

Manual de Programação em Português



Índice

Capítulo 0 – Aspectos da PCL 1001

Capítulo 1 - Comandos e Funções da PCL1001

- 1.1 Acionando as saídas
- 1.2 Lendo as entradas
- 1.3 Comunicação Serial
- 1.4 Data e Hora
- 1.5 Atrasos
- 1.6 PWM
- 1.7 Acesso ao display
- 1.8 Velocímetro
- 1.9 Contador
- 1.10 Frequencímetro
- 1.11 Comparador
- 1.12 AD
- 1.13 DA
- 1.14 Cronômetro
- 1.15 Matemática
- 1.16 Jogo de Palavras
- 1.17 Periodímetro
- 1.18 Velocidade Angular
- 1.19 Memória EEPROM
- 1.20 Chamada de Telefone
- 1.21 Chamadas rotinas e saltos
- 1.22 Controle da Máquina
- 1.23 Variáveis
- 1.24 Constantes Químicas
- 1.25 Controle da Saída

Capítulo 2 – Controle de Fluxo

- 2.1 Laço ENQUANTO
- 2.2 Laço REPITA
- 2.3 Seleção de Casos

Capítulo 3 - Operadores Lógicos e Estruturas Condicionais Lógicas

- 3.1 Operadores
- 3.2 Estruturas Condicionais Lógicas

Capítulo 4 – Exemplos de Programação

Capítulo 0

Aspectos da PCL 1001

A PCL 1001 é um projeto nacional de baixo custo que agrega várias comandos que facilitam a programação de qualquer evento em uma linguagem de fácil entendimento chamada *AutoEasy*. Ao adquirir a PCL o desenvolvedor pode baixar pela Internet a da AutoEasy para desenvolvimento dos seus trabalhos.

Esta placa pode ser usada em diversos modos, como:

- Robótica Educacional;
- Mecatrônica;
- Controle de Processos Industriais;
- Automação Residencial;
- Controle de Aquários;
- Segurança;
- Sua imaginação...

Em seguida podemos apreciar os aspectos de hardware principais da PCL 1001:

- 6 entradas de contato seco;
- 5 saídas energizadas com controle de reversão;
- 1 saída de contato seco;
- Mostrador LCD;
- Saída PWM;
- 2 saídas de Leds;
- AD Converter;
- DA Converter;
- RS-232 Full Duplex;
- Possibilidades de expansão de I/Os através da porta serial.
- Porta de Interface Paralela;
- Contas Matemáticas em 16 bits;

Capítulo 1

Comandos e Funções da PCL1001

1.1 Acionando as saídas

LIGAR

Sintaxe:

ligar(parâmetro)

Descrição:

Este tipo de comando aciona o relê ou o led especificado em parâmetro. O parâmetro deve ser um número entre 1 e 6 ou led1 ou led2. Caso o parâmetro do relê seja “todos”, todos os relês serão acionados.

Exemplo:

ligar(1)	; Liga relê 1
ligar(2)	; Liga relê 2
ligar(led1)	; Liga o Led1

INVERTER

Sintaxe:

inverter(parâmetro)

Descrição:

Este tipo de comando inverte o relê especificado em parâmetro. O parâmetro deve ser um número entre 1 e 6.

Exemplo:

inverter(1)	; Inverte o relê 1
inverter(2)	; Inverte o relê 2

DESLIGAR

Sintaxe:

DESLIGAR(parâmetro)

Descrição:

Este tipo de comando desliga o relê ou o led especificado em parâmetro. O parâmetro deve ser um número entre 1 e 6 ou led1 ou led2. Caso o parâmetro do relê seja "TUDO", todos os relês serão desligados.

Exemplo:

DESLIGAR(TUDO)	; Desliga todos os relês
DESLIGAR(1)	; Desliga relê 1
DESLIGAR(2)	; Desliga relê 2
DESLIGAR(led1)	; Desliga o Led1
DESLIGAR(led2)	; Desliga o Led2

OSCILAR

Sintaxe:

OSCILAR(parâmetro)

Descrição:

Este comando oscila o relê ou o led especificado em parâmetro. O parâmetro deve ser um número entre 1 e 6 ou led1 ou led2.

Exemplo:

OSCILAR(1)	; OSCILAR com frequência de 1 HZ o relê 1
OSCILAR(2)	; OSCILAR com frequência de 1 HZ o relê 2
OSCILAR(led1)	; OSCILAR com frequência de 1 HZ o led1
OSCILAR(led2)	; OSCILAR com frequência de 1 HZ o led2

TROCAR

Sintaxe:

TROCAR(parâmetro)

Descrição:

Efetua a inversão do estado atual da saída. Caso a saída esteja ativa, a mesma irá desligar e vice-versa.

Exemplo:

TROCAR(SAÍDA(1)) ; caso a saída 1 esteja ligada, a mesma irá desligar

1.2 Lendo as entradas

SE SENSOR (número_da_entrada_digital) ENTÃO

Sintaxe:

```
SE SENSOR (número_entrada_digital) ENTÃO
    .
    .
    .
    comandos 1
    .
    .
    .
SENÃO
    .
    .
    .
    comandos 2
    .
    .
    .
FIM DO SE
```

Descrição:

Testa a condição atual da entrada definida em número_entrada_digital. Se for verdadeiro, executa os comandos 1. Se for falso, executa os comandos 2. O número_entrada_digital deve ser um número entre 1 e 8.

Exemplo:

```
SE SENSOR(1) ENTÃO                                     ;Testa entrada 1
    LIMPAR DISPLAY                                     ;se verdadeiro, executa estes comando...
    Delay_seg(1)
    DISPLAY(1)(SENSOR 1 fechado)
    TRANSMITIR(SENSOR 1 fechado)
    Delay_seg(1)
SENÃO                                                    ;se falso executa o que está após SENÃO
    LIMPAR DISPLAY
    Delay_seg(1)
    DISPLAY(1)(SENSOR 1 aberto)
    TRANSMITIR(SENSOR 1 aberto)
    Delay_seg (1)
FIM DO SE
```

SE NÃO SENSOR (número_da_entrada_digital) ENTÃO

Sintaxe:

```
SE NÃO SENSOR (número_entrada_digital) ENTÃO
```

```

.
.
.
comandos 1
.
.
.
SENÃO
.
.
.
comandos 2
.
.
.
FIM DO SE

```

Descrição:

Testa a condição atual da entrada definida em número_entrada_digital. Se for falso, executa os comando 1. Se for verdadeiro, executa os comando 2. O número_entrada_digital deve ser um número entre 1 e 6.

Exemplo:

```

SE NÃO SENSOR (1) ENTÃO                                ;Testa entrada 1
  LIMPAR DISPLAY                                       ;se falso, executa estes comando...
  Delay_seg(1)
  DISPLAY(1)(SENSOR 1 fechado)
  TRANSMITIR(SENSOR 1 fechado)
  Delay_seg (1)
SENÃO                                                    ;se verdadeiro executa o que está após
  LIMPAR DISPLAY
  Delay_seg (1)
  DISPLAY(1)(SENSOR 1 aberto)
  TRANSMITIR(SENSOR 1 aberto)
  Delay_seg (1)
FIM DO SE

```

1.3 Comunicação Serial

PAINEL

Descrição:

A PCL1001 pode comunicar-se com o painel PE01 externo conectado na porta serial. Para isso existe uma série de comandos que se corretamente seguidos farão o mesmo funcionar.

Exemplo

Modo com painel
modo painel data

; prepara portta para funcionar no modo painel
; prepara para funcionar para mostra a data

ACESSANDO A PLACA DE EXPANSÃO

Descrição:

A PCL1003 pode ter mais entradas e saídas, permitindo um total de 14 entradas e 14 saídas. Para isto, a placa de i/o deve estar conectada na rs232 da PCL. Os comando para ligar/desligar relês são os mesmos assim como os comando de teste de entrada digital.

Exemplo

LIGAR(14) ; liga o relê 14
DESLIGAR(14) ; desliga o relê 14

SE SENSOR(7) ENTÃO ; se entrada 7 verdadeira...

TRANSMITIR PROGRAMA

Sintaxe:

TRANSMITIR PROGRAMA

Descrição:

Este comando envia pelo canal serial todo o programa gravado na memória EEPROM. Este comando é importante para debugação.

Exemplo:

TRANSMITIR PROGRAMA ;envia pelo canal serial o programa residente na memória EEPROM.

ECOAR DADOS

Sintaxe:

ECOAR DADOS ON ou OFF

Descrição:

Caso este comando esteja ativo, todos os caracteres recebidos pela serial da placa serão retornados pela mesma. O estado default é desligado.

TERMINAL

Sintaxe:

TERMINAL (tempo)

Descrição:

Apresenta no DISPLAY todos os caracteres recebidos do canal serial durante <tempo> em segundos. Caso tempo seja 0, o comando TERMINAL será executado indefinidamente.

TAXA RATE

Sintaxe:

TAXA(taxa de transmissão)

Descrição:

Permite alterar o taxa de transmissão da máquina. Apenas dois TAXA rates estão disponíveis, o de 9600 e 19200 bps.

Exemplo:

TAXA(9600) ; configura taxa de transmissão para 9600 bps (default)
TAXA(19200) ; configura taxa de transmissão para 19200 bps

TRANSMITIR BIN

Sintaxe:

TRANSMITIR(bin\$(variável))

Descrição:

Este comando envia pelo canal serial o conteúdo da variável em binário especificada em variável.

Exemplo:

a%=1000 ; carrega a% com 1000
TRANSMITIR(bin\$(a%)) ; envia o mesmo pelo canal serial em binário

TRANSMITIR HEX

Sintaxe:

TRANSMITIR(hex\$(variável))

Descrição:

Este comando envia pelo canal serial o conteúdo da variável em hexadecimal especificada em variável.

Exemplo:

a%=1000 ; carrega a% com 1000
TRANSMITIR(hex\$(a%)) ; envia o mesmo pelo canal serial em hexadecimal

TRANSMITIR OCT

Sintaxe:

TRANSMITIR(oct\$(variável))

Descrição:

Este comando envia pelo canal serial o conteúdo da variável em octal especificada em variável.

Exemplo:

a%=1000 ; carrega a% com 1000
TRANSMITIR(oct\$(a%)) ; envia o mesmo pelo canal serial em octal

TRANSMITIR DEC

Sintaxe:

TRANSMITIR(variável)

Descrição:

Este comando envia pelo canal serial o conteúdo da variável em decimal especificada em variável.

Exemplo:

a%=1000 ; carrega a% com 1000
TRANSMITIR(a%) ; envia o mesmo pelo canal serial em decimal

TRANSMITIR CRONÔMETRO

Sintaxe:

TRANSMITIR(cronômetro\$)

Descrição:

Envia pelo canal serial o valor das variáveis de contagem do cronômetro.

Exemplo:

TRANSMITIR(cronômetro\$)

TRANSMITIR(MEMÓRIA\$)

Sintaxe:

TRANSMITIR(MEMÓRIA\$)

Descrição:

Envia pelo canal serial o caracter posicionado em ENDERECO DA MEMÓRIA.

Exemplo:

ENDERECO DA MEMÓRIA(100)
TRANSMITIR(MEMÓRIA\$)

TRANSMITIR DATA\$

Sintaxe:

TRANSMITIR(DATA\$)

Descrição:

Envia pelo canal serial a data corrente programada no dispositivo.

Exemplo:

TRANSMITIR(DATA\$) ; envia pelo canal serial a data corrente

TRANSMITIR

Sintaxe:

TRANSMITIR(string_de_caracteres)

Descrição:

Envia pelo canal serial os dados contidos em string_de_caracteres.

Exemplo:

TRANSMITIR(Cerne Tecnologia)

TRANSMITIR_ASC

Sintaxe:

TRANSMITIR_asc(código da tabela ASCII)

Descrição:

Envia pelo canal serial os dados contidos em código da tabela ASCII.

Exemplo:

TRANSMITIR_asc(65,66,67,68,69,) ;transmite a string “ABCDE” pelo canal serial.

TRANSMITIR_HORA\$

Sintaxe:

TRANSMITIR(HORA\$)

Descrição:

Envia pelo canal serial a hora corrente programada no dispositivo.

Exemplo:

TRANSMITIR(HORA\$) ; envia pelo canal serial a hora corrente

TRANSMITIR_FREQ\$

Sintaxe:

TRANSMITIR(freq\$)

Descrição:

Envia pelo canal serial o número de pulsos ocorridos em 1 segundo na entrada 6.

Exemplo:

etiqueta1 para novamente
ENTRADA6 como frequencimetro

novamente

TRANSMITIR(freq\$)

vá para novamente

RECEPÇÃO DE DADOS

Sintaxe:

```
SE RECEPÇÃO DE DADOS="parâmetro" ENTÃO
    TRANSMITIR(Character recebido!)
FIM DO SE
```

Descrição:

Verifica se há algum dado no buffer de recepção, e caso haja, faz o teste com o caractere recebido e com o parâmetro.

Exemplo:

```
SE RECEPÇÃO DE DADOS="N" ENTÃO
    TRANSMITIR(Character recebido com sucesso!)
FIM DO SE
```

1.4 Data e Hora

SE HORA\$>

Sintaxe:

```
Se HORA$>hora_minuto ENTÃO
    .
    .
    comandos
    .
    .
FIM DO SE
```

Descrição:

Este comando testa se hora_minuto são maiores que o do RTC da máquina. Caso seja, comando serão executados. Este comando também pode funcionar como um despertador.

