Ano 49 • Nº 474
Europa € 5,30
Brasil R$15,90

www.sabereletronica.com.br

SABER ELETRÔNICA

http://twitter.com/editora_saber

Industrial

Interface gráfica para comunicação entre Arduino e PC



Conversores Buck-Boost
desenvolvidos para "aplicações de micropotências" e "internet das coisas"

Utilização do Protocolo SPI na comunicação entre microcontroladores

Aquisição de dados com Smartphones e Tablets

Regulador chaveado de baixo ripple para MCUs e FPGAs

ISSN 0101-6717

Mais de 4 milhões de componentes eletrônicos oferecidos.
MOUSER ELECTRONICS
br.mouser.com

skype

Assistência em Português

GO br.mouser.com

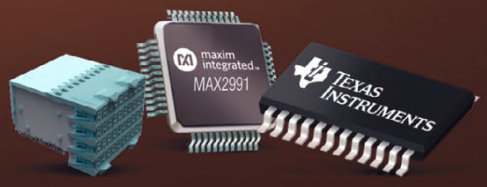
Os mais novos produtos para os projetos mais inovadores®

Acesso instantâneo.

Mais de **4 milhões** de componentes eletrônicos oferecidos por mais de **500 fabricantes** líderes reconhecidos da indústria.



Mouser é sua fonte autorizada para os mais novos semicondutores e componentes eletrônicos para os seus projetos mais inovadores



AUTHORIZED ecia SOURCE

Distribuimos semicondutores e componentes eletrônicos para Engenheiros de Desenvolvimento e Projetistas

MOUSER ELECTRONICS

Mouser e Mouser Electronics são marcas registradas da Mouser Electronics, Inc. Outros produtos, logotipos e nome de empresas mencionados aqui podem ser marcas de seus respectivos proprietários.

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 A. Zuffo

Redação
Rafaela Turiani

Revisão Técnica
Eutíquio Lopez

Designers
Carlos C. Tartaglioni,
Diego M. Gomes

Publicidade
Caroline Ferreira,
Marileide de Oliveira

Colaboradores
Bill Messner,
Dave Salerno,
Dawn Tilbury,
Edriano C. de Araújo,
Eutíquio Lopez,
Flávio da Cruz,
Guilherme Kenji Yamamoto,
Gustavo Leão Moreira,
Gustavo Peixinho,
Renan Machado de Azevedo,
Renato Araújo de Andrade.

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

Capa
Arquivo Editora Saber

Impressão
EGB 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

Os projetistas que o Brasil necessita


Hélio Fittipaldi

Nesta edição apresentamos uma interface gráfica para comunicação entre o Arduino (Tatuino) e o PC, artigo desenvolvido pelo engenheiro Edriano Carlos de Araújo. Há pouco tempo atrás, uma pessoa sem conhecimento avançado em programação, eletrônica e sistemas operacionais para PCs não conseguiria fazer o seu próprio projeto.

Graças ao avanço tecnológico e, principalmente, à internet, que reúne as pessoas em comunidades, oferecendo acesso a softwares distribuídos gratuitamente para uso individual e às publicações de artigos técnicos. Todo este novo mundo está acessível para quem tenha um mínimo de conhecimento em programação e eletrônica básica.

Vimos em épocas passadas pequenas empresas que, a partir de um conhecimento básico, evoluíram e se tornaram grandes empresas. Uma delas foi adquirida recentemente por uma multinacional.

Em diversas ocasiões recebemos solicitações de indústrias espalhadas pelo Brasil para indicação de profissionais de projetos versados em telecomunicações, eletrônica analógica e digital, TI etc.

Hoje em dia, projetos de novos produtos são postergados em até dois anos por falta de gente competente para desenvolvê-los. A realidade atual mostra que um engenheiro que domine estas diversas áreas não é encontrado com facilidade. A solução cada vez mais frequente é a engenharia ser contratada por serviço de empresa terceirizada, como uma Design House. O difícil é encontrá-las. Em vista disso, resolvemos fazer o nosso primeiro evento "Saber Electronics Design Show", em 5 e 6 de maio de 2014, em São Paulo.

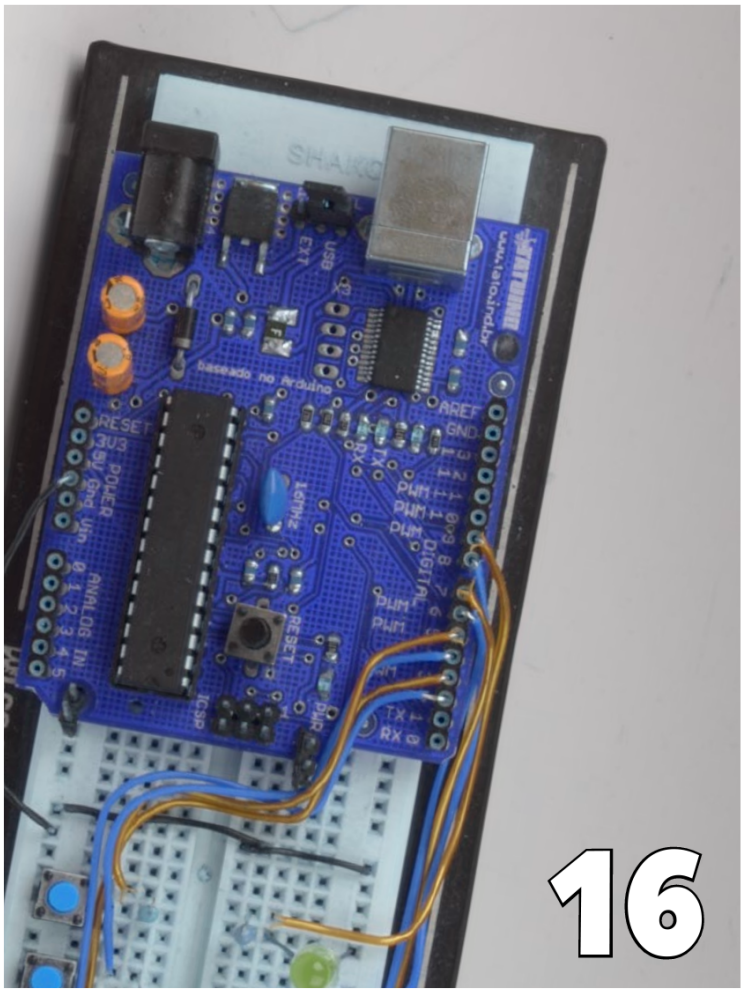
Aproximaremos a indústria que precisa contratar projetos, dos projetistas, das empresas fabricantes e das distribuidoras de componentes eletrônicos. Em breve, daremos maiores detalhes.

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 ideias 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





Instrumentação

10 Aprenda os Conceitos Básicos da Amostragem Analógica

Projetos

16 Interface Gráfica para Comunicação entre Arduino e PC

24 Aprenda como Projetar um Sistema de Controle: Modelagem de Simulação

31 Projeto de um Regulador Chaveado de Baixo Ripple para MCUs e FPGAs

Componentes

36 Conversores Buck-Boost com Corrente Quiescente de 1,3 µA, desenvolvidos para "Aplicações de Micropotências" e "Internet das Coisas"

Microcontroladores

41 Utilização do Protocolo SPI na Comunicação entre Microcontroladores

Eletrônica Aplicada – Telecomunicações

46 Aquisição de Dados com Smartphones e Tablets, passo a passo

03 Editorial

06 Acontece

06 Edital TI Maior, da Finep, contempla associadas da Abinee

06 Termômetro IR Visual VT04, da Fluke, detecta problemas instantaneamente

07 Lexmark investe em soluções para a área de saúde

07 Memórias SDRAM CMOS de alta velocidade, da Alliance Memory

08 Desempenho adequado com menor consumo

08 Usinas solares podem gerar empregos e reduzir as emissões de gases que causam o efeito estufa

09 RAD lança rede para comunicações em empresas de energia.

09 Solução em automação predial, da Schneider Electric

Índice de anunciantes

Cika 05
Globtek 15
KeyStone 23
Tato 30
Patola 30
Nova Saber 45
Mouser ..... 2ª Capa
Saber Educacional ..... 3ª Capa
National Instruments ..... 4ª Capa



Holtek agora também é ARM

NOVA LINHA DE MICROCONTROLADORES HT32Fxxx

Holtek inova com o lançamento de sua nova linha de microcontroladores HT32Fxxx que une todo o poder de processamento do core ARM Cortex-M3 com sua experiência de mais de 20 anos desenvolvendo microcontroladores de 8 bits, resultando em um microcontrolador poderoso com um set de periféricos abrangente altamente competitivos em preço.

CARACTERÍSTICAS:

Com a integração do núcleo Cortex-M3 em seus dispositivos os desenvolvedores contam com as seguintes vantagens:

- **Frequência de operação de até 72MHz.**
- **1,25 DMIPS/MHz**
- **Multiplicação e divisão em um único ciclo por hardware.**
- **Interrupções vetoradas e aninhadas através do controlador NVIC**
- **Timer de 24-bits**

A arquitetura ARM presente em seus novos microcontroladores de 32 bits, proporciona o uso dos melhores ambientes de programação da atualidade como:

- **Keil MDK-ARM**
- **IAR EWARM**

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 2696-6428 / Fax: 55 11 2696-6422

vendas@cikaeletronica.com

www.cikaeletronica.com

Edital TI Maior, da Finep,
contempla associadas da Abinee

A Finep publicou o resultado final da chamada pública do edital TI maior. Foram 28 propostas aprovadas, no total de R$ 79,6 milhões em recursos de subvenção econômica à inovação. Cinco das propostas aprovadas foram de empresas associadas à Abinee (IBM, HT Micron, Novus, Instramed e EMC), representando 23% do montante de recursos.

O TI Maior visa conceder recursos de subvenção econômica para que empresas desenvolvam produtos e/ou processos inovadores que envolvam significativo risco tecnológico, associados à clara oportunidade de mercado nas áreas de Tecnologias da Informação e Comunicação. O edital é parte do Plano Inova Empresa, lançado em março de 2013. A chamada abrangeu os seguintes temas: Computação em Nuvem; Petróleo, Gás e Mineração; Defesa Cibernética; Componentes, Dispositivos Semicondutores e Eletrônicos; e Aeroespacial. Para ver a lista completa das 28 propostas aprovadas, acesse: **www.download.finep.gov.br/chamadas/subvencao/resultados//Resultado_Final_TI_MAIOR_12112013.pdf**

Produtos

Termômetro IR Visual VT04, da Fluke, detecta problemas instantaneamente

A Fluke Corporation apresenta o Termômetro IR Visual VT04, da Fluke, a mais nova ferramenta para resolução de problemas com câmera digital integrada e sobreposição de mapa de aquecimento, fazendo a ligação entre os termômetros de IR tradicionais e os termovisores. Desenvolvido a partir do Fluke VT02, o VT04 adiciona PyroBlend™ Plus com resolução quatro vezes mais nítida que o VT02 e recursos de alarme automático. Trata-se de ferramenta ideal para resolução de problemas em manutenção industrial e elétrica, HVAC/R e aplicativos automotivos.

O Fluke VT04 é totalmente automático e contém inteligência integrada para que os problemas sejam detectados instantaneamente sem precisar de treinamento. "Compactos e acessíveis para toda a equipe, o VT02 e VT04 foram desenvolvidos para sempre ajudar a todos, visto que é uma ferramenta para resolução de problemas," afirma René Guiraldo, gerente nacional de vendas da Fluke do Brasil.

O VT04 tem recursos avançados de alarme para resolução de problemas intermitentes e difíceis, incluindo alarme de temperatura alta e baixa, que pisca na tela para avisar se a temperatura selecionada pelo usuário ultrapassa a faixa selecionada e captura de imagem em intervalo de tempo, que pode ser definido para capturar em intervalos de 30 segundos e 1 hora. Além disso, conta com alarme de monitoramento automático que inicia a captura de imagem automaticamente após o disparo do alarme de temperatura, possibilitando que os usuários capturarem as imagens assim, graças ao tripé universal, mesmo quando o VT04 não estiver em uso.

Exibe e salva imagens em modos totalmente digitais, infravermelhos ou em três modos mesclados (25%, 50% e 75%) com campo de visão 40% mais amplo que o VT02. Os marcadores detectam pontos quentes e frios indicando as temperaturas mais quentes e mais frias na tela. A leitura de temperatura é informada no ponto central. As imagens são salvas no cartão SD incluso e excluem a necessidade de salvar medições únicas ou múltiplas. O Termômetro IR Visual VT04 contém uma bateria recarregável de íon de lítio.

As imagens do VT04 podem ser importadas para o software para análise e relatório SmartView®, possibilitando a criação de relatórios profissionais que documentam problemas ou reparos feitos para análise do cliente e gerenciamento.



Lexmark investe em soluções
para a área de Saúde

A Lexmark International, uma das maiores companhias do mundo da área de tecnologia, tem ampliado sua participação na área de Saúde, entre outros setores. A atuação da empresa, nesta área, se destaca tanto pelo fornecimento de hardwares (impressoras e multifuncionais), quanto de softwares, os quais compõem soluções integradas específicas e customizáveis. Graças a esse portfólio, a Lexmark está presente em mais de 1,2 mil organizações de saúde em todo o mundo, com mais de 4 mil instalações.

Um dos grandes projetos para esse setor é o Hospital Sustentável, que busca reduzir (ou eliminar) o trânsito de papéis dentro dos hospitais. Em algumas ocasiões, o papel ainda é necessário, mas não precisa circular. Por isso, uma vez gerados ou recebidos, esses documentos são digitalizados de forma descentralizada e disponibilizados para consulta, praticamente em tempo real.

Aliadas às multifuncionais da marca, as soluções da Lexmark mais demandadas e implantadas na área de Saúde são: Mobile Print, que permite solicitar a impressão a partir de um comando disparado por meio de qualquer dispositivo móvel (seja ele iOS ou Android); Captura Única com Classificação Automática de Documentos, que direciona documentos digitalizados para locais específicos de tratamento; e Captura Direcionada, que auxilia o usuário a capturar documentos obrigatórios de determinado processo por meio de um aplicativo instalado na multifuncional.

Todas essas soluções se apresentam como grande oportunidade para consolidar as informações não-estruturadas em todas as áreas funcionais do dia a dia das organizações de saúde, sejam elas grandes conglomerados hospitalares, hospitais individuais, comunitários ou de pequeno porte instalados em áreas rurais.

A Lexmark disponibiliza tecnologia para a área de saúde de forma a abranger as diversas necessidades dos setores de um hospital, que incluem registros e admissões, relatórios, formulários de rotina e ordens administrativas até comunicações internas sobre pacientes, *workflows* via mobile e tratamentos para criar um ambiente de tecnologia seguro.

"A Lexmark deixou de ser somente provedora de soluções de impressão para se tornar fornecedora de soluções *end-to-end*. No setor de admissão de um hospital, por exemplo, estas soluções permitem agilizar o processo e dão eficiência nos trâmites internos, com aderência à resolução 1821/07 do CFM, que trata do registro eletrônico de informações de saúde", comenta Fernando Fagundes, gerente de soluções da Lexmark.

Além de equipamentos e softwares, a empresa continua a oferecer serviços gerenciados de impressão, segmento do qual é líder no Brasil em volume de páginas impressas. O serviço abrange também o gerenciamento do uso dos equipamentos por meio de monitoramento, serviço de suporte proativo e gestão de configuração que, além de aumentar a disponibilidade das máquinas, identifica desperdícios e impressões desnecessárias, reduzindo o volume de impressões.

Memórias SDRAM CMOS
de alta velocidade, da Alliance Memory

Essa empresa americana acaba de estender suas linhas de 64M e 128M de SDRAMs , CMOS, de alta velocidade, com duas novas memórias.

Os componentes recém-lançados fornecem *drop-in* confiável, e podem substituir outras memórias similares em produtos industriais, de telecom, ou de consumo que exijam uma grande largura de banda de memória. Para essas aplicações, a AS4C2M32S-7TCN e a AS4C4M32-7TCN apresentam características de rápido tempo de acesso de clock (menos de 5,4 ns) e altas velocidades de *clock* (até 166 MHz).

As duas SDRAMs funcionam com fonte de alimentação simples de 3,3 V (+ ou – 0,3 V) dentro da faixa comercial de temperaturas (entre 0 e 70 °C). Seus terminais são livres de chumbo e halogênios.

Elas fornecem leitura programável ou comprimentos de escrita *burst* de 1, 2, 4, 8, ou *full page*. Uma função de pré-carga automática permite uma pré-carga de linha temporizada , que é iniciada ao término da sequência de burst. Funções de *refresh* fáceis de usar incluem atualização automática (*self-refresh*), ao passo que um registrador programável possibilita ao sistema a escolha dos modos mais apropriados para maximizar o desempenho.

Os dois *Part-Numbers* colocados no mercado representam os últimos lançamentos da empresa em termos de SDRAMs de alta velocidade, cuja linha completa abrange memórias com densidades de 16 Mb, 64 Mb, 128 Mb, 256 Mb e 512 Mb encapsuladas em packages 50-pin TSOPII, 54-pin TSOPII, 54-ball TFBGA, 86-pin TSOPII e 90-ball TFBGA, respectivamente.

Especificações das SDRAMs		
Part Number	AS4C2M32S-7TCN	AS4C4M32S-7TCN
Densidade	64 Mb	128 Mb
Configuração	2M x 32 bit	4M x 32 bit
Tempo de Acesso	5,4 ns	5,5 ns
Velocidade	143 MHz	166 MHz
Invólucro	90-ball TFBGA	86-pin TSOP II

RAD lança Rede para Comunicações em Empresas de Energia

A RAD, fornecedora global de soluções de comunicação para concessionárias de energia, apresenta a nova versão de sua Rede com Garantia de Serviços (*Service Assured Networking*) voltada para as comunicações de empresas elétricas. A solução conta com o novo recurso de Duplicação de Tráfego. Com esta nova tecnologia, a solução oferece, ao mesmo tempo, o transporte de missão crítica de dados operacionais e administrativos da concessionária e de outras formas de tráfego sobre duas modalidades simultâneas de rede: as tradicionais redes SDH/SONET e as novas estruturas Ethernet com nível de operadora.

"A Duplicação de Tráfego não exige uma implementação customizada, possibilitando aproveitar a estrutura já existente. Isto significa, em síntese, que um expressivo ganho de confiabilidade pode ser obtido a partir de um pequeno investimento em CapEx, uma vez que a solução da RAD lança mão das instalações operacionais e de rede já disponíveis, tornando esta melhoria praticamente livre de custos", explica Mati Epstein, da RAD.

Rede com Garantia de Serviços

A Duplicação de Tráfego é uma das funcionalidades mais avançadas já incorporadas à solução de Rede com Garantia de Serviços da RAD, oferecendo confiabilidade, segurança, escala, gerenciamento e garantia de performance para as operações da rede do núcleo operacional.

Ela fornece comunicação multisserviços para as subestações, comunicações de teleproteção, automação da distribuição e *backhaul* de medição inteligente. Tais capacidades incluem o provisionamento de serviços, gerenciamento de tráfego, sincronização de tempo, TDM, monitoramento contínuo de performance, gerenciamento de falhas e vários mecanismos de resistência a falhas.

Um benefício adicional do uso da Duplicação de Tráfego é que ela permite a transição controlada para núcleo operacional baseado em pacotes. Através dessa abordagem, a rede de pacotes pode ser usada inicialmente apenas para o transporte em modo redundante, tornando-se paulatinamente a principal rede operacional à medida que for ampliando sua confiabilidade.

"A Duplicação de Tráfego pode também reduzir significativamente a latência e garantir níveis de atraso bem mais baixos do que os observados no padrão SDH/SONET, o que melhora a performance para aplicações críticas, como é o caso de teleproteção", completa Epstein.

A lista de clientes RAD no setor de *utilities* inclui empresas como AES, CELPE, Endesa, Enel, Eskom, Hydro Quebec, IEC, Kepco, Meralco, E-on, Elia, Nampower, National Grid USA, Novosibirsk Energo, Powercor, and Terna Italia.

Produtos

Solução em automação predial, da Schneider Electric.

A arquitetura SmartStruxure, capaz de reduzir custos com energia em até 30%, foi lançada no Brasil. Trata-se de um sistema inteligente para gerenciamento de energia em edifícios, que tem como diferencial a versão Lite, voltada para a automação de pequenos e médios edifícios. A nova plataforma facilita o intercâmbio e análise de dados de energia e iluminação, além do sistema de aquecimento, ventilação e ar condicionado (HVAC), contribuindo para otimização do uso de energia no edifício. Como parte essencial da infraestrutura predial, a solução SmartStruxure e o SmartStruxure Lite proporcionam integração com apenas um clique, podendo ser acessada de um computador com Web browser, dispensando a aquisição de softwares adicionais. Os programas de gerenciamento possuem fácil manuseio, onde é possível personalizar a área de trabalho, acessar as informações a qualquer hora e em qualquer lugar e simplicidade nas operações do dia a dia. O SmartStruxure Lite- engloba especificações de sistema menores, uma vez que é destinado àqueles que não precisam de um grande sistema de gerenciamento de edifícios. "Além da redução no consumo de energia e do controle mais eficiente dos sistemas prediais, o tempo de retorno do investimento com o SmartStruxure Lite é reduzido pelo fato destes possuírem a característica de comunicação wireless, ou seja, menos fios ou sem fios, diminuindo os custos de mão de obra de instalação, aceleração na entrega da obra e menor custo de investimento", explica José Antônio Torelli Bolota, gerente de produto da Schneider Electric.



As soluções gerenciam os sistemas de edifícios em uma só rede — em toda a empresa. Os softwares permitem a integração de dados externos à instalação, tais como informações meteorológicas e custos de utilidades, para garantir que, cada vez mais, o edifício esteja operando com a máxima eficiência. Outro diferencial é que conversa com sistemas pré-existentes, operando em uma plataforma aberta.

Desempenho adequado
com menor consumo

O software desenvolvido pelo engenheiro de computação Leonardo de Paula Rosa Piga, obteve resultados significativos na redução do consumo de energia elétrica por parte de Data Centers que oferecem serviços web. De acordo com o autor, o programa proporcionou entre 5% e 16% de economia de eletricidade, dependendo da eficiência energética do sistema considerado. Nos Estados Unidos, onde se concentram em maior número, os Centros de Dados representam 2% de todo o consumo de energia elétrica do país. Graças à importância do trabalho, Piga foi convidado para assumir o cargo de pesquisador na norte-americana AMD, uma das grandes fabricantes de processadores do mundo.

Atualmente o orientador, Sandro Rigo, e o autor do estudo estão iniciando conversações com a Agência de Inovação "Inova Unicamp" para analisar o registro intelectual do software. O professor explica que o programa é composto por um conjunto de algoritmos. Alguns são conhecidos, mas não tinham sido arranjados da forma como Piga o fez. "Além disso, há um método que ele desenvolveu, cuja técnica se baseia em 'dizer ao computador', por exemplo, para que ele permaneça num determinado estado de consumo, numa dada frequência e utilizando somente 70% da sua capacidade de processamento. Essa estratégia, que nos parece inovadora, talvez possa render um pedido de depósito de patente", explica o docente.

Usinas solares podem gerar empregos e
reduzir as emissões de gases que causam o efeito estufa

Caso já estivessem em operação, os 13 projetos contemplados pela Chamada Estratégica Nº 13, da Agência Nacional de Energia Elétrica (Aneel), que, objetiva fomentar a instalação de Usinas Solares Fotovoltaicas (USFs) no Brasil, teriam evitado a emissão de 6.285 toneladas de carbono equivalente (tCO2eq) em 2011 e de 11.229 (tCO2eq) no ano posterior, considerando-se a capacidade máxima de geração de energia elétrica das plantas.

Além disso, a iniciativa também teria gerado 454 empregos diretos com a instalação das USFs. As projeções fazem parte da tese de doutorado do engenheiro agrônomo Davi Gabriel Lopes, defendida recentemente na Faculdade de Engenharia Mecânica (FEM) da Unicamp, sob a orientação da professora Carla Kazue Nakao Cavaliero.

O autor da tese explica que antes do lançamento da Chamada 13, em 2011, o Brasil contava, praticamente, com painéis fotovoltaicos operando somente em sistemas isolados. Ou seja, os equipamentos, instalados em residências, por exemplo, não estavam interligados à rede de distribuição. A ação da Aneel pretendeu justamente estimular tanto as empresas públicas quanto privadas que atuam na geração, distribuição e transmissão de energia a investir na construção de USFs que pudessem ser conectadas à rede. Os recursos têm origem no montante que essas empresas são obrigadas a aplicar em programas de eficiência energética e em pesquisa e desenvolvimento (P&D).

Conforme Lopes, cerca de 100 projetos foram apresentados para participar da Chamada 13, mas somente 18 foram aprovados pela Aneel. Desses, entretanto, cinco desistiram de dar sequência às propostas. Dos 13 que sobraram, somente a CPFL Energia colocou a sua usina em operação. Inaugurada no final de dezembro de 2012, a unidade instalada na Subestação Tanquinho, em Campinas, exigiu investimentos da ordem de R$ 13 milhões, para produzir 1.090 MW. Participam do empreendimento, além da empresa proponente, pesquisadores da Unicamp – Lopes entre eles – e três empresas colaboradoras: Hytron, Eudora Solar e Instituto Aqua Genesis. Os demais projetos aprovados ainda estão em fase de planejamento ou os seus responsáveis estão providenciando licitações para a aquisição de equipamentos e contratação de serviços. Veja mais em: **www.unicamp.br/unicamp/ju/585/o-sol-como-aliado.**



Aprenda os conceitos básicos da amostragem analógica

Neste artigo, descrevemos os conceitos básicos sobre amostragem analógica, incluindo as definições e cálculos de largura de banda, taxa de amostragem, frequência de Nyquist, aliasing, filtros anti-aliasing, erro de quantização e dithering.

Guilherme Kenji Yamamoto
Renan Machado de Azevedo
Renato Araújo de Andrade
National Instruments

Largura de banda: definição e cálculos

Largura de banda é uma característica de um circuito ou canal de transmissão, definida pela faixa de frequência dentro da qual um sinal é transmitido por este circuito ou canal sem sofrer atenuação significativa. A largura de banda é medida entre os pontos de frequência inferior e superior, nos quais a amplitude do sinal cai 3 dB abaixo da frequência da banda de passagem. Os pontos de -3 dB são chamados de pontos de meia potência.

Unidades
Hertz (Hz)

Exemplo: Se você introduzir uma onda senoidal de 100 MHz com amplitude de 1 V em um digitalizador de alta velocidade que tenha uma largura de banda de 100 MHz, o sinal será atenuado pela entrada analógica do digitalizador e a forma de onda amostrada terá uma amplitude de aproximadamente 0,7 V. O valor de aproximadamente 0,7 V pode ser calculado pela fórmula abaixo:

$$-3\ dB = 20 \log (V_{ppout} / V_{ppin})$$

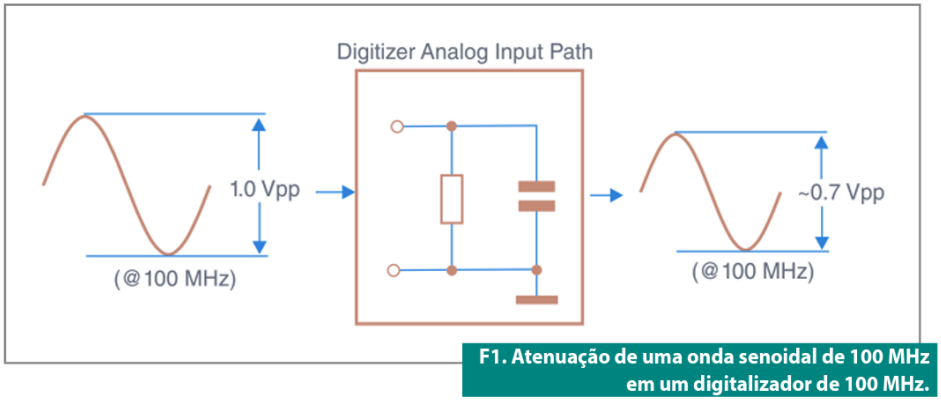
Onde:
Vppout = Tensão pico a pico da forma de onda da saída
Vppin = Tensão pico a pico da forma de onda na entrada = 1 V (no exemplo acima)

$$-3 = 20 \log (V_{ppout} / 1)$$
$$V_{ppout} = 0{,}7079\ V \cong 0{,}7\ V$$

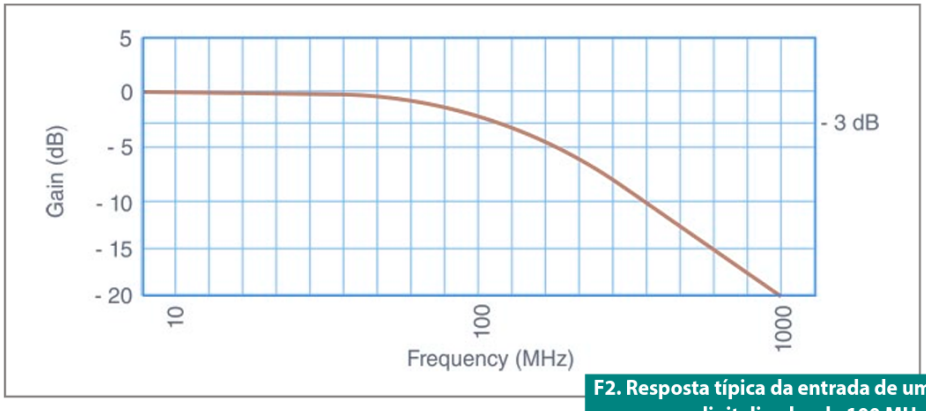
Veja as **figuras 1 e 2,** a seguir.

Erro de amplitude teórico de um sinal medido

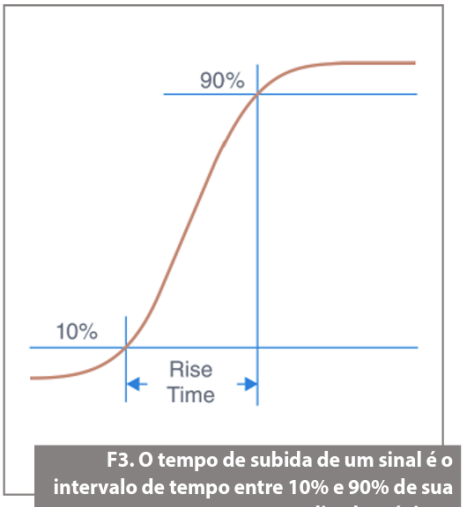
Para que tenhamos o menor erro possível de amplitude no sinal capturado, é recomendado que a largura de banda de seu digitalizador seja de 3 a 5 vezes a componente de maior frequência de interesse no sinal medido (largura de banda necessária = (3 a 5) x a frequência de interesse). O erro de amplitude teórico de um sinal medido pode ser calculado a partir da relação (R) entre a largura de banda do digitalizador (B) e a frequência do sinal de entrada (fin).



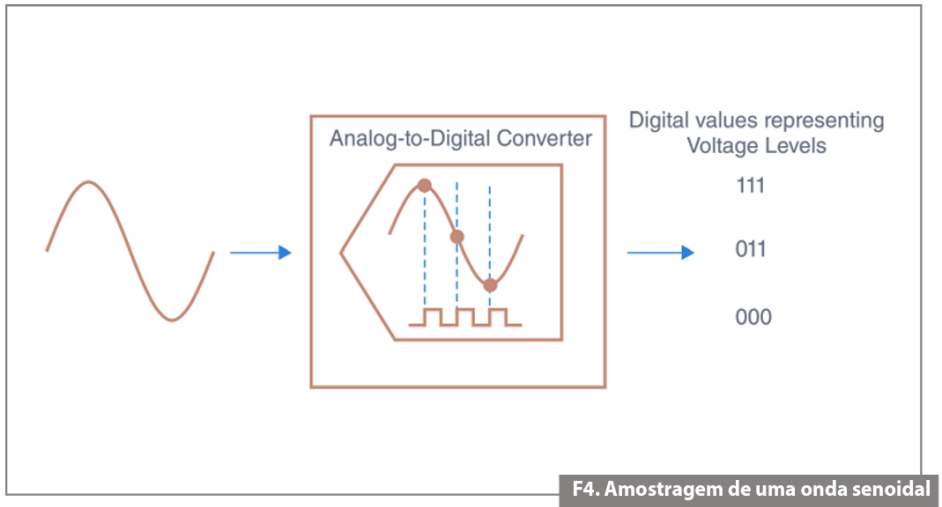
F1. Atenuação de uma onda senoidal de 100 MHz em um digitalizador de 100 MHz.



F2. Resposta típica da entrada de um digitalizador de 100 MHz.



