quando o projetista aproveita todos os seus recursos.

Cada projeto em VHDL é sempre distribuído em um ou mais arquivos de texto, geralmente coma extensão **.vhd**. Cada arquivo contém sempre a descrição funcional completa de um elemento do projeto ou mesmo do projeto inteiro. De fato, o arquivo VHDL de um elemento é um projeto em si sendo usado como um componente. O arquivo principal, topo do projeto, que contém referências de interconexão com os outros arquivos pode ser imaginado como a descrição de um diagrama de blocos. Geralmente, funções complexas e já verificadas são mantidas como um projeto individual (análogo a uma sub-rotina bem depurada quando programamos um computador).

```
    type bit is ('0', '1'); Sel <= '1';
    type boolean is (false,true); if enable then
    type integer is range 0 to 103; soma := conta + 32;
    - se não especificar o range: assume 32 bits
    type real is range - 31.5 to 47.2;
    - útil apenas para simulação. Não é sintetizável
    type std_logic is ('Ú','X','0','1','Z','W','L','H',' ');
    - também existe o std_ulogic mas este é mais rígido e não permite situações conflitantes como múltiplas saídas
    tri- state conectadas
    type my_type is (RST, LOAD, FETCH, SHIFT);
    - este é o tipo enumerated
```

Quadro 2: Tipos Escalares e alguns exemplos de uso.

Assim o projetista não precisa re-escrever ou copiar trechos extensos em VHDL para um único arquivo, mantendo a legibilidade e repetibilidade dos resultados.

A estrutura de um arquivo VHDL contém sempre a declaração das bibliotecas utilizadas e duas seções básicas e obrigatórias: ENTITY e ARCHITECTURE, como pode ser visto no exemplo do **quadro 1**. No texto deste quadro todas as palavras e caracteres que são reservadas do VHDL ou tipicamente obrigatórias em qualquer projeto (como as bibliotecas) estão marcados em negrito. As poucas partes que não estão em negrito são aquelas dependentes e produzidas pelo próprio projetista.

A seção ENTITY (entidade) serve como um cabeçalho onde são indicados todos os sinais de entrada e de saída do projeto ou elemento a ser descrito. No exemplo, a palavra main é o nome da entidade e normalmente também será o nome do arquivo VHDL. Os sinais no "Port" da entidade possuem um descritor indicando o sentido do sinal: In, Out e Inout e o tipo do sinal. Os tipos mais usados são: Bit, Std_logic, Bit_vector e Std_logic_vector. Na prática utiliza-se preferencialmente std_logic ou std_logic_vector. A razão disto será vista mais adiante.

A seção ARCHITECTURE (arquitetura) é onde o projetista descreve o funcionamento da entidade. A primeira linha da arquitetura deve indicar duas coisas: o tipo de arquitetura (no exemplo: RTL) e o nome da entidade à qual a arquitetura está relacionada (no exemplo: main). O tipo da arquitetura pode, de fato, ser qualquer nome escolhido pelo projetista. Programas que geram a estrutura de um arquivo VHDL automaticamente (como o ISE da Xilinx) costumam colocar o tipo como BEHAVIORAL. O VHDL não examina esta informação. Você pode usar quaisquer palavras tais como: COMPORTAMENTO, DESCRICAO, MINHA_ARQUITETURA, ABC123, etc. Em suma, o tipo da arquitetura serve mais para a pessoa que vai ler o projeto ter uma ideia da finalidade e a palavra usada não é importante para o VHDL.

A arquitetura tem um início (*begin*) e um fim (*end*). Entre o *begin* e a primeira linha (*architecture*) são declarados os sinais internos que serão usados neste projeto ou elemento de projeto. Os sinais devem ser entendidos como fios que vão interligar os componentes e portas lógicas do circuito. No exemplo do quadro 1 o único sinal utilizado que já não foi declarado no "port" da entidade é um barramento de 8 bits que é a saída de um contador de 8 bits chamado DIV. Entre o begin e o end deve ser incluído um texto descrevendo a lógica do circuito. No caso do exemplo esta descrição possui um PROCESS (processo) e uma linha de lógica combinacional muito simples que diz que o sinal LED deve ser fornecido a partir do bit 7 do contador DIV.