Exemplo:

```
SE HORA$>06_50 ENTÃO                ; se passou das 6h50mim então...
    LIGAR(1)                          ; liga o relê 1
    DISPLAY(1)(E hora de acordar!)    ; apresenta mensagem no display
FIM DO SE
```

SE HORA\$<

Sintaxe:

```
SE HORA$<hora_minuto ENTÃO
    .
```

```
.  
comando  
.  
FIM DO SE
```

Descrição:

Este comando testa se hora_minuto são menores do que o do RTC da máquina. Caso seja, comando serão executados. Este comando também pode funcionar como um despertador.

Exemplo:

```
SE HORA$<06_50 ENTÃO ; se não passou das 6h50mim então...  
DESLIGAR(1) ; desliga o relê 1  
DISPLAY(1)(Pode dormir...) ; apresenta mensagem no display  
FIM DO SE
```

SE DATA\$

Sintaxe:

se DATA\$ = data de teste ENTÃO

Descrição:

Verifica se a data da máquina é igual a data de teste.

Exemplo

```
SE DATA$="10_01" ENTÃO ; hoje é 10 de Janeiro?  
LIGAR(3) ; sim, então liga o relê 3  
FIM DO SE
```

AJUSTE DE DATA E HORA

Sintaxe:

```
HORA$="20_57_00"  
DATA$="21_01_05"
```

Descrição:

Ajusta via programa a data e hora do sistema.

DISPLAY DATA\$

Sintaxe:

DISPLAY(número da linha)(DATA\$)

Descrição:

Apresenta no DISPLAY a data corrente programada no dispositivo.

Exemplo:

DISPLAY(1)(DATA\$) ; escreve na linha 1 do DISPLAY a data corrente
DISPLAY(2)(DATA\$) ; escreve na linha 2 do DISPLAY a data corrente

DISPLAY HORA\$

Sintaxe:

DISPLAY(número da linha)(HORA\$)

Descrição:

Apresenta no DISPLAY a hora corrente programada no dispositivo.

Exemplo:

DISPLAY(1)(HOUA\$) ; escreve na linha 1 do DISPLAY a hora corrente
DISPLAY(2)(HOUA\$) ; escreve na linha 2 do DISPLAY a hora corrente

SE HORA\$

Sintaxe:

Se HORA\$ =hora_minuto ENTÃO
.
.
comando
.
.
FIM DO SE

Descrição:

Este comando testa se hora_minuto são iguais ao do RTC da máquina. Caso seja, os comando são executados. Este comando também pode funcionar como um despertador.

Exemplo:

SE HORA\$=20_50 ENTÃO


```
LIGAR(1)
DISPLAY(1)(E horaa de acordar!)
FIM DO SE
```

1.26 Atrasos

TEMPO_MS(VARIÁVEL)

Sintaxe:

```
Variável = valor
tempo_ms(variável)
```

Descrição:

O comando tempo_ms conta um tempo em milisegundos em função do parâmetro carregado na variável.

Exemplo

```
a%=1000 ; atribui 1000 a a%
tempo_ms(a%) ; conta 1000 ms
```

TEMPO_SEG(VARIÁVEL)

Sintaxe:

```
Variável = valor
tempo_seg(variável)
```

Descrição:

O comando tempo_seg conta um tempo em segundos em função do parâmetro carregado na variável.

Exemplo

```
a%=10 ; atribui 10 a a%
tempo_seg(a%) ; conta 10 segundos
```

TEMPO_MS

Sintaxe:

```
tempo_ms(tempo)
```

Descrição:

Este comando conta um tempo em milisegundos especificado em tempo.

Exemplo:

```
LIGAR(1) ; liga o relê 1
tempo_ms(1000) ; aguarda 1000 ms
```

DESLIGAR(1) ; desliga o relê 1

TEMPO_SEG

Sintaxe:

tempo_seg(tempo)

Descrição:

Realiza a contagem de um tempo em segundos especificado em tempo.

Exemplo:

```
LIGAR(1) ; liga o relê 1
tempo_seg(10) ; aguarda 10 seg
DESLIGAR(1) ; desliga o relê 1
```

1.17 PWM

INCREMENTAR PWM

Sintaxe:

INCREMENTAR PWM

Descrição:

Incrementa o *duty cycle* da saída PWM. A frequência do PWM é de 1kHz.

Exemplo:

INCREMENTAR PWM ; incrementa o ciclo ativo da saída

DECREMENTAR PWM

Sintaxe:

DECREMENTAR PWM

Descrição:

Decrementa o *duty cycle* da saída PWM. A frequência do PWM é de 1kHz.

Exemplo:

DECREMENTAR PWM ; decrementa o ciclo ativo da saída

LIGAR PWM

Sintaxe:

LIGAR PWM

Descrição:

Este comando coloca o *duty cycle* em 100%.

Exemplo:

LIGAR PWM ; põe o duty cycle em 100%

DESLIGAR PWM

Sintaxe:

DESLIGAR PWM

Descrição:

Este comando coloca o *duty cycle* em 0%.

Exemplo:

DESLIGAR PWM ; põe o duty cycle em 0%

PWM

Sintaxe:

pwm(duty cycle)

Descrição:

Ajusta o *duty cycle* da saída PWM. O *duty cycle* deve estar entre 0 até 100.

Exemplo:

pwm(50) ; ajusta o PWM para 50%

1.7 Acesso ao Mostrador

LIMPAR MOSTRADOR

Sintaxe:

LIMPAR MOSTRADOR

Descrição:

LIMPAR MOSTRADOR

Exemplo:

LIMPAR MOSTRADOR ;Limpa as duas linhas do MOSTRADOR

DISPLAY

Sintaxe:

DISPLAY(número_da_linha)(string_de_caracteres)

Descrição:

Apresenta em número_da_linha a string_de_caracteres.

Exemplo:

LIMPAR MOSTRADOR ;Limpa DISPLAY
DISPLAY(1)(PCL1001) ; Apresenta na linha 1 do DISPLAY "PCL1001"
DISPLAY(2)(A placa inteligente) ; Apresenta na linha 2 do DISPLAY "A placa inteligente"

DISPLAY DECIMAL

Sintaxe:

DISPLAY(número da linha)(CAD\$(decimal))

Descrição:

Este comando apresenta em número da linha a tensão de entrada do conversor analógico em decimal.

Exemplo:

DISPLAY(1)(CAD\$(decimal))

DISPLAY VOLTS

Sintaxe:

DISPLAY(número da linha)(CAD\$(voltagem))

Descrição:

Apresenta através do número da linha a tensão de entrada do conversor analógico.

Exemplo:

```
DISPLAY(1)(CAD$(voltagem))
```

DISPLAY FREQ\$

Sintaxe:

```
DISPLAY(número da linha)(freq$)
```

Descrição:

Apresenta o número de pulsos que ocorreram em 1 segundo na entrada 6.

Exemplo:

```
etiqueta1 para novamente  
ENTRADA6 como frequencímetro
```

novamente

```
DISPLAY(1)(freq$)  
vá para novamente
```

DISPLAY MONTH

Sintaxe:

```
DISPLAY(número da linha)(mês$)
```

Descrição:

Apresenta em número da linha o mês corrente do rtc da placa.

Exemplo:

```
DISPLAY(1)(mês$) ; apresenta na linha 1 do DISPLAY o mês corrente
```

DISPLAY VELOCIDADE

Sintaxe:

```
DISPLAY(número da linha)(velocidade$)
```

Descrição:

Apresenta em número da linha, a última medição de velocidade realizada. Para que uma nova medição seja feita, o comando VELOCIDADE ON deve ser utilizado novamente.

DISPLAY(VARIÁVEL)

Sintaxe:

DISPLAY(número da linha)(variável%)

Descrição:

Mostra em número da linha do DISPLAY o valor corrente da variável.

Exemplo

```
a%=100 ;carrega a% com 100 decimal
DISPLAY(1)(a%) ;apresenta conteúdo da variável
```

DISPLAY BIN\$

Sintaxe:

DISPLAY(número da linha)(bin\$(variável))

Descrição:

Mostra em número da linha do DISPLAY o valor corrente da variável em binário.

Exemplo

```
a%=1000 ;carrega a% com 1000 decimal
DISPLAY(1)(bin$(a%)) ;apresenta conteúdo da variável em binário
```

DISPLAY HEX\$

Sintaxe:

DISPLAY(número da linha)(hex\$(variável))

Descrição:

Mostra em número da linha do DISPLAY o valor corrente da variável em hexadecimal.

Exemplo

```
a%=100 ;carrega a% com 100 decimal
DISPLAY(1)(hex$(a%)) ;apresenta conteúdo da variável em hexa
```

DISPLAY OCT\$

Sintaxe:

DISPLAY(número da linha)(oct\$(variável))

Descrição:

Mostra em número da linha do DISPLAY o valor corrente da variável em octal.

Exemplo

a%=100 ;carrega a% com 100 decimal
DISPLAY(1)(oct\$(a%)) ;apresenta conteúdo da variável em octal

OSCILAR DISPLAY

Sintaxe:

OSCILAR(DISPLAY)

Descrição:

Este comando faz com que a mensagem que fique oscilando no DISPLAY a uma frequência de 1Hz.

NÃO OSCILAR DISPLAY

Sintaxe:

NÃO OSCILAR(DISPLAY)

Descrição:

Este comando faz com que o DISPLAY saia do modo oscilante.

DISPLAY CONTADOR DE PULSOS

Sintaxe:

DISPLAY(número da linha)(CONTADOR DE PULSOS\$)

Descrição:

Este comando permite que seja apresentado em número da linha o valor corrente da contagem de pulsos externa. O parâmetro número da linha deve ser 1 ou 2.

Exemplo:

DISPLAY(1)(CONTADOU DE PULSOS\$); apresenta na linha 1 a contagem de pulsos

DISPLAY CRONÔMETRO

Sintaxe:

DISPLAY(número da linha)(CRONÔMETRO\$)

Descrição:

Apresenta o estado do contador no DISPLAY. O número da linha determina em qual linha esta será apresentada.

Exemplo:

INICIAR CRONÔMETRO
DISPLAY(1)(CRONÔMETRO \$)

ROTACIONAR DISPLAY PARA ESQUERDA

Sintaxe:

ROTACIONAR DISPLAY PARA ESQUERDA

Descrição:

Executa uma rotação para a esquerda com a mensagem que está apresentada no DISPLAY.

Exemplo:

1001 *** " DISPLAY(1)(*** PCL 1001 ***) ; escreve no DISPLAY " *** PCL
ROTACIONAR DISPLAY PARA ESQUERDA ; faz uma rotação para a esquerda

ROTACIONAR DISPLAY PARA DIREITA

Sintaxe:

ROTACIONAR DISPLAY PARA DIREITA

Descrição:

Executa uma rotação para a direita com a mensagem que está apresentada no DISPLAY.

Exemplo:

1001 *** " DISPLAY(1)(*** PCL 1001 ***) ; escreve no DISPLAY " *** PCL
ROTACIONAR DISPLAY PARA DIREITA ; faz uma rotação para a direita

1.8 Velocímetro

VELOCIDADE ON

Sintaxe:

VELOCIDADE ON

Descrição:

Ativa o medidor de velocidade externa da máquina. As entradas 1 e 2 funcionam de modo a captar a diferença entre os tempos. O espaço deve estar previamente definido em ESPAÇO PADRÃO.

Exemplo

ESPAÇO PADRÃO=100	; diferença entre os sensores é de 100 m
VELOCIDADE ON	; habilita a medição de velocidade
DISPLAY(1)(VELOCÍMETRO\$)	; apresenta o resultado da medição

ESPAÇO PADRÃO

Sintaxe:

ESPAÇO PADRÃO

Descrição:

Este comando determina o espaço que há entre os sensores. O valor máximo de espaço padrão é 255 metros.

SE VELOCIDADE\$=

Sintaxe:

SE VELOCIDADE\$=velocidade ENTÃO

Descrição:

Verifica se a velocidade é igual á especificada.

Exemplo

SE VELOCIDADE\$=30 ENTÃO	; velocidade é igual a 30 m/s?
LIGAR(1)	; sim, então liga o relê 1
FIM DO SE	

SE VELOCIDADE\$>

Sintaxe:

SE VELOCIDADE\$ > velocidade ENTÃO

Descrição:

Verifica se a velocidade é maior que a especificada.

Exemplo

```
SE VELOCIDADE$>30 ENTÃO ; velocidade é maior que 30 m/s?  
LIGAR(1) ; sim, então liga o relê 1  
FIM DO SE
```

SE VELOCIDADE\$<

Sintaxe:

```
SE VELOCIDADE$ < velocidade ENTÃO
```

Descrição:

Verifica se a velocidade é menor que a especificada.

Exemplo

```
SE VELOCIDADE$<30 ENTÃO ; velocidade é menor que 30 m/s?  
LIGAR(1) ; sim, então liga o relê 1  
FIM DO SE
```

1.9 Contador

ENTRADA6 COMO CONTADOR

Sintaxe:

```
ENTRADA6 COMO CONTADOR
```

Descrição:

Promove o funcionamento da entrada 6 como um contador de pulsos.

Exemplo:

```
ENTRADA6 COMO CONTADOR
```

CONTADOR

Sintaxe:

```
contador(valor)
```

Descrição:

Este tipo de comando permite que seja contado n pulsos especificados por valor. O valor deve ser um número entre 0 e 65535.

Exemplo:

etiqueta1 para again	; cria etiqueta chamado again
ENTRADA6 COMO CONTADOR	; faz com que a entrada 6 funcione como contador
contador(1000)	; ajusta para contar 1000 pulsos

again

SECONTADOR ENTÃO	; quando a contagem se encerrar...
LIGAR(1)	; processa esses comando
tempo_ms(1000)	
DESLIGAR(1)	
tempo_ms(1000)	
FIM DO SE	
vá para again	; salta para again

LIMPAR PULSE COUNTER

Sintaxe:

LIMPAR CONTADOR DE PULSOS\$

Descrição:

Este comando limpa as variáveis de contagem de pulsos externo.

