

μDX

Série 100

CARTÃO DE REGISTRO DE USUÁRIO

Número de Série: _____
Nome: _____
Empresa: _____ Cargo: _____
Endereço: _____
Cidade: _____ Estado: _____
CEP: _____
Telefone: _____ Data da compra: ____/____/____
Local: _____
Página na Internet: _____ E-mail: _____

Motivação da compra:
 Propaganda de revista ou jornal. Qual: _____

 Feiras de eletrônica e informática. Qual: _____

 Recomendação de amigos ou colegas.
 Mala direta.
 Internet.
 Outra: _____

Interesse de aplicação:
 Uso próprio, residencial ou predial (alarme, automação).
 Uso próprio, educativo ou "hobby".
 Para prestar serviço de automação residencial ou predial.
 Para prestar serviço de automação industrial.
 Para automatizar máquinas ou processos industriais.
 Como parte adicional de máquinas ou equipamentos.
 Outro: _____

Alguma sugestão sobre o μDX, o Programador Gráfico ou sobre este Manual : _____

Preencha, destaque e envie esta página, por carta, à DEXTER. Desta forma você será prontamente informado sobre todas as novidades a respeito do μDX, como novas versões de programas, acessórios disponíveis e artigos de divulgação técnica. Além disso, estará contribuindo para o cadastramento nacional de todos os usuários do μDX e, através de suas sugestões, dando origem a aperfeiçoamentos de projeto e de produto.

DEXTER Indústria e Comércio de Equip. Eletrônicos Ltda.
Av. Pernambuco, 1328, Cj: 309 Porto Alegre RS CEP: 90.240-001

ñ"

Direitos Reservados

Nenhuma parte desta publicação pode ser reproduzida, armazenada ou transmitida sob qualquer forma (mecânica, fotocopiada, gravada), sem permissão escrita da DEXTER.

Embora todos os cuidados tenham sido tomados na elaboração deste manual, a DEXTER não assume qualquer responsabilidade por erros ou omissões contidos neste manual.

O programa PG não pode sofrer engenharia reversa, decompilação ou qualquer outro esforço de cópia e/ou modificação não autorizada expressamente pela DEXTER.

O programa PG é de propriedade da DEXTER Indústria e Comércio de Equipamentos Eletrônicos Ltda. que permite ao usuário realizar cópias de proteção ("backup") e/ou transferir o programa para um único disco rígido.

Todas as marcas e nomes de produtos de outros fabricantes citados neste manual são marcas ou marcas registradas de seus respectivos proprietários.

A DEXTER não se responsabiliza pela montagem e funcionamento dos circuitos descritos no manual. O usuário pode reproduzi-los apenas para seu próprio uso, sendo vedado sua comercialização sem a permissão expressa da empresa.

μ DX
Série 100

μ DX⁺
Série 100

CONTROLADOR PROGRAMÁVEL

Manual de Utilização

Rev. 2.3

Jun/2003

DEXTER Indústria e Comércio de Equipamentos Eletrônicos Ltda.

Av. Pernambuco, 1328 Cj.309 - Porto Alegre - RS - Fones: (0xx51) 3343-2378, 3343-5532

Página Internet: www.dexter.ind.br - E-mail: dexter@dexter.ind.br

Conteúdo

Prefácio	1
Produtos DEXTER.....	4
Apresentação	14
Funcionamento do μ DX	16
Aplicações	19
A Rede Local DXNET	20
Instalação inicial.....	22
Instalando os arquivos do disquete	23
Instalação no Windows 3.x.....	24
Instalação manual em Windows 95 ou 98	26
Copiar o disquete	26
Gerar um atalho	27
Editar propriedades do atalho	30
Instalação automática em Windows 95,98,Millenium	38
Instalação em Windows NT, 2000, XP	42
Instalação e Troca das Pilhas	43
Pilhas recarregáveis	45
Conexões nas Entradas e Saídas.....	46
Sensor tipo Reed-Switch.....	47
Sensor de Umidade	48
Sensor de Luz	48
Detector de tensão AC opto-acoplado	49
Conversor Tensão / PWM.....	50
Supressores de Ruído Elétrico.....	53
Programação em PDE - Utilização do PG	55
Introdução ao PG	56
Área de Programação	57
Menu de Programação	58
Teclas de Operação	59
Área de Status	59
Tecla de Operação - μ DX.....	61
COMPILAR.....	61
ENVIA PROGRAMA	62
EXECUTAR	62
PARAR	62
MONIT. VAR	63
DXNET	64
RESET	65
INFO	65
Porta DXNET	65
Porta Serial	66
ACERTA HORA	68
LÊ HORA	68

Força VAR_DX	69
Força Nodo_DX	69
?	69
Varredura	69
Modo DXNET	70
Modo TX	71
Modo Discar	73
Verifica EEPROM	73
Periférico	73
OK	75
Tecla de Operação - Arquivo	76
LER	76
Gravar	76
Apagar	77
Sai do PG	77
DIR	77
?	78
OK	78
Tecla de Operação - Tela	79
Imprime	79
Regenera	80
Apaga Tudo	80
Opção NODOS	80
Opção TEXTO	80
Opção BLOCOS	81
CORES	81
?	81
OK	81
Opção INGLÊS	82
Calibra	82
Tecla de Operação - MON	83
Monitor	83
Simulador	85
Elaborando Programas	88
Comandos Auxiliares de Edição	98
Tecla F3	98
Tecla F4	98
Convertendo de "LADDER" para PDE	99
Blocos de Instruções	104
Funcionamento dos Blocos de Tempo	106
TEMPO - Relógio	109
TEMPO - MonoEstável	110
TEMPO - Pulso	111
TEMPO - Atraso	112
TEMPO - Oscilador	113
FUNÇÃO - Aritmética/Lógica	
Comparação/Teste	114
FUNÇÃO - PWMIn	116
FUNÇÃO - Bi-Estável	118
CHAVES - Normal Aberto	
Normal Fechado	
Inversor	119
EXPANSÃO - Entrada/Saída de 8 bits	120
EXPANSÃO - Teclado/Display	122

EXPANSÃO - Entrada/Display de 5 dígitos _____	124
DXNET - Envia Nodo	
Envia Variável _____	125
CONFIGURAÇÃO - Nodo EL	
Nodo ED _____	126
CONFIGURAÇÃO - μ DX _____	127
PROGRAMAÇÃO - Energia _____	128
PROGRAMAÇÃO - Rótulo _____	129
SÍMBOLO - Entrada _____	130
SÍMBOLO - Saída _____	131
Operações aritméticas com mais de 8 bits _____	132
Conexão em rede - DXNET _____	136
Características da rede DXNET _____	136
Operação da rede DXNET _____	137
Programação usando a rede DXNET _____	138
Características Técnicas _____	140
ANEXO A - Conectores _____	142
Entradas _____	142
Saídas _____	143
Energia _____	143
Expansão _____	143
DXNET _____	144
ANEXO B - Exemplos de Aplicações _____	145
Minuteria para prédios (Arq: MINUT) _____	145
Alarme residencial (Arq: ALARME) _____	147
Árvore de Natal (Arq: NATAL) _____	147
Irrigação de jardim (Arq: JARDIM) _____	148
Seqüenciador (Arq: SEQUENC) _____	149
Simulador de presença (Arq: TEMPO) _____	149
Controle Térmico (Arq: TEMPERAT) _____	150
Semáforo duplo (Arq: SEMAFORO) _____	151
Controle reserv. industrial (Arq: RESERV) _____	152
Prensa Industrial (Arq: PRENSA) _____	156
Horário de Escolas (Arq: ESCOLA3) _____	159
ANEXO C - Diagramas Esquemáticos _____	160
Opto-acoplador _____	160
Expansão para 8 Entradas e 8 Saídas _____	161
Expansão para Teclado e Display _____	161
Expansão para 8 Entradas e Display _____	162
Fixação Mecânica _____	163
Área ocupada por μ DX + Expansão _____	164
Manutenção _____	165
Garantia _____	166

Prefácio

Esta Revisão 2.3 do Manual de Utilização atualiza o usuário do μDX sobre as novidades técnicas do produto, informa como instalar dispositivos de proteção contra interferências elétricas, acrescenta algumas dicas sobre a programação e uso do PG e lista os produtos DEXTER para o μDX.

Também apresenta o Controlador Programável μDX Série 100+ (Plus), com maior capacidade de blocos, nodos e variáveis.

Além disso, foi incluída uma explicação sobre o simulador do PG; utilização de "carry" em operações de soma ou subtração com mais de 8 bits (operacional para μDXs com firmware igual ou superior a 6.0); verificação do conteúdo da memória do μDX; uso de porta serial ou modem para comunicação remota; e uso de instruções SET e RST (para firmware igual ou superior a 6.9). O software PG versão 4.7 identifica cinco periféricos programáveis: Modem (.MOD), Interface Homem/Máquina (.IHM), Conversor Analógico/Digital (.PWM) e Registrador (.REG). Além disso, permite programar controlador μDX com firmware especial para leitura de código de barras (firmware versão 6.6).

Foram inseridos dois exemplos de programação por diagrama de estados (RESERV.UDX e PRENSA.UDX). É altamente recomendável o estudo da metodologia de programação do μDX utilizada nestes exemplos. O exemplo do semáforo (SEMAFORO.UDX) foi programado sem método e é visivelmente mais complicado para entender ou modificar.

A velocidade de operação de 1/256 s só é operacional em versões de firmware iguais ou superiores à 4.2. A partir da versão 6.3 de firmware o μDX roda em velocidade quádrupla (cristal de 16,77721 MHz em vez de 4,194304 MHz). Estes controladores são chamados de μDX TURBO.

As teclas de Varredura e ModoDXNET só estão disponíveis no programa PG com versão maior ou igual a 2.4. Outras características, como transmissão via serial ou via linha telefônica só estão disponíveis em versões recentes do PG (3.5 ou superior).

As versões quando da confecção deste manual (junho de 2003) são as seguintes:

Firmware μDX	Versão 7.2
Firmware μDX+	Versão 8.2
Software PG	Versão 6.0
Placa Impressa μDX	Versão 2.1
Manual μDX	Revisão 2.3

Os periféricos e softwares disponíveis para o μDX e μDX+ até esta data são:

Expansão de Entradas/Saídas (8 entradas e 8 saídas)

Opto-acoplador para as 4 entradas do μDX

Interface Homem/Máquina (Teclado, Display e Entradas Analógicas)

Modem (Discagem por pulso, atendimento de ligação automática)

Conversor Analógico/Digital (com Sensor de Temperatura)

Regulador Chaveado (permite alimentar μDX com tensão de 15 a 60V)

Sensor de Tensão de Rede Trifásica

Supressores de Ruído Elétrico

Sensor de Temperatura Ambiente (-55°C a 125°C)

Sensor de Umidade Ambiente (10 a 90% UR)

Leitor de Código de Barras (para uso com μDX versão 6.6)

Biblioteca de Funções (para C, Pascal, Clipper e Basic)

Programa de Gerenciamento Remoto (PGR)

Registrador de Eventos (data-logger)

Placa de Saída Analógica para Conversor A/D (0-5V ou 0-10V)

Além disso, a Dexter comercializa soluções integradas, em que vários itens acima são conectados em uma única caixa metálica, formando uma solução para determinada aplicação de automação. Atualmente, o exemplo mais completo desta integração é o Controlador de Arrefecimento e Telemetria para ERBs (estações rádio-base para telefonia celular).

Visite nosso endereço na Internet - www.dexter.ind.br - para obter as últimas informações sobre os produtos Dexter.

Produtos DEXTER

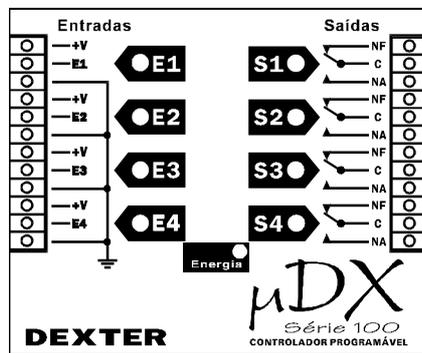
Controlador Lógico Programável (CLP) μDX Série 100

Trata-se de um controlador programável de 4 entradas e 4 saídas (contato seco de relé), compacto e de baixo custo, programável através de microcomputador compatível IBM-PC. Acompanha o software de programação (programa PG). Este software facilita muito a programação do μDX, pois é baseado em ícones (blocos de funções) ligados por fios (nodos), desenhados em tela gráfica VGA colorida, com utilização do "mouse" para posicionamento.

O controlador permite conexão a outros 14 controladores, via rede local DXNET. A rede local é multi-mestre com topologia tipo barramento, ou seja, todos os dispositivos ligados à rede podem iniciar comunicação e recebem simultaneamente as mensagens.

O programa é gravado no CLP em memória não-volátil (EEPROM). Além disso, possui previsão para pilhas comuns ou recarregáveis para manter execução do programa durante até 15 dias sem alimentação externa.

O controlador programável μDX Série 100 pode ser utilizado para controle de automatismos, racionalização do consumo de energia elétrica (ligando e desligando luminárias automaticamente, por exemplo), contagem de peças, e ainda telemetria e/ou telecomando, associado ao modem.

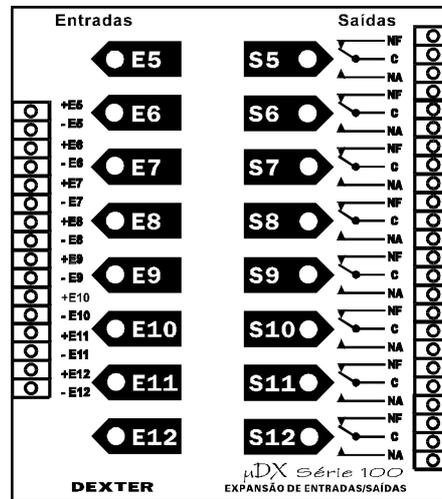


Controlador Lógico Programável μDX Série 100 Plus

Trata-se do controlador programável de 4 entradas e 4 saídas μDX, com maior capacidade de blocos, nodos e variáveis. O μDX+ permite o dobro de blocos (256 blocos), o triplo de nodos (192 nodos), e o quádruplo de variáveis (64 variáveis). Além disso, o controlador μDX+ admite até 4 Expansões de Entradas/Saídas, perfazendo um total de 36 entradas e 36 saídas. Também possui uma entrada de contagem rápida (E2), capaz de ler frequências até 3000 Hz. A versão atual do software PG suporta tanto o μDX normal quanto o μDX+ (Plus).

Expansão de Entradas/Saídas para μDX Série 100

Adiciona 8 entradas opto-isoladas e 8 saídas (contato seco de relé) ao controlador programável μDX Série 100. Assim, o conjunto fica com 12 entradas + 12 saídas (no caso do μDX+ pode-se utilizar até 4 Expansões, perfazendo 36 entradas + 36 saídas).



Interface Homem/Máquina para μDX Série 100

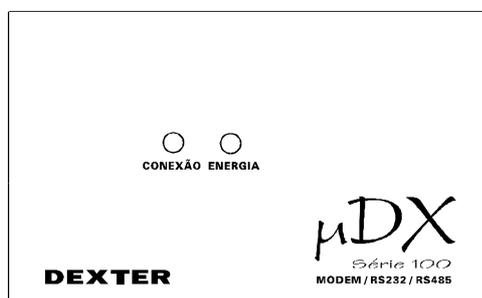
Esta unidade consiste de um display alfanumérico de 2 linhas, 16 caracteres por linha, e 4 teclas para seleção de parâmetro e valor.



Pode-se modificar e/ou visualizar temporizadores, variáveis e constantes do programa no controlador μDX. Além disso, a interface possui 8 entradas analógicas de 0 a 5 V (resolução de 8 bits), possibilitando medir variáveis como temperatura, por exemplo (mediante sensor com saída padrão 0 a 5V).

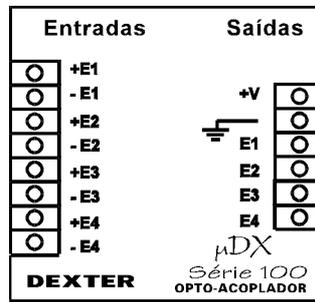
Modem para Controlador μDX Série 100

Trata-se de um modem (modulador/demodulador) capaz de estender a rede local DXNET via rede telefônica discada. O modem é compatível com norma Bell 103. O modem é capaz de discar e/ou atender chamadas telefônicas. A aplicação principal é telemetria e/ou telecomando. Assim, pode-se ter uma estação remota, por exemplo, que de tempos em tempos transmite seus dados à central via rede telefônica. O software PG (Programador Gráfico) que acompanha o controlador μDX permite conexão remota via modem do microcomputador com modems para μDxs.



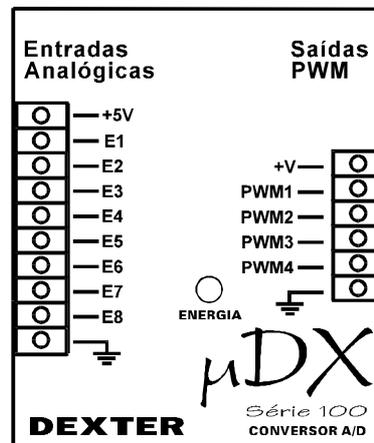
Opto-Acopladores para entradas do μDX Série 100

Permite isolar galvanicamente as 4 entradas do controlador programável μDX Série 100, possibilitando sua ligação direta à rede elétrica.



Conversor Analógico/Digital para μDX Série 100

Este módulo possui 8 entradas analógicas (programáveis via jumper para 0-5V, 0-10V ou ainda 0-20mA) e possibilidade de conexão de até 8 sensores de temperatura (-55 a 125°C), ou 4 sensores de umidade (10 a 90% UR). Acompanha o equipamento um sensor de temperatura e os cabos para sua conexão. Permite programar resolução (1°C; 0,5°C; 0,2°C; 0,1°C; 0,05°C; 0,02°C) e faixa de leitura da temperatura (entre -55 e 125°C). O conversor A/D pode se comunicar com controladores μDX via rede DXNET ou via 4 saídas moduladas por largura de pulso (PWM).



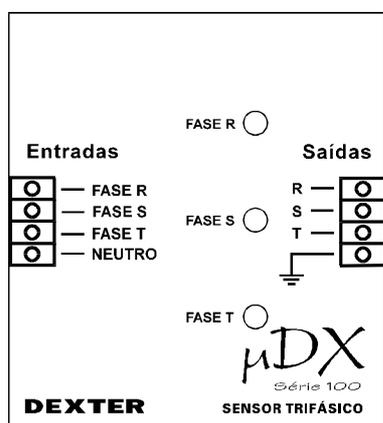
Regulador Chaveado para μDX Série 100

Permite alimentar o controlador μDX e acessórios com tensão contínua de 15V a 60V.



Sensor de Tensão de Rede Trifásica

O Sensor de Rede Trifásica utiliza três entradas do Conversor A/D para monitorar a tensão de uma rede elétrica trifásica, de 0 a 255 VCA, com resolução de 1 VCA.



Supressores de Ruído para saídas do μDX Série 100

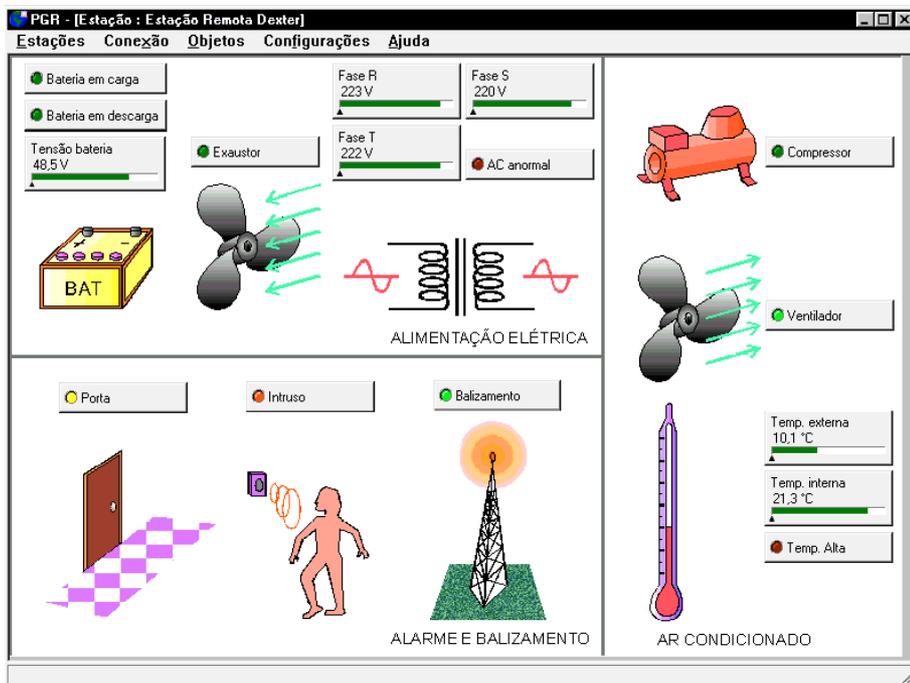
Os supressores de ruído elétrico são necessários no caso de cargas indutivas, como solenóides ou contactoras (vide capítulo sobre saídas do μDX).

Biblioteca de Funções para μDX Série 100

A DEXTER dispõe de um conjunto de subrotinas para serem utilizadas em linguagens como C, Clipper, Pascal ou Basic, que permitem a um programa desenvolvido pelo usuário total controle sobre o controlador programável μDX Série 100 e sobre a rede local DXNET. Essas rotinas são escritas para microcomputadores compatíveis IBM-PC com sistema operacional MS-DOS.

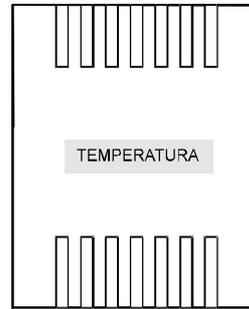
Programa de Gerenciamento Remoto (PGR)

O Programa de Gerenciamento Remoto permite sobrepôr a uma tela gráfica objetos com o valor de variáveis e/ou nodos de uma instalação com controladores μDX Série 100. Assim, é possível monitorar e atuar dispositivos via rede telefônica discada. O PGR utiliza plataforma Windows 95 ou Windows 98 e placa de fax/modem do microcomputador. Oferece ainda facilidades adicionais, como associar níveis de alarme e cadastro de estações remotas.



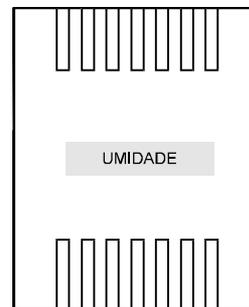
Sensor de Temperatura para Conversor A/D

Um sensor de temperatura acompanha o Conversor A/D para μDX Série 100. Unidades adicionais podem ser adquiridas separadamente, pois cada Conversor A/D pode ler até 8 sensores de temperatura. Estes sensores possuem resolução até 0,02°C e precisão de 0,5°C. A faixa de operação é de -55°C a 125°C.



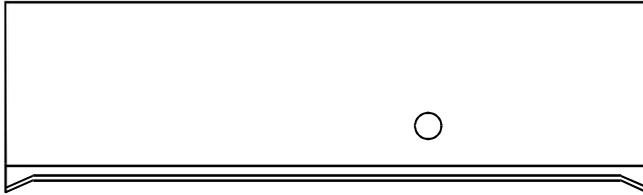
Sensor de Umidade para Conversor A/D

Cada Conversor A/D pode ler até 4 sensores de umidade. Estes sensores possuem resolução até 0,5% de UR (umidade relativa) e precisão de 5% de UR. A faixa de operação é de 10% a 90% UR.



Leitor de Código de Barras

O Controlador μDX Série 100, com firmware versão 6.6, permite leitura de código de barras via entrada E1. Este controlador é uma versão especial, em que o bloco de PWM, no caso da entrada E1, em vez de ler a modulação por largura de pulso na entrada, decodifica código de barras 2 de 5 intercalado com 10 dígitos, e coloca o valor lido em 4 variáveis sucessivas do μDX. O periférico representado abaixo é comercializado para permitir a leitura de cartões com tarjas codificadas em código de barras 2 de 5 intercalado. Como o sensor utilizado é infravermelho, pode-se usar tarjas pretas (para evitar duplicação heliográfica).



Placa de Saída Analógica para Conversor A/D

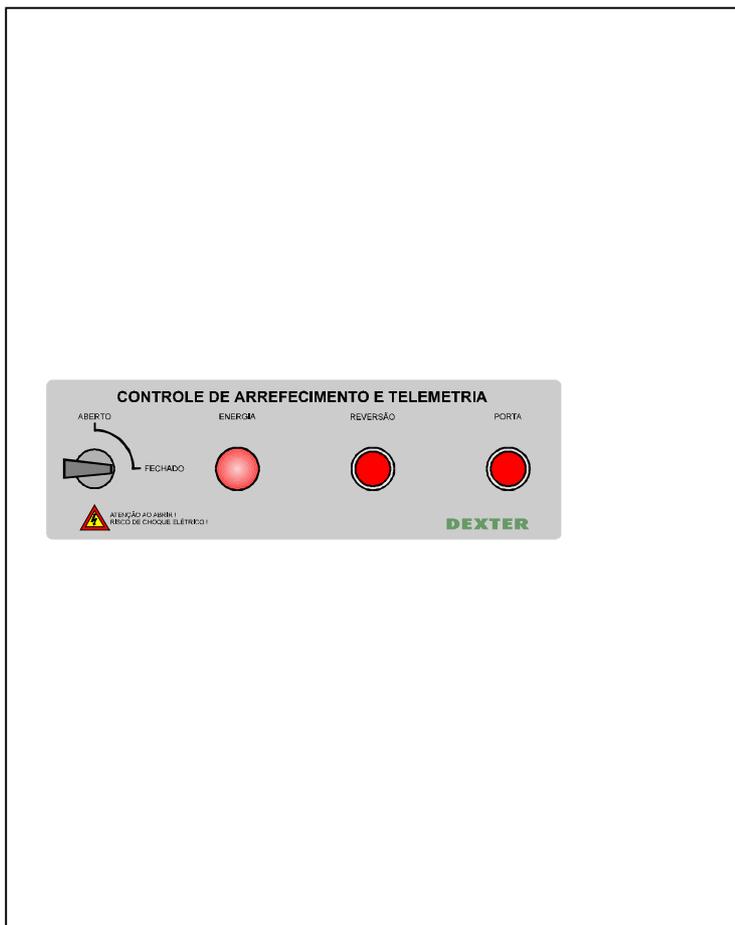
Cada Conversor A/D pode acionar até 4 Placas de Saída Analógica (saídas PWM1 a PWM4 do Conversor A/D). Esta placa converte os pulsos existentes na saída do Conversor A/D em tensão analógica entre 0-5V ou 0-10V.

Registrador para Controlador μDX Série 100

O Registrador permite armazenar dados, como estado de nodos e valor de variáveis, para posterior análise. Assim, é possível, por exemplo, armazenar temperatura e umidade ambientes (lidas via módulo de Conversor A/D) e gerar gráficos destas grandezas em função do tempo. O Registrador possui memória não-volátil de 64 Kbytes, e o ciclo de amostragem é programável entre 15 segundos e 32 minutos.

Controlador de Arrefecimento e Telemetria

O Controlador de Arrefecimento e Telemetria é, na verdade, uma unidade integrada em caixa metálica IP-55 (40x30x20cm), composta de 1, 2 ou 3 controladores μDX, Conversor A/D, Modem, Regulador Chaveado, Expansão de Entradas/Saídas e Sensores de Temperatura e Umidade. O controlador permite comandar 1, 2 ou 3 unidades de ar-condicionado da estação remota, assim como efetuar telemetria/telecomando na estação remota, via rede telefônica.



Controlador de Arrefecimento e Telemetria

Apresentação

O CONTROLADOR PROGRAMÁVEL μ DX (pronuncia-se micro-D-X) é um dispositivo eletrônico destinado a controlar processos de automação ou supervisão.

Possui quatro entradas que permitem detectar a ocorrência de determinados eventos como a abertura de portas ou janelas, a presença de luz ou ainda perceber a energização de equipamentos elétricos como um ar-condicionado, lâmpadas ou motores.

As entradas permitem alimentar sensores com 12 V, dispensando fontes adicionais. Assim, é possível conectar diretamente ao μ DX sensores de proximidade magnéticos ou capacitivos, por exemplo. Também chaves mecânicas de fim de curso, muito comuns em aplicações industriais, são ligadas diretamente ao μ DX.

Cada entrada está relacionada a um LED (indicador luminoso) que liga quando o sinal de entrada é considerado ativo. As entradas são denominadas de E1, E2, E3 e E4.

O μ DX possui também quatro saídas tipo relé que, atuando como interruptores elétricos, permitem ligar ou desligar aparelhos ou circuitos em corrente contínua ou alternada (DC/AC). Estas saídas podem ligar lâmpadas, motores, eletrodomésticos ou mesmo computadores.

Cada saída também está relacionada com um LED que liga indicando o acionamento do respectivo relé. As saídas são denominadas de S1, S2, S3 e S4.

Para que o μ DX possa exercer a função que queremos destinar-lhe ele possui, internamente, um microcontrolador para executar as tarefas e uma memória para armazenar o programa.

Este programa que instrui o μ DX sobre como tratar cada entrada e cada saída fica guardado numa memória que não perde seu conteúdo mesmo quando a bateria interna ou a energia externa não estiverem presentes. Sua construção permite guardar todo o programa por até 40 anos, sem precisar de qualquer fonte de energia.

O programa memorizado possui um ou mais comandos que ordenarão, um de cada vez, o que o μDX deve fazer. Depois que todos os comandos do programa forem executados pelo μDX ele voltará a fazer tudo novamente, recomeçando desde o primeiro. Os comandos podem ser de vários tipos: chaves comutadoras, temporizadores, operações aritméticas e lógicas, relógios, entre outros.

Para criar esta Programação utiliza-se uma ferramenta de "software" com recursos gráficos que permite montar programas intuitivamente, compondo-se, na tela de um computador, figuras interligadas que representarão os comandos que o μDX deverá executar. Esta ferramenta é chamada Programador Gráfico (PG).

O projeto do μDX levou em consideração ainda outras capacitações como: detecção de falta de energia, operação contínua em bateria (com os relés e LEDs desligados) por um período de até 15 dias sem provimento de energia externa e um conector de expansão para entrada e saída de valores com 8 bits, permitindo ter mais oito pontos de entrada e oito de saída ou um terminal de teclado e display para entrada e saída de dados.

Finalmente, a comunicação em rede local também está presente no μDX. Através da Rede Local DXNET (especificações desenvolvidas na DEXTER) é possível interconectar até 15 dispositivos que podem trocar informações em alta velocidade. Para melhor aproveitar o potencial da rede DXNET o μDX possui uma instrução multi-modo que o habilita a interferir no trabalho de outros μDX conectados na rede. Efetivamente isto significa um aumento do número de pontos de E/S e capacidade para elaboração de programas mais complexos que permitam alterar o processo caso algum dos μDX na rede sofra avarias ou detecte condições especiais.

Assim, graças ao emprego de uma tecnologia muito recente, o μDX pode reunir estas características:

- Muito baixo consumo: menor que 3 Watts.
- Operação em rede local: até 15 dispositivos independentes.
- Memória de programa não volátil, independente de bateria.
- Relógio de tempo real.
- Operação sustentada por bateria para quando falta energia.

- Programação intuitiva, através de ferramentas gráficas.
- 4 entradas AC/DC de -48 a 48 Volts (expansível à 12 entradas) (expansível a 36 entradas no caso de μ DX⁺).
- 4 saídas tipo relé com contatos para 10 Amperes (expansível a 12 saídas) (expansível a 36 saídas no caso do μ DX⁺).
- 3 entradas analógicas por leitura de largura de pulso (PWM).
- Conector de expansão para mais 16 pontos de E/S ou terminal com teclado e display.
- Ciclo de execução de programa de 62,5ms/31,25ms/15,6ms ou 3,9 ms.

Funcionamento do μ DX

O μ DX possui uma memória não volátil (que não perde o conteúdo mesmo quando falta energia) onde são armazenadas as instruções que lhe dizem o que deve ser feito a cada instante.

Para o μ DX cada instrução é chamada de bloco de instrução ou de bloco funcional. A memória do μ DX pode conter até 127 blocos de instrução. No caso do μ DX⁺, este número aumenta para 256 blocos.

O μ DX lê e interpreta cada um destes blocos, na seqüência em que foram armazenados, e chama-se de ciclo a leitura e interpretação completa de todos os blocos memorizados.

Como para qualquer controle ou automatização é necessário o maior grau de paralelismo possível (em qualquer processo sempre pode ocorrer mais de um evento diferente ao mesmo tempo) foi empregado no μ DX um método de execução de programa chamado Paralelismo Lógico.

Neste método os parâmetros de entrada (estado de ligações e valores de variáveis) são mantidos numa tabela acessível por qualquer um dos blocos de instrução que esteja sendo interpretado. Uma segunda tabela, com os resultados produzidos pela interpretação de cada bloco, vai sendo montada a medida que os blocos vão sendo lidos e interpretados. Assim, cada bloco poderá utilizar qualquer um dos parâmetros de entrada sem que estes sejam alterados devido à interpretação de algum outro bloco. Depois, no final do ciclo, a tabela de saída (com os resultados) é movida diretamente para a tabela de entrada para que os novos valores estejam disponíveis igualmente para todos os blocos no próximo ciclo.

É fácil perceber que esta forma de funcionamento faz com que todos os blocos sejam interpretados em paralelo, o que permite a elaboração de programas segmentados, onde cada parte pode controlar um processo independentemente e ao mesmo tempo que as demais.

Este paralelismo, operado em ciclos, faz com que a atualização da saída de um bloco de instrução para a entrada de um ou mais blocos demore o equivalente ao tempo de um ciclo. Esta demora, ou atraso, deve ser considerado no planejamento de um programa pois a conexão "encadeada" de, por exemplo, 10 blocos de instrução terá um atraso de 10 ciclos desde o estímulo na entrada do primeiro bloco até a saída no último. Com um tempo de ciclo de 1/16s isto resultaria em um atraso de 1,6 segundos.

A partir da versão de firmware 3.8 o μDX realiza um ou mais ciclos extras de execução especial em alta velocidade dentro do tempo de um ciclo normal. Esta execução especial serve apenas para as instruções de chaves interruptoras, acionando a saída destas mais rapidamente e, portanto, reduzindo o atraso total. Note-se que esta aceleração somente ocorre quando a saída da chave será ligada. Devido a lógica de operação do firmware esta aceleração não tem efeito ao desligar-se uma saída de chave.

Além disso, em velocidades de ciclo muito rápidas (como 1/256 s) e programas longos, pode ocorrer que o atraso seja maior que o valor de ciclo multiplicado pelo número de blocos encadeados. Isto porque a execução do programa pode levar mais tempo que um ciclo de execução.

Por limitações do "hardware" interno do microcontrolador empregado, o μ DX não coloca as variáveis (dados de 8 bits) nas tabelas de entrada e de saída. Assim quando um bloco altera o valor de uma variável este novo valor será empregado por algum outro bloco que utilize a mesma variável ainda no mesmo ciclo. Por exemplo, se um bloco altera o valor da variável V10 para 150 e outro bloco a ser interpretado a seguir (ainda no mesmo ciclo) testa se V10 é maior que 120 o resultado do teste será positivo ainda no mesmo ciclo, como se a Programação fosse seqüencial.

As tabelas de entrada e de saída, no μ DX, contém exclusivamente o estado de cada conexão entre os blocos de instrução: ligado ou desligado. Estas Conexões são também chamadas de NODOS.

A duração de cada ciclo determina a velocidade e a resolução de medida de tempo que o μ DX pode trabalhar. Existem quatro durações programáveis: 1/16s, 1/32s, 1/64s e 1/256s (62.5ms, 31.25ms, 15.625ms e 3,90625ms).

Estas durações são sincronizadas pelo relógio interno mas podem ser dilatadas caso o tempo necessário para a execução de todas as instruções do programa e para as comunicações via DXNET seja maior que a duração escolhida. O tempo total para leitura e interpretação do programa armazenado na memória pode variar conforme o número e tipo das instruções. Ainda, cada comunicação enviada ou recebida via DXNET consome de 10ms até 15ms.

Portanto, a escolha da duração de ciclo pode variar conforme a aplicação do μ DX: processos rápidos e sem comunicação via DXNET podem empregar a duração de 1/64s ou até 1/256s e os processos com muita comunicação podem utilizar a duração de 1/16s.

As entradas (4, digitais) e as saídas (4, tipo relé) do μ DX são lidas e atualizadas no início de cada ciclo. Além disso elas também são consideradas apenas NODOS, ou ligações, porque representam apenas um único estado (ligado ou desligado). Para cada entrada existe um LED (indicador luminoso) vermelho para indicar o estado atual dela. Estes LEDs são atuados pelo próprio microcontrolador do μ DX, de forma que seu estado representa exatamente o que o μ DX está detectando em cada uma das entradas.

As quatro saídas atuam diretamente os respectivos relés, cada um com capacidade de comutar correntes elétricas de até 10 Ampéres. Para cada saída existe um LED que indica se ela está atuada ou não.

Para se programar o μDX utiliza-se uma ferramenta fornecida juntamente com ele: o Programador Gráfico (PG).

Com o PG é possível programar o μDX utilizando-se uma linguagem avançada chamada de PDE : Programação por Diagrama Esquemático. Esta linguagem utiliza recursos gráficos intuitivos para montar os circuitos de controle que irão instruir o μDX.

Aplicações

A versatilidade e a capacidade do μDX permitem imaginar aplicações desde as mais simples até as mais complexas, que podem envolver a intercomunicação de vários μDX e utilização de programação distribuída.

O μDX pode ser empregado em minuterias inteligentes para edifícios, controle de irrigação de jardins, automatização para paisagismo como o controle de chafarizes, iluminação, ventilação e outros efeitos especiais. Pode operar um semáforo rodoviário completo, inclusive com recursos especiais como onda de verde e horários com temporização diferenciada.

Pode servir como simulador de presença e alarme residencial, controlar o funcionamento do ar-condicionado mantendo o ambiente com temperatura controlada. Pode medir o consumo de energia elétrica de eletrodomésticos, servir como temporizador para fotografia ou mesmo para lembrar o horário para se tomar medicamentos. Até o horário permitido para ligar a televisão ou abrir a geladeira pode ser monitorado e controlado pelo μDX!

Liga e desliga computadores em horários ou condições específicas. Permite posicionar antenas direcionais, animar festas (luzes e sons de efeito especial) e até mesmo fazer um pisca-pisca para árvore de natal.

Industrialmente ele pode ser empregado para controlar pequenas câmaras de teste por ciclo térmico ("burn-in"), inclusive submetendo o produto em teste a provas de energização intermitente ou verificações de estados progressivos. Pode controlar o uso de máquinas e reduzir o consumo de energia elétrica minimizando o tempo ocioso de máquinas ligadas ou iluminação integral em áreas sem presença de pessoas.

Para o "hobby" o μ DX pode controlar linhas férreas e até o andamento de modelos de trens, direcionar modelos de embarcações ou mesmo controlar um braço mecânico de robô.

Empregando a Rede Local DXNET é possível automatizar completamente uma residência ou até um parque, quando as entradas e saídas podem ser virtualmente expandidas para 360 ao todo (considerando-se que sejam utilizados 15 μ DX e todos eles com expansão de mais 8 entradas e 8 saídas - $15 \times (12+12) = 360$). No caso do μ DX+ (Plus) este valor pode ser estendido a 1080 entradas/saídas - $15 \times (36+36) = 1080$.

A Rede Local DXNET

A DEXTER desenvolveu um meio de intercomunicação entre diversos μ DX ou aparelhos quaisquer que precisem trocar informações para efetuar algum controle de processo: a Rede Local DXNET.

Funciona em topologia tipo barramento, isto é, todos os dispositivos ligados à rede recebem qualquer comunicado simultaneamente.

O modo de operação é Multi-Mestre, ou seja, qualquer um dos dispositivos pode iniciar um comunicado com qualquer outro independentemente de sinalizações (nas redes Mestre-Escravo é sempre um único dispositivo que sinaliza quando algum outro poderá realizar uma comunicação). Este modo é o mais recomendado para os controles de processos já que oferece equilíbrio de prioridade a todos os dispositivos e permite implementar sistemas mais imunes a falhas.

Eletricamente a rede DXNET é formada por um cabo blindado com uma malha conectada ao comum (0v) e um fio central que conduz as informações. Opera com níveis de 0 e 5 Volts e com velocidade de aproximadamente 9.000 bits por segundo. Esta velocidade é suficiente para permitir conexões a boas distâncias e garantir a troca de dados a uma taxa adequada à própria duração de ciclo dos μDX nela conectados.