F3. O tempo de subida de um sinal é o intervalo de tempo entre 10% e 90% de sua amplitude máxima.



F4. Amostragem de uma onda senoidal usando um digitalizador de 3 bits.

$$\text{Error}(\%) = \left(1 - \frac{R}{\sqrt{1+R^2}}\right) * 100$$

Onde:
R = B / fin

Conforme o cálculo mostrado, o erro em amplitude obtido quando medimos uma onda senoidal de 100 MHz com um digitalizador de alta velocidade de 100 MHz (o que nos dá uma relação R=1) será de aproximadamente 29,3%. No exemplo mostrado na **figura 1**, isso significa que se a forma de onda de entrada tiver uma amplitude de pico de 1 V, a forma de onda de saída teria uma amplitude pico a pico de aproximadamente 0,707 V. Em outro exemplo, se introduzirmos uma onda senoidal de 75 MHz em um osciloscópio da National Instruments que tenha uma largura de banda de 150 MHz, teremos R=2. Baseando-nos nesse cálculo, teríamos um erro teórico de amplitude de aproximadamente 10,6%.

Tempo de subida

Outro tópico importante relacionado à largura de banda é o tempo de subida. O tempo de subida de um sinal de entrada é o tempo que o sinal leva para passar de 10% a 90% da amplitude máxima do sinal. Esse parâmetro é inversamente proporcional à largura de banda. Observe a **figura 3**.

É recomendável que o tempo de subida na entrada do digitalizador seja de 1/3 a 1/5 do tempo de subida do sinal medido, para que possamos capturar o sinal com um mínimo de erro de tempo de subida. O tempo de subida teórico medido (Trm) pode ser calculado a partir do tempo de subida do digitalizador (Trd) e o tempo de subida real do sinal de entrada (Trs).

$$Tr_m = \sqrt{Tr_d^{\,2} + Tr_s^{\,2}}$$

Taxa de amostragem

A taxa de amostragem mostra a quantidade de dados coletados em um determinado intervalo de tempo. Ela **não** está diretamente relacionada à especificação de largura de banda do digitalizador de alta velocidade. A taxa de amostragem é a velocidade na qual o ADC do digitalizador converte o sinal analógico da entrada em valores digitais. Dessa forma, o digitalizador amostra o sinal após as etapas de atenuação, ganho, filtragem e outras similares terem sido aplicadas na entrada analógica, convertendo a forma de onda resultante em uma representação digital. A taxa de amostragem de um digitalizador de alta velocidade é baseada no *clock* de amostragem que controla os pontos no tempo em que o ADC faz a conversão dos valores analógicos instantâneos em valores digitais.

$$\frac{\textbf{Unidades}}{\textbf{Amostras/segundos (S/s)}}$$

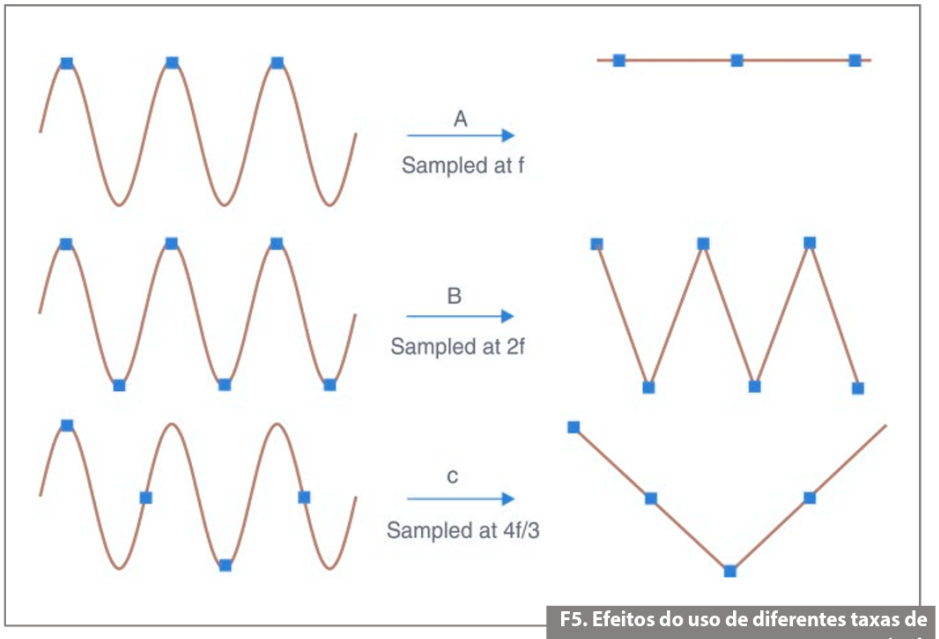
Exemplo: Os digitalizadores de alta velocidade da National Instruments podem oferecer uma taxa variável de amostragem, derivada da taxa máxima de amostragem do dispositivo. A taxa máxima de amostragem do dispositivo é determinada pela taxa de oscilação de seu oscilador a cristal (um componente do hardware do dispositivo). É possível obter, entretanto, taxas de amostragem menores, dividindo a taxa máxima de amostragem por um valor inteiro. Por exemplo, o digitalizador de alta velocidade NI 512 tem uma taxa máxima de amostragem de 200 MS/s. Dessa forma, ele pode ser configurado em taxas de 200/n MS/s, onde n = 1,2,3,4... Atente para a **figura 4**.

Teorema de Nyquist e frequência de Nyquist

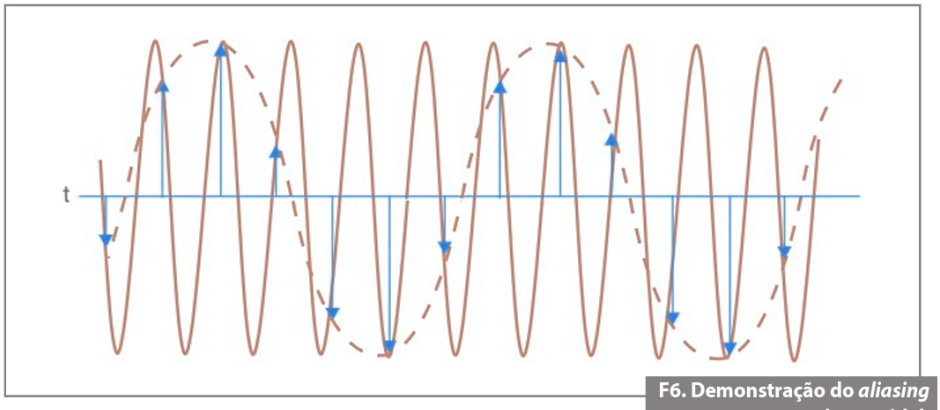
Teorema de Nyquist: Taxa de amostragem (fs) > 2 x componente da maior frequência de interesse do sinal medido. O teorema de Nyquist diz que um sinal deve ser amostrado a uma taxa maior que duas vezes a componente de maior frequência de interesse no sinal para que essa componente seja capturada. Caso isso não ocorra, o conteúdo de alta frequência será representado incorretamente em uma frequência que está dentro do espectro de interesse (banda de passagem). Veja **box 1**.

Qual é a taxa de amostragem que devo usar?

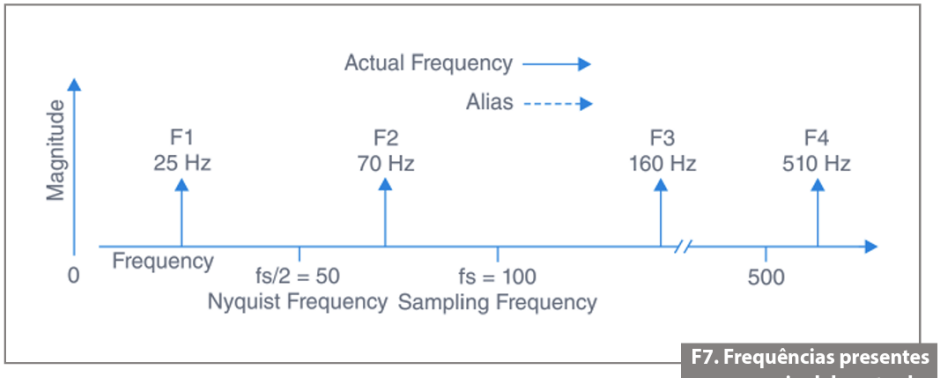
A **figura 5** mostra os efeitos das diversas taxas de amostragem. No caso A, a onda senoidal de frequência f é amostrada na mesma frequência f. A forma de onda reconstruída aparece como uma onda "alias" CC, de frequência igual a zero. Entretanto, se você elevar a taxa de amostragem a 2f, a forma de onda digitalizada terá a frequência correta (mesmo



F5. Efeitos do uso de diferentes taxas de amostragem em um sinal.



F6. Demonstração do *aliasing* com uma onda senoidal.



F7. Frequências presentes no sinal de entrada.

Box 1

A definição de frequência de Nyquist não é consistente no mundo da medição. Ela é, às vezes, usada no teorema para descrever a taxa de amostragem; outras vezes, para descrever a componente de frequência mais alta. Neste artigo, usaremos frequência de Nyquist para descrever a componente de maior frequência que podemos usar sem termos *aliasing* em uma dada frequência de amostragem.

número de ciclos), mas será exibida como uma forma de onda triangular. Neste caso (B), f será igual à frequência de Nyquist.

Aumentando a taxa de amostragem a um valor muito acima de f, por exemplo, 5f, você poderá reproduzir a forma de onda com exatidão. No caso C, a taxa de amostragem será 4f/3. A frequência de Nyquist, nesse caso, será (4f/3)/2 = 2f/3. Como f está acima da frequência de Nyquist, essa taxa de amostragem reproduzirá uma forma de onda com *aliasing*, de frequência e formato incorretos.

Aliasing e filtros anti-aliasing

Se um sinal for amostrado em uma taxa de amostragem menor que duas vezes a frequência de Nyquist, uma ou mais componentes de frequência mais baixa aparecerão nos dados amostrados. Esse fenômeno é denominado aliasing. A **figura 6** mostra uma onda senoidal de 5 MHz digitalizada por um ADC de 6 MS/s. A linha pontilhada indica o sinal com aliasing registrado pelo ADC. O efeito de aliasing mostra a frequência de 5 MHz incorretamente dentro da banda de passagem, representando-a como se fosse uma onda senoidal de 1 MHz.

Frequência de aliasing

A frequência de aliasing é o valor absoluto da diferença entre a frequência do sinal de entrada e o múltiplo inteiro mais próximo da taxa de amostragem.

Frequência de aliasing = ABS (múltiplo inteiro mais próximo da frequência de amostragem - frequência de entrada), na qual ABS significa valor absoluto. No mundo real, os sinais normalmente contêm componentes de frequência que estão muito acima da frequência de Nyquist. Pelo efeito de aliasing, essas frequências são incorretamente incluídas aos componentes do sinal amostrado corretamente, resultando em uma distorção dos dados amostrados.

Se em um sistema você quiser executar medições de alta exatidão usando dados amostrados, precisará usar uma taxa de amostragem suficientemente alta (de 5 a 10 vezes a componente de maior frequência no sinal) para evitar aliasing, ou introduzir um filtro anti-aliasing (filtro passa-baixas que atenua quaisquer fre-

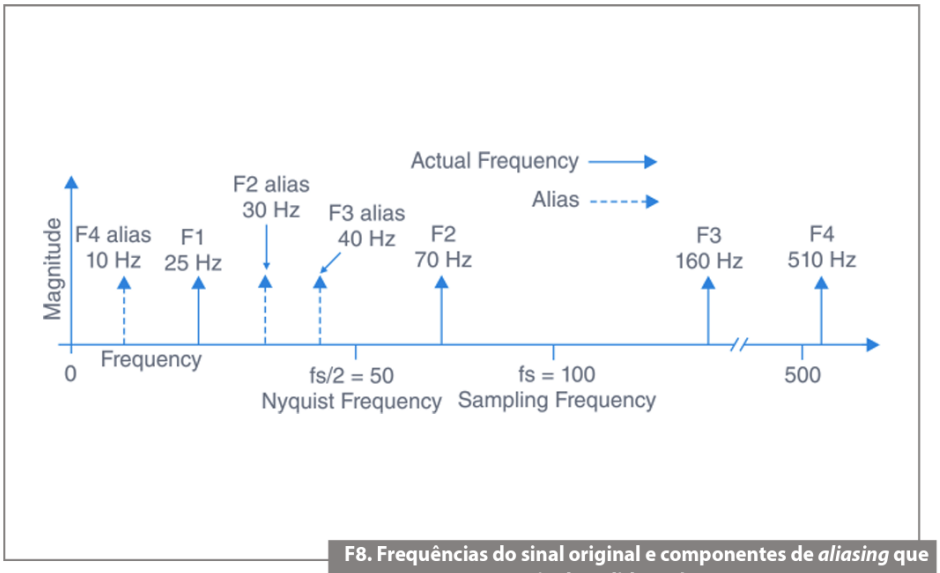
quências no sinal de entrada que sejam maiores que a frequência de Nyquist) antes do ADC, para restringir a largura de banda do sinal de entrada de forma a atender o critério de amostragem.

Por exemplo, no dispositivo de aquisição dinâmica de sinais NI 4461 são implementados filtros analógicos e digitais de hardware nas entradas analógicas, para evitar o aliasing. Os sinais de entrada são primeiramente passados por um filtro analógico fixo, para remover quaisquer componentes de frequência que estejam fora da faixa dos ADCs. Em seguida, filtros digitais anti-aliasing ajustam automaticamente suas frequências de corte de forma a remover quaisquer componentes de frequência que estejam acima da metade da taxa de amostragem programada. **Exemplo**: Assuma que fs, a frequência de amostragem, seja igual a 100 Hz e que o sinal de entrada contenha as seguintes frequências: 25 Hz, 70 Hz, 160 Hz e 510 Hz. Essas frequências são mostradas na **figura 7**.

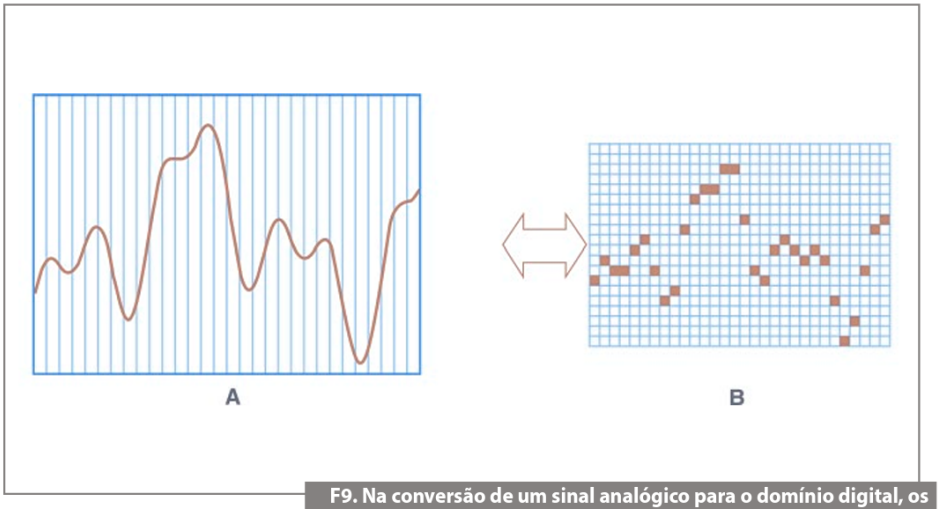
Partindo da **figura 8**, as frequências abaixo da frequência de Nyquist (fs/2 = 50 Hz) são amostradas corretamente. Frequências acima da frequência de Nyquist são mostradas como *aliases*. Por exemplo, F1 (25 Hz) é mostrada na frequência correta, mas F2 (70 Hz), F3 (160 Hz) e F4 (510 Hz) têm aliases em 30 Hz, 40 Hz e 10 Hz, respectivamente. Veja, agora, a **tabela 1**.

Erro de quantização

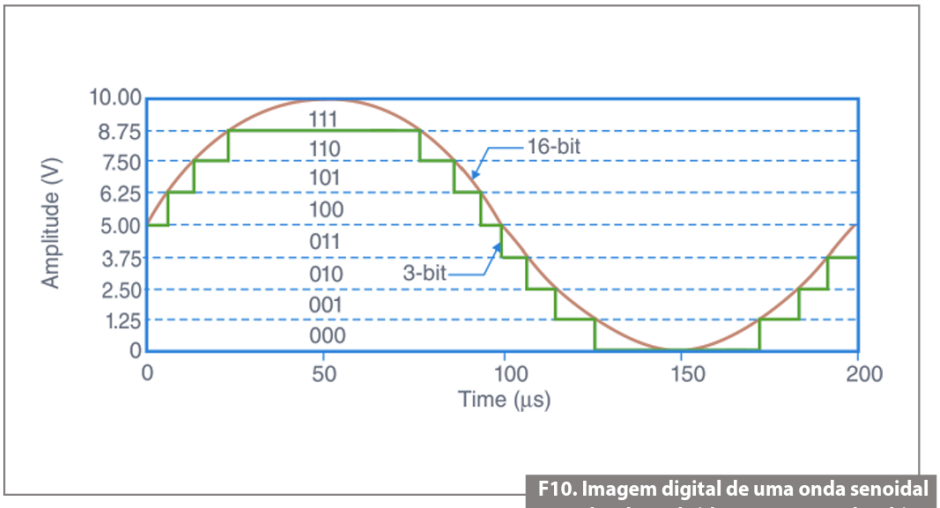
Quantização é o processo pelo qual um sinal analógico é convertido em uma representação digital. A quantização é realizada por um conversor analógico-digital (conversor A/D, ou ADC). Convertendo nossos sinais analógicos em uma sequência de dados digitais, podemos aproveitar a capacidade do computador pessoal e do software para tratar ou aplicar cálculos nos sinais. Para fazer isso, devemos amostrar nossa forma de onda analógica em momentos discretos bem definidos (mas limitados), de forma a mantermos uma relação bem próxima entre o tempo no domínio analógico e o tempo no domínio digital. Se fizermos isso, poderemos reconstruir o sinal no domínio digital, passá-lo por nosso processamento e, posteriormente, reconstruí-lo no domínio analógico, se necessário. Observe na **figura 9**.



F8. Frequências do sinal original e componentes de *aliasing* que aparecem no sinal medido após a passagem por um ADC.



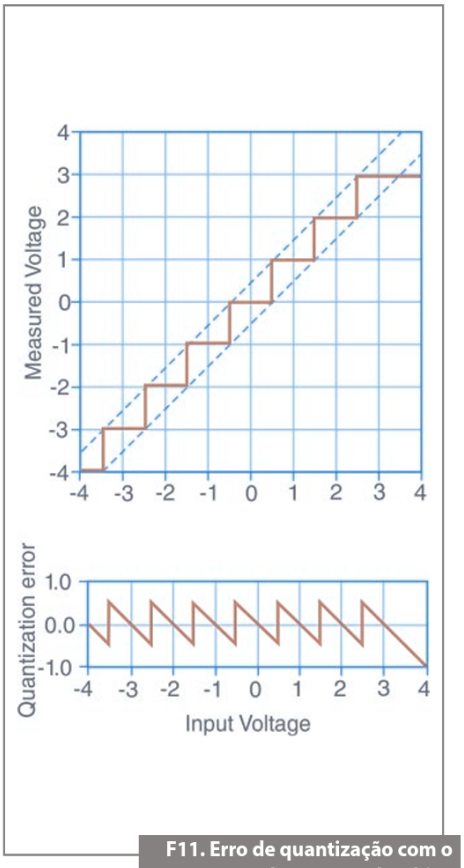
F9. Na conversão de um sinal analógico para o domínio digital, os valores do sinal são coletados em pontos discretos no tempo.



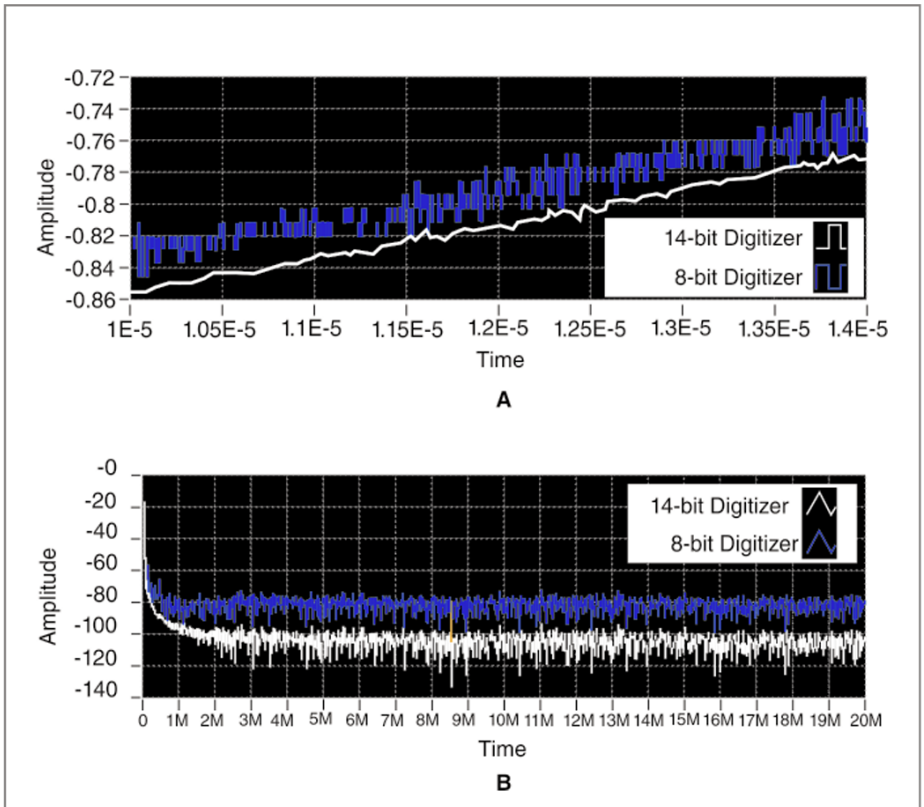
F10. Imagem digital de uma onda senoidal de 5 kHz obtida por um ADC de 3 bits.

Alias F2 =	100 – 70	= 30 Hz
Alias F3 =	(2)100 – 160	= 40 Hz
Alias F4 =	(5)100 – 510	= 10 Hz

F1. Frequência mostradas como "aliadas".



F11. Erro de quantização com o uso de um ADC de 3 bits.



F12. Medições com 8 bits e 14 bits.

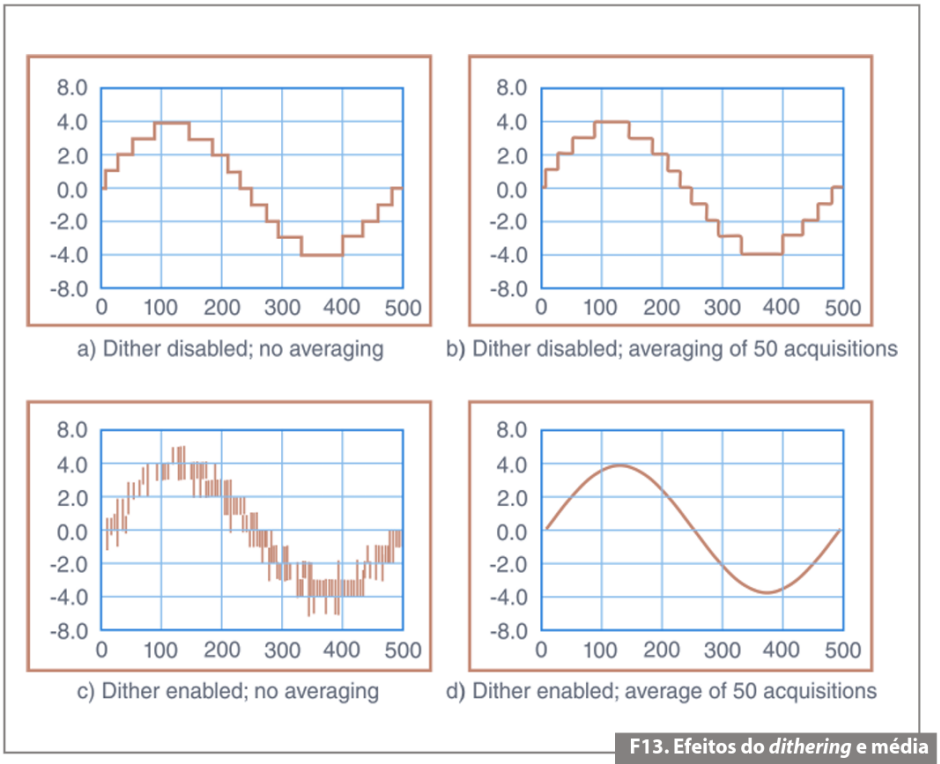
A resolução de tempo que teremos será limitada pela taxa de amostragem máxima do ADC. Mesmo se pudéssemos aumentar nossa taxa de amostragem indefinidamente, ela nunca seria totalmente um "tempo contínuo", como acontece com o sinal de entrada mostrado na **figura 9.** Mas para a maior parte das aplicações do mundo real, ela ainda será muito útil, apesar de sua natureza limitada. Obviamente, a utilidade de nossa representação digital será maior quanto maiores forem nossa resolução em tempo e amplitude. A resolução em amplitude é limitada pelo número de níveis de saída discretos que tiver o ADC. Por exemplo, um ADC de 3 bits divide a faixa em 2^3 (ou oito) divisões. Cada uma dessas divisões é representada por um código binário ou digital entre 000 e 111. O ADC traduz cada medição do sinal analógico em uma das divisões digitais. A **figura 10** mostra a imagem digital de uma onda senoidal de 5 kHz obtida por um ADC de 3 bits. Como constatado na **figura 11**, o sinal digital não representa adequadamente o sinal original, porque o conversor tem uma quantidade pequena demais de divisões digitais para representar as diversas tensões do sinal analógico. Entretanto, aumentando a resolução para 16 bits, para aumentar o número de divisões do ADC de oito (2^3) para 65.536 (2^{16}), teremos, com o ADC de 16 bits, uma representação extremamente exata do sinal analógico. Essa incerteza inerente à digitalização de um valor analógico é denominada erro de quantização. O erro de quantização depende do número de bits no conversor, juntamente com os seus erros, ruído e não linearidades.

A **figura 12** mostra como seria adquirir um sinal na faixa de entrada de 2,5 V usando um digitalizador de alta velocidade NI 5122 de 14 bits e um digitalizador de 8 bits NI 5112. Nesta figura, você pode ver a exatidão na representação de um sinal de entrada obtida com o digitalizador de 14 bits, com 16.384 pontos discretos de tensão, e compará-la com os 256 níveis obtidos com um digitalizador ou osciloscópio de 8 bits. Usando digitalizadores de alta resolução, você poderá também obter vários tipos de medições nos domínios do tempo e da frequência, usando um único instrumento. Esse gráfico mostra claramente as vantagens do uso de um digitalizador de alta resolução nas medições feitas nos domínios do tempo e frequência:

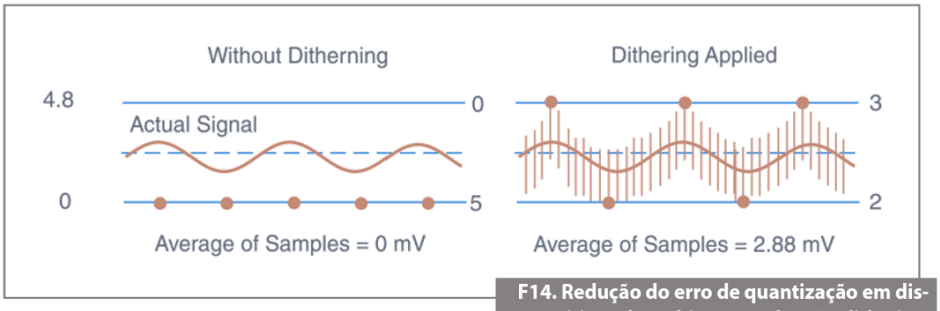
- 8 bits = 256 níveis discretos;
- 12 bits = 4.096 níveis discretos;
- 14 bits = 16.384 níveis discretos.

Dithering

Durante a quantização, no domínio do tempo, podemos preservar quase completamente as informações da forma de onda, fazendo uma amostragem suficientemente rápida. No domínio da amplitude, podemos preservar a maior parte das informações da forma de onda pelo uso da técnica denominada *dithering*. Dithering é uma inclusão intencional de ruído em nosso sinal de entrada. Essa função atenua pequenas diferenças na resolução de amplitude. O princípio básico dessa função é incluir ruído aleatório de forma a fazer o sinal subir e descer repetidamente entre níveis sucessivos. Obviamente, isso por si só apenas faria o sinal ter mais ruído. Mas esse ruído será suavizado pela função de cálculo do valor médio do sinal no formato digital.



F13. Efeitos do *dithering* e média em uma entrada senoidal.



F14. Redução do erro de quantização em dispositivos de 12 bits com a função *dithering*.

O cálculo matemático do valor médio dos sinais digitais sem o dithering não remove os degraus de quantização. Ele simplesmente os arredonda um pouco, como mostrado na **figura 13b**.

Exemplo: Em alguns produtos de aquisição de dados (DAQ) da série E da National Instruments, como o dispositivo DAQ multifuncional de alta exatidão da série M da National Instruments, o dithering é totalmente habilitado ou desabilitado por software (você não decide como a média é feita no instrumento). Quando habilitado, o software soma aproximadamente 0,5 LSB rms de ruído branco gaussiano no sinal de entrada. Esse ruído é incluído ao sinal antes da entrada no ADC. Como resultado, um sinal que está em algum ponto da menor diferença de tensão detectável pela placa (conhecida como largura de código) agora sobe e desce aleatoriamente até os limites desse código. Quando amostrados, os pontos agora serão exibidos nos limites superior e inferior, e as quantidades de pontos no limite superior ou inferior da largura de código serão proporcionais à localização do sinal. Você poderá então usar a média para aumentar a resolução especificada da placa e obter medições de maior exatidão, menos influenciadas pelo ruído de banda ampla. Por exemplo, quando o dithering estiver habilitado, uma placa de 12 bits poderá trabalhar com resolução de 14 bits. Você pode também desabilitar o dithering em aplicações de alta velocidade, que não utilizam a função de média. Atente para a **figura 14**.

Necessita Potência?
Pense GlobTek!

Carregadores inteligentes de baterias oferecem operação trifásica

Disponível em versões fornecendo 4,2 V, 8,4 V ou 12,6 V a 1 A para atender configurações de simples ou múltiplas baterias, a família GTM91128 de carregadores inteligentes de baterias Li-Íon da GlobTek opera com três modos de carga: condicionamento, corrente constante e voltagem constante. Os dispositivos de entrada universal possuem recurso de finalização de carga mínima com temporizador de backup totalmente carregado e LEDs indicadores do estado da carga. A família de carregadores inteligentes de baterias também vem com plugues intercambiáveis, simplificando



...para mais informações visite www.globtek.com



Comutadores aprovados para área médica com estrutura aberta fornecem até 240 W

Adequada para ampla variedade de aplicações médicas, ITE e PoE, a família GTM91110P240 de fontes de alimentação ca/cc de estrutura aberta da GlobTek fornece até 240 W de potência, ocupando espaço de 3 x 5 polegadas. Os dispositivos são fornecidos configurados em fábrica com saídas de 12 a 55 V (em incrementos de 0,1 V). Disponíveis em versões Classe I ou II, as fontes de alimentação de 1,75 polegada de altura apresentam eficiência de 85% com carga plena e incorporam recursos como PFC ativo, filtro EMI e uma saída de

...para mais informações visite www.globtek.com

Bateria recarregável fornece dados sobre estado da carga

Fornecendo capacidade inteligente de energia recarregável para avançados dispositivos portáteis e remotos, o conjunto BL3100C1865004S1PSQA de baterias Li-Íon da GlobTek incorpora funcionalidade de medição do estado para fornecer dados importantes sobre a carga. O conjunto de 14,4 V possui capacidade de 3.1 Ah e também incorpora um circuito de proteção. "Você não pode mais colocar uma bateria em



um dos produtos comuns atuais sem fornecer um meio para verificar o estado da carga, pois a vida útil é um aspecto crítico do desem

...para mais informações visite www.globtek.com

GlobTek® Inc.
"your power partner"

www.globtek.com

Interface gráfica para comunicação entre Arduino e PC

Há não muito tempo atrás, uma pessoa sem conhecimentos avançados em programação de baixo nível, Assembler, eletrônica e fundamentos de sistemas operacionais para PCs, não tinha a oportunidade de desenvolver seus próprios projetos.