INICIAR CONTADOR DE PULSOS

Sintaxe:

INICIAR CONTADOR DE PULSOS\$

Descrição:

Este comando habilita o funcionamento do contador de pulsos.

PARAR PULSE COUNTER

Sintaxe:

PARAR CONTADOR DE PULSOS\$

Descrição:

Este comando desabilita o funcionamento do contador de pulsos.

ENTRADA6 COMO CONTADOR

Sintaxe:

ENTRADA6 COMO CONTADOR

Descrição:

Este comando faz com que a entrada 6 funcione como contador de pulsos.

SE CONTADOR DE PULSOS\$=

Sintaxe:

Se CONTADOR DE PULSOS\$ = parâmetro ENTÃO

Descrição:

Este comando verifica se o contador de pulsos é igual a parâmetro. O parâmetro deve ser um número compreendido entre 0 e 65535 inclusive.

Exemplo:

```
SE CONTADOR DE PULSOS$=10000 ENTÃO           ; se atingiu 10000 contagens
então...
    TRANSMITIR(10000 pulsos!)
    LIMPAR CONTADOR DE PULSOS$
FIM DO SE
```

SE CONTADOR DE PULSOS\$>

Sintaxe:

SE CONTADOR DE PULSOS\$>parâmetro ENTÃO

Descrição:

Este comando verifica se o contador de pulsos é maior que o parâmetro. O parâmetro deve ser um número compreendido entre 0 e 65535 inclusive.

Exemplo:

```
SE CONTADOR DE PULSOS$>5000 ENTÃO           ; se passou 5000 contagens
então...
    TRANSMITIR(5000 pulsos!)
    LIMPAR CONTADOR DE PULSOS$
FIM DO SE
```

SE CONTADOR DE PULSOS\$<

Sintaxe:

SE CONTADOR DE PULSOS\$<parâmetro ENTÃO

Descrição:

Este comando verifica se o contador de pulsos é menor que o parâmetro. O parâmetro deve ser um número compreendido entre 0 e 65535 inclusive.

Exemplo:

```
SE CONTADOR DE PULSOS$<100 ENTÃO           ; se é menor que 100 contagens
então...
    TRANSMITIR(100 pulsos!)
    LIMPAR CONTADOU DE PULSOS$
FIM DO SE
```

SE ENTRADA\$=

Sintaxe:

SE ENTRADA\$ = valor ENTÃO

Descrição:

Este comando verifica se a entrada de dados corresponde ao valor especificado em valor.

Exemplo:

```
SE ENTRADA$=1 ENTÃO           ;se a entrada 1 estiver fechada...
    TRANSMITIR(100 pulsos!)    ;executa comando
    LIMPAR CONTADOU DE PULSOS$
FIM DO SE
```

1.10 Frequencímetro

ENTRADA6 COMO ENTRADA

Sintaxe:

ENTRADA6 COMO ENTRADA

Descrição:

Este tipo de comando permite que a entrada 6 funcione como uma entrada normal.

Exemplo:

ENTRADA6 COMO ENTRADA

ENTRADA6 COMO FREQUENCÍMETRO

Sintaxe:

ENTRADA6 COMO FREQUENCÍMETRO

Descrição:

Este tipo de comando permite que a entrada 6 funcione como uma entrada de frequencímetro.

Exemplo:

ENTRADA6 COMO FREQUENCÍMETRO

RPM\$

Sintaxe:

DISPLAY(número da linha)(rpm\$)

Descrição:

Apresenta em número da linha o valor atual da medição do tacógrafo.

ALETAS

Sintaxe:

ALETAS PADRÃO= valor

Descrição:

É utilizado no comando RPM para medir frequências externas. Na verdade, aletas é um divisor caso o encoder tenha mais de uma aleta. O valor padrão é 1 e o máximo permitido é 60.

1.11 Comparador

TENSÃO DE REFERÊNCIA

Sintaxe:

TENSÃO DE REFERÊNCIA = tensão de referência

Descrição:

Este comando gera uma tensão para ser comparada com a tensão de entrada. Caso a tensão de referência seja maior que a tensão de entrada, a função comparador retorna 0. Caso contrário, retorna 1.

Exemplo:

TENSÃO DE REFERÊNCIA=2.00
etiqueta1 para again

again

SE COMPARADOR ENTÃO ; se a tensão de entrada for maior que a de referência...
LIGAR(1) ; liga o relê 1
FIM DO SE

SE NÃO COMPARADOR ENTÃO ; se a tensão de entrada for menor que a de
referência...
DESLIGAR(1) ; desliga o relê 1
FIM DO SE

1.12 AD

MÁXIMA VOLTAGEM

Sintaxe:

MÁXIMA VOLTAGEM tensão_máxima

Descrição:

Permite que sejam medidas e apresentadas no DISPLAY tensões acima de 5V. Para isso um circuito condicionador de sinal deve ser utilizado. Deve-se lembrar que a tensão máxima de medida é de 65V e que o valor default é 5V.

Exemplo:

MÁXIMA VOLTAGEM 50 ; permite que sejam medidas tensões até 50V

SE CAD\$>

Sintaxe:

SE CAD\$>tensão de referência ENTÃO

Descrição:

Verifica se a tensão de entrada é maior que a tensão de referência. Caso este teste seja afirmativo, os comando seguintes serão executados.

Exemplo:

SE CAD\$>2.00 ENTÃO ; se a tensão de entrada for maior que 2 V
LIGAR(1) ; então liga o relê 1
FIM DO SE

SE CAD\$<

Sintaxe:

SE CAD\$<tensão de referência ENTÃO

Descrição:

Verifica se a tensão de entrada é menor que a tensão de referência. Caso este teste seja afirmativo, os comando seguintes serão executados.

Exemplo:

```
SE CAD$<2.00 ENTÃO           ; se a tensão de entrada for menor que 2 V
    LIGAR(1)                   ; então liga o relê 1
FIM DO SE
```

SE CAD\$>=

Sintaxe:

SE CAD\$>=tensão de referência ENTÃO

Descrição:

Verifica se a tensão de entrada é maior ou igual a tensão de referência. Caso este teste seja afirmativo, os comando seguintes serão executados.

Exemplo:

```
SE CAD$>=2.00 ENTÃO           ; se a tensão de entrada for maior ou igual a 2 V
    LIGAR(1)                   ; então liga o relê 1
FIM DO SE
```

SE CAD\$<=

Sintaxe:

SE CAD\$<=tensão de referência ENTÃO

Descrição:

Verifica se a tensão de entrada é menor ou igual a tensão de referência. Caso este teste seja afirmativo, os comando seguintes serão executados.

Exemplo:

```
SE CAD$<=2.00 ENTÃO           ; se a tensão de entrada for menor ou igual a 2 V
    LIGAR(1)                   ; então liga o relê 1
FIM DO SE
```


SE CAD\$=

Sintaxe:

SE CAD\$ = tensão de referência ENTÃO

Descrição:

Verifica se a tensão de entrada é igual a tensão de referência. Caso este teste seja afirmativo, os comando seguintes serão executados.

Exemplo:

```
SE CAD$=2.00 ENTÃO      ; se a tensão de entrada for igual a 2 V
    LIGAR(1)             ; então liga o relê 1
FIM DO SE
```

SE CAD\$> E CAD\$<

Sintaxe:

SE CAD\$>tensão de referência mínima E CAD\$<tensão de referência máxima ENTÃO

Descrição:

Este comando permite que o ad da máquina funcione como um comparador de janela, dentro de uma faixa de tensões o teste condicional seja positivo.

Exemplo:

```
SE CAD$>2.00 E CAD$<4.00 ENTÃO      ; se a tensão medida for maior que 2.00 V e
                                        ;menor 3.00
    LIGAR(1)                           ; então liga o relê 1
FIM DO SE
```

1.13 DA

SAÍDA DE TENSÃO

Sintaxe:

SAÍDA DE TENSÃO(tensão de saída)

Descrição:

Cria uma tensão de saída correspondente à tensão de saída. O valor de tensão de saída deve estar entre 0 e 5.

Exemplo:

SAÍDA DE TENSÃO (3.2)

1.14 Cronômetro

INICIAR CRONÔMETRO

Sintaxe:

```
INICIAR CRONÔMETRO
```

Descrição:

Inicia o funcionamento do cronômetro interno da máquina.

Exemplo:

```
INICIAR CRONÔMETRO  
DISPLAY(1)( CRONÔMETRO$)
```

PARAR CRONÔMETRO

Sintaxe:

```
PARAR CRONÔMETRO
```

Descrição:

Este comando para o funcionamento do cronômetro interno da máquina.

Exemplo:

```
PARAR CRONÔMETRO  
DISPLAY(1)( CRONÔMETRO$)
```

LIMPAR CRONÔMETRO

Sintaxe:

```
LIMPAR CRONÔMETRO
```

Descrição:

Executa a limpeza das variáveis de contagem de cronômetro.

Exemplo:

```
LIMPAR CRONÔMETRO  
DISPLAY(2)( CRONÔMETRO$)
```

1.15 Matemática

RAIZ

Sintaxe:

variável %=valor ;carrega variável% com valOU
variável% = raiz variável% ; tira a raiz quadrada.

Descrição:

A função raiz permite que se retire a raiz quadrada de um número qualquer até 65535. A mesma é armazenada em uma variável.

Exemplo

a%=100 ; carrega a% com valor
a%=raiz a% ; tira a raiz quadrada.

PORCENTAGEM

Sintaxe:

variável%=valor ;carrega a% com valor
variável%=variável%+ 20% ;soma mais 20 % da própria variável.

Descrição:

A função porcentagem calcula a porcentagem de um número com base nele mesmo. O resultado é armazenada na própria variável.

Exemplo

a%=100 ;carrega a% com 100 decimal
a%=a% + 20% ;soma mais 20 % da própria variável, ou seja, 120.
B%=1000 ;carrega b% com 1000 decimal
b%=b% - 50 % ;retira 50 por cento da mesma.

SOMA

Sintaxe:

variável%=1
variável%=variável% + constante

Descrição:

A função soma uma constante a uma variável do sistema. O sistema possui oito variáveis de usuário.

Exemplo

a%=100
a%=a% + 20

;carrega a% com 100 decimal
;soma 20 ao conteúdo da variável

SUBTRAÇÃO

Sintaxe:

variável%=1
variável%=variável% - constante

Descrição:

A função subtrai uma constante de uma variável do sistema. O sistema possui oito variáveis de usuário.

Exemplo

a%=100
a%=a% - 50

;carrega a% com 100 decimal
;subtrai 50 da variável

MULTIPLICAÇÃO

Sintaxe:

variável%=1
variável%=variável% * constante

Descrição:

Esta função multiplica uma constante e armazena em uma variável do sistema. O sistema possui oito variáveis de usuário.

Exemplo

a%=100
a%=a% - 50

;carrega a% com 100 decimal
;subtrai 50 da variável

DIVISÃO

Sintaxe:

variável%=variável% / 2

Descrição:

A função divide uma constante por uma variável do sistema. O resultado é armazenado na própria variável.

Exemplo

a%=100
a%=a% / 50

;carrega a% com 100 decimal
;subtrai 50 da variável

POTÊNCIA

Sintaxe:

variável%=variável% ^ 2

Descrição:

Eleva variável% pela constante e salva nela mesma. O sistema possui oito variáveis de usuário.

Exemplo

```
a%=10                ;carrega a% com 10 decimal
a%=10%^2            ;eleva a segunda potência
```

SE VARIÁVEL=

Sintaxe:

SE variável%=constante ENTÃO

Descrição:

Verifica se a variável é igual à constante. O sistema possui oito variáveis de usuário.

Exemplo

```
a%=100                ;carrega a% com 100 decimal
SE a%=100 ENTÃO      ;se a% igual a 100 então...
```

SE VARIÁVEL>

Sintaxe:

SE variável%>constante ENTÃO

Descrição:

Verifica se a variável é maior que à constante. O sistema possui oito variáveis de usuário.

Exemplo

```
a%=100                ;carrega a% com 100 decimal
SE a%>50 ENTÃO      ;se variável maior que 50 então
```

SE VARIÁVEL<

Sintaxe:

```
SE variável%<3 ENTÃO
```

Descrição:

Verifica se a variável é menor que a constante.

Exemplo

```
a%=100 ;carrega a% com 100 decimal  
SE a%<150 ENTÃO ; se for menor que 150 então...
```

HIPOT

Sintaxe:

```
a%=hipot(b%,c%)
```

Descrição:

Permite calcular a hipotenusa de um triângulo retângulo. A variável a% é o resultado do cálculo, ou seja, a hipotenusa e as variáveis b% e c% são os catetos.

Exemplo:

```
b%=6 ; atribui 6 a variável b%  
c%=8 ; atribui 8 a variável c%  
a%=hipot(b%,c%) ; calcula a hipotenusa e armazena em a%  
DISPLAY(1)(a%) ; apresenta o resultado, ou seja, a hipotenusa
```

CIRCUNF

Sintaxe:

```
b%=raio  
a@= circunf(b%)
```

Descrição:

Permite calcular a circunferência de um círculo. O raio deve estar previamente carregado em b%.