Esta Rede Local foi desenvolvida especificamente para permitir a intercomunicação entre dispositivos desprovidos de "hardware" especial ou sem capacidade para tratamento convencional de interrupções. Assim qualquer microprocessador ou microcomputador pode ser facilmente programado para efetuar comunicações via DXNET.

Instalação Inicial

Leia com atenção este capítulo. Mesmo que as instruções lhe sejam familiares é importante conhecer todos os detalhes de Instalação.

Na embalagem do μDX são fornecidos os seguintes itens:

- μDX Série 100
- Adaptador com cabo para DXNET
- Fonte de Alimentação
- Manual de Utilização (este manual)
- Disquete com o PG e programas de exemplos

O adaptador serve para conectar um microcomputador (tipo IBM-PC) ao μDX. Ele deve ser ligado na porta paralela (onde conecta-se a impressora) e a extremidade livre do cabo deve ser ligada em qualquer um dos conectores para Rede Local DXNET, no μDX.

A Fonte de Alimentação é que supre de energia o μDX para que ele possa acionar os relés e os LEDs, além de executar o programa.

ATENÇÃO: Antes de ligar a fonte de alimentação na rede elétrica verifique se ela está ajustada adequadamente para a tensão da rede no local (127 ou 220 Vca).

A extremidade livre do cabinho que sai da Fonte de Alimentação deve ser ligada ao conector no μDX que está indicado como ENERGIA.

Instalando os arquivos do disquete em DOS

O disquete que acompanha o conjunto do μDX contém o arquivo executável do Programador Gráfico (PG) e seus arquivos de dados (ícones, tabela de cores e logotipo). Além destes estão presentes alguns arquivos contendo programas de exemplo para o μDX.

Para instalar o Programador Gráfico no disco rígido ("winchester") siga as instruções abaixo (as instruções abaixo pressupõem que o microcomputador esteja em uma sessão DOS):

- Ligue o computador e aguarde a inicialização feita pelo DOS.

- Gere um diretório para o μDX digitando, por exemplo:

```
MD UDX <enter>
```

- A seguir passe para este novo diretório digitando:

```
CD UDX <enter>
```

- Insira o disquete, que acompanha o produto, no drive A.

- Copie o programa digitando:

```
COPY A:PG60_DOS.ZIP <enter>
```

- Descompacte o programa PG60_DOS no diretório criado (utilize, por exemplo, o software PKZIP para DOS) e execute o software PG:

```
PG <enter>
```

Atenção: Para executar o software PG são necessários 500 KBytes de memória convencional livre. Para obter este dado digite o comando MEM em DOS. É possível alocar a memória alta para o sistema operacional DOS, de forma a obter mais memória convencional livre (veja o comando HIMEM no manual do DOS). Além disso, é necessário instalar um software para interface de mouse (normalmente acompanha o próprio mouse).

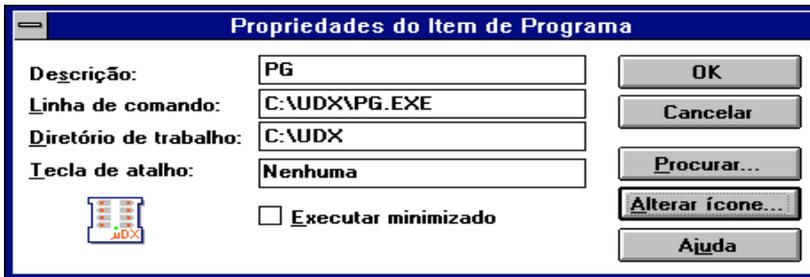
Instalação no Windows 3.x

Para instalação no sistema operacional Windows 3.x é fornecido com o programa PG um arquivo PG.PIF (arquivo de inicialização do programa PG no Windows) e um arquivo PG.ICO (ícone para chamada do programa PG no Windows).

Siga os passos descritos na página anterior para instalar o programa PG no sistema operacional DOS. A seguir, entre no Windows. Selecione <Arquivo> e <Novo>. O sistema operacional irá perguntar se deve criar um Grupo de Programas ou Item de Programa. Escolha a segunda opção caso queira inserir a chamada do PG em algum grupo de programas já existente. A primeira opção permite criar um novo grupo de programas. Digamos que se insira a chamada do programa PG em algum grupo de programas já existente (por exemplo, em Aplicativos). Neste caso, selecione a segunda opção. A seguir, o Windows irá perguntar:

- Descrição: É o nome que aparece embaixo do ícone.
Sugestão: PG
- Linha de Comando: É o comando a ser executado quando o ícone é chamado. No caso de o PG estar no disco C:\UDX devemos digitar: C:\UDX\PG.EXE
- Diretório de Trabalho: É o diretório que será utilizado pelo programa PG. Digite: C:\UDX
- Tecla de Atalho: Deixe este campo em branco.

Agora basta selecionar o desenho (ícone) para o PG. Pressione a tecla <Alterar Ícone>. O sistema operacional irá advertir que não tem nenhum ícone para este programa. Pressione <OK> e, na janela que surgir com vários ícones para seleção, pressione <Procurar>. A seguir selecione o ícone PG.ICO no diretório C:\UDX e pressione <OK>. O Windows deve apresentar a seguinte janela, se deu tudo certo:



Basta pressionar a tecla <OK> para finalizar a operação. Deve surgir um ícone com o desenho do μDX no grupo de programas selecionado que, uma vez ativado, chama o programa PG. Ao sair do PG, retorna-se automaticamente ao sistema operacional Windows.

Estes são, de maneira sumária, os passos necessários para chamar o programa PG via Windows. Maiores detalhes podem ser obtidos no Guia do Usuário da Microsoft para o sistema operacional Windows.

Instalação Manual do Software PG (Programador Gráfico) em Windows 95 ou 98

O software PG foi elaborado para rodar em ambiente DOS. Entretanto, é possível especificar um atalho no Windows 95 ou 98, de forma a chamar o programa PG sem a necessidade de reinicializar o computador no modo DOS. Na verdade, o atalho abre uma janela de modo DOS, com tela cheia, para uso do programa PG. O procedimento descrito a seguir especifica como gerar um atalho em Windows 95 ou 98 manualmente. Note que as versões atuais do software PG (versão 6.0) já possuem Assistente de Instalação incorporado e, com isso, geram automaticamente os atalhos necessários. Veja o capítulo seguinte (Instalação Automática do Software PG em Windows 95, 98 e Millenium) para maiores informações.

1. Copiar o disquete

Crie um diretório chamado PG e copie todo o conteúdo do disquete recebido com o μDX. É possível utilizar as ferramentas disponíveis em Windows (como o Explorer) ou o sistema operacional DOS. Caso prefira efetuar esta tarefa via DOS, abra uma janela DOS e digite:

```
CD \ <enter>
```

Esta instrução irá retornar a raiz de seu disco rígido.

```
MD PG <enter>
```

Esta instrução irá criar um diretório chamado PG.

```
CD PG <enter>
```

A seguir passe para este novo diretório com a instrução acima.

Por fim, coloque o disco que acompanha o controlador programável μDX no drive A e digite:

```
COPY A:PG60_DOS.ZIP <enter>
```

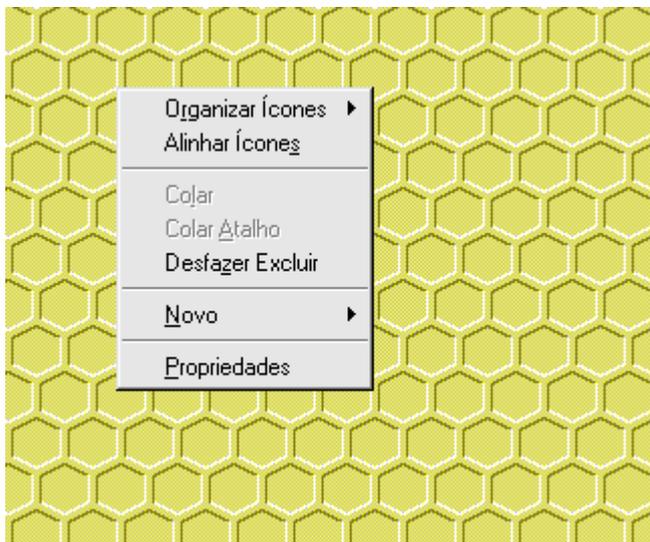
Descompacte o arquivo ZIP (através de programa PKZIP, por exemplo) e execute PG.EXE para iniciar o programa PG (caso queira executar o programa a partir da própria janela DOS):

PG <enter>

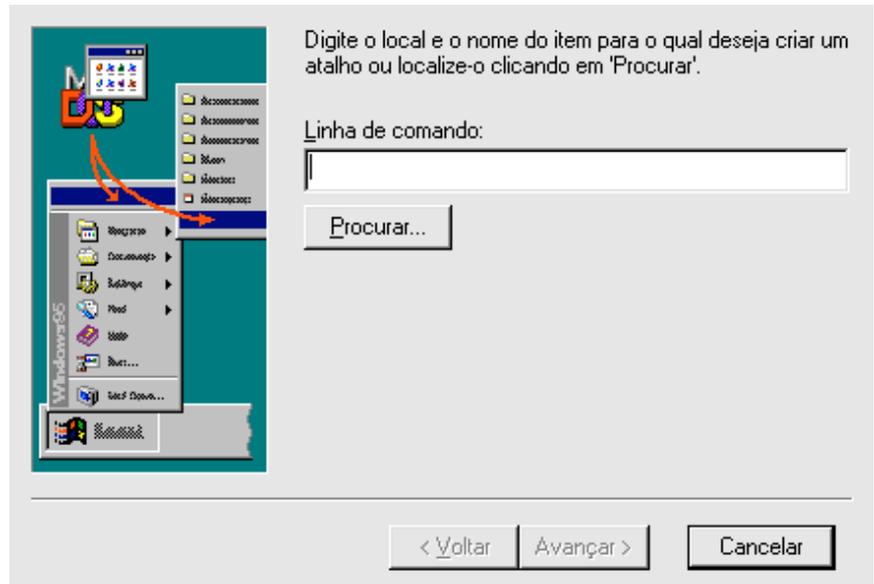
Note que <enter> significa pressionar a tecla <enter> (e não digitar esta palavra). Agora o Programador Gráfico (PG) e os demais arquivos já estarão instalados em seu disco rígido. A seguir feche a janela DOS, retornando ao Windows 95 ou 98.

2. Gerar um atalho

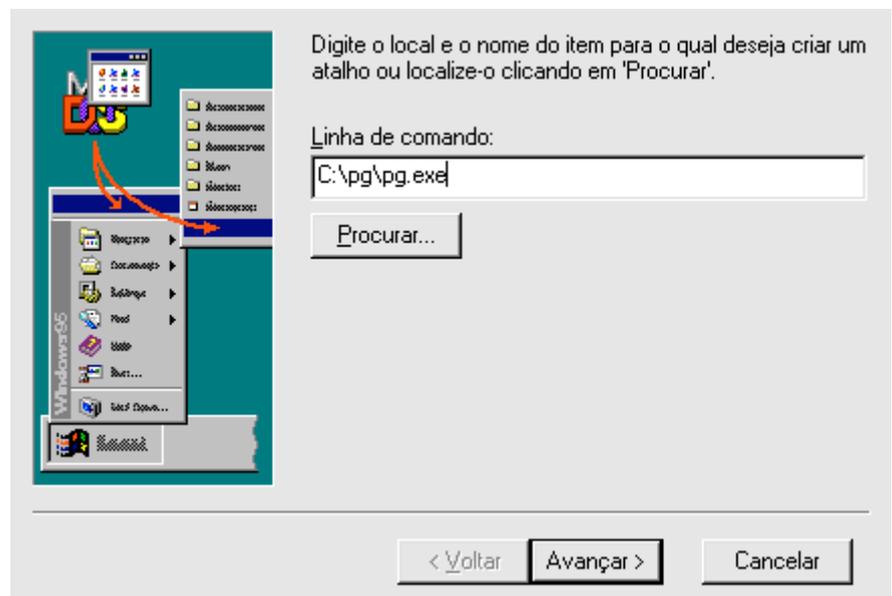
Clique na tecla direita do mouse apontando para a tela de fundo do Windows95 ou 98. Aparecerá a janela abaixo com as seguintes opções:



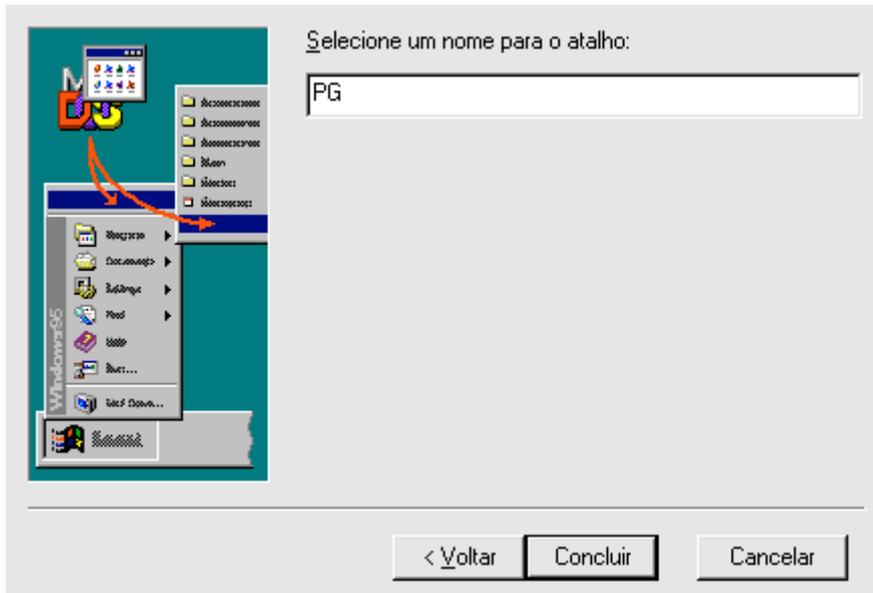
Selecione Novo e Atalho. Irá surgir a tela abaixo:



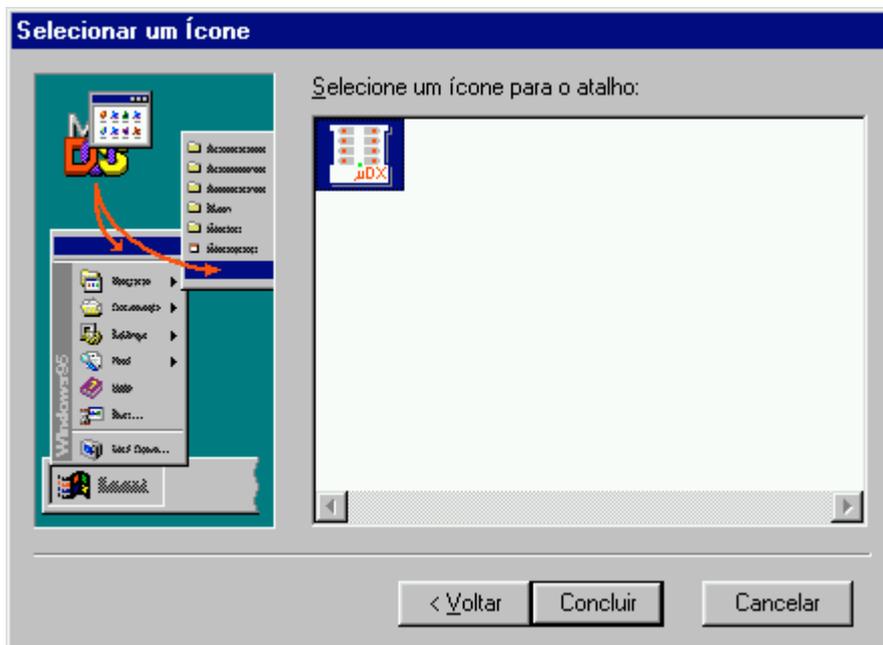
Então digite: C:\PG\PG.EXE como mostrado abaixo e clique em Avançar:



A seguir seleccione o nome do atalho: PG.



Por fim, seleccione o ícone do µDX para o atalho:



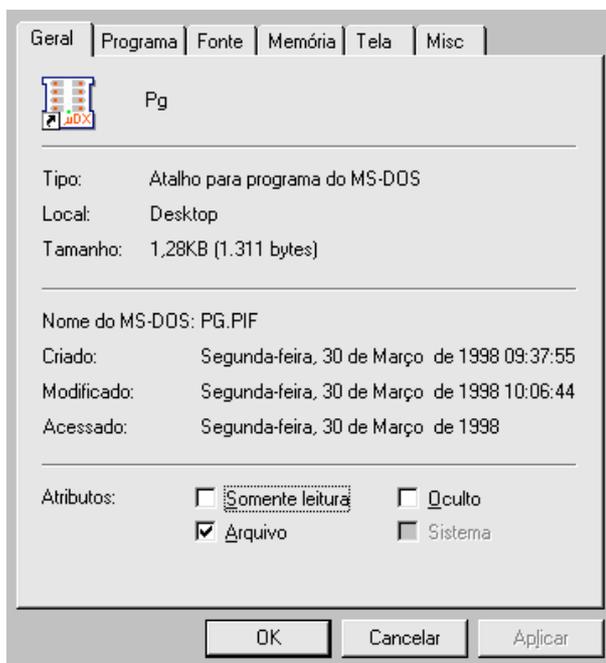
Depois de gerar o atalho é necessário editar suas propriedades.

3. Editar propriedades do atalho

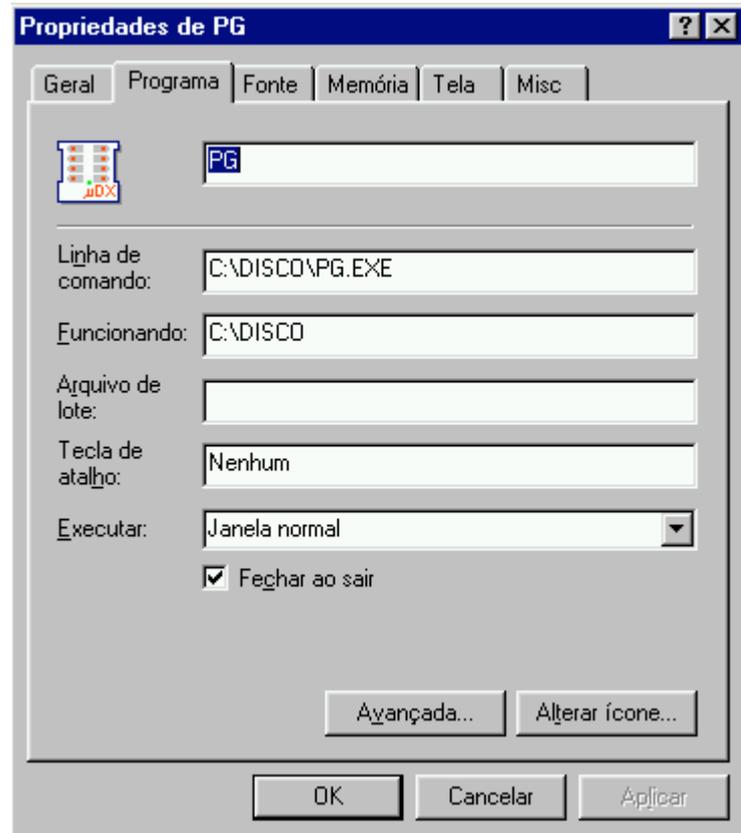
Irá surgir um atalho com o ícone do μDX e de nome PG na tela do Windows95 ou 98. Clique uma vez com a tecla direita do mouse sobre este atalho. Deve aparecer o seguinte menu:



Selecione Propriedades. Será apresentada a tela a seguir:



Selecione Programa:



Verifique se as informações mostradas na janela estão conforme a figura acima. A seguir selecione Fonte, Memória, Tela e Misc, sucessivamente, e modifique os dados destas janelas conforme mostrado nas páginas seguintes deste manual.

Selecione Fonte:



Selecione Memória:

The image shows a Windows-style dialog box titled "Propriedades de PG" with a blue title bar and standard window controls. The "Memória" tab is selected, showing settings for different memory types. The "Memória convencional" section has "Valor total" set to "Auto" and "Ambiente inicial" set to "Auto", with an unchecked "Protegido" checkbox. The "Memória expandida (EMS)" section has "Valor total" set to "1024". The "Memória estendida (XMS)" section has "Valor total" set to "1024" and the "Utiliza HMA" checkbox checked. The "Memória em modo protegido do MS-DOS (DPMI)" section has "Valor total" set to "Auto". At the bottom, there are "OK", "Cancelar", and "Aplicar" buttons.

Memória convencional	Valor total	Ambiente inicial	Protegido
	Auto	Auto	<input type="checkbox"/>

Memória expandida (EMS)	Valor total
	1024

Memória estendida (XMS)	Valor total	Utiliza HMA
	1024	<input checked="" type="checkbox"/>

Memória em modo protegido do MS-DOS (DPMI)	Valor total
	Auto

Selecione Tela:



Selecione Misc:



A seguir clique em OK para fechar a janela de edição de propriedades do atalho para PG. Basta clicar neste atalho para rodar o Programador Gráfico (PG) para controlador programável μDX a partir do sistema operacional Windows 95 ou 98.

ATENÇÃO: O software PG - Versão 4.1 ou superior permite ser executado a partir do sistema operacional Windows 95 ou 98, em janela DOS. Entretanto, para efetuar comunicações DXNET via porta paralela é necessário calibrar o software na primeira execução (depois não é mais necessário tal procedimento). Para isso, ligue um controlador μDX a porta paralela do computador, chame o software PG através do Windows e selecione a janela [Tela]. A seguir pressione a tecla [Calibra] . Uma vez feita a calibração ela é salva no disco rígido do computador, e não é necessário repetí-la em chamadas posteriores ao software PG.

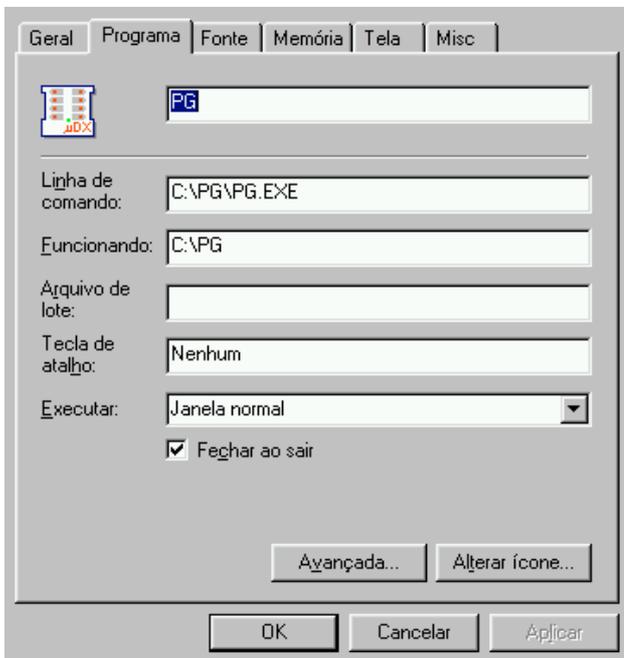
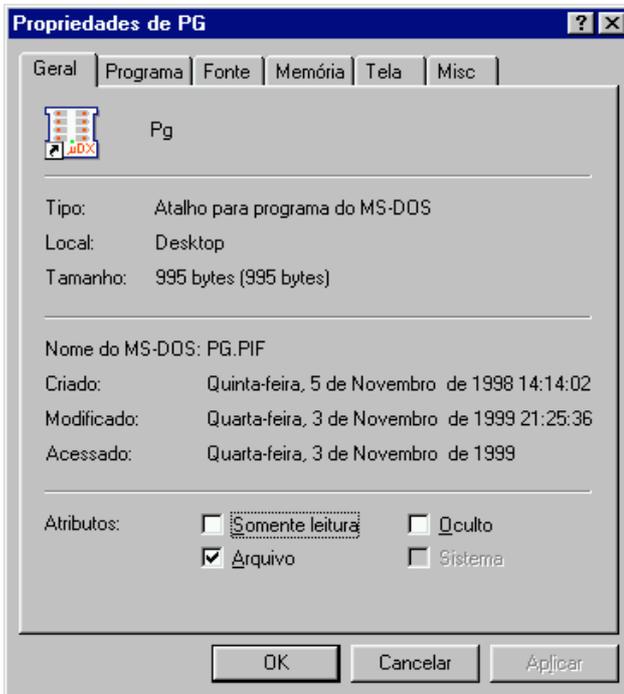
Instalação Automática do software PG (Programador Gráfico) em Windows 95, 98 e Millenium

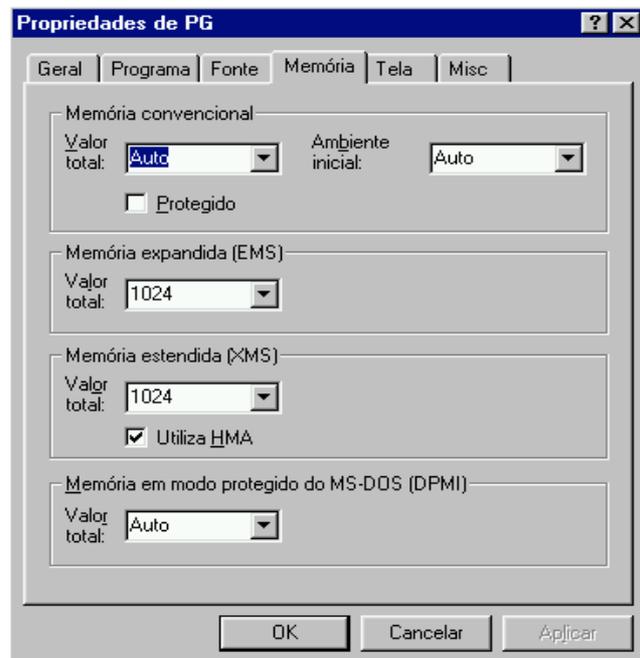
A versão atual do software PG – Programador Gráfico – possui instalador automático, capaz de gerar os atalhos em Windows. Para instalar o PG basta executar o programa SETUP.EXE, que irá acionar o Assistente de Instalação do PG. Uma vez instalado, o próprio Assistente de Instalação gera um atalho no Windows 95, 98 ou Millenium. Entretanto, pode ser necessário editar as propriedades do atalho, conforme descrito adiante. Além disso, para efetuar comunicação com Controlador μDX via DXNET deve-se calibrar o PG uma única vez, de forma que este avalie a velocidade de seu microcomputador e sistema operacional.

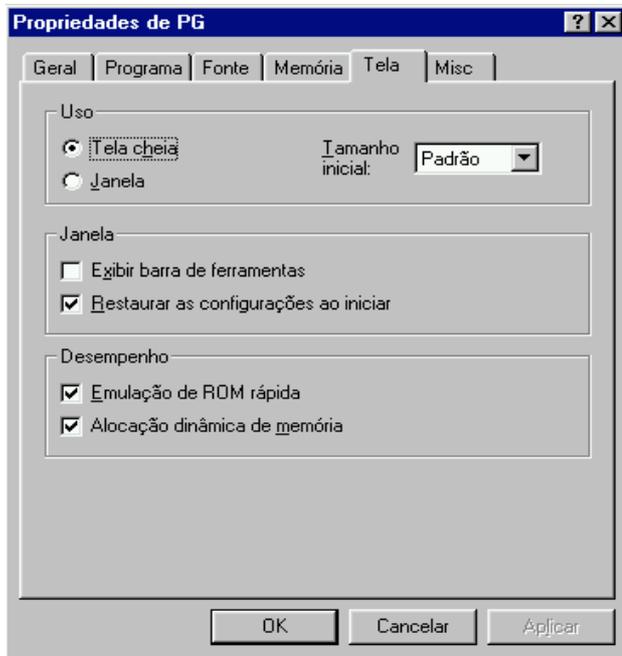
Os passos descritos neste capítulo estão explicitados também no documento INSTPG59.DOC, que acompanha o software PG.

Caso surjam dificuldades na utilização do atalho para o PG edite suas propriedades conforme especificado nas figuras que seguem. Note que para entrar em edição das propriedades basta selecionar o atalho e clicar a tecla direita do mouse e, a seguir, selecionar o item Propriedades.









ATENÇÃO: O SOFTWARE PG - Versão 6.0 permite ser executado a partir do sistema operacional Windows 95, 98 ou Millenium. Entretanto, para efetuar comunicações DXNET via porta paralela é necessário calibrar o software na primeira execução (depois não é mais necessário tal procedimento). Para isso, ligue um controlador μDX a porta paralela do computador, chame o software PG através do Windows e selecione a janela [Tela] (canto superior direito da tela). A seguir pressione a tecla [Calibra] . Uma vez feita a calibração ela é salva no disco rígido do computador, e não é necessário repeti-la.

INSTALAÇÃO EM WINDOWS NT, 2000, XP: O software PG - Versão 6.0 não permite utilização direta de sistema operacional Windows NT, 2000 ou XP. Estes sistemas operacionais bloqueiam o acesso ao hardware do computador (como porta paralela e porta serial). Para desbloquear este acesso é necessário utilizar software UserPort, de Tomas Franzon. Este software está disponível no site da Dexter – www.dexter.ind.br. Basta descompactá-lo e copiar o arquivo UserPort.sys (que acompanha o software) para o diretório c:\Windows\System32\Drivers. Depois disso execute o programa UserPort.exe e clique na tecla [Start]. Pronto, o I/O do computador estará liberado. Caso queira desinstalar esta liberação, basta entrar novamente no UserPort e clicar em [Stop]. Uma vez liberado o acesso as portas de comunicação do PC, rode o PG normalmente (a instalação é idêntica a efetuada em Windows 95, 98, ou Millenium). No caso de Windows XP, convém editar as propriedades do atalho do PG e selecionar as opções de Memória para “Auto” em vez de “Nenhuma”, Compatibilidade como “Rodar como Windows 98/Windows Me”, marcar as opções “Rodar em 256 cores”, “Rodar em resolução 640x480”, “Desativar temas visuais”. Por fim, em Diversos, desabilitar “Ativar proteção de tela” e habilitar “Suspende sempre” o plano de fundo. Muitas vezes o PG não acessa corretamente a porta paralela para acesso a rede DXNET. Então, ao entrar no PG clique na tecla [μDX] existente no canto superior direito e selecione [Porta DXNET]. A seguir clique no primeiro botão à esquerda [378h]. Também as mesmas providências descritas neste quadro para calibração do PG em Windows 95, 98 ou Millenium devem ser seguidas.

Instalação e Troca das Pilhas

Internamente ao μDX existe um suporte para 4 pilhas do tipo AA (pilhas pequenas). Estas pilhas irão suprir de energia o microcontrolador do μDX quando ocorrer interrupção no suprimento através da Fonte de Alimentação (falta de energia na rede elétrica ou desligamento da Fonte de Alimentação).

Com este suprimento de energia auxiliar o μDX mantém o funcionamento do relógio interno e também continua a execução completa do programa que estiver em sua memória. Apenas os relés, os LEDs e algum eventual circuito de expansão é que não recebem força desta energia auxiliar fornecida pelas pilhas.

Desta forma o consumo fica em torno de 2mA, o que permite o processamento contínuo (inclusive com comunicação via DXNET) por até 15 dias (empregando pilhas alcalinas novas).

Quando a energia suprida pela Fonte de Alimentação está presente as pilhas são eletronicamente desligadas, mantendo suas cargas preservadas.

ATENÇÃO: *O μDX não possui circuito para reconhecimento de quando as pilhas estiverem sem carga. Exemplificando, se o μDX precisar funcionar apenas com as pilhas por cerca de 1 hora por dia (mantendo a Fonte de Alimentação energizada durante as outras 23 horas) as pilhas somente precisarão ser substituídas uma única vez por ano.*

Para substituir as pilhas (ou instalá-las a primeira vez) é preciso abrir a caixa do μDX.

Desligue os conectores da Fonte de Alimentação, da Rede Local DXNET e da expansão (se tiver). Abra a caixa do μDX forçando levemente as laterais para afastarem-se dos encaixes que prendem a tampa ao fundo.

Puxe a tampa cuidadosamente para cima. Cuidado com os LEDs que estão montados presos a pequenos conectores de dois contatos. Caso algum dos LEDs saia do lugar observe a posição dos LEDs adjacentes para saber como encaixar o que saiu (ao contrário de uma lâmpada, o LED tem polaridade e não liga se for invertido).

Caso já existam pilhas (que neste momento estão mantendo o μDX em operação) volte a ligar o conector da Fonte de Alimentação. Assim o μDX não vai parar (e o relógio será mantido em funcionamento) quando as pilhas forem retiradas.

Retire o suporte das pilhas do encaixe, tomando o cuidado para não puxar em demasia e forçar os fios de ligação.

Coloque 4 pilhas novas (com carga completa) do mesmo tipo e observando a posição de cada uma devido a polaridade (a mola do suporte de pilhas é sempre ligada ao terminal negativo das pilhas).

ATENÇÃO: *Nunca utilize pilhas de tipos diferentes ao mesmo tempo, como uma mistura entre pilhas convencionais e alcalinas. Cada tipo possui um modo de funcionamento próprio que não permite combinação adequada com outros modelos.*

Depois de instaladas as pilhas no suporte volte a encaixá-lo na fixação deste na parte de baixo da caixa. Desligue o conector da Fonte de Alimentação (se estiver ligado).

Verifique se os LEDs estão bem alinhados entre si antes de aproximar a tampa para o fechamento. Encaixe a tampa alinhando os furos superiores com os LEDs. As quatro pequenas depressões da tampa irão fixar-se nos furos da parte de baixo da caixa feitos para este fim.

Volte a ligar os conectores da Fonte de Alimentação, DXNET e expansão (se estiver utilizando).

q Note que, uma vez instaladas as pilhas, sempre que não houver energia elétrica sendo suprida pelo conector de ENERGIA as pilhas estarão sendo gastas. Assim, não basta parar o programa instalado no μDX para evitar o consumo das pilhas.

Nunca armazene o μDX com pilhas instaladas. Caso ele não vá ser usado por algum tempo retire as pilhas para evitar que descarreguem.

Pilhas Recarregáveis

É possível instalar pilhas recarregáveis no μDX, evitando o incômodo da troca periódica destas em aplicações que necessitem do relógio de tempo real. Podem ser instaladas 4 pilhas recarregáveis tipo AA, de 500mAh ou 750 mAh. Para que as pilhas sejam recarregadas quando o μDX estiver com alimentação elétrica, basta fechar a conexão ("jumper") JP14, existente na placa de circuito impresso (esta conexão vem aberta de fábrica, prevendo pilhas comuns)(Esta conexão só está presente em placas versão 1.4 ou maior).

Nunca feche esta conexão ("jumper") com pilhas comuns (não recarregáveis) instaladas no μDX. Pode ocasionar vazamentos de ácido das pilhas.

ATENÇÃO: *No caso de pilhas comuns, a tensão de alimentação do μDX (Vcc) atinge 6V (4x1,5V). Isto não provoca nenhum problema no circuito do μDX. Entretanto, tal fato deve ser considerado ao projetar-se circuitos a serem ligados ao conector de expansão. Já no caso de pilhas recarregáveis, a tensão fica em 4,8V (4x1,2V); mais próximo dos usuais 5V (quando alimentado pela rede elétrica a tensão no μDX - Vcc - é de 5V).*

Conexões nas Entradas e Saídas

As Entradas

As quatro entradas digitais do μ DX estão localizadas, todas, numa única lateral da caixa (lado esquerdo quando a caixa é vista por cima).

Elas estão codificadas como E1, E2, E3 e E4. Cada uma utilizando três contatos: um para o +V, um para a entrada (E1 até E4) e um para o 0V (conhecido também como "GND", massa, retorno ou comum); conforme representado pela serigrafia da tampa.

O +V é uma conexão internamente ligada ao conector da Fonte de alimentação (Energia) e permite que algum circuito externo de baixo consumo seja energizado (emprega-se também o contato de 0V). Tal circuito pode servir como um conversor ou adaptador ativo para que seja possível, por exemplo, detectar com uma entrada digital os sinais vindos de sensores de luz, pressão, umidade ou mesmo temperatura.

Além disso o contato de +V é empregado para ativar a entrada quando um interruptor é utilizado como sensor. Neste caso o interruptor fica ligado entre o +V e o contato da entrada. Quando o interruptor fechar (permitir a passagem de corrente elétrica) o circuito da entrada detectará o sinal tornando-a ativa.

A entrada é considerada ativa quando a tensão presente nela estiver entre +2 Volts e +48 Volts (valor positivo medido em relação ao 0V), e é considerada inativa quando a tensão ficar entre -48 Volts e cerca de +0.9 Volt. Esta região de incerteza, entre +0.9 Volt e +2 Volt, é comum para os circuitos digitais convencionais e suficiente para todas as aplicações práticas do μ DX.

CUIDADO: Os μ DX com número de série a partir de 03800011 não possuem resistor limitador entre o contato +V e o conector de Energia. Assim algum curto-circuito acidental entre o contato +V e o 0V poderá causar danos ao circuito impresso ou à Fonte de Alimentação.

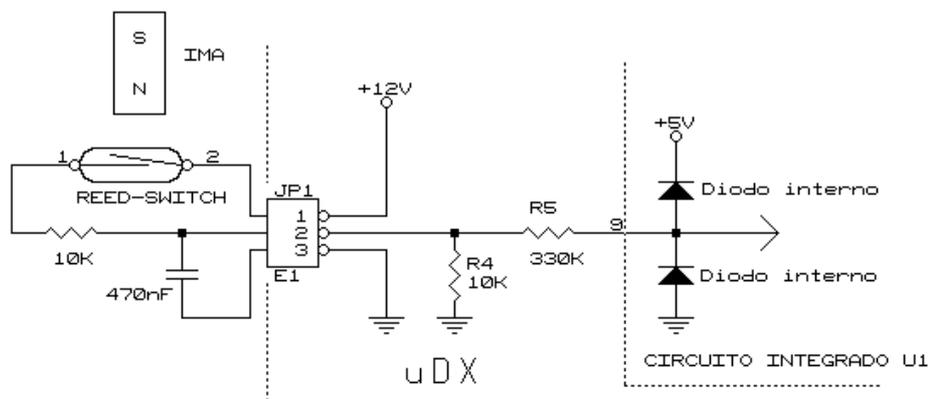
ATENÇÃO: As entradas do μDX NÃO SÃO ISOLADAS GALVANICAMENTE, isto é, não é possível ligar a qualquer uma delas algum fio conectado à rede elétrica domiciliar. Se isto for feito poderá causar danos graves no computador que estiver ligado na DXNET e no próprio μDX. Se for preciso detectar a presença de energia elétrica utilize um circuito isolador (tipo OPTO-ACOPLADOR) como sugerido no anexo DIAGRAMAS ESQUEMÁTICOS. A DEXTER comercializa um circuito com 4 opto-acopladores para o μDX.

Se alguma aplicação em que for empregado o μDX gerar muito ruído elétrico que possa ativar alguma das entradas por interferência, deve-se utilizar um pequeno circuito de filtragem nas entradas sensíveis. Este filtro é formado por um resistor e um capacitor, montados como mostrado nas figuras seguintes.

Alguns exemplos de circuitos de sensoriamento são mostrados a seguir.

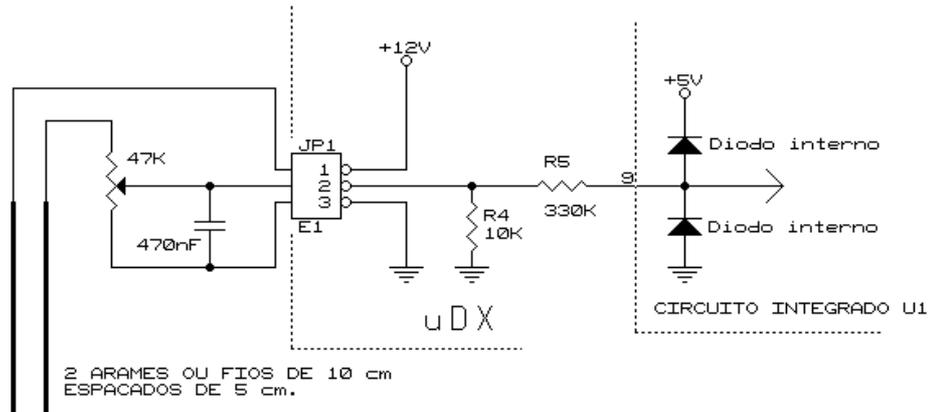
Sensor tipo Reed-Switch

Este tipo de sensor é adequado nas instalações de alarmes, detectando a abertura de portas ou janelas. O esquema mostrado na figura apresenta, também, parte do circuito interno de uma das entradas do μDX (ver Anexo C - Diagramas Esquemáticos). O resistor interno de 10KΩ serve para descarregar o capacitor utilizado como filtro quando o Reed-Switch abre.



