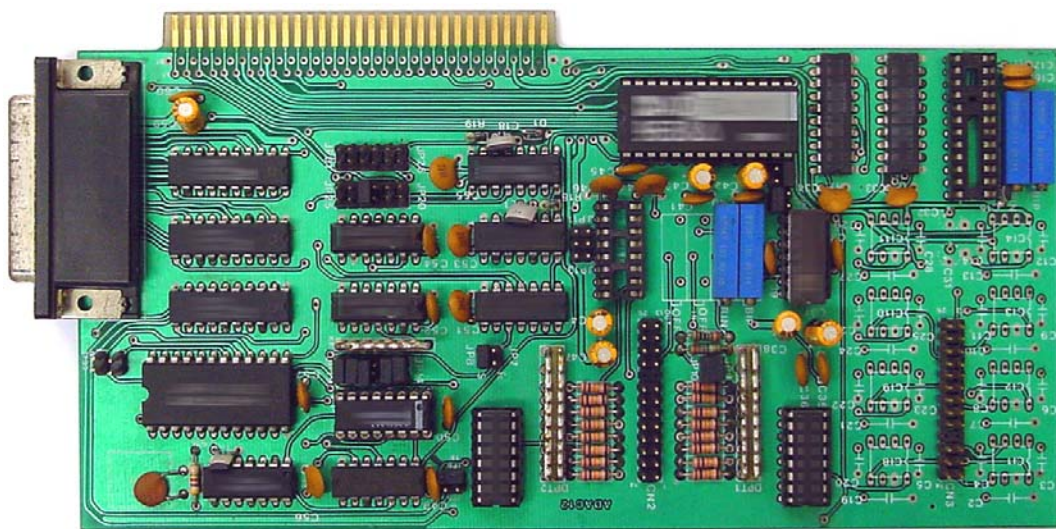


MANUAL DO USUÁRIO

VERSÃO 3.0



INTERDATA

INTERFACE DE AQUISIÇÃO E CONTROLE DE DADOS

SUMÁRIO

1 INTRODUÇÃO	01
2 CARACTERÍSTICAS DA PLACA	01
3 CONVERSOR ANALÓGICO DIGITAL	03
4 CONVERSOR DIGITAL ANALÓGICO	04
5 ENTRADAS E SAÍDAS DIGITAIS	04
6 CONECTORES DA PLACA	05
6.1 - CONECTOR DIGITAL CN1	05
6.2 - CONECTOR ANALÓGICO/DIGITAL CN2	05
6.3 - Conector Digital/Analógico CN3d	06
7 CONFIGURAÇÃO GERAL DA PLACA INTERDATA	07
7.1 - CONFIGURAÇÃO DE ENDEREÇOS DA PLACA	07
7.2 - SELEÇÃO DO NÚMERO DA INTERRUPTÃO DO MICROCOMPUTADOR A SER UTILIZADO PELA PLACA	08
7.3 - CONFIGURAÇÃO DO CONVERSOR ANALÓGICO/DIGITAL	09
8 INSTALAÇÃO DA PLACA	10
9 ESCOLHA DA FREQUÊNCIA DE AQUISIÇÃO	14
9.1 - MEDIÇÃO EM FREQUÊNCIA PROGRAMÁVEL GERADA POR INTERRUPTÃO	14
9.2 - MEDIÇÃO EM MÁXIMA FREQUÊNCIA	14
10 "SOFTWARE" DISPONÍVEL	15
DESCRIÇÃO DOS PROCEDIMENTOS IMPLEMENTADOS EM TURBO PASCAL	16
10.1 - PROG_MED_FREQ	17
10.2 - LÊ_FREQÜÊNCIA	17
10.3 - FUNÇÃO LAD	17
10.4 - ROTINA EDA	18
10.5 - FUNÇÃO LED	18
10.6 - ROTINA DSD	19
10.7 - ROTINA ASD	19
10.8 - ROTINA EXC_LAD_AF	19
10.9 - ROTINA EXC_LAD_MULTI_CANAL	21
10.10 - FUNÇÃO LAD_AF	22
10.11 - ROTINA GUARDA_AD_AF	22
10.12 - INICIA_PLACA	23
10.13 - MOSTRAMSGINICIPLACA	23
10.14 - FIM_PLACA	24
11 ROTINAS DE MANIPULAÇÃO DA PLACA	24
12 CONSIDERAÇÕES SOBRE A UTILIZAÇÃO DAS ROTINAS	25
13 CONSIDERAÇÕES SOBRE A UTILIZAÇÃO DA PLACA	25

1 INTRODUÇÃO

A placa interdata 2 foi projetada para preencher uma lacuna do mercado ainda não coberta por nenhum fabricante. Testada nas mais adversas condições de uso, como em soldagem a arco, esta placa, acompanhada de um "software" altamente potente, permite ao usuário uma elevada flexibilidade de utilização em todos os ramos científicos e tecnológicos.

2 CARACTERÍSTICAS DA PLACA

A placa com dimensões de 205 X 100 mm, possui um conector compatível com o barramento de 16 bit de microcomputadores da linha IBM PC. Ela possui as seguintes unidades:

- a) Um conversor A/D de 16 canais, com 14 dos quais plenamente disponíveis para aquisição de dados. Dois deles são utilizados com alimentação interna de 0 V e 9 V para uma autocalibração;
- b) Um conversor D/A de 8 canais;
- c) Um freqüencímetro;
- d) Uma porta digital com 8 entradas e 8 saídas.

Na figura 1 é apresentada um desenho esquemático da placa, apresentando os conectores para acesso a essas unidades, bem como os "jumpers" para adaptação de endereços, interrupções e seleção de entrada analógica unipolar ou bipolar, e os potenciômetros para ajustes de ganho. Na figura 2 é apresentada uma foto da placa.