Graças ao avanço das tecnologias e ao empenho de pessoas em forma de comunidades, ou até mesmo ao desenvolvimento de softwares profissionais que são distribuídos de maneira gratuita para uso doméstico, todo este mundo novo está ao alcance de estudantes, arquitetos, profissionais liberais, ou qualquer pessoa com conhecimentos básicos de programação.

Edriano C. De Araujo

Downloads

O Visual Studio Express:
www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx

Software do Arduino:
arduino.cc/en/Main/Software

Hardware Arduino no site oficial:
arduino.cc/en/Main/Products

No Brasil:
www.tato.ind.br/categoria/Arduino-%7B47%7D-Tatuino/

Levando-se em conta as características dos produtos atuais e as quase infinitas maneiras de se implementar um projeto pessoal, vamos desenvolver, neste artigo, uma interface gráfica que se comunicará com um hardware Arduino, o qual informará através de LEDs o *status* de quatro chaves e, ao mesmo tempo, se comunicará com um PC que, também, informará o status de forma gráfica.

Assim sendo, serão desenvolvidos os seguintes itens:

- Hardware e *firmware*.
- Interface gráfica.
- Protocolo de comunicação entre os dois.

Hardware: Para o desenvolvimento do projeto será utilizado um Arduino, que é uma plataforma de código aberto baseada em um hardware simples e fácil de usar. Focada para o uso de *hobistas*, artistas, *designers*, engenheiros ou quaisquer outras pessoas interessadas em aprender.

Interface gráfica: Para o desenvolvimento da Interface gráfica será utilizado o Visual Studio, da Microsoft, que, apesar de ser um software comercial, é distribuído gratuitamente em sua versão express para uso pessoal ou estudantil, proporcionando uma maneira fácil e barata de se aprender a trabalhar com a ferramenta.

Protocolo de comunicação: Um protocolo de comunicação pode ser definido de maneira geral como o conjunto de regras que permite a comunicação entre dois (ou mais) equipamentos ou pessoas. Existem diversos tipos de protocolos disponíveis atualmente, que podem ser abertos, proprietários, simples ou complexos. Alguns exemplos de protocolos industriais bem disseminados são ModBus, ProfiBus, MDB "multidrop bus" muito utilizado em máquinas de autosserviço.

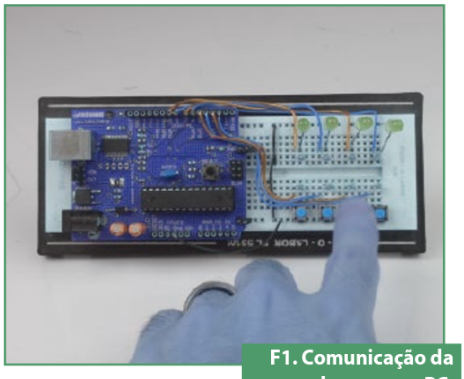
Neste artigo será desenvolvido um protocolo "proprietário" simples e de fácil entendimento.

Visão geral do projeto

O projeto desenvolvido neste artigo funcionará em mão dupla, como pode ser visto nas **figuras 1, 2, 3 e 4** a seguir.

Se um botão for pressionado na placa montada, o LED correspondente acenderá e o seu correspondente na Interface gráfica também.

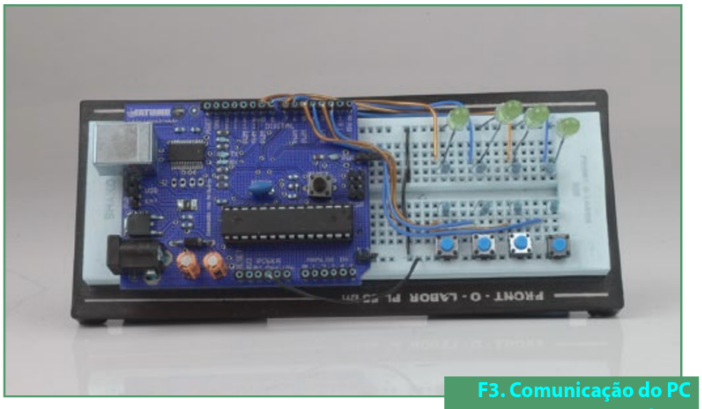
Em contrapartida, se o botão correspondente da interface gráfica for pressionado, o LED correspondente na placa acenderá.



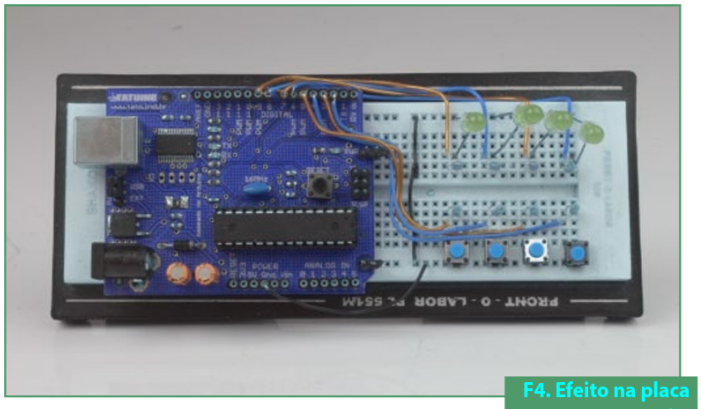
F1. Comunicação da placa para o PC.



F2. Efeito na interface gráfica.



F3. Comunicação do PC para a placa.



F4. Efeito na placa montada.

Mãos à obra

Todo o desenvolvimento, tanto do projeto no Arduino quanto no C#, será feito simultaneamente, de tal forma que, ao termino do artigo, o leitor será capaz de reproduzi-lo em sua própria placa e fazer modificações para deixar o projeto com o seu toque pessoal. Veja o **box downloads**.

Com todos os softwares instalados, já é possível iniciar o desenvolvimento do projeto. Não serão abordadas neste momento as informações básicas sobre o Arduino como sua história, conceitos básicos, instalação, entre outras, devido ao fato de que estas já foram previamente tratadas em edições anteriores.

Iniciaremos pelo desenvolvimento do firmware básico para o Arduino, incluindo as rotinas de leitura do teclado, acendimento dos LEDs e configuração da serial.

Abra o Arduino e crie um novo Sketch, incluindo as duas rotinas básicas do Arduino, setup e loop. Atente para a **figura 5**.

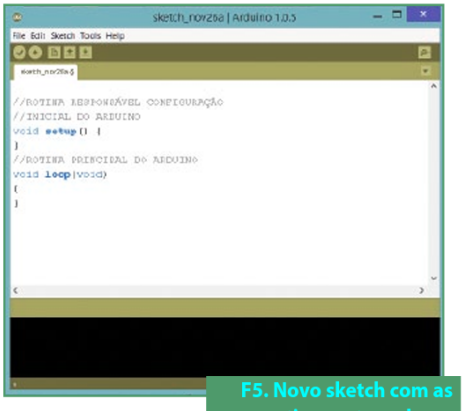
Criar as definições que serão utilizadas na configurações das portas de entrada e saída. Para o projeto, os LEDs serão ligados nas portas digitais 6, 7, 8 e 9. As chaves serão ligadas nas portas 2, 3, 4 e 5.

Configurar as portas ligadas ao LED como saída e as portas ligadas às chaves como entrada. Veja nas **figuras 6 e 7.**

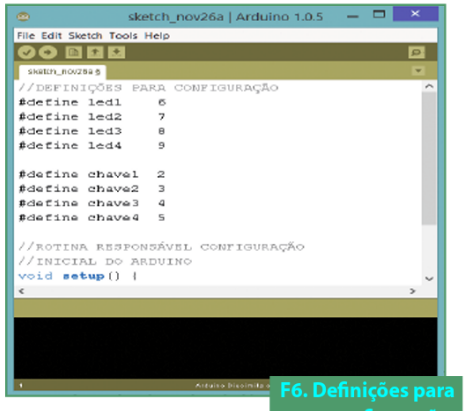
Criar uma rotina que será responsável por ler o estado das chaves. Esta rotina será chamada pela rotina principal.

Na rotina principal será inserida a chamada desta rotina. Acompanhe nas **figuras 8 e 9**.

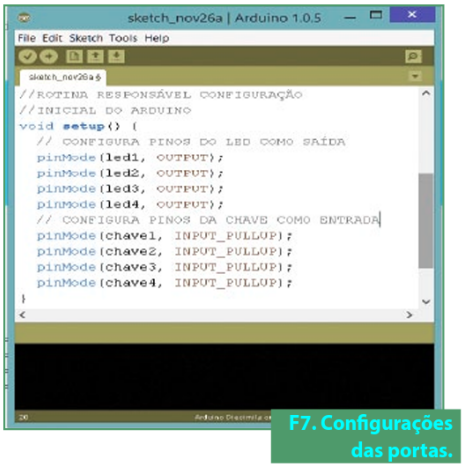
Criar a rotina responsável pelo acionamento dos LEDs. Esta rotina será invocada sempre que o estado atual das chaves for diferente de seu estado anterior. Como se pode ver, na **figura 10**, dependendo do



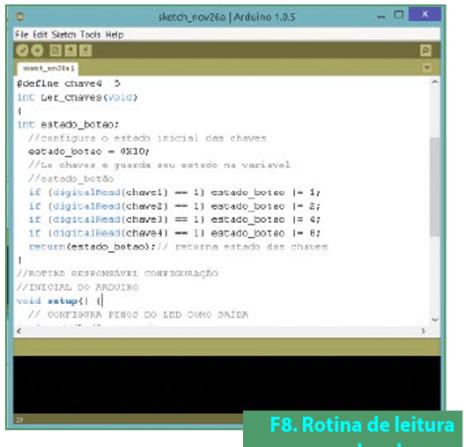
F5. Novo sketch com as rotinas setup e loop.



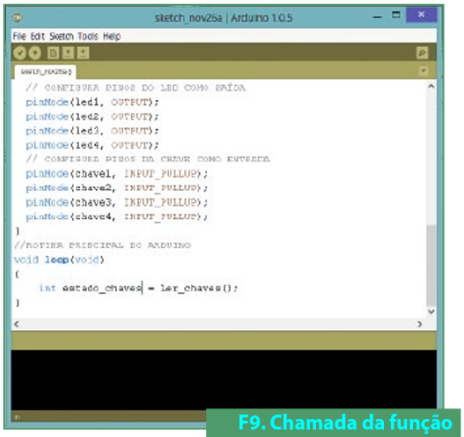
F6. Definições para configuração.



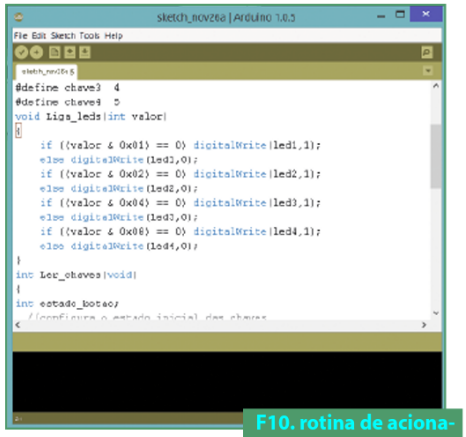
F7. Configurações das portas.



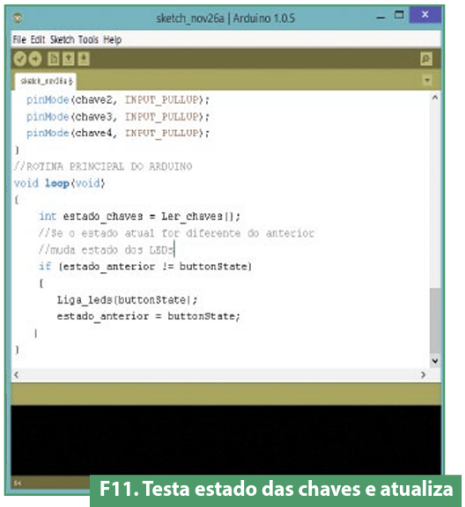
F8. Rotina de leitura das chaves.



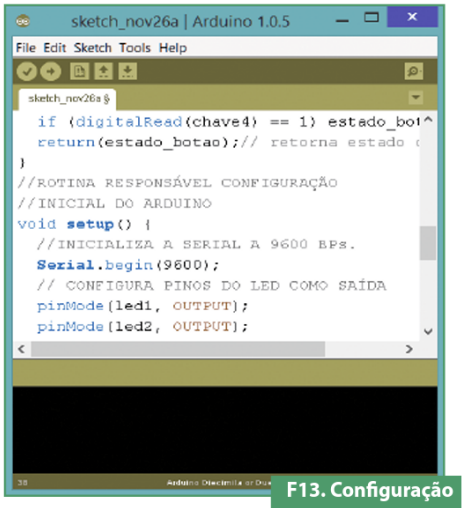
F9. Chamada da função ler chaves.



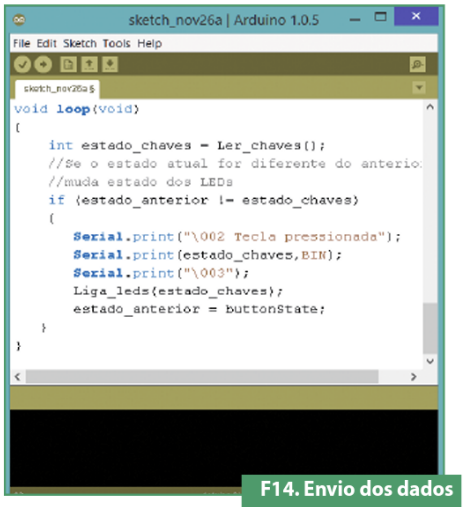
F10. rotina de acionamento dos LEDs.



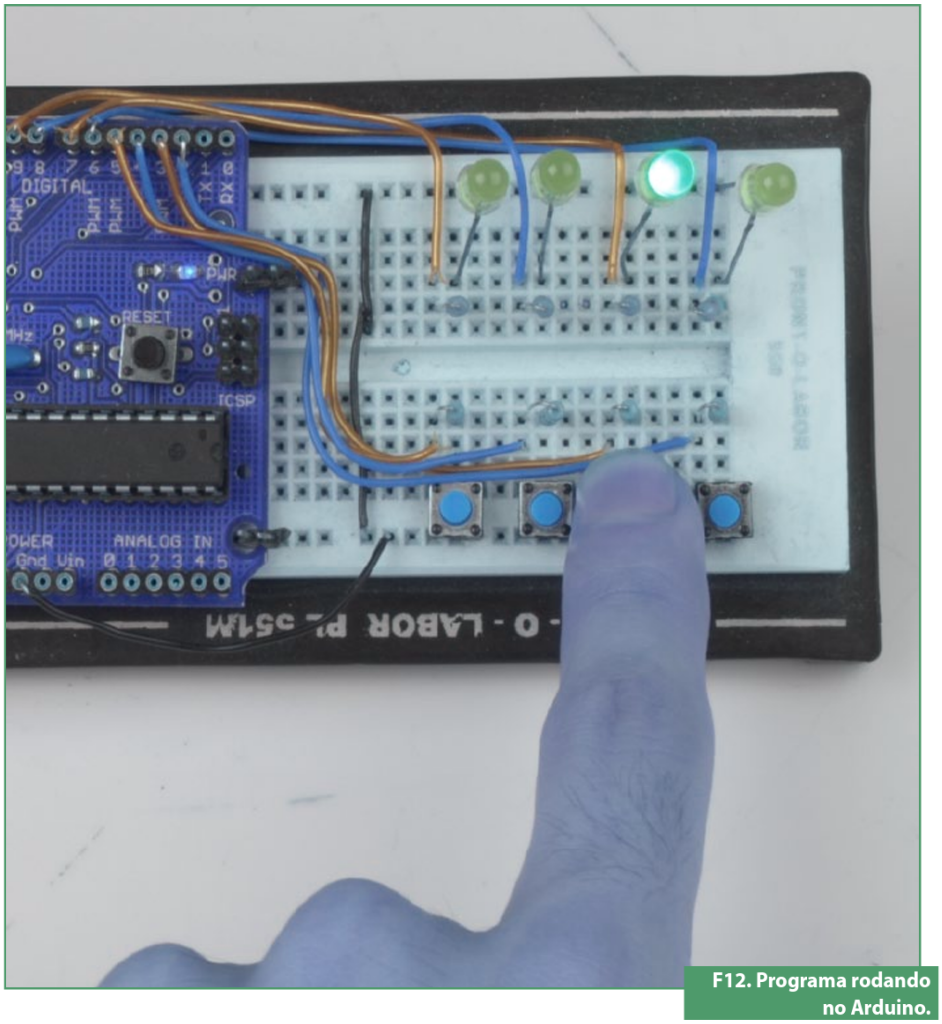
F11. Testa estado das chaves e atualiza LEDs, se necessário.



F13. Configuração da serial.



F14. Envio dos dados para o PC.



F12. Programa rodando no Arduino.

estado das chaves, o LED correspondente à chave pressionada será acionado. Observe também a **figura 11**.

Pronto, agora compilar e gravar o firmware no Arduino. E a exemplo da **figura 12**, o programa estará totalmente funcional.

Com o firmware básico operacional, será iniciado o processo de construção das rotinas de comunicação entre o Arduino e o PC.

Toda comunicação entre dois equipamentos deve ser baseada em regras claras e bem definidas para que não haja erro na codificação (ou decodificação) dos comandos, desta forma, para que a placa Arduino possa se comunicar com o PC. Um protocolo deve ser especificado.

Neste caso, será utilizado um protocolo simples e de fácil entendimento. Veja a **tabela 1**.

O funcionamento básico se dá de tal maneira que, ao se enviar um comando, as partes saibam exatamente onde o pacote enviado começa e termina.

STX: "start of text" caracter 0x02.

Payload: Pacote enviado, nele estarão presentes o comando e os dados.

ETX: "End of text" caracter 0x03.

Exemplo: Tecla pressionada

Comando para envio do estado das chaves, quando uma chave é pressionada. Acompanhe na **tabela 2**.

O comando enviado pelo PC e recebido pela placa Arduino será discutido depois.

Uma vez definido o Protocolo e sua estrutura básica, será inserido no firmware do Arduino as rotinas referentes a comunicação serial.

Protocolo		
STX	Dados ou "Payload" em ASCII	ETX

T1. Especificação do protocolo.

0x02	Tecla pressionada	1	0	0	0	0	0x03
STX	Comando em ascii	1	Estado chave 1	Estado chave 2	Estado chave 3	Estado chave 4	ETX

T2. Funcionamento básico.

Primeiramente, configurar a serial. Esta configuração deve ser feita na rotina setup. Atente para a **figura 13**.

Mantendo o intuito do Arduino, configurar e utilizar a serial deve ser uma tarefa fácil.

Desta forma, basta inserir a linha de código Serial.begin(9600) na rotina setup para que a serial do Arduino esteja pronta para uso.

Próximo passo, enviar o estado da serial para o PC através da serial. Observe a **figura 14**.

Para que o dado seja enviado ao PC, foram inseridas as seguintes linhas de código: **Serial.print("\002 Tecla pressionada");**

Note que \002 instrui o compilador a inserir o caracter 0x002 no início da transmissão e logo em seguida o comando Tecla pressionada, definido anteriormente: **Serial.print(estado_chaves,BIN);**

Nesta linha o estado das chaves será enviado para o PC formatado em binário: **Serial.print("\003");**

Nesta linha o caracter 0x03 ETX é enviado.

Compilar o programa gravar e verificar seu funcionamento. Se tudo correr de acordo, o programa estará totalmente funcional. Repita a ação demostrada na figura 12.

Para que o leitor possa testar as rotinas de comunicação desenvolvidas no Arduino, o seu desenvolvimento para neste ponto e o da Interface gráfica se inicia.

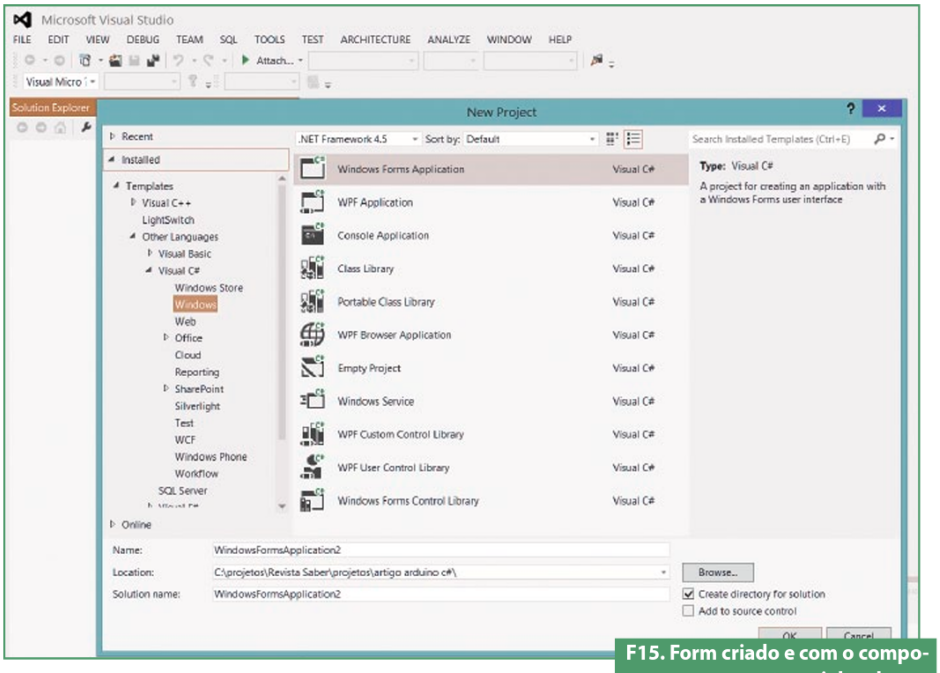
O próximo passo será começar o desenvolvimento da interface gráfica para o PC. Para isso, abra o Visual Studio express ou Visual studio. Vá em ***file/new/project/windows forms application***, conforme mostra a **figura 15**.

Neste ponto será apresentada uma tela similar à da **figura 16**, com um Windows form em branco, porém, totalmente funcional, podendo até mesmo ser executado.

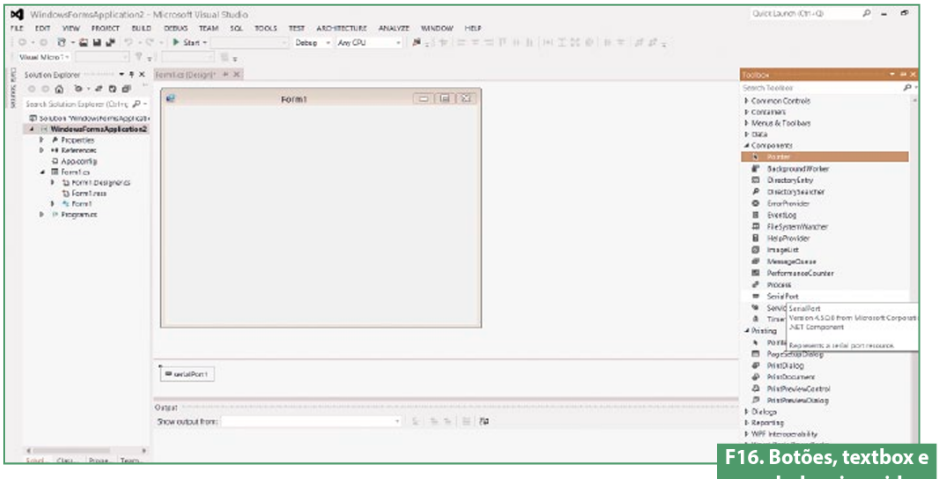
Da mesma forma que no Arduino, utilizar a serial no C# é fácil e indolor.

Vá em Toolbox/Componentes/SerialPort. E arraste o mesmo para dentro do form.

Da mesma forma que Componente SerialPort foi inserido no Form, colocar 9 Buttons (botões) 1 TextBox e 1 ComboBox no Form, resultando na **figura 17**.



F15. Form criado e com o componente serial no lugar.



F16. Botões, textbox e combobox inseridos.

Após colocados todos os componentes, será dado início à configuração do programa.

As linhas de código a seguir terão a função de varrer as portas seriais disponíveis e disponibilizá-las no combobox, proporcionando ao usuário, selecionar, a porta correspondente a sua placa Arduino.

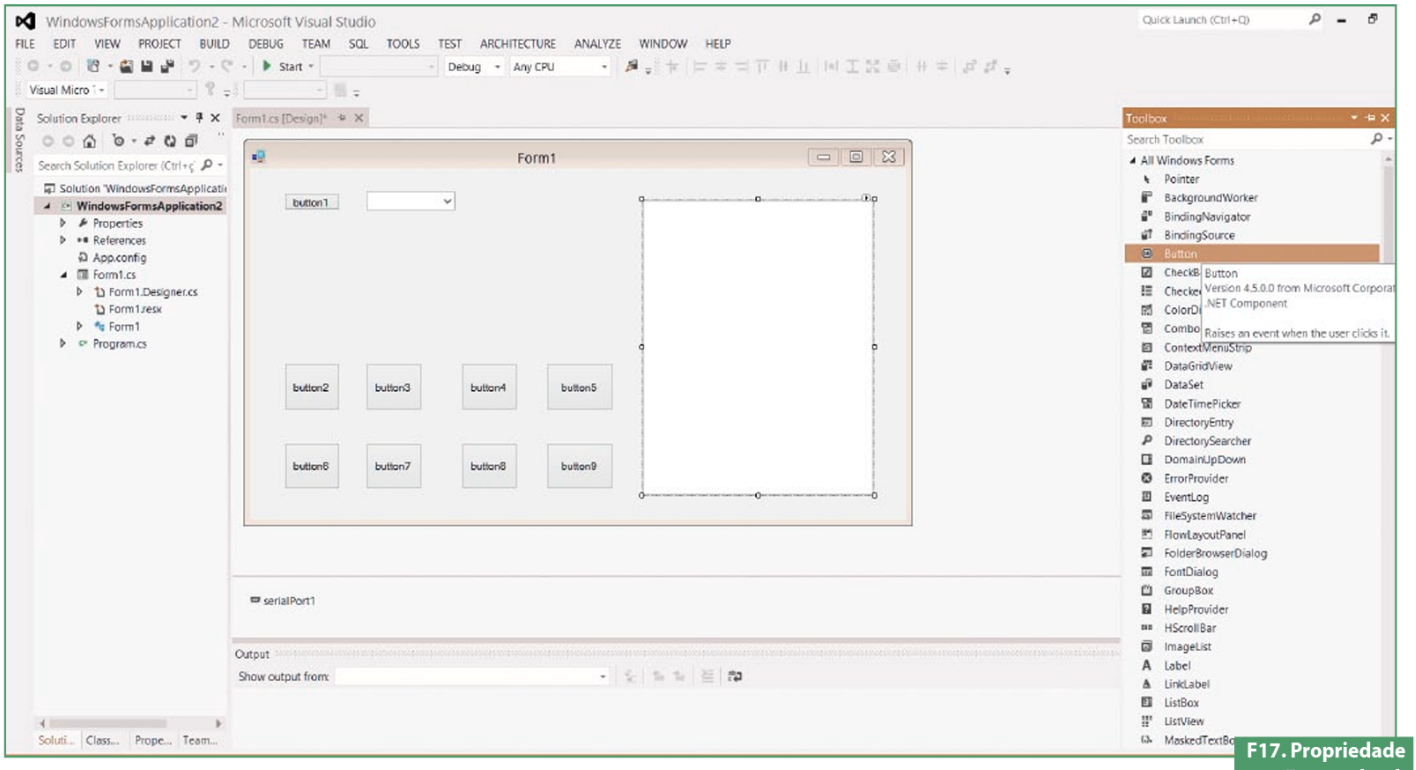
Clique no Form, vá em properties e procure pela propriedade Load e dê dois cliques na propriedade Form1_load conforme indica a **figura 18**. O seguinte código será apresentado.

```
private void Form1_Load(object sender, EventArgs e)
{
}
```

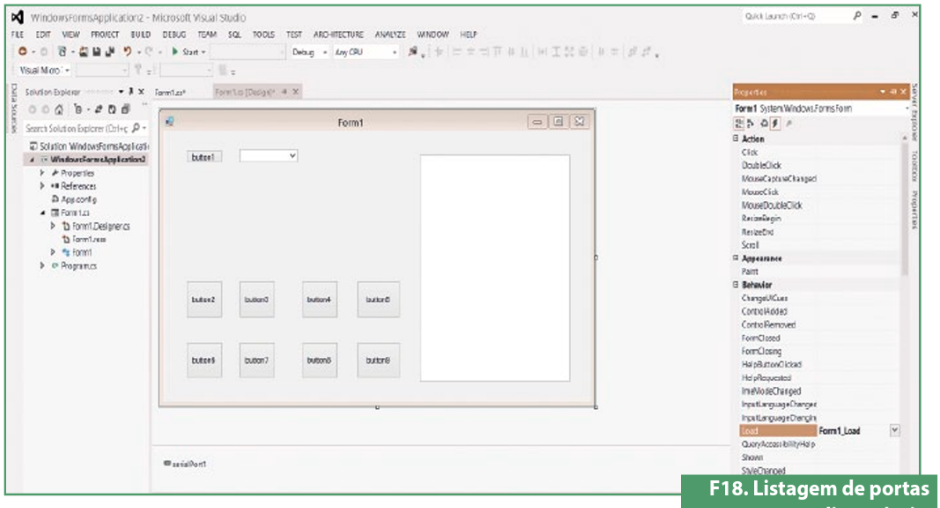
Insira as linhas de código a seguir. Estas linhas serão responsáveis por preencher o comboBox com as portas disponíveis no momento.

```
private void Form1_Load(object sender, EventArgs e)
{
this.comboBox1.Items.Clear();
string[] thePortNames = System.IO.Ports.SerialPort.
GetPortNames();
foreach (string item in thePortNames)
{
this.comboBox1.Items.Add(item);
}
}
```

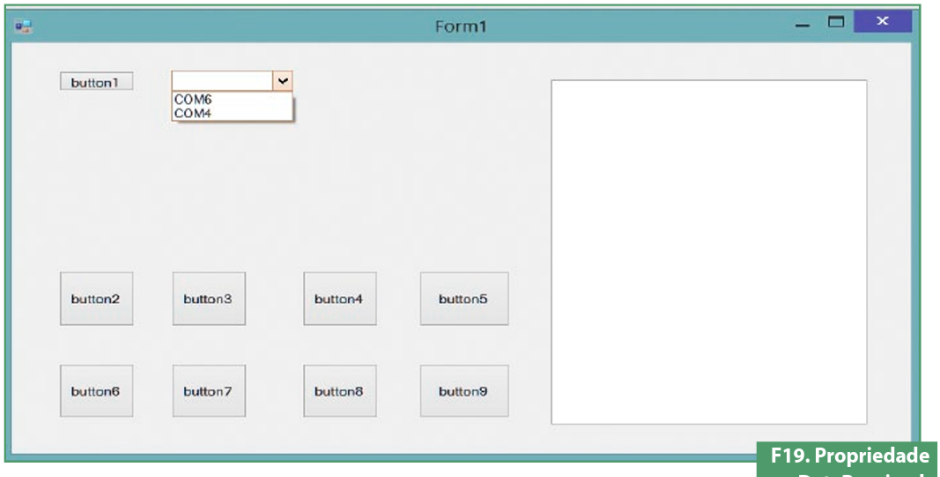
Execute o programa. O resultado será similar ao mostrado na **figura 19**.



F17. Propriedade Form_1 load.



F18. Listagem de portas disponíveis.



F19. Propriedade DataReceived.

No momento em que este artigo estava sendo escrito, duas portas estavam disponíveis no computador: a placa Arduino Com4 e um conversor USB-Serial Com6.

Em continuação, será dada ao botão 1 a função de abrir ou fechar a comunicação serial. Para tal, termine o programa e volte a edição do form. Clique duas vezes em cima do botão e insira o código a seguir.

```
private void button1_Click(object sender,
EventArgs e)
{
if (serialPort1.IsOpen)
{
serialPort1.DiscardInBuffer();
serial = 0;
serialPort1.Close();
button1.Text = "Conectar";

}
else
{
serialPort1.PortName = comboBox1.Text;
serial = 1;
serialPort1.Open();
button1.Text = "Desconectar";
}
}
```

Neste ponto, já será possível selecionar a porta e iniciar o desenvolvimento da rotina de recepção que será responsá-

vel por receber os pacotes enviados pela placa Arduino e decodificá-los.

Para isto, volte à edição do Form e selecione o componente SerialPort. Vá em properties e encontre a propriedade DataReceived. Dê um clique duplo e insira o código fornecido. Acompanhe na **figura 20**.