Exemplo:

```
b%=2 ; atribui 2 a variável b%  
a@=circunf(b%) ; calcula a circunferencia e armazena em a@  
DISPLAY(1)(a@) ; apresenta o resultado de a@
```

AREA_CIRCUNF

Sintaxe:

```
b%=raio
a@= area_circunf(b%)
```

Descrição:

Permite calcular a área da circunferência de um círculo. O raio deve estar previamente carregado em b%.

Exemplo:

```
b%=2 ; atribui 2 a variável b%
a@=area_circunf(b%) ; calcula a área da circunferencia e armazena em a@
DISPLAY(1)(a@) ; apresenta o resultado de a@
```

CONSTANTES

```
pi=3.14 ; pi
e=2.72 ; exponencial
g=9.81 ; gravidade da terra
c=3.00 ; velocidade da luz
cte_g=6.67 ; constante gravitacional
Na=6.02 ; Número de Avogrado
R=8.31 ; Constante universal dos gases
c2=8.99 ; Relação massa-energia
&0=8.85 ; Constante de permissividade
u0=1.26 ; Constante de permeabilidade
h=6.63 ; Constante de Plank
k=1.38 ; Constante de Boltzmann
e=1.60 ; Carga elementar
me=9.11 ; Massa do elétron
mp=1.67 ; Massa do próton
mn=1.68 ; Massa do neutrôn
md=3.34 ; Massa dp dêuteron
a=5.29 ; Raio de Bohr
ub=9.27 ; Magnéton de Bohr
atm=1.01 ; Atmosfera da Terra
```

Descrição:

Iguala uma variável de duas casas decimais uma constante.

Exemplo

```
a@=pi ; atribui a a@ pi, ou seja, 3.14
b@=e ; atribui a b@ e, ou seja, 2.72
```

SIN(), COS(), TAN()...

Funções:

SIN	; calcula o seno de um ângulo
COS	; calcula o cosseno de um ângulo
TAN	; calcula a tangente de um ângulo
SEC	; calcula a secante de um ângulo
COSEC	; calcula a cosecante de um ângulo
COTAN	; calcula a cotangente de um ângulo
EXP	; calcula a exponencial de um número
LOG	; calcula o log na base 10 de uma constante
LN	; calcula o log na base e

Descrição:

O parâmetro deve sempre vir em graus para que seja calculada na função acima. A mesma será armazenada em uma variável de casa decimal dupla.

Exemplo

a@=sin(30) ; atribui a a@, o seno de 30 graus, ou seja, 0,5
b@=cos(90) ; atribui a b@, o coseno de 90 graus, ou seja, 0.

FATORIAL

Sintaxe:

variável= constante !

Descrição:

A função fatOUial retOUina um número mediante o valOU infOUmado pOU constante. Constante deve ser um número entre 0 e 8 inclusive.

Exemplo

a%=0! ; atribui o fatOUial de 0 a a%
b%=1! ; atribui o fatOUial de 1 a b%
c%=5! ; atribui o fatOUial de 5 a c%

1.16 Jogo

JOGO

Sintaxe:

JOGO

Descrição:

Inicia o jogo de palavras existente na placa. Caso o jogador tenha achado a frase, a palavra JOGO será positiva. Caso contrário negativa.

Exemplo

```
JOGO          ; chama o jogo existente na placa

SE NÃO JOGO ENTÃO          ; se o jogador não encontrou a frase ...
  DISPLAY(1)(Voce perdeu...) ; apresenta mensagem que o jogo está perdido
FIM DO SE

SE JOGO ENTÃO
  DISPLAY(1)(Voce ganhou!) ; apresenta mensagem que o jogo foi vencido
FIM DO SE
```

1.17 Periodímetro

PERIODÍMETRO

Sintaxe:

```
DISPLAY(número da linha)(PERIODÍMETRO$)
```

Descrição:

Esta função permite calcular o período da onda que chega através da entrada 6 do sistema. Não podemos esquecer que a entrada deve estar configurada como entrada de frequência.

Exemplo

```
ENTRADA6 COMO FREQUENCÍMETRO ; entrada 6 como entrada de pulsos
DISPLAY(1)(PERIODÍMETRO$)    ; apresenta na linha 1 do DISPLAY o período da onda
```

1.18 Velocidade Angular

VEL_ANG

Sintaxe:

```
variável de duas casas decimais= vel_ang$
```

Descrição:

Calcula a velocidade angular e atribui a alguma variável. Para que este comando seja utilizado, a entrada 6 deve estar configurada como entrada para frequencímetro, para que satisfaça a equação $W=2\pi \cdot f$.

Exemplo

```
ENTRADA6 COMO FREQUENCÍMETRO ; entrada 6 como entrada de pulsos
a@= vel_ang$                  ; calcula a velocidade angular
```

1.19 Memória EEPROM

INCREMENTAR ENDERECO DA MEMÓRIA

Sintaxe:

INCREMENTAR ENDERECO DA MEMÓRIA

Descrição:

Este comando incrementa o endereçamento da memória não volátil contida na máquina.

Exemplo:

INCREMENTAR ENDERECO DA MEMÓRIA
ESCREVER NA MEMÓRIA(A)

DECREMENTAR ENDERECO DA MEMÓRIA

Sintaxe:

DECREMENTAR ENDERECO DA MEMÓRIA

Descrição:

Este comando decrementa o endereçamento da memória não volátil contida na máquina.

Exemplo:

DECREMENTAR ENDERECO DA MEMÓRIA
ESCREVER NA MEMÓRIA(B)

ESCREVER NA MEMÓRIA

Sintaxe:

ESCREVER NA MEMÓRIA(caracter)

Descrição:

Este comando escreve na posição endereçada por ENDERECO DA MEMÓRIA o caracter especificado pelo programa. Deve-se verificar que o comando escreve caracter e não strings.

Exemplo:

ESCREVER NA MEMÓRIA(J)

LEITURA DA MEMÓRIA

Sintaxe:

```
SE LEITURA DA MEMÓRIA$=A ENTÃO
    LIGAR(1)
FIM DO SE
```

Descrição:

O comando verifica se na posição endereçada se existe o caracter usado para comparação. Caso exista, o comando é executado.

Exemplo:

```
ENDERECO DA MEMÓRIA=100
SE LEITURA DA MEMÓRIA=A ENTÃO
    TRANSMITIR (Há o caracter A nesta posição de memória)
FIM DO SE
```

1.20 Chamada de Telefone

CHAMAR O TELEFONE

Sintaxe:

```
CHAMAR O TELEFONE(número_do_telefone)
```

Descrição:

Efetua a geração de pulsos para a chamada de algum telefone.

Exemplo:

```
CHAMAR O TELEFONE (88316621) ;liga para o telefone 88316621
```

1.21 Chamada a rotinas e saltos

ETIQUETA

Sintaxe:

```
ETIQUETA(N) PARA nome_etiqueta
```

Descrição:

Define um etiqueta para ser usado pelo comando vá para ou chamar. O Parâmetro N não pode exceder 8.

Exemplo:

etiqueta1 para LAÇO
etiqueta2 para Teste

```
LIGAR(TUDO)
DESLIGAR(TUDO)
LAÇO

DISPLAY(1)(Teste,,)
Delay_seg(1)
LIMPAR DISPLAY
Delay_seg(1)
vá para LAÇO
```

VÁ PARA

Sintaxe:

vá para ETIQUETA

Descrição:

Volta para o ETIQUETA especificado pelo comando etiqueta.

Exemplo:

```
etiqueta LAÇO
LIMPAR DISPLAY           ;limpa o DISPLAY
Delay_seg(1)             ;temporiza 1 segundo
DISPLAY(1)(Aguarde . . .) ;Escreve "Aguarde . . ." no DISPLAY
Delay_seg (1)            ;temporiza 1 segundo
LAÇO:
LIMPAR DISPLAY           ;limpa o DISPLAY
DESLIGAR(1)
Delay_seg (1)            ;temporiza 1 segundo
LIGAR(1)
DISPLAY(1)(Rele 1 On)    ;Escreve "Aguarde . . ." no DISPLAY
Delay_seg (1)            ;temporiza 1 segundo
vá para LAÇO             ;volta ao etiqueta LAÇO
```

CHAMAR

Sintaxe:

Chamar etiqueta identificador

Descrição:

Chama uma rotina definida pelo comando etiqueta.

Exemplo

```
etiqueta1 para liga_rele      ; declara etiqueta liga_rele
  Chamar liga_rele           ; chama liga_rele
.
.
programa do usuário
.
.
FIM                          ; fim do programa do usuário

liga_rele                    ; rotina para ligar o relê
  LIGAR(1)                   ;liga o relê 1
  LIGAR(2)                   ; liga o relê 2
  LIGAR(3)                   ; liga o relê 3
  RETORNAR                   ; retorna da rotina
```

1.22 Controle da Máquina

PARAR

Sintaxe:

```
PARAR
```

Descrição:

O comando cessa todo o processamento da máquina. O funcionamento só ocorre após um reset físico.

Exemplo:

```
SE SENSOR(1) ENTÃO
  PARAR
FIM DO SE
```

RESET

Sintaxe:

```
RESET
```

Descrição:

Este tipo de comando resseta a máquina.

Exemplo:

```
SE SENSOR (1) ENTÃO
  RESET
```

SHUTDOWN

Sintaxe:

SHUTDOWN ON ou OFF

Descrição:

Caso o parâmetros seja on, a máquina desligará automaticamente após decorrido 1 minuto. Este comando é importante para economia de energia. O shutdown tanto pode estar on quanto off.

Exemplo:

SHUTDOWN ON ; após 1 minuto desliga a máquina
SHUTDOWN OFF ; desabilita o comando shutdown

SEM OPERAÇÃO

Sintaxe:

SEM OPERAÇÃO

Descrição:

Gasta um ciclo de máquina sem fazer absolutamente nada.

Exemplo:

SEM OPERAÇÃO ; gasta um ciclo de máquina sem fazer absolutamente nada

1.23 Variáveis

GENERAL

Sintaxe:

general=variável inteira de 16 bits
variável inteira de 16 bits=general
general=general+variável inteira de 16 bits
general=general-variável inteira de 16 bits
general=general*variável inteira de 16 bits(só considera a parte baixa)
general=general/variável inteira de 16 bits(só considera a parte baixa)

Descrição:

É uma variável de acesso geral para todas as variáveis inteiras do sistema. Com esta variável podemos somar o conteúdos das variáveis e realizar outras operações.

Exemplo

; Neste exemplo iremos somar o conteúdo das variáveis a% e b%
; e salvar o resultado em a%

a%=1000 ; atribui 1000 a variável a%
b%=2000 ; atribui 2000 a variável b%
general=a% ; general recebe a%
general=general+b% ; soma o conteúdo de general mais b%
a%=general ; a% recebe general

TIPOS DE VARIÁVEIS

Descrição:

O sistema possui:
oito variáveis inteiras (0 á 65535);
oito variáveis caracter;
quatro variáveis com uma casa decimal (0 á 6553,5);
quatro variáveis com duas casas decimais (0 á 655,35);

ASC\$

Sintaxe:

Variável Inteira=asc\$('a')

Descrição:

Atribui á alguma variável inteira o valor numérico da tabela ASCII do caracter.

Exemplo

a%=asc\$('A') ; atribui 65 a variável a%

CHR\$

Sintaxe:

Variável Caracter=chr\$(65)

Descrição:

Atribui á alguma variável caracter um valor até 127 da tabela ASCII.

Exemplo

a\$= chr\$(65) ; atribui o caracter 'A' a variável a\$

BIN\$

Sintaxe:

Variável Inteira=bin\$(valor binário)

Descrição:

Atribui á alguma variável inteira o valor binário do parâmetro valor binário.

Exemplo

a%=bin\$(0000000011111111) ; atribui 255 a variável a%

HEX\$

Sintaxe:

Variável Inteira=hex\$(valor hexadecimal)

Descrição:

Atribui á alguma variável inteira um valor em hexadecimal de 16 bits.

Exemplo

a%=hex\$(FFFF) ; atribui 65535 a variável a%

VARIÁVEL CARACTER

Sintaxe:

a\$="caracter"

Descrição:

Atribui algum valor a uma variável caracter.

Exemplo

a\$='a' ; a\$ recebe 'a'

NOVO

Sintaxe:

NOVO

Descrição:

Apaga a memória de programa.

LIMPAR

Sintaxe:

LIMPAR

Descrição:

Apaga a memória de dados.

RANDOM

Sintaxe:

variável=random\$

Descrição:

A função random gera um número aleatório de 8 bits que pode ser armazenado em uma variável.

Exemplo

a%=random\$; a% recebe um valor randomico da função random\$

FLAG

Sintaxe:

flag=valor lógico

Descrição:

Permite que sejam armazenados os valores lógicos para posteriores testes do programa. O total de flags são 4.

Exemplo:

```
SE SENSOR(1) ENTÃO
    flag=1
FIM DO SE

SE SENSOR(2) ENTÃO
    flag=0
FIM DO SE

.
.
.

SE FLAG1 ENTÃO
    DISPLAY(1)(Flag verdadeiro)
FIM DO SE
```

1.24 Constantes Químicas

Z

Sintaxe:

variável inteira= z(elemento químico)

Descrição:

Atribui á alguma variável inteira de 16 bits o número atômico de algum elemento químico da tabela periódica.

Exemplo

a%=z(H) ; a% recebe o número atômico de Hidrogênio

MASS

Sintaxe:

variável inteira= mass(elemento químico)

Descrição:

Atribui á alguma variável inteira de 16 bits a massa de algum elemento químico da tabela periódica.

Exemplo

a%=mass(H) ; a% recebe massa de Hidrogênio

ELET

Sintaxe:

variável inteira= elet(elemento químico)

Descrição:

Atribui á alguma variável inteira de 16 bits a eletrosfera de algum elemento químico da tabela periódica.