Equações lógicas podem ser facilmente implementadas e todos os operadores (funções) comumente usados estão à disposição. Uma equação deste tipo seria, por exemplo: clk_x <= clock and (not div(0)) and div(3);. Neste caso um sinal de saída chamado "clk_x" seria o resultado de uma porta AND de três entradas com os sinais clock, bit 0 de DIV invertido e o bit 3 de DIV. Já um decodificador de endereço poderia ser descrito como chipselect <= '0' when (addr = "0011000000000001" and wr = '0' and ale = '1') else '1';. Note-se que o sinal "chipselect" vai ficar em nível lógico zero quando o barramento de endereço for 0x3001 e ao mesmo tempo tendo os sinais "wr" em nível zero e "ale" em um. Caso contrário, o sinal ficará em 1. Esta operação lógica utilizou um recurso do VHDL que é um teste condicional. Para fazer isso em esquemático levaria um pouco mais de tempo...

Tipos de sinais

Existem diversos tipos pré-definidos para serem atribuídos aos sinais. Eles são divididos em dois grupos: ESCALARES e COMPOSTOS. O **quadro 2** mostra os tipos escalares.

O tipo BIT somente pode ter dois estados: 1 e 0. Por esta razão ele não poderia ser associado a um sinal que possa ficar em alta impedância (*tri-state*). O tipo BOOLEAN serve para verificação em testes condicionais. O tipo INTEGER é útil para equações matemáticas e contadores, mas não é prático quando se precisa extrair um ou mais bits para serem usados em outras equações. Do ponto de vista do hardware um sinal INTEGER é, de fato, um barramento de n bits, sendo o número de bits correspondente ao necessário para expressar toda a gama de valores que o sinal pode ter (indicado no range).

O REAL é um tipo apenas aplicável em simulações e não poderá ser sintetizado, isto é, a ferramenta de síntese não saberá como traduzir um sinal REAL em hardware. Note-se que operações aritméticas com números reais somente podem ser feitas através da implementação de um hardware correspondente, para o processamento em ponto flutuante. Tal implementação fica por conta do projetista.

O tipo STD_LOGIC é o mais usado. Ele permite que o sinal possa assumir até 9 estados diferentes e isso permite fazer verificações funcionais mais completas além de tornar viável a implementação de sinais em tri-state (Z). O significado de cada um dos estados possíveis é: U : Não inicializado; X : Forçado "Unknown"; 0 : Forçado Zero; 1 : Forçado Um; Z : Alta impedância; W : "Unknown" fraco; L : Zero fraco; H : Um fraco; - : Don't care. A maioria dos estados é somente útil em projetos mais complexos. Os estados '0', '1', 'Z' e '-' são mais usados.

O tipo MY_TYPE (apenas no exemplo pois poderia ter qualquer nome) é também chamado de tipo enumerado, ou seja, onde cada estado possível recebe um nome escolhido pelo projetista. Durante o processo de síntese os sinais de tipo enumerado são montados como barramentos com tantos bits quantos necessários para, através de números binários ou *one-hot* (barramento de n bits onde apenas um bit de cada vez pode ficar em nivel 1), representar todos os estados declarados. Observar que, na representação binária poderão ser possíveis mais estados do que os declarados (com 3 bits existem 8 combinações e quando apenas 5 estados são declarados ainda sobram outros três). Durante a síntese estes estados adicionais são tratados internamente como nomes ou códigos atribuídos aleatoriamente. O projetista deve saber disso pois o hardware final, não sabendo como atuar durante a presença destes estados, pode funcionar incorretamente.

Os tipos enumerados são muito usados para definir os possíveis estados de uma máquina de estados finitos (FSM ou Finite State Machine). Obviamente o nome de cada estado ajuda a entender a sua finalidade e é mais amigável que séries de valores binários.