Ao receber dados através da serial, um evento é gerado e a quantidade de *bytes* disponíveis no buffer informada pela propriedade **serialPort1.BytesToRead**.

```
private void serialPort1_DataReceived(object
sender, System.IO.Ports.
SerialDataReceivedEventArgs e)

{
int teste, qtd;
teste = 0;
qtd = serialPort1.BytesToRead;
do
{
teste = serialPort1.ReadChar(); //quantidade de
bytes no buffer
--qtd;
switch (teste)
{
case 0x02://testa se recebeu um STX
recebe = "";
break;
case 0x03://testa se um ETX foi recebido.

//Em caso afirmativo, a rotina que decodifica os
dados é chamada.

this.Invoke(new EventHandler(Mostra_display));
break;
default:
//adiciona um novo dado a string recebida.
recebe += Convert.ToChar(teste);
break;
}

}
while (qtd != 0); //Enquanto existirem dados no
buffer, continua a leitura.
}
```

Rotina responsável por decodificar o pacote recebido e mudar a cor dos botões que representam os LEDs. Observe que para manter a compatibilidade com o protocolo especificado anteriormente, a primeira ação executada pela função é checar se o comando Tecla pressionada foi recebido. E somente se o comando foi recebido a decodificação das chaves é executada.

```
private void Mostra_display(object sender,
EventArgs e)
{
textBox1.AppendText(recebe + Convert.
ToChar(0X0D) + Convert.ToChar(0X0A));
if (recebe.Contains("Tecla pressionada"))// verifica
se o comando foi recebido corretamente.
{
// Se a tecla zero foi pressionada a cor do botão
muda para verde escuro, caso contrário
// se mantém em verde claro. O mesmo é válido
para as outras chaves.
if (recebe[19] == '0') button2.BackColor = Color.
ForestGreen;
else button2.BackColor = Color.LightGreen;
if (recebe[20] == '0') button3.BackColor = Color.
ForestGreen;
else button3.BackColor = Color.LightGreen;
if (recebe[21] == '0') button4.BackColor = Color.
ForestGreen;
else button4.BackColor = Color.LightGreen;
if (recebe[22] == '0') button5.BackColor = Color.
ForestGreen;
else button5.BackColor = Color.LightGreen;
}
}
```

Neste ponto, já é possível executar o programa e verificar que ao se pressionar uma chave na placa Arduino o botão referente ao respectivo LED irá mudar de cor. Porém, como se pode notar, os botões não ficam automaticamente redondos, visto no exemplo mostrado anteriormente, na figura 2.

Para que isso ocorra, será necessário mudar o estilo do botão em tempo de execução. Então, clique duas vezes no form. E na classe principal insira o código a seguir:

```
public partial class Form1 : Form
{
int serial;
string recebe;
string chaves = "chaves 0000";
public Form1()
{
InitializeComponent();
}

protected override void OnPaint
(PaintEventArgs e)

{
GraphicsPath forma = new GraphicsPath();

forma.AddEllipse(0, 0, button2.Width, button2.
Height);

button2.Region = new Region(forma);
button3.Region = new Region(forma);
button4.Region = new Region(forma);
button5.Region = new Region(forma);

}
```

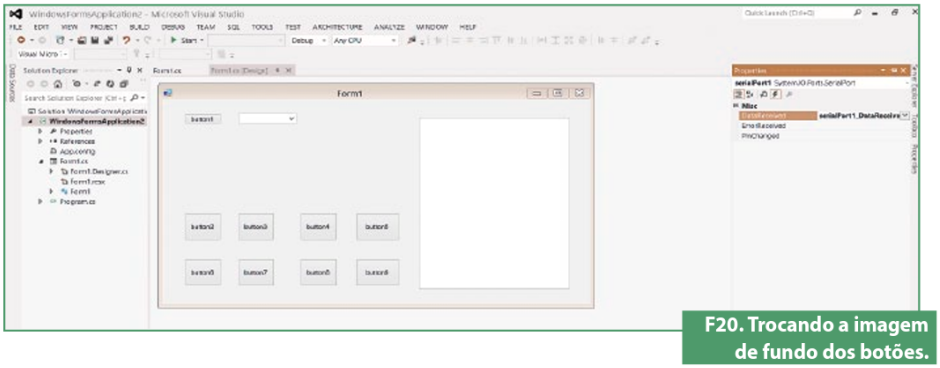
Agora, ao se executar o programa, os botões que simulam os LEDs, serão redondos.

Quanto às figuras, para que os botões tenham a figura de chave ou qualquer outra desejada, procure nas propriedades do botão a propriedade Image e insira uma imagem a seu gosto. Repita isto para todos os botões. Procedimento exibido na **figura 21**.

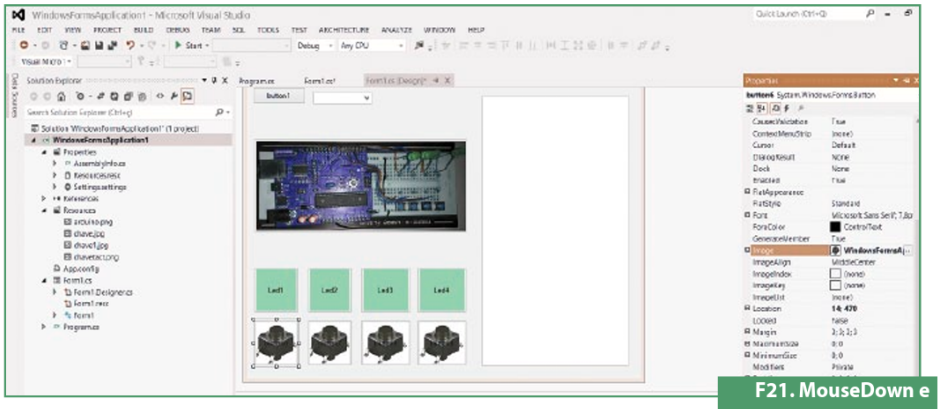
Após todas as modificações, executar o programa e testar. Agora que o PC está mostrando o estado dos botões pressionados na placa Arduino, o inverso deve ser desenvolvido. Toda vez que um botão que representa uma chave for pressionado, o LED na placa Arduino deve acender e ao ser solto o LED deve apagar.

Para que isto seja possível, as propriedades MouseUp e MouseDown serão utilizadas. Para acessá-las procure as duas propriedades, conforme ilustra a **figura 22**.

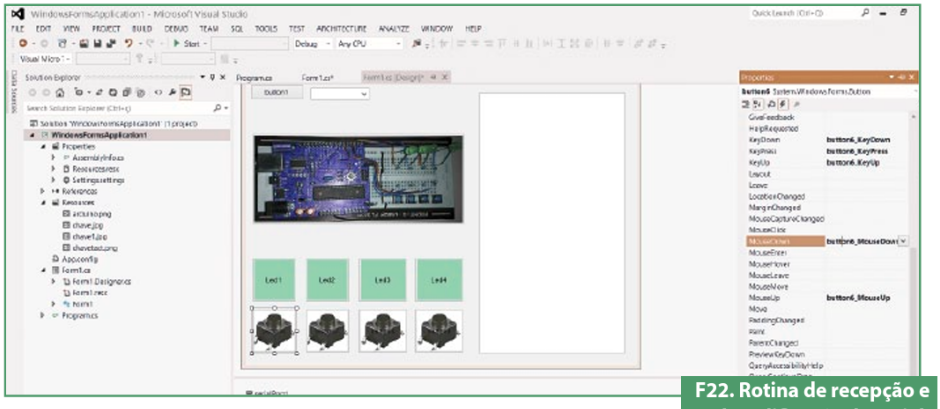
De um duplo clique na propriedade desejada e insira o código fornecido. Repita este procedimento para todos os botões.



F20. Trocando a imagem de fundo dos botões.



F21. MouseDown e MouseUp.



F22. Rotina de recepção e decodificação da serial.



F23. Chamada da rotina Testa_serial.

```
private void button6_MouseDown(object sender,
MouseEventArgs e){
//"chaves 1111"; se a tecla for precionada o estado
da chave muda de 1 para 0
button2.BackColor = Color.ForestGreen;
chaves = chaves.Remove(10, 1).Insert(10,"1");
//para enviar o dado pela serial basta usar Serial-
Port1.Write.
serialPort1.Write(Convert.ToChar(2) + chaves +
Convert.ToChar(3));
}
private void button6_MouseUp(object sender,
MouseEventArgs e)
{
button2.BackColor = Color.LightGreen;
//s= s.Remove(3, 2).Insert(3,"ZX");
chaves = chaves.Remove(10, 1).Insert(10,"0");
serialPort1.Write(Convert.ToChar(2) + chaves +
Convert.ToChar(3));
}
private void button7_MouseDown(object sender,
MouseEventArgs e)
{
button3.BackColor = Color.ForestGreen;
chaves = chaves.Remove(9, 1).Insert(9,"1");
serialPort1.Write(Convert.ToChar(2) + chaves +
Convert.ToChar(3));
}
private void button7_MouseUp(object sender,
MouseEventArgs e)
{
button3.BackColor = Color.LightGreen;
chaves = chaves.Remove(9, 1).Insert(9,"0");
serialPort1.Write(Convert.ToChar(2) + chaves +
Convert.ToChar(3));
}
private void button8_MouseDown(object sender,
MouseEventArgs e)
{
button4.BackColor = Color.ForestGreen;
chaves = chaves.Remove(8, 1).Insert(8,"1");
serialPort1.Write(Convert.ToChar(2) + chaves +
Convert.ToChar(3));
}
private void button8_MouseUp(object sender,
MouseEventArgs e)
{
button4.BackColor = Color.LightGreen;
chaves = chaves.Remove(8, 1).Insert(8,"0");
serialPort1.Write(Convert.ToChar(2) + chaves +
Convert.ToChar(3));
}
private void button9_MouseUp(object sender,
MouseEventArgs e)
{button5.BackColor = Color.LightGreen;
chaves = chaves.Remove(7, 1).Insert(7,"0");
serialPort1.Write(Convert.ToChar(2) + chaves +
Convert.ToChar(3));
}
private void button9_MouseDown(object sender,
MouseEventArgs e){
button5.BackColor = Color.ForestGreen;
chaves = chaves.Remove(7, 1).Insert(7,"1");
serialPort1.Write(Convert.ToChar(2) + chaves +
Convert.ToChar(3));
}
```

Para testar se esta parte do programa está funcionando, devemos voltar ao desenvolvimento do Arduino. A função de recepção deve ser implementada nele para que os LEDs acendam no momento em que o botão que representa a chave é pressionado.

A grande vantagem de se trabalhar com o Arduino é a de que não se faz acessar diretamente o hardware. Desta forma, como pode ser notado na **figura 23**, as rotinas de recepção e decodificação dos dados recebidos são praticamente idênticas as desenvolvidas em C#. Perceba que:

- A rotina de recepção tem a mesma estrutura da desenvolvida em C#.
- Os dados são decodificados da mesma maneira, sendo que a única diferença é a de que a decodificação está sendo feita na mesma rotina que trata a serial.
- Ao receber o estado das chaves, o Arduino retorna para o PC a quantidade de dados recebidos, desta forma o usuário pode confirmar que a comunicação está funcionando.
- Serial.print("\002 Recebi : ");
- Serial.print(contador_bytes, DEC);
- Serial.print("\003");

A última ação a ser tomada será inserir a chamada da rotina inicia serial na rotina loop.

Após terminado todo o desenvolvimento, compilar e executar os dois programas. Neste ponto será possível observar o funcionamento de ambos.

Conclusão

Utilizando o Arduino em conjunto com uma ferramenta como o Visual Studio, é possível desenvolver projetos de boa qualidade em um espaço reduzido de tempo.

Vale salientar que o Visual Studio é uma ferramenta muito poderosa e, nem de longe, utilizamos o que ela é capaz de fornecer. Todavia, tomando este artigo como base, o leitor pode modificar e desenvolver aplicações muito mais avançadas.

Da mesma forma, o Arduino vem avançando muito e cada vez mais inserindo novas funcionalidades em sua plataforma, proporcionando experiências cada vez mais prazerosas.



Pense Suporte Vertical para Baterias do tipo Botão da Keystone

• Suporte com polaridade • Montagem segura para soldagem à onda • Mola de tensão confiável assegura baixa resistência no contato • Retém a bateria com segurança para suportar choques e vibrações • Faixa de temperatura operacional -50°C até +145°C; Cat. No. 1069 (CR1612, BR1616, CL1620, CR1632) Cat. No. 1065 (BR2012, CR2016, CL2020, DL2020, CR2032) Cat. No. 1068 (CL2430, CR2450) • Solicite o Catálogo M60.2

O que importa é o que

KEYSTONE ELECTRONICS CORP.

... está do lado de dentro.

Tel: (718) 956 8900 • (800) 221-5510
e-mail: kec@keyelco.com • Web: www.keyelco.com

Há uma pedra fundamental em toda grande invenção... Siga, Pense & aprenda em:

Aprenda como projetar um sistema de controle: Modelagem de simulação

Esta última parte mostra como construir um sistema dinâmico a partir de leis básicas da Física, por meio do software NI LabVIEW e do módulo LabVIEW Control Design and Simulation. O link para download do software NI LabVIEW, do módulo Control Design and Simulation e dos VIs pode ser encontrado no final deste artigo.

Professor Bill Messner
Professor Dawn Tilbury
Guilherme Kenji Yamamoto
Gustavo Guirao Licinio Peixinho
Renan Machado de Azevedo
National Instruments

Parte Final

Sistema de trem

Neste exemplo, vamos considerar um trem de brinquedo que consiste em um motor e um carro. Assumindo que o trem viaja somente em uma direção, nós queremos criar um sistema de controle que permita uma partida e parada suaves, juntamente com um passeio em velocidade constante.

As massas do motor e do vagão serão representadas por M_1 e M_2, respectivamente. As duas são unidas por uma mola, que possui um coeficiente de rigidez k. A força aplicada pelo motor é representada por F e a letra grega μ (mi), que também será representada pela letra u, representa o coeficiente de atrito de rolamento. Veja a **figura 1**.

Diagrama de corpo livre e lei de Newton

O sistema pode ser representado pelos Diagramas de Corpo Livre vistos na **figura 2**.

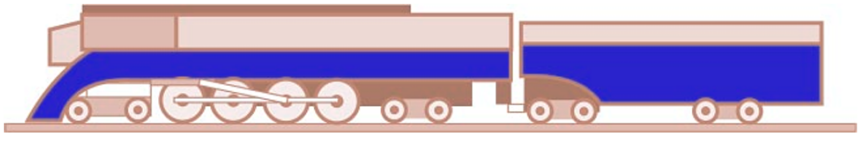
A partir da Lei de Newton, você sabe que a soma das forças que agem em uma massa é igual à massa vezes a sua aceleração. Neste caso, as forças que agem sobre M_1 são a mola, o atrito e a força aplicada pelo motor. As forças que agem sobre M_2 são a da mola e o atrito. Na direção vertical, a força gravitacional é cancelada pela força normal aplicada pelo chão, então não haverá aceleração na direção vertical. Começaremos a construir o modelo simplesmente a partir das expressões:

$$F_{M_1} = ¿\, M_1 . \dot{x}_1$$
$$\sum ¿$$
$$F_{M_2} = ¿\, M_2 . \dot{x}_2$$
$$\sum ¿$$

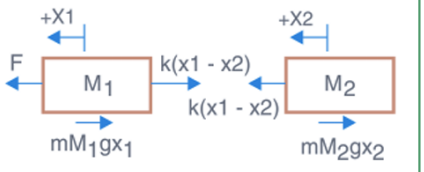
Construindo o modelo

Este conjunto de equações do sistema pode ser representado graficamente, sem manipulação adicional. Comece criando um *loop* de simulação e controle (*Control & Simulation Loop*), da sessão *Simulation* da paleta *Control Design & Simulation*.

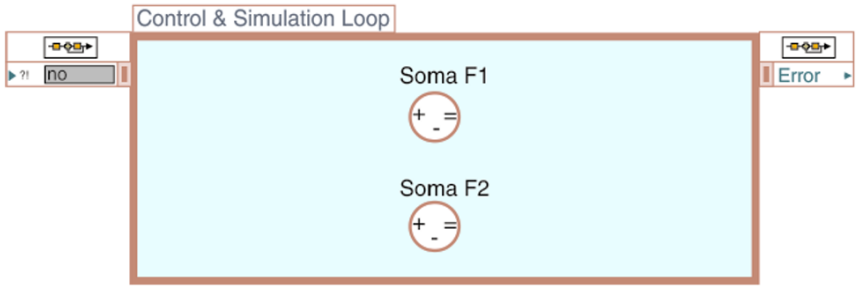
Primeiramente, vamos construir duas cópias (uma para cada massa) das expressões $\Sigma F = M \cdot a$ ou a = 1/M. Σ F. Arraste dois blocos somadores (*Summation*) (da sessão Signal Arithmetic da paleta *Simulation*) no *Control & Simulation Loop*. Dê um clique duplo em uma parte vazia do diagrama de blocos para criar etiquetas (*labels*). Coloque labels nos blocos somadores como "Soma F1" e "Soma F2", conforme mostra a **figura 3**.



F1. Trem de brinquedo.



F2. Diagramas de Corpo Livre.



F3. Blocos somadores (Summation).

As saídas de cada um desses blocos somadores representam a soma das forças agindo em cada massa. Multiplicar por 1/M nos dará a aceleração. Arraste dois blocos de ganho (*Gain*) ao seu modelo (da sessão Signal Arithmetic da paleta Simulation). Conecte cada uma às saídas dos blocos somadores.

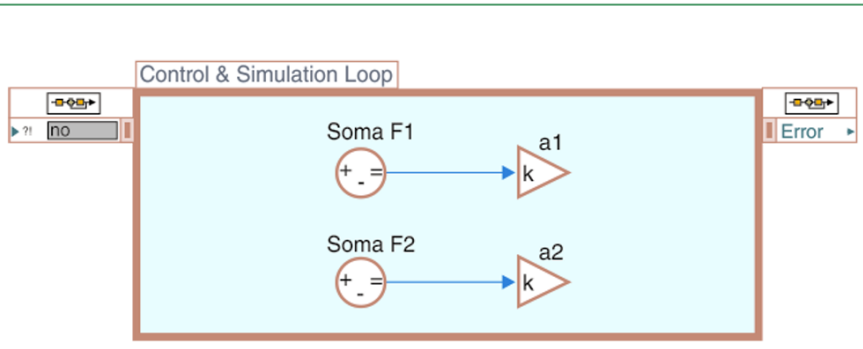
Esses blocos de ganho devem conter 1/M para cada uma das massas. Nós vamos inserir esses valores externamente. Dê um clique duplo em cada bloco de ganho, e altere o campo *Parameter Source de Configuration Dialog Box* para *Terminal*.

Coloque labels nesses dois blocos de ganho como "a1" e "a2", observe a **figura 4**.

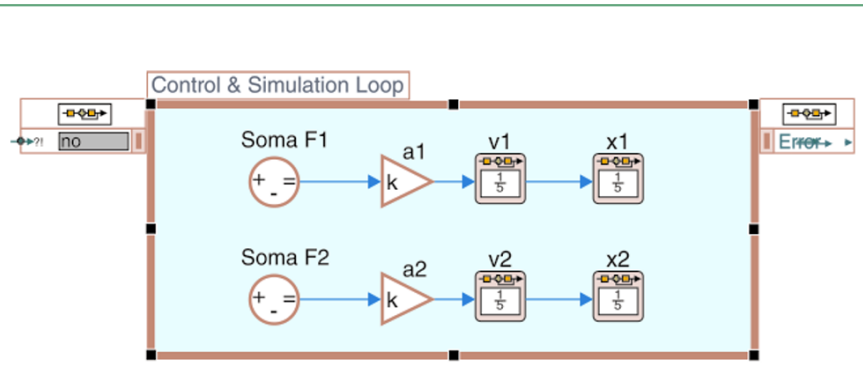
As saídas desses blocos de ganho serão as acelerações de cada uma das massas. Nós estamos interessados em ambas as velocidades e posições das massas. Uma vez que a velocidade é a integral da aceleração e a posição é a integral da velocidade, podemos gerar estes sinais usando blocos integradores.

Arraste dois blocos integradores (*Integrator*, da sessão *Continuous Linear Systems* da paleta *Simulation*) no seu modelo para cada uma das duas acelerações. Conecte-os com fios em duas sequências, como mostrado abaixo. Coloque labels nesses integradores como "v1", "x1", "v2" e "x2", uma vez que estes serão os sinais que os integradores vão gerar (**figura 5**).

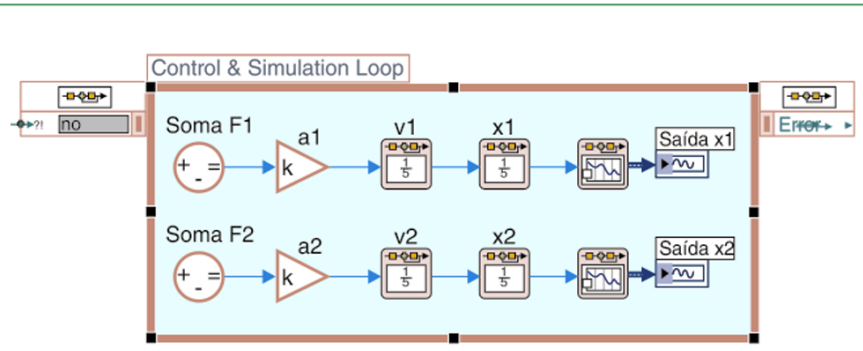
Agora, arraste dois blocos *SimTime Waveform* da sessão *Graph Utilities* da paleta Simulation, e conecte-os às saídas desses integradores. Nomeie-os como "Saída x1" e "Saída x2". Atente para a **figura 6**.



F4. Blocos de ganho.



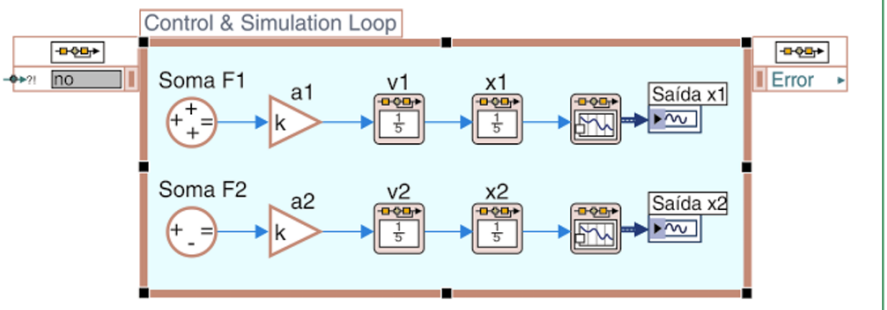
F5. Blocos integradores (Integrator).



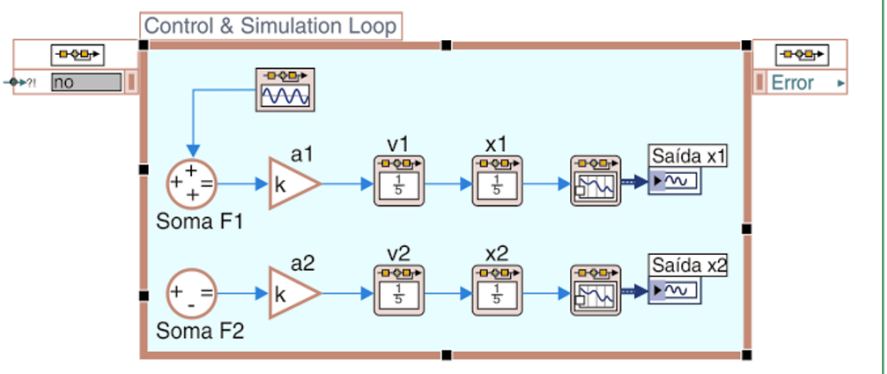
F6. Waveform Graphs.

Agora, estamos prontos para somar as forças que estão agindo em cada massa. Primeiro, você precisa ajustar as entradas em cada bloco somador para representar o número apropriado (nos preocuparemos com o sinal depois) de forças. Há um total de 3 forças agindo em M_1 – clique com o botão direito no bloco "Soma F1" e configure-o para ter três entradas (+++).

Existem somente 2 forças agindo em M_2, então podemos deixar F_2 sozinha no momento. Acompanhe na **figura 7**.



F7. Somador com três entradas.



F8. Signal Generator.

A primeira força agindo em M_1 é a força de entrada, F. Insira um bloco *Signal Generator* da sessão *Signal Generation* da paleta Simulation e conecte-o à entrada do ponto mais alto do bloco "Soma F1". Coloque uma label no Signal Generator contendo "F", conforme podemos ver na **figura 8**.

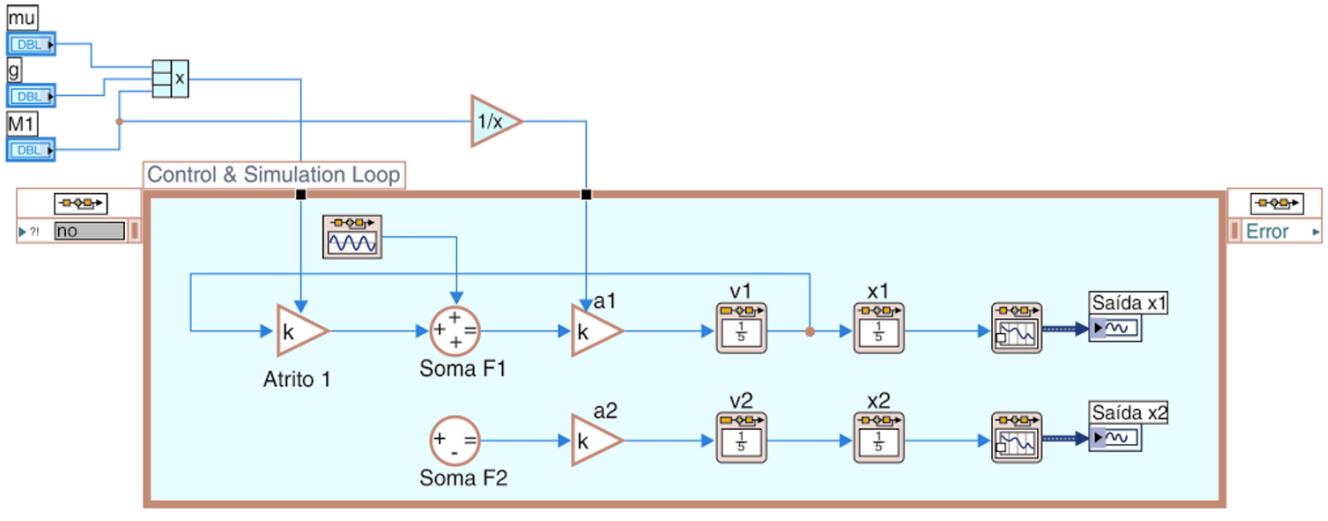
A próxima força agindo em M_1 é a força de atrito. Esta força é igual a:

$$F_{atrito\,1} = \mu . g . M_1 . v_1$$

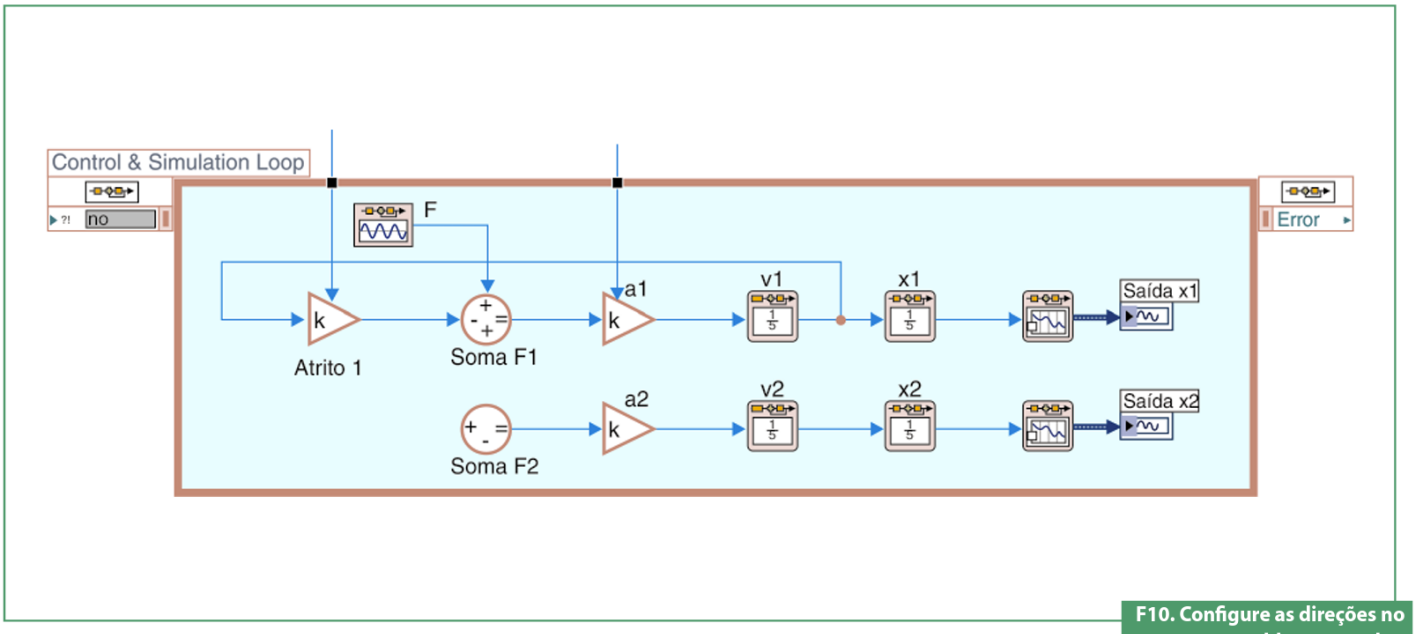
Para gerar esta força, nós podemos pegar o sinal da velocidade e multiplicá-lo por um ganho, $\mu \cdot g \cdot M_1$. Arraste um bloco de ganho à sua janela de modelo. Encontre o fio vindo do integrador v1 e conecte-o à entrada do bloco de ganho. Conecte a saída do bloco de ganho à segunda entrada do somador F1. Configure o bloco de ganho de forma que ele use um terminal para *Parameter Source* em vez da caixa de diálogo de configuração (*Configuration Dialog Box*). Nomeie este bloco como "Atrito 1".

O ganho deste bloco será $\mu \cdot g \cdot M_1$. Para obter isto, vá ao painel frontal e crie 3 controles numéricos. Nomeie-os como μ (mi), g, e M1.

No diagrama de blocos, use um VI *Compound Arithmetic* da paleta *Numeric* para multiplicar essas 3 variáveis. Conecte a saída ao novo bloco de ganho. Ainda,



F9. Adicione o atrito 1 e controles numéricos.



F10. Configure as direções no bloco somador.

use o VI *Reciprocal* para inverter M1 e conecte essa saída invertida ao terminal de a1. **Figura 9**.

Esta força, entretanto, age na direção negativa de x1. Portanto, ela deve ser conectada ao bloco "Soma F1" com um sinal negativo. Dê um clique duplo no bloco "Soma F1" e altere os sinais para + - +, como é visto na **figura 10**.

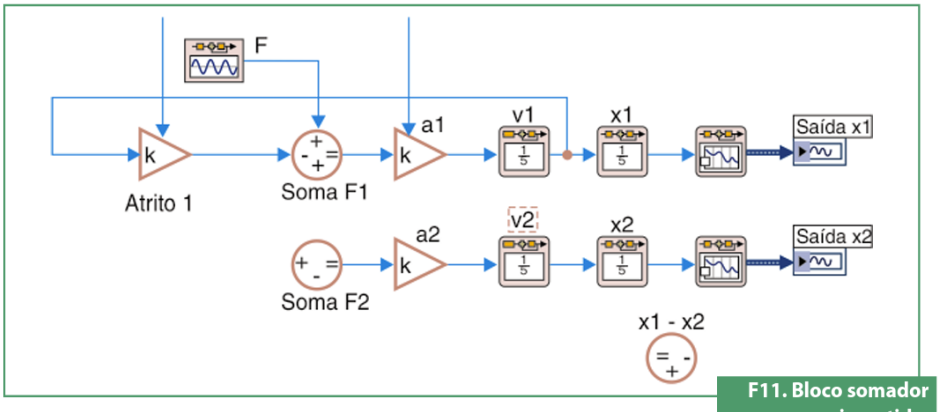
A última força agindo em M_1 é a força da mola entre as massas. Esta é igual a K . (x_1 - x_2).

Em primeiro lugar, nós precisamos gerar (x_1 - x_2) que então podemos multiplicar por k para gerar a força. Insira um bloco somador (Summation) abaixo do resto do seu modelo. Coloque nele uma label contendo "(x1-x2)" e altere sua lista de sinais para - +.