Exemplo

a%=elet(H) ; a% recebe a eletrosfera de Hidrogênio

TROCAR

Sintaxe:

variável%=TROCAR variável%

Descrição:

Este comando inverte a parte-alta da variável pela baixa e vice-versa.

Exemplo

```
a%=255 ;carrega a% com 255 decimal  
a%=TROCARf a% ; inverte a parte-alta pela baixa
```

NÃO

Sintaxe:

variável%=NÃO variável%

Descrição:

Este comando inverte o estado de todos os bits da variável.

Exemplo

```
a%=100 ;carrega a% com 100 decimal  
a%=NÃO a% ; lógica NÃO
```

1.25 Controle da Saída

SAIDA=ENTRADA

Sintaxe:

SAIDA=ENTRADA

Descrição:

Movimenta o conteúdo da porta de entrada para a saída.

Exemplo:

SAIDA=ENTRADA

COMPLEMENTAR

Sintaxe:

COMPLEMENTAR(SAÍDA)

Descrição:

Modifica o estado atual de todas as 6 saídas dispostas na placa PCL 1003.

Exemplo:

COMPLEMENTAR(SAÍDA) ;modifica todas as saídas. O que estava ligado agora está
;desligado e vice-versa.

SAÍDA=

Sintaxe:

SAÍDA=parâmetro

Descrição:

Carrega o valor especificado em parâmetro.

Exemplo:

SAÍDA=63 ;liga todos os relês
SAÍDA=1 ;liga somente o relê 1

DECREMENTAR

Sintaxe:

DECREMENTAR SAÍDA

Descrição:

Decrementa de uma unidade a saída.

Exemplo:

DECREMENTAR SAÍDA ;decrementa o valor anterior de 1.

INCREMENTAR

Sintaxe:

INCREMENTAR SAÍDA

Descrição:

Incrementa de uma unidade a saída.

Exemplo:

INCREMENTAR SAÍDA ;decrementa o valor anterior de 1.

SAÍDA

Sintaxe:

SAÍDA(número_da_saída)

Descrição:

Verifica o estado atual de alguma saída informada em número_da_saída. Número_da_saída deve ser um número entre 1 e 6.

Exemplo:

```
SE SAÍDA(1) ENTÃO
  DISPLAY(1)(Rele 1 esta)
  DISPLAY(2)(Ligado.)
FIM DO SE
```

```
SE SAÍDA(1) ENTÃO
  DISPLAY(1)(Rele 1 esta)
  DISPLAY(2)(Ligado)
SENÃO
  DISPLAY(1)(Rele 1 esta)
  DISPLAY(2)(Desligado)
FIM DO SE
```

LED

Sintaxe:

LIGAR(número_do_led)
DESLIGAR(número_do_led)

Descrição:

Liga (LIGAR) ou desliga (DESLIGAR) o led especificado em número_do_led. Número_do_led pode ser 1 ou 2.

Exemplo:

```
SE SENSOU(1) ENTÃO
  LIGAR(led1) ;se SENSOU(1) acionado...
  DESLIGAR(led2) ;autOUiza a passagem no acesso.
LIMPAR DISPLAYLAY
DISPLAY(1)(SENSOU 1 on)
```

```
        DISPLAY(2)(Acesso AutOUizado)
        Delay_seg(1)
SENÃO                                     ;senão....
        LIGAR(led2)                       ;nega a passagem no acesso.
        DESLIGAR(led1)
        LIMPAR DISPLAYLAY
        DISPLAY(1)(SENSOU 1 off)
        DISPLAY(2)(Acesso Negado)
        Delay_seg(1)
FIM DO SE
```

SE SAÍDA\$=

Sintaxe:

SE SAÍDA\$=valor ENTÃO

Descrição:

Este comando verifica se a saída de dados corresponde ao valor especificado em valor.

Exemplo:

```
SE SAÍDA$=1 ENTÃO                       ;se a saída 1 estiver fechada...
    TRANSMITIR(100 pulsos!)             ;executa comando
    LIMPAR CONTADOU DE PULSOS$
FIM DO SE
```

Capítulo 2

Controle de Fluxo

2.1 Laço ENQUANTO

FAÇA - ENQUANTO SENSOR

Sintaxe:

```
FAÇA
    .
    comando
    .
    .
ENQUANTO SENSOR(número_da_entrada_digital)
```

Descrição:

Testa SENSOR(número_da_entrada_digital) após executar, ao menos uma vez, o que está após de “FAÇA”. Enquanto, o SENSOR(número_da_entrada_digital) for verdadeiro, volta a executar a partir de “FAÇA”, se a condição for falsa passa a executar o que está após o comando ENQUANTO.

Exemplo:

```
FAÇA
    TRANSMITIR(Estrutura posfixada)
    LIMPAR DISPLAYLAY
    LIGAR(le1)
```

```
LIGAR(TUDO)
Delay_seg(4)
DESLIGAR(TUDO)
ENQUANTO SENSO(1)
```

FAÇA - ENQUANTO NÃO SENSOR

Sintaxe:

```
FAÇA
.
.
comando
.
.
ENQUANTO SENSOR NÃO (número_da_entrada_digital)
```

Descrição:

Testa SENSOR (número_da_entrada_digital) após executar, ao menos uma vez, o que está após de “FAÇA”. Enquanto, o SENSOR(número_da_entrada_digital) for falso, volta a executar a partir de “FAÇA”, se a condição for verdadeira passa a executar o que está após o comando ENQUANTO.

Exemplo:

```
FAÇA
TRANSMITIR(Estrutura posfixada)
LIMPAR DISPLAYLAY
LIGAR(led1)
LIGAR(TUDO)
Delay_seg(1)
DESLIGAR(TUDO)
ENQUANTO NÃO SENSOR(1)
```

ENQUANTO SENSOR...LAÇO

Sintaxe:

```
ENQUANTO SENSOR (número_da_entrada_digital) FAÇA
.
.
comando
.
.
LAÇO
```

Descrição:

Testa primeiro o SENSOR (número_da_entrada_digital). E caso verdadeiro, executa os comando.

Exemplo:


```
ENQUANTO SENSOU(8) FAÇA
    LIGAR(1)
    LIMPAR DISPLAYLAY
    DISPLAY(1)(Rele 1 on)
    tempo_seg(6)
    DESLIGAR(1)
    LIMPAR DISPLAYLAY
    DISPLAY(1)(Rele 1 off)
    tempo_seg(6)
LAÇO
```

ENQUANTO NÃO SENSOR...FAÇA

Sintaxe:

```
ENQUANTO SENSOR(número_da_entrada_digital) FAÇA
    .
    .
    comando
    .
    .
LAÇO
```

Descrição:

Testa primeiro o SENSOR (número_da_entrada_digital). E quando falso, executa os comando.

Exemplo:

```
ENQUANTO NÃO SENSOR(8) FAÇA
    LIGAR(1)
    LIMPAR DISPLAYLAY
    DISPLAY(1)(Rele 1 on)
    Delay_seg(6)
    DESLIGAR(1)
    LIMPAR DISPLAYLAY
    DISPLAY(1)(Rele 1 off)
    Delay_seg(6)
LAÇO
```

ENQUANTO ... LAÇO

Sintaxe:

```
ENQUANTO variável FAÇA
    comando
LAÇO
```

Descrição:

Enquanto a variável for diferente de 0, comando serão executados.-

Exemplo

```
a%=10
ENQUANTO a% do
    TRANSMITIR(PCL 1001)
    a%=a%-1
LAÇO
```

2.2 Laço REPITA

REPITA...LAÇO DO REPITA

Sintaxe:

```
REPITA(número_de_repetições)
.
.
comando
.
.
LAÇO DO REPITA
```

Descrição:

Repete os comando pelo número de vezes especificado em número_de_repetições. O número_de_repetições deve ser maior que zero e menor ou igual a 65535.

Exemplo:

```
REPITA(5)
    LIGAR(1)
    Delay_seg(1)
    DESLIGAR(1)
    Delay_seg(1)
    SE SENSOR(2) ENTÃO
        LIMPAR DISPLAY
        Delay_seg(1)
        DISPLAY(1)(Saindo do REPITA)
        Delay_seg(1)
        SAIR DO REPITA
    FIM DO SE
    SE SENSOR(3) ENTÃO
        RESTAURAR
    FIM DO SE
    LIGAR(TUDO)
    Delay_seg(1)
```

;liga e desliga o relé 1 5 vezes.

;se SENSOR(2) acionado
;sai do REPITA

;se SENSOR(3) volta para REPITA

```
DESLIGAR(TUDO)
Delay_seg(1)
LAÇO DO REPITA
```

REPITA...LAÇO REPITA

Sintaxe:

```
REPITA(variável)
    comando
LAÇO REPITA
```

Descrição:

O REPITA permite que uma variável seja utilizada para o seu funcionamento. Para isso, siga o exemplo abaixo.

Exemplo:

```
a%=100                                ;carrega a% com 100 decimal
REPITA(a%)                             ; executa estes comando 100 vezes
    LIGAR(1)
    tempo_ms(1500)
    DESLIGAR(1)
    tempo_ms(1500)
LAÇO REPITA
```

2.3 Seleção de Casos

SELEÇÃO DE CASO

Sintaxe:

```
SELEÇÃO DE CASO variável
    CASO 1
        comando 1
        .
        .
        TRAVAR
    CASO n
        comando n
        .
        .
        TRAVAR
    SENÃO SELEÇÃO
        comando 3
        .
        TRAVAR
FIM select
```

Descrição:

Faz uma seleção de casos para teste. Caso um seja verdadeiro, os comando entre o CASO e o TRAVAR serão executados. Esta estrutura funciona para variáveis character, variáveis inteiras e para a recepção de dados.

Exemplo

```
a%=100
SELEÇÃO DE CASO variável
  CASO 100
    TRANSMITIR(a%)
    TRAVAR
  CASO 200
    TRANSMITIR(b%)
    TRAVAR
SENÃO select
  TRANSMITIR(Nenhum e igual)
FIM SELEÇÃO
```

Capítulo 3

Operadores Lógicos e Estruturas Condicionais Lógicas

3.1 Operadores

E

Sintaxe:

variável%=variável% E constante

Descrição:

Este comando realiza uma lógica E entre a variável e a constante.

Exemplo

```
a%=100 ;carrega a% com 100 decimal
a%=a% E 1 ; lógica e com 1
```

OU

Sintaxe:

variável%=variável% OU 1

Descrição:

Este comando realiza uma operação OU entre a variável e uma constante.

Exemplo

```
a%=100 ;carrega a% com 100 decimal
a%=a% OU 1 ; lógica OU com 1
```

XOR

Sintaxe:

variável%=variável% xor constante

Descrição:

Este comando realiza uma operação xor entre a variável e a constante.

Exemplo

```
a%=100                ;carrega a% com 100 decimal
a%=a% xor 1          ; lógica xor com 1
```

3.2 Estruturas Condicionais Lógicas

SE E COM...

Sintaxe:

se e com SENSOR(número_entrada_digital 1), SENSOR(número_entrada_digital 2),
SENSOU(número_entrada_digital n), ENTÃO

```
.
.
.
comando 1
```

```
.
.
.
```

SENÃO

```
.
.
.
comando 2
```

```
.
.
.
```

FIM DO SE

Descrição:

Operação E com as entradas digitais. O parâmetro n, não pode passar de 6 já que este é a última entrada digital. Se a operação E com as entradas for verdadeira, os comando 1 serão executados, caso contrário, os comando 2 serão executados.

Exemplo:

```
SE E COM SENSOR(1),SENSOR(2), ENTÃO                ;Lógica e com SENSOU(1) e SENSOU(2)
  LIMPAR DISPLAYLAY                                ;caso verdadeiro, executa estes comando...
  Delay_seg(1)
  DISPLAY(1)(SENSOR 1 fechado)
  TRANSMITIR(SENSOR 1 fechado)
  Delay_seg(1)
SENÃO                                               ;caso falso executa o que está após SENÃO
  LIMPAR DISPLAYLAY
  Delay_seg(1)
  DISPLAY(1)(SENSOU 1 aberto)
  TRANSMITIR(SENSOU 1 aberto)
  Delay_seg(1)
```

FIM DO SE

SE OU COM...

Sintaxe:

SE OU COM SENSOR(número_entrada_digital 1), SENSOR(número_entrada_digital 2),
SENSOR(número_entrada_digital n), ENTÃO

.
. .
comando 1

.
. .
.

SENÃO

.
. .
comando 2

.
. .
.

FIM DO SE

Descrição:

Operação OU com as entradas digitais. O parâmetro n, não pode passar de 6 já que este é a última entrada digital. Se a operação OU com as entradas for verdadeira, os comando 1 serão executados, caso contrário, os comando 2 serão executados.

Exemplo:

SE OU COM SENSOU(1),SENSOU(2), ENTÃO ;Lógica ou com SENSOU(1) e SENSOU(2)
LIMPAR DISPLAY ;se verdadeiro, executa estes comando...

Delay_seg(1)
DISPLAY(1)(SENSOU 1 fechado)
TRANSMITIR(SENSOU 1 fechado)
Delay_seg(1)

SENÃO ;caso falso executa o que está após SENÃO

LIMPAR DISPLAYLAY
Delay_seg(1)
DISPLAY(1)(SENSOU 1 aberto)
TRANSMITIR(SENSOU 1 aberto)
Delay_seg(1)

FIM DO SE

SE NÃO E COM...

Sintaxe:

SE NÃO E COM SENSOR(número_entrada_digital 1), SENSOR(número_entrada_digital 2), SENSOR(número_entrada_digital n), ENTÃO

.
.
.
comando 1
.
.
.