Os tipos compostos apareçem no **quadro 3**. Os VECTORs são pré-definidos e não exigem uma declaração de tipo (type). Um std_logic_vector ou bit_vector representa um barramento de tantos bits quanto os declarados na expressão X downto Y, onde X é o número do bit mais a esquerda do barramento e Y o mais a direita. Embora Y possa ter qualquer valor (sempre menor que X) recomenda-se usar 0 (zero) pois algumas ferramentas de síntese causam erros quando Y é maior que zero. A declaração de largura pode ser tanto DOWNTO como TO.



```
Signal teste: bit_vector (3 downto 0) : = 0"0100";
```

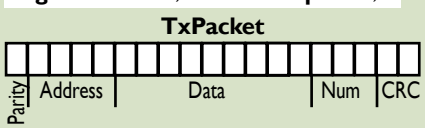
```
array
type memoria is array (3 downto 0) of std_logic;
type dados is array (255 downto 0) of integer range 0 to 9;
```

*__bit_vector__ e __std_logic_vector__ são tipos compostos pré-definidos

a)

```
type OPCODE is recorded
   parity : bit;
   address : std_logic_vetor (3 downto 0);
   data : std_logic_vector (7 downto 0);
   num_value : integer range 0 to 5
   crc : bit_vector (1 downto 0);
end record;
```

signal TxPacket, RxPacket : **opcode**;



b)

```
signal word : bit_vector (15 downto 0);
signal data : bit_vector (7 downto 0);
signal aux : bit vector (3 downto 0);
signal zz : std_logic_vector (0 to3);
signal a, b, c, d : bit;
(a,b,c,d) <= aux; (*)
data<(c,aux(1),aux(2),data(0),'0','0',b,a);
word <= (15 => '1', 14 => data(2), 0
=> c,2 a, others => '0');
zz <= ( others => '1');
```

***Apenas** tipos escalares podem ser agregados no lado esquerdo da instrução.

c)

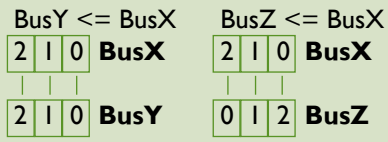
Quadro 3. Tipos Compostos: a) Vetores e Arrays - Os vetores são os mais comuns; b) Record - Barramento composto de vários sinais; c) Agregados (agregates) - forma de associação de sinais.

1. Os arrays e vetores devem ser do mesmo tipo;

2. Os arrays e vetores devem ser do mesmo tamanho;

3. As designações são posicionais, da esquerda para direita;

```
signal BusX, BusY : bit_vector (2 downto 0)
signal BusZ: bit_vector (0 to 2);
```



Quadro 4. Atribuições (conexão) com vetores e arrays. A importância na ordem do barramento.

O VHDL não impõe restrições e a ordem (do bit mais significativo para o bit menos significativo) sempre é feita da esquerda para a direita.

Recomenda-se usar consistentemente apenas uma orientação, pois do contrário uma conexão entre dois sinais pode significar a inversão de um barramento (veja o **quadro 4**).

Normalmente, usa-se a ordem DOWNTO por ser a mesma da representação em esquemas e a mais natural aos projetistas de circuitos lógicos.

A inicialização, feita na declaração dos sinais (como indicado no exemplo do **quadro 3a** – :="0100"), é útil para a simulação e não é sintetizável. Quando um sinal não é inicializado, os simuladores não conseguem calcular os resultados das lógicas dependetes deste sinal. No caso do ModelSIM, um ou mais sinais e outras lógicas dependentes deste de um sinal não inicializado aparecem na janela de análise como uma linha vermelha.

Um tipo muito interessante e que desperta a imaginação dos projetistas é o RECORD. Com ele, monta-se um barramento complexo onde cada parte pode ser tratada individualmente.

Depois este barramento pode ser carregado, digamos, em um registrador de deslocamento (*shift register*) e a saída resultante usada numa interface serial de comunicação. Outro exemplo é o uso do record para descrever microinstruções de um processador. Para acessar as partes individuais de um record basta usar:

```
<signal_name>.<field_name>
```

Exemplo:

```
Signal contador : std_logic_vector (3 downto 0);
TxPacket.address <= contador;
TxPacket.num_value := 3;
TxPacket.parity <= contador (0) xor contador
(1) xor contador (2) xor contador (3);
...
```

Finalmente, os agregados são na verdade a forma de associar partes individuais de outros sinais e constantes de forma a compor um sinal diferente. O uso da palavra reservada "others" indica que todas as demais posições não explicitamente indicadas devem assumir o estado apontado. Os agregados somente podem ser compostos de elementos associados de forma posicional ou nomeada.