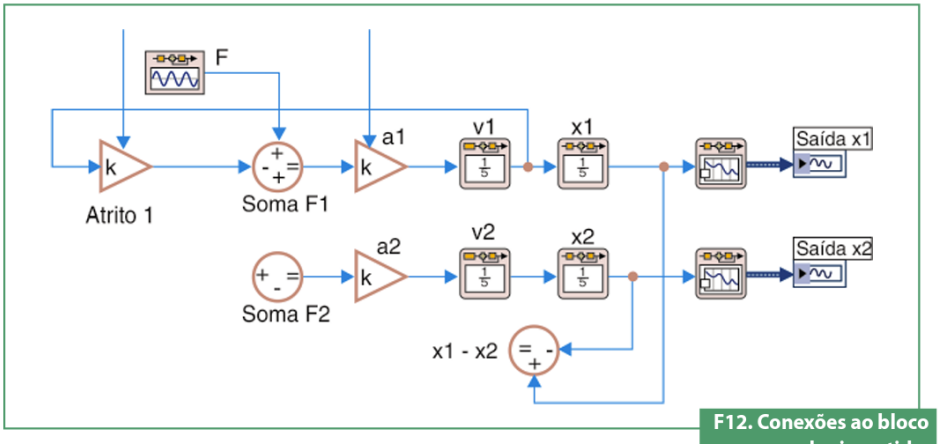
Uma vez que essa soma vem da direita para a esquerda, nós precisamos inverter o bloco. Clique com o botão direito no bloco e selecione "*Reverse Terminals*". Veja a **figura 11**.

Agora, divida o sinal x2 e conecte-o à saída negativa do bloco somador (x1-x2). Divida o sinal x1 e conecte-o à entrada positiva. **Figura 12**.

Podemos multiplicar essa diferença de posições pela constante da mola para gerar a força elástica. Insira um bloco de ganho no seu modelo à esquerda dos blocos somadores. Configure o bloco de ganho de forma que o "Parameter Source" seja um terminal.



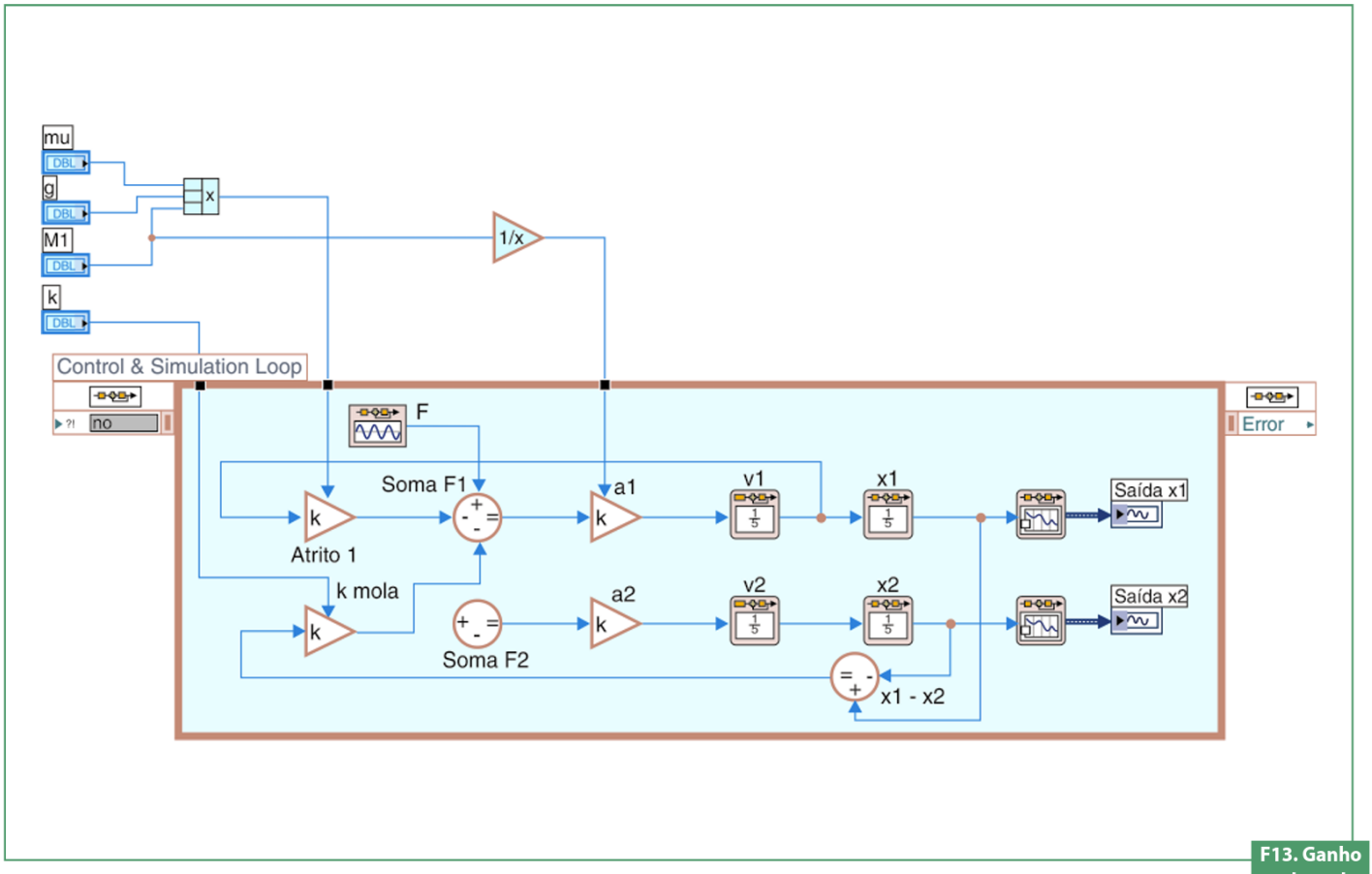
F11. Bloco somador invertido.



F12. Conexões ao bloco somador invertido.

No painel frontal, crie um controle numérico e nomeie-o como k. No diagrama de blocos, conecte este controle ao bloco de ganho que você acabou de criar. Coloque uma label no bloco de ganho como "mola".

Conecte a saída do bloco (x1-x2) à entrada do bloco da mola e conecte a saída do bloco da mola à terceira entrada de Soma F1. Altere o terceiro sinal de Soma F1 para negativo (+ - -). Observe a **figura 13**.



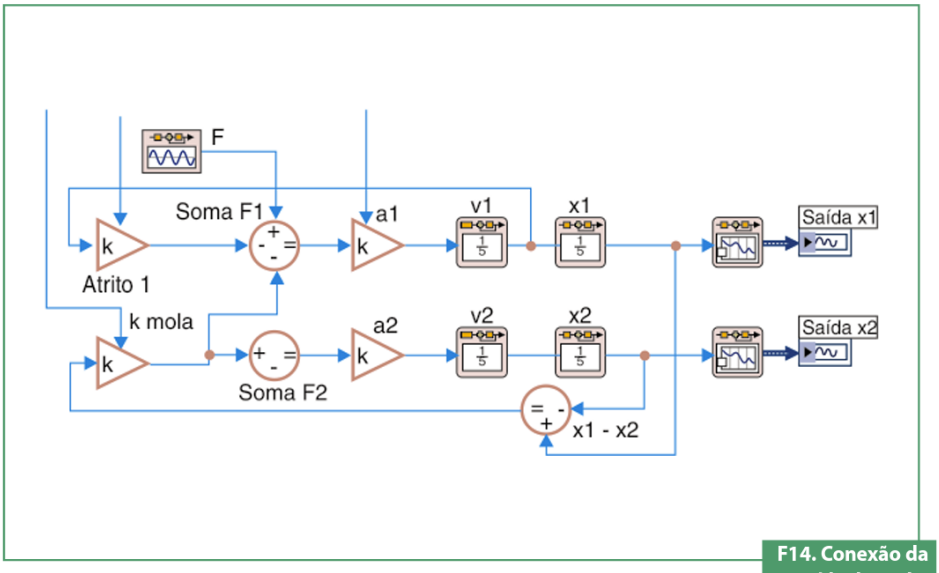
F13. Ganho da mola.

Agora, nós podemos aplicar forças a M_2. Para a primeira força usaremos a mesma força da mola que acabamos de gerar, porém, ela é somada com sinal positivo. Simplesmente pegue a saída do bloco da mola e conecte-o à primeira entrada de Soma F2. Repare na **figura 14**.

A última força a ser adicionada é o atrito em M_2. Isso é feito exatamente da mesma maneira que o atrito em M_1: pegando v2, multiplicando por um ganho de $\mu \cdot g \cdot M_2$ e adicionando a Soma F2 com sinal negativo. Você terá que criar um controle numérico para M_2 também. Deveremos ter, então, a seguinte imagem da **figura 15**.

Agora o modelo está completo. Nós somente precisamos fornecer a entrada adequada e visualizar a saída adequada.

A entrada do sistema será a força, F, fornecida pelo motor. Nós já colocamos o gerador de função na entrada. A saída do sistema será a velocidade do motor. Insira outro bloco *SimTime Waveform* no Simulation Loop, da sessão Graph Utilities da paleta Simulation. Conecte um fio da saída do bloco integrador "v1" a esse novo *Waveform Graph*, para visualizar a saída. Veja a **figura 16**.

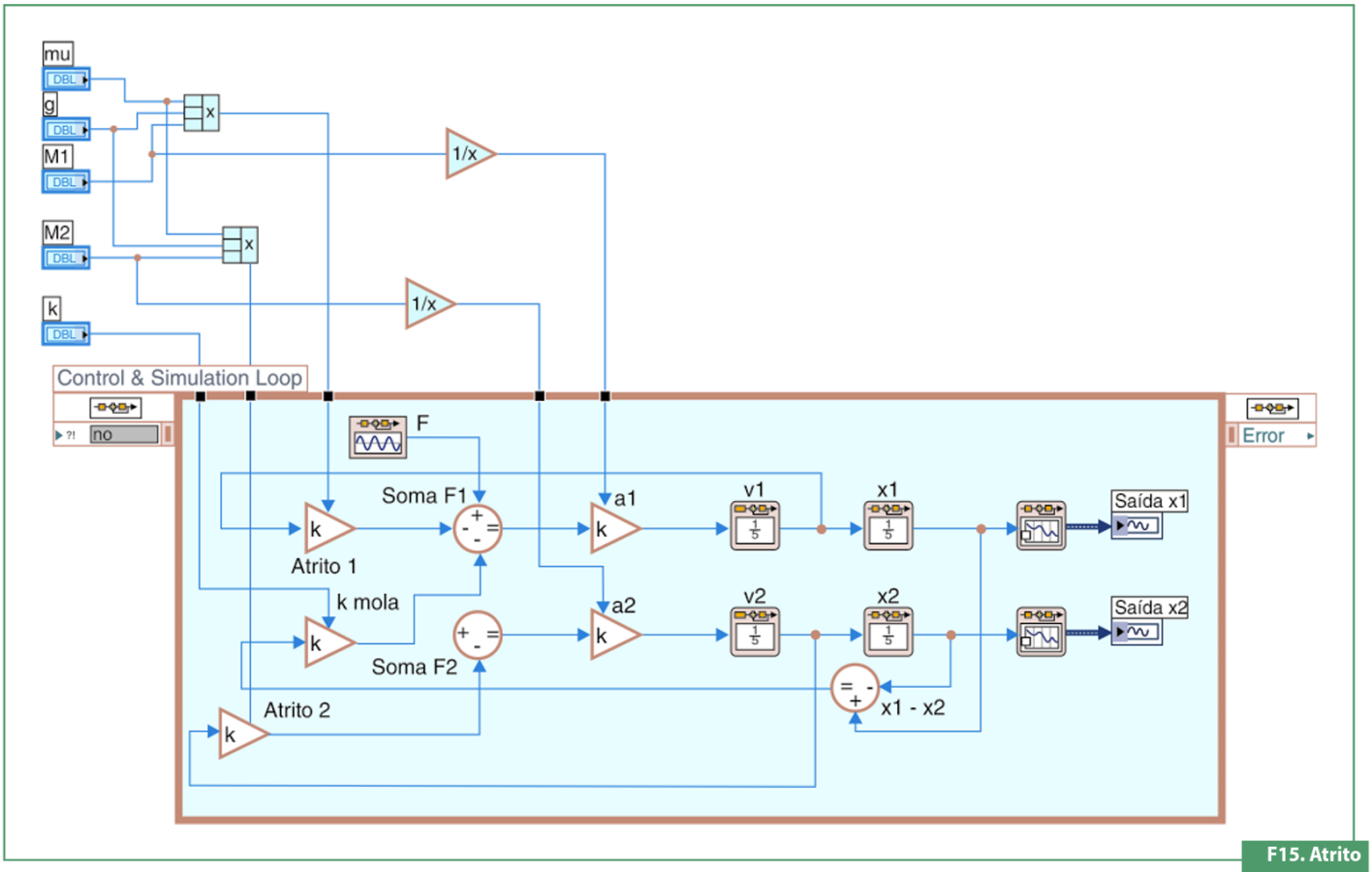


F14. Conexão da saída da mola.

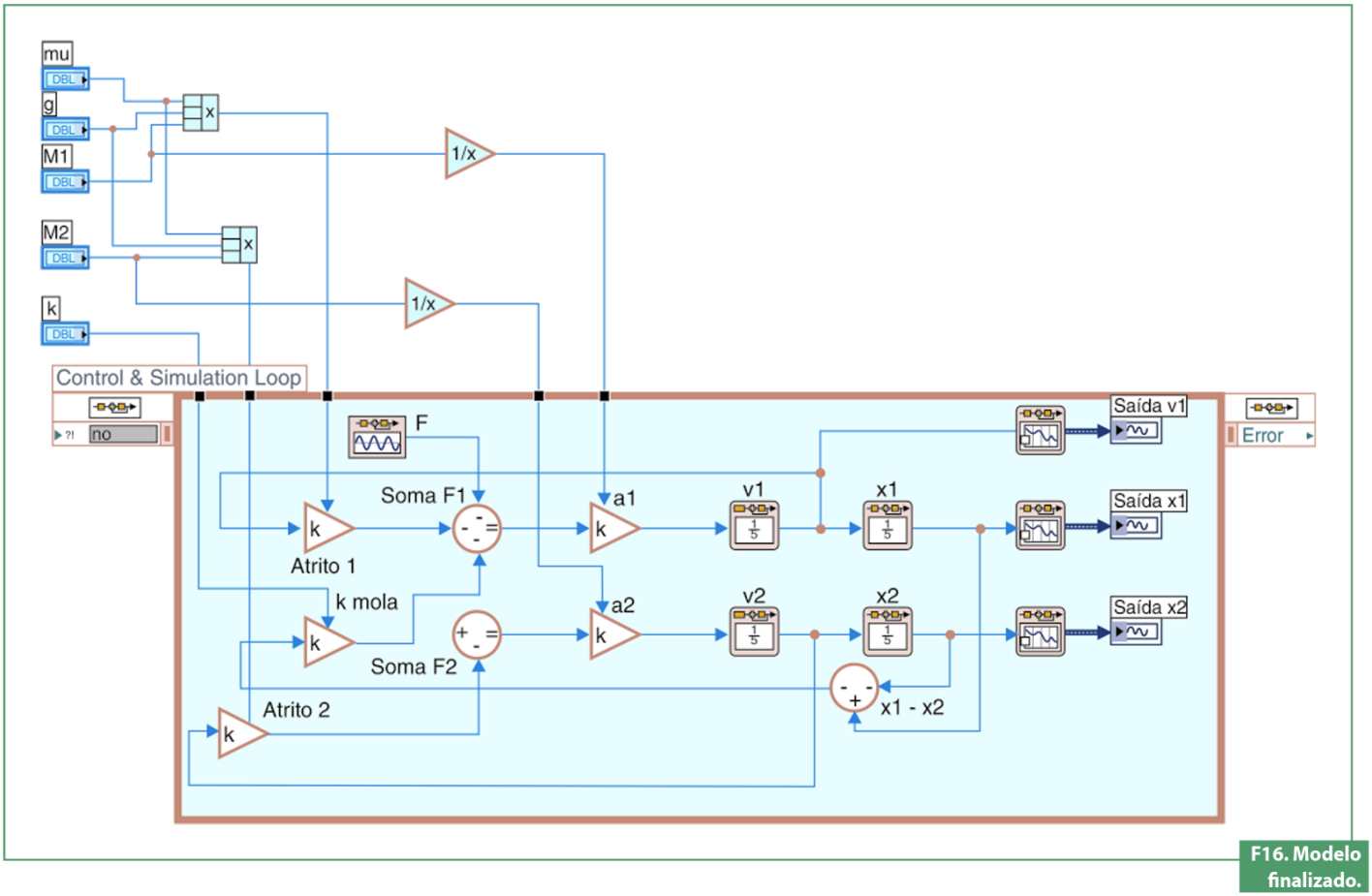
Executando o modelo

Antes de executar o modelo, precisamos atribuir valores numéricos a cada uma das variáveis usadas no modelo. Assim sendo, use os seguintes valores para o sistema do trem:

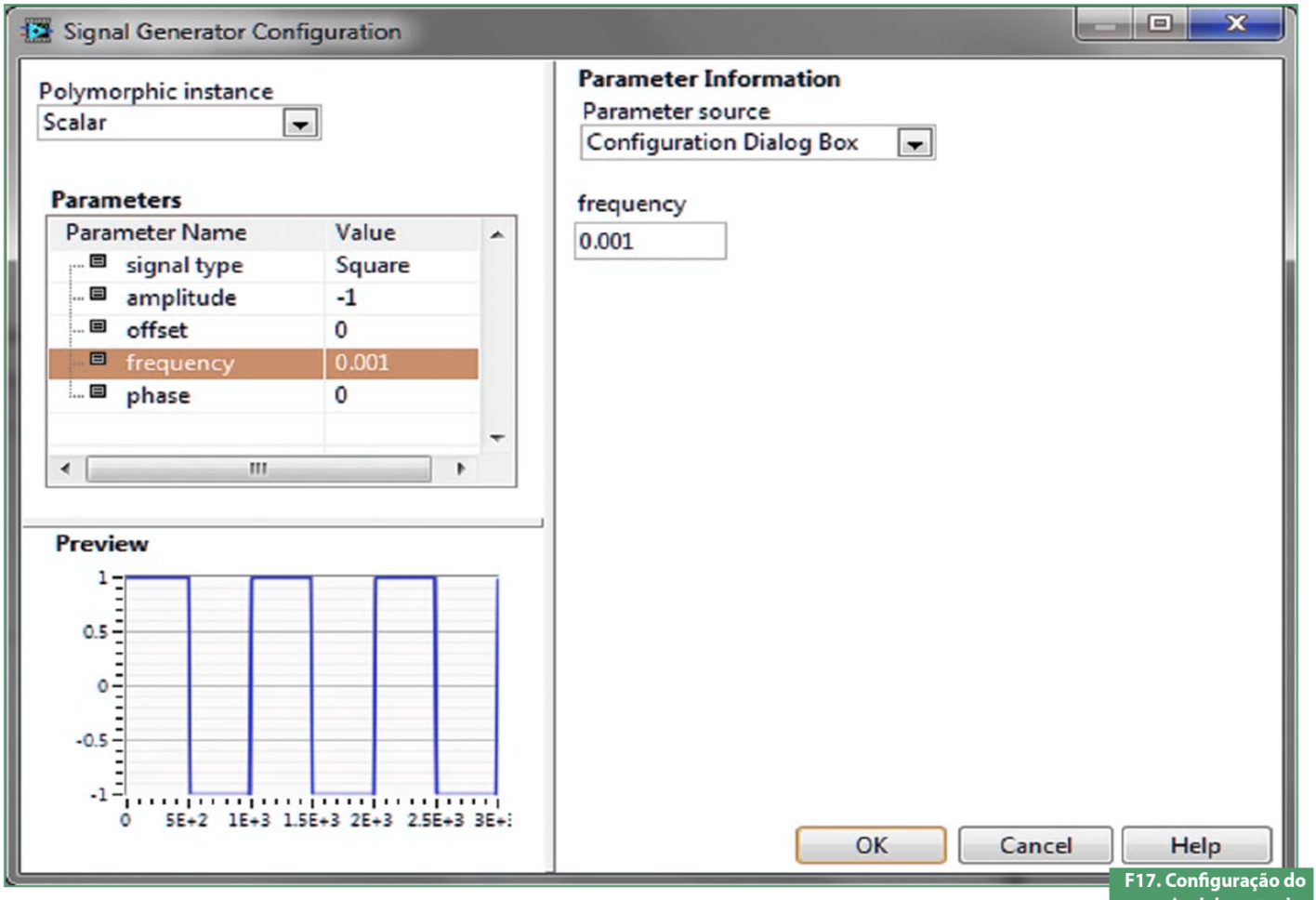
- M1 = 1 kg
- M2 = 0,5 kg
- k = 1 N/s
- F = 1 N
- u = 0,002 s/m
- i = 9,8 m/s²



F15. Atrito em M2.



F16. Modelo finalizado.



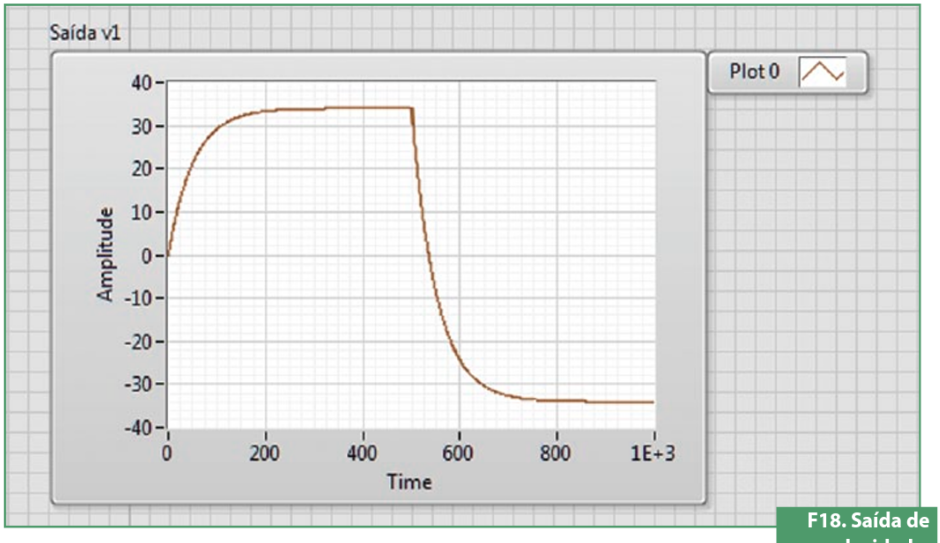
F17. Configuração do sinal de entrada.

Insira esses valores usando os controles do painel frontal.

Agora, precisamos dar uma entrada apropriada ao motor. Dê um clique duplo no gerador de sinais (Signal Generator – bloco F). Selecione uma onda quadrada com frequência 0,001 Hz e amplitude -1 (degraus de amplitude positiva que estão negativos antes de ir para positivo), como mostra a **figura 17**.

O último passo antes de executar a simulação é selecionar um tempo de simulação apropriado. Para visualizar um ciclo da onda quadrada de 0,001 Hz, nós devemos simular por 1000 segundos. Dê um clique duplo nos terminais localizados no lado esquerdo do Control & Simulation Loop. Na sessão *Simulation Timing*, no topo da caixa de diálogo, coloque o tempo inicial para 0 segundos, e o final para 1000 segundos. Feche a caixa de diálogo.

A seguir, alterne para o painel frontal. Clique com o botão direito no eixo x de cada Waveform Graph e selecione "*Autoscale X*".



F18. Saída de velocidade.

Execute a simulação e olhe no gráfico "Saída v1" (**figura 18**) para examinar a saída da velocidade. A entrada foi uma onda quadrada com dois degraus, um positivo e outro negativo. Fisicamente, isso significa que o motor primeiro foi para a frente depois para trás.

Download

Faça o download do software NI LabVIEW, do módulo Control Design and Simulation e dos VIs utilizados neste artigo por meio do link: **http://brasil.ni.com/saber-eletronica**

Projeto de um regulador chaveado de baixo ripple para MCUs e FPGAs

O uso de processadores e lógica programável vem crescendo muito nos últimos anos. Para atender essa demanda crescente de aplicações, com restrições cada vez maiores na alimentação, estão surgindo conversores chaveados de alto desempenho e integração.

Gustavo Leão Moreira
Engenheiro de Aplicações na MacnicaDHW
gustavo.leao@macnicadhw.com.br

Ao longo da última década, a porcentagem de projetos com FPGAs (*Field Programmable Gate Arrays*) e MCUs (*Microcontrollers*) deu um salto, passando de poucos milhões em 2003 a dezenas de milhões em 2013 no mercado nacional. Mercado que exige fontes mais limpas e baratas que performem elevada eficiência e garantam o bom funcionamento da parte digital.

Para atender essa demanda, houve a necessidade de criar uma nova arquitetura para a distribuição de potência na placa, denominada *Point-of-Load* (POL). Os POL fazem o trabalho árduo de converter uma tensão alta e pouco regulada do barramento de alimentação da PCI numa tensão de poucos volts, e precisa o bastante para ser utilizada pela lógica digital.

Uso de Point-Of-Load (POL)

O uso de POL vem para solucionar o problema da demanda de correntes elevadas e baixo *ripple* na tensão necessária para alimentar semicondutores mais modernos. Para isso, é adicionado um conversor (seja ele chaveado ou linear) próximo do seu ponto de uso e/ou carga. Neste artigo, o foco será o uso de POL chaveado, do tipo abaixador (*buck*). Normalmente, os POL funcionam com uma tensão de entrada de 12 V_{DC} a 5 V_{DC} e fornecem à carga tensões de 0,65 V a 3,3 V, com corrente na ordem de centenas de miliampères a uma centena de ampères.

No projeto de um POL, vários fatores devem ser levados em conta como a resistência de condução das chaves, retificação síncrona, tipo e valor dos capacitores de filtragem, modelo do indutor, resistência interna do indutor e capacitores, variação da capacitância com a tensão DC – principalmente quando o alvo é atingir baixo ripple de saída e alta eficiência na conversão. As etapas de seleção da frequência de chaveamento e ponto ótimo de trabalho também devem ser de domínio do projetista para obter-se um projeto de sucesso.

A vantagem em aumentar a frequência de chaveamento do POL

O conversor chaveado abaixador (*buck*) funciona através do armazenamento de energia da fonte no indutor - na forma de campo magnético - para posterior entrega dessa energia à carga. Esse processo de chaveamento entre armazenagem e entrega deve ser repetido periodicamente, para que o conversor opere de maneira satisfatória. Assim, a frequência de chaveamento (f_{SW}) influencia na operação do circuito, conforme pode ser visto na **equações** abaixo (variação da indutância com a frequência e variação da capacitância com a frequência):

$$L_{min} = \frac{V_{OUT} \cdot (V_I - V_{OUT})}{V_I \cdot LIR \cdot I_{OUT(MAX)} \cdot f_{SW}}$$

$$C_{min}^{OUT} = \frac{LIR \cdot I_{OUT(MAX)}}{8 \cdot f_{SW} \cdot V_{RIPPLE}}, \text{ onde } LIR \approx 0{,}3$$

Com o aumento da frequência, o indutor (normalmente o maior componente de uma fonte chaveada de pequena potência 0,5 ~ 20 W) e os capacitores de entrada/saída podem ser diminuídos, conforme pode ser visto pelas equações. A resposta transitória do circuito, com o aumento da frequência também é beneficiada, pois atinge mais rapidamente o valor de regime permanente, o ripple de entrada e saída da fonte são reduzidos e as perdas por ligar/desligar o canal do MOSFET também são menores. Ou seja, o projetista só tem a ganhar com o aumento da frequência!? Infelizmente não, o próximo tópico sobre perdas irá abordar o ponto negativo. Entretanto, mesmo havendo pontos negativos, as vantagens pesam mais que as desvantagens.

Ponto ótimo de operação do POL: perdas IR igual a perdas no chaveamento

O aumento da frequência de operação traz muitas vantagens para o funciona-

mento e diminuição dos componentes passivos do circuito. No entanto, ela também acaba causando problemas, pois acima de uma determinada frequência, as perdas por chaveamento (como a carga e descarga do *gate* do MOSFET, o *loop* de histerese do indutor e as emissões de ondas eletromagnéticas) aumentam demasiadamente, atingindo valores impraticáveis para produtos comerciais.

Uma boa relação sobre frequência de trabalho e ponto ótimo de operação, envolve o comparativo das perdas estáticas (ou IR) com as perdas dinâmicas (ou por chaveamento). De uma maneira simples elas podem ser exemplificadas como:

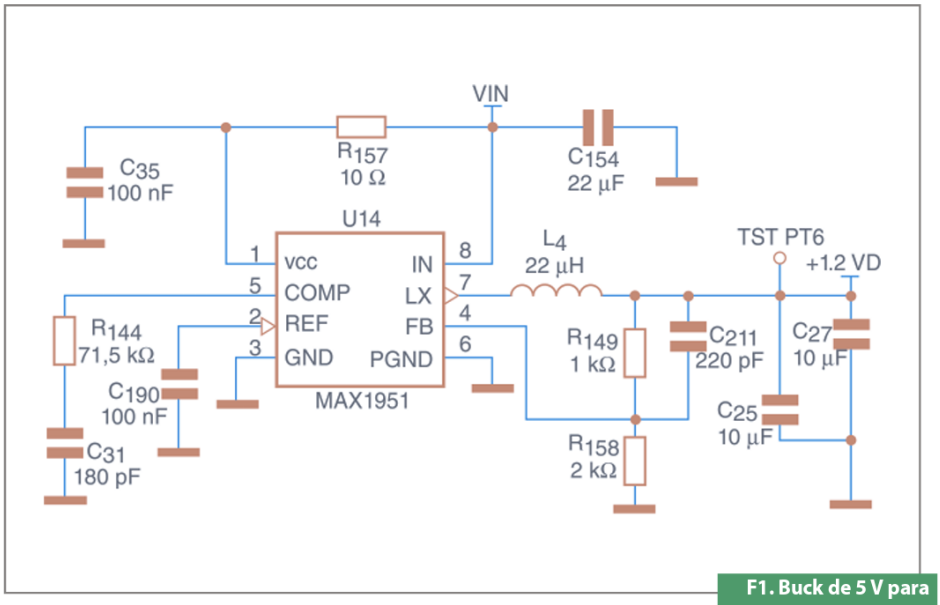
$$P_{IR} = I_{max}^2 . R_{DS}$$
$$P_{SW} = C_{GATE} . V_{DD}^2 . F_{SW}$$
$$P_{IR} = P_{SW}$$

Onde:
R_{DS}: soma das resistências de condução dos transistores da retificação síncrona;
I_{MAX}: é a corrente de saída máxima;
C_{GATE}: é a capacitância de gate dos transistores MOSFET;
V_{DD}: é a tensão aplicada no gate dos transistores MOSFET;
f_{SW}: é a frequência de operação.

Aumentando a eficiência do POL: substituindo LDO por reguladores chaveados

Os reguladores lineares (LDO) têm dominado o mercado no que se refere a alimentação de FPGAs e MCUs de média e baixa complexidade – nos quais correntes de até 2 A são necessárias. Essa dominação deve-se ao legado das fontes lineares, da baixa necessidade de reguladores eficientes (soluções normalmente alimentadas pela rede elétrica), além de serem circuitos mais simples de se projetar – na maioria dos casos basta apenas selecionar a corrente do regulador e *setar* a tensão de saída. Porém, está havendo uma grande mudança nesse mercado.

O regulador chaveado com custos cada vez mais reduzidos, *package* menor e ripple de saída na faixa de milivolts, além de possuir maior eficiência quando comparado ao LDO, estão fazendo com



F1. Buck de 5 V para 1,2 Vcore@2A.

que este seja obsoletado. Outro motivador dessa mudança é a busca de muitas empresas pela redução da corrente total consumida pela equipamento (seja ligado, ou em *standby*) e por uma área menor de placa. Uma regra simples para descobrir onde utilizar LDO e onde utilizar conversor chaveado é dada analisando-se a relação entre tensão de saída e tensão de entrada, e a corrente de saída, como apresentado a seguir:

$$\frac{V_{OUT}}{V_I} \geq 0{,}7 \quad e \quad I_{out} \leq 500\,mA$$

Utilizar LDO; se não, utilizar regulador chaveado"

Numa visão macro do sistema, percebe-se que a utilização do regulador chaveado no projeto não implica apenas numa maior eficiência do POL. Mas, como o regulador chaveado irá drenar uma corrente menor nos barramentos de alimentação, o projetista pode utilizar trilhas de cobre com espessura mais fina na PCI, assim como menor área de dissipação. Além disso, pode-se reduzir também a fonte externa (reduzindo o custo), visto que a corrente drenada pelos POLs será menor.