Sensor de Umidade

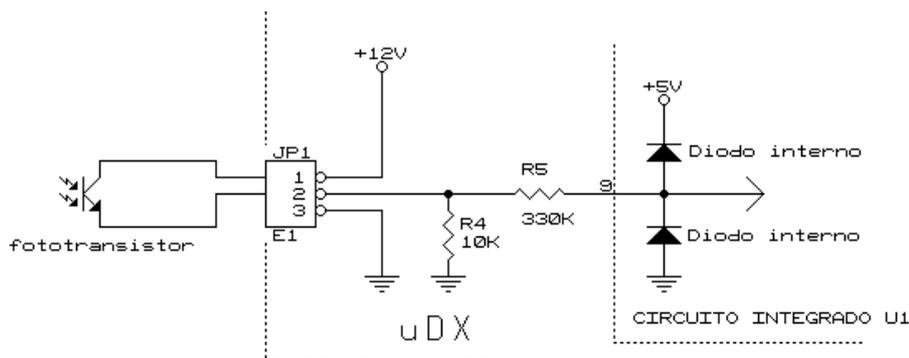
Para que alguma das entradas do μDX detecte a presença de umidade no solo de um gramado ou de vasos (também o transbordamento de piscinas ou um sensor de nível para algum processo industrial) é preciso apenas um circuito externo muito simples. Como a impedância de entrada é cerca de 10KΩ (resistência entre o sinal de entrada e o comum), e considerando que o μDX detecta entrada ativa a partir de 2 Volts, para isto bastará uma resistência menor que 50KΩ (entre o +V e o sinal de entrada).



Nas experiências práticas constatou-se que a resistência medida na presença de umidade é bem menor do que isso. Portanto, com um resistor variável (trimpot) montado como o esquema abaixo pode-se ajustar o limiar de sensibilidade do sensor adequado para cada caso.

Sensor de Luz

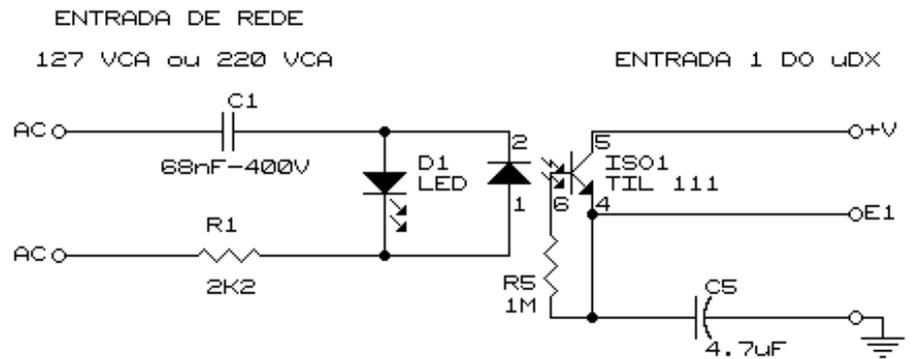
Com este esquema muito simples pode-se detectar a incidência de luz em um foto-transistor (pode ser o TIL81 ou outro equivalente). Basta conectar o foto-transistor como mostrado. Se for empregada uma barreira ótica (componente com um emissor de luz e um foto-transistor) o emissor de luz poderá ser alimentado a partir do +V presente no conector de entrada do μDX. Utilize um resistor de 1K8 em série com o emissor de luz para limitar a corrente ao nível necessário.



Detector de tensão AC Opto-acoplado

Com este circuito pode ser detectada a presença de tensão AC em 110V ou 220V, com a segurança do sinal ficar isolado do circuito do μDX. Esta isolação é necessária para evitar curto-circuito entre diversas entradas que monitorem sinais AC de alta tensão e também protegem o μDX de eventuais descargas elétricas que possam ocorrer. O LED D1 ilumina-se quando a tensão AC está presente, também protegendo o emissor de luz do opto-acoplador contra a tensão reversa do semi-ciclo não aproveitado por ele.

O capacitor de 4,7μF(C5) age como filtro, com um período maior que o ciclo de rede 60Hz (16,67 ms). Para ligarmos a esta entrada uma tensão de 12Vdc em vez dos 127 ou 220 VAC basta substituir o capacitor C1 por um curto-circuito (e colocar o LED em série com o LED do opto-acoplador, de forma que ambos liguem quando for aplicada tensão na entrada).

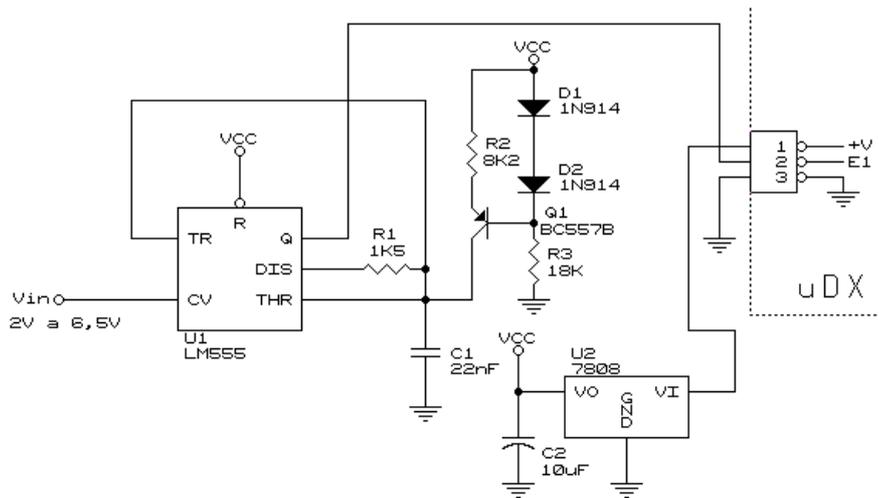


Conversor Tensão / Largura de Pulso (PWM)

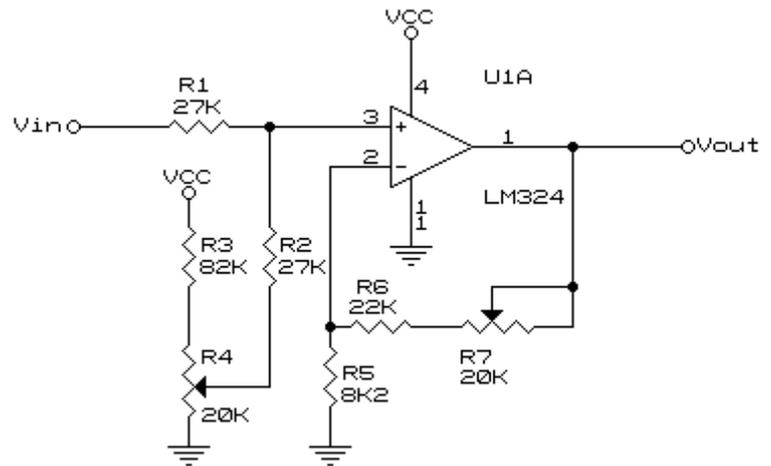
O circuito apresentado faz a conversão de um nível de tensão da entrada para uma largura de pulso proporcional na saída. Utilizando-se o bloco de instrução PWMIn, do μDX, pode-se fazer a leitura desta largura de pulso em um valor de 8 bits a ser colocado em uma variável interna (ver FUNÇÃO - PWMIn).

Foi utilizada a entrada de ajuste de comparação do 555 para que a posição de gatilhamento, sobre a rampa linearizada pela fonte de corrente, mova-se conforme a tensão aplicada. O regulador de tensão 7808 garante uma tensão de alimentação para o circuito com boa estabilidade, já que a tensão disponível no conector de entrada do μDX (+V) não é regulada.

Utilizando-se os limites indicados na figura (2V a 6,5V) o resultado da função PWMIn terá os valores de 50 a 200, aproximadamente. A conversão dentro destes limites é linear.



Como os limites de tensão de entrada estão em uma faixa pouco usual, recomenda-se empregar um circuito adicional de correção da faixa que ajusta os limites de 0 a 1 Volt. Além disso, o circuito da página seguinte, quando conectado à entrada 5 do LM555, permite uma alta impedância de entrada. Com isso, podemos fazer uma entrada analógica de corrente - 4 a 20 mA. Basta colocar um resistor de 47R em paralelo com a entrada e ajustar os trimpots para obter uma leitura correspondente à corrente de entrada no μDX (por exemplo, 40 para 4mA e 200 para 20mA).



Aqui os trimpots $R4$ e $R7$ servem, respectivamente, para calibrar o zeramento e o ganho do circuito em função da tolerância dos componentes. Com este circuito pode-se ajustar para que a medida de 0 a 1 Volt corresponda a valores como 80 a 180 obtidos pela função PWMIn. Depois bastará subtrair 80 do valor da variável para conseguir uma correspondência de 0 a 1 Volt = 0 a 100.

As Saídas

As quatro saídas do μDX são do tipo relé com um contato reversor, isto é, possui um contato central e dois adjacentes, ficando um deles ligado e outro desligado quando a saída está inativa. Quando a saída fica ativa o contato central troca de lado deixando o primeiro contato adjacente desligado e o outro ligado.

Esta particularidade permite que se ligue ou que se desligue algum dispositivo quando a saída for acionada, bastando escolher qual dos contatos adjacentes deva ser empregado.

A serigrafia na tampa da caixa (lado direito da tampa) mostra estas conexões e a representação dos contatos dos relés.

A designação dada às saídas é S1, S2, S3 e S4.

Note que os relés dispõem de isolamento galvânica entre a bobina de acionamento e os contatos. Isto torna possível ligar dispositivos à rede elétrica domiciliar para serem atuados diretamente pelos contatos do relé, sem risco algum para o μDX ou algum outro equipamento conectado na Rede Local DXNET.

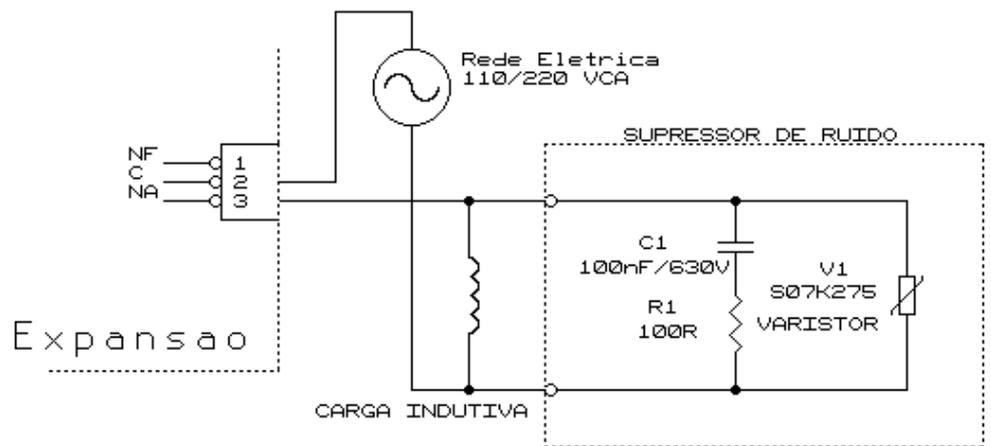
Supressores de Ruído Elétrico

No caso do μDX acionar cargas indutivas (contactoras, válvulas pneumáticas, etc) é necessário instalar supressores de ruído junto à carga indutiva (em paralelo com essa). O supressor evita que, ao abrir o relé do μDX ligado à carga indutiva, forme um arco de alta tensão nos contatos. Este arco, embora não danifique permanentemente o controlador programável μDX, diminui muito a vida útil dos relés e, em casos extremos, pode abortar a execução do programa no μDX, sendo necessário reinicializá-lo. A seguir temos o diagrama elétrico do supressor e sua ligação a uma carga. A DEXTER pode fornecer supressores de ruído já montados para a sua aplicação.

Abaixo temos uma lista de precauções contra ruídos elétricos na instalação do controlador programável:

1. Instalação de supressores de ruído, como já frisado.

2. Separação física dos cabos de sinal (DXNET, alimentação elétrica do μDX, sinais de entrada para μDX) dos cabos de potência (contactoras, motores, etc.). Nos casos em que os cabos devem correr juntos, providenciar um eletroduto metálico (devidamente aterrado) para blindar os cabos de sinal.
3. O terra lógico do μDX (disponível no conector das entradas E1 a E4) não deve ser aterrado.
4. No caso de ruídos advindos da fonte de alimentação, é necessário providenciar uma linha de alimentação separada para o μDX e outros equipamentos eletrônicos da instalação.



A partir da versão 2.1 de placa impressa do Controlador Programável μDX Série 100 foram inseridos vários componentes adicionais para supressão de transientes elétricos. Isso evita que o transiente perturbe o funcionamento do controlador. Entretanto, no caso de cargas indutivas, ainda é interessante a inserção de supressores de ruído elétrico para aumentar a vida útil dos relés de saída do μDX.

Programação em PDE - Utilização do PG

Como mencionado nos capítulos anteriores, junto com o μDX é fornecido um disquete contendo um programa para micro computadores (tipo IBM-PC) chamado PG (Programador Gráfico).

O PG é uma ferramenta com a qual se pode elaborar programas para o μDX (em linguagem PDE) e que permite monitorar e interferir em qualquer μDX conectado na Rede Local DXNET.

Esta linguagem PDE - Programação por Diagrama Esquemático - foi desenvolvida pela DEXTER como um meio de programação intuitiva, de fácil compreensão e que dispensa conhecimentos especializados em electricidade e informática.

Para elaborar um programa basta "pegar" os blocos de instrução que sejam necessários (dispostos em forma de Menu na lateral direita da tela do computador), colocando-os depois na área de programação. Depois liga-se uns aos outros com fios e a representação artística final é a de um diagrama esquemático, com as chaves, relés, temporizadores, fios, fontes de energia e tudo o mais que o programa precisar.

Pequenos textos podem ser inseridos no meio do desenho para explicar alguma operação ou indicar a finalidade das entradas e saídas.

E quando se quiser alterar o programa basta apontar o mouse e mover linhas e blocos que se desejar, apagar ou inserir novos blocos e linhas.

Finalmente, através de um único comando, o programa pode ser enviado a qualquer μDX e posto para funcionar, sem perda de tempo.

Além disso tudo o PG permite monitorar valores de variáveis, forçar outros valores ou ainda ligar algum nodo de qualquer μDX. Por exemplo, através de um só comando pode-se ligar um relé de um μDX que está até 100 metros distante do computador.

Introdução ao PG

Para utilizar o PG é necessário empregar um microcomputador com a seguinte configuração (ou melhor):

- CPU IBM-PC/AT (486,586,686,K6-II,Pentium) ou compatível
- Vídeo VGA colorido ou monocromático
- Mouse

Utilize um Sistema Operacional MS-DOS superior à versão 3.0 ou algum outro capaz de executar programas baseados neste sistema, ou Windows.

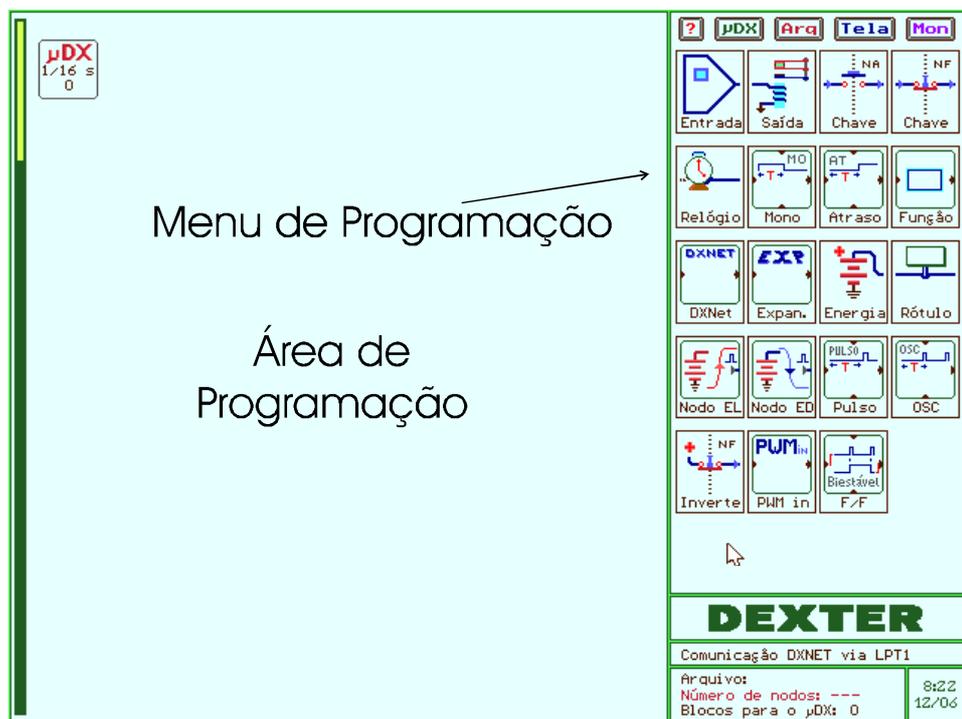
Caso ainda não tenha instalado o PG e os outros arquivos copie o disquete fornecido com o μDX inteiramente para algum diretório ou sub-diretório em seu disco rígido (winchester). Para isto siga as instruções de instalação dadas anteriormente (veja Instalando os arquivos do disquete).

Quando tudo estiver pronto passe para o diretório escolhido para os arquivos do PG. Teclle: PG <enter> e aguarde a inicialização do programa (ou chame o programa via Windows).

A primeira tela que aparece mostra a logo marca do μDX e da DEXTER. No canto esquerdo superior aparece a versão corrente do PG. Para sair desta tela basta pressionar qualquer tecla no teclado do computador ou no mouse.

Depois aparece a tela principal do PG: uma grande área retangular que é onde se desenha os programas, uma área menor onde aparecem os blocos utilizados para o desenho (Menu de Programação) e duas áreas menores sendo uma na parte inferior direita, reservada para mensagem de "Status" e outra no canto superior direito (sem demarcação de borda) onde estão as Teclas de Operação do PG.

Teclas de Operação



Área de Status

Área de Programação

A área de programação mostrada é apenas uma parte da área total disponível. Os desenhos feitos na área de programação podem ser maiores verticalmente e a visualização nesta área de programação muda conforme o desenho vai sendo feito em outros lugares da área total.

Existe uma barra vertical à esquerda da área de programação mostrando um retângulo claro (amarelo se o vídeo for colorido) dentro. Este retângulo claro indica o tamanho proporcional da área de programação relativamente à área disponível. Conforme a posição deste retângulo sobre a barra escura a área de programação desloca-se ao longo da área total disponível, ou vice-versa.

Para fazer a área de programação mostrar outra parte da área total existem três caminhos: utilizando-se as teclas PgUp e PgDn no teclado do computador, movendo o mouse até "bater" no lado inferior ou no lado superior da área de programação (Auto-Pan) ou clicando o mouse ao apontar para o retângulo claro e, sem deixar de pressionar a tecla do mouse, arrastá-lo para cima ou para baixo.

No canto superior esquerdo da área de programação sempre aparece um bloco já colocado. Este bloco (μ DX) não pode ser apagado e nunca haverá mais de um no programa. Sua finalidade é fornecer ao compilador dados sobre a configuração de funcionamento que o μ DX deve assumir ao receber o programa. Mais adiante ele será descrito em detalhes.

Menu de Programação

Nesta área aparecem todos os blocos de desenho que estão disponíveis para a elaboração dos programas. Como será explicado mais adiante, alguns destes blocos representam instruções específicas para o μ DX e outros servem para auxiliar a programação ou simbolizar alguma função.

Alguns blocos estão presentes em quantidade limitada enquanto outros podem ser utilizados à vontade. Quando este limite for excedido o retângulo que delimita o bloco em questão ficará amarelo, aparecendo a palavra VAZIO embaixo.

Os blocos podem ser "capturados" utilizando o mouse: aponte-o sobre o bloco desejado e pressione sua tecla da esquerda. A figura do bloco descolará de seu lugar e deslizará pela tela do computador conforme o mouse se movimentar (note-se que quando "pegar" a figura a seta do mouse desaparece). Depois de capturar a figura você deve soltar a tecla do mouse. Para fixar a figura na área de programação basta posicioná-la no lugar escolhido e pressionar mais uma vez a tecla esquerda do mouse.

Veja mais adiante as instruções para a programação.

Teclas de Operação

Das cinco teclas a única que não abre novos Menus é a de ?. Esta tecla serve para mostrar uma pequena tela de ajuda que fornece instruções simplificadas sobre a Tela Principal do PG.

Devido a quantidade de informação necessária para explicar cada uma das outras teclas foi elaborado um sub-capítulo específico para tratar delas que você poderá consultar mais adiante.

Área de Status

Neste retângulo emoldurado aparecerão duas linhas de texto informando a quantidade de blocos para o μDX (portanto instruções) que já foram colocados no atual desenho de programa e um indicador de quantos nodos foram encontrados após a compilação.

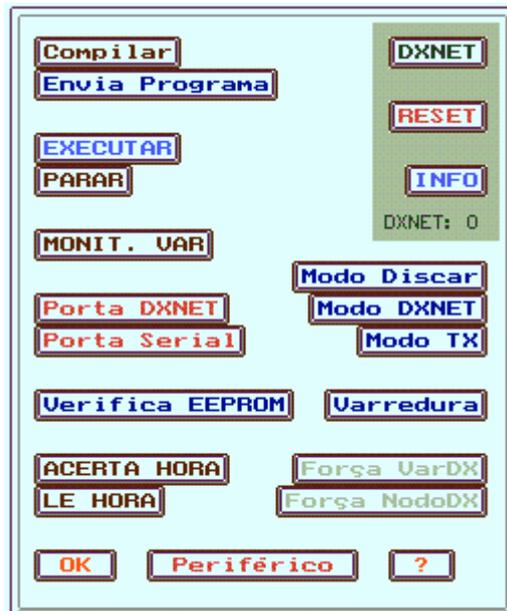
Com estes valores você poderá dimensionar melhor a construção de algum programa comparando-os com seus limites: 127 blocos de instrução para o μDX e 64 nodos de interconexão (ou, no caso do μDX+, 256 blocos de instrução e 192 nodos de interconexão).

Além disso, a área de status mostra a data e hora do computador. Note que não se trata da hora e dia da semana do μDX, acessíveis via tecla de operação - μDX.

Note que, abaixo do logotipo DEXTER, aparece uma linha informando qual o tipo de comunicação que está sendo usada. Ao iniciar o PG esta linha indica "Comunicação DXNET via LPT1". É possível efetuar comunicação com μDXs via linha serial ou via linha telefônica (em ambos os casos é necessária a utilização de modem para μDX).

Tecla de Operação - μDX

Com esta tecla abre-se uma janela com diversas opções relativas ao μDX e à Rede Local DXNET. Todas elas são ativadas utilizando-se o mesmo tipo de representação gráfica de tecla.



COMPILAR

Examina o programa desenhado na área de programação, montando a seqüência de códigos apropriada para enviar posteriormente ao μDX. Calcula automaticamente a numeração dos nodos de ligação entre os blocos de instrução, atribuindo um número pré-estabelecido para as entradas e saídas (0 a 7) e nas ligações que tiverem o bloco de rótulo.

Depois de compilado o programa, o número de nodos encontrados é apresentado na tela principal, na Área de Status.

ENVIA PROGRAMA

Esta tecla permite transmitir o programa desenhado para o μDX. Caso o programa não tenha sido compilado ainda, o compilador é automaticamente executado antes do envio.

Uma pequena janela será mostrada para que o usuário confirme o endereço na Rede Local DXNET do μDX que vai receber o programa, ou permite que a tarefa seja cancelada.

No início da transmissão o PG envia ao μDX um comando de PARAR que o faz interromper o programa que estiver executando (o comando é dado mesmo que o μDX já esteja parado). Depois o programa é enviado e, por fim, o PG manda um comando de RESET, que força o μDX a inicializar toda a memória.

Note que o μDX não executa o novo programa antes de receber ordem para tal (após receber programa, o μDX permanece parado, por segurança. É necessário pressionar a tecla EXECUTAR para rodar o novo programa).

EXECUTAR

Este comando força o μDX a executar o programa. Caso o μDX já esteja em execução a instrução é ignorada.

PARAR

Este comando faz o μDX interromper o programa que estiver em execução, inclusive mantendo o estado dos nodos (até os de saída).

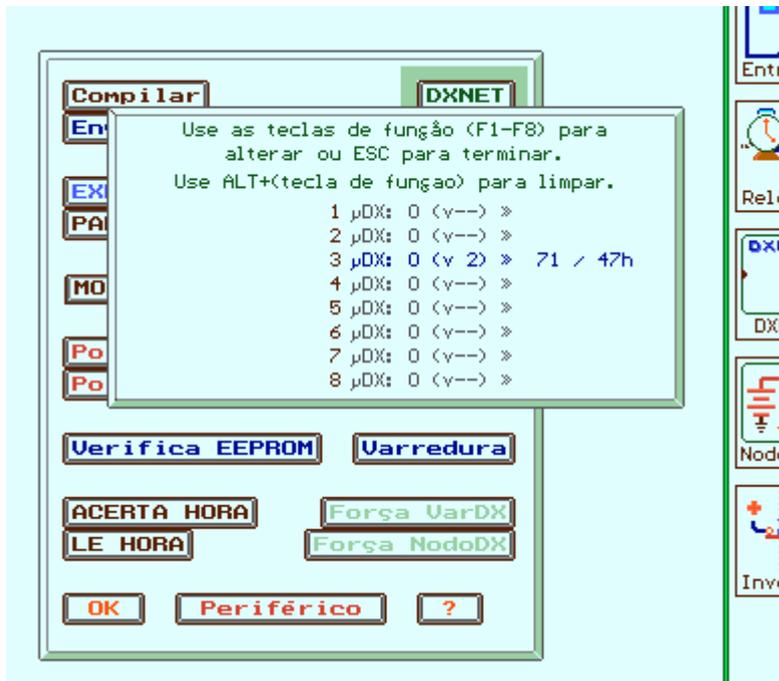
O programa pode recomeçar a execução depois que o μDX receber um comando de EXECUTAR ou um comando de RESET seguido de comando EXECUTAR, para reinicializar variáveis e recomeçar programa do início.

MONIT. VAR

Permite monitorar até oito variáveis que podem estar em qualquer μDX conectado na Rede Local DXNET.

Quando seleccionado faz aparecer uma nova janela com o seguinte funcionamento:

- Para alterar ou iniciar a monitoração de alguma das oito linhas pressione alguma das teclas F1 até F8.
- Para apagar uma das linhas pressione, juntas, as teclas ALT e a tecla de função (F1 até F8) desejada.
- Para encerrar a operação e fechar a janela tecle ESC.



A figura mostra o aspecto desta janela de monitoração na tela do computador. Como pode ser visto o PG está monitorando a variável V2 do μDX endereçado como 0 (zero). Lembre-se: o endereço zero somente pode ser utilizado quando existe apenas um μDX conectado ao computador; caso contrário qualquer μDX (via DXNET) responderia à monitoração, causando erros por colisão de dados.

CONJ.

Só é operacional para controladores μDX Plus. Abre uma pequena janela onde o usuário pode escolher um novo conjunto para comunicação em DXNET. Cada conjunto DXNET é composto de 16 endereços DXNET. Os valores possíveis vão de 0 a 15. O conjunto DXNET permite expandir os endereços DXNET a 256 endereços possíveis (16 conjuntos, cada um com 16 endereços DXNET). Note que este endereço será empregado por qualquer uma das três formas de comunicação possíveis do PG: comunicação via porta paralela, comunicação via porta serial ou ainda comunicação via rede telefônica.

Como existe limitação de corrente na rede DXNET, só é possível a conexão de mais de 15 dispositivos em rede via rede RS485 ou rádio-comunicação.

DXNET

Abre uma pequena janela onde o usuário pode escolher um novo endereço para comunicação em DXNET. Este endereço será o endereço de destino de todos os comandos ou leituras efetuadas. Os valores possíveis vão de 0 a 15. Note que este endereço será empregado por qualquer uma das três formas de comunicação possíveis do PG: comunicação via porta paralela, comunicação via porta serial ou ainda comunicação via rede telefônica.

RESET

Força um comando de RESET (re-inicialização) no μDX endereçado (via DXNET). Após o RESET o μDX apaga todas as variáveis (colocando o valor 255 em cada uma) e "flags" internos de controle. Os leds das entradas piscam, juntos, por cerca de um segundo para indicar a ocorrência.

INFO

Lê do μDX, no endereço especificado pela tecla DXNET, o estado atual (Status) mostrando as seguintes informações: Modelo do μDX, Versão do Firmware, Estados de execução (Parado ou Executando), duração de ciclo programada e indica se houve falta de energia. A partir da versão 3.2 do Programador Gráfico (PG), a tecla INFO também apresenta o nome do programa existente na memória do μDX. Como este nome ocupa as 4 posições finais desta memória, o programa deve ter menos de 124 blocos para que esta facilidade esteja habilitada (ou, no caso do μDX+, 252 blocos). Além disso, mostra informações dos periféricos programáveis disponíveis para o μDX: Modem, Conversor Analógico/Digital, Interface Homem/Máquina e Registrador.

Porta DXNET

Permite escolher como porta paralela para a comunicação DXNET entre os endereços 378h, 278h e 3BCh. Abaixo do endereço absoluto da porta aparece sua correspondência com o nome da porta, ou seja, LPT1, LPT2 e LPT3 (quando o computador possui mais de uma porta paralela). A porta paralela selecionada é salva no arquivo CFG_PG.PG, e sempre que o PG é executado ele seleciona esta porta para comunicação DXNET. A porta selecionada aparece em amarelo (mais claro, no caso de vídeo monocromático), enquanto as outras ficam em azul.

Porta Serial

Permite escolher como porta serial para a comunicação serial ou telefônica entre os endereços 3F8h, 2F8h, 3E8h e 2E8h. Abaixo do endereço absoluto da porta aparece sua correspondência com o nome da porta, ou seja, COM1, COM2, COM3 e COM4 (quando o computador possui mais de uma porta serial). A porta serial selecionada é salva no arquivo CFG_PG.PG, e sempre que o PG é executado ele seleciona esta porta para comunicação serial ou telefônica. A porta selecionada aparece em amarelo (mais claro, no caso de vídeo monocromático), enquanto as outras ficam em azul.

Comunicação serial:

No caso de comunicação via serial (RS232C ou RS485) a porta serial do microcomputador deve ser ligada a porta serial do Modem para Controlador μDX (conector DB-9). Se o modem para μDX tiver versão 2.3 ou superior, é possível programar-se a taxa de transmissão do modem para conexão serial (via cabo RS232C ou RS485). Pode-se escolher as seguintes taxas de transmissão para o modem do μDX: 300, 600, 1200, 2400, 4800 e 9600 bps. Devido as limitações da velocidade da rede DXNET não há um ganho significativo de velocidade de comunicação para taxas maiores que 2400 bps. Assim, recomenda-se a taxa de 2400 bps para comunicação serial via cabo.

O programa PG permite programar a porta serial do microcomputador em 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 e 115200 bps. No caso de comunicação via porta serial, a velocidade máxima fica limitada a 9600 bps, já que é a velocidade máxima de comunicação serial do modem para μDX (se for versão inferior a 2.3 esta velocidade é fixa em 300 bps).

Caso seja utilizada uma taxa de comunicação maior que 2400 bps, recomenda-se o uso de interrupção para a comunicação serial, de forma que não haja perda de caracteres. Selecione um endereço de interrupção para a porta serial selecionada (IRQ). Verifique este valor chamando Painel de Controle -> Sistema -> Gerenciador de Dispositivos -> Portas (COM & LPT), no sistema operacional Windows. A seguir clique sobre a porta serial que será utilizada para comunicação com modem para μDX e selecione Recursos. Nesta janela aparecerá o endereço da porta serial (3F8h, 2F8h, 3E8h, 2E8h) e a interrupção utilizada (IRQ n). Normalmente, a porta 3F8h (COM1) utiliza IRQ4, e as demais utilizam IRQ3. Se selecionarmos IRQ desligada a recepção de dados seriais será feita por varredura da porta serial.

Por fim, desabilite as linhas de controle (RTS desligado), uma vez que estas linhas não são utilizadas pelo Modem para μDX.

No caso de conexão via RS485, é necessário intercalar um Conversor RS232C para RS485 entre o microcomputador e o Modem para μDX, além de conectar as linhas A e B da RS485 nos pinos corretos do conector DB-9 do Modem (ver pinagem no manual do Modem para μDX). Note que a interface RS485 é opcional no Modem para Controlador Programável μDX.

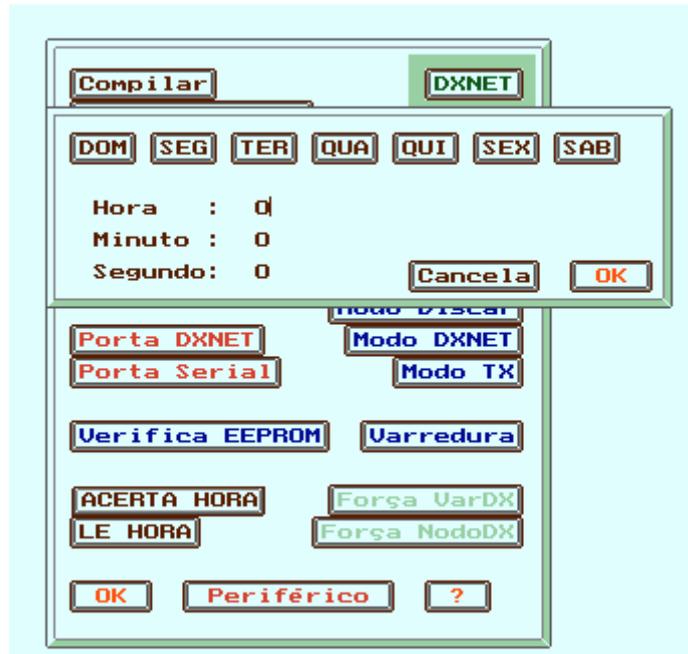
Comunicação Telefônica:

Para conexão via linha telefônica é necessário que o microcomputador possua um modem instalado. A primeira etapa consiste em descobrir qual a porta serial ocupada pelo modem. Vá em Painel de Controle -> Modems -> Propriedades, no Windows 95 ou 98. Anote a Porta (COM1, COM2, COM3, COM4) e a velocidade máxima de conexão. A seguir, especifique esta porta serial no programa PG (item [Porta Serial] na janela [μDX]). A velocidade de comunicação deve ser testada: tente 300 bps (velocidade do Modem para μDX), IRQ desligada, RTS ligado.

Caso não funcione tente velocidade máxima de conexão (115200 bps) do modem do microcomputador, IRQ ativa (normalmente, COM1 está na IRQ4 e COM2, COM3, COM4 na IRQ3. Verifique em Painel de Controle -> Sistema -> Gerenciador de Dispositivos -> Portas (COM & LPT) -> Recursos, no Windows 95 ou 98), RTS ativo.

ACERTA HORA

Permite corrigir o relógio interno do μDX. Uma janela é aberta para o usuário informar o dia da semana e a hora atual. A cada valor digitado tecla ENTER. Após tecla ENTER, depois de digitar os segundos, o horário é transmitido ao μDX. Para escolher o dia da semana selecione com o mouse uma das teclas desenhadas que aparecem no topo da janela aberta.



LÊ HORA

Abre uma pequena janela para mostrar a hora atual do relógio do μDX. Não deve ser utilizado com endereço de DXNET igual a 0 (zero) quando existe mais de um μDX conectado na rede.

Força VAR_DX

Força um valor (8 bits) em uma variável do μDX endereçado na rede DXNET.

Força Nodo_DX

Permite forçar um nodo qualquer do μDX endereçado na rede DXNET. Lembre-se que, devido a lógica "wired-or" o forçamento pode ser apenas como LIGAR. A opção de DESLIGAR serve para deixar de forçar como ligado. Um quadrado (LED) indica o estado atual do nodo: preto - desligado; vermelho - ligado.

?

Mostra uma janela de ajuda ("help") com dicas rápidas de como utilizar as opções do Menu μDX. A partir da versão 2.0 do software PG todas as teclas de ajuda ("help") foram representadas por "?".

Varredura

Esta tecla faz uma varredura em todos endereços DXNET e lista todos os dispositivos conectados ao microcomputador, via porta paralela especificada na tecla <Porta DXNET>, ou via porta serial especificada na tecla <Porta Serial>. No exemplo abaixo, foi detectado um μDX no endereço 5 e um Modem no endereço 7. Note que o μDX responde no seu endereço (5), naturalmente, e também no endereço 0. Já outros dispositivos, como Modem, Conversor Analógico/Digital ou Interface Homem/Máquina respondem apenas no seu endereço.

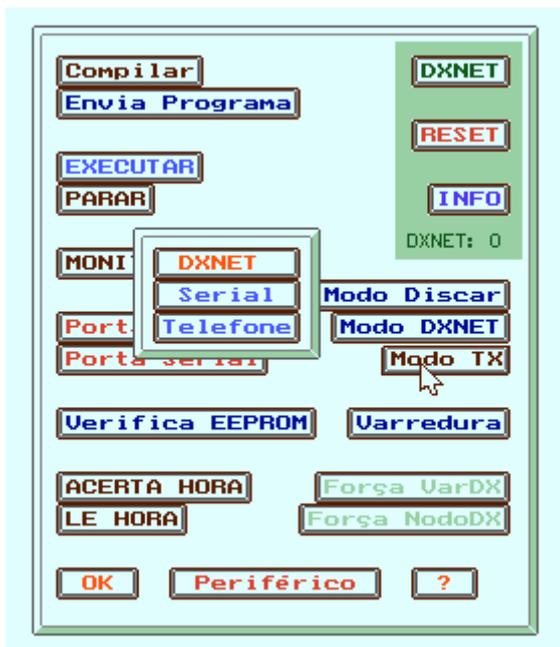


Modo DXNET

Esta tecla permite escolher o modo de operação do microcomputador na DXNET entre Normal e Modem. O modo Normal é o "default", ou seja, o PG sempre inicializa neste modo. O modo Normal permite rápido acesso à rede local DXNET. Já no modo Modem, o acesso é estendido, via modems ligados à rede telefônica, a qualquer rede DXNET remota. No modo Modem, o microcomputador pode acessar um μDX remotamente via telefone. Para isso, a velocidade de transmissão é diminuída. Maiores detalhes no manual do Modem para Controlador Programável μDX. Note que, no caso de usar-se o modem interno do computador para a comunicação via rede telefônica - ou seja, selecionar comunicação telefônica via tecla <Modo TX> - não é necessário selecionar modo Modem. O PG faz isso automaticamente e, ao retornar para comunicação DXNET, retorna ao modo Normal.

Modo TX

O modo TX é o modo de transmissão de dados entre o microcomputador rodando o software PG e os μDXs e periféricos interligados por rede DXNET. É possível escolher entre três modos de transmissão de dados: DXNET, Serial ou Telefone.



O modo DXNET é o modo usual. Através da porta paralela selecionada via tecla <Porta DXNET> o microcomputador se comunica com os μDXs e periféricos. Utiliza-se o cabo de comunicação que acompanha o controlador μDX.

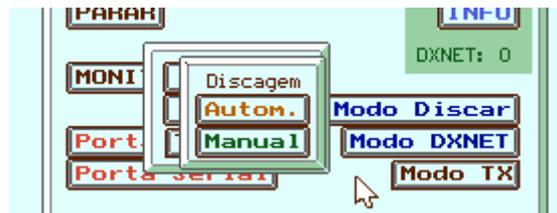
Já o modo Serial pressupõe a existência de um modem para μDX conectado a porta serial do microcomputador especificada via tecla <Porta Serial>. Assim, o microcomputador se comunica via porta serial (RS-232C ou RS485) com o modem para μDX, que converte os dados para DXNET. É necessário um cabo especial, capaz de conectar a porta serial do computador ao conector DB-9 existente no Modem para μDX (ver manual do modem).

Já a terceira opção é utilizar o modem do computador (instalado na porta serial especificada via tecla <Porta Serial>) para comunicação de dados via rede telefônica. Neste caso, o computador irá discar para um número selecionado, que deve estar ligado a uma instalação com modem para μDX.