SENÃO

.
.
.
comando 2
.
.
.

FIM DO SE

Descrição:

Operação E com as entradas digitais. O parâmetro n, não pode passar de 6 já que este é a última entrada digital. Se a operação E com as entradas for verdadeira, os comando 1 serão executados, caso contrário, os comando 2 serão executados.

Exemplo:

```
SE E COM SENSOU(1),SENSOU(2), ENTÃO           ;Lógica e com SENSOU(1) e SENSOU(2)
  LIMPAR DISPLAY                               ;caso verdadeiro, executa estes comando...
  Delay_seg(1)
  DISPLAY(1)(SENSOU 1 fechado)
  TRANSMITIR(SENSOU 1 fechado)
  Delay_seg(1)
SENÃO                                           ;caso falso executa o que está após SENÃO
  LIMPAR DISPLAYLAY
  Delay_seg(1)
  DISPLAY(1)(SENSOU 1 aberto)
  TRANSMITIR(SENSOU 1 aberto)
  Delay_seg(1)
FIM DO SE
```

Se NÃO OU COM...

Sintaxe:

SE OU COM SENSOR(número_entrada_digital 1), SENSOR(número_entrada_digital 2),
SENSOR(número_entrada_digital n), ENTÃO

.
.
.
comando 1
.
.
.

SENÃO

.
 .
 . comando 2
 .

FIM DO SE

Descrição:

Operação OU com as entradas digitais. O parâmetro n, não pode passar de 6 já que este é a última entrada digital. Se a operação OU com as entradas, for verdadeira, os comando 1 serão executados, caso contrário, os comando 2 serão executados.

Exemplo:

```
SE OU COM SENSOU(1),SENSOU(2), ENTÃO           ;Lógica ou com SENSOU(1) e SENSOU(2)
  LIMPAR DISPLAY                               ;se verdadeiro, executa estes comando...
  Delay_seg(1)
  DISPLAY(1)(SENSOU 1 fechado)
  TRANSMITIR(SENSOU 1 fechado)
  Delay_seg(1)
SENÃO                                           ;caso falso executa o que está após SENÃO
  LIMPAR DISPLAYLAY
  Delay_seg(1)
  DISPLAY(1)(SENSOU 1 aberto)
  TRANSMITIR(SENSOU 1 aberto)
  Delay_seg(1)
FIM DO SE
```

Capítulo 4

Exemplos de Programação

Neste capítulo iremos estudar alguns exemplos de programação com a placa PLC 1001. Observe que em cada comando, há uma linha de comentários para facilitar o entendimento da mesma.

Exemplo 1 - Oscilando uma saída de relê

Este programa ficará ligando e desligando um relê interminavelmente. O tempo entre o ligar e desligar é de 1 segundo.

```

10 ; Programa para oscilar a Saída de um relê
20 ; Autor: Cerne Tecnologia
30 ; Data: 28/01/2005
40
50
60 LIGAR(1) ; liga a saída 1
70 tempo_ms(1000) ; conta um tempo de 1 segundo
80 DESLIGAR(1) ; desliga a saída 1
90 tempo_ms(1000) ; conta um tempo de 1 segundo
100 PULAR PARA O INICIO ; volta para o início do programa

```

Exemplo 2 - Botão e Led

Este programa irá acender o led1 da placa assim que a entrada 1 ficar verdadeira.

```

10 ; Programa Botão e Led
20 ; Autor: Cerne Tecnologia
30 ; Data: 28/01/2005
40
50 SE SENSOU(1) ENTÃO ; entrada 1 está ativa?
60 LIGAR(led1) ; sim, então liga led1
70 FIM DO SE ; fim do se
80
90 SE NÃO SENSOU(1) ENTÃO ; entrada 1 está aberta?
100 DESLIGAR(led1) ; sim, então desliga led1
110 FIM DO SE ; fim do se
120
130 PULAR PARA O INICIO ; volta para o
início

```

Exemplo 3 - Mensagem no DISPLAY

Este programa irá apresentar uma mensagem no DISPLAY LCD da placa

```

10 ; Programa Mensagem no DISPLAY
20 ; Autor: Cerne Tecnologia
30 ; Data: 28/01/2005
40
50 DISPLAY(1)(PCL 1001) ; mostra na linha 1 "PCL 1001"
60 DISPLAY(2)(Projeto Nacional) ; mostra na linha 2 "Projeto..."

```

Exemplo 4 - Mensagem no DISPLAY e Entradas

Este programa irá apresentar uma mensagem no DISPLAY LCD caso a entrada 3 seja acionada.

```

10 ; Programa que apresenta mensagem se chave estiver on
20 ; Autor: Cerne Tecnologia
30 ; Data: 28/01/2005
40
50 SE SENSOR(5) ENTÃO
60     LIMPAR DISPLAYLAY
70     DISPLAY(1) (Chave 5 on!)
80     tempo_ms(1000)
90     LIMPAR DISPLAYLAY
90 FIM DO SE
100
110 PULAR PARA O INICIO

```

Exemplo 5 - Rotacionar uma mensagem no DISPLAY

Este exemplo apresenta uma mensagem no DISPLAY e depois fica rotacionando a mesma.

```

10 ; Programa para rotacionar mensagens no DISPLAY
20 ; Autor: Cerne Tecnologia
30 ; Data: 28/01/2005
40
50 etiquetal para novamente ; define um etiqueta
60
70 DISPLAY(1) (PCL 1001) ; mostra na linha 1 a mensagem
90 novamente
100 ROTACIONAR DISPLAY PARA DIREITA
110 tempo_ms(500)
120 ROTACIONAR DISPLAY PARA DIREITA
130 tempo_ms(500)
140 ROTACIONAR DISPLAY PARA DIREITA
150 tempo_ms(500)
160 ROTACIONAR DISPLAY PARA ESQUERDA
170 tempo_ms(500)
180 ROTACIONAR DISPLAY PARA ESQUERDA
190 tempo_ms(500)
200 ROTACIONAR DISPLAY PARA ESQUERDA
210 tempo_ms(500)
220 vá para novamente

```

Exemplo 6 - Transmissão de Dados pelo Canal Serial

Este programa irá transmitir dados pelo canal serial continuamente a uma taxa de 9600 bps.

```

10 ; Programa de Transmissão de Dados pelo Canal Serial
20 ; Autor: Cerne Tecnologia
30 ; Data: 28/01/2005
40
50 TRANSMITIR(PCL 1003) ; transmite "PCL 1003"
60 tempo_ms(1000) ; espera 1000 ms
70 PULAR PARA O INICIO ; volta para 0

```

Exemplo 7 - Recepção de Dados pelo Canal Serial

Este programa irá receber dados do canal serial e dependendo do caracter que receber irá ligar/desligar um dos 2 relês da placa e display uma mensagem no DISPLAY.

```

10 ; Programa de Recepção de Dados
20 ; Autor: Cerne Tecnologia
30 ; Data: 28/01/2005
40
50
60 SE recepção de dados="A" ENTÃO ; caracter recebido é o "A"?
70     LIGAR(1) ; sim, então liga o relê 1
80     DISPLAY(1)(Rele 1 On) ; mostra mensagem na linha 1
90 FIM DO SE ; fim do se
100
110 SE recepção de dados="B" ENTÃO ; caracter recebido é o "B"?
120     DESLIGAR(1) ; sim, então desliga o relê 1
130     DISPLAY(1)(Rele 1 Off) ; mostra mensagem na linha 1
140 FIM DO SE ; fim do se
170
150 SE recepção de dados="C" ENTÃO ; caracter recebido é o "C"?
180     LIGAR(2) ; sim, então liga o relê 2
190     DISPLAY(1)(Rele 2 On) ; mostra mensagem na linha 1
200 FIM DO SE ; fim do se
210
220 SE recepção de dados="D" ENTÃO ; caracter recebido é o "D"?
230     DESLIGAR(2) ; sim, então desliga o relê 2
240     DISPLAY(1)(Rele 2 Off) ; mostra mensagem na linha 1
250 FIM DO SE ; fim do se
260
270 PULAR PARA O INICIO ; volta para início do programa

```

Exemplo 8 - Utilizando a Data e Hora da Placa

A PCL 1001 possui internamente um relógio de data e hora que permite ao usuário controlar a data e hora do dia. Este relógio de alta precisão já tem tratamento para ano bissexto. Neste próximo exemplo iremos apresentar a data e hora no DISPLAY. Quando a hora passar um valor, um dos relês irá acionar para liga por exemplo uma sirene.

```

10 ; Programa de Controle de Data e Hora
20 ; Autor: Cerne Tecnologia
30 ; Data: 28/01/05
40
50 DISPLAY(1)(DATA$) ; mostra na linha 1 a data
60 DISPLAY(2)(HOUA$) ; mostra na linha 2 a hora
70
80 SE HORA$>"12_00" ENTÃO ; se passou do meio-dia então...
90     LIGAR(1) ; liga o relê 1
100 FIM DO SE ; fim do se
110
120 PULAR PARA O INICIO ; volta para início do programa

```

Exemplo 9 - Utilizando o Cronômetro

Neste exemplo veremos um exemplo para medir um determinado intervalo de tempo com o comando cronometro\$.

```

10 ; Programa para medir um determinado intervalo de tempo
20 ; Autor: Cerne Tecnologia

```

```

30 ; Data: 28/01/05
40
50 SE SENSOR(1) ENTÃO ; entrada 1 está fechada?
60 LIMPAR CRONÔMETRO ; sim, então limpa o cronômetro
70 FIM DO SE ; fim do se
80
90 SE SENSOR(2) ENTÃO ; entrada 2 está fechada?
100 INICIAR CRONÔMETRO ; sim, então inicia o timer 2
110 FIM DO SE
120
130 SE SENSOU(3) ENTÃO ; entrada 3 está fechada?
140 PARAR CRONÔMETRO ; sim, então pára o cronômetro
150 FIM DO SE ; fim do se
160
170 DISPLAY(1)(Medicao do Tempo); mostra mensagem no DISPLAY
180 DISPLAY(2)(CRONÔMETRO $) ; mostra na linha 2 a medição
190 PULAR PARA O INICIO ; volta para o início

```

Exemplo 10 - Implementando um controle de semáforo

Neste exemplo iremos implementar um controle de semáforo.

```

LIMPAR DISPLAY ; limpa o DISPLAYlay
DESLIGAR(TUDO) ; desliga todos os relés
DISPLAY(1)(Semaforo) ; escreve no mensagem
DISPLAY(2)(Verde) ; no DISPLAY
LIGAR(1) ; liga o relé 1
TRANSMITIR(Semaforo Verde Acionado) ; envia para o canal serial
tempo_seg(10) ; Espera 10 s
DESLIGAR(1) ;desliga o semáFOUo verde
LIMPAR DISPLAY ;limpa o DISPLAYlay
DISPLAY(1)(Semaforo) ;escreve mensagem
DISPLAY(2)(Amarelo)
LIGAR(2) ;liga o relê 2
TRANSMITIR(Semaforo Amarelo Acionado) ; transmite dados
tempo_seg(2) ;aguarda 2 segundos
DESLIGAR(2) ;desliga o relé 2
LIMPAR DISPLAY ;limpam lcd
DISPLAY(1)(Semaforo) ; mostra mensagem
DISPLAY(2)(Vermelho) ; no DISPLAY
LIGAR(3) ; liga o relê 3
TRANSMITIR(Semaforo Vermelho Acionado) ; transmite dados
tempo_seg(5) ; espera 5 segundos
PULAR PARA O INICIO ; volta para o início

```

Exemplo 11 - Medindo Velocidades Externas

Este exemplo demonstra como deve-se proceder para medir uma velocidade externa.

```

10 ; Programa de Medição de Velocidade Externa

```

```

20 ; Autor: Cerne Tecnologia
30 ; Data: 28/01/05
40
50 ESPAÇO PADRÃO=200 ; distância entre os sensores é de 100m
120 LIMPAR DISPLAYLAY ; limpa o DISPLAY
60 DISPLAY(1) (Medindo...) ; mostra mensagem
70 VELOCIDADE ON ; habilita a medição de velocidade
80 LIMPAR DISPLAYLAY ; limpa o lcd
90 DISPLAY(1) (velocidade$); mostra o valor da medição de
velocidade
100 tempo_ms(2000) ; aguarda 2 segundos
130 PULAR PARA O INICIO ; volta para o início do programa

```

Exemplo 12 - Contador de Pulsos

Este exemplo irá demonstrar os passos necessários para a contagem de pulsos externos.

```

10 ; Programa para medição de pulsos externos
20 ; Autor: Cerne Tecnologia
30 ; Data: 28/01/2005
40
50 ENTRADA6 COMO CONTADOR DE PULSOS$ ; config. entrada 6 como
contador
60 etiquetal para again ; define etiquetal
100 INICIAR CONTADOU DE PULSOS$
190
120 again
110
70 DISPLAY(1) (CONTADOR DE PULSOS$) ;apresenta o contadOU de
pulsos
80
170 SE SENSOR(1) ENTÃO ; entrada 1 é verdadeira?
130 LIMPAR CONTADOR DE PULSOS$ ; sim, então limpa
contadOU
140 FIM DO SE ; fim do se
150
160 SE SENSOR(2) ENTÃO ; entrada 2 é verdadeira?
200 PARAR CONTADOR DE PULSOS$ ; sim, então pára a
contagem
210 FIM DO SE ; fim do se
220
230 vá para again ; salta para again