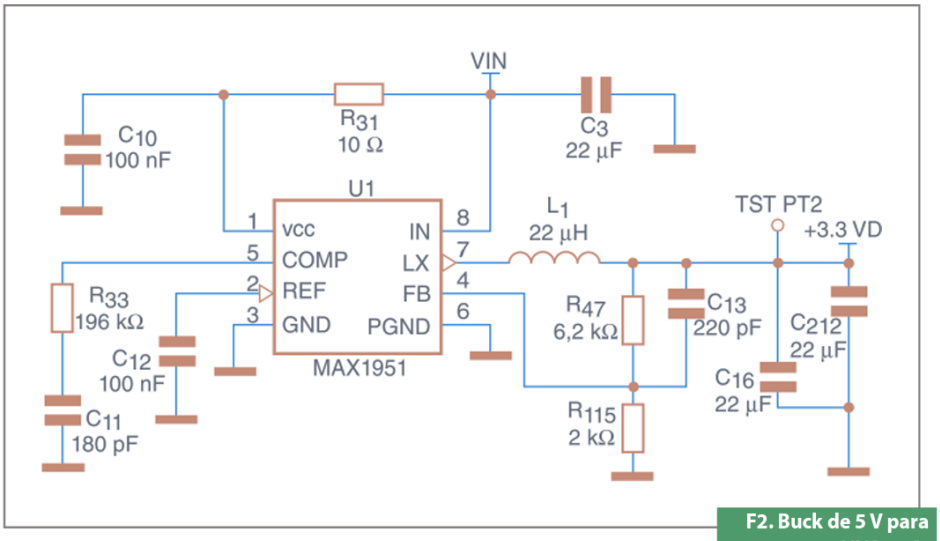
É possível citar como outros pontos a favor dos reguladores chaveados a redução da fadiga térmica na placa (por não esquentarem tanto), a ampla gama de CIs disponíveis no mercado com tensões de entrada de 1,3 V até 76 V e corrente de saída na faixa de 75 mA até 120 A, além de inúmeros componentes com eficiência na conversão chaveada em torno de 95%!

Exemplo prático: Conversor buck (abaixador) para alimentar Core e I/O de FPGA CycloneIV, da Altera

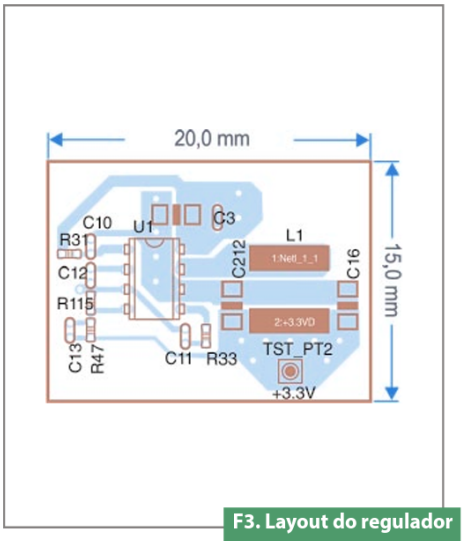
O circuito mostrado a seguir é utilizado no *kit* de FPGA MercurioIV® da MacnicaDHW. O mesmo consiste de dois conversores abaixadores, independentes, que geram 1,2 VDC e 3,3 VDC para alimentar, respectivamente, o Core (**figura 1**) e I/O (**figura 2**). É utilizada uma tensão de 5 V_{DC} - através de uma fonte externa ou USB2.0 - para alimentar os circuitos. Decidiu-se usar o MAX1951A para ambos os conversores, visto que a corrente máxima necessária em cada barramento é de aproximadamente 2 A.

O circuito elétrico e o layout

O integrado MAX1951A é o cérebro e o coração do circuito. Ele trabalha numa frequência fixa de 1 MHz e possui internamente as chaves do lado alto e do lado baixo (retificação síncrona), o que garante reduzido espaço de placa (**figura 3**) aliado a alta eficiência (superior a 90% em alguns casos). O valor e tipo dos componentes passivos utilizados, vide **tabela 1**, foram definidos de acordo com a sugestão do fabricante Maxim Integrated, presentes na *data sheet* do componente.



F2. Buck de 5 V para 3,3 VI/O@2A.



F3. Layout do regulador 15 mm x 20 mm.

O *layout* também requer atenção no seu desenvolvimento. Como sugestões estão o uso de planos e polígonos para interligar componentes que exigem correntes altas (evitando ao maximo trilhas finas); manter os capacitores de entrada e saída, indutor e CI o mais próximos possível – para reduzir loops grandes de corrente; e afastar a parte de controle (resistores de realimentação e compensação) da parte de potência – para não haver *crosstalk*. Utilizar sempre boas práticas de layout aumenta a probabilidade de que o circuito na PCI funcione de acordo com o projeto!

Melhorando o ripple: Uso da malha de controle a seu favor

A malha de controle dos conversores tem a função de estabilizar a tensão de saída, independentemente da tensão de entrada e da corrente de saída. Para isso, é utilizado um sinal de realimentação da tensão de saída, que é comparado com um valor de referência interno para controlar as chaves de potência. Nos conversores que suportam ajuste da tensão de saída, normalmente é usado um divisor resistivo para adequar a tensão de saída à tensão de realimentação de referência. Porém, quando se utiliza apenas resistores para fazer essa adequação de tensão (exemplo: R149 e R158 da figura 1), o ripple que aparece no pino de realimentação é também um valor fracionário do ripple que está na saída. Assim, a malha de controle, por melhor que seja, não conseguirá reduzir o ripple de saída.

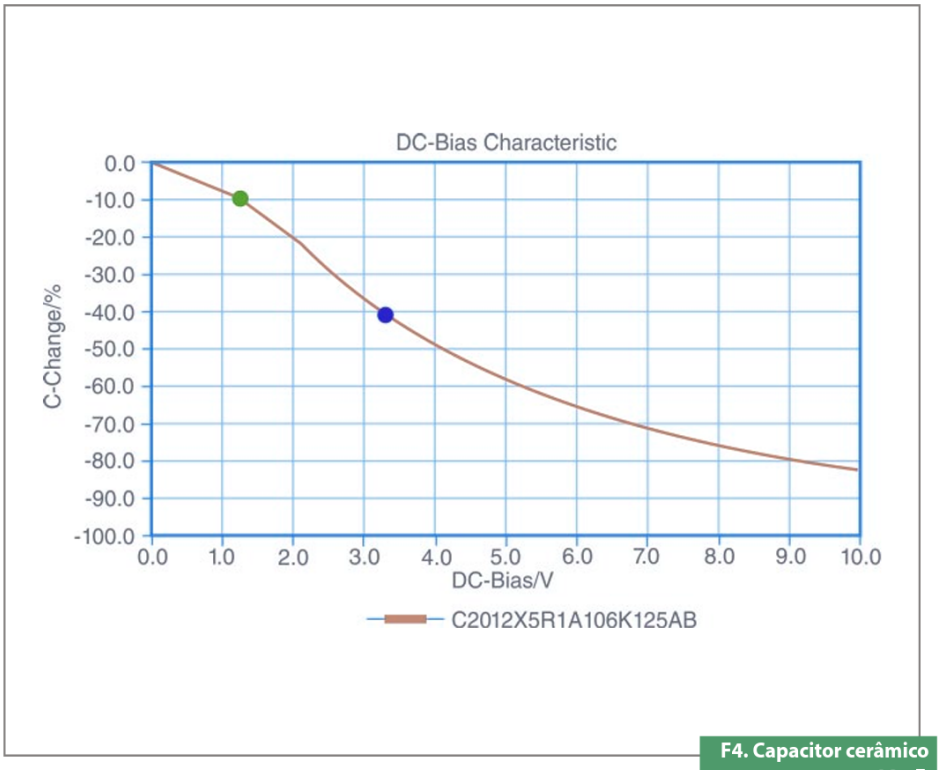
Uma maneira de resolver esse problema, é adicionar um capacitor de pequeno valor (capacitores cerâmicos na faixa de dezenas a centenas de pF são recomendados) entre o pino de realimentação e a saída do conversor (exemplo: C211 da figura 1). Ele tem a função de deixar passar apenas o valor AC ou o ripple por inteiro (e não apenas uma parte dele, caso haja um divisor resistivo somente no loop de realimentação) da saída para ser sentida através da malha de controle. Essa alteração causará uma diminuição do ripple a um valor que depende, basicamente, da tensão de realimentação do regulador (V_{fb}) e da tensão de saída (V_{out}). Esse novo ripple pode ser calculado como:

$$V_{Ripple}^{new} = \frac{V_{Ripple}^{old} \cdot V_{FeedBack}}{V_{out}}$$

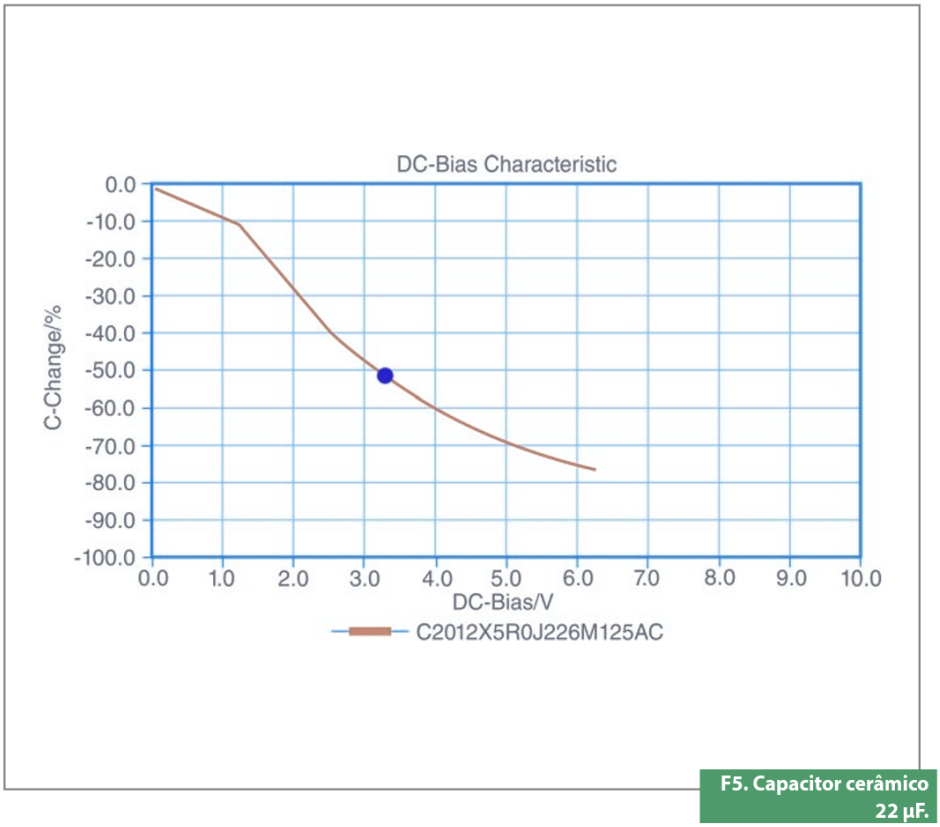
Pode-se fazer uso desse capacitor sempre que for necessário reduzir o ripple da sua fonte (chaveada ou linear), porém, cuidado com certas malhas de controle e valores de capacitância, pois a inserção desse componente no circuito pode causar instabilidade na saída. Simulações e testes práticos são sempre bem-vindos!

Comment	Package	Part number
100 nF	0402	C1005X5R1A104K050BA
180 pF	0402	C1005C0G1H181J050BA
220 pF	0402	C1005X7R1H221K050BA
22 µF	0805	C2012X5R0J226M125AC
10 µF	0805	C2012X5R1A106K125AB
1 k	0402	ERJ-2RKF1001X
10 R	0402	ERJ-2RKF10R0X
196 k	0402	ERJ-2RKF1963X
2 k	0402	ERJ-2RKF2001X
6k2	0402	ERJ-2RKF6201X
71k5	0402	ERJ-2RKF7152X
2,2 µH	L5050	VLC5045T-2R2N
MAX1951	SOIC8	MAX1951ESA+

T1. Componentes utilizados nos conversores.



F4. Capacitor cerâmico 10 µF.



F5. Capacitor cerâmico 22 µF.

Como medir o ripple?

O processo de medição do ripple de saída é simples, mas se executado de maneira incorreta, poderá introduzir erros nos resultados. Esses erros normalmente são originados pelo fio com "jacaré" de GND - entre a ponteira do osciloscópio e o circuito a ser testado - ao funcionar como antena, captando ondas eletromagnéticas presentes no ambiente e afetando o resultado. Para minimizar esses erros, ao invés de conectar o GND da ponteira através do fio com jacaré ao circuito a ser testado, utilize a ponteira de GND em forma de mola (acompanha os modelos mais novos de osciloscópio). Um outro detalhe para garantir uma medida fiel, é usar sempre os terminais do capacitor de saída e/ou entrada para conectar a ponteira do osciloscópio (sinal e GND) ao circuito a ser testado.

Capacitores cerâmicos e a sua variação com tensão DC

O capacitor cerâmico apresenta uma série de características que o tornam preferido para a filtragem da tensão de entrada e saída de fontes chaveadas de baixa potência. Pode-se citar como exemplos a vida útil prolongada, reduzido ESR e ESL e baixo custo. Entretanto, o capacitor cerâmico sofre grande influência negativa quando é polarizado com uma tensão DC (4) – no caso dos conversores chaveados, essa tensão DC é V_{in} e V_{out}. As **figuras 4** e **5** mostram esse comportamento para os capacitores de saída da fonte em discussão de 3,3 V e 1,2 V (22 µF e 10 µF respectivamente).

O projetista deve atentar a esse efeito e aumentar, caso necessário, o valor da capacitância de entrada e/ou saída.

Como nesse exemplo, onde o conversor de 3,3 Vout deve ter uma capacitância de saída em torno de 20 µF de acordo com o projeto, tal fato implica em dois capacitores de 10 µF em paralelo, correto? Não, pois, devido à queda de capacitância, o valor de 10 µF em 3,3 V é 40% menor (6 µF). Com isso, a capacitância de saída seria apenas 12 µF (ponto vermelho na figura 4). Assim fez-se necessário aumentar esses capacitores para 22 µF – que em 3,3 V se torna 50% menor, ou seja, 11 µF cada um (ponto vermelho na figura 5). Para o caso da fonte de 1,2 V, a queda na capacitância é de apenas 10% (ponto verde na figura 4), o que não causa nenhum problema ao circuito.

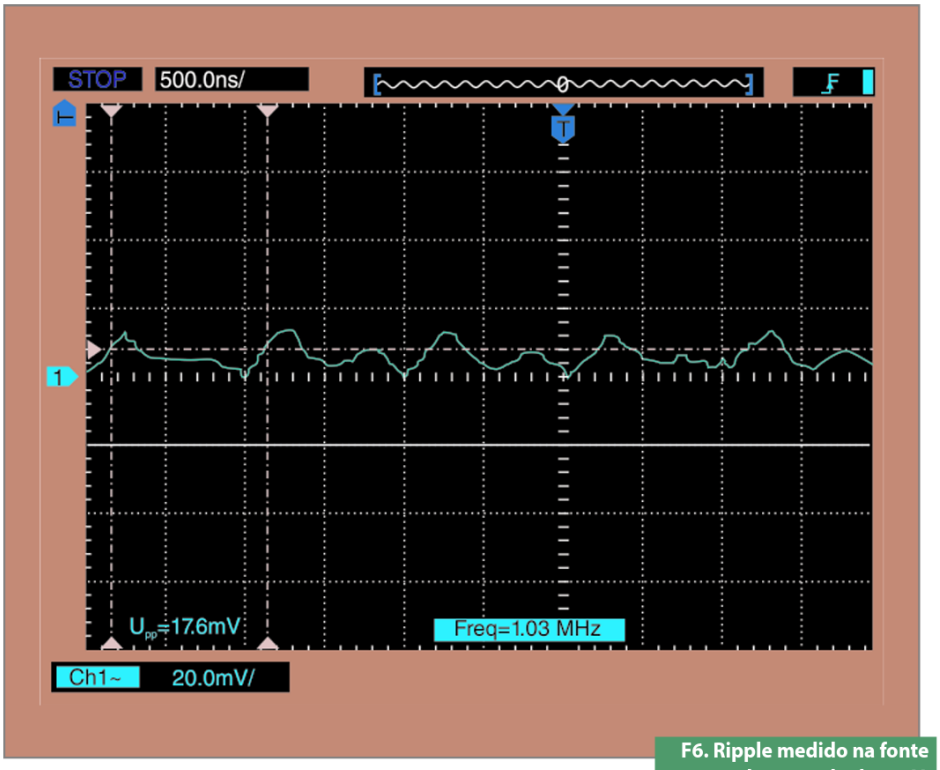
Conclusão

Fica evidente que as empresas de desenvolvimento de hardware estão vendo os benefícios de se utilizar reguladores chaveados em seus produtos, onde

uma eficiência global maior é atingida – economizando assim energia elétrica do consumidor - e custos menores de fabricação são envolvidos.

Os resultados obtidos com a implementação prática dos reguladores para o CORE e I/O do *kit* MercurioIV foram comprovadamente validados e atenderam todos os requisitos do projeto. Na parte do ripple de saída, chegou-se ao valor pico a pico de apenas 17,6 mV, como pode ser visto na **figura 6**. Um excelente resultado, visto que o CycloneIV suporta até 100 mV de oscilação na sua alimentação nesses barramentos. Caso a aplicação do leitor necessite de um ripple ainda menor, deve-se apenas adicionar uma maior capacitância na saída do conversor.

O mercado mostra que o aumento da frequência de chaveamento, na faixa de megahertz veio para ficar. Reguladores chaveados com baixo ripple de saída - comparável ao LDO e de apenas alguns milivolts - eficiência na conversão acima dos 90%, preços reduzidos e área de placa menor são apenas alguns dos motivadores dessa mudança. Inclusive já estão disponíveis reguladores que operam em frequências de até 4 MHz. Ao que tudo indica, com os avanços dos semicondutores e materiais, em alguns anos teremos fontes trabalhando na frequência de dezenas de megahertz.



F6. Ripple medido na fonte do exemplo de 1,2 V.





Conversores Buck-Boost

com Corrente Quiescente, de 1,3 μA, desenvolvidos para "Aplicações de Micropotências" e "Internet das Coisas"

A proliferação do uso de sensores wireless dentro da "Internet das Coisas"aumentou a procura pelos pequenos e eficientes conversores CC/CC, feitos sob medida para alimentar os dispositivos de baixa potência. Neste artigo, vamos estudar algumas aplicações importantes desses novos chips da Linear Technologies.

Dave Salerno
Tradução/Adaptação: Eutíquio Lopez

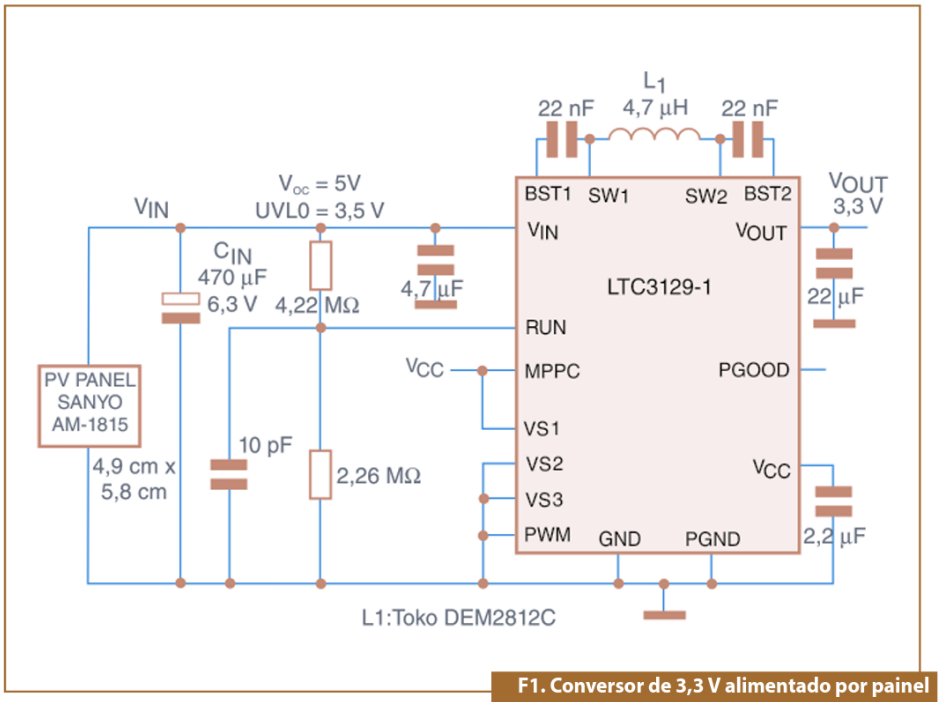
Os novos conversores buck-boost de 15 V, denominados LTC3129 e LTC3129-1, da Linear Technologies, foram projetados para satisfazer a necessidade acima citada.

Eles consistem de dispositivos monolíticos com faixa de tensões de entrada entre 2,42 V e 15 V. O primeiro deles apresenta um range de tensões de saída de 1,4 V a 15,75 V, ao passo que o segundo oferece oito tensões de saída fixas (selecionadas por pinos) entre 1,8 V e 15 V. Ambos componentes podem fornecer uma corrente mínima de 200 mA, quando operam no modelo buck.

Os sensores de baixa potência podem beneficiar-se da corrente nula desses conversores, quando estão desabilitadas ambas tensões V_{IN} e V_{OUT}, e também de uma corrente quiescente em V_{IN} = 1,3 μA, poupando energia quando operam em Modo Burst (rajada). Assim, eles se tornam ideais para uso em aplicações de micropotências e de coleta de energia, onde é decisivo um alto rendimento com cargas extremamente baixas. Suas arquiteturas buck-boost tornam esses conversores muito apropriados para uma grande variedade de fontes de alimentação.

Outras características fundamentais dos CIs LTC3129 e LTC3129-1 dizem respeito à frequência de operação de 1,2 MHz (fixa); ao controle em modo corrente; compensação de *loop* interna; operação automática em modo Burst, ou em modo PWM com baixo ruído; um preciso *threshold* no pino RUN para permitir o limiar UVLO a ser programado; uma saída *power good* e um MPPC (máximo controle do ponto de potência) para otimizar a transferência de energia quando funcionando com células fotovoltaicas.

O compacto invólucro QFN, de 3 mm x 3 mm, e o seu alto nível de integração, fazem desses dois CIs componentes ideais em aplicações com espaço de placa limitado. Um projeto de fonte de alimen-



F1. Conversor de 3,3 V alimentado por painel solar que opera com luz interna.

tação será facilmente desenvolvido com apenas alguns componentes externos e um pequeno indutor de (2 mm x 3 mm), por exemplo. A compensação de loop interna, geralmente, simplifica o processo de projeto.

Os LTC3129 e LTC3129-1 podem começar a operar a partir de uma fonte interna de 7,5 µW, o que lhes permite funcionar com pequenas células solares de baixo custo (com dimensão menor que 1 pol²) com níveis de luz interna abaixo de 200 lux, viabilizando sua aplicação em sensores wireless energizados por luz interior.

Acompanhe, a seguir, a descrição do 1º exemplo de projeto de um conversor com os citados CIs.

Conversor de 3,3 V funcionando com a luz interna de uma pequena célula solar

O circuito mostrado na **figura 1** consiste de uma aplicação como sensor wireless energizado pela luz interior da célula fotovoltaica (PV Panel Sanyo AM-1815), no qual o conversor CC/CC opera com uma exigência de potência extremamente baixa devido ao seu baixo *duty cycle* de operação.

Para possibilitar sua baixa corrente de partida, esses CIs drenam somente 2,0 µA (menos ainda em *shutdown*) até que três condições sejam satisfeitas:

- O nível de tensão do pino RUN deve exceder 1,22 V, tipicamente;
- O nível de tensão do pino V_{IN} deve ser maior que 1,9 V (típico);
- V_{CC} deve ser superior a 2,25 V, tipicamente.

Enquanto essas condições não estiverem realizadas, o conversor permanece no estado de "soft-shutdown" ou *standby*, drenando justamente os 2,0 µA.

A tensão de entrada em que o componente fornece o nível UVLO pode ser retirada de qualquer valor entre 2,4 V e 15 V, usando-se o divisor resistivo externo pendurado no pino RUN. Com uma corrente nesse pino menor que 1 nA (típica), resistores de alto valor ôhmico devem ser empregados para reduzir a corrente drenada de V_{IN}.

Nesta aplicação, a energia armazenada no capacitor C_{IN} é utilizada para regular V_{OUT}, logo que o conversor começa a operar. Se a demanda média de potência em V_{OUT} for menor que a liberada pela célula solar, o CI permanecerá operando em Modo Burst, e o nível V_{OUT} continuará regulado. Em caso contrário, o nível V_{IN} cairá até alcançar a tensão UVLO, na qual o conversor retorna ao shutdown. Nesse ponto, V_{IN} começa a recarga, permitindo que o ciclo se repita.

Nesse modo de operação variante (*hiccup*), V_{IN} é posicionado histereticamente com respeito ao nível UVLO com um *ripple* de aproximadamente 290 mV. Esse ripple é estabelecido pela histerese de 100 mV no pino RUN, ampliada pela relação do divisor de UVLO. Note que, sendo feito assim, a célula solar funcionará sempre próximo de sua máxima tensão de transferência de energia (potência).

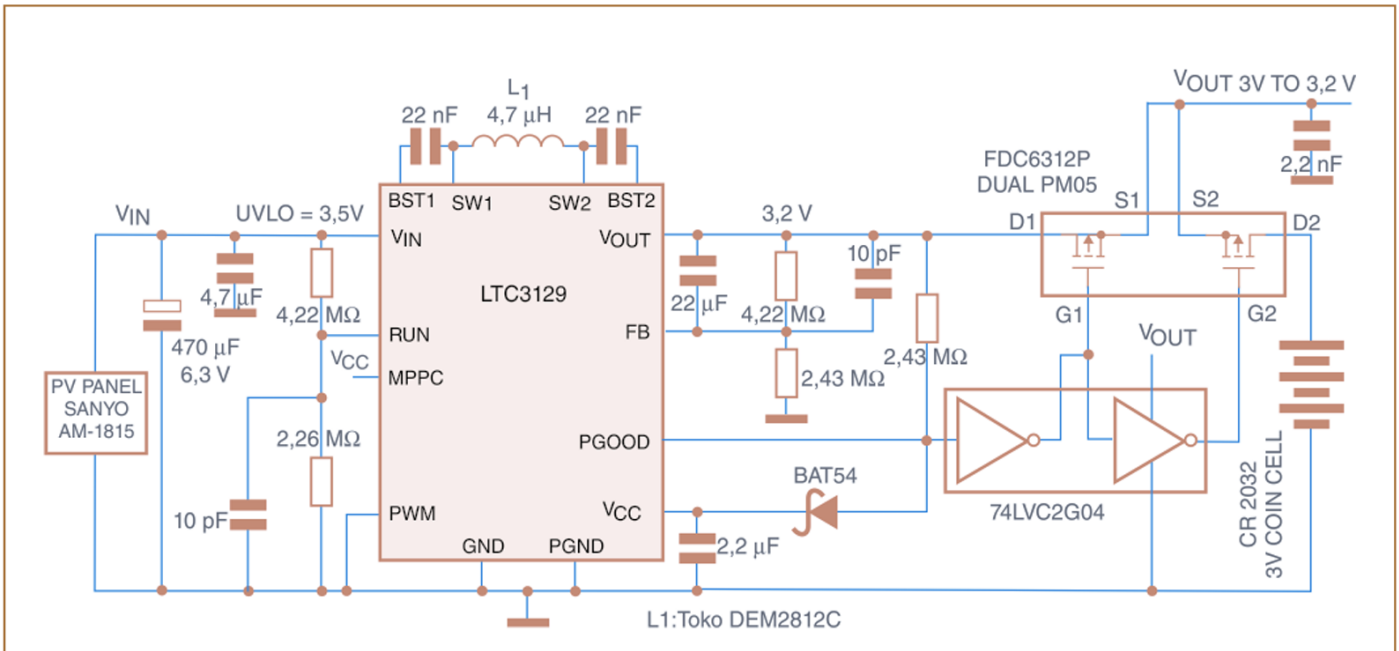
Para otimizar o rendimento e eliminar um carregamento desnecessário de V_{OUT}, o LTC3129/ 3129-1 não drena nenhuma corrente dessa saída durante o *soft-start* ou em qualquer outro tempo, caso a operação em Modo Burst tenha sido escolhida. Isso permite que o conversor descarregue V_{OUT} durante o soft-start, e, portanto, preserve a carga do capacitor de saída.

De fato, quando o LTC3129 está em modo *sleeping* (soneca), não existe nenhuma corrente sendo drenada de V_{OUT}. No caso do LTC3129-1, a corrente drenada é de sub microampères devido às altas resistências do divisor de *feedback* interno.

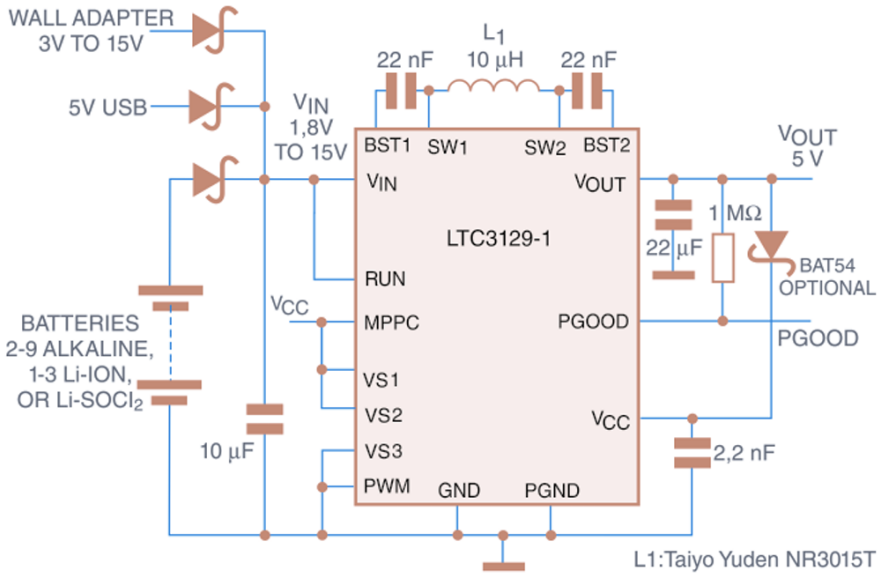
Acrescentando ao Conversor uma Bateria Reserva

Em muitas aplicações energizadas pela luz solar, uma bateria reserva fornece potência complementar quando necessário (no caso da energia solar ser insuficiente).

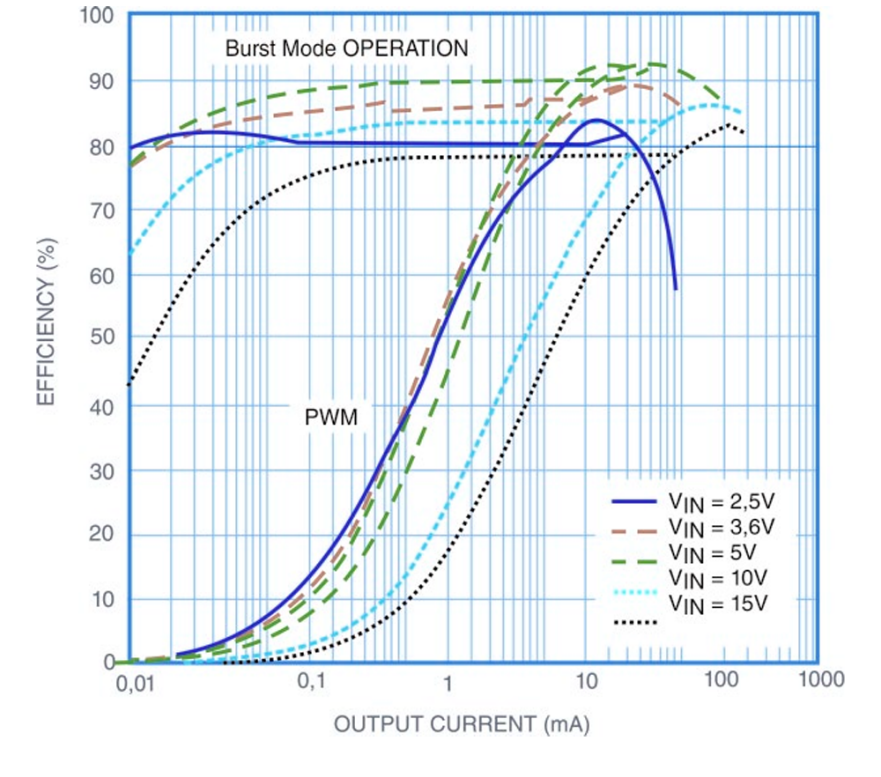
A **figura 2** ilustra uma aplicação onde uma célula de lítio primária (tipo moeda) e alguns componentes externos foram acrescentados ao circuito do conversor



F2. Conversor alimentado por painel solar com bateria reserva (tipo "moeda").