Uma vez estabelecida a conexão, todas as operações do PG estarão disponíveis remotamente, inclusive descarregar novo programa no μDX remoto! A atual versão do PG (versão 5.9) permite uma ampla compatibilidade com placas de modem para microcomputadores IBM-PC compatíveis. Modems da U.S. Robotics costumam estabelecer conexão tanto com baud-rate de 300 bps (velocidade do modem para μDX) quanto na máxima velocidade de conexão, especificada no Windows (ver item Porta Serial, algumas páginas atrás). Já outros modems exigem que a conexão seja efetuada na taxa máxima de transferência de dados (normalmente, 115200 bps), e eles negociam a taxa de transmissão após a discagem para 300 bps.

No caso de versões de PG anteriores a 4.5, nem todas as placas de modem para microcomputadores IBM-PC compatíveis funcionam. Normalmente, modems da U.S. Robotics funcionam sem problemas (tanto placas internas quanto modems externos). Se for o caso requirite a Dexter uma versão atualizada do PG.

Uma vez selecionado o modo Telefone, o programa pergunta se a discagem será efetuada automaticamente ou não:



Tanto no caso de discagem automática quanto manual, o programa pergunta o endereço DXNET do modem para μDX remoto e sua senha. Para que a conexão se efetue é necessário que estes dados estejam corretos (veja manual do modem para detalhes sobre a senha de acesso).

Se for utilizada discagem automática, será apresentada a janela abaixo, permitindo digitar um número telefônico de até 16 dígitos, e um prefixo para obtenção de linha externa (caso o modem do computador esteja conectado a um ramal):



Se tudo estiver correto, deve aparecer uma mensagem dizendo que a conexão foi efetuada. A partir deste momento todas as operações do PG estão atuando na instalação remota. Para desconectar basta selecionar a tecla <Modo TX> novamente e <Telefone>.

Modo Discar

Esta tecla seleciona entre duas formas de discagem possíveis: pulso ou tom. O PG salva este dado no arquivo CFG_PG.PG, reiniciando com a última seleção. Atualmente, quase todas centrais telefônicas brasileiras aceitam discagem por tom (mais rápida).

Verifica EEPROM

Permite verificar se o conteúdo da memória de programa do μDX coincide com o programa existente na tela do PG.

Periférico

Esta tecla acessa as janelas de programação dos periféricos para o controlador programável μDX: Modem, Interface Homem/Máquina, Conversor Analógico/Digital, Registrador e Código de Barras. Maiores detalhes de cada uma destas janelas podem ser obtidos nos respectivos manuais destes periféricos.

Apenas a opção Código de Barras será mais explicitada aqui, uma vez que não se trata de um periférico, e sim um Controlador μDX com firmware especial (versão 6.6).

Código de Barras:

O Controlador Programável μDX Série 100 versão 6.6 permite a leitura de código de barras 2 de 5 intercalado (10 dígitos), ao conectar-se o Leitor de Código de Barras a sua entrada E1. O bloco PWMin, quando utilizado no programa aplicativo deste μDX, e direcionado para leitura na entrada E1, lê o valor do código de barras e armazena em 4 variáveis sucessivas do controlador (se o bloco PWMin estiver programado para Ent: 1 (entrada E1) e R: v2 (variável v2), por exemplo, a leitura do código será gravada em v2, v3, v4 e v5). Note que o cartão com o código pode ser passado indistintamente nos dois sentidos. O μDX se encarrega de ordenar o valor lido.

A tecla [Periférico], quando selecionada para [Cód. Barras], permite descarregar números de códigos válidos (o total de códigos que pode ser descarregado depende do tamanho do programa aplicativo existente no μDX versão 6.6. Cada código ocupa cerca de 2 blocos de programação). Caso o código de barras lido corresponda a um dos valores cadastrados na memória do controlador, o bloco PWMin pulsa seu nodo de saída ao ler este código. Com isso é possível, por exemplo, acionar uma fechadura elétrica via monoestável, permitindo o acesso a áreas restritas para pessoas munidas de cartões com códigos de barras específicos.

Inicialmente o programa PG lê todos os valores de códigos de barras já cadastrados no controlador μDX versão 6.6 e os apresenta na tela (caso não existam códigos cadastrados esta janela aparecerá limpa). Para editar um código existente basta apontar para o código com o mouse e pressionar a tecla esquerda desse. Para inserir um código aponte para uma área livre e pressione a tecla esquerda do mouse. Para retirar um código do cadastro basta editá-lo e limpar a área de edição. Ao fechar a janela no PG, o arquivo com os códigos de barras válidos são transferidos para a memória não-volátil do controlador μDX versão 6.6.

OK

Fecha a janela de μDX, voltando o controle à Tela Principal.

Tecla de Operação - Arquivo

Esta tecla abre uma janela que permite ler, gravar ou mesmo apagar arquivos de programas para o μDX. Nesta janela é que existe a tecla para terminar o PG.



Existem duas linhas de entrada de dados alfa-numéricos: Caminho e Nome. Elas servem para indicar ao PG qual o diretório e o nome do arquivo que se quer trabalhar. A digitação está sempre ativa e o conteúdo das linhas, a cada momento, é considerado válido. Um cursor, em forma de uma linha vertical piscante, indica onde aparecerá o próximo carácter digitado. Para trocar de linha basta pressionar a tecla TAB no computador (ou aponte a linha com o mouse e pressione a tecla esquerda). A tecla DEL, no teclado do computador, apaga o conteúdo da linha.

LER

Faz o PG ler o programa guardado no arquivo indicado por CAMINHO\NOME.

Gravar

Faz o PG gerar um arquivo, conforme CAMINHO\NOME, guardando nele o programa atualmente existente na área de programação.

Apagar

Apaga do disco magnético o arquivo de programa para μDX especificado em CAMINHO\NOME.

Sai do PG

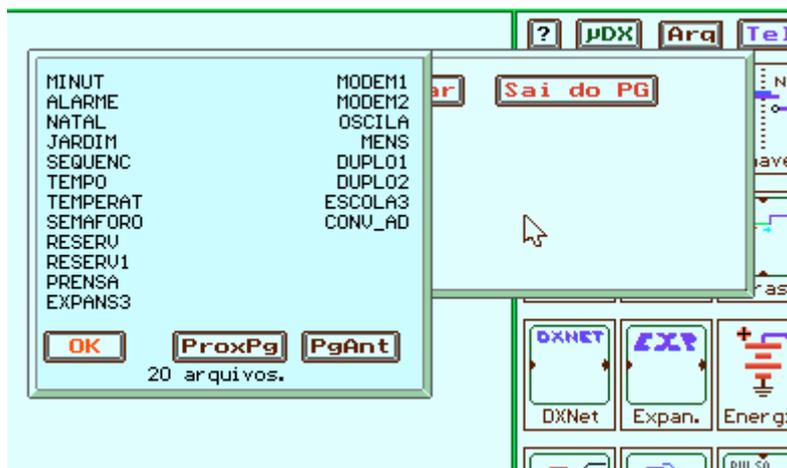
Permite terminar a execução do PG, retornando ao DOS ou Windows.

DIR

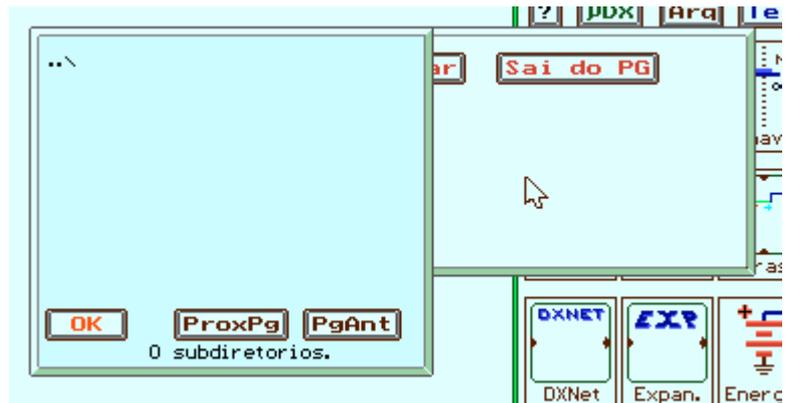
Se o cursor estiver piscando no campo Nome, mostra uma lista dos arquivos de programa para μDX presentes atualmente no diretório indicado por CAMINHO. Nesta lista pode-se apontar com o mouse um nome qualquer e pressionar a tecla esquerda. Assim o nome escolhido será transferido para a linha NOME.

As teclas <ProxPg> e <PgAnt> permitem folhar páginas com as listas de arquivos, no caso desta lista exceder a capacidade da janela.

Abaixo destas teclas é apresentado o número total de arquivos existentes para controlador programável μDX no subdiretório especificado em CAMINHO.



Já se o cursor estiver piscando em Caminho, ao pressionar a tecla <DIR> é apresentada a estrutura de diretórios existentes no disco rígido, permitindo navegar nos diretórios do disco rígido. Para ir para um diretório, basta clicar com o mouse sobre o nome deste diretório. Para sair do diretório, indo para o imediatamente acima, clique nos dois pontos (..\). Abaixo das teclas é apresentado o número de subdiretórios existentes no diretório corrente.



?

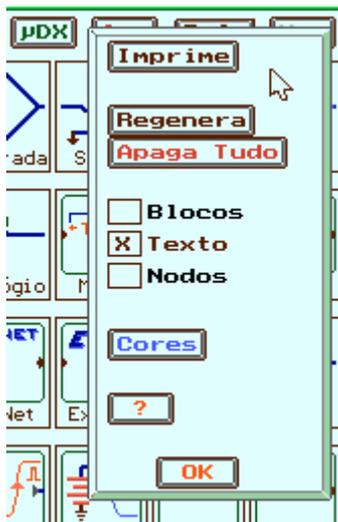
Mostra uma janela com a explicação simplificada das opções deste Menu de Arquivo. A partir da versão 2.0 do software PG todas as teclas de ajuda ("help") foram representadas por "?".

OK

Fecha a janela de Arquivo, voltando o controle à Tela Principal.

Tecla de Operação - Tela

Esta tecla abre uma janela de Menu com opções genéricas a respeito da Tela Principal e do PG.



Imprime

Permite imprimir o programa atualmente na área de programação em uma impressora tipo MATRICIAL (EPSON ou compatível) ou em uma impressora LASER (que reconheça os comandos PCL5 (linguagem padronizada pela Hewlett-Packard para impressoras Laser Jet), como a EPSON AL-1500 ou qualquer impressora laser da HP). Impressoras "desk-jet" (jato de tinta) também podem ser utilizadas. Neste caso selecione impressora tipo JATO DE TINTA. Algumas impressoras jato de tinta permitem impressão tipo LASER (em especial impressoras jato de tinta HP).

Depois de escolhida a impressora aparecerá uma janela de aviso solicitando que o usuário desconecte o adaptador para DXNET e conecte a impressora. Depois de terminada a impressão outra janela solicitará que a impressora seja desligada e que o cabo para DXNET seja novamente conectado.

O retorno da conexão do adaptador para DXNET não é obrigatório, sendo apenas utilizado quando houver alguma comunicação via DXNET para ser feita.

Regenera

Redesenha todo o programa feito na área de programação.

Apaga Tudo

Permite limpar toda a área de programação para que se comece a desenhar um novo programa.

Opção NODOS

Este quadrado à esquerda do nome (NODOS) indica se a numeração dos nodos, atribuída depois de feita a compilação, deve aparecer junto das linhas desenhadas após uma regeneração (tecla Regenera).

Para ativar ou desativar esta opção aponte o mouse no quadrado e pressione a tecla esquerda.

Esta opção facilita o controle sobre os programas enviados e em execução no μDX, mostrando qual o número de determinado nodo que se queira forçar como ligado (ver Tecla de Operação - μDX / Força NodoDX).

Portanto, para que o número dos nodos apareça no desenho é necessário que esta opção esteja selecionada e após que seja mandado REGENERAR o desenho.

Opção TEXTO

Como a opção descrita anteriormente, esta indica ao PG se deve mostrar ou não os textos que existam no meio do programa desenhado, a cada regeneração.

Opção BLOCOS

Este quadrado à esquerda do nome (BLOCOS) indica se a numeração dos blocos que contém constantes deve aparecer junto dos blocos após uma regeneração (tecla Regenera). Para ativar ou desativar esta opção aponte o mouse no quadrado e pressione a tecla esquerda.

Os blocos que podem conter constantes são: Relógio, Monoestável, Atraso, Pulso, Oscilador e Função. Esta constante pode ser editada via Interface Homem/Máquina (neste caso atribua um número de constante para cada bloco via tecla <P> ou <C> do teclado do microcomputador), ou via PGR (Programa Gerenciador Remoto). No caso do PGR, assim como os nodos têm numeração que deve ser informada ao programa para que ele monitore estes nodos, as constantes editáveis devem ter seu número de bloco informado ao programa PGR para que este permita a leitura ou modificação destes valores (set-points).

CORES

Abre uma janela mostrando as cores disponíveis (16) para que o usuário tenha como alterar as tonalidades conforme desejado. Note-se que apenas as cores 0, 1, 2, 3, 4, 5 e 7 podem ser modificadas.

Utilize as teclas desenhadas + e -, de cada componente cromático, para fazer o ajuste. Para terminar pressione ESC. As modificações feitas serão sempre armazenadas em arquivo. Utilize a tecla Cores Originais para obter as tonalidades padrão da DEXTER.

?

Abre uma janela com explicações simplificadas sobre o Menu de Tela.

OK

Fecha a janela do Menu de Tela. Se as opções NODOS, BLOCOS ou TEXTO tiverem sido modificadas faz uma regeneração automaticamente.

Opção Inglês

Comuta a linguagem utilizada no PG para inglês. É necessário reinicializar o PG para que este comute para a língua inglesa (disponível para versão 4.1 ou posterior do PG).

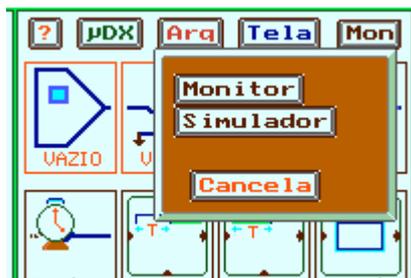
Calibra

Calibra a velocidade da comunicação DXNET, efetuada com Controlador μDX via porta paralela. Esta calibração é necessária ao rodar o programa PG em janela DOS, dentro do sistema operacional Windows 95 ou 98 (em DOS puro ou Windows 3.x a calibração é desnecessária). Conecte um controlador μDX a porta paralela do microcomputador e pressione [Calibra]. Uma vez efetuada a calibração, o valor médio obtido é salvo no arquivo CFG_PG.PG e não há mais necessidade deste procedimento (obviamente, se o PG for transferido para outro computador com velocidade distinta do que foi feita a calibração, esta terá que ser refeita).

ATENÇÃO: O software PG - Versão 4.1 ou superior permite ser executado a partir do sistema operacional Windows 95,98 ou Millenium, em janela DOS. Entretanto, para efetuar comunicações DXNET via porta paralela é necessário calibrar o software na primeira execução (depois não é mais necessário tal procedimento). Para isso, ligue um controlador μDX a porta paralela do computador, chame o software PG através do Windows e selecione a janela [Tela]. A seguir pressione a tecla [Calibra]. Uma vez feita a calibração ela é salva no disco rígido do computador, e não é necessário repeti-la em chamadas posteriores ao software PG.

Tecla de Operação - MON

Esta tecla permite seleccionar o monitoramento ou simulação de circuitos, duas ferramentas poderosas do PG para a depuração de programas para o μDX.



Monitor

Faz o PG monitorar o estado dos nodos internos de um μDX, indicando no desenho do programa as ligações, entradas, saídas e chaves que estiverem ativas. Com isso, é possível observar o que ocorre internamente no controlador. Durante a monitoração o PG fica inteiramente dedicado a esta tarefa. A tecla ESC permite sair da monitoração, as teclas PgUp e PgDown permitem visualizar outra região da área de programação e, pressionando a tecla esquerda do mouse sobre blocos ou nodos é possível ativá-los.

Ou seja, além de permitir a observação dos nodos e blocos do programa, via mouse é possível ligar e desligar determinados nodos, forçando condições para testar o programa. Note que, assim como no caso de tecla <Força Nodo>, já descrita no menu μDX, a ativação de nodos durante a monitoração também é feita por lógica OU. Assim, se o nodo estiver desligado, ao clicar sobre ele com o mouse, ele será ligado. Ao clicar novamente ele se desligará. Já se o nodo estiver originalmente ligado devido ao programa, não adiantará clicar sobre ele para tentar desligá-lo.

Os nodos (fios de conexão entre os blocos) são representados em cor azul quando desenergizados e em cor vermelha quando energizados. Os blocos assumem cor vermelha e amarela quando têm sua saída energizada (seja ou não devido a sua própria condição), e as chaves e relés de saída apresentam uma animação na tela, para representar seu estado (aberto ou fechado, ligado ou desligado).

Na versão 3.5 do PG foi incluído um teste para verificar se o programa existente na memória do μDX corresponde ao existente na tela do PG (a mesma função é disponível via tecla <Verifica EEPROM> no menu μDX). Só se isso for verdade o programa entra em monitoração. Além disso, caso o programa na tela do PG não tenha sido compilado, esta função será efetuada automaticamente.

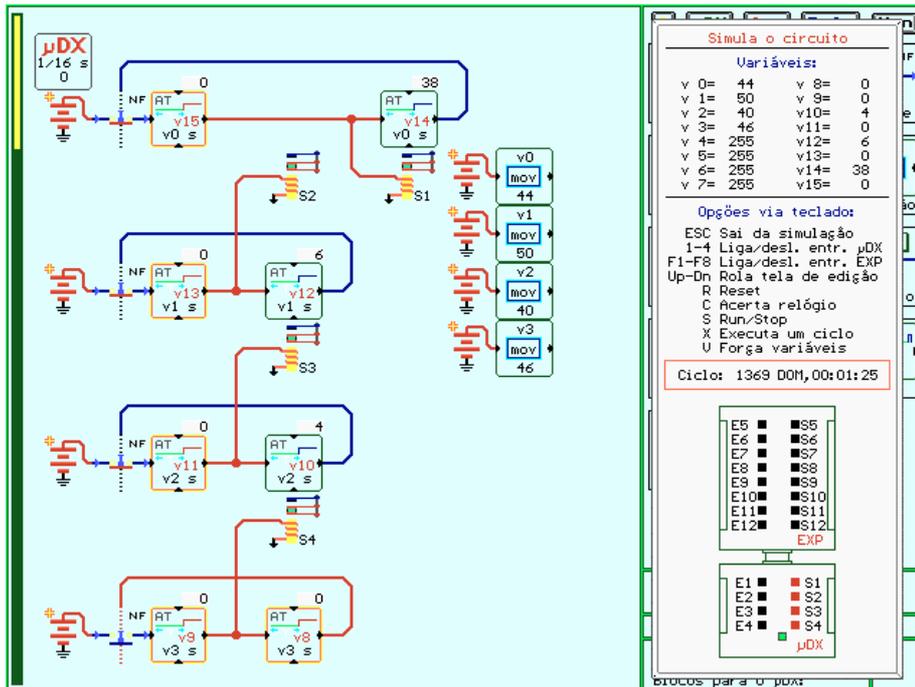
ATENÇÃO: Para que o PG mostre a monitoração corretamente é preciso que:

- ☐ O μDX esteja em execução.
- ☐ O μDX esteja conectado ao computador e tenha o mesmo endereço DXNET que o escolhido para o PG (veja Tecla de Operação - μDX / DXNET).
- ☐ O programa que está no μDX também esteja na área de programação.
- ☐ O PG tenha compilado o programa (de forma que ele possa conhecer quais nodos representam quais conexões).

O Monitor serve para verificar o comportamento de um programa aplicativo dentro do controlador μDX, facilitando a correção de erros e permitindo interagir com o controlador.

Simulador

O simulador, como o nome diz, simula o comportamento de um programa aplicativo dentro de um Controlador μDX + Expansão de Entradas/Saídas virtuais. Ao contrário do monitor, não necessita de nenhum μDX conectado ao computador. Ao entrar no simulador, é mostrado na tela um desenho representando o μDX+Expansão simulados.



Acima temos um exemplo de simulação do programa NATAL.UDX. Note que é possível, com o simulador, andar o programa passo a passo (tecla <X>), ou rodar o programa continuamente (tecla <S>). Acima dos blocos de temporização são impressos os valores das variáveis usadas para a temporização. Uma simulação de um circuito como o acima, principalmente passo a passo, é uma poderosa ferramenta de aprendizado e depuração. As seguintes instruções estão disponíveis no simulador:

<ESC>	Sai do simulador, retornando a tela principal do PG.
<1> a <4>	Liga/desliga as entradas do μDX virtual.
<F1> a <F8>	Liga/desliga as entradas da Expansão virtual.
<PgUp>	Rola a tela de programa para cima.
<PgDn>	Rola a tela de programa para baixo.
<R>	Reseta μDX virtual.
<C>	Acerta relógio de tempo real do μDX virtual.
<S>	Rodar/parar programa no μDX virtual.
<X>	Executa um passo do programa no μDX virtual.
<V>	Força variáveis do μDX virtual.
<+> e <->	Aumenta ou diminui a velocidade de simulação.

Note que, além do desenho dos equipamentos e um descritivo das teclas acima, o simulador apresenta o valor de todas as variáveis do μDX simulado, o horário do relógio de tempo real e um contador de ciclos de execução do programa. Caso o programa na tela do PG esteja programado para velocidade de 1/16s, o contador de ciclos incrementa 16 vezes para cada incremento dos segundos no relógio. Da mesma forma que no Monitor, o Simulador também efetua uma animação na tela, mostrando chaves fechando e relés batendo os contatos quando energizados.

Caso esteja simulando programa para μDX+ (Plus), o simulador apresenta instruções adicionais:

<↑> e <↓>	Rola Variáveis.
<<-> e <->>	Rola Expansões.

Ou seja, as setas permitem comutar quais as variáveis do simulador devem aparecer na tela (já que o μDX+ possui 64 variáveis e não havia espaço para representá-las simultaneamente). Também permitem comutar qual das quatro expansões possíveis de serem conectadas ao μDX+ deve ser monitorada no momento.

O Simulador serve para verificar o comportamento de um programa aplicativo dentro do próprio ambiente do Programador Gráfico (PG), facilitando a depuração e o aprendizado, antes de utilizar o controlador μ DX no mundo real.

Elaborando Programas

A criação de um programa para o μDX é muito fácil e pode ser dividida em quatro etapas:

- Colocação dos blocos na área de programa
- Interligação dos blocos com linhas (fios)
- Configuração dos blocos programáveis
- Compilação e envio do programa ao μDX

Tenha em mente o processo que você quer controlar com o μDX. Fazer um rascunho em papel é muito útil, tanto para analisar a solução idealizada como para escolher a disposição mais adequada dos blocos na área de programação. Depois vá "pegando e colocando" cada um dos blocos de que você precisa para fazer este desenho de programa na área de programação.

É altamente recomendável utilizar uma metodologia de programação por diagrama de estados. Veja exemplos no final deste manual (Controle de reservatório e Prensa).

As figuras a seguir mostram em mais detalhes como isto pode ser feito:

Por exemplo, escolheu-se o bloco de relógio para ser colocado na área de programação: aponte o mouse sobre a figura do relógio que está no Menu de Programação.

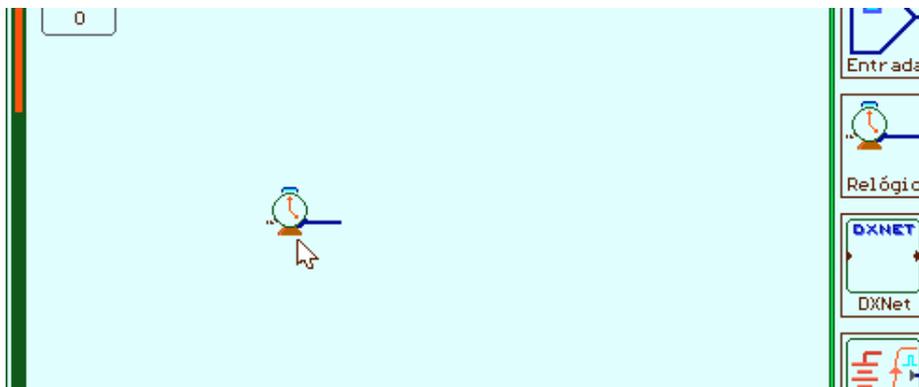


Pressione por um instante a tecla esquerda do mouse. A seta que representa o mouse desaparecerá e uma cópia da figura do relógio "descolará", sendo arrastada para qualquer ponto da tela do computador conforme os movimentos do mouse.



Mova a figura para o ponto desejado da área de programação. Escolha o lugar em função da disposição que você planeja dar às outras partes do desenho. Lembre-se que a área de programação desloca-se verticalmente (ela é maior, verticalmente, do que os limites visíveis na tela do computador) mas não horizontalmente, já que a largura desta área é fixa.

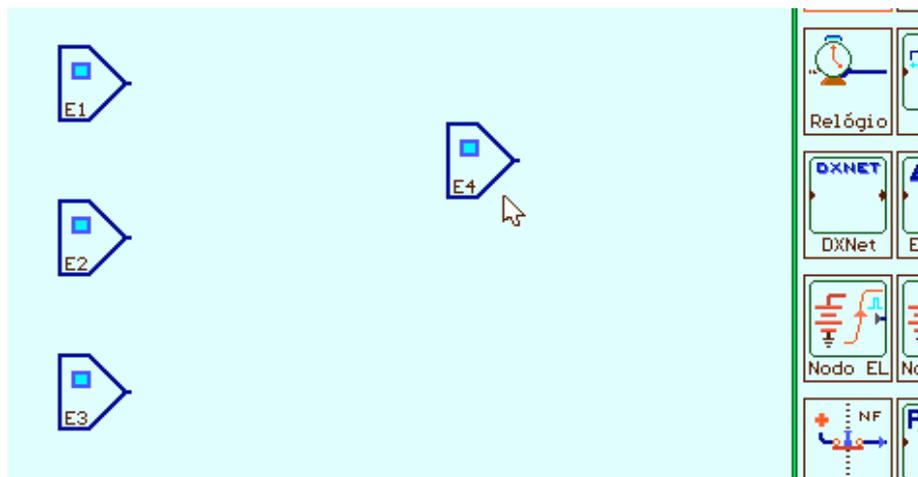
Depois de escolhido o lugar pressione por um instante a tecla esquerda do mouse, novamente, para que a figura seja fixada no local.



A seta do mouse voltará a aparecer. Se você precisar mudar a posição onde a figura foi fixada basta utilizar a tecla esquerda do mouse, apontando-o para o centro da figura, e movê-la como explicado anteriormente.

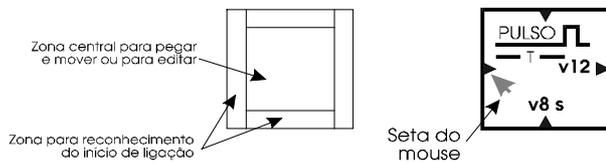
Note que as entradas e saídas são numeradas automaticamente quando você pega alguma delas do Menu de Programação. Assim se você quer utilizar a saída S3 é preciso pegar o bloco de saída duas vezes, colocando-os em um ponto qualquer da área de programa. Depois pegue mais uma vez o bloco de saída que possuirá o número correto. O mesmo ocorre com os blocos de entrada.

Alguns blocos possuem um certo limite de cópias que podem ser "retiradas" do Menu de Programação. Quando este limite é esgotado a figura correspondente, no Menu de Programação, fica cercada por um retângulo amarelo e com a palavra VAZIO escrita embaixo.

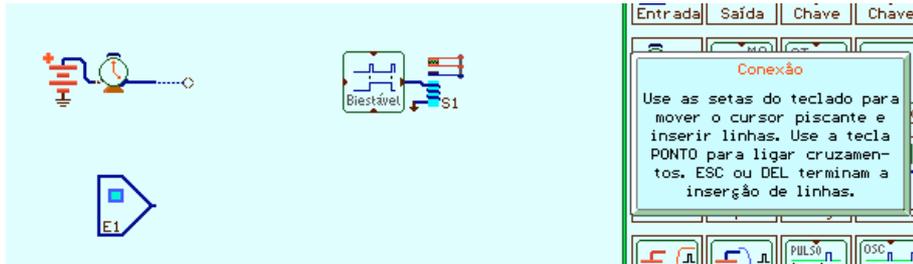


Depois de colocadas as figuras necessárias deve-se interligar cada nodo (entrada ou saída de bloco) de acordo com o esquema elétrico que o programa representará. Para fazer a interligação você deve apontar o mouse para a lateral do bloco onde quer iniciar a ligação e pressionar a tecla esquerda do mouse.

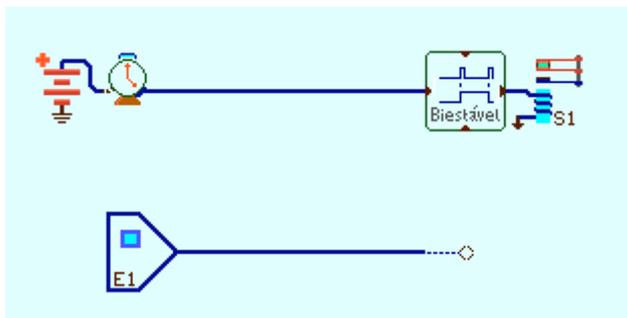
Note-se que é preciso apontar o lado de dentro do bloco, estando, portanto, a ponta da seta do mouse junto a lateral e na parte interna do bloco. Isto devido à técnica empregada no PG para reconhecimento da operação de inserção de linhas, onde cada figura na área de programa é dividida em setores de "sensibilidade" (veja o desenho).



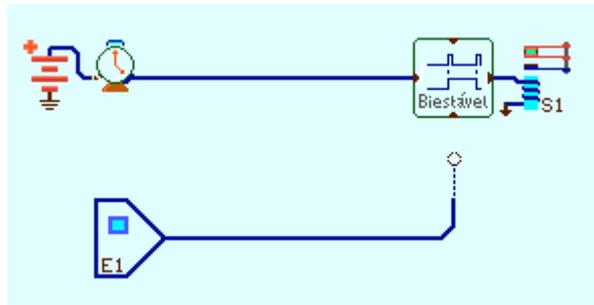
Quando iniciada a interconexão uma janela explicativa aparece sobre o Menu de Programação e uma linha curta (que pisca rapidamente mostrando um pequeno losângulo na extremidade) surge para mostrar a posição em que será colocado um segmento de linha.



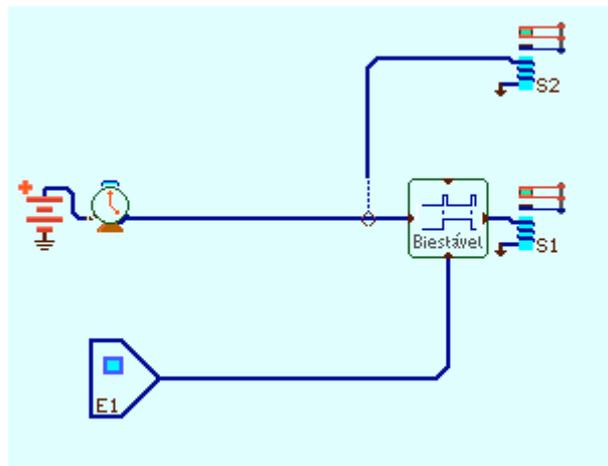
As linhas são colocadas automaticamente conforme se pressiona uma das teclas de setas no teclado do computador. Se com a próxima inserção de linha esta tocar em um dos nodos de uma figura o PG encerrará a inserção de linhas, completando a conexão.



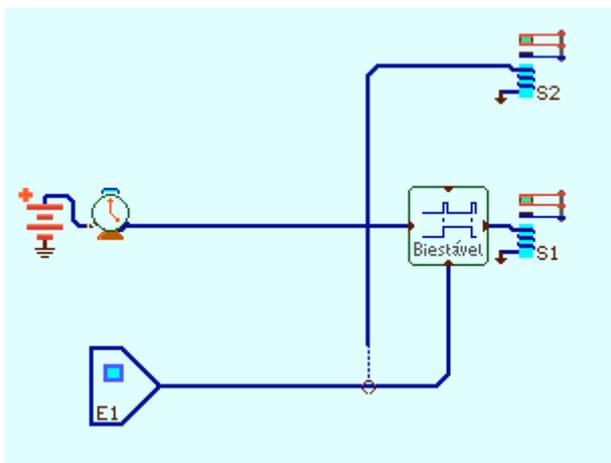
O PG coloca um segmento de linha na direção apontada pela seta pressionada. Se a direção mudar (horizontal para vertical ou vice-versa) o PG colocará um segmento de linha curva.



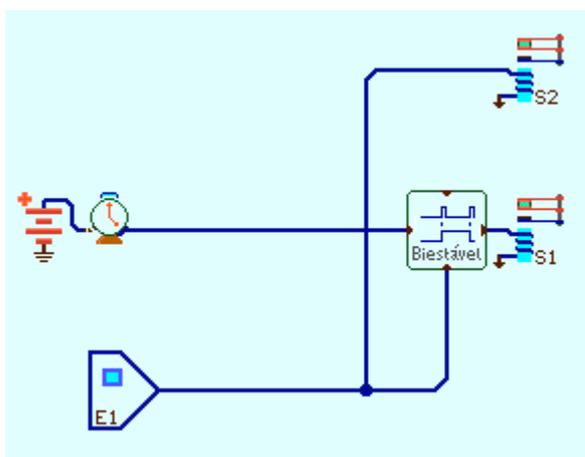
Quando a pequena linha piscante aponta para outra linha perpendicularmente pode-se tanto conectá-las, fazendo um cruzamento unido, ou passar por cima sem conexão, fazendo um cruzamento livre.



Para escolher basta seguir a direção da linha pressionando mais uma vez a tecla de seta no mesmo sentido.



Ou então pressionar a tecla . (ponto) no teclado do computador para indicar ao PG que se quer uma conexão:

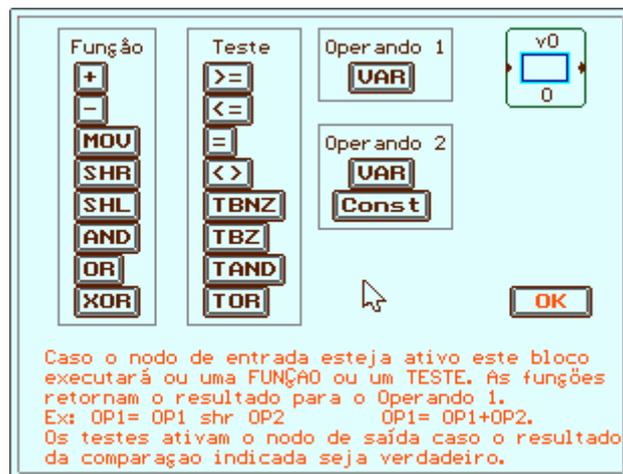


No programa os blocos (instruções) são como dispositivos elétricos bem determinados (chaves comutadoras, relés, etc.) e os fios simbolizam a ligação elétrica entre eles. Assim um desenho (programa) onde apareçam três blocos: Uma entrada, um temporizador tipo Atraso e uma saída, interligados (a Entrada no Temporizador e o Temporizador na Saída) fará com que o μDX ligue um temporizador tipo Atraso quando a entrada estiver ativa e ligue a saída depois que o tempo determinado ao temporizador tenha transcorrido.

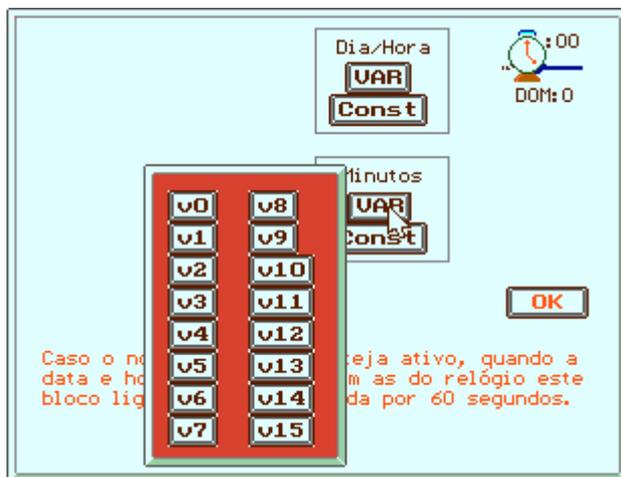
Se quiser apagar algum bloco basta apontá-lo com o mouse e pressionar a tecla DEL no teclado do computador ou a tecla direita do mouse.

Antes de enviar o programa ao μDX é preciso configurar os blocos programáveis, se tiver algum deles no programa. O temporizador tipo Atraso, por exemplo, precisa que seja especificado o tempo de atraso para que a saída seja ligada.

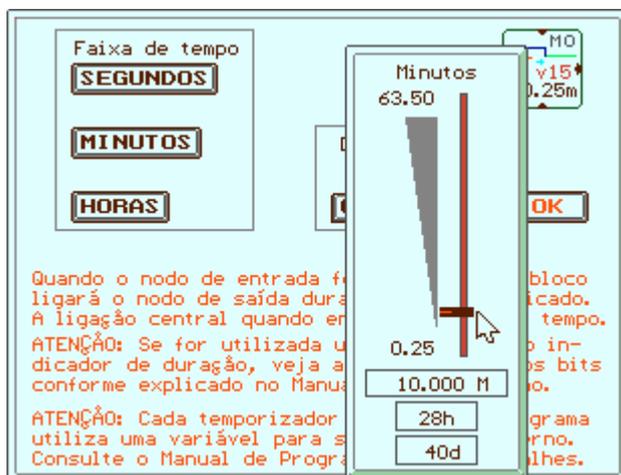
Para editar a configuração deste tipo de bloco de instrução basta apontar o mouse para o centro do bloco (depois de ele ter sido colocado na área de programa) e pressionar a tecla central do mouse ou a tecla E do teclado do computador. Aparecerá uma janela mostrando as diversas opções disponíveis para a programação do bloco apontado, como o mostrado a seguir, para o bloco de função:



Ainda dentro destas janelas é possível surgirem outras para que se especifique algum parâmetro qualquer, como a variável escolhida...

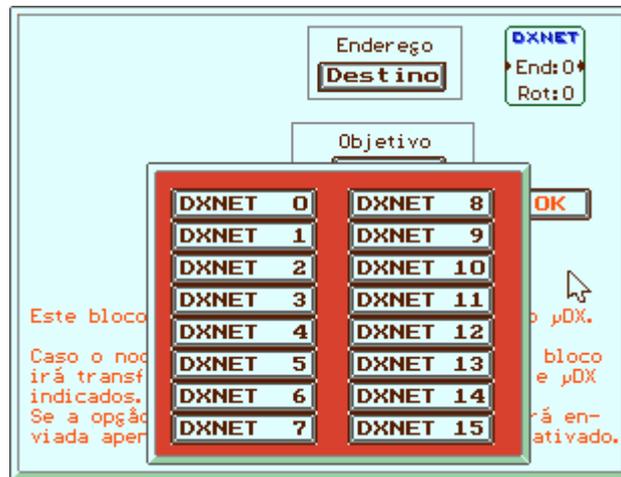


Ou o valor para ser atribuído como constante. Neste caso deve-se utilizar o mouse, apontado para a barra horizontal do dial, para movê-lo escolhendo o valor adequado. Também as setas no teclado do computador podem ser utilizadas.



Em alguns casos, como no bloco DXNET, o parâmetro escolhido para se alterar faz surgir uma pequena janela para entrada manual de algum valor.

Note que tanto as variáveis quanto as constantes do µDX são de 8 bits, ou seja, assumem valores entre 0 e 255.



Observação: A partir da versão 2.4 do programa PG, a seleção de endereço DXNET é feita via 16 teclas selecionáveis pelo mouse e não mais digitando-se o valor (vide exemplo acima).