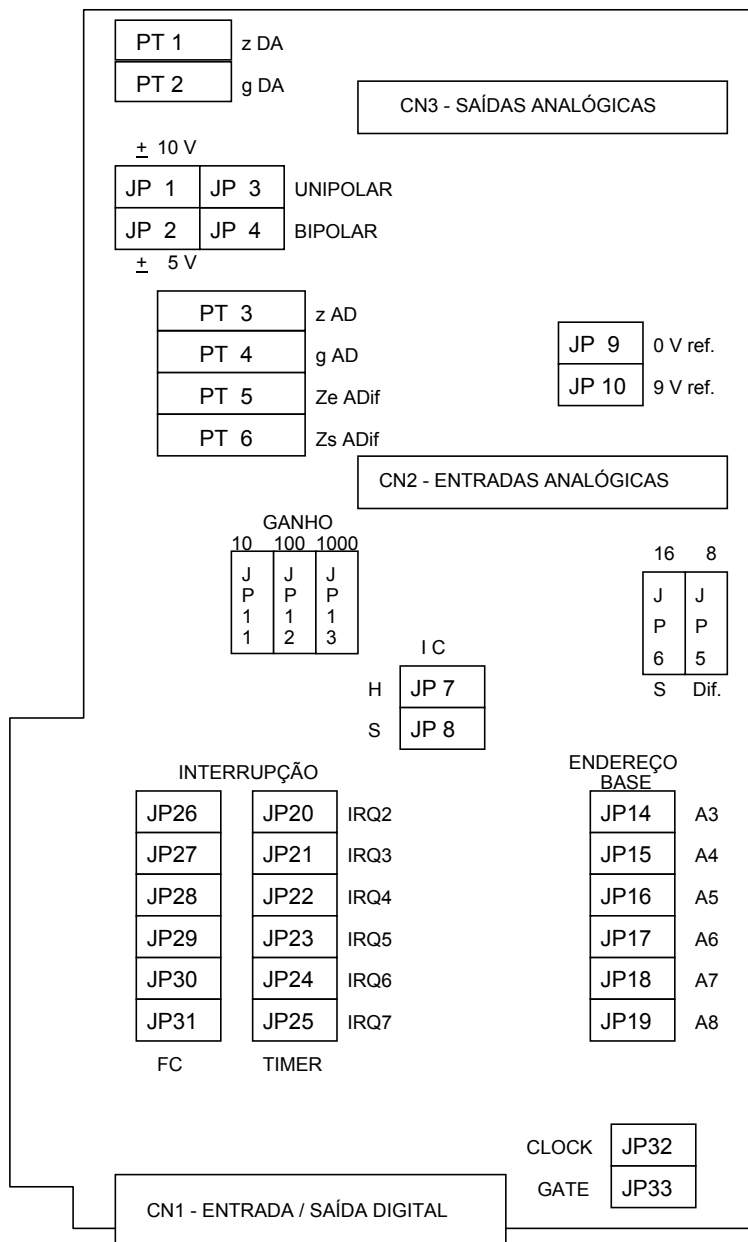


Fig. 1 - Vista Superior com a disposição dos componentes, conectores, jumpers e potenciômetros

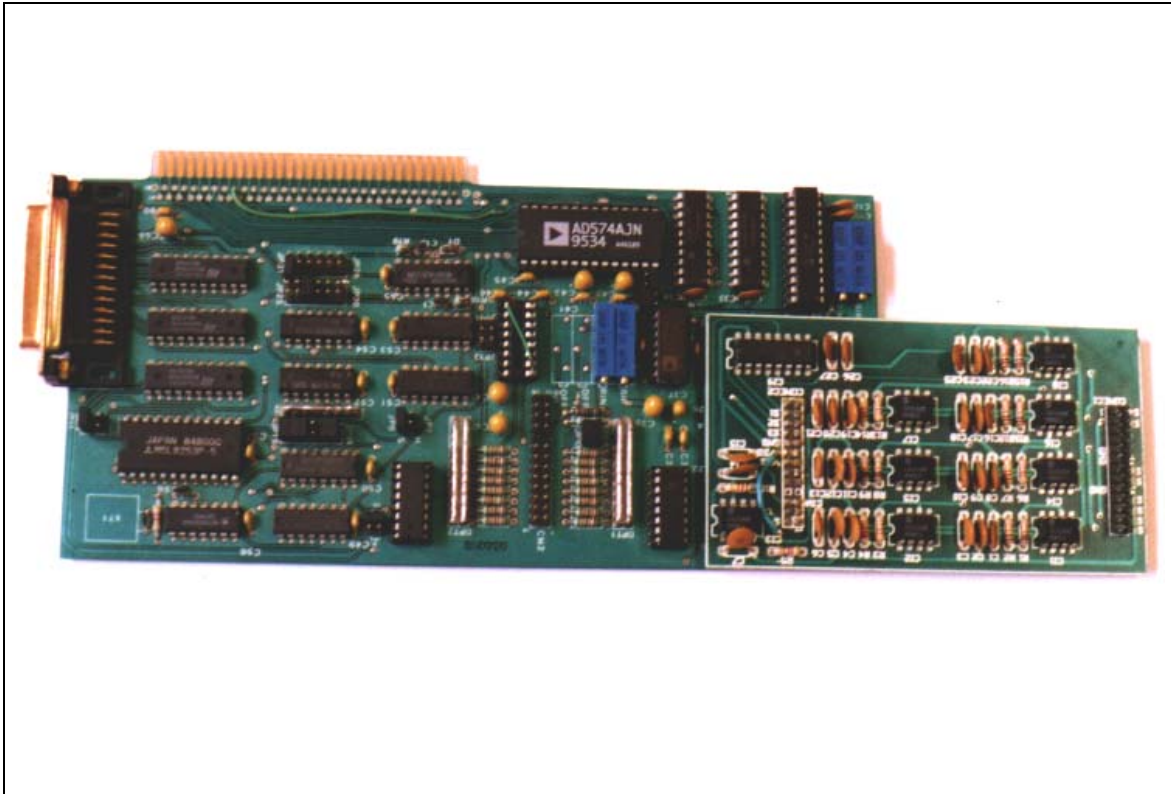


Fig. 2 - Foto da Placa de Aquisição de Dados e Comando de Equipamentos INTERDATA 2

3 CONVERSOR ANALÓGICO / DIGITAL

É utilizado um A/D rápido de 12 bit que pode medir tensões nas faixas 0-10 V e -10 a +10 V, com a seleção feita por "jumper" na placa. Podem ser feitas leituras também em 8 bit, aumentando a taxa de aquisição. A seleção de leitura em 8 ou 12 bit é realizada por "software".

Estão disponíveis sinais de referência de 9 V e 0 V nos canais 15 e 16, respectivamente. Estes sinais permitem uma auto calibração da placa, minimizando os erros gerados por variação da temperatura. Os erros causados por mudanças de características elétricas devido a envelhecimento dos componentes da placa também são atenuados.

Dependendo do computador utilização podemos obter diferentes taxas de aquisição, assim foram levantadas as frequências máximas de aquisição para alguns modelos.

a) PC 286; 12 MHz

- Aquisição em 8 bit - 40 kHz
- Aquisição em 12 bit - 26 kHz

b) PC 386; 33 Mhz

- Aquisição em 8 bits - 43 kHz
- Aquisição em 12 bits - 28 kHz

A frequência máxima por canal é aproximadamente o resultado da divisão da frequência máxima pelo número de canais lidos simultaneamente.

A frequência de aquisição pode ser programada por "software", dentro de uma determinada faixa, utilizando a base de tempo do temporizador existente na placa. O tempo de aquisição é dependente da memória disponível do microcomputador e da frequência de aquisição escolhida, o que é calculado e fornecido ao usuário pelo "software".

A impedância de entrada de cada canal é $> 10\text{ K}$.

4 CONVERSOR DIGITAL ANALÓGICO

É utilizado um conversor de 12 bit com 8 canais de saída na faixa de $\pm 5\text{ V}$ a $\pm 10\text{ V}$. A multiplexação das saídas é feita por circuitos de amostragem (Sample-hold) que devem ter seus valores periodicamente atualizados. A corrente máxima das saídas é de 5 mA. Taxa de decaimento do sinal de saída (CONSULTAR COM FABRICANTE).

5 ENTRADAS E SAÍDAS DIGITAIS

Estão disponíveis 8 entradas digitais e 8 saídas digitais, operando em nível TTL (0 a 5 V). A corrente máxima das saídas é de aproximadamente 10 mA.