F3. Conversor de 5 V com múltiplas entradas.



F4. Curvas de (Rendimento x VIN e Carga) do Conversor de 5 V da anterior.

nº 1 (anterior) para prover energia de reserva à sua saída, caso a fonte de luz seja incapaz de fornecer a potência necessária para manter a tensão de saída V_{OUT}. O LTC3129 é empregado nesta circunstância, permitindo que V_{OUT} seja programado para 3,2 V de modo a casar melhor com a tensão da célula.

Neste 2º exemplo, a bateria é colocada no lado de saída do conversor e o LTC3129 opera para regular V_{OUT} um pouco acima da tensão da bateria. Isso garante que não haja nenhuma carga na bateria enquanto V_{OUT} possa ser energizado pela entrada de luz solar.

Na eventualidade de V_{OUT} cair devido à luz insuficiente para alimentar a carga, a saída PGOOD do LTC3129 vai para um nível baixo, chaveando a carga da saída do conversor para a bateria e retendo então V_{OUT} na tensão da bateria. Durante esse tempo, a entrada do conversor e os capacitores de saída estarão aptos para a recarga (caso haja alguma luz disponível), habilitando o chaveamento periódico da carga de volta da bateria para o conversor através do sinal PGOOD.

Sendo assim, a carga será alimentada pela entrada de luz solar o quanto for possível, e a bateria será usada de modo compartilhado estendendo sua vida.

O diodo ligado entre PGOOD e V_{CC} foi colocado para manter PGOOD baixo durante a partida (*start-up*), antes que V_{CC} esteja valendo.

Escolhendo onde colocar a bateria reserva

Neste conversor, a bateria foi posicionada junto à saída. Em aplicações com cargas leves, isso é vantajoso por não expor a bateria (que poderá ser de baixa capacidade com resistência extremamente alta) a *bursts* de corrente de entrada relativamente altos no start-up do conversor, provocando uma queda significativa da tensão da bateria e dissipação de potência interna também, que, por sua vez, reduz o tempo de vida dela.

As desvantagens de posicionar essa bateria reserva na saída do conversor residem no fato de que a tensão deve ser bem casada com o nível de saída requerido, e que ela deve possuir uma curva de descarga relativamente plana de forma a manter uma regulação ra-

zoável de V_{OUT}. A célula de lítio de 3 V satisfaz esse dois requisitos.

Colocando-se a bateria reserva no lado de entrada do conversor, isso permite que sua tensão possa ser diferente da tensão de saída exigida, mas ela deverá ser capaz de resistir às altas correntes que o circuito drena durante transientes de start-up ou de carga. Neste caso, uma bateria de lithium-thionyl chloride representa a melhor escolha para aplicações de longa vida. Ela poderá formar uma "porta OR a diodo" com a célula solar, ou ainda ser chaveada pela "chaves a MOSFET"conforme mostra a figura 2.

Conversor de 5 V opera suavemente a partir de uma variedade de fontes de entrada

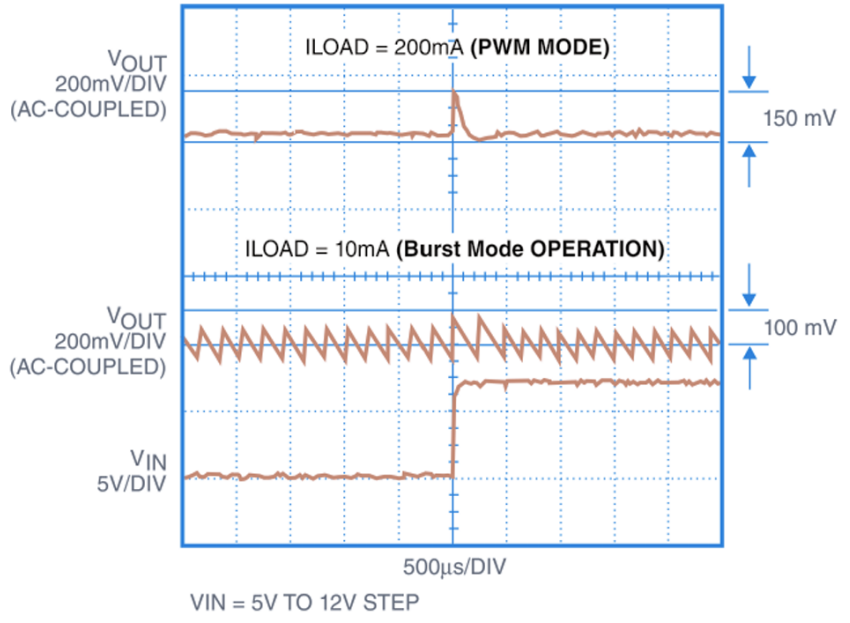
A capacidade do LTC3129-1 poder funcionar com alto rendimento e com uma larga faixa de cargas e tensões de entrada (usando um nº mínimo de componentes externos) é ilustrada pelo circuito da **figura 3**.

Neste exemplo, a saída que foi programada para 5 V usando-se os pinos VS1 – VS3, poderá ser energizada a partir da entrada USB 5 V, ou de uma variedade de opções de baterias, ou mesmo de um adaptador para tomada de 3 V a 15 V.

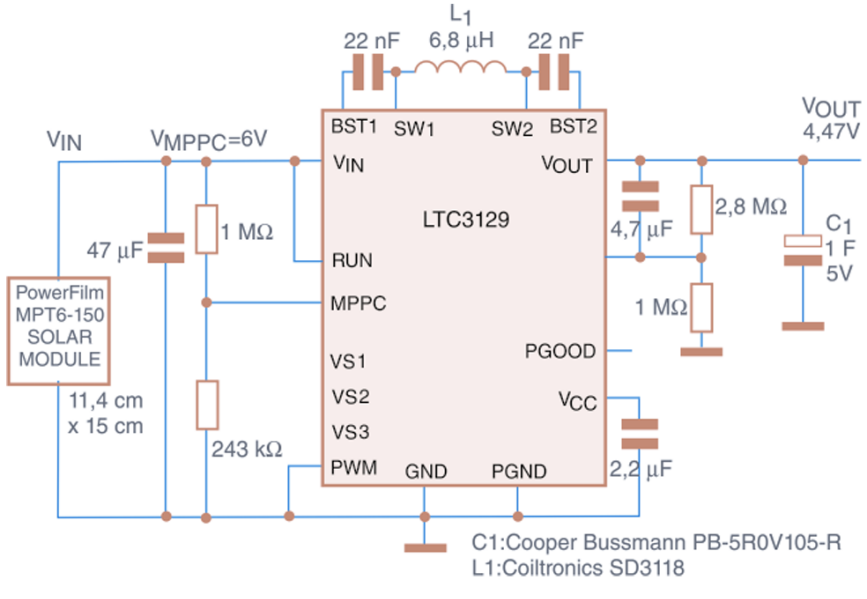
A baixa corrente quiescente I_Q do LTC3129-1, justamente de 1,3 µA em modo sleep, combinada com um divisor interno de *feedback* de alta resistência faz com que o CI mantenha um alto rendimento para uma larga faixa de cargas, veja o gráfico da **figura 4.** Com uma corrente de carga de 100 µA, o rendimento é de aproximadamente 80% sobre praticamente toda a faixa de V_{IN}.

A resposta a um degrau da linha (V_{IN} é alçada de 5 V a 12 V) é mostrada na **figura 5,** onde V_{OUT} é medido para ambas condições de carga (intensa e leve). Para uma carga de 200 mA, o componente está operando no modo PWM e a sobrecarga de V_{OUT} é de apenas 150 mV (3%). Já com uma carga de 10 mA, ele funciona no modo Burst com um ripple burst de 100 mVpp (2%) e uma sobrecarga em V_{OUT} menor que 100 mV devido ao degrau de linha.

O pino V_{CC} é a saída de um regulador LDO interno que gera uma tensão nominal de 3,9 V de V_{IN} para alimentar o CI. O LDO foi projetado de maneira que ele



F5. Resposta do Conversor de 5 V ao transiente de linha (step).



F6. Carregador de supercapacitor alimentado por módulo solar, que funciona com máximo controle do nível de potência (MPPC).

possa ser controlado externamente para níveis acima de 5 V. Neste exemplo, um diodo boot-strap é colocado entre V_{OUT} e V_{CC}, conforme visto na figura 3.

O acréscimo desse diodo externo traz duas vantagens:

A primeira é que ele aumenta o rendimento com V_{IN} baixo e altas correntes de carga, uma vez que fornece uma tensão de *gate drive* maior para as chaves internas, diminuindo sua $R_{DS(on)}$. E, por outro lado, com alto V_{IN} e cargas leves ele também aumenta o rendimento pela redução da perda de potência do LDO interno utilizado para gerar V_{CC}.

A segunda vantagem é que colocação desse diodo permite operação desde um V_{IN} mais baixo.

Após o start-up, se o V_{CC} for mantido acima do valor mínimo de 2,2 V (pela tensão de saída, neste caso), então, o conversor poderá funcionar com uma tensão de entrada menor (abaixo de 1,75 V) onde o limiar fixado de V_{IN} (UVLO) será alcançado. Essa capacidade estende o range de tensões utilizáveis, embora para possibilitar isso, a operação deverá ser feita com duas baterias alcalinas descarregadas. Observe que, se a tensão de bateria for menor que 2,4 V e o conversor estiver em shutdown (ou se V_{OUT} estiver cortado), o CI será incapaz de reiniciar.

Conversor Solar para Outdoor/Carregador com MPPC

Os CIs LTC3129 e LTC3129-1 têm uma característica conhecida como MPPC – *Maximum Power Point Control*, a qual previne o conversor contra falha da tensão de entrada V_{IN}, quando ele estiver operando com uma fonte limitada em corrente.

O *loop* de controle MPPC funciona com a redução da corrente média no indutor, comandada pelo conversor, e, portanto, isso mantém a mínima tensão V_{IN} programada com carga. Essa tensão é aplicada usando-se um divisor resistivo externo, ligado entre V_{IN} e o pino MPPC, conforme mostra o exemplo do carregamento de um supercapacitor na **figura 6**.

A maioria das aplicações que utilizam o MPPC diz respeito ao carregamento de um grande capacitor de armazenagem, ou à carga lenta de uma bateria, a partir da operação de uma célula solar.

É importante notar que, quando esses CIs estão no controle MPPC, o funcionamento em Modo Burst fica inibido e a corrente quiescente de V_{IN} alcança vários miliampères, desde que eles estejam chaveando continuamente na frequência de 1,2 MHz. Portanto, o MPPC não é apropriado para uso com fontes incapazes de fornecer um mínimo de 10 mA.

Segurança Intrínseca com uso de MPPC

A característica MPPC poderá ser usada em outras aplicações, incluindo aquelas projetadas para segurança intrínseca onde a fonte de entrada tem um resistor de limitação de corrente entre ela e o conversor CC/CC.

Nesse caso, o laço de MPPC previne o CI de drenar demasiada corrente, especialmente durante a partida, quando o capacitor de saída está sendo carregado e a tensão de entrada falhando. Um exemplo deste conversor é exibido na **figura 7,** no qual a tensão de entrada é mantida no mínimo de 3 V fixado pelo divisor MPCC.

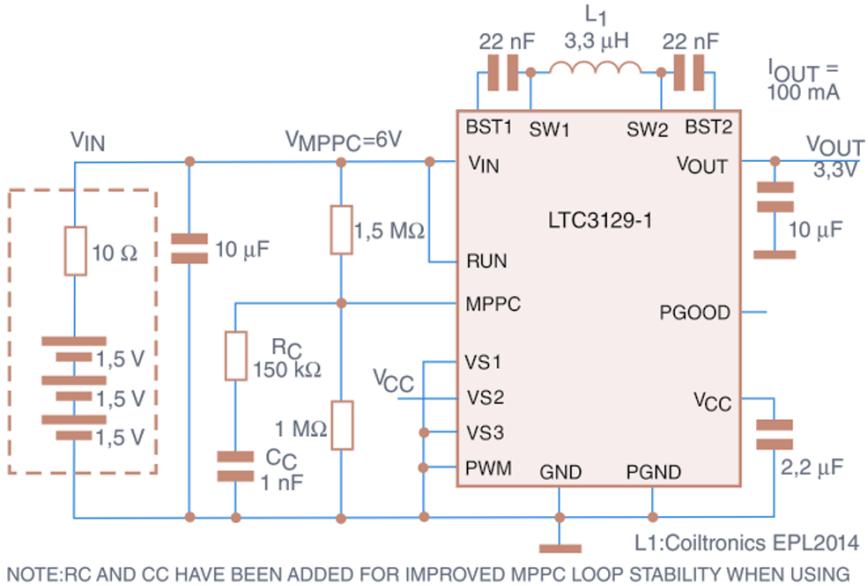
Dessa forma, devido ao valor do capacitor de entrada ser limitado justamente em 10 µF por segurança (abaixo do mínimo recomendado de 22 µF, quando se usa MPPC), um circuito de compensação RC é acrescentado no pino MPPC para aumentar a margem de fase do laço.

Note que a característica MPPC poderá ser utilizada, inclusive, para fixar um valor máximo da corrente de entrada. Escolhendo-se um valor adequado para o resistor de entrada em série e fixando-se a tensão MPPC em valor inferior da tensão da fonte de entrada, a máxima corrente de entrada fica limitada por: $I_{IN} = (V_{Source} - V_{MPCC})/ R_{SERIES}$.

Conclusão

Os CIs LTC3129 e LTC3129-1 oferecem uma performance excelente em baixas potências e a flexibilidade exigida das fontes de alimentação para os sensores wireless e outros instrumentos eletrônicos portáteis. Sua ultra baixa corrente quiescente de 1,3 µA e seu alto rendimento de conversão podem estender o tempo de vida da bateria indefinidamente, se esta for usada combinadamente com coleta de energia.

Uma escolha certa de circuitos MPPC permite a otimização do desempenho de potência em uma larga faixa de fontes de alimentação. O alcance da expansão das aplicações de monitoração wireless requer facilidade de uso, um ótimo rendimento e soluções flexíveis de conversores CC/CC. Estes CIs da Linear Technologies estão prontos para responder a tais desafios. E



F7. Conversor de 3,3 V usando MPPC para uma aplicação de segurança intrínseca.

Utilização do Protocolo SPI
na Comunicação entre Microcontroladores

Este artigo demonstra a aplicação do protocolo SPI na comunicação síncrona entre dois microcontroladores PIC16F877A, da Microchip. Um dos microcontroladores terá o papel de mestre e o outro terá o papel de escravo. No mestre, um botão irá incrementar uma contagem que será enviada ao microcontrolador escravo encarregando-o de exibir o valor acumulado em um display de 7 segmentos.

Flávio da Cruz
Professor de Controlador Programável e Tecnologia de Comunicação - SENAI

Atualmente, o controle de muitos sistemas não ocorre de forma centralizada em um único dispositivo de controle. Em muitos processos industriais contínuos (ou discretos), o controle ocorre de forma distribuída.

Em sistemas com controle embarcado é possível aplicar controle distribuído, em que *drives* de atuadores e sensores de um determinado sistema se comuniquem através de protocolos síncronos. Exemplos de protocolos síncronos para comunicação são o Inter-Integrated Circuits, I²C, desenvolvido pela Philips, e o Serial Peripheral Interface, desenvolvido pela Motorola.

O canal de comunicação síncrono tem, além das vias de transmissão e recepção, uma via para transmissão de *clock*. Com base nos pulsos de clock, o recebedor lê e captura a informação em uma das bordas do período do clock. O recebedor não estará pronto para ler uma informação no canal de dados até o próximo pulso de clock. Segundo AXELSON, o fato dos dados e os pulsos de clock serem gerados unicamente pelo transmissor garante o sincronismo da transmissão mesmo quando os circuitos geradores de clock são precários.

Revistas especializadas têm muitos exemplos atuais do uso de protocolos síncronos para a transmissão de dados.

Avinash, D. *et al* descrevem a aplicação do protocolo SPI em sistema de monitoramento da pressão em pneus de veículos.

Fredenic *et al* combinam o protocolo SPI com fibra óptica, "*Power-over-fiber*" em um caso de comunicação de 10 km de distância com baixo consumo para aplicações no fundo mar.

Outro exemplo de aplicação sofisticada de protocolos síncronos é descrito por Parag Parandkar e Deepak Suryavanshi que, aplicam o protocolo I²C na transmissão de imagens de câmeras de vigilância em tempo real.

Microcontrolador PIC16F877A

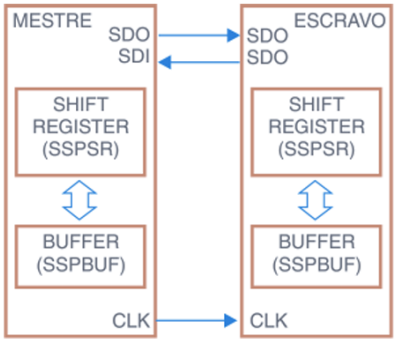
Os microcontroladores têm cada vez mais funcionalidades como modulação de largura de pulso (PWM), conversão de sinal analógico em digital (ADC), sistemas de comunicação serial e paralela, entre outras. Essa funcionalidades nos microcontroladores são tratadas como recursos periféricos.

O microcontrolador usado nesse artigo é o PIC16F877A, da Microchip. Esse microcontrolador tem três áreas de memória:

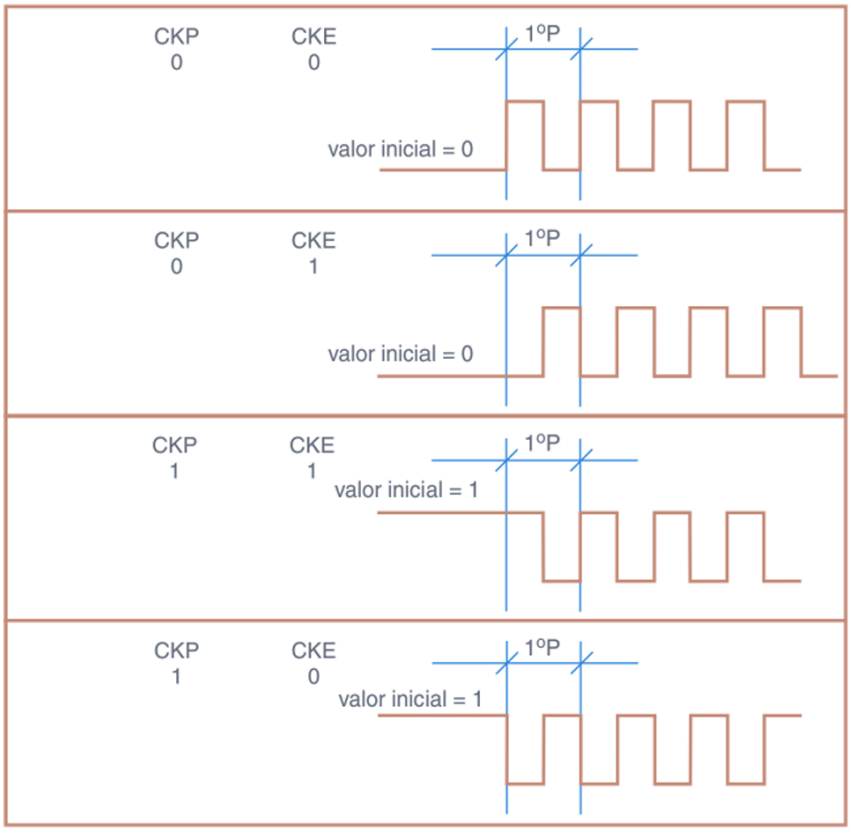
- Memória de programa;
- Memória de dados;
- Memória EEPROM.

A memória de programa tem um contador de 13 bits capaz de endereçar 8 kwords (2^13), sendo que cada palavra de controle tem 14 bits. Essa memória é organizada em 4 páginas.

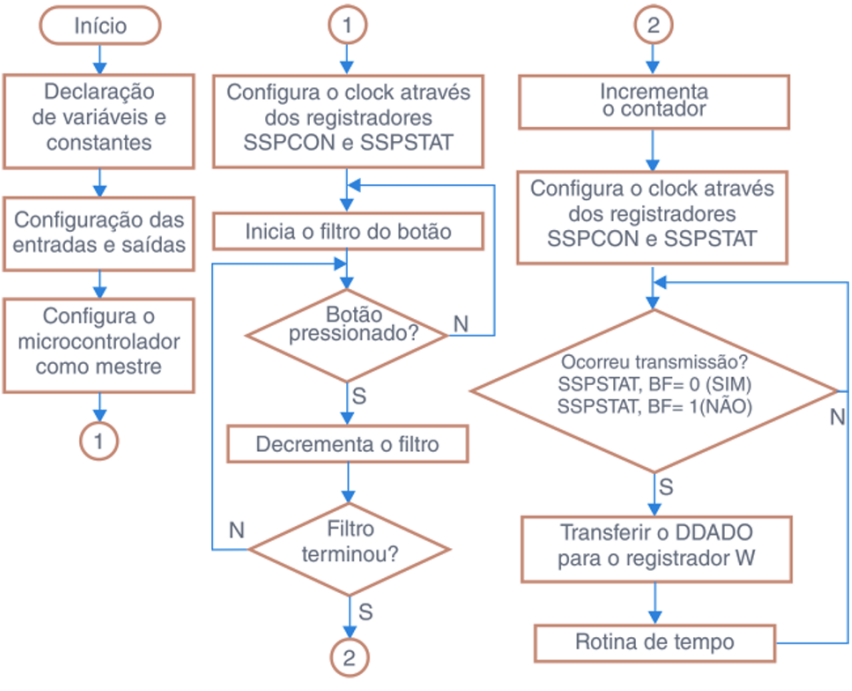
A memória de dados é dividida em 4 bancos e contém os registradores que executam as instruções da memória de programa. Cada posição de 8 bits na memória de dados é chamada de registrador, existem registradores de uso geral e os



F1. Funcionamento do protocolo.



F2. Configurações do clock.



F3. Fluxograma do spi_mestre.

registradores de uso específico. Nos registradores de uso geral os programadores reservam espaços para o processamento dos algoritmos conforme a necessidade. Os registradores de uso específico têm posição fixa nas posições iniciais dos bancos de memória. Esses registradores configuram e controlam o funcionamento do microcontrolador e seus recursos periféricos.

O recurso periférico usado na comunicação síncrona é de *Master Synchronous Serial Port* (MSSP). Esse recurso periférico usa quatro registradores. O controle e configuração da comunicação síncrona é feita pelos registradores SSPSTAT e SSPCON. A transferência dos dados é feita pelo registrador *Syncrhonous Serial Port Shift Register* (SSPSR). Os dados recebidos são acumulados no registrador *Syncrhonous Serial Port Buffer* (SSPBUF).

A edição nº 413 da Revista Saber Eletrônica, da Editora Saber, traz um artigo com descrições mais detalhadas do microcontrolador.

Uma breve descrição do Protocolo SPI

O protocolo SPI é um padrão de comunicação serial síncrona. Uma característica importante desse protocolo é que a transmissão e o recebimento ocorrem sempre simultaneamente. Outras características são:

- Comunicação ponto a ponto entre um controlador mestre e um controlador escravo.
- A forma de comunicação é síncrona, ou seja, o mestre gera o sinal de *clock* que é transmitido junto com a informação.
- Comunicação de dados com 8 bits de comprimento.

O protocolo usa um canal *full duplex* com três vias: transmissão, recebimento e clock.

A **figura 1** mostra o funcionamento esquemático entre o microcontrolador MESTRE e o ESCRAVO.

O controlador mestre é responsável pela geração do clock, que é iniciado no início da comunicação. A cada pulso do clock, um bit é lido na porta *serial data input SDI* escrito no registrador interno SSPSR que é um registrador de deslocamento (*flip-flops* tipo D ligados em série). Esse registrador não pode ser acessado diretamente.

Simultaneamente, o bit mais significativo do dado é enviado para a porta serial data *output* SDO, ou seja, a transmissão e o recebimento sempre acontecem ao mesmo tempo.

Ao final de 8 períodos do clock, o registrador de deslocamento esta cheio e o dado é transferido para o registrador SSPBUF que é o acumulador do dado.

Configurações

Como os microcontroladores têm cada vez mais funções periféricas usando o mesmo encapsulamento, é necessário uma extrema atenção com a configuração para evitar conflito entre as funcionalidades.

A configuração dos recursos periféricos do microcontrolador é feita através de registradores de uso especial. Esses registradores têm posição fixa na memória de trabalho do microcontrolador. O microcontrolador escolhido para esse trabalho tem o recurso periférico MASTER SYNCHRONOUS SERIAL PORT (MSSP) e é esse recurso que suporta a aplicação do protocolo SPI. Para o uso desse recurso periférico são necessárias 4 configurações básicas:

- Configurar a direção das portas do microcontrolador.
- Definir o papel do microcontrolador, MESTRE ou ESCRAVO e a frequência do clock de transmissão serial.
- Definir a forma de amostragem dos bits de dados, *sampling*.
- Definir a polaridade inicial do clock e posição da onda no período. Essas configurações estão melhor descritas a seguir, na **tabela 1**.

Deve-se determinar a função do microcontrolador na comunicação, MESTRE ou ESCRAVO e taxa de transmissão através do registrador de uso especial SSPCON. Veja a **tabela 2**.

A cada período do clock de transmissão serial um bit de dado é lido. Essa leitura pode ocorrer na borda intermediária do clock ou na última borda. Para determinar essa opção, o bit mais significativo do registrador de uso especial SSPSTAT precisa ser configurado da seguinte forma:

- SSPSTAT,SMP = **0**: A leitura do dado é feita na borda intermediária do período.
- SSPSTAT,SMP = **1**: A leitura do dado é feita no final do período. Essa opção só é possível no MESTRE.

O início da comunicação depende de uma instrução programada no microcontrolador MESTRE. Quando a comunicação começa, uma sequência de pulsos de clock é transmitida do MESTRE para o ESCRAVO. O pino de clock no microcontrolador MESTRE está conectado ao pino de clock no microcontrolador ESCRAVO. No mestre esse pino é saída e no ESCRAVO esse pino é entrada.

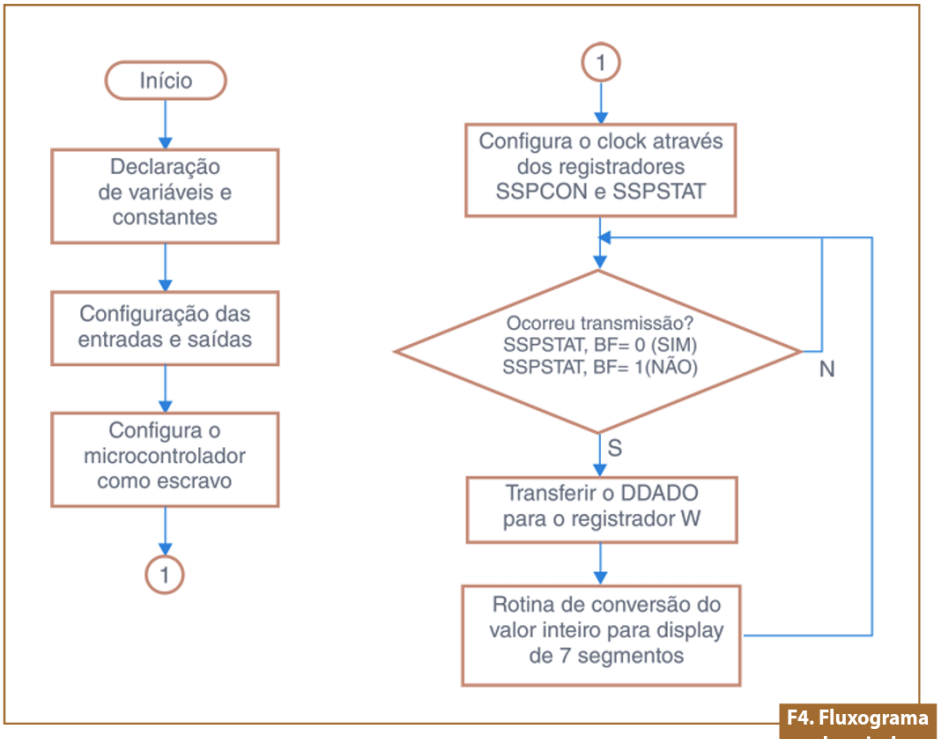
Através do *Clock Polarty Select Bit* do registrador especial SSPCON é possível determinar a polarização inicial do pino de clock. Se o bit CKP (bit 4 do registrador SSPCON) for 1, a polarização inicial será um valor de tensão próximo a 5 volts. Se CKP for 0, a polarização inicial do pino CKP terá um valor próximo de 0. É possível, também, escolher se a transmissão do dado ocorrerá na borda de subida ou na borda de descida do clock serial através do bit CKE do registrador SSPSTAT.

Porta/Pino	Descrição
PORTC, bit 4, Serial Data In SDI	É configurado automaticamente pelo sistema como entrada.
PORTC, bit 5, Serial Data Out SDO	Deve ser configurado manualmente como saída através registrador TRISC.
PORTC, bit 3, Serial Clock SCK	Deve ser configurado no TRISC da seguinte forma: No mestre deve ser configurado com saída. No escravo deve ser configurado como entrada.
PORTA, bit 5, Slave Select/SS	Esse pino serve para habilitar ou desabilitar a comunicação. 0 habilita; 1 desabilitada.

T1. Configurações básicas.

Valor	Função do microcontrolador e frequência
0000	SPI Master com clock =FOSC/4
0001	SPI Master com clock FOSC/16
0010	SPI Master com clock = FOSC/32
0011	SPI Master com clock = metade do TMR2
0100	SPI Slave com SS habilitado (somente para mais de um escravo)
0101	SPI Slave com SS desabilitado

T2. Função e taxa de transmissão do microcontrolador.

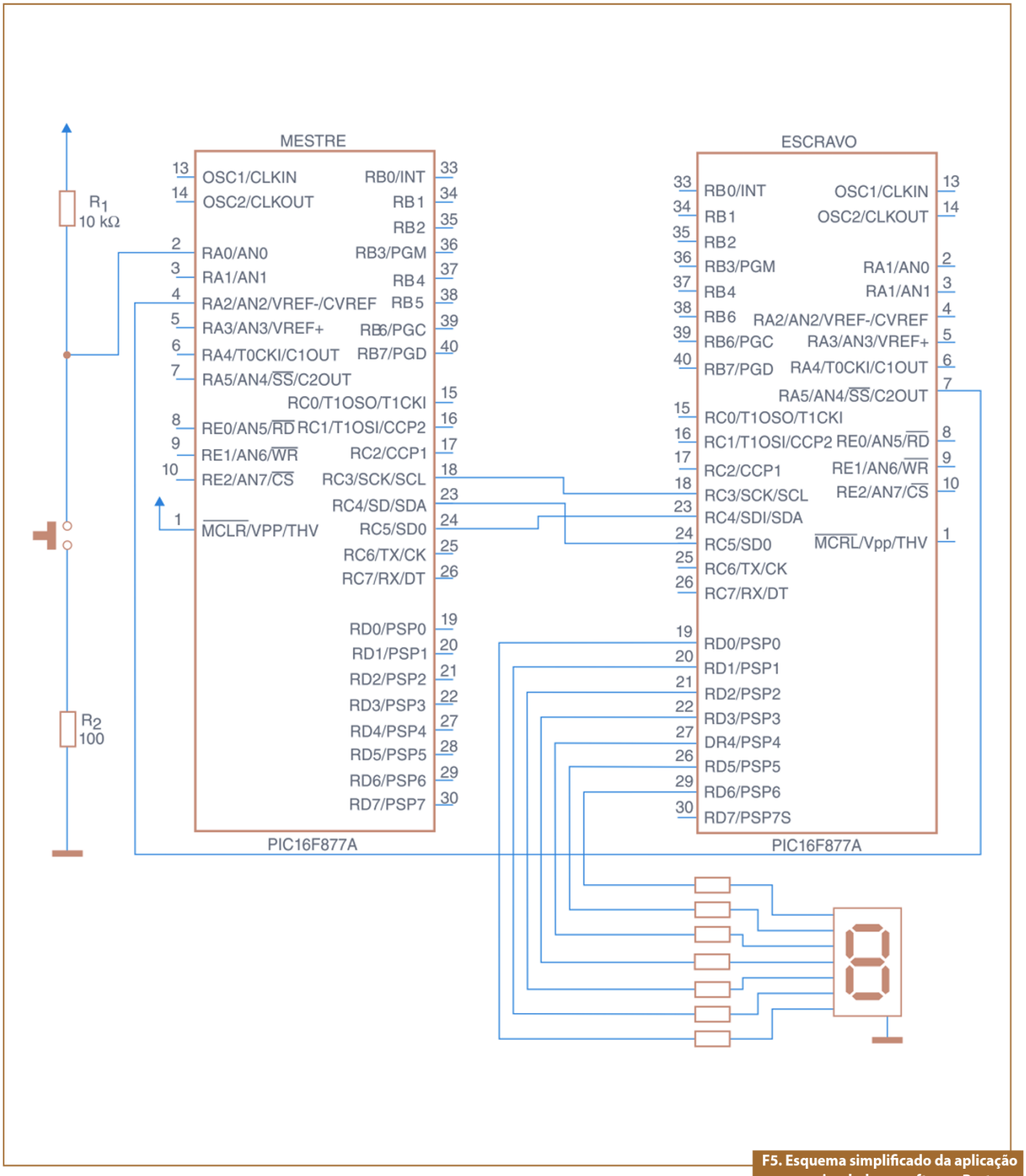


F4. Fluxograma do spi_slave

O funcionamento do clock no protocolo SPI depende, então, de configurações feitas em dois registradores especiais diferentes SSPCON e SSPSTAT. Um cuidado adicional nessa configuração é que esses registradores se encontram em bancos de memória diferentes. A **figura 2** mostra as possibilidades de funcionamento do clock.