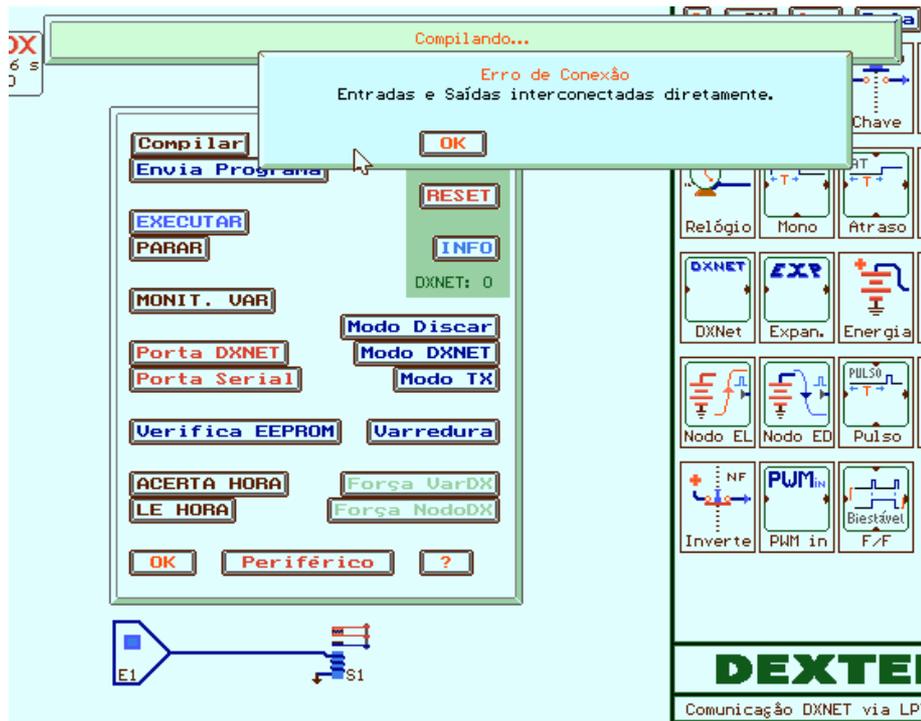
Alguns blocos não são configuráveis, como as entradas e saídas, porque seu funcionamento já está explícito na própria representação gráfica.

Depois é só enviar o programa ao μDX, utilizando para isso o Menu da tecla μDX disponível na tela principal (ver Teclas de Operação).

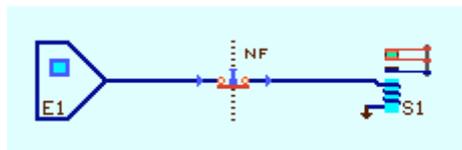
ATENÇÃO: Os nodos de saída dos blocos de instrução podem ser ligados entre si, neste caso o estado do nodo desta ligação será ligado caso qualquer uma das saídas estiver ligada (operação tipo "wired-or").

ATENÇÃO: Os blocos de entradas e de saídas não podem ser interconectados diretamente. É necessário utilizar algum bloco de instrução que sirva como intermediário (uma chave, por exemplo).

Se você tentar compilar um programa que tenha alguma entrada ligada diretamente a uma saída o PG informará um erro. Isto ocorre porque as entradas e saídas são tratadas como nodos e, sendo todos os nodos isolados entre si, não podem assumir este tipo de conexão. Os erros de compilação são mostrados através de uma janela, como aparece na figura. No caso, Erro de Conexão - Entradas e Saídas interconectadas diretamente.



Para conectar-se uma entrada à uma saída utiliza-se como artifício um bloco intermediário, que pode não ter nenhuma função própria.



Comandos Auxiliares para Edição de Programas

O PG dispõe de dois comandos que auxiliam a edição e a criação de programas:

Tecla F3: Insere linha em branco.

Tecla F4: Retira linha.

Tecla F3

Quando a tecla F3 do teclado do computador é pressionada o PG abre, na área de programação, um espaço horizontal equivalente a uma linha de blocos de instruções. O local onde este espaço surge é onde o mouse estiver apontando.

O conteúdo da área de programação é "empurrado" para baixo a partir da posição do mouse.

Eventuais conexões existentes no local onde é aberto este espaço serão interrompidas, sendo necessário que o usuário refaça as conexões.

Tecla F4

Retira uma linha do desenho feito na área de programação.

Se a linha de desenho a ser retirada possuir conexões ou blocos de instrução esses serão apagados.

*Não existe forma de retornar uma linha que tenha sido apagada.
Neste caso será preciso redesenhar a parte retirada.*

Convertendo de "LADDER" para PDE

"LADDER" é a linguagem de programação para Controladores Programáveis mais comumente empregada por indústrias e empresas de automação.

Inteiramente baseada nos métodos de projetos elétricos para controles por relés e circuitos discretos, a "LADDER" carregou junto com esta compatibilidade de conceitos uma complexidade de estrutura, precisando de especialistas para seu bom aproveitamento.

A linguagem PDE, que lembra um circuito elétrico convencional - e, portanto, mais fácil de ser interpretada - pode ser empregada para reproduzir um programa feito em "LADDER".

A programação em "LADDER" atende a algumas convenções:

- Existe uma barra de energia (+V) à esquerda e uma barra de OV à direita do circuito desenhado.

- As conexões do circuito sempre obedecem a este sentido de fluxo de energia: para ligar um relé ele precisa ter uma extremidade conectada (diretamente ou através de outros circuitos) à barra de energia e a outra extremidade conectada na barra de OV.

- Um relé pode ter diversos contatos auxiliares.

- O retorno de algum sinal, como a saída de um temporizador para sua própria entrada (formando um oscilador), somente pode ser feita através de um relé (acionado pelo temporizador) e um contato auxiliar deste relé conectado na entrada do temporizador.

Em PDE estas convenções podem ser esquecidas ou replicadas.

As barras de energia e de OV não são necessárias, uma vez que a energia (simbolizada pela figura de pilhas ou resultante de alguma entrada ativada) pode ser aplicada em qualquer local do esquema desenhado. O nível de OV (zero volts) está implícito nas conexões em aberto ou no símbolo aparente no desenho dos relés de saída.

O sentido do fluxo de corrente, da esquerda para a direita, é importante apenas nas próprias instruções (como as chaves e outros blocos). As conexões feitas pelos fios desenhados podem dar voltas à vontade em qualquer região do desenho, como ocorre em um diagrama esquemático convencional.

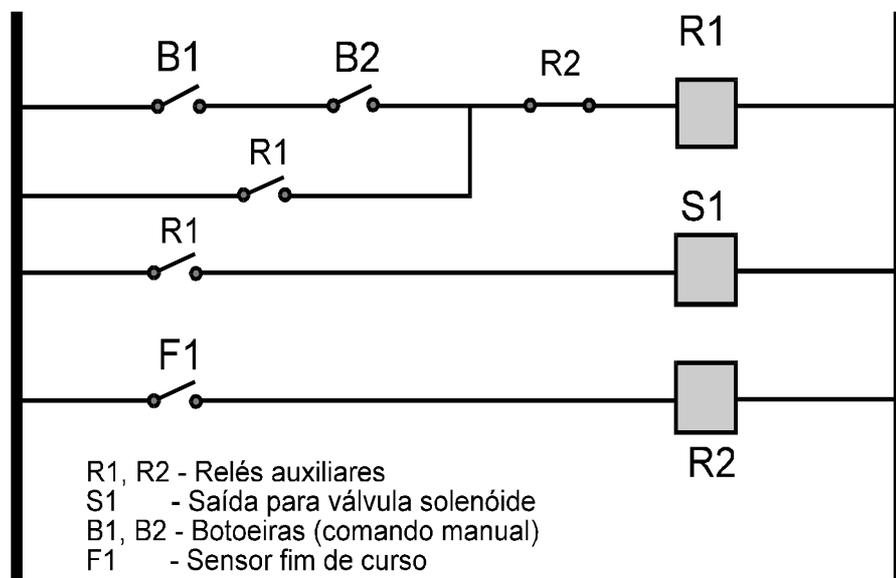
Os relés e contatos auxiliares empregados na linguagem "LADDER" podem ser representados pelas chaves (NA e NF) do μDX. O controle de uma chave funciona como a bobina do relé auxiliar e os contatos da chave como os contatos do relé.

Assim como em "LADDER" um relé pode ter mais de um contato auxiliar, também em PDE pode-se implementar esta idéia utilizando-se mais de uma chave e unindo-se os controles para serem atuadas todas de uma vez só.

É importante entender que no uso das chaves do μDX o controle é uma conexão (elétrica) atuada por alguma das entradas do μDX ou por uma saída de algum bloco de função. Assim, uma conexão em PDE pode representar um relé em "LADDER".

O desenho a seguir mostra um pequeno programa feito em "LADDER" que serve para controlar a lógica de operação de uma prensa (hipotética). As botoeiras B1 e B2 servem para iniciar a operação e precisam ser acionadas juntas (na prática elas devem ser montadas distantes de forma que o operador precise das duas mãos para pressionar as botoeiras). Uma vez iniciada a operação o relé auxiliar R1 é ligado (já que o contato auxiliar do relé R2 é normalmente fechado). O primeiro contato auxiliar de R1 irá manter o próprio R1 ligado mesmo que as botoeiras sejam liberadas (travamento). O segundo contato auxiliar de R1 energiza o relé S1 que, na prática, deve atuar uma válvula solenóide. Esta válvula comanda o movimento da prensa.

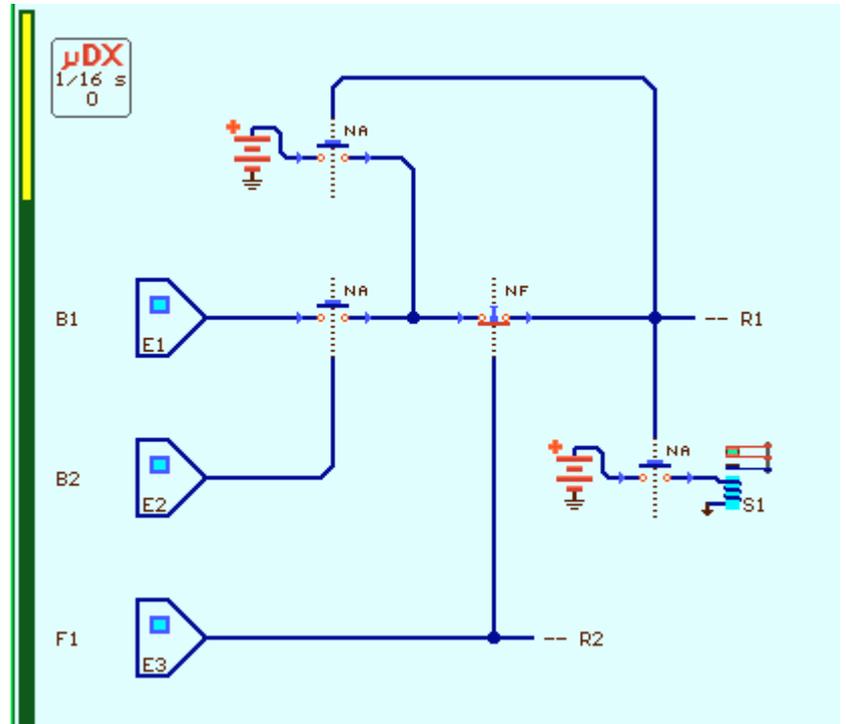
No fim do movimento da prensa um sensor tipo fim-de-curso é atuado ligando o relé auxiliar R2. Este relé, quando ligado, abre seu contato auxiliar desligando o relé R1, que desligará a saída S1, com a lógica toda voltando ao estado inicial.



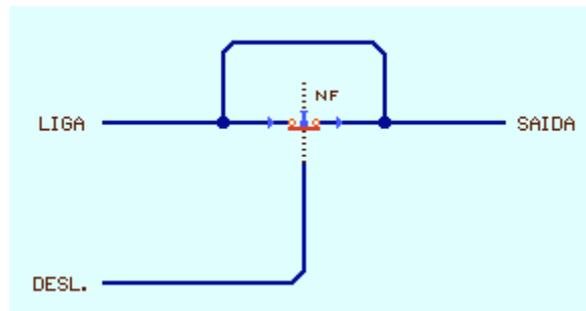
O próximo desenho mostra o equivalente em PDE ao programa de controle da prensa. As conexões que funcionam como os relés auxiliares R1 e R2 estão apontadas (note que este apontamento foi feito apenas para fins didáticos e não é necessário para o funcionamento do programa).

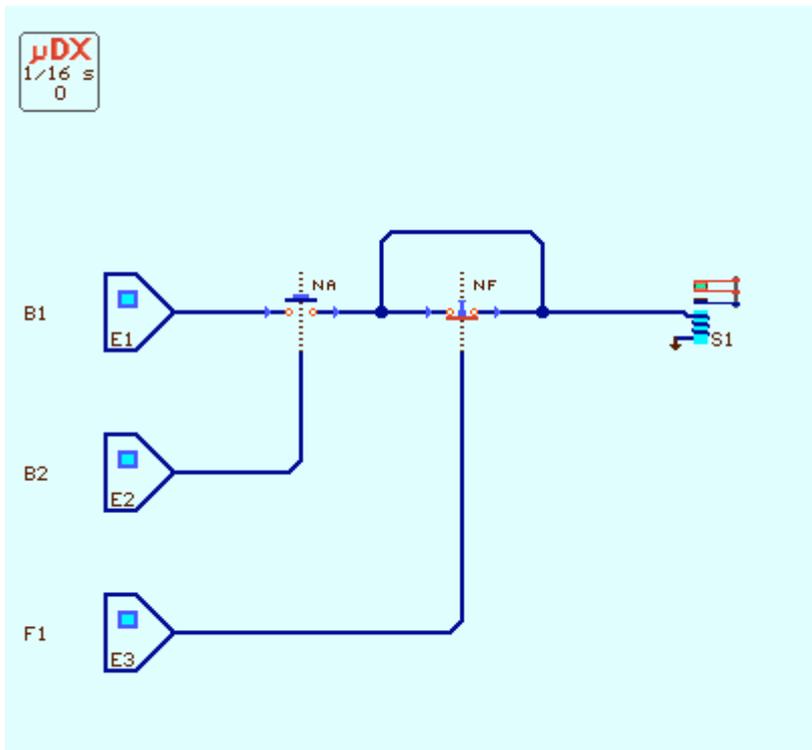
O símbolo de bateria foi utilizado nos pontos onde é preciso haver energia para a atuação do circuito. As entradas E1 e E2 (para as botoeiras) efetuam a lógica "AND" através da chave NA. A outra chave NA acima é utilizada para funcionar como o contato de travamento (retenção), suprimindo de energia a conexão antes da chave NF caso alguma das entradas fique inativa (e após "R1" ter sido ligado).

A entrada do sensor fim-de-curso conecta-se diretamente no controle da chave NF, uma vez que estando E3 ativa o relé "R2" deve desligar o relé "R1".



Pode-se simplificar ainda mais o esquema acima, utilizando uma particularidade da linguagem PDE. A retenção pode ser feita por realimentação direta do sinal de saída, sem a necessidade da chave NA superior. Neste caso, o esquema seria o da página seguinte. Note que utilizei o esquema de retenção mostrado abaixo. Ao energizar a linha LIGA ela se auto-sustenta pela realimentação da saída da chave NF. Ao energizar a linha DESL. abre-se a porta NF, interrompendo a auto-sustentação e desligando a linha de saída.





Portanto, o esquema final em PDE fica como mostrado acima. Pode-se perceber uma simplificação em relação à linguagem "LADDER", pois não há necessidade dos relés auxiliares.

Após compilar o programa e transmiti-lo ao μDX, mande-o executar este programa exemplo. A seguir, entre em monitoração (tecla de operação MON). Apontando com o mouse para a entrada E1 e pressionando a tecla esquerda do mouse é possível acionar esta entrada. Fazendo o mesmo com E2 a chave NA se fecha, energizando o relé de saída S1. Aponte para E1 e E2 e pressione novamente a tecla esquerda do mouse, desativando estas entradas. Apesar de desativadas, a saída continuará energizada devido a auto-sustentação. Entretanto, ao ativar a entrada E3 a chave NF se abre, interrompendo a auto-sustentação e desligando a saída S1. Faça isso apontando com o mouse para E3 e pressionando sua tecla esquerda. Para sair do modo de monitoração basta pressionar a tecla ESC do teclado do microcomputador.

Blocos de Instruções

O μDX reconhece e interpreta 16 blocos de instruções (52 no total se consideradas as várias opções):

TEMPO	Relógio	(4 modos)
	Mono-Estável	(2 modos)
	Pulso	(2 modos)
	Atraso	(2 modos)
	Oscilador	(2 modos)
FUNÇÃO	Aritmética/Lógica	(8 tipos / 2 modos)
	Comparação/Teste	(8 tipos / 2 modos)
	PWMin	
	Bi-Estável	
CHAVES	Normal Aberto	
	Normal Fechado	
	Inversor	
EXPANSÃO	Entrada/Saída 8 bits	
	Teclado/Display	
DXNET	Envia NODO	
	Envia VARIÁVEL	

Além destes o PG possui sete outros blocos que atuam na configuração do μDX ou representam símbolos operacionais e de programação:

CONFIGURAÇÃO	Nodo EL
	Nodo ED
	μDX

PROGRAMAÇÃO	Energia Rótulo
SÍMBOLOS	Entrada Saída

Os blocos de Configuração são enviados ao μDX como um único bloco que sempre aparece no começo de cada programa. Os blocos de Programação servem apenas para facilitar a elaboração de programas permitindo interconectar blocos distantes ou suprir de energia onde for preciso, não sendo enviados ao μDX.

Os símbolos de Entrada e de Saída representam exatamente os pontos de Entrada e os relés de Saída do μDX. Eles não são blocos de instrução porque não efetuam qualquer transformação ou operação, restringindo-se a energizar (ou não) algum nodo (quando Entrada) e acionar (ou não) um relé de Saída conforme o estado do nodo a ele conectado.

Foi convencionalizado que para qualquer bloco de instrução podem existir até três conexões independentes. Cada conexão, como já mencionado, recebe o nome de NODO. Assim, os três tipos de conexões dos blocos são chamados: Nodo de Entrada, Nodo de Controle e Nodo de Saída.

Alguns blocos não utilizam o Nodo de Controle. Naqueles em que é empregado ele serve para acionar o bloco (como o caso das CHAVES) ou para zerar ("reset") o bloco, como é o caso dos temporizadores e do "Flip-Flop" bi-estável.

Em qualquer dos casos, estes três nodos têm sempre um lado específico nos blocos: o Nodo de Entrada fica no lado esquerdo, o Nodo de Saída fica no lado direito e o Nodo de Controle (quando existir) fica no lado inferior e no lado superior. Aliás estes lados do Nodo de Controle estão internamente conectados, como se houvesse um fio ligando um extremo ao outro.

Funcionamento dos Blocos de Tempo

No μDX os blocos de tempo (Mono-Estável, Pulso, Atraso e Oscilador) funcionam da seguinte maneira:

A constante ou a variável especificadas para programar o tempo escolhido são reconhecidas apenas uma vez a cada temporização pelo programa que controla estes blocos. Assim, quando o nodo de entrada de um bloco de mono-estável é ligado o programa do μDX reconhece este estado e lê o valor da constante ou o valor atual da variável que indica o tempo programado para o bloco. O valor obtido é então memorizado numa variável auxiliar para ser decrementado a cada intervalo de tempo (conforme programado pelo usuário e pode ser de Horas, Minutos ou Segundos).

Quando o valor nesta variável auxiliar chegar a zero o programa do μDX vai executar a segunda parte da função do bloco. No caso do mono-estável isto significa desligar o nodo de saída e para os outros tipos seria o momento de ligar o nodo de saída.

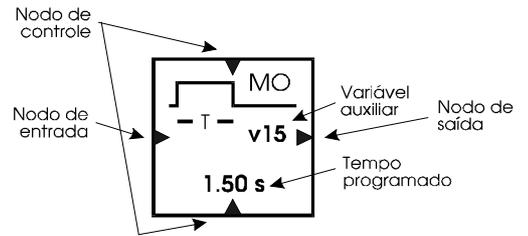


Figura de um bloco de instrução de tempo (mono-estável)

Depois o programa do μDX manterá o valor na variável auxiliar em zero até que o nodo de entrada seja desligado e novamente ligado ou que o nodo de controle seja acionado, causando um zeramento do bloco. Este zeramento significa atribuir o valor 255 à variável auxiliar, o que indica ao μDX que este bloco está zerado e pronto para nova temporização.

Como o μDX não dispõe de espaço de RAM interna para variáveis auxiliares, estas são obtidas pela utilização das variáveis de cálculo disponíveis.

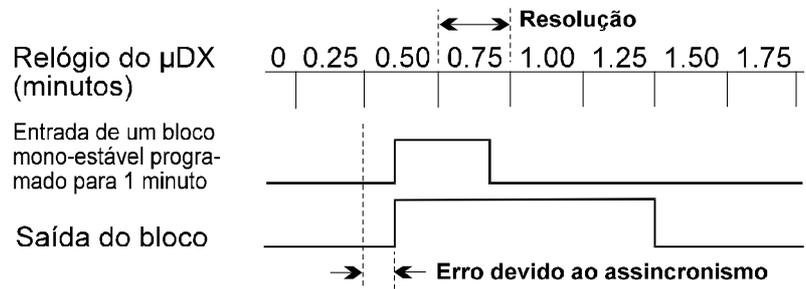
Para reduzir a confusão na elaboração de programas para cada bloco de tempo colocado na área de programação o consumo das variáveis do μDX é em ordem decrescente, isto é, da variável V15 até a variável V0 (ou, no caso da μDX+, de variável V63 até variável V0).

A variável escolhida aparece dentro da figura do bloco de tempo escrita em cor vermelha. Para cada bloco de tempo inserido na área de programação a variável escolhida pode mudar. Este "efeito" ocorre porque o μDX mesmo é que obtém qual variável ele vai empregar como auxiliar conforme a ordem em que aparecem os blocos de tempo ao longo do programa todo.

Note-se que qualquer modificação no valor de uma variável que se usa para programar o tempo de um destes blocos não será notada senão quando ele for ativado. Por outro lado, se o programa feito pelo usuário interferir no valor da variável utilizada como auxiliar, isto vai resultar em alteração do período de tempo de atividade do bloco (por exemplo empregando-se um bloco de função para mudar o valor da variável auxiliar quando ocorrer determinada situação). Esta última condição de funcionamento pode ser bem empregada, na programação avançada, para a elaboração de controles mais complexos e inteligentes.

ATENÇÃO: *Caso a duração de ciclo especificada no bloco μDX (do tipo Configuração) seja diferente de 1/16s as temporizações mudarão suas escalas de acordo. Assim para 1/32s um intervalo especificado para 2,5 segundos passará em 1,25 segundos e para 1/64s passará em apenas 0,625 segundos (No caso de μDX com firmware igual ou superior à versão 4.2 existe a opção de 1/256s. Neste caso o intervalo de 2,5 segundos teria duração de 0,15625 segundos, desde que o programa seja pequeno para não comprometer a velocidade de execução).*

ATENÇÃO: Para cada faixa de tempo escolhida existe uma resolução para a duração ajustada. Assim, por exemplo, a faixa de horas tem resolução de 4 minutos (0.066 hora), o que quer dizer que a duração programada é sempre um intervalo múltiplo de 4 minutos. Além disso, esta resolução torna o intervalo total programado dependente do momento em que o temporizador é disparado. Isto ocorre porque para cada faixa os sinais internos que fazem a contagem do tempo programado ocorrem em momentos fixos. Veja a figura:



Assim, para temporizadores seleccionados para a faixa de segundos existe uma incerteza de 0,06s. Já para temporizadores seleccionados para a faixa de minutos esta incerteza é de 15s. Por fim, para temporizadores seleccionados para a faixa de horas a incerteza é de 4 minutos. Estes valores são para configuração de 1/16s. Caso seja seleccionada uma configuração de 1/32s todas as incertezas acima são divididas por 2. Para as outras faixas de operação do μDX (1/64 ou 1/256s) vale a regra de divisão correspondente. Entretanto, em velocidades muito altas de operação, programas grandes e temporizações curtas, podem ocorrer atrasos adicionais devido ao processamento do programa.



TEMPO - Relógio

Esta instrução utiliza o relógio interno do μDX para comparar com o dia, as horas e os minutos programados e, quando coincidirem, acionar o nodo de saída durante 15 segundos. Mesmo que o relógio interno tenha correspondência, se o nodo de entrada não estiver ativo o nodo de saída permanecerá desligado.

O horário de acionamento pode ser programado utilizando-se variáveis ou constantes (combinadas livremente). Assim pode-se escolher uma variável para os minutos e deixar o dia e a hora indicados por constante.

Na figura deste bloco o dia e a hora aparecem na parte inferior, abaixo do desenho do relógio. Os minutos aparecem na parte superior, a direita do relógio. A posição dos ponteiros do relógio nada tem a ver com os valores especificados, servindo apenas para composição artística do desenho.

Quando for escolhida uma constante para o dia e hora é possível assinalar uma condição para que o relógio seja atuado em qualquer dia da semana e num determinado horário. Para isso deve-se escolher a opção de TODOS dentre os dias da semana (abaixo do relógio aparecerá DIA).

Quando utilizada uma variável para minutos obtém-se uma resolução de 1/4 de minuto (15 segundos) para o relógio devido à resolução própria do relógio do μDX, onde um byte conta 240 quartos de minuto. Neste caso o nodo de saída deste bloco ficará ativo apenas por 15 segundos. Quando utilizada uma variável para o dia e hora deve-se também adequar o valor dela com o formato do byte de dia e hora do relógio do μDX:

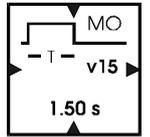
```
7 6 5 4 3 2 1 0
d d d h h h h h
```

Onde: d - indica o dia (000=domingo, 001=segunda,... e 111 para todos os dias)

h - indica as horas (0 a 23 em decimal ou 00000 a 10111 em binário)

Quantidade disponível : μDX = 127, μDX+ = 256.

TEMPO - MonoEstável



Esta instrução é um tipo de temporizador que, depois de ativado, mantém o nodo de saída ligado durante o tempo programado.

Para ser ativado é necessário que o nodo de entrada passe do estado de desligado para o estado de ligado. Durante o tempo programado o mono-estável (já ativado) não reconhecerá qualquer mudança no estado do nodo de entrada. Somente o nodo de controle têm influência sobre o mono-estável neste período. Se ligado ele fará com que o mono-estável seja zerado e que o nodo de saída seja desligado. Portanto, o monoestável é "não-retrigável".

Depois de passado o tempo programado o nodo de saída é desligado e o programa do μDX volta a verificar o nodo de entrada. Note-se que o mono-estável somente será disparado quando o nodo de entrada passa de desligado para ligado. Esta condição é chamada de sensibilidade de borda de subida (relativa à representação gráfica do sinal elétrico que sugere um degrau). Note que o monoestável é "não-retrigável", como já frisado.

O tempo programado pode ser especificado por uma constante ou por uma variável. Existem três faixas: Horas, Minutos e Segundos. A resolução do intervalo de tempo, tanto para constante como para variáveis é:

Horas - 4 minutos (0.066 hora)

Minutos - 15 segundos (0.25 minuto)

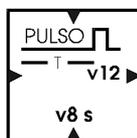
Segundos - 0.062 segundo

Veja o bloco Configuração - μDX que permite alterar esta resolução conforme a duração do ciclo programada.

Cada mono-estável inserido no programa desenhado ocupará também uma variável do μDX como variável auxiliar. O número da variável ocupada aparecerá dentro da figura do mono-estável. Veja no anexo B (Exemplos de Aplicações) deste manual o programa de minuteria para prédios (arquivo MINUT.UDX) para transformar o bloco de monoestável em "retrigável".

Quantidade disponível : μDX = 16, μDX+ = 64.

TEMPO - Pulso



Esta instrução produz um pulso no nodo de saída após o nodo de entrada haver sido acionado e o tempo programado tiver transcorrido. Este pulso traduzido como ligar o nodo de saída por um período de duração de um único ciclo do μDX.

A ativação desta instrução ocorre apenas quando o nodo de entrada passa do estado de desligado para o ligado. Após ativado, durante o tempo programado, o estado do nodo de entrada não é mais testado.

Caso o nodo de controle seja atuado durante o período de tempo a instrução de Pulso zera a variável auxiliar e interrompe a temporização, sem ligar o nodo de saída.

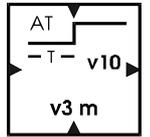
Para especificar o tempo pode-se utilizar uma constante ou uma variável. As faixas de duração disponíveis são: Minutos e Segundos. A resolução do intervalo de tempo, tanto para constante como para variáveis é:

Minutos - 15 segundos (0.25 minuto)
 Segundos - 0.062 segundo

Veja o bloco Configuração - μDX que permite alterar esta resolução conforme a duração do ciclo programada.

Cada instrução de Pulso inserida no programa desenhado ocupará uma variável do μDX como auxiliar.

Quantidade disponível : μDX = 16, μDX+ = 64.



TEMPO - Atraso

Esta instrução liga o nodo de saída depois que o nodo de entrada tiver sido ligado e que o tempo programado tiver transcorrido.

O nodo de saída permanecerá ligado enquanto o nodo de entrada continuar ligado. Se o nodo de entrada for desligado antes do tempo programado terminar a variável auxiliar será zerada e o programa do μ DX voltará a esperar que o nodo de entrada volte a ser ligado.

O tempo programado pode ser especificado por uma constante ou por uma variável. Existem três faixas: Horas, Minutos e Segundos. A resolução do intervalo de tempo, tanto para constante como para variáveis é:

Horas - 4 minutos (0.066 hora)

Minutos - 15 segundos (0.25 minuto)

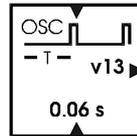
Segundos - 0.062 segundo

Veja o bloco Configuração - μ DX que permite alterar esta resolução conforme a duração do ciclo programada.

Cada instrução de Atraso inserida no programa desenhado ocupará também uma variável do μ DX como variável auxiliar. O número da variável ocupada aparecerá dentro da figura da instrução.

Quantidade disponível : μ DX = 16, μ DX+ = 64.

TEMPO - Oscilador



A instrução de oscilador produz pulsos em sua saída continuamente, com um intervalo que pode ser programado. Este bloco não possui nodo de entrada, uma vez que sua ativação é independente de condição. O nodo de controle, se ligado, faz parar o oscilador e mantém a saída desligada.

A duração do pulso equivale a um ciclo do μDX.

Para especificar o tempo pode-se utilizar uma constante ou uma variável. As faixas de duração disponíveis são: Minutos e Segundos. A resolução do intervalo de tempo, tanto para constante como para variáveis é:

Minutos - 15 segundos (0.25 minuto)

Segundos - 0.062 segundo

Veja o bloco Configuração - μDX que permite alterar esta resolução conforme a duração do ciclo programada.

Cada instrução de Oscilador inserida no programa desenhado ocupará uma variável do μDX como auxiliar.

Quantidade disponível : μDX = 16, μDX+ = 64.



FUNÇÃO - Aritmética/Lógica Comparação/Teste

Este bloco de instrução é o que oferece maior quantidade de opções de configuração, permitindo escolher uma entre dezesseis funções que incluem aritmética, lógica, comparação e testes.

Sua operação é executada enquanto o nodo de entrada estiver ligado, sendo que o nodo de saída será ligado sempre que a operação for completada com sucesso. Isto sempre ocorre nas operações de aritmética ou de lógica e somente quando o resultado é verdadeiro nas operações de comparação e de teste.

Todas as operações são tratadas utilizando-se dois operandos: Operando 1 e Operando 2. Nas funções aritméticas e nas lógicas o resultado do cálculo com os dois operandos é sempre retornado à variável do Operando 1. Nos testes e comparações o resultado não é retornado para o operando e sim para o nodo de saída. Os cálculos e comparações são sempre na seguinte ordem:

Operando 1 <função> Operando 2

O Operando 1 é sempre uma variável. Já o Operando 2 pode ser uma variável ou uma constante.

As operações disponíveis são as seguintes:

FUNÇÃO

OP1 = OP1 + OP2	- soma
OP1 = OP1 - OP2	- subtração
OP1 = OP1 mov OP2	- move OP2 para OP1
OP1 = OP1 shr OP2	- desloca OP1 à direita por OP2 bits
OP1 = OP1 shl OP2	- desloca OP1 à esquerda por OP2 bits
OP1 = OP1 and OP2	- operação lógica AND (E)
OP1 = OP1 or OP2	- operação lógica OR (OU)
OP1 = OP1 xor OP2	- operação lógica XOR (OU EXCLUSIVO)

TESTE

OP1 >= OP2	- liga a saída se OP1 maior ou igual a OP2
OP1 <= OP2	- liga a saída se OP1 menor ou igual a OP2
OP1 = OP2	- liga a saída se OP1 igual a OP2
OP1 <> OP2	- liga a saída se OP1 diferente de OP2
OP1 tbnz OP2	- liga a saída se o bit [OP2] de OP1 estiver em 1
OP1 tbz OP2	- liga a saída se o bit [OP2] de OP1 estiver em 0
OP1 tand OP2	- liga a saída se todos os bits em 1 de OP2 estiverem também em 1 no OP1
OP1 tor OP2	- liga a saída se pelo menos um dos bits em 1 de OP2 estiver também em 1 no OP1

O μDX não trabalha com valores negativos, apesar dos cálculos aritméticos serem feitos em complemento de 2. Assim o resultado do cálculo 10 - 15 não será -5 e sim 251 (256 - 5 = 251).

Note que todas as variáveis e constantes do μDX são de 8 bits, ou seja, assumem valores entre 0 e 255. Na edição do bloco de função sempre aparece o valor hexadecimal da constante utilizada (além do valor decimal) para facilitar as operações que envolvam manipulação de bits.

A partir da versão de firmware igual ou superior a 6.9, o Controlador μDX permite duas funções adicionais:

FUNÇÃO

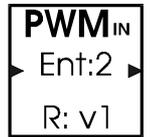
OP1 = OP1 SET OP2 - Faz o bit OP2 de OP1 = nodo de entrada.

OP1 = OP1 RST OP2 - Faz o bit OP2 de OP1 = nodo de entrada invertido.

Estas funções adicionais permitem modificar apenas um bit de determinada variável (especificada por OP1), conforme o estado do nodo de entrada do bloco. São especialmente úteis para acionar saídas da Expansão de Entradas/Saídas (lembre-se que a Expansão utiliza variáveis do μDX para acessar suas entradas e saídas).

Quantidade disponível : μDX = 127, μDX⁺ = 256.

FUNÇÃO - PWM_{IN}



Este bloco de instrução permite que o μDX converta o período de um pulso repetitivo em uma das entradas em um valor de 8 bits para ser guardado em uma variável. Em última análise, este bloco permite fazer a leitura de sinais analógicos desde que se utilize, por exemplo, um circuito oscilador controlado por tensão conectado em uma das entradas.

Pode-se escolher uma de três entradas (E1, E2 ou E3) e qualquer das dezesseis variáveis para armazenar o resultado.

O bloco será ativado enquanto o nodo de entrada estiver ligado e o nodo de saída será ligado após a conversão.

Se a largura do sinal da entrada em 1 ou em 0 exceder o limite de leitura o nodo de saída não será ligado.

Os limites são:

Nível em 1 máximo: 1.215,23 μs

Nível em 0 máximo: 1.460 μs (mínimo de 10 μs)

A medida de largura para a conversão é feita sobre o nível em 1 do sinal de entrada e está compreendida na seguinte faixa:

4,77 μs = 0

1215,23 μs = 254

A resolução da medida é de +/- 1, ou seja, um sinal em nível 1 com 14,3 μs de largura e que resultaria em uma medida correta de 3 pode ser medido como 2 ou como 4. Durações do nível 1 menores que 4,77μs poderão não ser detectadas, o que pode acabar causando ultrapassagem do limite do nível 0. Em μDX versão 4.2 em diante, caso a conversão PWM não ocorra corretamente, a variável associada ao PWM assume valor 255.

No caso de μDX versão 6.6, este bloco pode ser usado para leitura de código de barras (entrada E1 conectada a Leitor de Código de Barras).

Entrada de Contagem Rápida:

No caso de controlador μ DX⁺ o bloco PWMin pode ser usado para atribuir a entrada E2 um contador de alta velocidade, capaz de discernir até mais de 3000 Hz (3000 contagens por segundo). Note que a entrada E2 continua a se comportar como uma entrada normal, mas além disso a cada energização de E2 as variáveis de contagem rápida serão incrementadas.

Caso o programa seja para μ DX⁺, ao abrir o bloco PWMin surgirão duas opções selecionáveis:

Contador E2

Zerar Cont. E2

Caso seja selecionada apenas a primeira opção, este bloco irá incrementar a variável especificada e a subsequente (de forma a formar um contador de 16 bits) conforme os pulsos na entrada E2.

Já se forem selecionadas as duas opções este bloco irá zerar o contador rápido sempre que for energizado. Com isso, é fácil implementar, por exemplo, um freqüencímetro.

Quantidade disponível : μ DX = 127, μ DX⁺ = 256.

FUNÇÃO - Bi-Estável



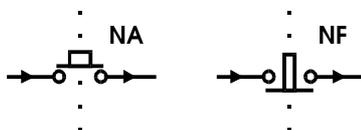
Este bloco reproduz a função de um multi vibrador biestável ("Flip/Flop") que troca o estado do nodo de saída cada vez que o nodo de entrada passa do estado de desligado para ligado.

O nodo de controle serve para forçar o bloco ao estado de repouso, isto é, com o nodo de saída desligado.

Sua melhor aplicação ocorre quando se precisa memorizar um estado ("flag") para teste posterior ou quando se quer que a saída do oscilador seja um sinal simétrico (onda quadrada), bastando para isso ligar a saída do oscilador na entrada do bi-estável.

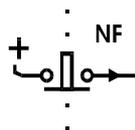
Quantidade disponível: 4.

CHAVES - Normal



Aberto

Normal Fechado
Inversor



As chaves são blocos cuja função está descrita pela própria representação gráfica: como um interruptor de corrente o botão que aparece no centro tem dois contatos por onde "passará" a corrente em determinadas condições.

A chave Normal Aberto (NA) precisa que o nodo de controle seja ligado para que ela permita passar o estado do nodo de entrada para o nodo de saída.

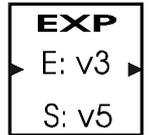
A chave Normal Fechado (NF) sempre passa o estado do nodo de entrada para o nodo de saída até que o nodo de controle seja ligado, quando interrompe esta transferência.

A chave Inversor é uma simplificação de uma chave NF cujo nodo de entrada estivesse ligado diretamente à fonte (positivo ou estado ligado). Assim atuando-se o nodo de controle o nodo de saída fica desligado e, caso contrário, o nodo de saída fica ligado.

É possível editar-se as chaves (pressionando a tecla [E] no teclado do microcomputador). A edição permite atribuir um número de nodo ao nodo de controle da chave. Assim, não há necessidade de usar o bloco Rótulo para conectar os nodos de controle das chaves. O número do nodo selecionado aparece sobre a representação pontilhada do controle da chave.

Quantidade disponível : μDX = 127, μDX+ = 256.

EXPANSÃO - Entrada/Saída de 8 bits



O bloco de expansão serve para que se possa ler e escrever valores de 8 bits em um circuito ligado no conector de expansão.

Quando o nodo de entrada é ligado o conteúdo da variável de saída é enviado à expansão ao mesmo tempo que dela é lido um valor de 8 bits para dentro da variável de entrada.

Para que isto funcione é preciso que o circuito montado na expansão obedeça a um esquema padrão, detalhado no ANEXO C - Diagramas Esquemáticos.

De qualquer forma a operação é sempre completada, mesmo que o circuito na expansão não esteja presente. Assim o nodo de saída sempre será ligado depois da operação completada. Note-se que este comportamento do nodo de saída é diferenciado quando se utiliza a expansão para a aplicação de display e teclado (veja EXPANSÃO - Teclado/Display).

Se o mesmo número de variável for utilizado para a entrada e para a saída na expansão, no momento em que o nodo de entrada é ligado o valor atual contido na variável é transferido para a saída da expansão e, ao término da operação, esta variável receberá os 8 bits da entrada da expansão.

Múltiplas Expansões:

No caso de controlador μDX+ é possível selecionar o uso de até 4 Expansões de Entradas/Saídas. Note que o número de Expansões é determinado pela seleção existente no bloco EXP (quando o programa aplicativo é para μDX+).

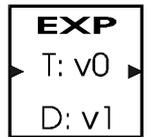
O bloco EXP irá selecionar sequencialmente variáveis para as entradas e saídas das Expansões. Assim, se selecionarmos como variável de entrada v0 e variável de saída v10, se selecionarmos as 4 Expansões o bloco EXP irá selecionar as variáveis v0, v1, v2 e v3 para as entradas e v10, v11, v12 e v13 para as saídas das Expansões.

As entradas e saídas das Expansões podem ser utilizadas normalmente no programa aplicativo do μ DX⁺. As entradas da primeira Expansão são E5 a E12 e as saídas S5 a S12; da segunda Expansão são 2E-E5 a 2E-E12 e as saídas 2S-S5 a 2S-S12; e assim em diante.

ATENÇÃO: *A fonte de alimentação que acompanha o controlador μ DX⁺ possui capacidade de corrente para suprir até 12 relés acionados simultaneamente (4 do μ DX⁺ e 8 de uma Expansão). No caso de uso de múltiplas Expansões é necessário verificar se o programa aplicativo não excede esta capacidade em algum momento ou substituir a fonte de alimentação por uma mais potente (para acionar todos os 36 relés do μ DX⁺ e das 4 Expansões a fonte deve suprir 12V @ 1,5A (18 W)).*

Quantidade disponível : μ DX = 127, μ DX⁺ = 256.