```

Exemplo 13 - Frequencímetro

Esta placa vêm equipada com um medidor de frequência externo. Vejamos um exemplo deste fantástico módulo.

```

10 ; Programa para medição de frequências externas
20 ; Autor: Cerne Tecnologia
30 ; Data: 28/01/2005
40
50 ENTRADA6 COMO FREQUENCÍMETRO ; config. entrada6 como
;frequencímetro
60 etiquetal para again ; define etiqueta 1
70
80 again
90

```

```
100      DISPLAY(1)(freq$) ; apresenta a frequencia medida
110      vá para again      ; salta para again
```

Exemplo 14 - Módulo Contador

Este módulo se parece um pouco com o contador de pulsos, diferindo que neste a recarga de dados é automática além de a informação de contagem encerrada ser informada por meio de flags.

```
10      ; Programa Contador de Pulsos externo com recarga
; automática
20      ; Autor: Cerne Tecnologia
30      ; Data: 28/01/2005
40
50      ENTRADA6 COMO CONTADOR
60      contador(1000)
70      etiqueta1 para again
80
90      again
100     SE CONTADOR ENTÃO
110         DISPLAY(1)(Contou 200 pulsos!)
120         tempo_ms(2000)
130         LIMPAR DISPLAYLAY
140     FIM DO SE
150     vá para again
```

Exemplo 15 - Chamando um Telefone

Este exemplo demonstra como deve ser feito para se fazer uma chamada de algum telefone pela linha telefônica. Neste caso, toda vez que a entrada 4 fOU pressionada haverá uma chamada de telefone via o relê 1.

```
10      ; Programa para demonstrar o comando para chamar um
; telefone
20      ; Autor: Cerne Tecnologia
30      ; Data: 28/01/2005
40
50      SE SENSOR(4) ENTÃO ; chave 4 está on?
60         DISPLAY(1)(Chamando Telefone)
100        chamar o telefone(88316621) ; sim, então chama
120        LIMPAR DISPLAY
70      FIM DO SE
80
90      PULAR PARA O INICIO
```

Exemplo 16 - Implementando um Jogo de Palavras

A PCL1003 vem equipada com um jogo educativo de palavras. Esta pode ser acessada através do comando JOGO. Caso o jogador tenha ganho a partida, o comando JOGO será positivo, caso contrário negativo.

```
10      ; Programa para demonstrar a utilização do jogo de
; palavras
20      ; Autor: Cerne Tecnologia
```

```

30 ; Data: 28/01/2005
40
50 DISPLAY(1)(Boa Partida!) ; apresenta mensagem
60 tempo_ms(2000) ; espera 2 segundos
70 JOGO ; chama o jogo
80
100 SE NÃO JOGO ENTÃO ; se jogador perdeu...
110 LIMPAR DISPLAYLAY ; limpa o DISPLAYlay
120 DISPLAY(1)(Voce perdeu...) ; apresenta mensagem
130 tempo_ms(2000) ; espera 2 segundos
140 FIM DO SE ; fim do se
150
160 SE JOGO ENTÃO ; se jogador ganhou...
170 LIMPAR DISPLAYLAY ; limpa o DISPLAY
180 DISPLAY(1)(Voce Ganhou!) ; apresenta mensagem
190 tempo_ms(3000) ; espera 2 segundos
200 FIM DO SE ; fim do se
210
240 LIMPAR DISPLAY ; limpa o lcd
250 DISPLAY(1)(Deseja Continuar?) ; apresenta mensagem
260
220 FAÇA ; faça
230 SEM OPERAÇÃO ; sem operação
270 ENQUANTO NÃO SENSOU(4) ; enquanto a chave 4
; estiver aberta
280 PULAR PARA O INICIO ; salta para o endereço 0

```

Exemplo 17 - Atribuindo valores e constantes as variáveis

Neste exemplo veremos como deve se proceder para atribuir um valor ou uma constantes a algumas variáveis dos sistema como as inteiras, as de duas casas decimais e uma casa decimal.

```

10 ; Programa de atribuição de valores as variáveis
20 ; Autor: Cerne Tecnologia
30 ; Data: 29/01/2005
40
50 a@=pi ; atribui a constante pi a variável
60 b@=e ; atribui a constante e(exponencial) a variável
70 c@=g ; atribui a constante g(gravidade) a variável 80
80
90 DISPLAY(1)(Variavel a@ ) ; apresenta mensagem
100 DISPLAY(2)(a@) ; apresenta conteúdo da variável
110 tempo_ms(2000) ; espera tempo
120
130 DISPLAY(1)(Variavel b@ ) ; apresenta mensagem
140 DISPLAY(2)(b@) ; apresenta conteúdo da variável
150 tempo_ms(2000) ; espera tempo
160
170 DISPLAY(1)(Variável c@ ) ; apresenta mensagem
190 DISPLAY(2)(c@) ; apresenta conteúdo da variável

```

Exemplo 18 - Calculando a hipotenusa de um triângulo

Neste exemplo iremos utilizar um pouco as funções matemáticas que PCL nos oferece. Começemos pelo cálculo da hipotenusa.


```
10 ; Programa para Calcular a hipotenusa de ângulo
20 ; Autor: Cerne Tecnologia
30 ; Data: 28/01/2005
40
50 ; Atenção!!!
60 ; a hipotenusa é o a% enquanto os catetos
70 ; são o b% e o c%.
80
90 b%=6 ; carrega um cateto
100 c%=10 ; carrega outro cateto
110 a%=hipot(b%,c%) ; calcula a hipotenusa
120 DISPLAY(1)(a%) ; apresenta o valor calculado
```

Exemplo 19 - Calculando o Perímetro e a Área de um Círculo

Vamos agora demonstrar as diretrizes para calcular o Perímetro e a Área de um círculo.

```
10 ; Programa para Cálculo de Perímetro e Área de um Círculo
20 ; Autor: Cerne Tecnologia
30 ; Data: 28/01/2005
40
50 ; Atenção!!!
60 ; O raio deve estar previamente
70 ; carregado em b%!
80
90 b%=3 ; raio igual á 3
100 a@=circunf(b%) ; calcula a circunferência
110 DISPLAY(1)(a@) ; apresenta resultado
120 a@=area_circunf(b%) ; calcula a área
130 DISPLAY(2)(a@) ; apresenta resultado
```

Exemplo 20 - Calculando o Fatorial de um Número

Neste tópico iremos calcular o fatorial de um número. O máximo suportado pela placa é até o fatorial de 8.

```
10 ; Programa para Cálculo de Fatorial
20 ; Autor: Cerne Tecnologia
30 ; Data: 28/01/2005
40
50 a%=0!
60 DISPLAY(1)(a%)
70 tempo_ms(1000)
80
90 b%=1!
100 DISPLAY(1)(b%)
110 tempo_ms(1000)
120
130 c%=4!
140 DISPLAY(1)(c%)
150 tempo_ms(1000)
```

Exemplo 21 - Soma, Subtração, Produto e Divisão com constantes

Neste exemplo iremos somar valores constante as variáveis.

```
10 ; Programa que demonstra as quatro operações básicas
```

```

20 ; Autor: Cerne Tecnologia
40 ; Data: 29/01/2005
50
50 a%=1000 ; atribui 1000 á variável
60 DISPLAY(1)(Antes da soma) ; apresenta mensagem
70 DISPLAY(2)(a%) ; apresenta variável
80 tempo_ms(2000) ; espera um tempo
90
100 a%=a%+1500 ; soma 1500 a variável
110 DISPLAY(1)(Depois da soma) ; apresenta mensagem
120 DISPLAY(2)(a%) ; apresenta variável
130 tempo_ms(2000) ; espera um tempo
140
150 a%=a%-200 ; subtrai 200 da variável
170 DISPLAY(1)(Apos Subtracao) ; apresenta mensagem
180 DISPLAY(2)(a%) ; apresenta variável
190 tempo_ms(2000) ; espera tempo
200
210 a%=a%*4 ; multiplica o último valOU pOU 4
220 DISPLAY(1)(Apos Multipli) ; apresenta mensagem
230 DISPLAY(2)(a%) ; apresenta variável
240 tempo_ms(2000) ; espera tempo
260
250 a%=a%/2 ; divide a% pOU 2
270 DISPLAY(1)(Apos Divisao) ; apresenta mensagem
280 DISPLAY(2)(a%) ; apresenta variável

```

Exemplo 22 - Soma, Subtração, Produto e Divisão com Variáveis

Agora iremos fazer as mesmas operações acima, porém entre as variáveis.

```

10; Programa que demonstras as quatro operações com Variáveis
20; Autor: Cerne Tecnologia
30; Data: 29/01/2005
40
50 a%=1000 ; atribui a variável a% 1000
60 general=a% ; a variável geral é igual a 1000
70 b%=200 ; atribui 200 a variável b%
80 general=general+b% ; soma general mais b%
90 a%=general ; a% recebe general
100 DISPLAY(1)(a%) ; o resultado é 1200
110 tempo_ms(2000) ; aguarda 2 segundos
120
130 c%=500 ; atribui 500 a variável c%
140 general=a% ; general recebe a%
150 general=general-c% ; subtrai general menos c%
160 a%=general ; a% recebe general
170 DISPLAY(1)(a%) ; o resultado é 700
180 tempo_ms(2000) ; aguarda 2 segundos
190
200 d%=4 ; atribui 4 a variável d%
210 general=a% ; general recebe a%
220 general=general * d% ; multiplica general * d%
230 a%=general ; a% recebe general
240 DISPLAY(1)(a%) ; o resultado é 2800
250 tempo_ms(2000) ; aguarda 2 segundos
260
270 e%=10 ; atribui 10 a variável
280 general=a% ; general recebe a%
290 general=general / e% ; divide general pOU e%
300 a%=general ; a% recebe general

```

```
310 DISPLAY(1)(a%) ; o resultado é 280
```

Exemplo 23 - Calculando a Porcentagem

Este exemplo permitirá que se calcule a porcentagem de uma variável mais um valor definido por uma constante.

```
10 ; Programa para Calculo de Porcentagem
20 ; Autor: Cerne Tecnologia
30 ; Data: 29/01/2005
40
50 a%=1000 ; atribui 1000 a a%
60 a%=a%+40% ; a% é igual a 1400
70 DISPLAY(1)(a%) ; apresenta o valOU do cálculo
80 tempo_ms(1000) ; aguarda tempo
90 b%=10000 ; atribui 10000 a a%
100 b%=b%+70% ; a% é igual a 17000
110 DISPLAY(1)(b%) ; apresenta o valOU do cálculo
```

Exemplo 24 - Tira a Raiz de um número

Este exemplo irá calcular a raiz de um número e apresentar no DISPLAY após o cálculo.

```
10 ; Programa para retirar a raiz quadrada de um número
20 ; Autor: Cerne Tecnologia
30 ; Data: 29/01/2005
40
50 a%=100 ; atribui 100 a a%
60 a%=raiz a% ; calcula a raiz e salva em a%
70 DISPLAY(1)(a%) ; apresenta o resultado
```

Exemplo 25 – Calculando a Potência de um Número

Neste exemplo iremos calcular a potência de um número mediante ao valor atribuído á variável e elevaremos o mesmo á uma constante.

```
10 ; Programa para Calculo de potência
20 ; Autor: Cerne Tecnologia
30 ; Data: 29/01/2005
40
50 a%=2 ; atribui 2 a variável
60 a%=a% ^ 20 ; eleva o número a 10
70 DISPLAY(1)(a%) ; o resultado é 1024
```

Exemplo 26 - Testes condicionais com variáveis

Neste exemplo iremos fazer testes com os valores armazenados nas variáveis.

```
10 ; Programa de Teste Condicional com Variáveis
20 ; Autor: Cerne Tecnologia
30 ; Data: 29/01/2005
40
50 f%=101 ; atribui 100 a variável
60
70 SE f%<100 ENTÃO ; se for menor que 100...
```

```

80          DISPLAY(1) (E menor que 100) ; apresenta mensagem
90          tempo_ms(1000)                ; espera 1 segundo
100 FIM DO SE                               ; fim do se
110
120 SE f%>100 ENTÃO                          ; se for maior que 100...
130     DISPLAY(1) (E maioe que 100) ; apresenta mensagem
140     tempo_ms(1000)                ; espera 1 segundo
150 FIM DO SE                               ; fim do se
160
170 SE f%=100 ENTÃO                          ; se for igual a 100...
180     DISPLAY(1) (E igual a 100)    ; apresenta mensagem
190     tempo_ms(1000)                ; espera 1 segundo
200 FIM DO SE                               ; fim do se

```

Exemplo 27 - Seleção de Casos com Variáveis

Neste exemplo veremos como devemos proceder para fazermos uma seleção de casos com as variáveis.

```

10 ; Programa para demonstrar o uso do SELEÇÃO DE CASO
20 ; Autor: Cerne Tecnologia
30 ; Data: 29/01/2005
40
50 d%=5000                                ; atribui 10000 a d%
60
70 SELEÇÃO DE CASO d%                      ; seleção de casos com d%
80
90     CASO 1000                            ; caso seja 1000...
100
110         DISPLAY(1) (E 1000)             ; apresenta mensagem
120         tempo_ms(1000)                 ; aguarda tempo
130         TRAVAR                          ; fim do CASO
140
150     CASO 5000                            ; caso seja 5000...
160
170         DISPLAY(1) (E 5000)             ; apresenta mensagem
180         tempo_ms(1000)                 ; aguarda tempo
190         TRAVAR                          ; fim do CASO
200
210     CASO 10000                           ; caso seja 10000...
220
230         DISPLAY(1) (E 10000)           ; apresenta mensagem
240         tempo_ms(1000)                 ; aguarda tempo
250         TRAVAR                          ; fim do CASO
260
270     SENÃO select                          ; caso não seja nenhum...
280
290         DISPLAY(1) (Nao encontrado) ; apresenta mensagem
300         TRAVAR
310
320 FIM SELEÇÃO
330