Programas

Procurou-se fazer programas enxutos para que a configuração e o algoritmo de transmissão e recepção ficassem claros, aprimoramentos valiosos desses programas podem ser encontrados no livro "Conectando o PIC" e no livro "Desbravando o PIC", ambos da editora Erica. Tanto o programa MESTRE quan-



F5. Esquema simplificado da aplicação simulada no software Proteus.

to o programa ESCRAVO começam com a declaração de variáveis e constantes, então se configura as entradas e saídas do PORTC e do PORTA e se configura os registradores SSPSTAT e o SSPCON que regulam a comunicação síncrona.

No programa MESTRE, um teste verifica se o botão está pressionado. Se estiver, um filtro assegura que o contador só será incrementado uma única vez. A cada pulso no botão, uma variável de contagem é incrementada em uma unidade e transferida para o *buffer* de transmissão, registrador SSPBUF. O módulo periférico do microcontrolador automaticamente transfere o valor do buffer para o registrador de deslocamento SSPSR e a transmissão inicia. Como explicado anteriormente, a transmissão é bidirecional, o bit 0 do registrador SSPSTAT sinaliza o final da transmissão. O fluxograma do programa mestre está na **figura 3**.

O início do programa do microcontrolador ESCRAVO é semelhante ao programa do microcontrolador MESTRE. A configuração dos microcontroladores é interdependente. O laço de programa verifica se houve recebimento de dados vindos do MESTRE, o bit 0 do registrador SSPSTAT sinaliza a ocorrência de uma transmissão. A cada conjunto de 8 bits recebidos, uma rotina converte o valor acumulado no registrador SSPBUF para um *display* de sete segmentos. Veja a **figura 4**.

Conclusão

O protocolo SPI utiliza apenas 4 fios e pode ser usado para transmitir dados entre sistemas microcontrolados com eficiência. Substituindo-se a contagem de pulsos no microcontrolador mestre por um sinal analógico, podem-se transferir leituras que estão feitas por um dispositivo para um display que está em outro ponto.

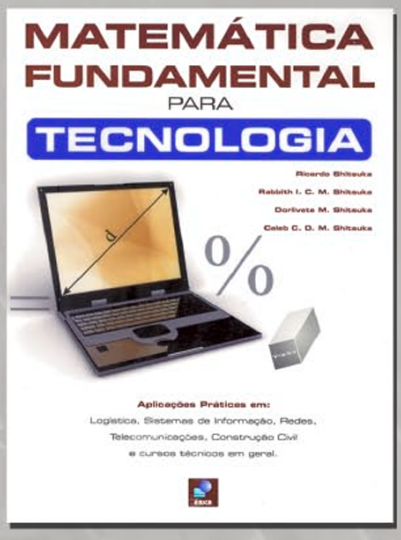
O microcontrolador escravo está usando uma parcela pequena de sua capacidade e, aplicações mais sofisticadas poderiam ser atribuídas, como, por exemplo, a exibição do valor recebido em um display monocromático. Um esquema simplificado da aplicação simulada no software PROTEUS está na **figura 5**.



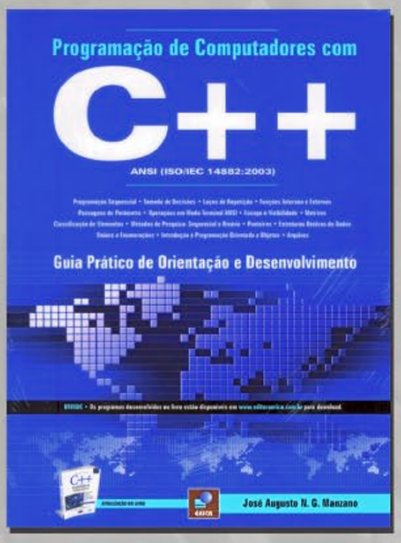
Projetos de Circuitos Digitais com FPGA

Mostra o emprego de novas ferramentas computacionais no desenvolvimento de projetos por meio de exemplos e casos práticos do dia-a-dia do projetista de circuitos digitais.
Destina-se a estudantes, professores, mestrandos e doutorandos, técnicos, autodidatas e profissionais da área.

Outros Destaques









novasaber
www.novasaber.com.br

Aquisição de dados com Smartphones e Tablets, passo a passo

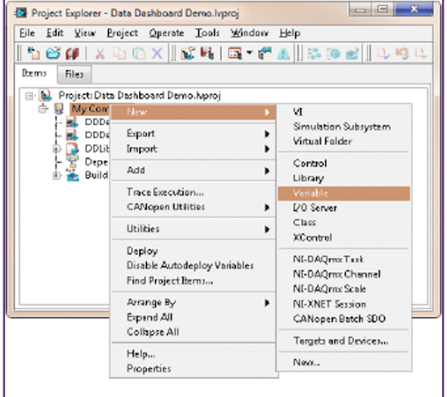
O Data Dashboard permite que você crie sistemas de monitoramento portáteis e customizados por meio de aplicações desenvolvidas no LabVIEW, então, você poderá criar painéis de indicadores para exibir os valores de variáveis compartilhadas publicadas na rede, e indicadores implementados no serviço Web do LabVIEW, como por exemplo: tabelas, medidores, caixas de texto e LEDs. O Data Dashboard está disponível para IOS e Android.

Guilherme Kenji Yamamoto
Renan Airosa Machado de Azevedo
National Instruments

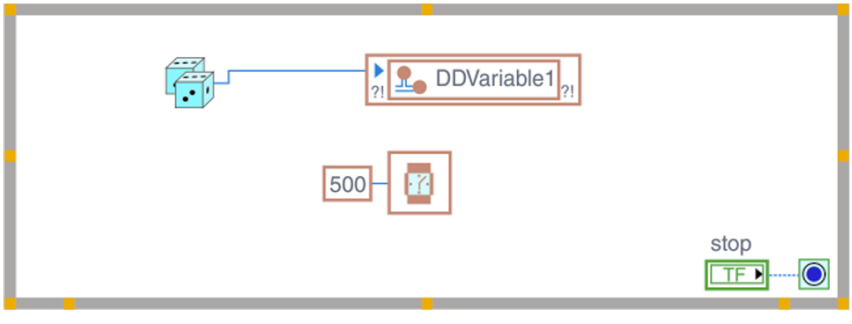
Como criar e implementar variáveis compartilhadas (shared variables)

Antes de você criar um painel de indicadores, crie uma aplicação no LabVIEW no qual possa gerenciar o código e as variáveis compartilhadas publicadas na rede (network-published shared variables).

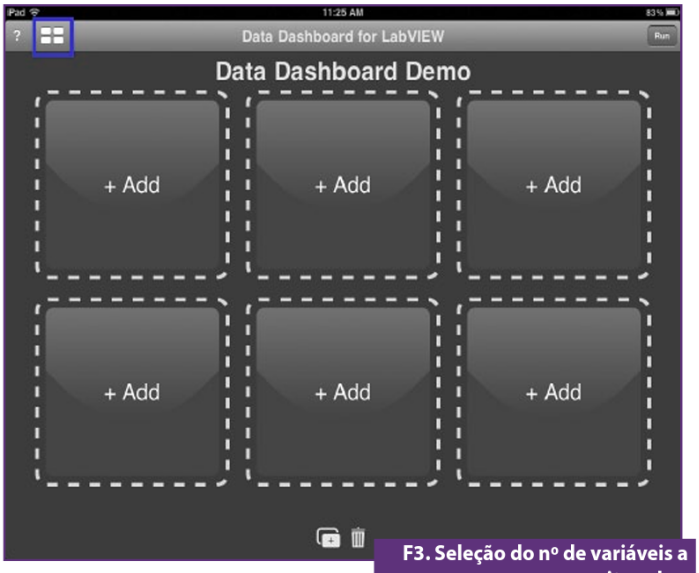
- Selecione *"File»New Project"* para criar um novo projeto no LabVIEW.
- Adicione uma variável compartilhada (shared variable) ao projeto. Clique com o botão direito em *"My Computer"* na janela *"Project Explorer"* e selecione *"New»Variable"*, como mostra a **figura 1**.
- Clique com o botão direito na *"shared variable"* e selecione *"Rename*. Renomeie a shared variable" para *DDVariable1*.
- Clique com o botão direito em *"My Computer"* e selecione *New»VI* para criar um novo VI e adicioná-lo ao projeto. Esse VI irá atualizar periodicamente o valor da shared variable com um valor aleatório.
- Salve o projeto.
- Adicione um *While Loop* ao diagrama de blocos do VI.
- Coloque a função *Random Number (0-1)* e a função *Wait (ms)* dentro do While Loop.
- Clique com o botão direito no terminal de entrada da função Wait (ms) e selecione *Create»Constant*. Coloque o valor da constante como 500 para que o valor seja atualizado a cada meio segundo.
- Arraste para o diagrama de blocos a shared variable que você criou na janela "Project Explorer". Clique com o botão direito na *shared variable* e selecione *"Access Mode»Write"*.



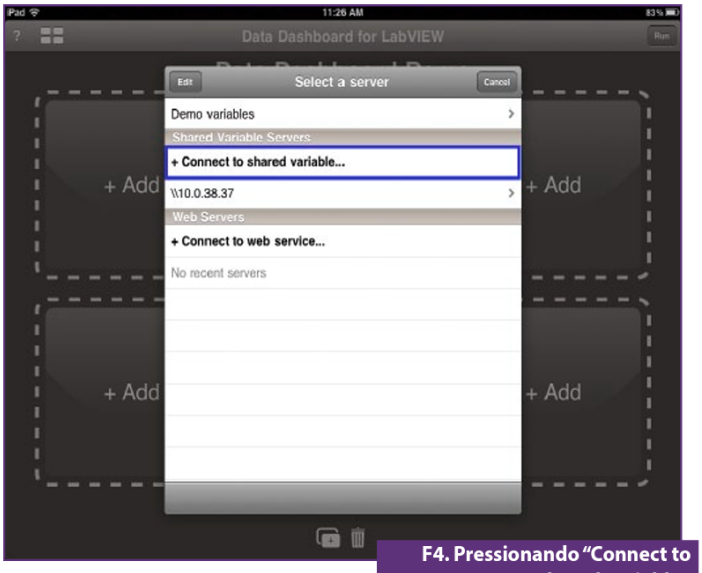
F1. Criação/ Implementação de variáveis compartilhadas.



F2. Diagrama de blocos.



F3. Seleção do nº de variáveis a serem monitoradas.



F4. Pressionando "Connect to shared variable".



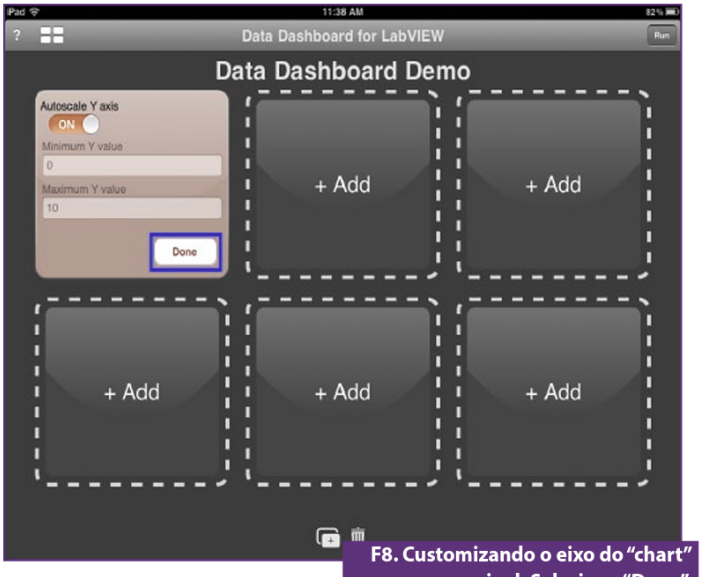
F5. Colocando o endereço do IP do computador onde a variável está hospedada.



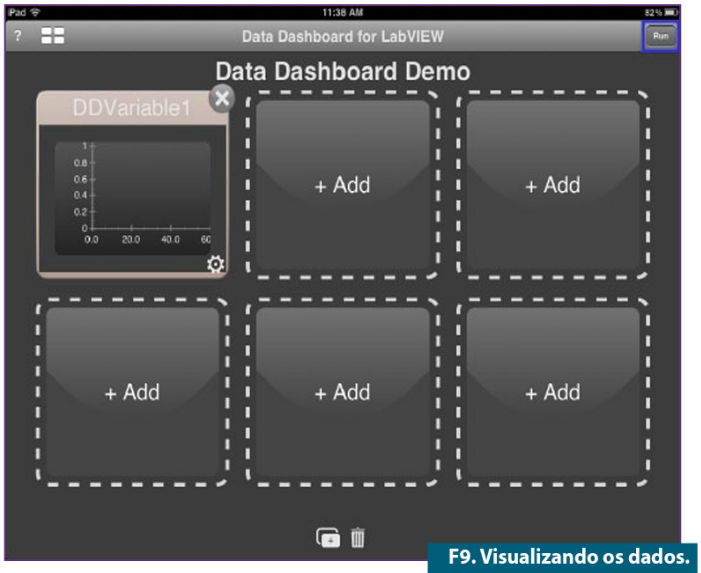
F6. Escolhendo a "shared variable".



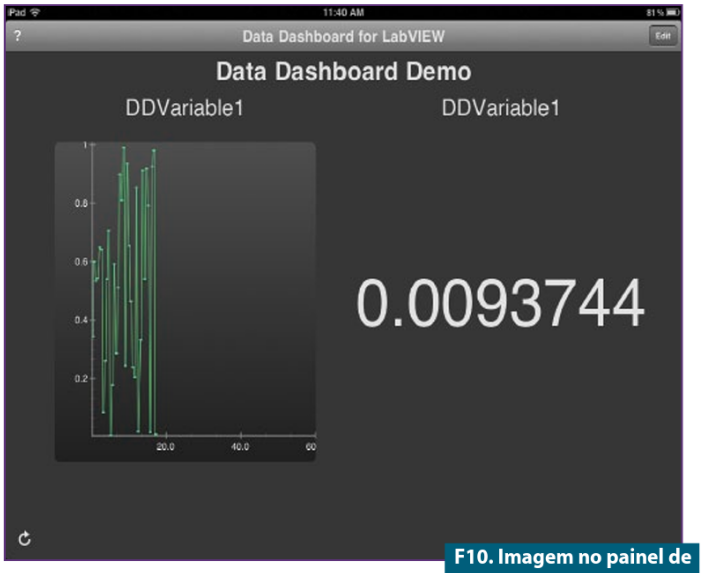
F7. Selecionando o indicador chart para exibir dados da variável.



F8. Customizando o eixo do "chart" no painel. Selecione "Done".



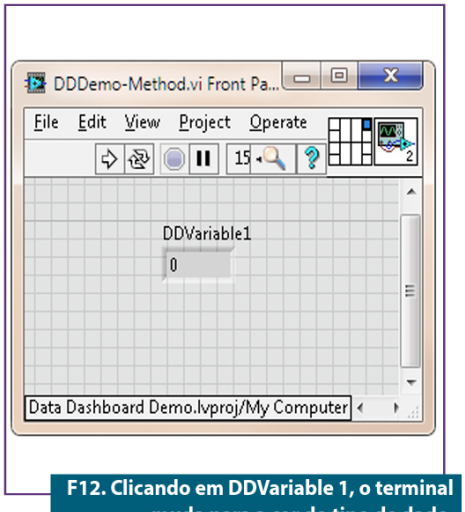
F9. Visualizando os dados. Aperte "Run".



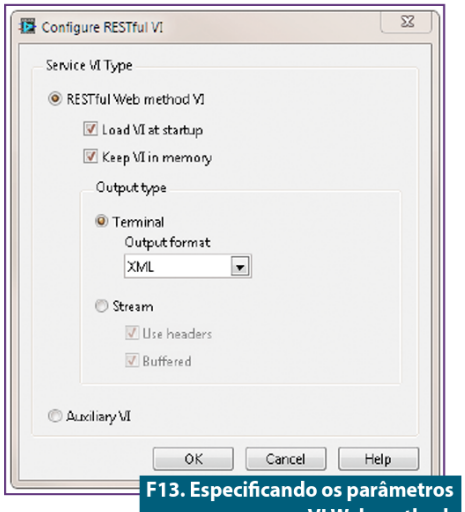
F10. Imagem no painel de indicadores.



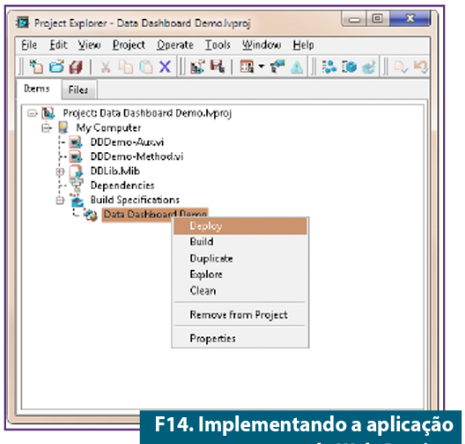
F11. Editando o "label" do indicador como DDVariable 1.



F12. Clicando em DDVariable 1, o terminal muda para a cor do tipo de dado.



F13. Especificando os parâmetros para o VI Web method.



F14. Implementando a aplicação de Web Service.

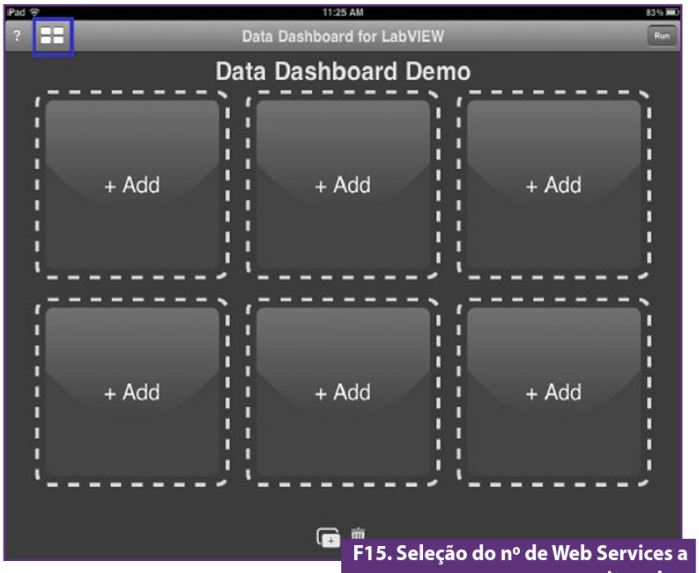
- Conecte o terminal de saída da função Random Number (0-1) ao terminal de entrada da shared variable.
- Clique com o botão direito no terminal de entrada do terminal de parada do *While Loop* e selecione "*Create»Control*", para criar um botão que para e inicia o programa. O seu diagrama de blocos deve ficar semelhante ao exemplo dado na **figura 2**.
- Você deve realizar um *deploy* na shared variable antes que possa exibir seu valor no *Data Dashboard*. Na janela "*Project Explorer*", clique com o botão direito na biblioteca (*library*) do projeto que contém a shared variable e selecione Deploy.
- Compile e execute a aplicação para atualizar continuamente a shared variable.

Como monitorar os dados de uma Shared Variable em um painel de indicadores

Abra o aplicativo Data Dashboard no seu dispositivo e aperte o botão pattern para selecionar o número de variáveis a serem monitoradas, como mostrado no exemplo da **figura 3**. Aperte +Add para adicionar um indicador, para que este possa exibir os dados da shared variable implementada, e, em seguida, aperte *Connect to shared variable* conforme ilustrado na **figura 4**.

Coloque o DNS do servidor (ou endereço de IP do computador) onde a shared variable está hospedada, como exibe a **figura 5**, e, após isso, aperte Connect.

Selecione o nome da biblioteca que contém a shared variable implementada, então escolha a shared variable em "*Select a variable*", conforme visto na **figura 6**. O tipo de dado de cada shared variable é listado abaixo do nome da shared variable.



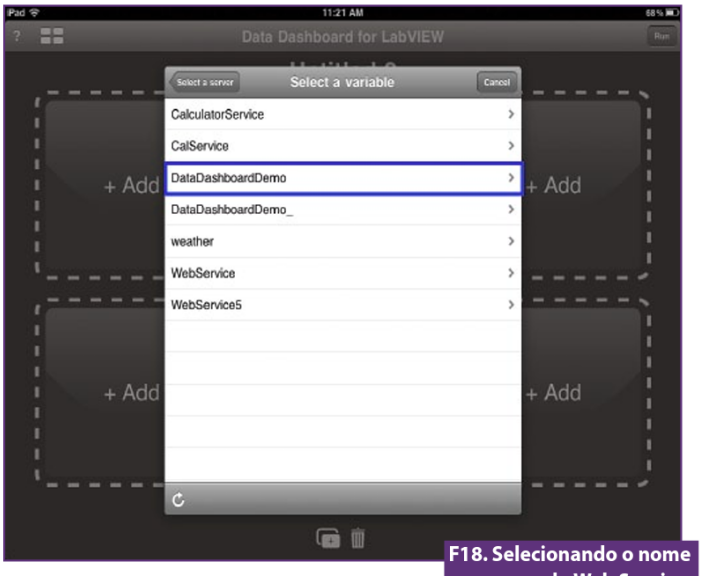
F15. Seleção do nº de Web Services a serem monitoradas.



F16. Pressionando "Connect to web service".



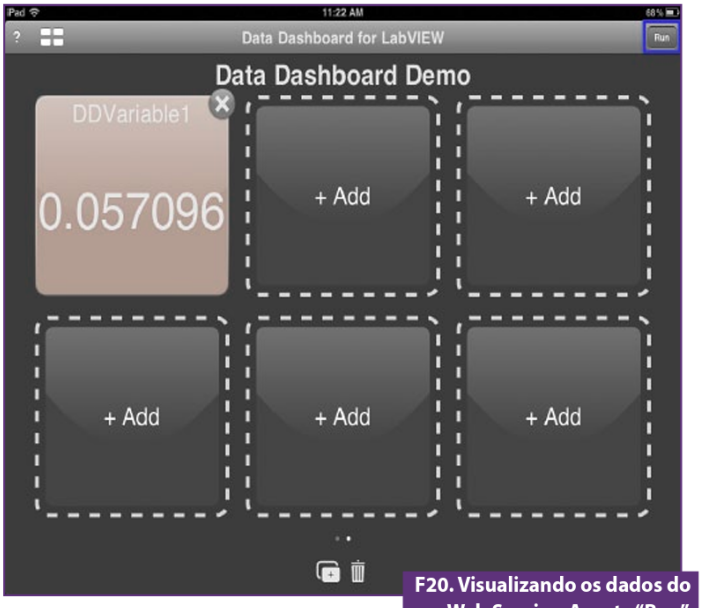
F17 Colocando o endereço do IP do computador que está hospedando o Web Service.



F18. Selecionando o nome do Web Service.



F19. Prosseguindo na escolha do Web Service a ser monitorado.

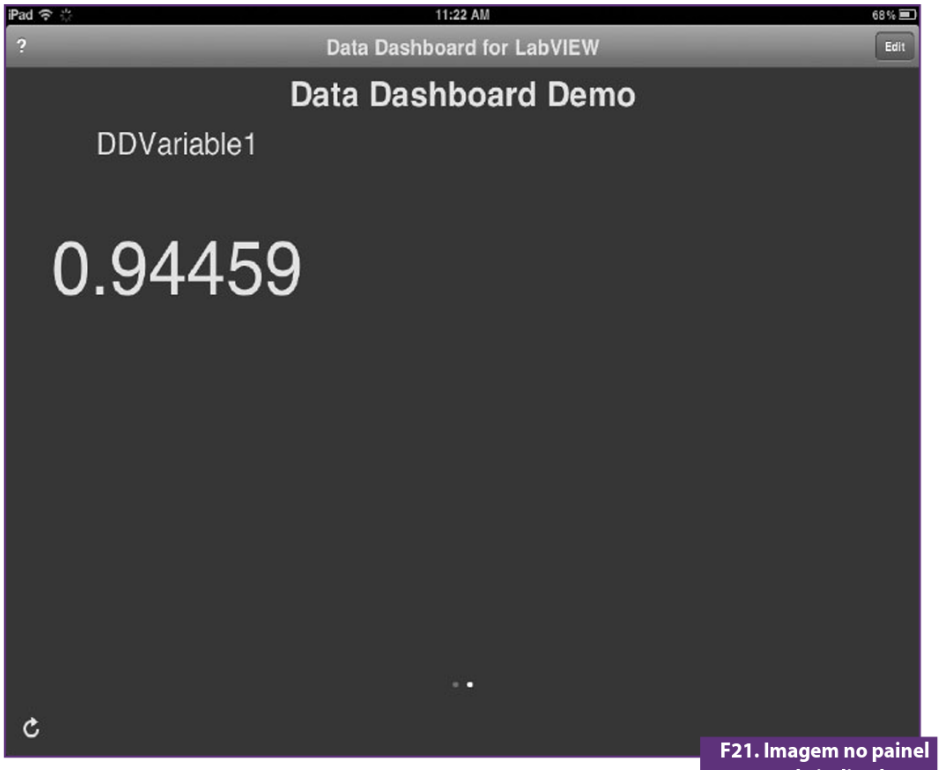


F20. Visualizando os dados do Web Service. Aperte "Run".

Selecione o indicador chart para exibir os dados da shared variable, veja na **figura 7**. O aplicativo Data Dashboard somente permite que você selecione indicadores que são compatíveis com o tipo de dado da shared variable selecionada. Neste exemplo, a shared variable implementada é um número de ponto flutuante de precisão dupla, que podemos exibir em um indicador *string*, *gauge* ou *chart*.

Você pode customizar o eixo do chart no painel. Para este exemplo, utilize os valores padrão, e selecione Done (**figura 8**).

O aplicativo Data Dashboard exibe uma prévia do painel de indicadores que você criou. Aperte *Run* para visualizar os dados publicados pela shared variable implementada (**figura 9**). O painel de indicadores em execução deve se parecer com a imagem da **figura 10**.



F21. Imagem no painel de indicadores.

Como criar e implementar serviços Web

Antes de você criar um painel de indicadores para monitorar um serviço Web (Web Service), crie uma aplicação no LabVIEW para implementar como um Web Service. Neste exemplo, você cria um VI de método Web (Web method) que troca dados com clientes Web (Web clients) pela rede.

Primeiramente, crie um novo projeto no LabVIEW e nomeie o projeto como *Data Dashboard Demo*. Adicione um novo VI ao projeto, nomeando-o como *VI DDDemo-Method*. Coloque uma função Random Number (0-1) no diagrama de blocos e clique com o botão direito no terminal de saída da função Random Number (0-1) e selecione Create»Indicator. Edite a label do indicador e renomeie o indicador como DDVariable1, como mostra a **figura 11**.

Atribua o indicador a um terminal do *connector panel* do VI, assim o Data Dashboard pode monitorar a saída do VI Web method. No painel frontal, selecione um padrão de *connector panel*. Clique em um terminal do *connector panel*. O terminal fica preto quando é selecionado. Clique no indicador DDVariable1 no painel frontal. Uma seleção destaca o indicador e o terminal muda para a cor do tipo de dado para indicar que você conectou o terminal, observe na **figura 12**.

Salve o projeto e crie um novo *Web Service*. Clique com o botão direito em *Build Specifications*, na janela Project Explorer e selecione *New»Web Service (RESTful)*.

Na página *Information*, selecione o nome do Web Service e o seu diretório de destino. Use as configurações-padrão. Na página *Source Files*, use a seta *Add Item* para mover o VI que você criou da árvore *Project Files* para a árvore *Service Vis*. Na caixa de diálogo Configure RESTful VI, especifique os seguintes parâmetros para o VI Web method (**figura 13**).

Compile (*build*) o projeto e expanda a opção Build Specification, depois clique com o botão direito em Data Dashboard Demo na janela Project Explorer, e selecione Deploy para implementar a aplicação de Web Service (**figura 14**).

Como monitorar Web Services em um painel de indicadores

Comece abrindo o aplicativo do Data Dashboard no seu dispositivo. Aperte o botão de padrão para selecionar o número de Web Services a serem monitorados, como mostrado na **figura 15**.

Aperte *+Add* para adicionar um indicador que exibirá os dados do Web Service implementado e Connect to web service, seguindo o exemplo da **figura 16**. Insira o DNS do servidor ou endereço de IP do computador em que está hospedando o Web Service (**figura 17**), e aperte *Connect*.

Selecione o nome do projeto que contém o *Web Service* implementado, então selecione o nome do Web Service (**figura 18**).

Selecione o Web Service a ser monitorado, como mostrado na imagem a seguir. Essas variáveis correspondem às entradas e saídas no connector panel do VI Web service method. O tipo de dados do valor do Web Service é listado abaixo do nome da variável (**figura 19**).

Selecione o indicador string, e então, clique em *Done*. O Data Dashboard exibe uma prévia do painel que você criou. Aperte *Run* para visualizar os dados do Web Service implementado (**figura 20**).

O painel de indicadores em execução deve se parecer com a imagem que esta representada pela **figura 21**. E



TREINAMENTOS PROFISSIONAIS

Agora a SABER está promovendo treinamentos técnicos, fazendo aquilo que mais sabe fazer que é ensinar. Nosso diferencial é saber muito antes, as novidades tecnológicas e preparar um bom material didático teórico, nunca esquecendo da prática (que é a nossa principal característica), para passar para você, leitor.

- **Treinamento Básico e Avançado de Programação de Microcontroladores em linguagem C**
- **Treinamento de Programação de Microcontroladores Atmel / Freescale / Holtek / Infineon / Renesas**
- **Treinamento Prático em SMD com Estações de Solda/Retrabalho**
- **Treinamento de Instrumentação e Controle Aplicados em Caldeiras à Óleo**

Aulas em três sedes:

SÃO PAULO - Rua Jacinto José de Araújo, 309 - Parque São Jorge - Tatuapé - Próximo à estação do Metrô Carrão

CARAPICUIBA - Av. Rui Barbosa, 540 – Piso 1 sala 40 Centro - Carapicuiba próximo à estação de trêm e também em frente de Tamboré e Alphaville.

SÃO BERNARDO DO CAMPO - SP -Rua Ângelo Dusi, 326 - Vila Dusi

CONSULTE A PROGRAMAÇÃO DE 2013

Ligue para: (11) 2095-5330 – Miriam das 10h às 16h.
Mais informações : (11) 2095-5333 - Mari das 10h às 16h.



>> Saiba como a National Instruments apóia a próxima geração da inovação em **ni.com/academic** **(11) 3149-3149**

NATIONAL INSTRUMENTS

©2012 National Instruments. Todos os direitos reservados. National Instruments, NI, e ni.com são marcas registradas da National Instruments.
Os outros nomes de produtos e das empresas mencionadas são marcas registradas e nomes comerciais das respectivas empresas. 05051