6 CONECTORES DA PLACA

A placa possui 3 conectores:

O conector das entradas e saídas digitais com acesso externo ao microcomputador, o conector das entradas e o das saídas analógicas, ambos com acesso interno, através de cabos.

6.1 - CONECTOR DIGITAL CN 1.

Conector fêmea de 25 pinos para entradas e saídas digitais com a configuração conforme a tabela 1.

S I N A L	P I N O
Freqüêncímetro	02 ("jumper" 32 aberto)

Gerador de pulsos retangulares	03 ("jumper" 32 fechado)
Saída digital 1	05
Saída digital 2	07
Saída digital 3	17
Saída digital 4	19
Saída digital 5	04
Saída digital 6	06
Saída digital 7	16
Saída digital 8	18
Entrada digital 1	09
Entrada digital 2	11
Entrada digital 3	23
Entrada digital 4	21
Entrada digital 5	10
Entrada digital 6	08
Entrada digital 7	20
Entrada digital 8	22
GND	13,25

Tabela 1 - Configuração do conector digital CNI

Para uso da placa como freqüencímetro (pino 02) ou como gerador de pulso (pino 03), é necessário o posicionamento correto do "jumper" 32 na placa. Para funcionamento como freqüencímetro, o "jumper" 32 deve estar aberto e como gerador de pulsos, fechado.

6.2 - CONECTOR A/D CN2

Conector macho de 24 pinos, com acesso interno ao microcomputador, para entradas analógicas. Ele já vem conectado, por meio de um cabo paralelo de 25 vias, a um conector DB 25, externamente ao microcomputador, cuja configuração é apresentada na tabela 2.

SINAL	PINO
Entrada analógica 1	14
Entrada analógica 2	15
Entrada analógica 3	16
Entrada analógica 4	17
Entrada analógica 5	18
Entrada analógica 6	19
Entrada analógica 7	20
Entrada analógica 8	21
Entrada analógica 9	01
Entrada analógica 10	02
Entrada analógica 11	03
Entrada analógica 12	04
Entrada analógica 13	05
Entrada analógica 14	06
Entrada analógica 15 (ref. 9.0 Volts)	07
Entrada analógica 16 (ref. 0.0 Volts)	08
GND	11,12,24,25
Guia do conector	9,22

Tabela 2 - Configuração do conector analógico - CN2

6.3 - CONECTOR D/A - CN3

Conector macho de 24 pinos com acesso interno ao microcomputador para saídas analógicas. Ele já vem conectado, por meio de um cabo paralelo de 25 vias, a um conector DB 25, externamente ao microcomputador, cuja configuração é apresentada na tabela 3.

SINAL	PINO
Saída analógica 1	12, 25
Saída analógica 2	11, 24
Saída analógica 3	10, 23
Saída analógica 4	01, 14
Saída analógica 5	08, 21
Saída analógica 6	13
Saída analógica 7	02, 20
Saída analógica 8	07, 20
GND	5, 6, 9, 18, 19, 22
Guia do conector	04, 17

Tabela 3 - Configuração do conector D/A - CN3

7 CONFIGURAÇÃO GERAL DA PLACA INTERDATA 2

Foram previstos na placa INTERDATA 2 oito conjuntos de pinagens com possibilidade de "jampemento" a fim de configurá-la à diversas necessidades de aplicações e adaptação ao microcomputador.

7.1 - CONFIGURAÇÃO DE ENDEREÇO DA PLACA

Cada placa conectada ao PC tem um endereço específico de acesso. A placa INTERDATA 2 está configurada por meio de "jumpers" para trabalhar em um dos endereços especificados na tabela 4.

ENDEREÇOS	Situação dos "Jumpers"					
	JP19	JP18	1P17	JP16	JP15	JP14
200	F	F	F	F	F	F
220	F	F	F	A	F	F
240	F	F	A	F	F	F
250	F	F	A	F	A	F
260	F	F	A	A	F	F
270	F	F	A	A	A	F

280	F	A	F	F	F	F
2F0	F	A	A	A	A	F
3E0	A	A	A	A	F	F
3F0	A	A	A	A	A	F

Tabela 4 - Situação dos "jumpers" para seleção de endereço (F - fechado, A - aberto).

O usuário deve escolher um endereço ainda não utilizado pelos periféricos instalados no micro (controladores de disco, interface GPIB, etc).

Caso o endereço selecionado já estiver sendo utilizado por um desses periféricos, aparecerá uma mensagem de erro ao se instalar a placa, através do programa instala. Neste caso, deve-se alterar o endereço na placa, mudando-se a posição dos jumpers, conforme a referida tabela. Após isto, deve-se executar novamente o programa instala.

7.2- SELEÇÃO DO NÚMERO DA INTERRUPTÃO DO MICROCOMPUTADOR A SER UTILIZADA PELA PLACA

Esta seleção deve ser feita com critério, a fim de que não se selecione uma já utilizada pelo sistema. As interrupções 2 e 6 não podem ser utilizadas de nenhuma maneira, pois são usadas pela configuração mínima de um microcomputador. Resta as interrupções 3, 4, 5 e 7, dependendo dos periféricos associados ao microcomputador. A seleção de uma delas deve ser feita observando-se o seguinte:

IRQ3 - Quando não for utilizada a saída serial N°2 (COM 2);

IRQ4 - Quando não for utilizada a saída serial n° 1 (COM 1);

IRQ5- Quando não for utilizada a saída paralela LPT2. Esta é a interrupção mais provável de estar livre, pois é destinada a utilização de uma segunda impressora;

IRQ7 - Quando não for utilizada pela a saída paralela LPT1.

A placa possui duas fontes para gerar o pulso para produzir a interrupção no microcomputador, uma é através do relógio interno (timer) e a outra pelo próprio conversor A/D, quando este termina cada conversão.

A interrupção gerada pelo "timer" foi projetada para ser selecionada pelos "jumpers" 20 a 25 e a gerada pelo fim de conversão (FC), foi projetada para ser selecionada pelos "jumpers" 26 a 31.

A versão já disponível do "software" de instalação da placa foi desenvolvida para a interrupção gerada pelo "timer"e, portanto, os "jumpers" 20 a 25 devem ser configurados conforme a tabela 5.

IRQ	JP 20	JP 21	JP 22	JP 23	JP 24	JP 25
2	F	A	A	A	A	A
3	A	F	A	A	A	A
4	A	A	F	A	A	A
5	A	A	A	F	A	A
6	A	A	A	A	F	A
7	A	A	A	A	A	F

Tabela 5 - Situação dos "jumpers" para seleção da interrupção

(F-fechado, A-Aberto).

7.3 - CONFIGURAÇÃO DO CONVERSOR A/D

O conversor A/D pode ser configurado para trabalhar bipolarmente (sinal alternado) e unipolarmente.

Para aquisição de sinais que não mudam de polaridade é mais vantajoso se trabalhar com a placa em modo unipolar para se conseguir melhor resolução de medição. Neste caso, o "jumper" JP3 deve ser usado. Para sinais bipolares deve-se usar o "jumper" JP4.

A excursão do sinal de entrada, ou seja, a faixa de variação do sinal a ser medido, pode ser modificada através dos "jumpers" 1 e 2.