EXPANSÃO - Teclado/Display



Cada bloco de expansão pode ser configurado para funcionar como Entrada e Saída de 8 bits ou como interface para Teclado e Display. Quando esta última opção é selecionada a variável de entrada servirá para a entrada de um valor de 8 bits através de um teclado de 12 teclas e a variável de saída terá seu valor informado em um display de 3 dígitos (mostrando 000 até 255).

Para que esta operação funcione é necessário ter um circuito apropriado conectado na expansão (ver ANEXO C - Diagramas Esquemáticos - Expansão tipo Teclado / Display).

Enquanto o nodo de entrada estiver ligado o valor atual na variável de saída para o display é mostrado no visor. Se o valor na variável de saída mudar também o display mudará para mostrar o valor atualizado.

Para entrar com o valor através do teclado basta pressionar a tecla *. Enquanto ela for mantida pressionada o display mostrará o número da variável para a qual o valor digitado será transferido. Note-se que neste momento o primeiro dígito (o mais significativo) ficará apagado ou piscando fracamente, para indicar que o valor mostrado refere-se ao número da variável.

Depois de soltar a tecla * os pontos decimais dos três dígitos ficarão ligados para indicar que o usuário pode digitar o valor desejado. Durante esta operação o conteúdo do display ficará mostrando o valor correntemente digitado. Inicialmente este valor é igual ao conteúdo que havia na variável destinada à entrada do teclado. Quando a primeira tecla numérica for pressionada o display será zerado e o valor da primeira tecla aparecerá no primeiro dígito (o mais significativo). Atenção para o valor digitado: não pode ser superior à 255. Valores maiores (como o primeiro dígito igual a 3) farão com que o programa no μDX (que converte BCD para binário utilizando 16 bits internamente) assuma para a variável de entrada um valor truncado em 8 bits.

Cada vez que uma tecla numérica for pressionada um novo dígito será lido, em seqüência, do primeiro ao terceiro e depois voltando ao primeiro, continuamente. Se, por exemplo, forem digitadas 4 teclas numéricas, a quarta será atribuída novamente ao primeiro dígito; a quinta ao segundo e assim sucessivamente.

Por fim, o valor que estiver aparecendo no display é precisamente o valor que será assumido para a variável de entrada quando a tecla # for pressionada. Esta tecla significa o encerramento da digitação. Depois que ela for pressionada o programa do μDX irá ligar o nodo de saída deste bloco de expansão.

Sempre que se tiver um bloco de expansão para teclado e display para que o nodo de saída seja ligado é preciso, no mínimo, pressionar uma vez a tecla * para iniciar a digitação e depois a tecla # para que o μDX assumo o valor corrente da variável de entrada.

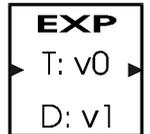
A duração do estado de ligado para o nodo de saída é de apenas um ciclo.

Note-se que qualquer alteração do valor contido na variável de entrada somente refletirá na digitação quando for pressionada a tecla *. Sempre que esta tecla for pressionada, mesmo após ter sido iniciada a digitação, o conteúdo da variável de entrada será novamente mostrado no display e o valor digitado anteriormente será esquecido.

Quantidade disponível : μDX = 127, μDX⁺ = 256.

Atenção: Os controladores μDX atualmente são fornecidos com opção para Expansão tipo Entradas / Display de 5 dígitos, em vez da Expansão tipo Teclado / Display. Isto porque já está disponível o periférico de Interface Homem / Máquina, que substitui com vantagens o circuito descrito acima. Entretanto, a Dexter pode fornecer controladores μDX compatíveis com a Expansão tipo Teclado / Display sob consulta.

EXPANSÃO - Entrada/Display de 5 dígitos



O controlador μDX é fornecido atualmente para aceitar um display de 5 dígitos + entrada de 8 bits, caso o bloco de expansão esteja programado como Teclado/Display (versão de firmware 6.1 para μDX). Neste caso, o μDX deixa de aceitar o circuito descrito nas páginas anteriores (Expansão tipo Teclado/Display), e aceita somente Expansão de Entradas/Saídas (Expansão Tipo 1) ou Expansão tipo Entradas/Display de 5 dígitos (vide esquema elétrico no final deste manual). Contacte a Dexter caso necessite que seu μDX seja compatível com o antigo tipo de expansão (Expansão tipo Teclado/Display). Neste caso, as entradas se comportam como entradas convencionais (8 bits), e as saídas acionam um display de leds de 5 dígitos. A variável eleita para Teclado no bloco de EXP irá receber o valor das 8 entradas digitais. Já a variável eleita para Display no bloco de EXP, será concatenada a variável imediatamente superior, de forma a obter-se um valor de 16 bits, que será apresentado no display numérico de 5 dígitos. Assim, se colocarmos a variável v1 para Display, (v1 + 256 * v2) será o valor que aparecerá no display, ou seja, um valor de 16 bits. Obviamente, o máximo valor será de 65535. Caso seja eleita a variável v15 para display, é utilizada v0 como byte mais significativo, ou seja, o display irá apresetar o valor (v15 + 256 * v0). .

Para que esta operação funcione é necessário ter um circuito apropriado conectado na expansão (ver ANEXO C - Diagramas Esquemáticos - Expansão tipo Entradas / Display de 5 dígitos).

Enquanto o nodo de entrada estiver ligado o valor atual na variável de saída para o display é mostrado no visor. Se o valor na variável de saída mudar também o display mudará para mostrar o valor atualizado. Além disso, o valor dos 8 bits de entrada são atribuídos a variável de entrada de Teclado.

Quantidade disponível : μDX = 127, μDX⁺ = 256.

DXNET - Envia Nodo
Envia Variável



Este é o bloco de instrução que permite a intercomunicação com vários μDX utilizando-se a Rede Local DXNET.

Com ele é possível fazer duas coisas: transferir para um nodo qualquer de outro μDX o estado atual do nodo de entrada deste bloco, ou transferir para uma variável qualquer de outro μDX o valor atual de uma variável local qualquer.

No caso de enviar variável a operação será realizada apenas quando o nodo de entrada passar do estado desligado para o ligado. Neste momento o μDX enviará o conteúdo da variável local para dentro da variável remota do μDX cujo endereço está especificado neste bloco de instrução (ou seja, apenas na borda de subida do sinal no nodo de entrada do bloco DXNET).

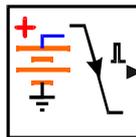
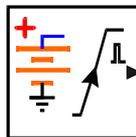
Se a instrução for para enviar nodo o estado atual do nodo de entrada será enviado a um rótulo (nodo) especificado no μDX de destino. Cada vez que o estado do nodo de entrada mudar, o μDX local vai transmitir ao μDX remoto o novo valor. O rótulo (nodo) no destino fica armazenado numa tabela especial onde existe uma posição (bit) para cada nodo do μDX. Esta tabela é sempre processada a cada ciclo e os estados nela contidos são reunidos com os estados de cada nodo do μDX destino, como se esta tabela estivesse representando as saídas de diversos blocos fantasmas.

Esta tabela é chamada, internamente, de NODO_DX (ver Menu da Tecla μDX).

Em ambos os casos, envio de variável ou de nodo, o nodo de saída do bloco será ligado apenas se a comunicação tiver sido completada com êxito. Enquanto isto não ocorrer o μDX continuará tentando transmitir a informação a cada ciclo até que seja completada ou que o nodo de entrada volte ao estado anterior (como se não tivesse sido acionado).

Quantidade disponível: 8.

CONFIGURAÇÃO - Nodo EL
 Nodo ED



Estes dois blocos produzem um único pulso de ligado no nodo de saída em uma determinada condição.

O Nodo EL produz o pulso no momento em que o suprimento de energia (fonte de alimentação) for ligado ao μDX. O Nodo ED, ao contrário, produz o pulso no momento em que a energia é desligada.

Naturalmente é preciso que o μDX esteja recebendo energia das pilhas para que continue funcionando e reconheça estas condições.

A principal aplicação é permitir que o programa desenhado tenha como saber quando faltou ou retornou a energia elétrica, já que as saídas (relés) não funcionam sem esta energia. Neste caso o programa pode assumir um outro estado operacional ou memorizar a ocorrência para processamento posterior.

A duração do pulso é de apenas um ciclo do μDX.

Note-se que sendo um bloco de configuração o Nodo EL e o Nodo ED não ocupam espaço de memória de programa do μDX. Portanto, um programa qualquer pode ter 127 blocos de instrução além dos blocos de configuração.

O bloco Nodo EL é muito utilizado para inicializar variáveis no programa do μDX. Neste caso, ele deve ser posicionado no início do programa (próximo ao bloco de configuração μDX). Assim, ele será executado antes do restante do programa (o μDX executa o programa escrito no PG da esquerda para direita e de cima para baixo).

Quantidade disponível: 1 de cada.

CONFIGURAÇÃO - μDX



Este bloco serve para indicar ao μDX que receber o programa qual o endereço na Rede Local DXNET que ele deve assumir, qual sua duração de ciclo e qual o tipo de CLP usado (μDX ou μDX+).

Além disso, este bloco não pode ser apagado do programa; nem pode ser movido para algum outro ponto.

Os valores "default" (de inicialização) são o endereço na rede igual a zero, duração de ciclo igual a 1/16 segundo, e CLP μDX.

Atenção: quando for escolhida outra duração de ciclo (1/32, 1/64, ou 1/256s). A programação de intervalo de tempo nos temporizadores (Mono, Pulso, Atraso e Oscilador) será dividida por 2, por 4, ou por 16, conforme a escolha. O valor final é apresentado na tela do computador.

Observe que o μDX demora cerca de 16 milissegundos para executar 128 blocos de instruções. Assim, do tempo restante devido a duração de ciclo escolhida, deixa-se livre a maior parte para alguma eventual comunicação via DXNET. Como cada comunicado pela DXNET, quando o endereço confere, consome em média 15 milissegundos, deve-se observar o limite de tempo disponível para eles em função da duração de ciclo escolhida.

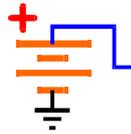
Se o tempo de execução de todas as instruções mais o tempo necessário aos comunicados via DXNET for superior à duração de ciclo poderá ocorrer que o controlador irá executar menos vezes o programa por segundo. Isso não traz nenhum problema ao relógio de tempo real nem aos temporizadores.

Por outro lado, utilizar uma duração de ciclo pequena permite empregar o μDX em aplicações que exijam mais velocidade de resposta.

Ao selecionar μDX+ (Plus) o número de blocos disponíveis sobe para 256, o número de nodos para 192, e o número de variáveis para 64.

Quantidade disponível: não se aplica.

PROGRAMAÇÃO - Energia



Este bloco serve para indicar quais nodos devem ser normalmente energizados (ligados). Esta indicação é interpretada pelo μDX como sendo todos os blocos de energia apenas um só e, internamente, ocupando o nodo número 63, que é mantido sempre em 1.

Assim, sempre que um bloco de instrução ou uma ligação que una um ou mais blocos de instrução precisar que fique continuamente no estado ligado, basta conectar a este bloco de energia.

Note-se que apesar da representação simbólica de pilhas (ou bateria) a energia pode ser suprida tanto pelas pilhas como pela fonte de alimentação, uma vez que importa apenas o estado ligado que o bloco indica.

Quantidade disponível: Sem limite.

PROGRAMAÇÃO - Rótulo

O bloco de rótulo serve para forçar que uma determinada ligação receba uma numeração de nodo específica.

Isto é útil devido ao compilador do PG que atribui uma numeração a cada nodo conforme o resultado da compilação, que pode mudar caso o programa seja alterado. Neste caso, se o nodo em questão pode receber um estímulo vindo de outro μDX (via DXNET), é preciso informar exatamente qual o número deste nodo ao bloco de instrução DXNET no μDX de origem. Identicamente quando se quer forçar o estado de ligado para este nodo utilizando-se o comando de Forçamento de Nodo_DX do PG (Menu μDX).

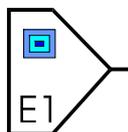
Se você não informar o número para o rótulo (nodo) o compilador do PG assumirá o valor 8.

Com o rótulo você pode interligar partes do programa sem precisar desenhar as ligações. Basta colocar o bloco de rótulo nos lugares que devem ser interconectados e atribuir a todos estes rótulos um número idêntico. Esta é outra grande utilidade deste bloco, pois permite elaborar programas com ligações complexas entre blocos.

Atenção para os números disponíveis para rótulo que vão de 0 (zero) até 61 (ou, no caso do μDX+, de 0 a 191). Além disso, os números 0, 1, 2 e 3 pertencem às quatro entradas do μDX e os números 4, 5, 6 e 7 pertencem às quatro saídas. Estes números podem ser empregados no rótulo porém seu uso indica que o estado das ligações associadas aos rótulos de mesmo número será entendido como o estado de uma entrada ou o acionamento de uma saída. Já o nodo 62 é terra (sempre em 0) e o nodo 63 é energia (sempre em 1). O nodo de terra é 62 e o de energia 63 também no caso de μDX+. Neste caso, os nodos genéricos são de 8 a 61, e de 64 a 191.

Quantidade disponível: Sem limite.

SÍMBOLO - Entrada



As quatro entradas do μDX podem ser utilizadas empregando-se um destes símbolos de Entrada para cada uma, assim como as oito entradas da Expansão de Entradas/Saídas.

Assim como indicado na tampa superior do μDX, cada entrada tem uma numeração própria: E1, E2, E3 e E4 (a Expansão possui as entradas E5 a E12).

Cada vez que se retira uma figura de entrada do Menu de Programação o próprio PG atribui a numeração à entrada conforme as que já estiverem em uso. Portanto, para escolher uma entrada específica fora da ordem de numeração que o PG entrega, basta pegar as figuras e colocar num local da área de programação que não seja utilizado. Depois pega-se novamente aquela que se quiser para mover até o lugar desejado.

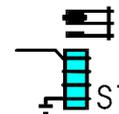
Internamente ao μDX, as entradas são associadas a nodos (0 a 3) e, a cada ciclo, o programa do μDX verifica o estado de cada entrada e liga ou não o nodo correspondente.

Além disso, as entradas E1, E2 e E3 podem servir para a leitura da largura de pulso quando se emprega o bloco de instrução PWMin (ver FUNÇÃO - PWMin).

ATENÇÃO: As entradas do μDX, fisicamente, não são isoladas galvanicamente (as entradas da Expansão são isoladas). Portanto não conecte os fios da energia elétrica domiciliar diretamente em qualquer das entradas do μDX. Utilize para isso o módulo de Opto-Acoplador ou um circuito externo com isolador ótico (veja ANEXO C - Diagramas Esquemáticos).

Quantidade disponível: 12.

SÍMBOLO - Saída



As quatro saídas do μDX - S1, S2, S3 e S4 - são feitas utilizando-se relés, cada um com um contato reversor, isto é, existe neles uma lâmina móvel que permite passar a corrente elétrica quando está encostando em um dos dois contatos elétricos montados de cada lado. Assim, quando o relé está desligado a lâmina fica encostando em um deles (Normal Fechado - NF) e afastada do outro (Normal Aberto - NA). Quando o relé é ligado a conexão se reverte, ficando o primeiro contato em aberto e o segundo fechado.

Como os dois contatos e a lâmina central estão disponíveis para conexões externas (através do conector lateral do μDX) ambos podem ser empregados para que se tenha corrente elétrica quando o relé está desligado (usando o NF) ou quando está ligado (usando o NA).

Como as entradas, as quatro saídas também são, internamente, consideradas como nodos (4 a 7). Logo, se o nodo correspondente a uma das saídas for ligado por algum bloco de instrução (ou mesmo forçado como ligado via DXNET) então o relé desta saída será energizado.

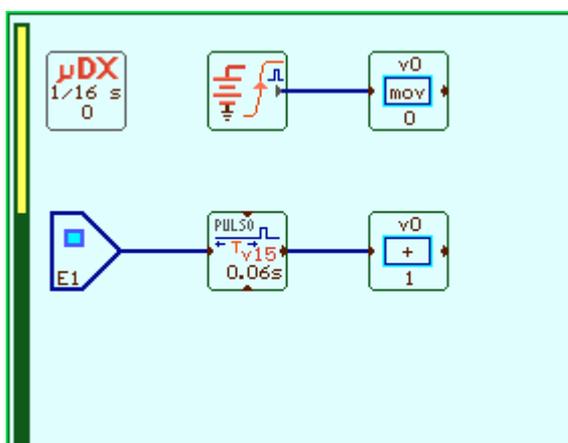
A Expansão de Entradas/Saídas possui as saídas S5 a S12.

ATENÇÃO: Como cada relé proporciona uma isolamento galvânica suficiente, você **pode** conectar em seus contatos os fios da energia elétrica domiciliar, observando apenas o limite de corrente máximo de **10 Amperes**. Tome sempre muito cuidado nas conexões devido à proximidade entre os contatos do conector.

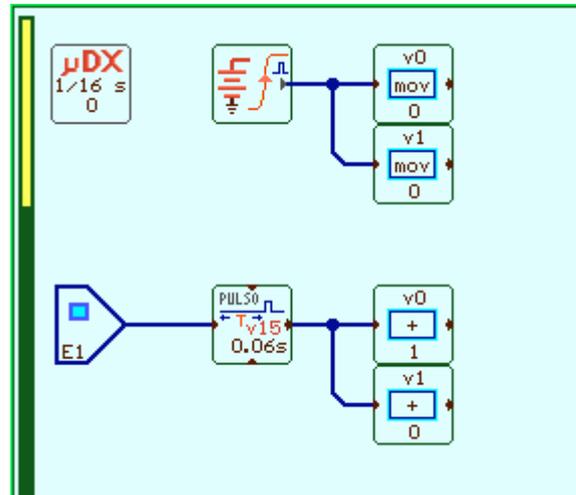
Quantidade disponível: 12.

Operações aritméticas com mais de 8 bits

O controlador μDX utiliza variáveis com 8 bits, sem sinal. Assim, o máximo valor que estas variáveis podem assumir é 255 (2^8-1). Entretanto, muitas vezes pode ser necessário manipular valores numéricos maiores. Para isso, a versão 6.0 do μDX ou superior prevê a propagação de estouro (carry ou borrow) no caso de operação de adição ou subtração. Neste caso, utiliza-se duas funções anteriormente inúteis, que eram somar ou subtrair 0 (zero) de uma variável. Quando é efetuada uma operação de soma ou subtração em que ocorre estouro, esta informação é armazenada pelo μDX e, no primeiro bloco com adição com zero ou subtração com zero este estouro é adicionado ou subtraído da variável. Por exemplo, se tivermos um bloco que incrementa v0 a cada energização da entrada E1 (para contar número de peças que passam por um sensor ligado a E1), o programa poderia ser como abaixo (note o bloco PULSO para garantir apenas um incremento para cada energização de E1, e o bloco NODO EL para inicializar a variável v0 com valor zero):



Este programa incrementa v0 a cada energização da entrada E1. Mas, após 256 energizações, a variável v0 volta ao valor zero (pois $255+1 = 256 \rightarrow$ estouro de variável). Então, pode-se usar uma segunda variável para continuar a contagem (usando a propagação de carry via bloco com adição com zero):



Note que, quando houver estouro em v0, o carry será considerado no bloco subseqüente (v1+0), incrementando v1. Assim, o valor de contagem será obtido pela fórmula:

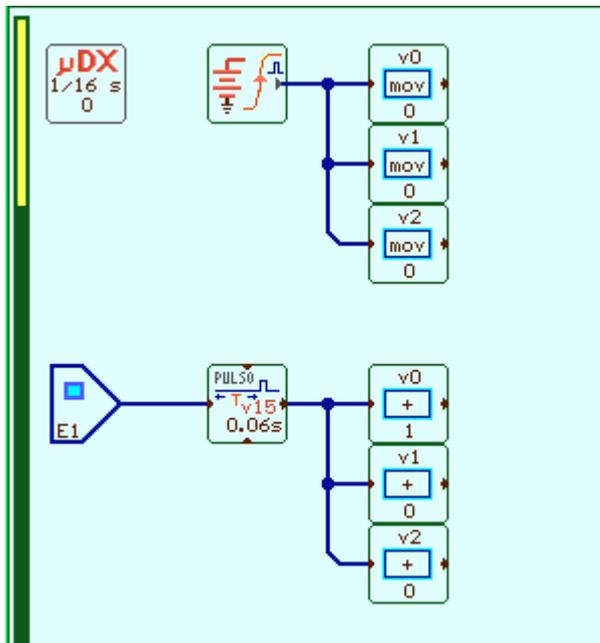
$$Contagem = 256 * v1 + v0$$

O valor máximo atingível é de 65535 contagens. Se ainda assim não for o suficiente basta adicionar mais variáveis. Neste caso, a contagem será:

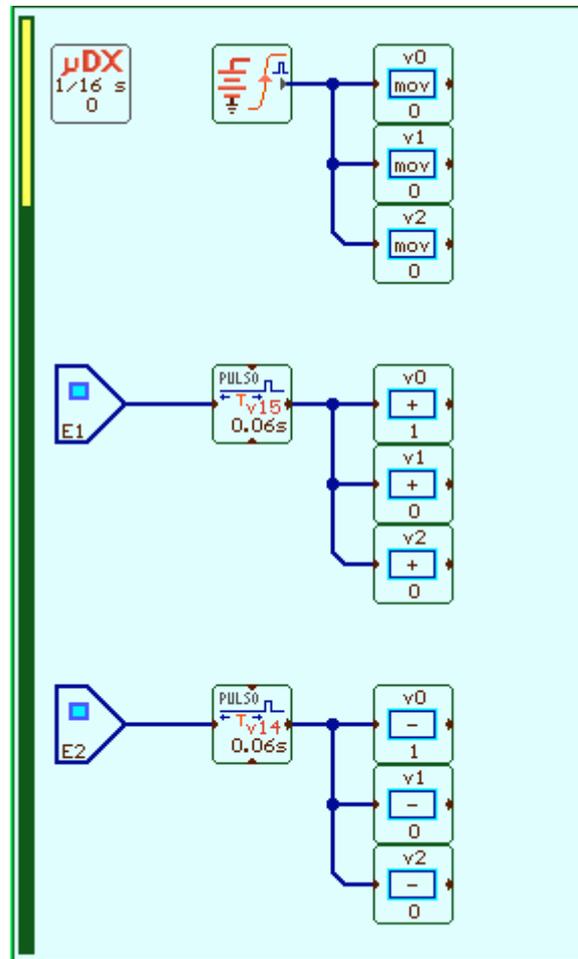
$$Contagem = 256 * 256 * v2 + 256 * v1 + v0$$

O limite, neste caso, é de 16777215 contagens!

Atenção: A propagação de estouro de variável (carry) só funciona para controladores μDX com versão de firmware igual ou superior a 6.0.



O mesmo raciocínio pode ser aplicado para a operação de subtração. Na página seguinte o programa CARRY.UDX exercita esta facilidade. A entrada E1 incrementa a contagem, enquanto a entrada E2 decrementa a mesma contagem. O simulador prevê este comportamento e, portanto, pode ser utilizado para obter um melhor entendimento do processo.



Conexão em rede - DXNET

Se você tiver mais de um μ DX eles poderão se comunicar através da Rede Local DXNET. Utilizando o bloco de instrução DXNET o programa elaborado para um μ DX pode alterar o conteúdo de uma variável de outro qualquer ou fazer com que seja ligada uma saída (ou mesmo forçar um nodo de entrada como se ela estivesse ligada).

Estes recursos permitem, entre outras coisas, que cada μ DX conectado à rede processe uma parte de um programa mais complexo, tornando possível controlar grandes ambientes, como fazer uma casa inteira inteligente, buscando conforto e economia, ou automatizando um processo industrial complexo.

Características da rede DXNET

A rede DXNET foi desenvolvida para ser de baixo custo de instalação, empregando-se um simples cabinho blindado (com um fio mais malha) para fazer a interconexão. Este cabinho deve ser pouco capacitivo.

Em cada μ DX existe um resistor de 1,8 Kohms ligado entre a linha de comunicação e o +5V. Por isso deve-se limitar em 15 o número de μ DX conectados à rede, não só devido ao endereçamento, limitado em 15, mas por que estes resistores ficariam conectados em paralelo, aumentando muito a corrente que cada μ DX precisaria drenar para comutar a linha de 1 para 0.

A distância, testada em laboratório, entre uma ponta e a outra da rede é de 100 metros. Este valor pode ser excedido porém sem garantia que o funcionamento seja constante. Ainda assim uma distância muito maior poderia causar danos ao circuito de comunicação, uma vez que a capacitância da linha aumentaria muito e conseqüentemente a corrente de drenagem seria maior.

No μ DX existem dois conectores tipo P2, fêmea, que estão interconectados em paralelo. É uma solução de engenharia para facilitar conexão com a rede. Deste modo pode-se interligar os μ DX utilizando um cabinho para conectar cada dois μ DX, sem precisar de emendas.

Operação da rede DXNET

A comunicação na rede DXNET segue os seguintes princípios:

- O modo de operação é Multi-Mestre, isto é, qualquer um dos dispositivos conectados na rede pode iniciar uma comunicação.
- Todos os μ DX estão sempre atentos à rede e "ouvem" a linha reconhecendo o cabeçalho da comunicação, onde existe um caracter de início e um segundo com o comando e o endereço de destino. Se o endereço de destino for diferente do endereço próprio do μ DX a rotina de recepção é interrompida e o μ DX continua o serviço que estava fazendo. A exceção é o endereço 0 (zero) que todos os μ DX reconhecem, sempre.
- Toda comunicação inicia com um bit de start (sinal em nível zero com duração de 2ms) e somente termina após o μ DX de destino haver respondido ao comando ou que tenha ocorrido um "Time-Out". Este último ocorre se nenhum μ DX na rede possui o mesmo endereço indicado no comunicado.
- Antes de qualquer dispositivo na rede iniciar uma comunicação ele deve verificar se a linha está livre por um período mínimo que varia conforme seu endereço na rede e a qual a tentativa de comunicação atual.
- O tempo de comunicação (incluindo-se o tempo gasto nas tentativas) consome, gradativamente, o tempo livre para comunicações por ciclo. Se este tempo livre for excedido a duração do ciclo será dilatada em função do excesso, podendo causar erros nas temporizações de frações de segundo.

- Ao programar um μDX conectado junto com outros na rede DXNET tenha certeza de que ele responde a um endereço diferente que os demais para que somente ele receba o programa. Com todo μDX sai de fábrica com endereço 0 (zero) deve-se programar cada um individualmente, com a conexão DXNET apenas entre o computador e o μDX. Depois disso, tendo ele recebido um endereço próprio, qualquer nova programação já poderá ser feita em rede.
- O endereço de cada μDX fica armazenado em memória não volátil, completamente independente das pilhas.

Programação usando a rede DXNET

O uso do bloco de instrução DXNET é muito simples.

Deve-se ter em mente que os blocos DXNET apenas transmitem. Assim para ler o conteúdo de uma variável em um certo μDX é preciso que o programa nele tenha um bloco DXNET que transmita o valor desta variável para o μDX local.

Apesar de parecer uma limitação, na maior parte dos processos controlados é preciso apenas transmitir informações quando ocorrem condições especiais. Para os outros casos basta elaborar os programas dos diversos μDX de acordo com um planejamento adequado.

É recomendável restringir o número de comunicados ao mínimo necessário, evitando principalmente os comunicados simultâneos. Programe o disparo da comunicação conforme o tempo de ciclo do processo controlado. Ele é determinado pelo tempo que o processo controlado precisa para modificar-se de forma significativa. Por exemplo o controle de temperatura de uma sala é bastante lento pois a mudança de um grau centígrado pode demorar desde dezenas de segundos até muitos minutos. Logo, informar a temperatura da sala a um outro μDX pode ser feito a cada minuto sem risco de perda de informação.

O nodo de entrada do bloco DXNET deve ser mantido em cada estado pelo máximo de tempo possível. No caso de enviar o valor de uma variável, quanto mais tempo o nodo de entrada ficar em 1 mais tempo haverá para retentativas caso haja algum erro de comunicação. No caso de enviar estado de nodo o tempo para retentativas vai ser conforme a duração que tiver o novo estado do nodo de entrada.

A DEXTER comercializa bibliotecas de funções para várias linguagens de programação, permitindo que o programa do usuário, rodando em IBM-PC/AT ou compatível, acesse a rede DXNET. As linguagens disponíveis até a data de edição deste manual são C, Pascal, Clipper e Basic.

Nota: No caso de μDX+ é possível expandir a rede DXNET a 256 endereços usando a informação adicional de CONJUNTO. Como cada um dos 16 conjuntos DXNET é formado de 16 endereços DXNET o total de dispositivos endereçáveis chega a 256. Entretanto, por limitações de corrente (como já comentado) esta opção só é operacional no caso de interligar os vários conjuntos via rede RS485 ou Rádio-Transmissão (com o módulo de Modem para μDX).

Características Técnicas

O μDX Série 100 apresenta as seguintes características técnicas:

Características Gerais

- 52 Instruções, incluindo lógica e aritmética de 8 bits.
- 16 timers de 8 bits (compartilhados com as variáveis) (64 para μDX+).
- 16 variáveis de 8 bits (64 para μDX+).
- Estruturação do programa por rede nodal.
- Execução do programa em modo de paralelismo lógico.
- Memória com capacidade para 127 blocos de instruções e 63 nodos. (256 blocos e 192 nodos, no caso de μDX+)
- Quatro durações de ciclo: 1/16, 1/32, 1/64 e 1/256 segundo.
- "Watch-Dog-Timer".
- 4 entradas e 4 saídas digitais, expansível para 12 entradas e 12 saídas (36 entradas e 36 saídas no caso do μDX+).
- Entrada de contagem rápida até 3000 Hz (apenas para μDX+).
- 3 entradas analógicas (por PWM).

Características Elétricas

- Alimentação (ENERGIA): 9-14VDC @ 250mA (máximo)
- Bateria Interna: 4 pilhas tipo AA (ou baterias recarregáveis)
- Consumo na alimentação pela bateria: aproximadamente 2mA. (5mA no caso de μDX+).
- Oscilador Central: 4,194304 MHz ou 16,777216 MHz (firmware ≥ 6.3)
- Temperatura de operação: 0°C até 60°C

- Entradas:

- Lógica normalmente a zero (com "pull-down" de 10K Ω).
- Detecção de 1 lógico: +2.0 até +48.0V
- Detecção de 0 lógico: -48.0V até +0.9V.
- Tensão máxima na entrada (1 minuto): 120VCA.
- Frequência de entrada: DC - 10Hz (260KHz para bloco PWMIn).
- Resistor limitador no terminal +V: 220 Ω (comum para todos).

ATENÇÃO: A partir do μ DX número 0380011 o resistor de limitação foi suprimido, permitindo sua alimentação elétrica pelo conector de entrada (bornes +V e GND).

- Tensão em +V (em aberto): 9-14VDC (conforme ENERGIA)

- Saídas:

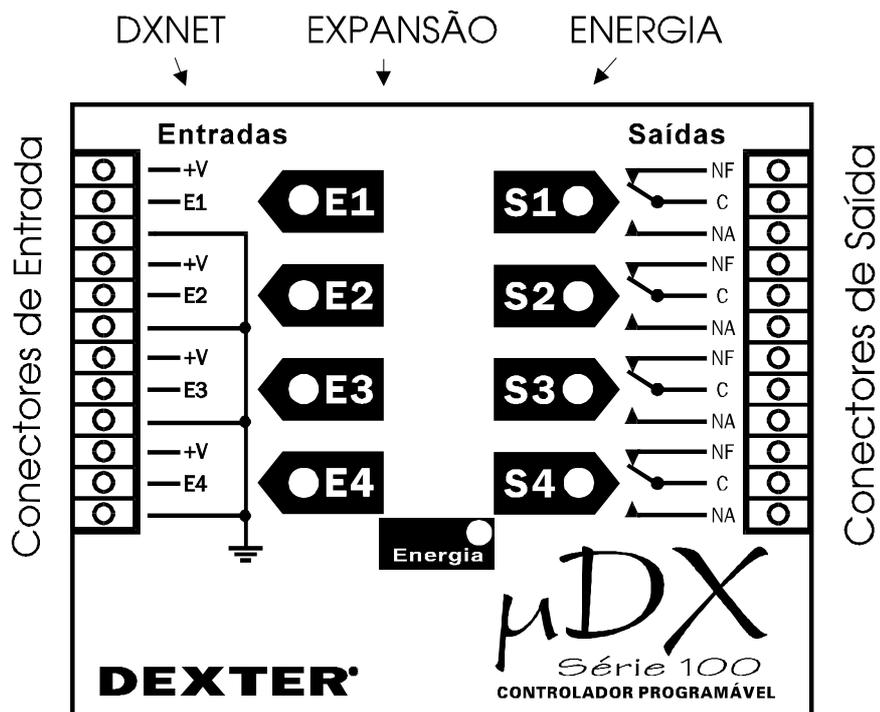
- Tipo de saída: 1 reversor (por cada saída)
- Tensão nos contatos: 30VDC/220VCA
- Potência de chaveamento: 150W ou 1220VA (CA)
70W (DC com 30V)
- Corrente máxima: 10A
- Vida útil sem carga: 100.000.000 operações
- Vida útil com carga resistiva máxima (CA): 80.000 operações
- Isolação (entre bobinas e contatos): 2.000Vef (1 minuto)

- Conector de Expansão:

- Saída de alimentação +V: máx. 100mA (9 a 14 VDC)
- Saída de alimentação +5V: máx. 8mA (+V=9VDC)
máx. 50mA (+V=14VDC)
- Saída de alimentação VCC: máx. 5mA (consumo direto das pilhas)

ANEXO A - Conectores

Os conectores do μDX são os seguintes:



Entradas

São quatro módulos de três terminais cada um, montados na lateral esquerda do μDX. Como indicado na serigrafia do painel o primeiro terminal de cada módulo é para o +V (Ligado nos modelos antigos ao conector da energia por um resistor de 220Ω, 1/8w. Nos novos modelos a ligação é direta, permitindo ligar a fonte de alimentação no conector de entrada). O segundo terminal é para o sinal de entrada e o terceiro é o comum (terra, Zero Volts, "ground"). Pode-se alimentar algum circuito externo utilizando-se o +V e o comum para isso.

Todos os +V estão interligados, assim como os comuns.

Não existe isolamento galvânica entre qualquer um dos terminais e o restante do circuito interno do μ DX.

Saídas

São quatro módulos de três terminais cada um, montados na lateral direita do μ DX. Como indicado na serigrafia do painel o segundo terminal de cada módulo corresponde ao contato móvel interno de cada relé de saída. Os outros dois terminais referem-se aos contatos dos relés, Normal Fechado e Normal Aberto.

Os relés possuem isolamento galvânica entre si e entre os contatos e o circuito interno do μ DX.

Não existe interconexão interna entre os relés.

Energia

O conector para ligação com a fonte de alimentação fica localizado na lateral traseira do μ DX, ao lado do conector da expansão.

A energia externa para alimentar o μ DX deve ser entre 9 e 14 VDC suprimindo até 250mA.

Como representado no desenho ao lado dele o contato central é ligado ao comum (Zero Volts, "ground"). O contato externo é ligado aos circuitos dos relés, na fonte interna que reduz a tensão para 5 Volts e aos terminais do conector das entradas.

Expansão

É um tipo barra de pinos, com 10 contatos em duas filas de 5. O primeiro fica no canto superior esquerdo, sendo a numeração dos contatos a seguinte:



Serve para conectar ao μDX algum circuito externo para a entrada e saída de dados de 8 bits ou para controle direto de um display e/ou teclado (ver ANEXO C - Diagramas Esquemáticos).

Significado dos contatos:

- 1 - Comum (OV)
- 2 - +V (diretamente ligado à energia)
- 3 - Pulso de carga ("Load")
- 4 - +5V (máx. 10mA)
- 5 - Pulso de relógio ("Clock")
- 6 - Sem conexão
- 7 - Entrada de dado serial ("Din")
- 8 - VCC (+5V garantidos pelas pilhas quando falta energia, máx. 5 mA)
- 9 - Saída de dado serial ("Dout")
- 10 - Sem conexão

DXNET

São dois conectores tipo P2, fêmea, que estão internamente ligados em paralelo. Em qualquer um dos dois podem ser conectados os cabos para rede DXNET ou o cabo do adaptador para a comunicação com o computador.

O contato da ponta é o sinal e o contato do corpo é o comum (Zero Volts, "Ground").

O sinal está ligado diretamente ao microcontrolador do μDX e a um resistor de 1,8KΩ (que vai ligado ao VCC).

Cuidado para não aplicar no sinal tensões superiores à +5 Volts, nem inferiores à -0,5 Volts, sob risco de dano irreversível do microcontrolador.

ANEXO B - Exemplos de Aplicações

Os seguintes exemplos foram incluídos no disquete que acompanha o μDX e podem ser utilizados imediatamente. Leia atentamente a descrição do funcionamento de cada um, assim como as instruções de como fazer para utilizá-los na prática.

Veja também o desenho do circuito de cada um, que estão juntos deste anexo.

Minuteria para prédios de até 4 andares (Arquivo: MINUT)

O programa MINUT transforma o μDX em uma minuteria com 4 saídas (em um prédio de 4 andares podemos acionar as lâmpadas de cada andar independentemente). Ao energizar momentaneamente uma entrada, a saída correspondente fica ligada durante 2 minutos. Aqui temos um bom exemplo de como transformar o bloco de mono-estável, que normalmente não é "retrigável", ou seja, se a entrada for ligada novamente durante os 2 minutos ele não contaria mais 2 minutos para desligar a saída, em um mono-estável "retrigável". Isto permite que o usuário, caso necessite de luz por mais de 2 minutos, pressione o botão ligado às entradas para obter outros 2 minutos adicionais.

Esta característica é obtida pelos blocos de função que forçam o valor 8 à variável auxiliar de cada mono-estável. Note que os mono-estáveis foram programados para minutos, onde a resolução é de 1/4 de minuto (1/4 de minuto X 8 = 2 minutos).

O circuito pode tornar-se mais inteligente com aperfeiçoamentos do programa, como acionar as lâmpadas dos andares imediatamente superior e inferior, piscar as lâmpadas uma vez alguns segundos antes do tempo se esgotar, empregar a expansão para atender a mais andares, ou mesmo ser combinado com outras atribuições como acionamento de portões ligando as luzes do pátio automaticamente.

Na montagem prática podem ser utilizados dois métodos: empregando energia elétrica da rede do prédio para acionar as entradas ou empregando o sinal +V, disponível em cada entrada, para o acionamento.

A primeira opção é a mais econômica pois aproveita, em muitos casos, a fiação elétrica já disponível para as minuterias convencionais. É preciso apenas que o neutro da rede elétrica seja conectado no terminal do COMUM no conector das entradas (basta em um apenas, já que internamente todos estão interligados). O sinal que vêm dos botões nos andares (o "vivo" da rede elétrica) deve ser conectado ao terminal de sinal de entrada através de um resistor de $470K\Omega$ (pode se de 1/8W ou 1/4W). A finalidade deste resistor é proteger a entrada do μDX, evitando aquecimento desnecessário do resistor de entrada interno.

Atenção: Neste modo de instalação é preciso ter certeza de que a energia que vem dos botões é sempre da mesma FASE.

CUIDADO: Nunca conecte os cabos para DXNET em um μDX que tenha alguma entrada ligada diretamente na rede elétrica, como neste exemplo. Isto poderia causar danos em outros μDX ou no computador, se ele estiver conectado na rede DXNET. Se for preciso utilizar a DXNET empregue opto-acopladores para isolar a tensão da rede elétrica em relação às entradas.

No outro modo de instalação o sinal +V suprido nos conectores de entrada deve ser utilizado para retornar ao sinal de cada entrada, passando através dos botões de cada andar. Isto somente será possível se a instalação destes botões for completamente isolada da rede elétrica.

Assim basta conectar um dos +V (caso todos os botões tenham um dos contatos em comum com os outros) aos botões e o retorno deles conectar aos terminais de sinal de entrada.

Alarme residencial temporizado (Arquivo: ALARME)

O programa ALARME demonstra como o μDX pode atuar em um alarme com sensoriamento com atraso, sensoriamento instantâneo e entrada de ativação.

A entrada E3 deve estar energizada para ligar o alarme. Enquanto ela estiver desligada todos os temporizadores estarão impedidos de funcionar e a saída se manterá inativa, independentemente das entradas E1 e E2.