Os dois modos não podem ser usados juntos na mesma atribuição. Vale dizer neste momento que a forma posicional serve para indicar a posição exata de um bit de um barramento. Como no exemplo em 3c, segunda linha, onde o sinal c vai ligado no bit 7 de data, o sinal aux(1) vai no bit 6 e assim por diante.

Já a forma nomeada indica em que bit do barramento o sinal em questão será ligado. Veja a terceira linha do exemplo onde o sinal data(2) é ligado ao bit 14 de word. Lembre-se disso, pois estas formas posicional e nomeada são empregadas em outras referências no VHDL.

Adicionalmente ao exemplo uma expressão data <= (a,b,c,d,aux); não funcionaria porque a ferramenta de síntese percebe que os tipos de data e aux são diferentes.

Também a expressão zz <= aux; não funcionaria pelo mesmo motivo, contudo isso pode ser resolvido, neste caso, empregando-se uma função de conversão de tipos: zz <= to_stdlogicvector (aux);

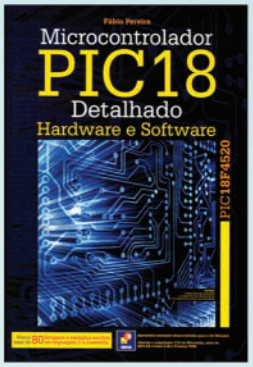
Ainda sobre sinais, na representação de valores, conforme o tipo usa-se:

```
Bit ou Std_logic = '0' ou '1' ou 'Z'
Bit_vector ou Std_logic_vector = "0010"
ou "1110101001Z00"
Intege = 23 ou 18594 ou 89234551
```

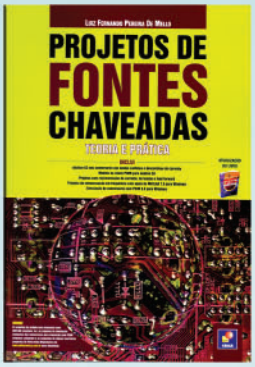
Ótimos livros por um excelente preço



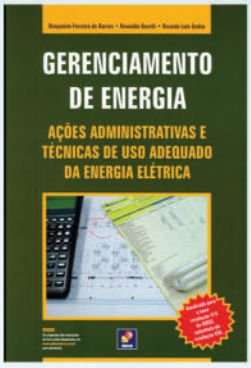


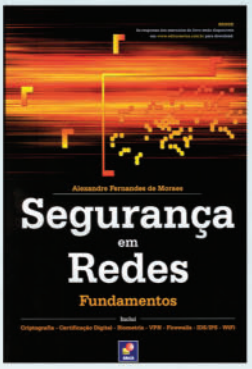


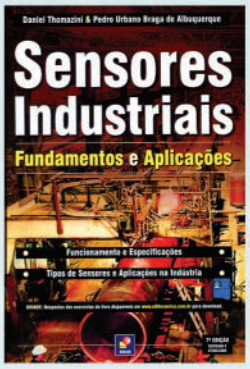






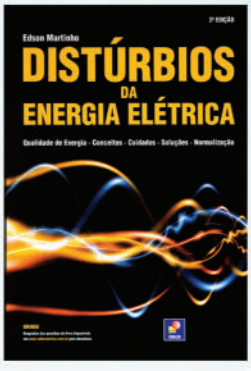


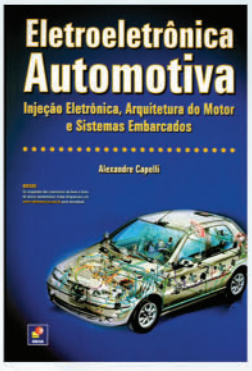


























novasaber
www.novasaber.com.br

Fonte fixa de
12/12,6/13,2 V x 10 A

Equipamentos de uso móvel como transmissores podem ser facilmente alimentados pela bateria do carro, no entanto, se eles forem usados em uma bancada ou precisarem passar por testes, ajustes ou reparações precisarão de uma fonte cuja capacidade de corrente está muito além da fornecida por projetos comuns. O circuito que descrevemos aqui é ideal para alimentar transmissores de comunicações de uso móvel com potências até 50 W e mesmo para circuitos especiais, a exemplo dos inversores.