Assim, em combinação com os "jumpers" 3 e 4 tem-se as seguintes possibilidades:

-JP1 e JP3 fechados - situação não recomendada, uma vez que, se teria a excursão do sinal de 0 a 20 V e como o limite do sinal a ser convertido pelo A/D é de 10 V, ter-se-ia uma perda de resolução, pois, por exemplo, em conversão em 12 bit se teria a faixa de 0 a 4095 distribuída entre 0 e 20 V e o que se aproveitaria seria a faixa de 0 a 2047.

-JP1 e JP4 fechados - excursão do sinal entre -10 e +10 V.

-JP2 e JP3 fechados - excursão do sinal entre 0 e +10 V.

-JP2 e JP4 fechados - excursão do sinal entre -5 e +5 V.

Os "jumpers" JP9 e JP10 são utilizados para colocar sinais de referência nos canais 15 e 16 com a finalidade de, através do "software" utilizado, se ter condições de se avaliar o estado do conversor A/D.

Para os casos de excursão do sinal entre 0 a 10 e -10 a +10 V, o sinal de referência utilizado é de 9 V para o canal 15 e 0 V para o canal 16, que é o caso padrão de fornecimento da placa.

Para o caso da excursão do sinal entre -5 e +5 V, a referência do canal 15 deve ser rebaixada para não saturar o conversor A/D. Para tanto, deve-se consultar o fornecedor.

Os "jumpers" JP7 e JP8 são utilizados para se definir se o início de conversão será por "software" ou por "hardware". Para o presente "software", será ele o encarregado disto, o que, para tanto, deve ser utilizado o "jumper" JP8, deixando-se a ligação de JP7 em aberto.

Se a placa for utilizada com entradas analógicas simples, o "jumper" JP6 deve ser usado. Caso se pretenda utilizar a placa com entradas diferenciais, deve-se usar o "jumper" JP5, desfazer-se o curto-circuitamente existente entre os pinos 1,2 e 9 do soquete do conversor diferencial, colocando este último no referido soquete, e escolher o ganho pretendido, atuando-se nos "jumpers" 11, 12 e 13. Sem nenhum desses "jumpers", o ganho é 1. Com o "jumper" 11, o ganho é 10, com o 12, o ganho é 100, e com o 13, o ganho é 1000.

8 INSTALAÇÃO DA PLACA

Para instalação da placa basta colocá-la em um conector disponível do barramento de um microcomputador da linha IBM PC.

Após a ligação da placa, deve ser rodado o programa ITD_INST.EXE, (instalador da placa) o qual solicita o endereço da placa, como mostrado na figura abaixo:

Esta tela irá solicitar a você o endereço de memória em hexadecimal que está configurado pelos jump's JP14 a JP19.

Inicialmente a sua placa vem configurada para trabalhar no endereço de dispositivo "220", e se não existir outro periférico instalado no seu microcomputador neste endereço não é necessário alterar os jumps.

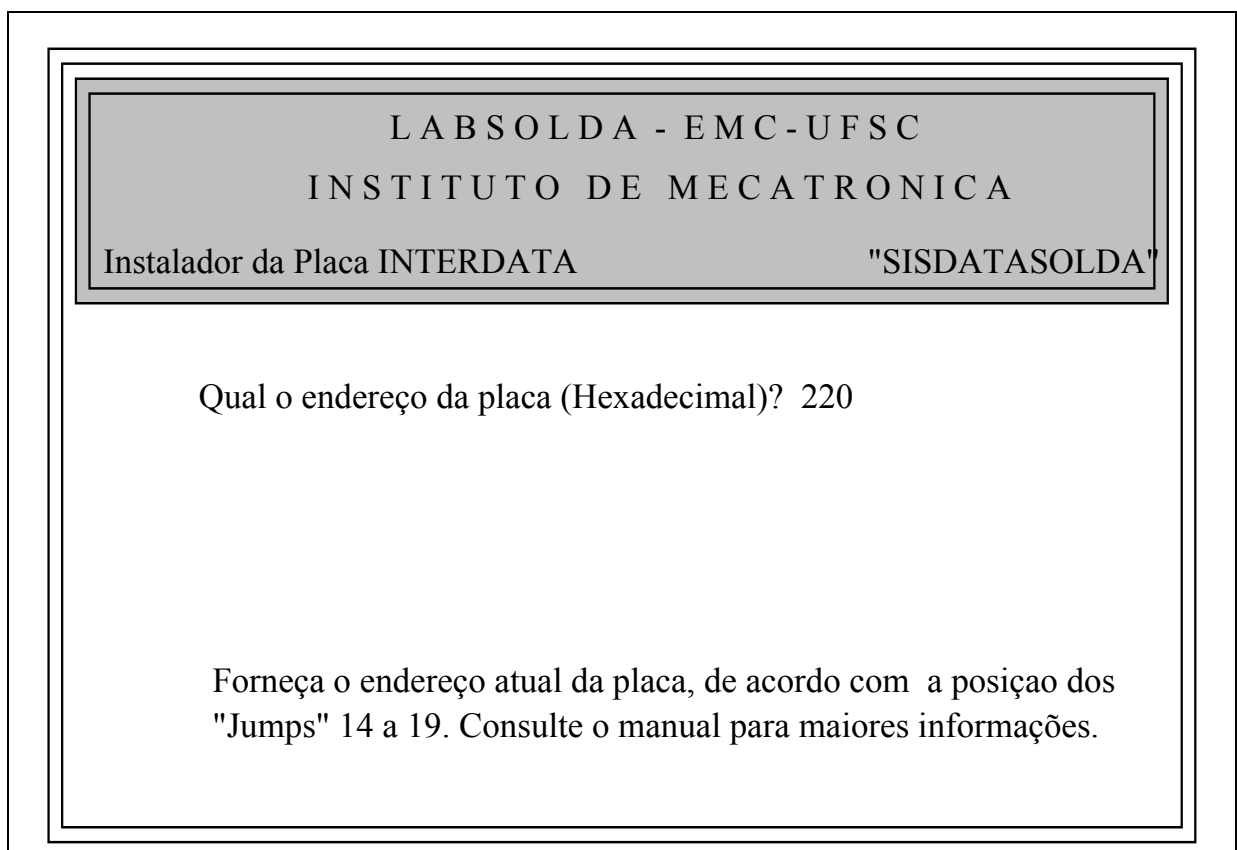


Fig. 3 – Tela para configuração do endereço de memória hexadecimal

Após você digitar o endereço tecele [ENTER], então uma nova tela aparecerá, solicitando o número da interrupção utilizada na mesma como na figura a seguir:

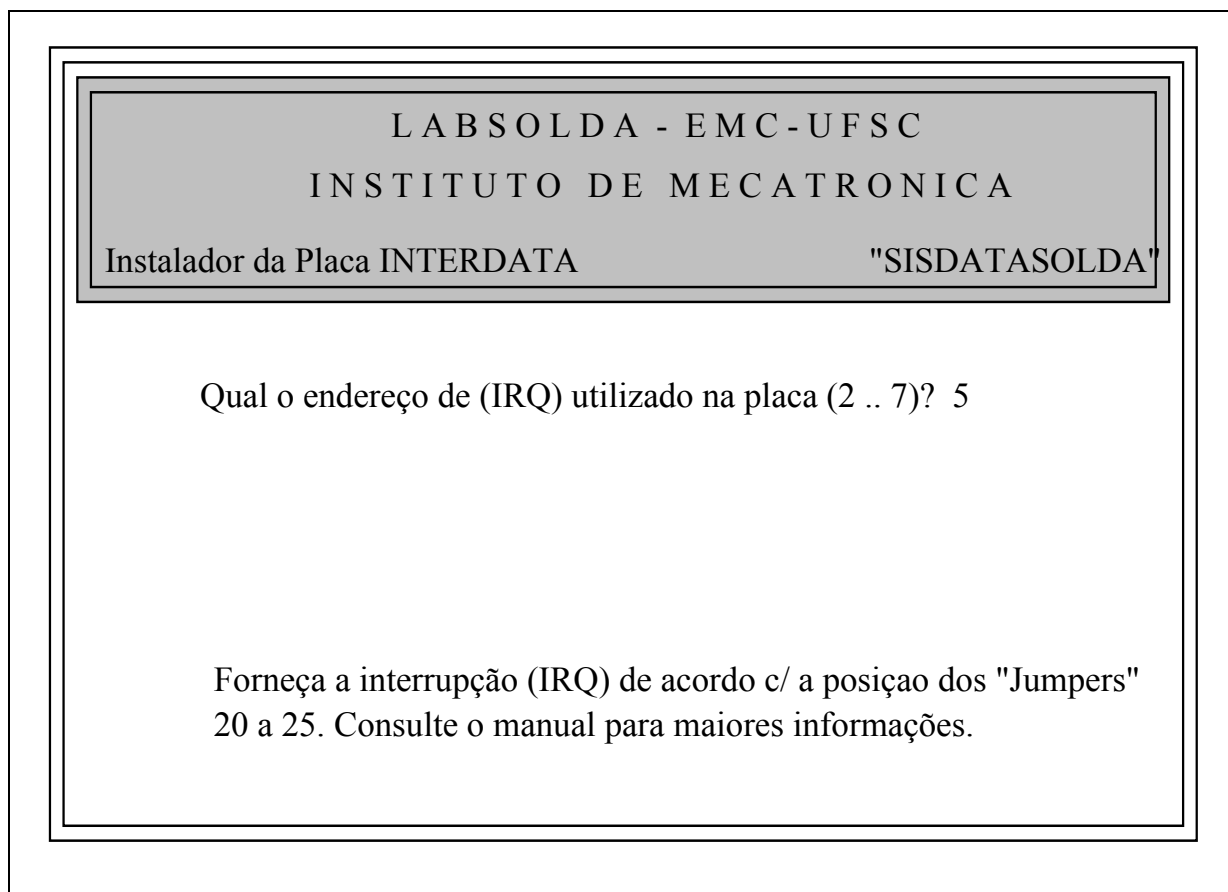


Fig. 4 – Tela para configuração do número de interrupção

Neste momento o sistema está solicitando o número da interrupção de hardware sobre o qual a placa irá trabalhar.

Inicialmente a sua placa vem configurada para trabalhar com a interrupção número 5 (cinco), e se não existir outro periférico instalado no seu microcomputador que utilize esta interrupção não será necessário alterar os jumps.

Após digitar o número "5" tecle [ENTER].

O sistema começará testando os contadores internos da placa, posteriormente testará o conversor, as interrupções da placa e então mostrará na tela o tempo de conversão que a sua configuração de hardware permite.

O tempo de uma conversão está relacionado com o tipo de conversor A/D (circuito integrado) que é utilizado na sua placa, e este tempo pode ser alterado caso você troque o conversor A/D.

Assim, na tela aparecerá:

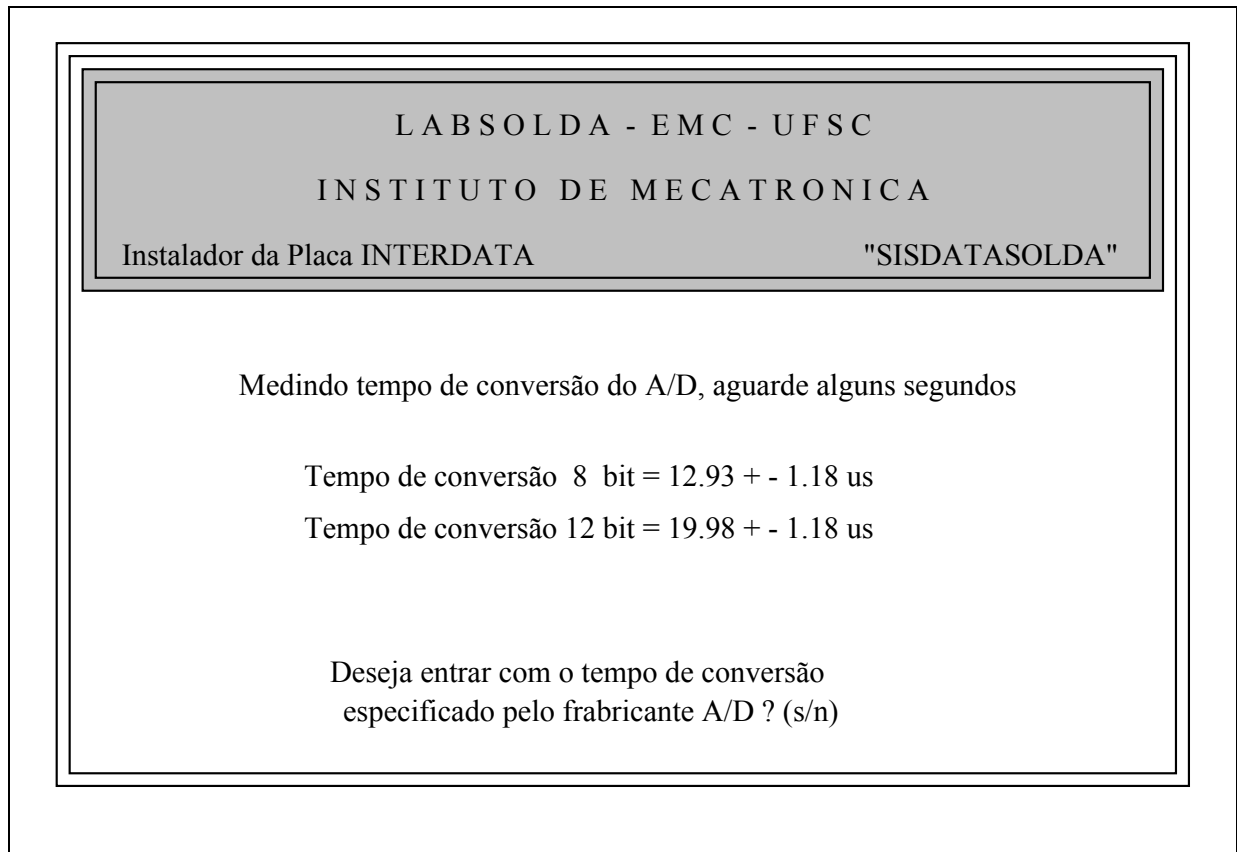


Fig. 5 – Tela para a configuração do tempo de converção A/D

Como a sua placa já vem configurada corretamente você deve digitar [n]. Assim o processo de instalação continuará, fazendo uma medida da máxima frequência de aquisição do seu sistema. Cabe ressaltar que o tempo de conversão está relacionado com o tipo do conversor que você utiliza. Ao final do processo de instalação na tela aparecerá:

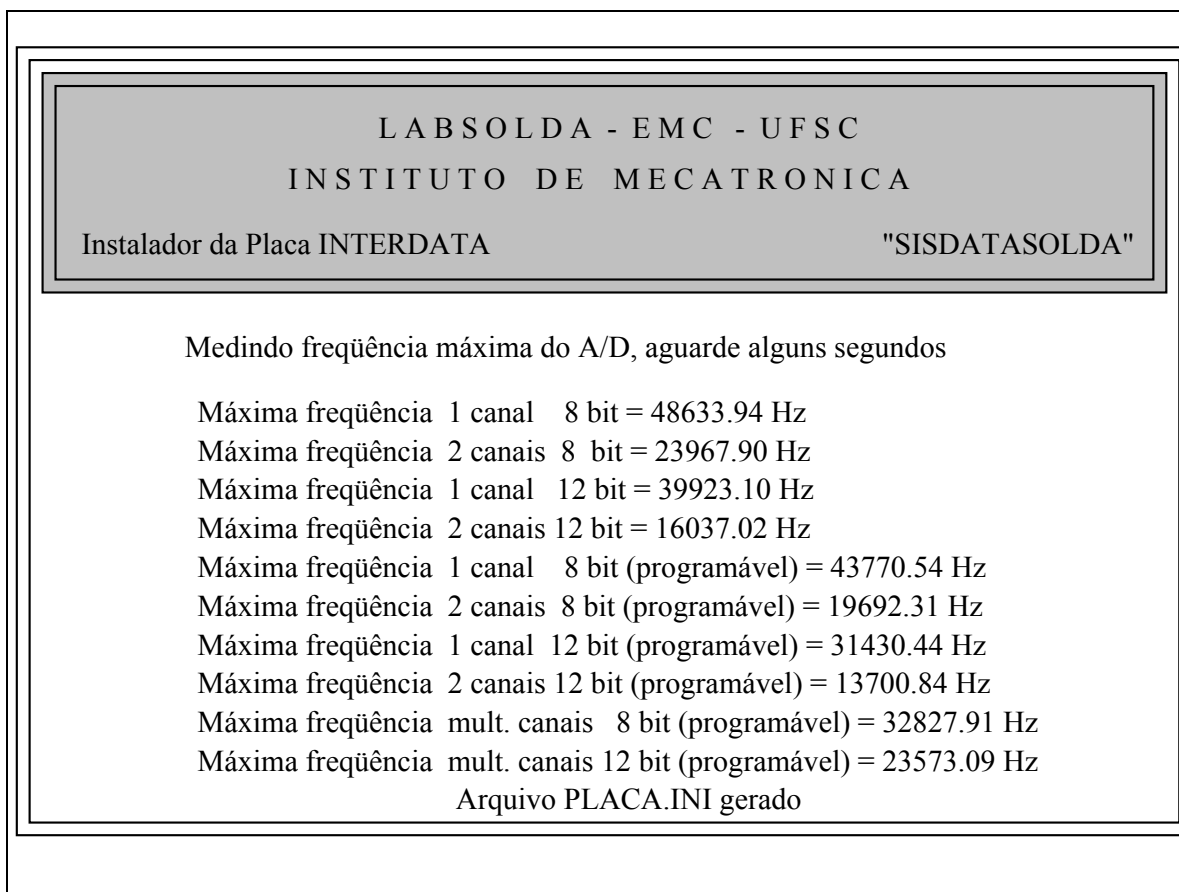


Fig. 6 –Tela final com as frequências de aquisição do sistema.

O tempo de aquisição máximo, está relacionado com o tipo de equipamento que você utiliza, por isto, estas frequências são típicas de um equipamento 386 DX de 40 MHz.

Se a placa estiver funcionando corretamente, o programa gera o arquivo PLACA.INI que contém informações que serão utilizadas pelas rotinas de aquisição e controle da placa.

9 - ESCOLHA DA FREQUÊNCIA DE AQUISIÇÃO

Para aquisição de dados analógicos em alta frequência podem ser lidos 1, 2 ou mais canais em 8 ou 12 bit. Podem ser utilizados duas formas de medição:

9.1 - MEDIÇÃO EM FREQUÊNCIA PROGRAMÁVEL GERADA POR INTERRUPÇÃO

A frequência programável pode ser programada na faixa de 1 Hz a X kHz, onde X depende do número de canais lidos, do número de bit, da frequência com que o micro pode ser interrompido e do tempo de conversão. A frequência X é fornecida pelo "software" de instalação conforme mostrado na figura anterior.

9.2 - MEDIÇÃO EM MÁXIMA FREQUÊNCIA

A frequência máxima depende do número de canais lidos, do número de bit, da frequência de clock do micro utilizado e do tempo de conversão, pois esta é feita por "pulling".

A seguir é mostrada uma tabela que contém o nome de variáveis globais das rotinas de aquisição que definem as frequências máximas, calculadas em função do "clock" do micro em que a placa estiver instalada:

Condição de leitura	Limite de frequência programável	Frequência máxima (não programável)
1 canal 8 bit	MED_FREQ8_1CN	MAX_FREQ8_1CN
1 canal 12 bit	MED_FREQ12_1CN	MAX_FREQ12_1CN
2 canais 8 bit	MED_FREQ8_2CN	MAX_FREQ8_2CN
2 canais 12 bit	MED_FREQ12_2CN	MAX_FREQ12_2CN

O valor destas variáveis é calculado pelo programa ITD_INST.EXE e os dados obtidos são gravados no arquivo PLACA.INI.

Devido ao método indireto utilizado, para o cálculo das frequências, que utiliza o relógio do PC como base de tempo (a precisão do relógio é de apenas 0.055 s), o valor medido para a frequência máxima de leitura apresenta um erro de até 0,5%.

10 "SOFTWARE" DISPONÍVEL

O "software" para controle da placa foi desenvolvido em TURBO PASCAL e as rotinas de aquisição em alta frequência foram desenvolvidas em ASSEMBLY.

Estão disponíveis os seguintes arquivos:

ITD_VAR.TPU - Arquivo de definição de variáveis.

ITD_FU40.TPU - Arquivo de rotinas básicas de aquisição.

ITD_BA40.TPU - Arquivo de rotinas para controle da placa.

PLACAIN.TBJ - Arquivo de rotinas de interrupção em assembly.

PLACAMAC.OBJ - Arquivo de rotinas de aquisição em assembly.

ITD_TEST.PAS - Arquivo exemplo de utilização das rotinas de controle da placa.

INST_40.EXE - Arquivo executável para instalação da placa.

Como já foi descrito, o programa INST_40 é utilizado para instalação e calibração da placa e atua de forma automática, não sendo necessária descrição detalhada de suas rotinas e funções.

O arquivo ITD_BA40.TPU contém todas as rotinas que serão utilizadas pelo usuário da placa no desenvolvimento de seus programas.