Ao energizar E3 o programa passa a monitorar as entradas E1 e E2. Caso E2 seja ligada, ativará o mono-estável de 1 minuto (tempo que fica tocando o alarme), que por sua vez ativará o oscilador de 0,5 segundos. O mono-estável de 0,25 segundos torna a saída do oscilador simétrica, ou seja, com o mesmo período de ligado e de desligado. S1 será ligado e desligado neste ritmo, podendo comutar uma sirene ou buzina.

Já E1 também aciona o mono-estável de 1 minuto, mas somente após 10 segundos de intervalo. isto permite ao proprietário do bem protegido pelo alarme desativá-lo antes do disparo.

Este alarme pode funcionar muito bem mesmo na falta de luz, caso seja utilizada uma bateria potente para permitir o funcionamento do μDX e da sirene ou buzina. Na instalação prática você pode empregar, para E3, um interruptor escondido ou com chave (semelhante aos utilizados para trancar o teclado dos computadores).

Todas as entradas, na instalação, utilizam o sinal +V (conectado ao sinal de entrada através dos sensores ou chaves) para serem acionadas.

Árvore de Natal (Arquivo: NATAL)

O programa NATAL foi elaborado com a intenção de servir como "pisca-pisca" para 4 conjuntos de lâmpadas decorativas para árvores de natal.

Embora a seqüência de ligação seja rigorosamente repetitiva, com os valores escolhidos para os atrasos a seqüência parece aleatória, já que os valores não tem um múltiplo comum próximo.

Note que foram feitos 4 osciladores com atrasos cujo funcionamento é bastante simples: inicialmente as saídas dos atrasos V15 e V14 estão desligadas. Com isto a chave NF está fechada, acionando a entrada do atraso V15. Após VO segundos (2,75 segundos - 44 para a resolução de 1/16 segundo) a saída deste bloco é ligada, acionando o relé de S1 e o atraso V14. Após mais 2,75 segundos a saída deste atraso liga, abrindo a chave NF. Isto forçará que a saída do atraso V15 seja desligada (desligando também o relé de S1) e logo a seguir a saída do próprio atraso V14, recomeçando o ciclo.

Um aperfeiçoamento interessante seria modificar a duração do ciclo do μDX para 1/32 ou 1/64 e depois utilizar o bloco PWMin para medir, simplificadaamente, a frequência aproximada de um sinal de entrada, proveniente de um amplificador, por exemplo. Se o resultado do PWMin for empregado para alterar a temporização dos osciladores o jogo de luzes poderá piscar com um ritmo influenciado pela música no amplificador.

Irrigação de jardim (Arquivo: JARDIM)

O programa JARDIM permite irrigar automaticamente um jardim todos os dias sempre às 18 horas e 30 minutos, desde que o sensor de umidade esteja desativado (ou seja, não detecte umidade na terra).

Antes de ligar a válvula de água (S2), o programa liga e desliga, em ciclos de 0,25 segundos, durante 5 segundos o relé S1, que pode ser ligado a algum sinalizador sonoro ou visual. Esta sinalização pode servir para avisar as pessoas que porventura estejam no gramado que vai ser iniciada a irrigação. Depois deste aviso o programa ainda espera 10 segundos e então liga a água por 10 minutos.

A entrada E1, quando ativada, inibe o sistema de irrigação. A entrada E2 é ligada a um sensor de umidade, descrito no ANEXO C - Diagramas Esquemáticos. Note que a chave NF em série com este sinal impede que o circuito detecte a umidade após ligar a água.

A válvula de água pode ser uma válvula tipo solenóide, adequada à vazão desejada, e que deve ser conectada em série com os terminais do relé S2 (C e NA). O interruptor para inibir a irrigação pode ser igual aos utilizados nas instalações residenciais, para ligar e desligar lâmpadas, conectado entre o terminal +V e o terminal de sinal de E1.

Não esqueça de manter o μDX com as pilhas (para manter o relógio funcionando mesmo nas faltas de energia) e de acertar o relógio corretamente.

Seqüenciador (Arquivo: SEQUENC)

Este programa liga as saídas do μDX em seqüência (uma saída de cada vez), invertendo o sentido de ligação a cada 4 segundos. Isto permite, por exemplo, ligar lâmpadas em seqüência para simular movimento. Para ligar mais de 4 lâmpadas basta colocar em paralelo:

... S1 S2 S3 S4 S1 S2 S3 S4 S1 S2 S3 S4 ...

Assim teremos "ondas" se movimentando da esquerda para a direita e da direita para a esquerda, alternadamente.

ATENÇÃO: Não se esqueça do limite de corrente das saídas do μDX que é de 10 Ampéres por relé.

Como sugestão de aperfeiçoamento, no caso de aplicação como atrativo de uma vitrine, pode-se utilizar um sensor para perceber quando alguma pessoa se aproxima e então a velocidade do seqüenciador mudar para chamar a atenção.

Simulador de presença (Arquivo: TEMPO)

Este programa utiliza o relógio interno do μDX para acionar as saídas, a partir das 19 horas, todos os dias, em uma seqüência aparentemente caótica. As saídas devem ser ligadas à lâmpadas da residência para simular a presença de pessoas na casa (pode-se conectar as saídas em paralelo com os interruptores convencionais das lâmpadas que se queira ligar).

As saídas são acionadas durante os intervalos de tempo mostrados no programa. Para modificar estes intervalos basta alterar a programação dos blocos temporizadores (veja Elaborando Programas).

A entrada E1 deve estar ligada para permitir o funcionamento do circuito.

Este programa pode ser adicionado ao de ALARME para fazer com apenas um μ DX tanto o alarme como um simulador de presença.

Não esqueça que o relógio interno do μ DX deve estar ajustado corretamente e que, nesta aplicação, é importante utilizar as pilhas para manter o μ DX funcionando corretamente (e o seu relógio) após eventuais faltas de energia elétrica.

Controle de Temperatura (Arquivo: TEMPERAT)

Este programa permite controlar a temperatura de uma sala ou do interior de uma caixa (por exemplo uma chocadeira).

Para isso, liga-se a resistência de aquecimento na saída S1. A entrada do sensor de temperatura é feita através de um circuito externo que converte o sinal analógico medido em um outro de frequência variável. Conectando-se este último na entrada E3 pode-se obter o valor da medida através do bloco PWMin. Quanto maior a temperatura, menor o valor lido pelo PWMin.

ATENÇÃO: Os valores de comparação indicados não representam diretamente a temperatura. Para isso seria necessário que o circuito conversor tensão-frequência estivesse precisamente calibrado.

Para explicar o funcionamento digamos que a temperatura esteja muito baixa. Neste caso o PWMin estará lendo um valor alto. Digamos 150. Ora, se $VO=150$ e, portanto, $VO>100$ então $V1$ ficara igual a 1, acionando S1. Com a resistência de aquecimento ligada (via S1) a temperatura sobe fazendo VO diminuir de valor. Quando VO ficar menor que 90, $V1$ ficara com zero, desligando S1. Assim a temperatura deve ficar oscilando entre 90 e 100.

Para mudar os pontos de decisão basta re-programar os limites nos blocos de função. Os valores disponíveis, pelo limite de 8 bits, estão entre 0 e 255.

O circuito conversor tensão/largura de pulso está apresentado no início deste manual (ver Conexões nas Entradas e Saídas). Para fazer a leitura de temperatura deve-se alterar o ganho do circuito de correção de nível (que usa 1/4 de um circuito integrado LM324) para uma sensibilidade capaz de detectar a variação da tensão de junção de um diodo que sirva como sensor (pode ser um 1N4148). Sugere-se utilizar um dos outros amplificadores operacionais presentes no circuito integrado para fazer esta amplificação.

A DEXTER comercializa um pequeno circuito capaz de ler temperatura ou outra variável analógica via entrada PWM do μ DX.

Semáforo duplo completo (Arquivo: SEMAFORO)

O programa SEMAFORO transforma o μ DX em um semáforo completo, com duas sinaleiras para veículos e duas sinaleiras para pedestres. É necessário utilizar o circuito de expansão de entradas e saídas, pois as duas sinaleiras para veículos consomem 6 saídas e as duas para pedestres mais 4 saídas, perfazendo 10 saídas (com a expansão temos 8 saídas adicionais).

As saídas do programa são:

- S1 : Sinal Vermelho - Sinaleira de veículos 1
- S2 : Sinal Amarelo - Sinaleira de veículos 1
- S3 : Sinal Verde - Sinaleira de veículos 1
- S4 : Sinal Vermelho - Sinaleira de pedestres 1
- S1e : Sinal Verde - Sinaleira de pedestres 1
- S2e : Sinal Vermelho - Sinaleira de veículos 2
- S3e : Sinal Amarelo - Sinaleira de veículos 2
- S4e : Sinal Verde - Sinaleira de veículos 2
- S5e : Sinal Vermelho - Sinaleira de pedestres 2
- S6e : Sinal Verde - Sinaleira de pedestres 2

S7e e S8e não são utilizadas.

A entrada E1, quando energizada, força o trancamento do semáforo em sinaleira para veículos 1 aberta (em Verde). Já a entrada E2 força o mesmo trancamento para a sinaleira de veículos 2. Isto é útil, por exemplo, para a passagem de comitivas ou veículos de urgência pelo cruzamento.

O semáforo tem uma duração de 20 segundos em aberto para cada lado e 3 segundos de amarelo na comutação do verde para o vermelho. Neste período o sinal vermelho da sinaleira de pedestres pisca, indicando o término do tempo de sinaleira de veículos fechada.

Note a utilização de rótulos para facilitar a conexão entre blocos distantes.

Controle de reservatório para processo industrial (
Arquivo: RESERV)

Digamos que devemos controlar um reservatório, no qual temos sensor de nível inferior, sensor de nível superior, válvula para entrada de água, válvula de drenagem, botoeira para operador, botoeira de pausa, sinal luminoso para operador, bomba para drenagem e agitador mecânico. O processo é o seguinte: Inicialmente o reservatório está vazio e, portanto, o sensor de nível alto está desativado e o sensor de nível baixo ativado. Caso a botoeira de pausa não esteja acionada devemos permitir a entrada de água (acionando a válvula de entrada de água), até que o sensor de nível alto energize. Desliga-se a válvula de entrada de água e liga-se o aviso luminoso para o operador. O operador irá diluir algum produto na água do tanque (hipoteticamente) e pressionará uma botoeira que avisa ao μDX que a diluição está completa. A seguir o μDX irá acionar o agitador mecânico durante 4 horas, de forma a homogeneizar a mistura. Após estas 4 horas o μDX desliga o agitador e espera outras 4 horas para decantar o material não solúvel. Por fim, é aberta a válvula de drenagem e acionada a bomba, de forma a esvaziar o reservatório. Ao atingir o nível inferior a bomba é desligada, a válvula de drenagem fechada e retorna-se ao estado inicial, ou seja, se a botoeira de pausa não estiver acionada o processo se reinicia, com a entrada de água no tanque.

Este exemplo é interessante para realçar a importância de, em processos mais complexos, fazer um diagrama de estados. Assim, neste processo pode-se identificar 6 estados:

<i>Estado 0</i>	<i>Tanque Parado</i>	<i>Nenhum acionamento ligado.</i>
<i>Estado 1</i>	<i>Entrada de água</i>	<i>Apenas válvula de água ligada.</i>
<i>Estado 2</i>	<i>Avisa operador</i>	<i>Aviso luminoso acionado.</i>
<i>Estado 3</i>	<i>Agitação</i>	<i>Agitador acionado.</i>
<i>Estado 4</i>	<i>Decantação</i>	<i>Nenhum acionamento ligado.</i>
<i>Estado 5</i>	<i>Descarga</i>	<i>Válvula de saída e bomba ligadas.</i>

As condições para trocar de estado são as seguintes:

<i>Estado 0 para Estado 1</i>	<i>Chave nível baixo tanque acionada. Botoeira de pausa não acionada.</i>
<i>Estado 1 para Estado 2</i>	<i>Chave nível alto tanque acionada.</i>
<i>Estado 2 para Estado 3</i>	<i>Botoeira de operador ciente acionada.</i>
<i>Estado 3 para Estado 4</i>	<i>Passaram 4 horas desde início estado 3.</i>
<i>Estado 4 para Estado 5</i>	<i>Passaram 4 horas desde início estado 4.</i>
<i>Estado 5 para Estado 0</i>	<i>Chave nível baixo tanque acionada.</i>

Ou seja, temos uma máquina de estados com todos os estados definidos. Com base nisso é relativamente fácil elaborar o programa para o μDX. No programa RESERV.UDX (presente no disquete), usei a variável vO para criar a máquina de estados. Basicamente esta variável vai assumindo os valores de 0 a 5, conforme o estado atual. A mudança de estado é determinada pelas condições descritas acima.

Esta maneira de resolver problemas de automação (via máquina de estados) é muito poderosa, pois mesmo processos muito complexos podem ser tratados de forma sistemática. Além disso, o traslado para o μDX é praticamente direto. Basta eleger uma variável que definirá o estado corrente e colocar as condições para a mudança de seu valor.

Agora falta definir as entradas e saídas necessárias. Os sinais de entrada e saída são:

<i>Entradas:</i>	<i>Sensor Inferior</i>	<i>E1</i>
	<i>Sensor Superior</i>	<i>E2</i>
	<i>Botoeira Operador</i>	<i>E3</i>
	<i>Botoeira Pausa</i>	<i>E4</i>
<i>Saídas:</i>	<i>Válvula d'água</i>	<i>S1</i>
	<i>Sinal Luminoso</i>	<i>S2</i>
	<i>Válvula Saída e Bomba de Drenagem</i>	<i>S3</i>
	<i>Agitador Mecânico</i>	<i>S4</i>

Note que a válvula de saída e a bomba de drenagem são acionadas simultaneamente pela saída S3. Examine cuidadosamente o programa RESERV.UDX. Foram inseridos comentários para ajudar...

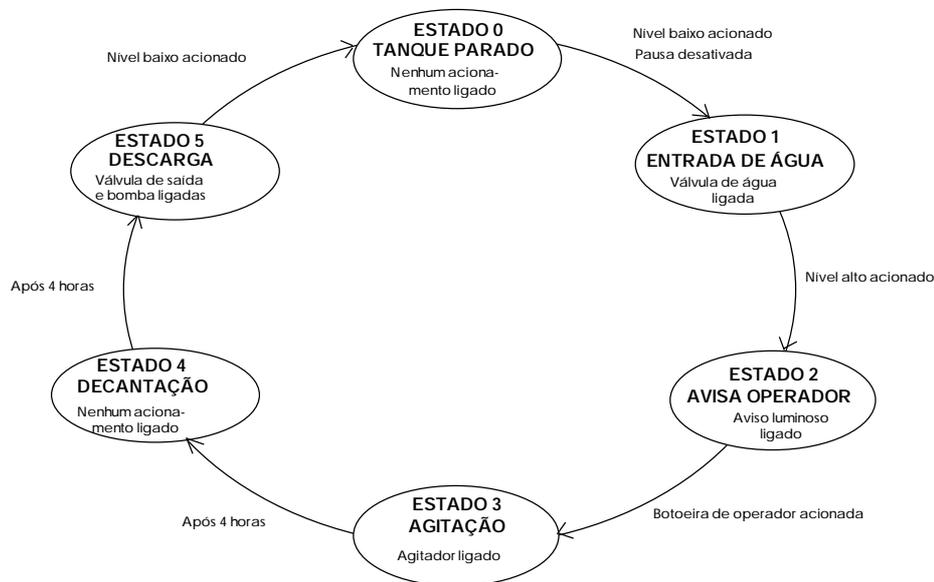


DIAGRAMA DE ESTADOS PARA RESERV.UDX

Agora digamos que, ao instalar o sistema, surgiu a necessidade de abortar o ciclo de decantação via comando do operador. Ou seja, se o operador quiser, ele pode encerrar o ciclo de decantação antes de concluir as 4 horas programadas. Para isso vou utilizar a própria botoeira de operador (E3). Se ela for acionada durante o estado 4 (decantação) o μDX pula para o estado 5 (descarga). O diagrama de estados fica:



DIAGRAMA DE ESTADOS PARA RESERV1.UDX

O programa que executa o diagrama acima é RESERV1.UDX.

Prensa Industrial (Arquivo: PRENSA)

Outro exemplo de programação via máquina de estados. Ao pressionar uma tecla ligada à entrada E4, a saída S1 (responsável pela pressurização do sistema hidráulico ou pneumático) é acionada. Esta entrada E4 serve como chave liga/desliga para a prensa. O pistão da prensa deve avançar (S2 - rótulo 5) até o sensor fim de curso ligado à E2 (rótulo 1). A seguir o pistão deve recuar (S3 - rótulo 6) até o fim de curso E1 (rótulo 0). Este movimento deve ser repetido 5 vezes (estado 1 e estado 2 da máquina de estados).

A seguir o µDX pula para o estado 3, que continua recuando o pistão via saída S3, mas temporiza 1 segundo antes de passar para o estado 4, que aciona a saída S4, que abre a tampa do compartimento da prensa (para expulsão do material prensado).

Após dois segundos a tampa já está aberta e o pistão é acionado até o fim de curso E3. Note que a tampa permanece aberta via saída S4 (rótulo 7).

Depois de atingir E3, o µDX pula para estado 6 (v1=6), que faz o pistão recuar até o fim de curso E2 (rótulo 1), mantendo a porta aberta.

Uma vez atingido E2, a porta é fechada e o pistão continua recuando até atingir o fim de curso E1 (rótulo 0).

Quando o pistão atinge E1, o sistema retorna ao estado 0 e, caso a entrada E4 esteja acionada, inicia novo ciclo de prensagem.

Assim, neste processo pode-se identificar 7 estados:

<i>Estado 0</i>	<i>Prensa Parada</i>	<i>Nenhum acionamento ligado.</i>
<i>Estado 1</i>	<i>Prensa Avançando</i>	<i>Apenas avanço acionado.</i>
<i>Estado 2</i>	<i>Prensa Recuando</i>	<i>Apenas recuo acionado.</i>
<i>Estado 3</i>	<i>Recuo p/abrir Porta</i>	<i>Recuo acionado durante 1 seg.</i>
<i>Estado 4</i>	<i>Abertura de Porta</i>	<i>Abertura de porta acionado.</i>
<i>Estado 5</i>	<i>Descarga</i>	<i>Avanço e abertura de porta acionados.</i>
<i>Estado 6</i>	<i>Recuo p/fechar Porta</i>	<i>Abertura de porta e recuo acionados.</i>
<i>Estado 7</i>	<i>Fechamento de Porta</i>	<i>Recuo acionado.</i>

As condições para trocar de estado são as seguintes:

<i>Estado 0 para Estado 1</i>	<i>Chave liga/desliga E4 acionada.</i>
<i>Estado 1 para Estado 2</i>	<i>Fim de curso E2 acionado.</i>
	<i>Contador de ciclos (v0) < 5.</i>
<i>Estado 1 para Estado 3</i>	<i>Fim de curso E2 acionado.</i>
	<i>Contador de ciclos (v0) = 5.</i>
<i>Estado 2 para Estado 1</i>	<i>Fim de curso E1 acionado.</i>
<i>Estado 3 para Estado 4</i>	<i>Passou 1 segundo desde início do estado 3.</i>
<i>Estado 4 para Estado 5</i>	<i>Passou 2 segundos desde início do estado 4.</i>
<i>Estado 5 para Estado 6</i>	<i>Fim de curso E3 acionado.</i>
<i>Estado 6 para Estado 7</i>	<i>Fim de curso E2 acionado.</i>
<i>Estado 7 para Estado 0</i>	<i>Fim de curso E1 acionado.</i>

Agora falta definir as entradas e saídas necessárias. Os sinais de entrada e saída são:

<i>Entradas:</i>	<i>Fim de curso pistão recolhido</i>	<i>E1</i>
	<i>Fim de curso pistão prensando</i>	<i>E2</i>
	<i>Fim de curso pistão expulsando material</i>	<i>E3</i>
	<i>Chave liga/desliga</i>	<i>E4</i>
<i>Saídas:</i>	<i>Pressurização do sistema</i>	<i>S1</i>
	<i>Avança pistão</i>	<i>S2</i>
	<i>Recua pistão</i>	<i>S3</i>
	<i>Abertura de tampa</i>	<i>S4</i>

Note que foram utilizadas duas variáveis neste programa (além das variáveis utilizadas nos blocos de temporização):

<i>v0</i>	<i>Contador de número de ciclos de prensagem.</i>
<i>v1</i>	<i>Indica estado da máquina de estados da prensa.</i>

A variável v0 é incrementada cada vez que a prensa entra no estado 1. Para isso foi utilizado um bloco de Pulso, pois este gera um pulso com duração de apenas um ciclo de execução do μDX, e assim a variável é incrementada somente uma vez.

O diagrama de estados para este exemplo fica:

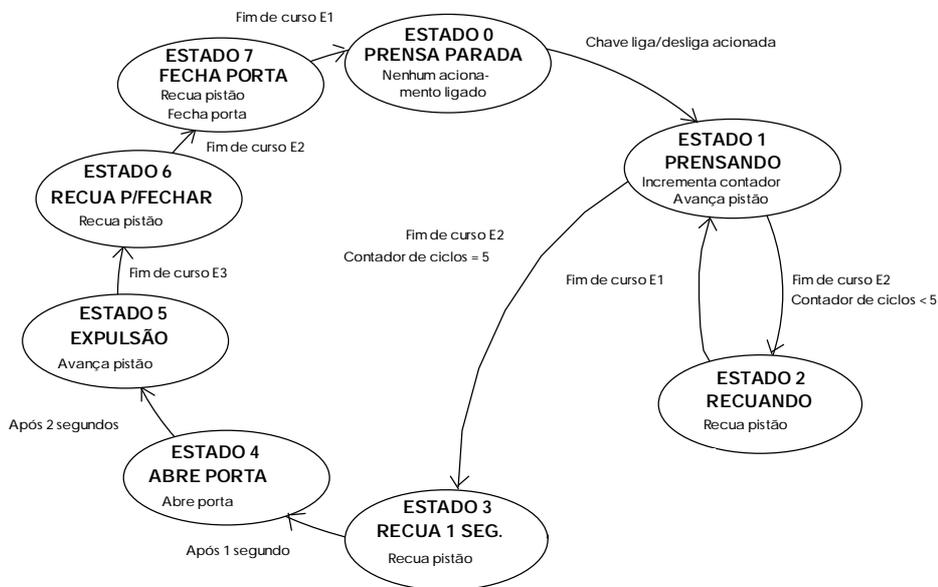
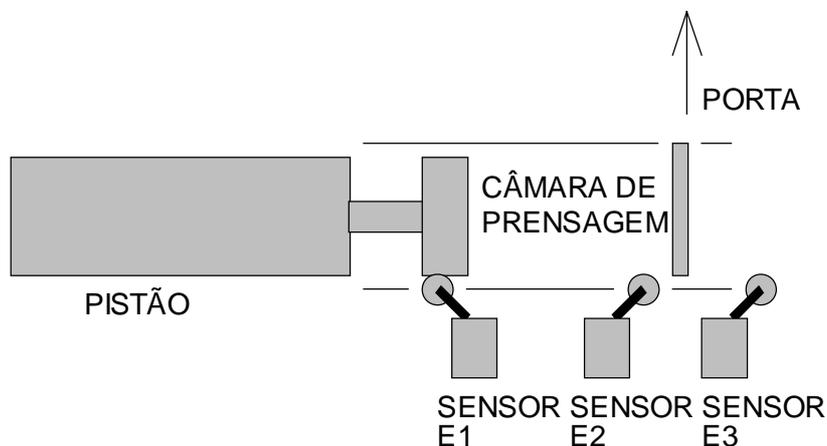


DIAGRAMA DE ESTADOS PARA PRENSA.UDX

O esquema de sensores e acionamentos da prensa é o seguinte:



Tanto a variável vO (contador de ciclos de prensagem) como vI (variável de estado da máquina de estados) são inicializados pelo bloco NodoEL no início do programa. Além disso, são utilizadas chaves NF em vários pontos do programa para isolar nodos, evitando que um estado interfira no outro.

Controle de Horário de Escolas (Arquivo: ESCOLA3)

Este programa permite controlar uma sirene, para que toque em horários determinados. Estes horários seriam, no caso deste exemplo, as 7:55, 8:55, 9:55, 10:55, 11:55, 13:55, 14:55, 15:55, 16:55, 17:55, 18:55, 19:55, 20:55, 21:55, e 22:55. Sábado e domingo o circuito é inibido. A entrada E1, caso energizada, permite atrasar em uma hora todos os horários. Assim, se o relógio de tempo real do μ DX for programado com o horário de verão (1 hora adiantado), durante a vigência deste horário mantem-se a entrada E1 desenergizada. Quando o horário de verão acabar, basta energizar E1 para atrasar todos os horários em uma hora, obtendo-se o horário convencional sem a necessidade de ajustar o horário do relógio de tempo real do μ DX.

Obviamente, este programa necessita que o μ DX possua pilhas ou baterias recarregáveis para manter o relógio funcionando no caso de uma interrupção da energia elétrica.

ANEXO C - Diagramas Esquemáticos

Neste anexo foram incluídos os diagramas esquemáticos do μDX e de alguns circuitos que podem ser empregados na expansão ou na conexão de sinais de entrada especiais.

O diagrama do μDX incluído aqui serve apenas como informação técnica e corresponde à versão do μDX que acompanha este manual. Não deve ser utilizado para manutenção ou alterações, o que implicaria na perda imediata da garantia.

A DEXTER não se responsabiliza por falhas ou mau funcionamento dos demais circuitos aqui mostrados.

Os diagramas apresentados são exclusivamente para montagens individuais e de uso próprio do comprador do μDX e não devem ser utilizados comercialmente.

Opto-acoplador

O esquema anexo mostra quatro circuitos opto-acopladores, uma para cada entrada do μDX. Eles podem ser montados separadamente, conforme quantos sejam necessários.

O LED na entrada de tensão alternada ilumina-se quando existe tensão presente. Ele também serve para proteger o LED do opto-acoplador contra a tensão reversa. O uso de um capacitor reduz muito a tensão que fica sobre o resistor de proteção, permitindo que ele seja de pequeno valor e de baixa potência de dissipação (1/8 W serve muito bem). Ele serve para limitar a corrente quando aparece tensão na entrada do circuito com o capacitor ainda descarregado.

Na saída do opto-acoplador o capacitor de 4,7 μ F forma um circuito RC com o resistor na entrada do μ DX, que é de 10K Ω , mantendo a tensão de saída no nível 1 por cerca de 47ms. Isto evita que a entrada, com semi-ciclos durando 17ms, possa ser detectada algumas vezes como desligada (se o μ DX testar o estado da entrada justamente no momento do semi-ciclo errado).

Expansão para 8 Entradas e 8 Saídas

Este circuito emprega apenas dois circuitos integrados, um "shift-register" conversor série-paralelo e o outro paralelo-série. Os controles de carga e o "clock" de deslocamento são gerados automaticamente pelo μ DX quando ele executa um bloco de instrução de expansão (EXP).

No esquema apresentado não foram incluídos nenhum circuito adicional para reforçar a corrente de saída (para ligar um relé, por exemplo) nem o circuito de proteção necessário para as entradas, isto porque cada circuito poderá ser diferente de acordo com a utilização específica de cada entrada ou saída.

Note-se que a alimentação do circuito e a que está presente nos conectores de entrada e de saída é proveniente do +5V produzido internamente no μ DX, sendo, portanto, de corrente limitada ao especificado nas características técnicas.

Expansão para Teclado e Display

Este circuito utiliza a estrutura do esquema anterior para a expansão. Devido ao aumento de consumo de corrente optou-se por regular a alimentação de +5V a partir do +V.

O gerador de pulsos com o 555 serve para reduzir o tempo em que os displays ficam ligados para cerca de 10%. Isto reduz muito a corrente média consumida e deixa brilho suficiente no mostrador.

O teclado é do tipo empregado em telefones, com uma matriz de 3x4 elementos, como indicado no diagrama esquemático.

A utilização do display e do teclado depende do μDX estar executando algum bloco de instrução de expansão, devidamente programado para atender a este modo (Opção Teclado/Display acionada).

O transistor Q1 somente é ligado quando o programa no μDX inicia a rotina de leitura de valores digitados no teclado, acendendo-se os pontos decimais nos displays.

O modelo dos displays pode ser alterado, respeitando-se que seja tipo catodo comum e com a pinagem idêntica a mostrada no esquema.

Atenção: Os controladores μDX atualmente são fornecidos com opção para Expansão tipo Entradas / Display de 5 dígitos, em vez da Expansão tipo Teclado / Display. Isto porque já está disponível o periférico de Interface Homem / Máquina, que substitui com vantagens o circuito descrito acima. Entretanto, a Dexter pode fornecer controladores μDX compatíveis com a Expansão tipo Teclado / Display sob consulta.

Expansão para 8 Entradas e Display de 5 dígitos

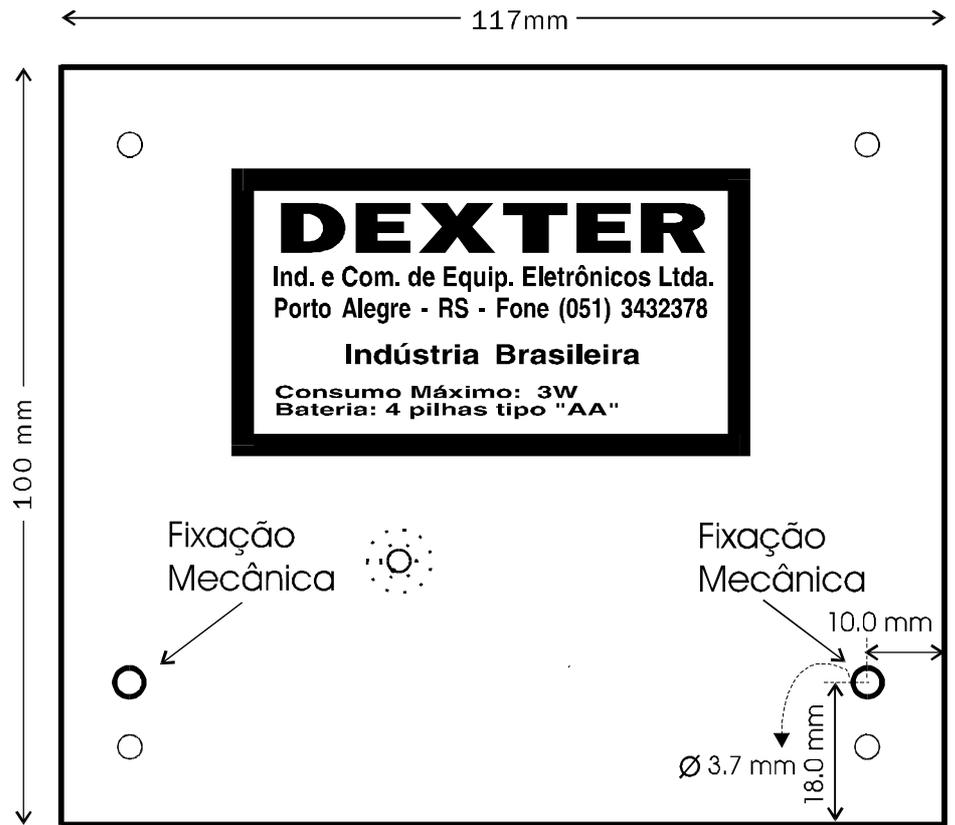
Este circuito é semelhante ao anterior, mas lê diretamente 8 entradas digitais e aciona um display de 5 dígitos (que pode mostrar um valor de 0 a 65535, usando duas variáveis do μDX concatenadas). Devido ao aumento de consumo de corrente optou-se por regular a alimentação de +5V a partir do +V.

O gerador de pulsos com o 555 serve para reduzir o tempo em que os displays ficam ligados para cerca de 10%. Isto reduz muito a corrente média consumida e deixa brilho suficiente no mostrador.

A utilização do display e das entradas depende do μDX estar executando algum bloco de instrução de expansão, devidamente programado para atender a este modo (Opção Teclado/Display acionada).

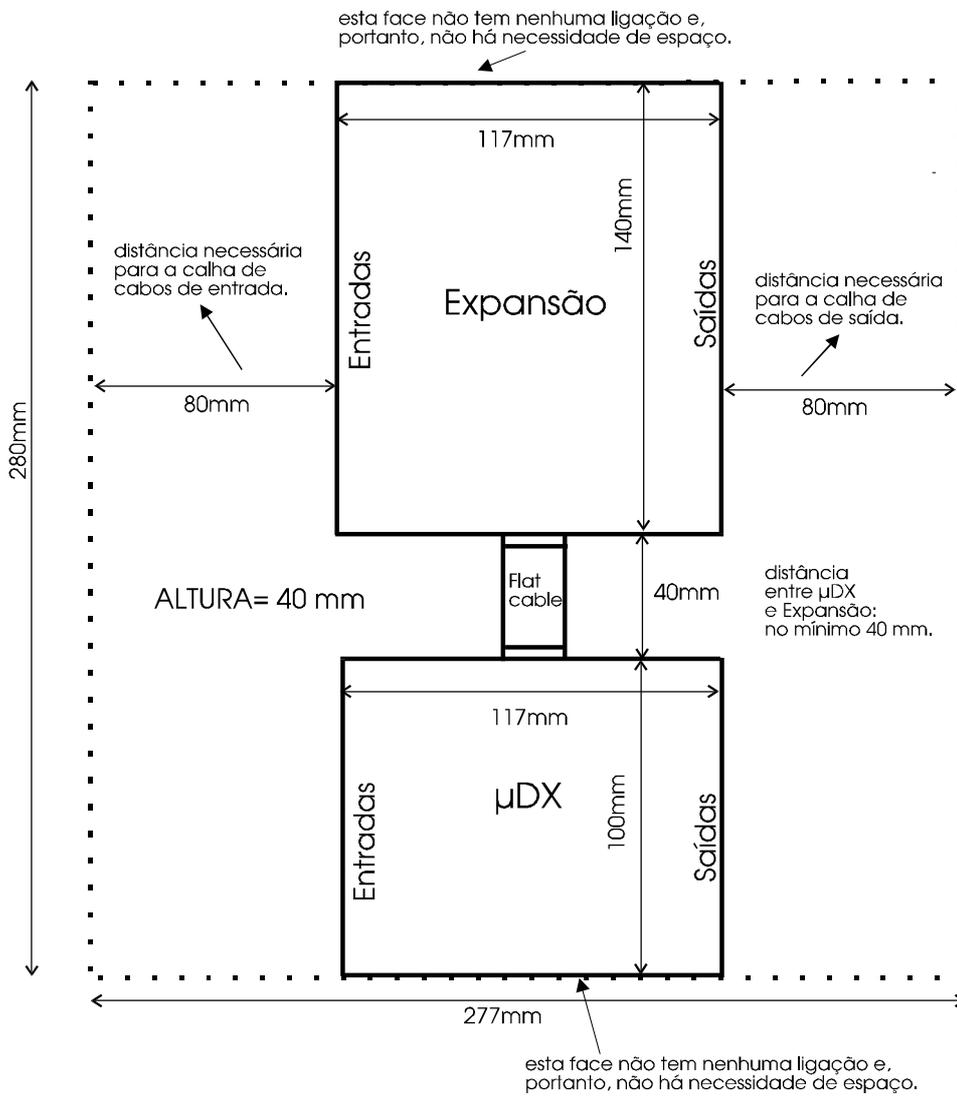
Fixação Mecânica

Para fixar o μDX existem dois furos na face inferior da caixa, conforme o desenho abaixo. Estes furos são disponíveis abrindo-se o μDX (retirando a tampa superior) e deslocando para fora de seu suporte o porta-pilhas.



VISTA DA PARTE INFERIOR DA CAIXA

Área ocupada por μDX + Expansão



Manutenção

O não funcionamento correto de qualquer uma das partes do μDX (seja o próprio μDX, a fonte de alimentação, o adaptador ou o programa) deverá ser comunicado diretamente à DEXTER.

Evite qualquer tentativa de conserto, adaptação ou configuração que não tenha sido cuidadosamente abordada neste manual.

A DEXTER não se responsabiliza pelo uso indevido ou incorreto do μDX ou das partes que o acompanham.

Leia este manual com atenção antes de energizar o μDX.

Garantia

A DEXTER oferece uma garantia de 1 (um) ano, a contar da data da compra, para reposição ou conserto do todo ou das partes do μDX no caso de mau funcionamento ou defeitos originários na fábrica.

Esta garantia deixa de vigorar caso o defeito apresentado for resultante do uso indevido ou incorreto do todo ou das partes do μDX, assim como no caso de serem feitas alterações de qualquer espécie em qualquer das partes do μDX, sem autorização por escrito da DEXTER.

A DEXTER garante que o disquete que acompanha o μDX está isento de contaminação pelos vírus de computador conhecidos até a data de fabricação.

Não estão incluídos nesta garantia os custos com transporte do μDX ou de suas partes, tanto para recebimento como para devolução.

Esta garantia se restringe ao controlador programável μDX, não se estendendo ao processo controlado, nem a sensores e/ou acionamentos ligados ao controlador. O bom funcionamento do μDX pressupõe uma linha de alimentação sem ruídos e, no caso de acionamento de cargas indutivas, a instalação de supressores de ruído.

A DEXTER não se responsabiliza pela aplicação do μDX em processos perigosos ou de risco de vida.

DEXTER Indústria e Comércio de Equipamentos Eletrônicos Ltda.