Newton C. Braga

Equipamentos de uso móvel tais como transmissores de VHF, FM, 11 metros, radioamadores das faixas de ondas curtas, inversores, etc., exigem correntes elevadas quando alimentados por uma fonte. Em alguns casos, a corrente de operação a máxima potência pode chegar aos 10 ampères, o que inviabiliza o uso de fontes comuns ou simples eliminadores de pilhas.

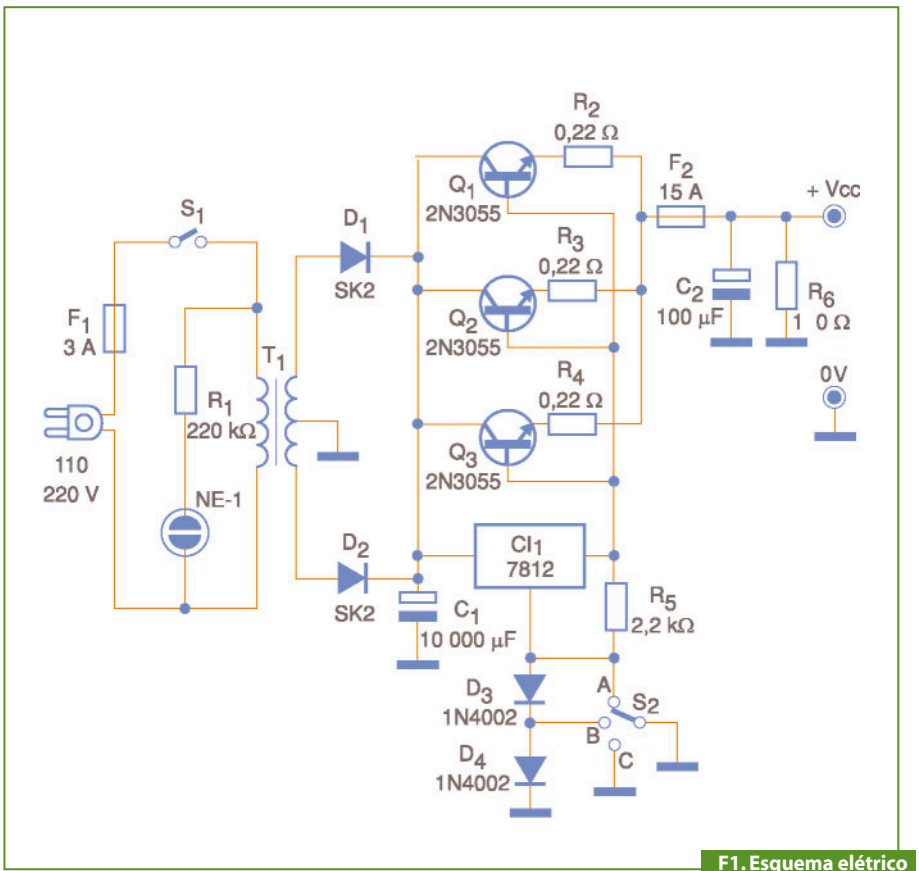
O que propomos neste projeto é uma excelente fonte de alimentação de 12 a 13,2 V que pode fornecer correntes até 10 A, servindo pois para a alimentação de todos os equipamentos descritos na introdução.

O uso de um circuito integrado de precisão, como referência de tensão, torna esta fonte muito estável e segura na alimentação de qualquer equipamento indicado para o carro ou bateria como fonte de alimentação.

Como funciona

O circuito de entrada consta de um transformador que abaixa a tensão da rede de energia de 110 V ou 220 V para uma tensão menor, entre 15 e 20 V sob corrente máxima de 10 A.