O arquivo ITD_VAR.TPU contém definições de variáveis globais que serão utilizadas pelo usuário da placa no desenvolvimento de seus programas.

O arquivo ITD_TETS.PAS é um exemplo de aplicação da placa e pode ser utilizado pelo usuário da placa como base para desenvolvimento de seus programas.

Os demais arquivos contém definições internas que não são utilizadas pelo usuário da placa.

DESCRIÇÃO DOS PROCEDIMENTOS IMPLEMENTADOS EM PASCAL

10.1 - PROG_MED_FREQ.

Descrição : Programa o CTC para medir frequência

FORMATO DE DEFINIÇÃO

procedure PROG_MED_FREQ;

Param. entrada : tempo_leitura - Tempo em que serão contados pulsos para medição da frequência. Pode variar de 0.05 a 100 segundos .

Param. saída : nenhum .

Obs : Quanto maior o tempo, maior a precisão da medida , porém o tempo máximo é dado por :

$\text{tempo_máximo} = 40000 / \text{máxima_frequência} ;$

A resolução da medida é dada por :

$\text{resolução} = 1 / \text{tempo_leitura} ;$

Portanto, para um tempo de 0.05 segundos, a máxima frequência que pode ser medida é 800 kHz ;

Exemplo de utilização :

Considerando que se deseja medir um sinal de frequência máxima igual a 1 kHz , com resolução de 0.1 Hz :
tempo_máximo = 40000/1000 = 40 s
tempo_leitura = 1 / 0.1 = 10 s
Utiliza-se o menor valor :

Prog_med_freq(10);

Considerando que se deseja medir um sinal de frequência máxima igual a 10 kHz , com resolução de 0.1 Hz :
tempo_máximo = 60000/10000 = 6 s
tempo_leitura = 1 / 0.1 = 10 s
Utiliza-se o menor valor :

Prog_med_freq(4);

10.2 - LE_FREQÜENCIA

Descrição : Esta função mede a frequência do sinal aplicado no contador.
Ela só é válida se for anteriormente chamada a função PROG-MED-FREG.

FORMATO DE DEFINIÇÃO:

Function Lê_frequência;

Param. entrada : Nenhum

Param. saída : valor da frequência em Hz . Se não existir valor disponível será enviado o valor -1 .

Exemplo de utilização :

```
X := Lê_frequência ;  
if X <> -1 then  
  write(' Valor medido = ',X)  
else  
  write(' Em medição');
```

10.3 - FUNÇÃO LAD

Descrição : Esta função lê um canal do conversor analógico digital em 12 bit.

Formato de definição:

Function LAD (canal: byte; número_leituras: integer): integer;

Param entrada : Canal = Número do canal a ser lido (de 1 até 16).

Número_leituras = Número de leituras das quais será feita uma média para obter o valor da função. Valor mínimo = 1, valor típico = 20.

Param saída : valor lido na faixa de 0 a 4095.

Exemplo de utilização: Placa config. para ler em $\pm 10V$

$X = LAD(2, 1)$; 1 leitura do canal 2

$X = LAD(3, 10)$; Média de 10 leituras do canal 3

O valor lido é dado pela seguinte fórmula:

$X = (V + 10) \times 4095/20$

Onde:

X = Valor numérico lido

V = Tensão aplicada

10.4 - ROTINA EDA

Descrição : Esta rotina escreve em um canal do conversor D/A.

Formato de definição:

Procedure EDA (canal: byte; valor :integer);

Param entrada : Canal = Número do canal a ser escrito (1 a 8).

Valor = Valor a ser escrito (0 .. 4095).

Param saída : Nenhum.

Exemplo de utilização:

EDA (1,2000); Escreve 2000 no canal 1.

EDA (2,4000); Escreve 4000 no canal 2.

10.5 - FUNÇÃO LED

Descrição: Esta função lê uma entrada digital.

Formato de definição:

Function LED (canal : byte) :integer;

Param entrada : Canal : Número do canal a ser lido (1 .. 8);
Param saída : Valor da entrada = 0 : entrada desativada,
1 : entrada ativada.

Exemplo de utilização:

X := LED (1); Lê entrada digital 1.

10.6 ROTINA DSD

Descrição: Esta rotina desativa uma saída digital.

Formato de definição:
Procedure DSD (canal : byte);

Param entrada : Canal = Número da saída a ser desativada .
Param saída : Nenhum.

Exemplo de utilização:

DSD(1) ; Desativa saída digital número 1;

10.7 ROTINA ASD

Descrição: Esta função ativa uma saída digital.

Formato de definição:
Procedure ASD (canal: byte);

Param entrada : Canal = Número da saída a ser ativada.
Param saída : Nenhum.

Exemplo de utilização:

ASD(1); Ativa saída digital número 1;

10.8 - ROTINA EXC_LAD_AF

Descrição : Esta rotina executa leitura do conversor A/D em alta frequência.

Formato de definição:

```
procedure EXC_LAD_AF; (nbits, canalA, canalB :byte; num_leituras:longint;  
    freq_prog: real;var freq_r: real );
```

param. entrada : Nbits = 8 : Leitura em 8 bit ,

12 : Leitura em 12 bit .

CanalA = Número do primeiro canal a ser lido (1 a 16);

CanalB = Número do segundo canal a ser lido (1a16),para leitura de apenas um canal, utilizar: Canal2= 0 .

Num_leituras = Número de leituras realizadas em cada canal. A variável

MAX_MEMÓRIA indica o número máximo de bytes disponíveis na memória. Os valores máximos admissíveis são:

Leitura 1 canal 8 bit = MAX_MEMÓRIA,

Leitura 2 canais 8 bit = MAX_MEMÓRIA/2 ,

Leitura 1 canal 12 bit = MAX_MEMÓRIA/2

Leitura 2 canais 12 bit=MAX_MEMÓRIA/4

Freq-prog = Frequência programada (em Hz)

para aquisição em cada canal. Esta frequência pode ser variada de um valor mínimo (1) até um valor programável máximo ou ser fixada na frequência máxima. Os valores máximos programáveis são dados pelas variáveis:

MED_FREQ8_1CN = 1 canal em 8 bit ,

MED_FREQ8_2CN = 2 canais em 8 bit ,

MED_FREQ12_1CN = 1 canal em 12 bit ,

MED_FREQ12_2CN = 2 canais em 12 bit .

As frequências máximas são dados pela variáveis

MAX_FREQ8_1CN = 1 canal em 8 bit ,

MAX_FREQ8_2CN = 2 canais em 8 bit ,

MAX_FREQ12_1CN = 1 canal em 12 bit ,

MAX_FREQ12_2CN = 2 canais em 12 bit .

Param. saída : Freq_r = Frequência real obtida na medição.

Exemplo de utilização:

EXC_LAD_AF (8 , 1 , 0 , 1000 , 3000 , X) ;

lê o canal 1 em 8 bit , 1000 leituras em 3 kHz , em X retorna a frequência real de leitura.

EXC_LAD_AF (8 , 2 , 3 , 5000 , MAX_FREQ8_2CN , X) ;

lê os canais 2 e 3 em 8 bit , 5000 leituras por canal (Total = 10000 leituras) na frequência máxima de 8 bit 2 canais, em X retorna a frequência real de leitura.