Av. Pernambuco, 1328 Cj.309 - Porto Alegre - RS - Fone: (0xx51) 3343-2378, 3343-5532

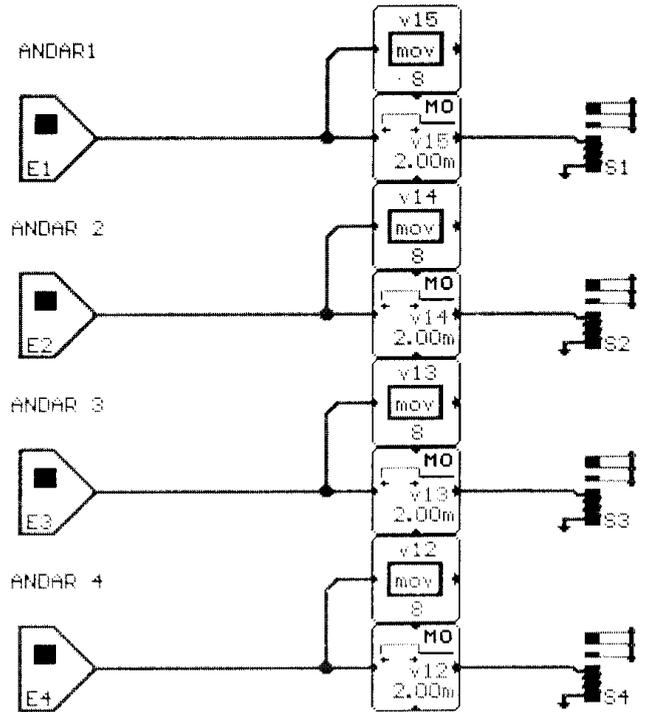
E-mail: dexter@dexter.ind.br

Internet: www.dexter.ind.br

Controlador Programavel uDX Serie 100

Nome do programa: MINUT.UDX

Numero de blocos para uDX utilizados: 8



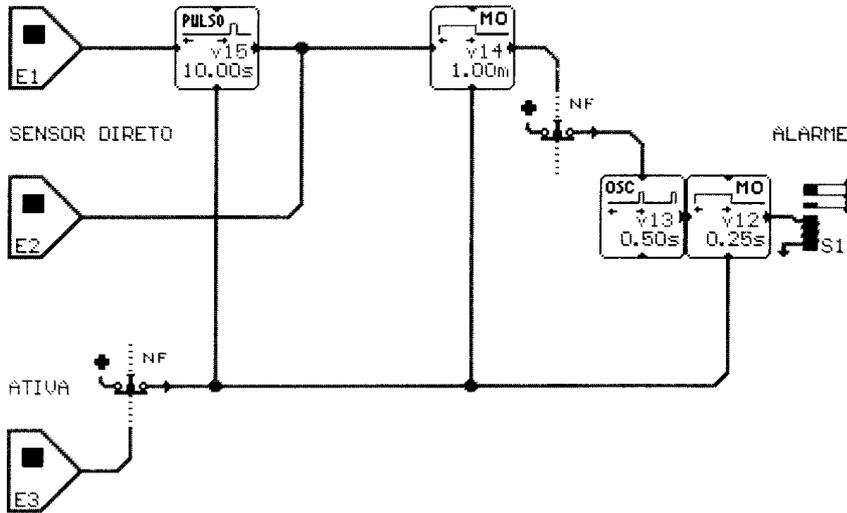
Controlador Programavel uDX Serie 100

Nome do programa: ALARME.UDX

Numero de blocos para uDX utilizados: 6



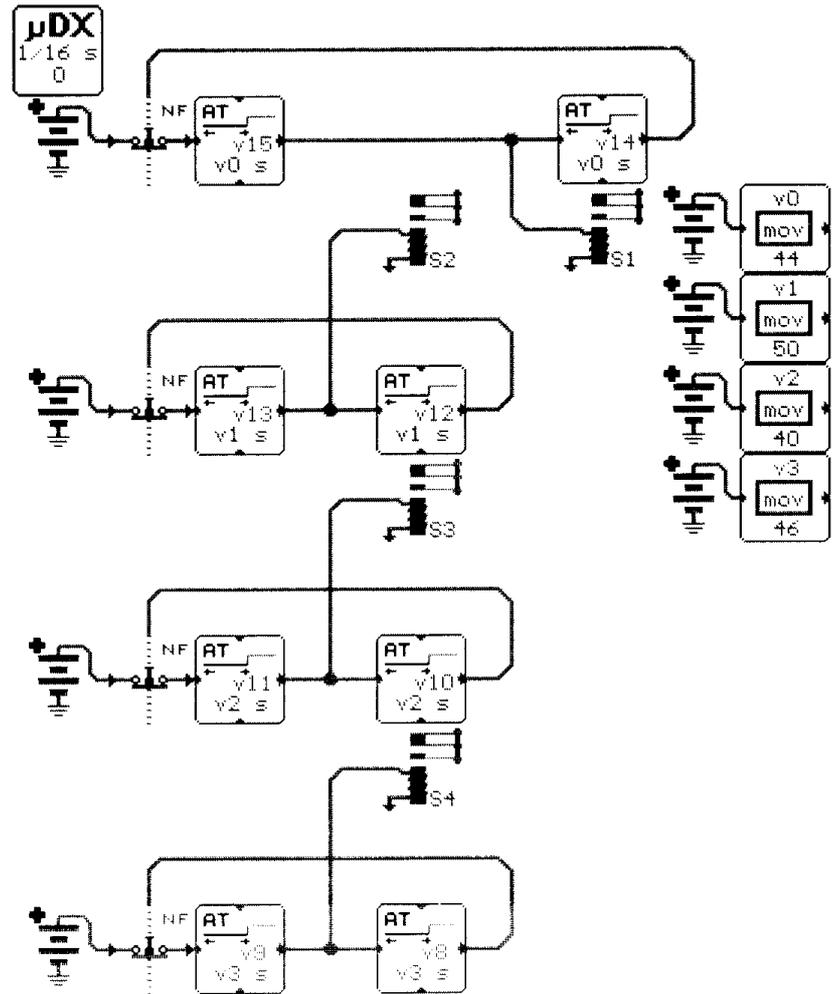
SENSOR C/ATRASO



Controlador Programavel uDX Serie 100

Nome do programa: NATAL.UDX

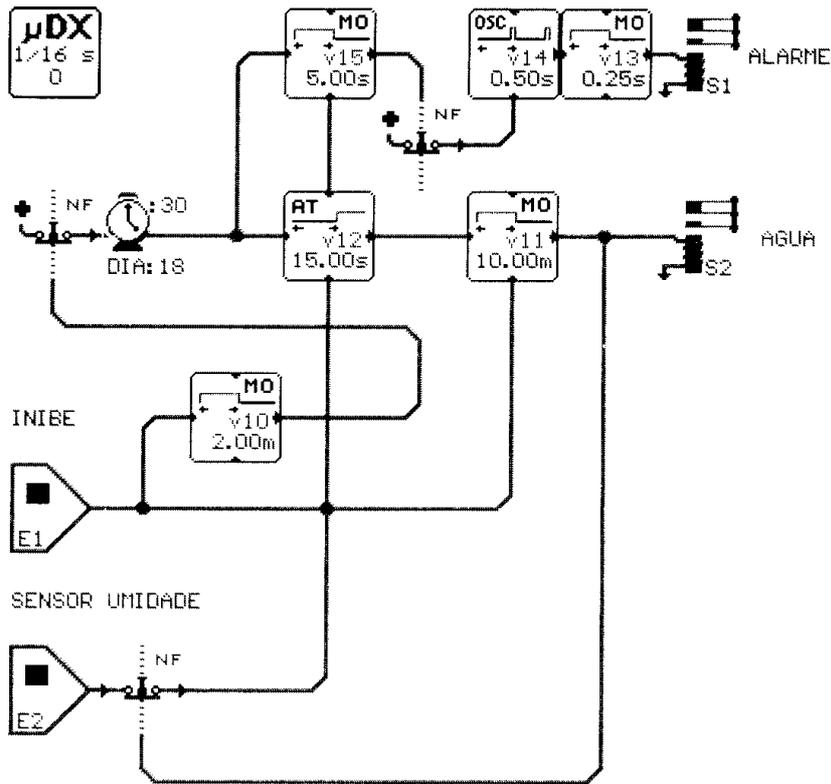
Numero de blocos para uDX utilizados: 16



Controlador Programavel uDX Serie 100

Nome do programa: JARDIM.UDX

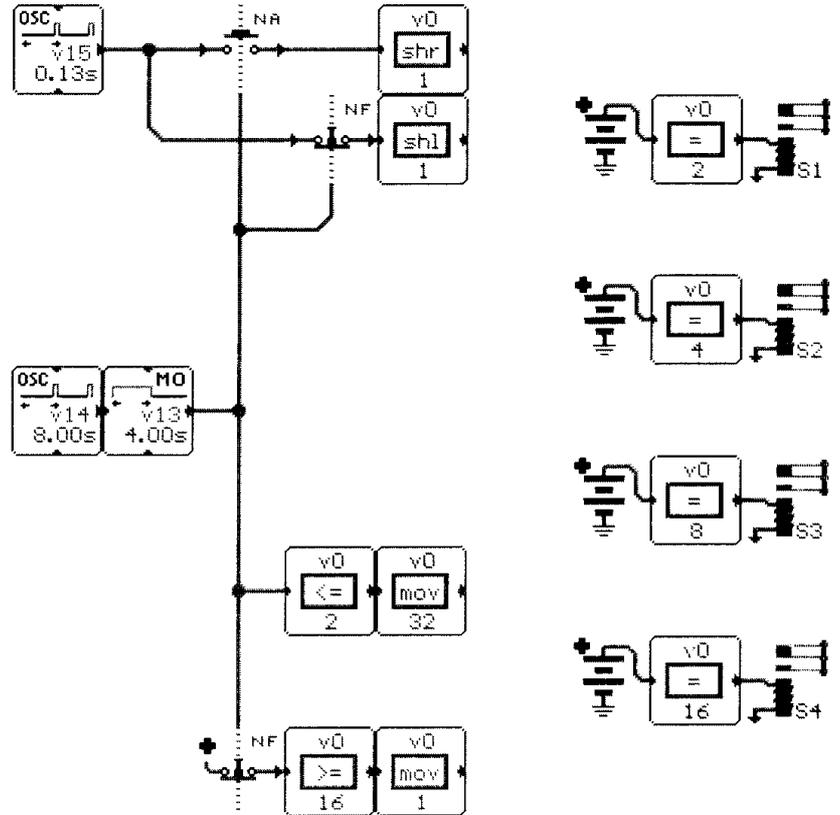
Numero de blocos para uDX utilizados: 10



Controlador Programavel uDX Serie 100

Nome do programa: SEQUENC.UDX

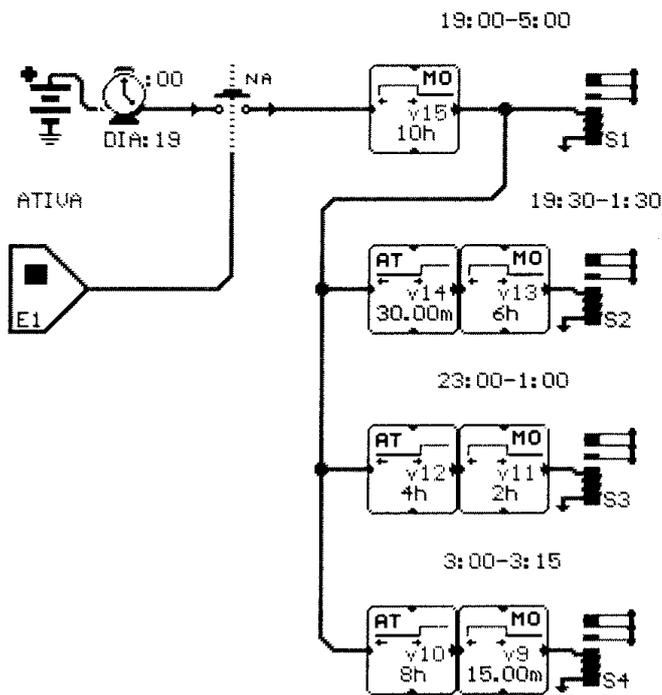
Numero de blocos para uDX utilizados: 16



Controlador Programavel uDX Serie 100

Nome do programa: TEMPO.UDX

Numero de blocos para uDX utilizados: 9



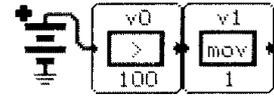
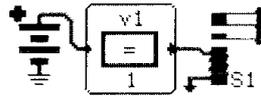
Controlador Programavel uDX Serie 100

Nome do programa: TEMPERAT.UDX

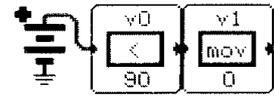
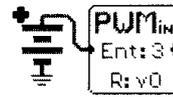
Numero de blocos para uDX utilizados: 6



Aquecedor



Limites

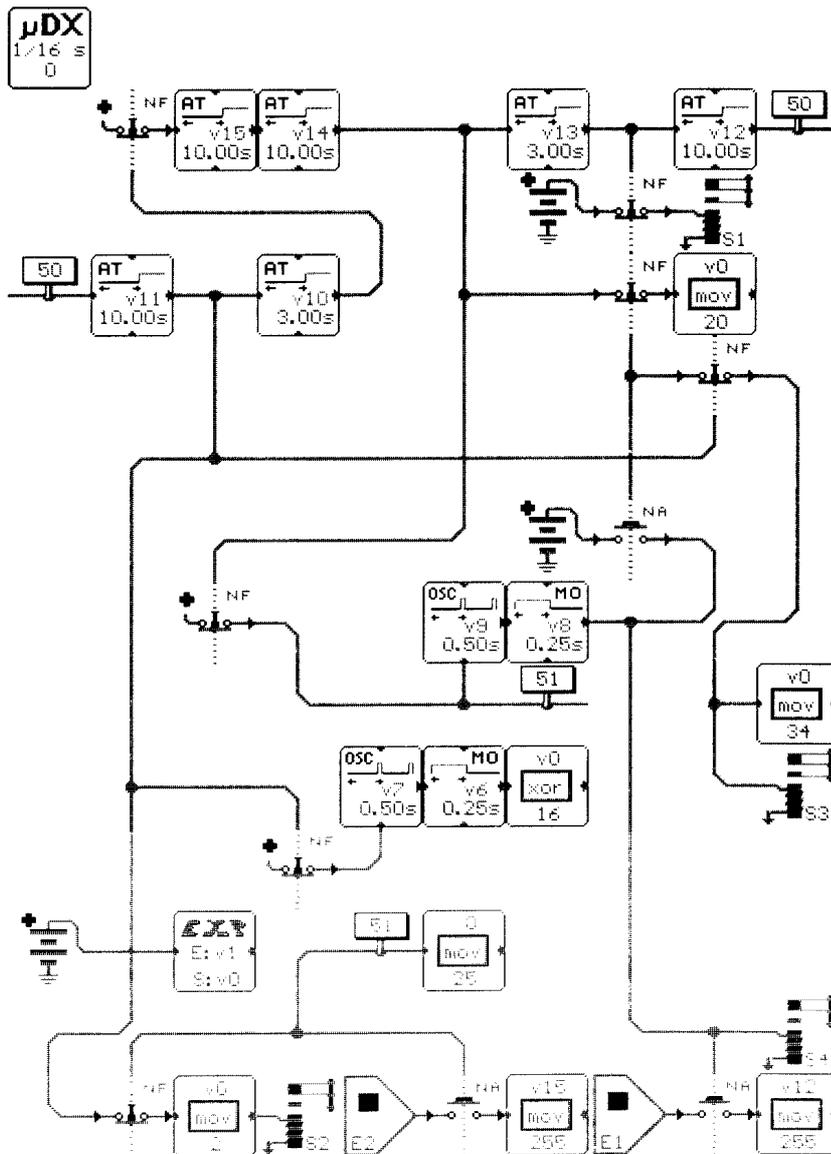


Leitura

Controlador Programável uDX Série 100

Nome do programa: SEMAFORO.UDX

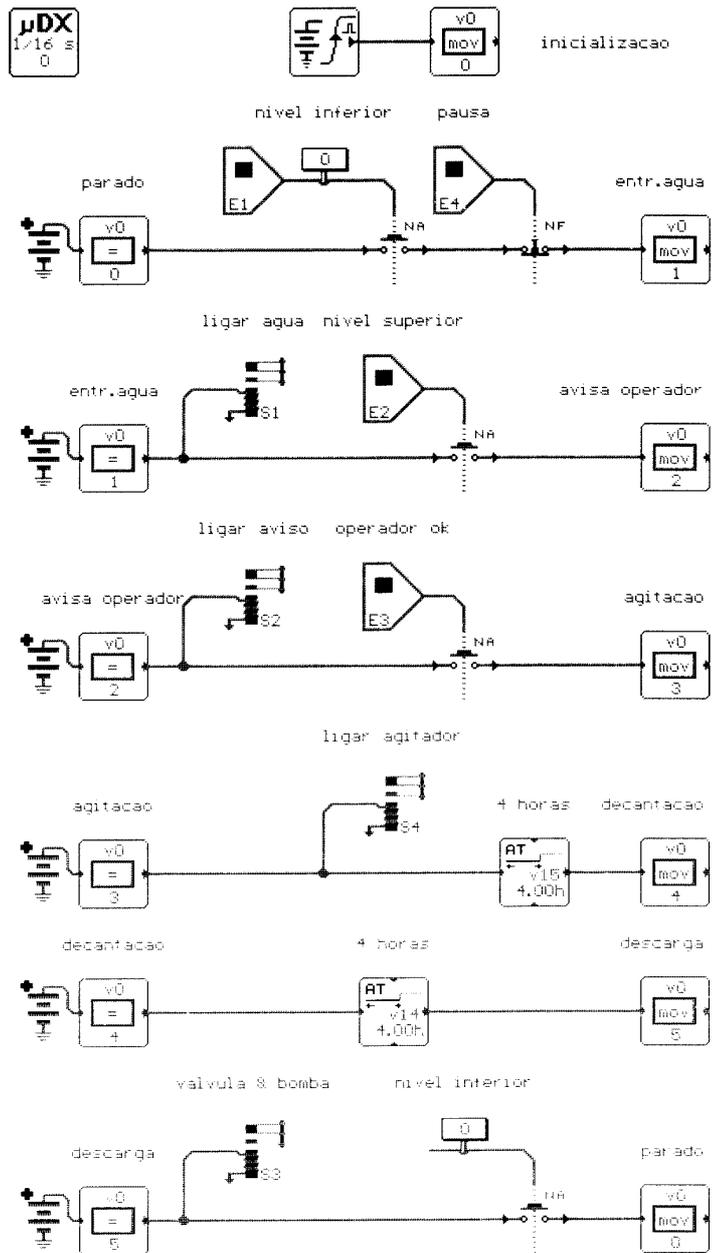
Numero de blocos para uDX utilizados: 28



Controlador Programavel uDX Serie 100

Nome do programa: RESERV.UDX

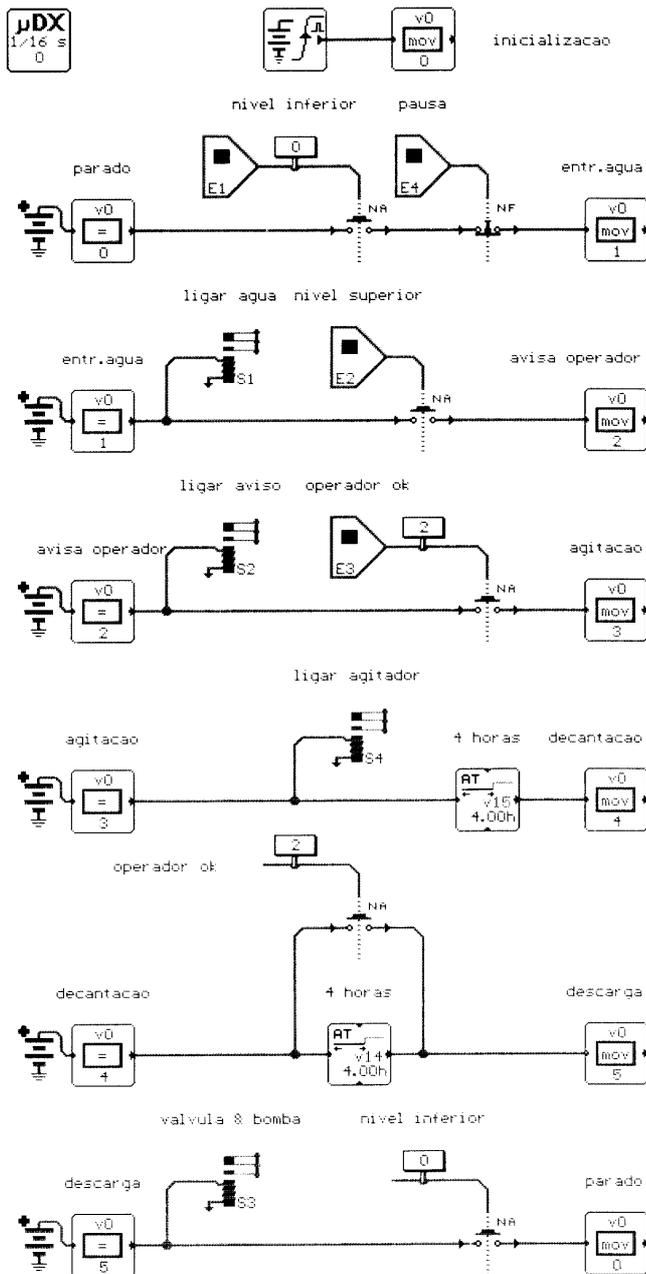
Numero de blocos para uDX utilizados: 20



Controlador Programavel uDX Serie 100

Nome do programa: RESERV1.UDX

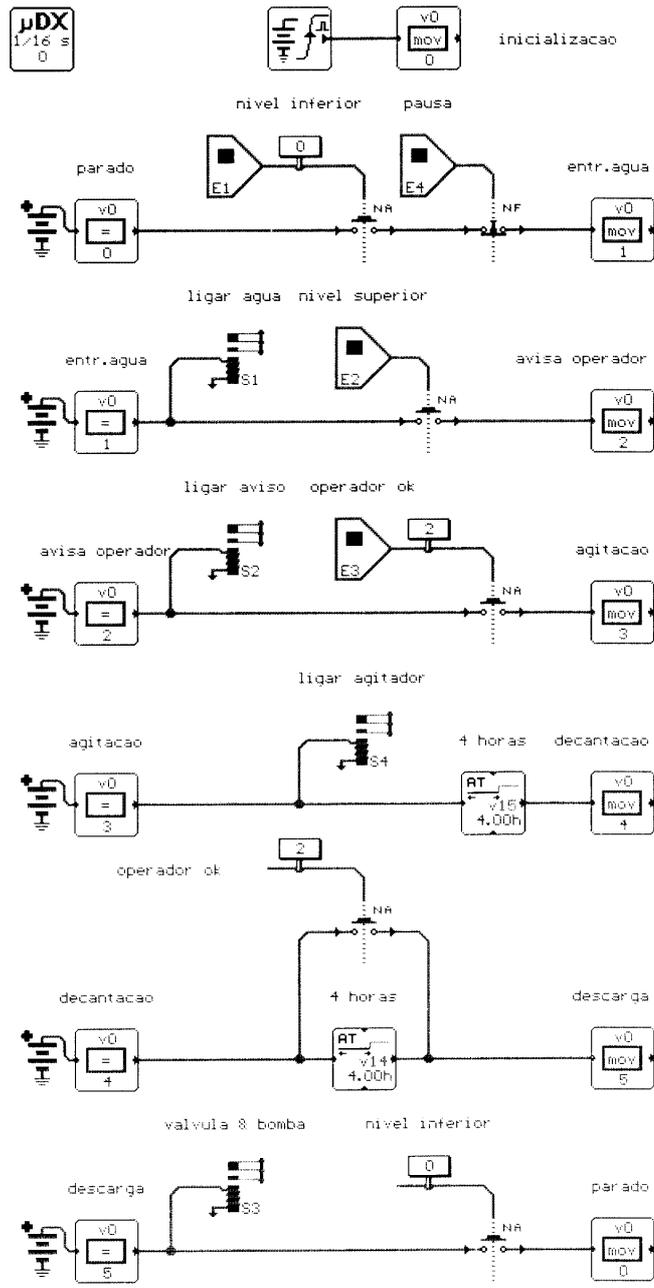
Numero de blocos para uDX utilizados: 21



Controlador Programavel uDX Serie 100

Nome do programa: RESERV1.UDX

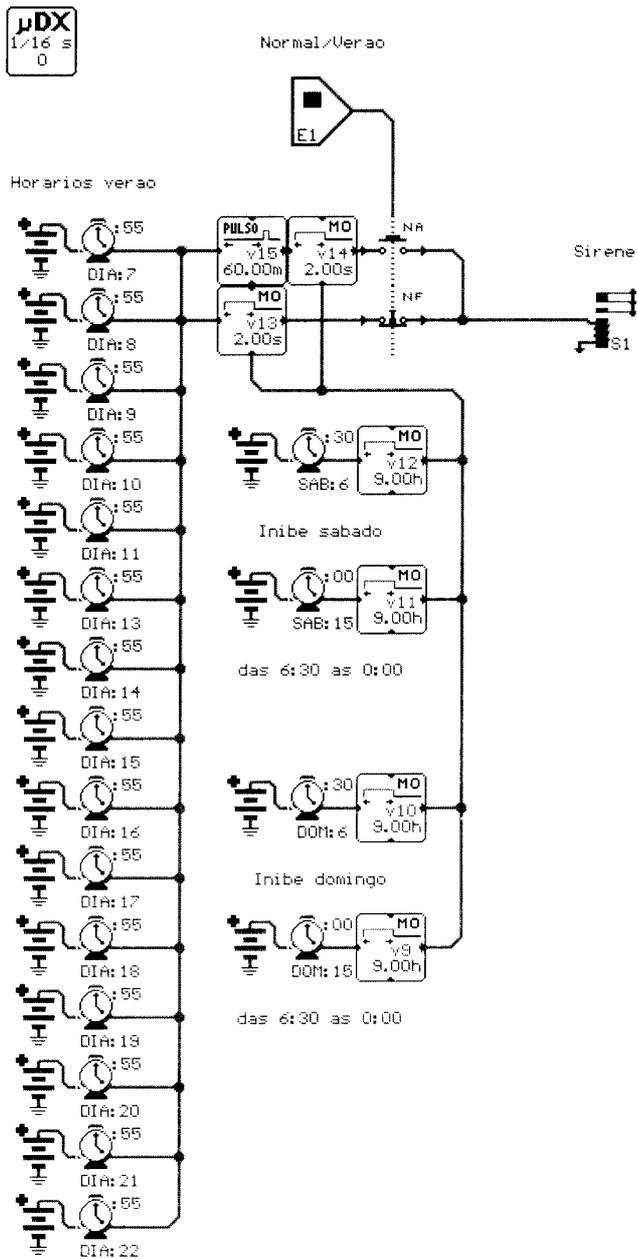
Numero de blocos para uDX utilizados: 21

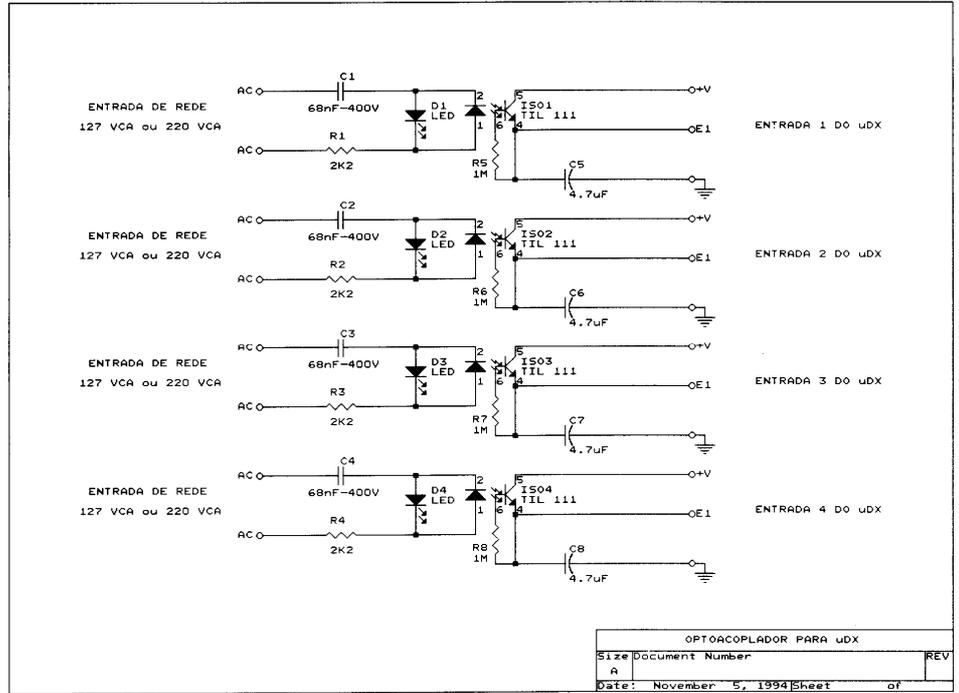


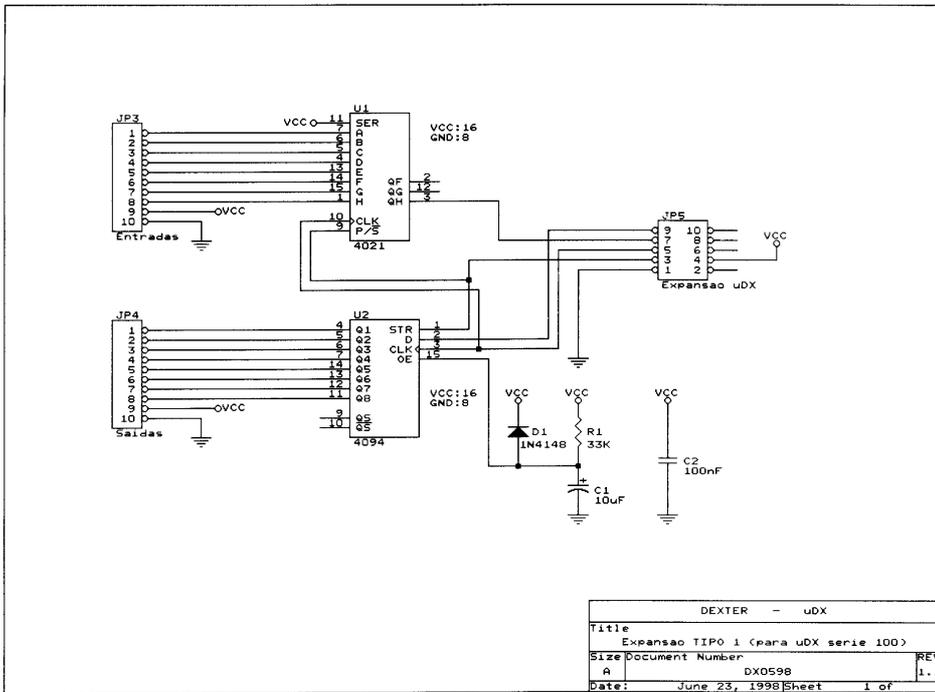
Controlador Programável uDX Série 100

Nome do programa: ESCOLA3.UDX

Numero de blocos para uDX utilizados: 28

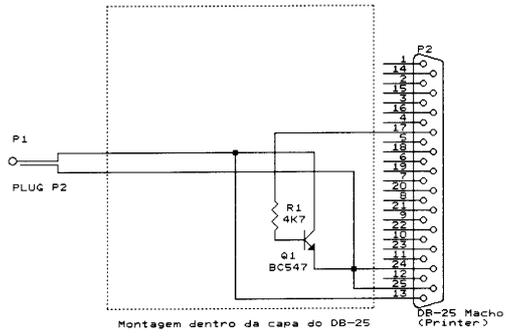




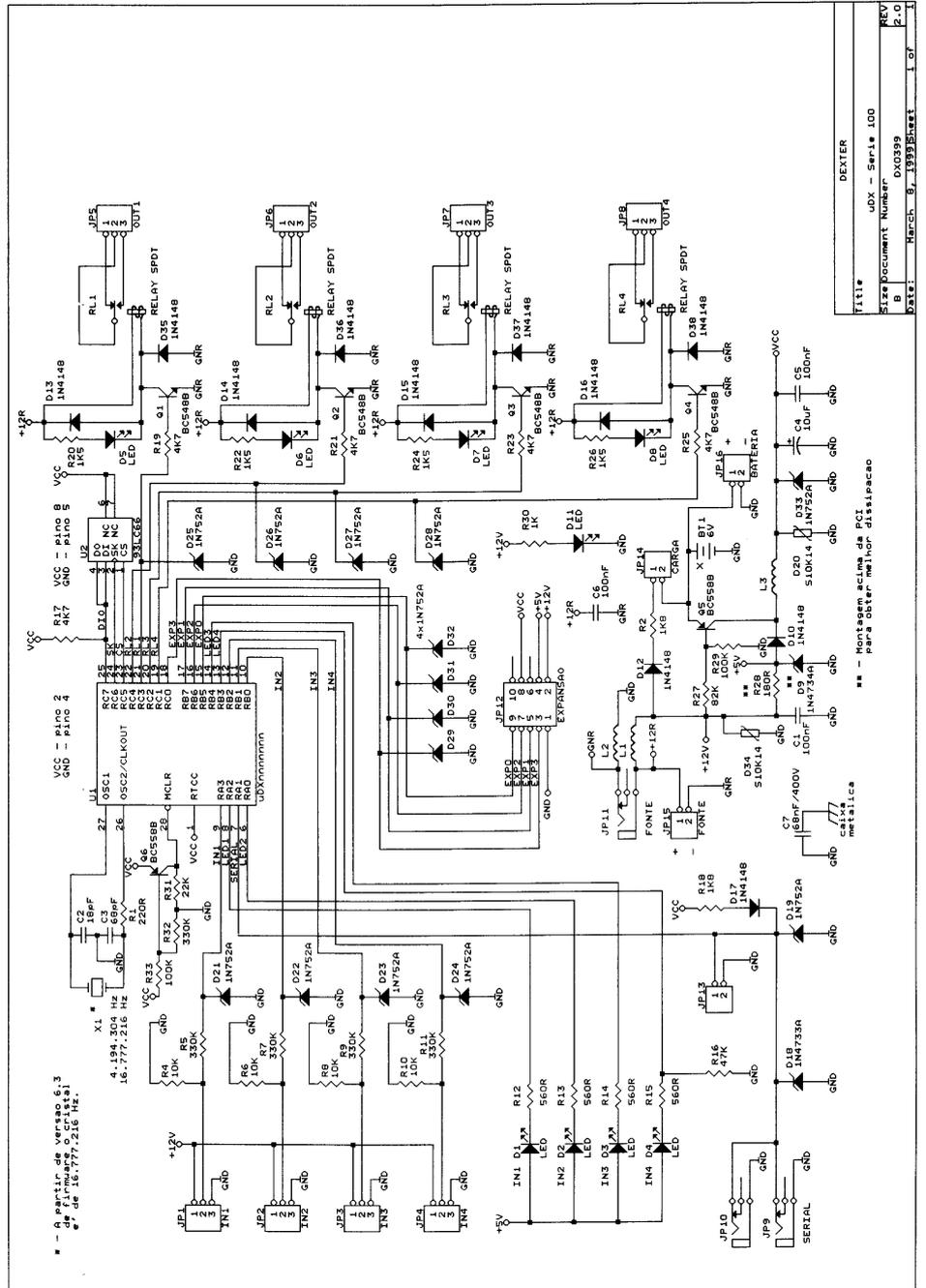




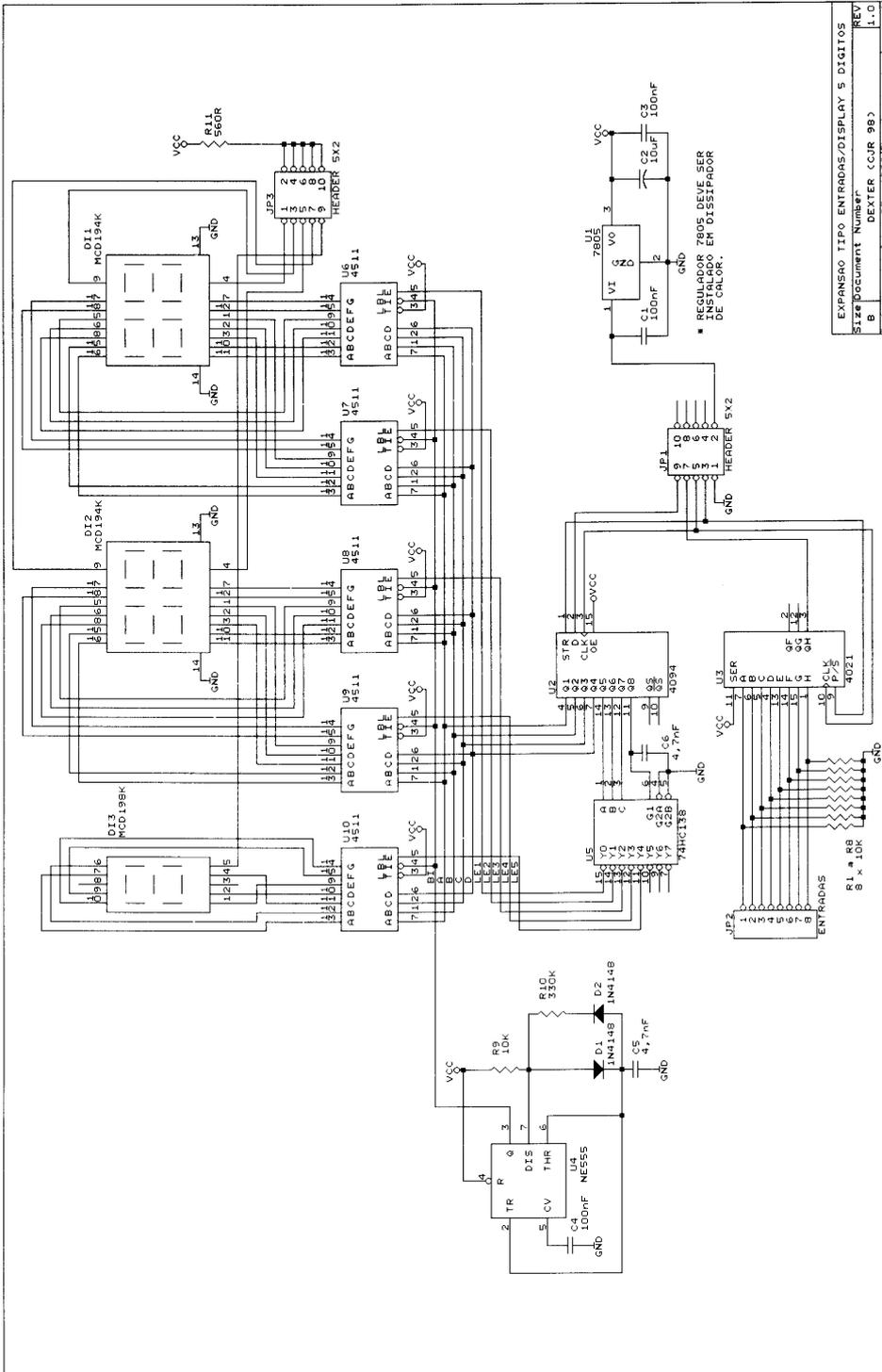
DEXTER - CJR		
Size	Document Number	REV
A	CABO uDX<>uDX	1.0
Date:	January 15, 1997	Sheet 1 of 1

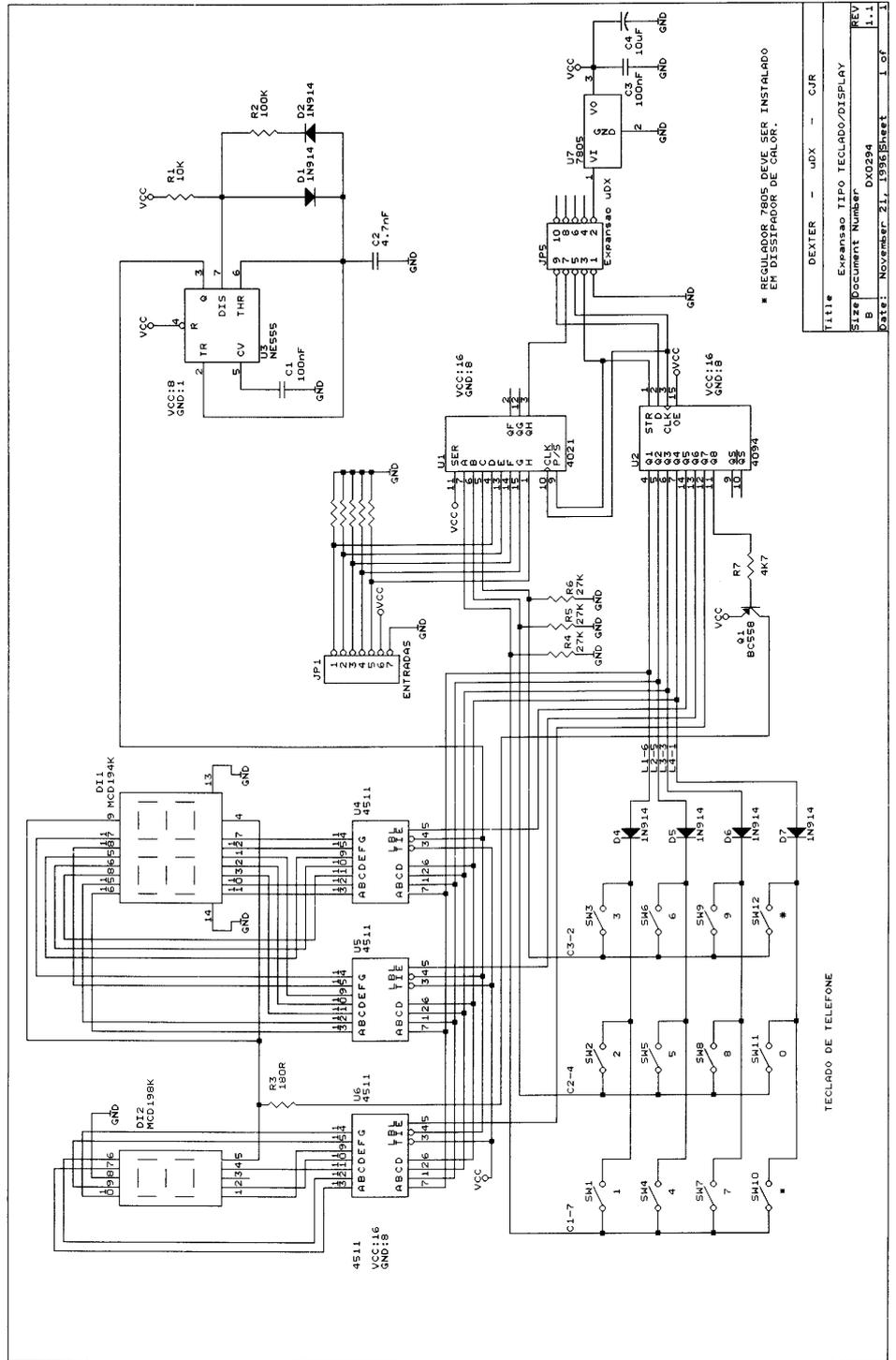


DEXTER - CJR		
Size	Document Number	REV
A	CABO PC > μDX	1.3
Date:	August 1, 1995	Sheet 1 of 1



REV	1	OF
Size	Document Number	DX0399
Date:	Mar-Ch.	9, 1993
Title		
DEXTER		





TITULO	DEXTER - µDX - CJR
Size	Expansao TIPO TECLADO/DISPLAY
Document Number	DX0294
REV	1.1
B	
Date:	November 21, 1996/Sheet 1 of 1