Há transformadores que podem ser encontrados prontos para esta finalidade. Mas, se isso não for possível, o leitor terá



F1. Esquema elétrico da fonte fixa.

duas alternativas: uma será mandar enrolá-lo e outra será usar um de tensão na faixa indicada, mas com corrente menor, caso em que a capacidade da fonte ficará reduzida. Neste caso, se ela puder ainda ser usada com o equipamento que o leitor dispuser, não haverá nenhum problema.

A tensão reduzida pelo transformador é então retificada por uma ponte de diodos. Podem ser colocados diodos separados ou pontes completas, conforme indicadas na lista de material.

Como cada diodo só conduz durante metade do ciclo de alimentação, sua especificação de corrente deve ser apenas um pouco maior que metade do valor máximo de saída. Assim, diodos para 8 A ou mais podem ser utilizados. Sua tensão inversa de pico (PIV) deve ser de pelo menos 100 V para maior segurança.

Logo após a retificação temos a filtragem, que deve ser excelente para que não ocorram zumbidos na reprodução sonora dos aparelhos alimentados.

É uma prática segura usar 1 000 µF para cada ampère de corrente de saída com uma fonte até 12 V. Recomendamos, portanto, um capacitor de 10 000 µF ou ainda dois capacitores de 4 700 µF x 35 V ligados em paralelo.

Temos finalmente a etapa de regulação: a referência de tensão é dada pelo circuito integrado 7812, o qual deverá ser montado num bom radiador de calor. Ligando ao terminal de referência deste integrado diodos, podemos alterar levemente a tensão de saída. A chave S_2 se encarrega disso.

Para cada diodo colocado no circuito temos um acréscimo de 0,6 V o que nos leva a 13,2 V, que é aproximadamente a tensão que encontramos nos terminais de uma bateria de uso automotivo de 12 V quando completamente carregada.

O circuito integrado 7812 só fornece correntes até 1 A, mas esta intensidade é suficiente para excitar três transistores 2N3055 ligados em paralelo. Cada transistor poderá então controlar 3,33 ampères, o que está bem dentro de sua capacidade.

Veja que nos emissores dos transistores temos resistores de 0,22 ohms. Sua finalidade é distribuir igualmente a corrente entre estes componentes, pois suas características não são exatamente iguais. Sem os resistores existe o perigo de um transistor ter de conduzir uma corrente maior que outro, sofrendo com isso uma sobrecarga que pode levá-lo à queima.

Na saída do circuito, além de um fusível adicional de proteção de 15 A temos também um capacitor de desacoplamento e um resistor de carga, cuja finalidade é manter a tensão nos terminais, mesmo na ausência de uma carga externa.

Montagem

O circuito completo da fonte é mostrado na **figura 1**.

A disposição dos componentes em uma placa de circuito impresso é ilustrada na **figura 2**.

Os transistores devem ser montados em radiadores de calor de pelo menos 50 cm quadrados (5 cm x 10 cm), com fixação em local de boa ventilação.

O regulador de tensão CI_1 deve ter um radiador de calor menor.

Para a retificação pode ser usada uma ponte Semikron SKB 7/02, ou ainda os diodos SK2 ou SK3 para 200V.

O capacitor eletrolítico de filtragem, de 10 000 µF, pode ser obtido com a ligação de dois eletrolíticos de 4 700 µF x 35 V em paralelo.



Os resistores R_2, R_3 e R_4 são de fio de 0,22 ohms com pelo menos 2 W de dissipação. Tendo dificuldades em se conseguir estes valores de resistores, existem alternativas:

- Ligação de dois resistores de 0,47 ohms x 1 W em paralelo.
- Ligação de quatro resistores de 1 ohm x 1W em paralelo.

A chave S_2 seleciona a tensão de saída: na posição A temos 12 V, na posição B temos 12,6 V e na posição C temos 13,2 V.

O resistor R_6 deve ser de fio com pelo menos 5 W de dissipação.

Observamos que os fios de ligação que conduzem as correntes principais devem ser todos grossos (18 ou 16).