10.9 - ROTINA EXC_LAD_MULTI_CANAL

Descrição: Esta função executa leitura do conversor A/D em alta frequência para até 14 canais.

Formato de definição:

Procedure EXC_LAD_MULTI_CANAL (nbits:byte;canais:lista_de_canais;

```
num_leituras:longint;freg_prog:real;
freg-r:real);
```

Para entrada : Nbits = 8 : Leitura em 8 bit,

12 : Leitura em 12 bit.

Canais = Lista de canais a serem lidos (1 a 16).

Num_leituras = Número de leituras realizadas em cada canal.

A variável MAX_MEMÓRIA indica o número máximo de bytes disponíveis na memória.

Valores máximos admissíveis:

Número de leitura n canais 8 bit= $MAX_MEMÓRIA / n$,

Número de leitura n canais 12 bit= $MAX_MEMÓRIA / (2*n)$

Freg_Prog = Frequência programada (em Hz) para aquisição em cada canal. Esta frequência pode ser variada de um valor mínimo (1) até o valor máximo programável.

Frequências máximas de leitura :

n canais e 8 bit = $MAX_FREQ8_MULTI_CN / n$

n canais em 12 bit = $MA_FREG12_MULTI_CN / n$

Param. saída : Freq_r = Frequência real obtida na medição.

Exemplo d utilização :

```
EXC_LAD_AF ( 8 , [1,2,3,5,6,7] , 1000 , 3000 , X );
```

lê os canais 1, 2, 3, 5, 6 e 7 em 8 bit ,

1000 leituras em 3 kHz , em X retorna a frequência real de leitura .

```
EXC_LAD_DF ( 12 , [15,14,13] , 5000 , 1000 , X );
```

lê os canais 13, 14 e 15 em 12 bit , 5000 leituras

em 1 kHz , em X retorna a frequência real de leitura.

10.10 - FUNÇÃO LAD_AF

Descrição : Esta função pega um ponto lido pelo procedimento EXC_LAD_AF.

Formato de definição:

```
function LAD_AF ( ponto:longint;canal:byte):integer;
```

Param. entrada : Ponto = Número do ponto a ser lido de 0 até o número de leituras definido no procedimento EX_LAD_AF .

Canal = Canal de leitura, igual a um dos canais definidos no procedimento EX_LAD_AF (canalA ou canalB)

Param. saída : Valor da função.

Exemplo de utilização:

X := LAD_AF (100 , 1); Pega o ponto número 100 lido no canal 1

10.11 - ROTINA GUARDA_AD_AF

Descrição: Esta rotina guarda um valor na matriz de leitura do A/D e é utilizada quando dados que foram lidos pela rotina LAD_AF e gravados em disco devem ser carregados novamente para a memória e tratados por rotinas que utilizem a função LAD_AF. Este procedimento é necessário, pois a maior parte da memória disponível no micro é alocada para armazenamento de dados adquiridos em alta frequência.

Formato de definição:

Procedure GUARDA_AD_AF (valor:integer;ponto:longint;canal, cnA,cnB: integer);

Param. entrada : Valor = Valor a ser guardado ;

Ponto = Número do ponto a ser lido de 0 até o número de leituras definido no procedimento EX_LAD_AF

Canal = Canal de leitura, igual a um dos canais definidos no procedimento EX_LAD_AF (canalA ou canalB)

CNA = Número do primeiro canal a ser lido (1 a 8)

CNB = Número do segundo canal a ser lido (1 a 8) para leitura de apenas um canal utilizar: CNB = 0.

Param. saída : nenhum.

Exemplo de utilização:

GUARDA_AD_AF (500,100,3,2,3) ;

Armazena o valor 500 no ponto número 100 lido no canal 3, para leitura em 12 bit dos canais 2 e 3.

10.12 - INICIA PLACA

Descrição: Esta função inicializa a interrupção, aloca a memória e testa o conversor A/D e os contadores. Carrega também os dados do arquivo "PLACA.INI" que contém valores para inicialização da placa.

FORMATO DE DEFINIÇÃO:

function INICIA_PLACA: byte;

Param entrada : Nenhum.

Param saída : Valor da função = 0, 1, 2, 3 ou 4, conforme a situação resultante da instalação da placa, que será informado ao usuário pela rotina MOSTRAMSGINICIAPLACA, a seguir apresentada.

Exemplo de utilização:

X := INICIA_PLACA;

10.13 - MOSTRAMSGINICIAPLACA

DESCRIÇÃO: Esta rotina apresenta na tela a mensagem descritiva do erro (0, 1, 2, 3 ou 4) que por acaso possa resultar da execução da função inicia-placa.

Param entrada: Código do erro proveniente da função inicia placa que apresenta na tela os seguintes textos:

- 0 = Inicialização Ok
- 1 = Erro do conversor A/D
- 2 = Erro nos contadores da placa INTERDATA
- 3 = Erro na leitura ou inexistência do arquivo
PLACA.INI.
Execute o programa [ITD_INST.EXE] e tente novamente.
- 4 = Inicialização da placa feita com clock diferente do atual.
Execute o programa [ITD_INST.EXE] ou altere o clock do micro.

Param saída: nenhum

Exemplo de utilização:

```
mostramsginiciaplaca (X);
```

10.14 - FIM - PLACA

Descrição : Esta rotina desabilita as interrupções e libera memória alocada.

FORMATO DE DEFINIÇÃO:

```
procedure FIM_PLACA;
```

Param entrada : Nenhum.

Param saída : Nenhum.

Exemplo de utilização:

```
FIM_PLACA;
```

11 ROTINAS DE MANIPULAÇÃO DA PLACA

Para utilizar as rotinas descritas a seguir deve ser incluído no programa que chame as rotinas o seguinte comando:

```
USES ITD_BA30, ITD_VAR;
```

12 CONSIDERAÇÕES SOBRE A UTILIZAÇÃO DAS ROTINAS

Todo programa de manipulação da placa deve chamar no início a rotina `INICIA_PLACA` e no fim a rotina `FIM_PLACA`.

Como a rotina `INICIA_PLACA` aloca toda memória disponível no micro, outras rotinas que necessitem alocar memória (Por exemplo, rotina inicialização de gráfico) só poderão ser executadas antes da mesma. A variável `MAX_MEMÓRIA` indica o número de byte alocados pela rotina `INICIA_PLACA`.

O vetor de interrupção do relógio (vetor 28) é utilizado para atualização das saídas digitais.

Durante a leitura das entradas analógicas na máxima frequência todas as interrupções e demais processamentos realizados são interrompidos, ficando o micro continuamente lendo o conversor analógico digital e armazenando o valor lido na memória.

13 CONSIDERAÇÕES SOBRE A UTILIZAÇÃO DA PLACA

Apesar da placa poder ser utilizada com um sistema completo de aquisição e controle, é recomendável que sejam utilizados circuitos externos para compatibilização da potência e da faixa dos sinais de entrada e saída da placa com os do sistema controlado. Como as entradas e saídas da placa não possuem isolamento, estando todos referenciados ao terra do micro, para algumas aplicações é necessário que os circuitos externos proporcionem a isolamento requerida.