Uma outra modificação possível para este projeto consiste em se alterar a quantidade de transistores 2N3055 ligados em paralelo, caso a corrente máxima desejada seja um pouco menor ou maior que a indicada aqui.

Por exemplo, com dois transistores 2N3055 poderemos ter uma fonte até 5 ou 6 A; e com apenas um, a fonte chegará aos 3 ou 4 A de saída.

Por outro lado, acrescentando mais um transistor, levando a 4, poderemos ter saídas que superam os 12 A. Evidentemente, o transformador e os diodos deverão ter suas especificações alteradas para as novas correntes que deverão ser fornecidas.

Lista de Materiais

Semicondutores:

CI_1 - 7812 - circuito integrado regulador de tensão
Q_1, Q_2, Q_3 - 2N3055 - transistores NPN de alta potência
D_1 a D_2 - diodos SK2 ou SK3 (Semikron)
D_3, D_4 - 1N4002 - diodos de silício

Resistores:

R_1 - 220 kΩ x 1/8 W
R_2, R_3, R_4 - 0,22Ω x 2 W - fio
R_5 - 2,2 kΩ x ½ W
R_6 - 100Ω x 5 W - fio

Capacitores:

C_1 - 10 000 µF x 35 V - eletrolítico - ver texto
C_2 - 100 µF x 16 V - eletrolítico

Diversos:

NE_1 - lâmpada néon comum
S_1 - Interruptor simples
S_2 - Chave de 1 polo x 3 posições, rotativa
T_1 - Transformador com primário conforme a rede local de energia e secundário de 15 a 22 V x 10 A - ver texto
F_1 - Fusível de 3 A
F_2 - Fusível de 15 A

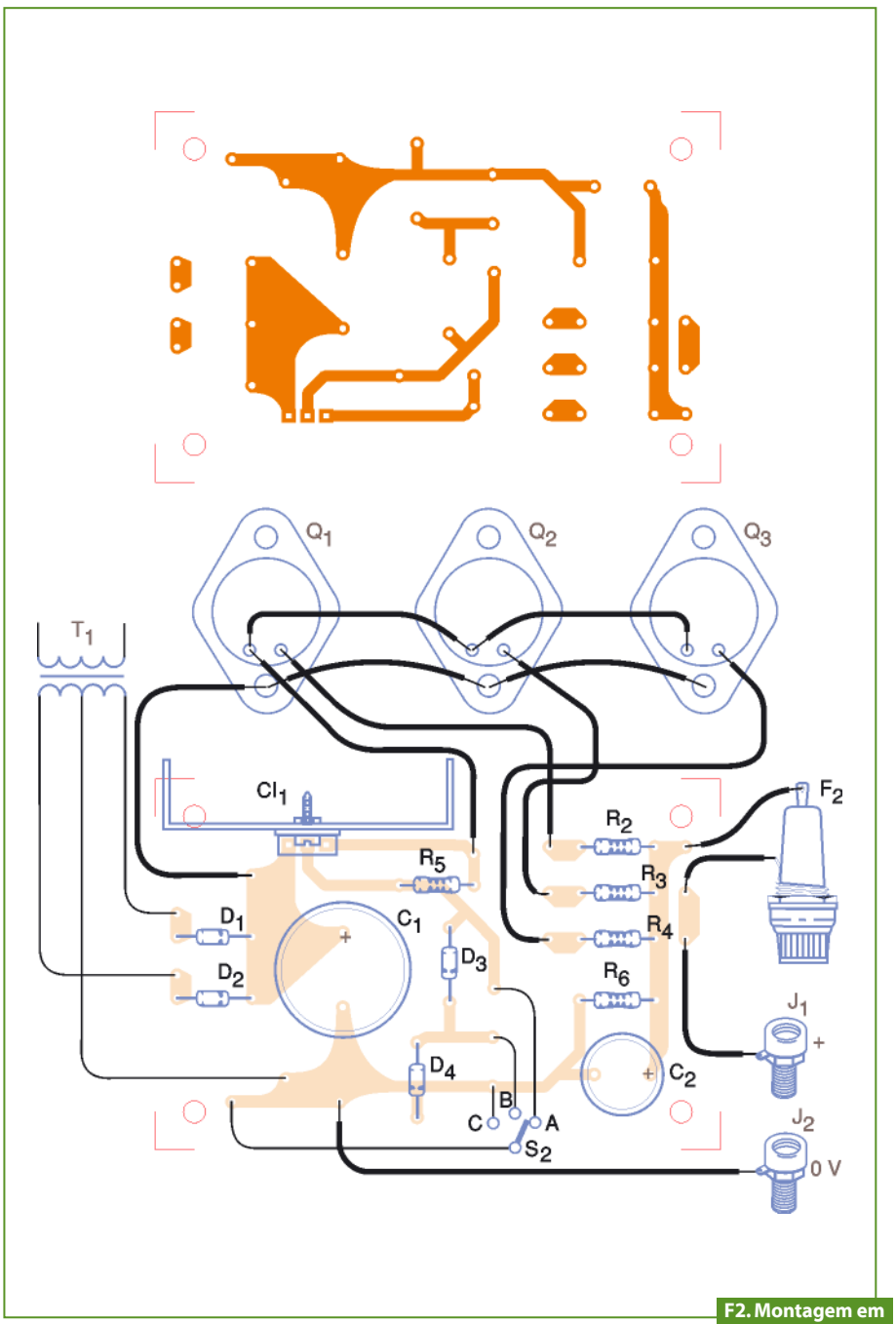
Placa de circuito impresso, radiadores de calor para os transistores, fios, solda, cabo de força, suporte para os fusíveis, caixa para montagem, fios com garras para conexão ao aparelho alimentado ou conector do tipo acendedor de cigarros de carro, fios comuns, solda etc.

Prova e uso

Para provar a fonte, é só ligar algum aparelho com consumo conhecido em sua saída, e verificar se a tensão se mantém estável. Uma lâmpada de 12 V de farol de carro é uma boa carga para esta prova.

Na ligação de equipamentos eletrônicos em geral tais como transmissores, aparelhos de som, transceptores e toca-fitas é preciso observar a polaridade dos fios.

A lâmpada néon é opcional, servindo apenas para indicar que a fonte se encontra ligada.



F2. Montagem em placa de circuito

Os Mais Confiáveis Dispositivos de Aquisição de Dados Agora Sem Fio!



Apresentando o Novo NI Wi-Fi DAQ - os dispositivos para aquisição de dados (DAQ) sem fio da National Instruments que simplificam aplicações de monitoração remota de fenômenos físicos, mecânicos, elétricos e acústicos. Os mais de 20 anos de experiência em aquisição de dados da National Instruments tornaram possível o desenvolvimento dos novos NI Wi-Fi DAQ, que integram conectividade direta com sensores, a flexibilidade da programação do NI LabVIEW e a maior segurança de rede disponível comercialmente para atender as necessidades das suas aplicações de monitoração estrutural, ambiental e industrial.

>> Conheça os novos dispositivos Wi-Fi acessando **ni.com/wifi** **(11) 3149-3149**

©2008 National Instruments. All rights reserved. LabVIEW, National Instruments, NI, and ni.com are trademarks of National Instruments
Other product and company names listed are trademarks or trade names of their respective companies. 2008-9486-301-101

NATIONAL INSTRUMENTS

Texas Intruments e National Semiconductor

Fazendo **MAIS** Juntas

- **MAIS** projetos e desenvolvimentos
- **MAIS** produtos
- **MAIS** suporte
- **MAIS** capacidade de manufatura

MAIS para nossos clientes

Juntas somos parceiras comprometidas com voce, nosso cliente, reconhecendo que nosso sucesso depende do seu sucesso.



Texas Intruments www.ti.com/brasil
Distribuidores: Arrow (11)3613-9300;
Avnet (11) 5079-2150;
Farnell-Newark (11)4066-9400
e-mail: texas-suporte@ti.com - tel: (11) 5504-5133

Texas Instruments

A barra vermelha é marca registrada da Texas Instruments © 2011 Texas Instruments incorporated.