```

Exemplo 28 - Operações Lógicas com Variáveis

Neste exemplo veremos como aplicar as operações lógicas nas variáveis inteiras do sistema.

```

10 ; Programa para demonstrar as operações lógicas
20 ; com as variáveis do sistema
30 ; Autor: Cerne Tecnologia
40 ; Data: 29/01/2005
50
60 a%=65535 ; atribui valor a variável
70 DISPLAY(1)(a%) ; apresenta o conteúdo da variável
80 tempo_ms(1000) ; espera 2 segundos
90 a%=a% E 255 ; lógica E com constante
100 DISPLAY(1)(a%) ; apresenta conteúdo da variável
110 tempo_ms(1000) ; espera 2 segundos
140
150 a%=1 ; atribui constante a variável
160 a%=a% OU 2 ; lógica OU com constante
170 DISPLAY(1)(a%) ; apresenta conteúdo da variável
180 tempo_ms(1000) ; aguarda tempo
190
200 a%=255 ; atribui constante a variável
210 a%=TROCAR a% ; inverte parte baixa pela alta
220 DISPLAY(1)(a%) ; apresenta conteúdo da variável
230 tempo_ms(1000) ; aguarda tempo
240
250 a%=1 ; atribui constante a variável
260 a%=NÃO a% ; inverte o estado dos bits
270 DISPLAY(1)(a%) ; apresenta conteúdo da variável
280 tempo_ms(1000) ; aguarda tempo
290
300 a%=100 ; atribui constante a variável
310 a%=a% xor 1 ; lógica xor com 1
320 DISPLAY(1)(a%) ; apresenta conteúdo da variável
330 tempo_ms(1000) ; aguarda tempo

```

Exemplo 29 - Atribuição nas Variáveis Caracter

Neste tópico vamos ver como é feita a atribuição de valores nas variáveis caracter.

```

10 ; Programa de atribuição de
20 ; Valores a variável caracter
30 ; Autor: Cerne Tecnologia
40 ; Data: 29/01/2005
50
60 a$='p' ; atribui caracter a a$
70 b$='c' ; atribui caracter a b$
80 c$='l' ; atribui caracter a c$
110 DISPLAY(1)(a$) ; apresenta a$
120 tempo_ms(1000) ; espera 1 segundo
130 DISPLAY(1)(b$) ; apresenta b$
140 tempo_ms(1000) ; espera 1 segundo
150 DISPLAY(1)(c$) ; apresenta c$
160 tempo_ms(1000) ; espera 1 segundo

```

Exemplo 30 - Transmissão de Variáveis Caracter pelo Canal Serial

Neste exemplo iremos abordar a transmissão das variáveis caracter pelo canal serial.

```

10 ; Program-a para demonstrar a transmissão
20 ; de variáveis caracter pelo canal serial.
30

```

```

40   a$='b'           ; carrega variável caracter
50   TRANSMITIR(a$)  ; transmite a mesma pelo canal serial
60
70   b$='A'           ; carrega variável caracter
80   TRANSMITIR(b$)  ; transmite a mesma pelo canal serial

```

Exemplo 31 - Testes condicionais com as Variáveis Caracter

Neste exemplo vamos abordar o teste condicional com a variável caracter.

```

10   ; Programa para demonstrar
20   ; testes condicionais com as variáveis caracter
30   ; Autor: Cerne Tecnologia
40   ; Data: 29/01/2005
50
60   e$='v'
70
80   if e$='m' ENTÃO
90       DISPLAY(1)(Variavel igual)
100      DISPLAY(2)(a 'm'          )
110      tempo_ms(2000)
120   FIM DO SE

```

Exemplo 32 - Seleção de casos com as Variáveis Caracter

Neste exemplo vamos abordar a seleção de casos com a variável caracter.

```

10   ; Programa para demonstrar o uso da seleção
20   ; de casos com a variável caracter
30   ; Autor: Cerne Tecnologia
40   ; Data: 29/01/2005
50
60   a$='z'           ; atribui caracter
70
80   SELEÇÃO DE CASO a$           ; seleção de casos de a$
90       CASO 'm'           ; caso 'm'
100          LIGAR(1)       ; liga relê 1
110          TRAVAR         ; fim do CASO
120       CASO 'o'           ; caso 'o'
130          LIGAR(2)       ; liga o relê 2
140          TRAVAR         ; fim do CASO
150       CASO 'z'           ; caso 'z'
160          LIGAR(3)       ; liga o relê 3
170          TRAVAR         ; fim do CASO
180   FIM SELEÇÃO           ; fim da seleção

```

Exemplo 33 - Utilizando as funções CHR\$ e ASC\$

Neste exemplo vamos utilizar as funções CHR\$ e ASC\$ para atribuir valores as variáveis que não se encontram no teclado.

```

10   ; Programa de demonstração da
20   ; utilização das funções CHR$ e ASC$
30   ; Autor: Cerne Tecnologia
40   ; Data: 29/01/2005
50
60   a%=asc$('A')      ; atribui o valor ASCII do 'A' em a%
70   DISPLAY(1)(a%)    ; apresenta no DISPLAY 80

```

```

90   a$=chr$(48)           ; atribui o '0' a a$
100  DISPLAY(2)(a$)       ; apresenta no DISPLAY

```

Exemplo 34 - Escrevendo na memória não volátil

O sistema possui 256 bytes de de memória de dados não voláteis, ou seja, após o sistema ter sido desligado, os dados não serão perdidos. Neste primeiro exemplo com estas memórias, iremos escrever alguns caracteres em posições específicas da memória.

```

10   ; Programa exemplo para escrita
20   ; de dados na memória não volátil
30   ; Autor: Cerne Tecnologia
40   ; Data: 29/01/2005
50
60   ENDERECO DA MEMÓRIA=100      ; aponta para o endereço 100
70   ESCREVER NA MEMÓRIA(a)      ; escreve o caracter neste
80                                   ; endereço

```

Exemplo 35 - Lendo a memória não volátil

Agora iremos ler na mesma posição em que foi feita a escrita e compararmos se os valores estão corretos.

```

10   ; Programa exemplo para
20   ; demonstrar a leitura da memória
100  ; Autor: Cerne Tecnologia
110  ; Data: 29/01/2005
30
40   ENDERECO DA MEMÓRIA=100      ; aponta para o endereço
50
60   SE LEITURA DA MEMÓRIA$='a' ENTÃO ; caracter lido é o 'a'?
70     DISPLAY(1)(Foi escrito o A!) ; sim, então apresenta
                                     ; mensagem
90     TRANSMITIR(MEMÓRIA$)        ; envia dado lido
80   FIM DO SE                    ; fim do se

```

Exemplo 36 - Comparador

Iremos utilizar nesta parte do manual, o comparador de tensão existente na placa. Para isso, devemos informar ao sistema qual será a tensão de referência utilizando o comando TENSÃO DE REFERÊNCIA. A tensão de referência deve ser um valor entre 0 e 5 VCC.

```

10   ; Programa exemplo para demonstrar
20   ; a utilização do comparador de tensão
30
40   TENSÃO DE REFERÊNCIA=2.5; define a t. de referência em 2.2V
50   etiquetal para again      ; define um etiqueta
70
80   again
90     SE COMPARADOR ENTÃO ; se a tensão de entrada é maior...
60     DISPLAY(1)(Tensao de Entrada) ; apresenta mensagem
100    DISPLAY(2)(E maior!)        ; apresenta mensagem
190    LIGAR(1)                    ; liga o relê 1
110    FIM DO SE                  ; fim do se
120
130    SE NÃO COMPARADOR ENTÃO ;se a tensão de entrada é menor...
140    DISPLAY(1)(Tensao de Entrada) ; apresenta mensagem
150    DISPLAY(2)(E menor!)        ; apresenta mensagem
200    DESLIGAR(1)                ; desliga o relê 1

```

```
160 FIM DO SE ; fim do se
170
180 vá para again ; salta para again
```

Exemplo 37 - Leitura de Tensão externa

AgOUa iremos ler uma tensão externa e verificar se ela ultrapassou um valOU. Caso tenha ultrapassado, iremos transmitir uma mensagem pelo canal serial, ligar um relê e display uma mensagem.

```
10 ; Programa de Leitura de Tensão externa
20 ; Autor: Cerne Tecnologia
30 ; Data: 29/01/2005
40
50 SE CAD$>3.00 ENTÃO ; se a tensão de entrada for
; maior que 3 Volts...
60
70 LIGAR(1) ; liga o relê 1
80 DISPLAY(1)(Maior que 3 V) ; mostra mensagem
90 TRANSMITIR(Passou 3 Volts) ; transmite dados
100
110 FIM DO SE ; fim do se
120
130 PULAR PARA O INICIO ; volta para o início
```

Exemplo 38 - Comparador de Janela

Este fantástico comando permite que verifiquemos se a tensão de entrada está entre dois limites. Caso esteja, o teste condicional será verdadeiro e os comando subseqüentes serão executados.

```
10 ; Programa exemplo para
20 ; demonstrar a utilização do
30 ; comparador de janela
40
50 SE CAD$>2.00 E CAD$<4.00 ENTÃO ; se a medição
; estiver nessa faixa...
60
70 INVERTER(SAÍDA(1)) ; inverte o estado do relê1
80 TRANSMITIR(Esta na faixa) ; transmite dados
90
100 FIM DO SE ; fim do se
110
120 PULAR PARA O INICIO ; volta para o início
```

Exemplo 39 - Controle por PWM

A PCL 1001 vem com um módulo PWM para controle externo. O PWM pode ser usado por exemplo para controlar a velocidade de um motor. O duty cycle é ajustável pelo software.

```
10 ; Programa para controle do PWM
20 ; Autor: Cerne Tecnologia
30 ; Data: 29/01/2005
40
50 SE SENSOR(1) ENTÃO ; se chave 1 pressionada...
60 pwm(0) ; duty-cycle é 0
70 FIM DO SE ; fim do se
80
```



```

90 SE SENSOR(2) ENTÃO ; se chave 2 pressionada...
100     pwm(25) ; duty-cycle é 25%
110 FIM DO SE ; fim do se
120
130 SE SENSOR(3) ENTÃO ; se chave 3 pressionada...
140     pwm(50) ; duty-cycle é 50%
150 FIM DO SE ; fim do se
160
170 SE SENSOR(4) ENTÃO ; se chave 4 pressionada...
180     pwm(75) ; duty-cycle é 75%
190 FIM DO SE ; fim do se
200
210 SE SENSOR(5) ENTÃO ; se chave 5 pressionada...
220     pwm(100) ; duty-cycle é 100%
230 FIM DO SE ; fim do se
240
250 PULAR PARA O INICIO ; volta para o início

```

Exemplo 40 - Saída de Tensão

A PCL ainda tem uma saída de tensão de 0 á 5V ajustável por software. Vejamos um exemplo deste módulo.

```

10 ; Programa exemplo para demonstrar
20 ; a saída de tensão externa
30 ; Autor: Cerne Tecnologia
40 ; Data: 29/01/2005
50
60 SE recepção de dados='A' ENTÃO ; se recebeu o 'A'...
70     SAÍDA_VOLTAGEM(0) ; saída de 0V
80 FIM DO SE ; fim do se
90
100 SE recepção de dados='B' ENTÃO ; se recebeu o 'B'...
110     SAÍDA_VOLTAGEM (1.21) ; saída de 1.21 V
120 FIM DO SE ; fim do se
130
140 SE recepção de dados='C' ENTÃO ; se recebeu o 'C'...
150     SAÍDA_VOLTAGEM (2.45) ; saída de 2.45 V
160 FIM DO SE ; fim do se
170
180 SE recepção de dados='D' ENTÃO ; se recebeu o 'D'...
190     SAÍDA_VOLTAGEM (4.10) ; saída de 4.1 V
210 FIM DO SE
220
230 PULAR PARA O INICIO

```

Exemplo 41 - Desligando a placa automaticamente

O Comando Shutdown permite que a placa seja desligada passado algum tempo fornecido em segundos. Vejamos uma aplicação.

```

10 ; Programa para desligar a placa
20 ; após um determinado horário
30
40 DISPLAY(1) (HORA$) ; apresenta a hora
50
60 SE HORA$>"22_00" ENTÃO ; se passou das 22:00...
70     DESLIGAR ON ; programa para desligar em 1 minuto

```

```

80     FIM DO SE                ; fim do se
90
100    PULAR PARA O INICIO      ; volta para início

```

Exemplo 42 - Apresentando o mês corrente

A PCL 1001 possui internamente um calendário do mês vigente e este pode ser apresentado no DISPLAY.

```

10     ; Programa para apresentar
20     ; o mês corrente
30     ; Autor: Cerne Tecnologia
40     ; Data: 29/01/2005
50
60     DISPLAY(1)(HOUA$)        ; apresenta a hora
70     DISPLAY(2)(DATA$)       ; apresenta a data
80     DISPLAY(2)(mês$)        ; apresenta o mês
90     PULAR PARA O INICIO      ; volta para o início

```

Exemplo 43 - Utilizando o terminal

Neste exemplo aprenderemos a utilizar o comando TERMINA. Este fantástico comando permite que os dados digitados no teclado pelo seu PC sejam enviados ao DISPLAY da PCL.

```

10     ; Programa para demonstrar a
20     ; utilização do comando TERMINAL
21     ; Autor: Cerne Tecnologia
80     ; Data: 29/01/2005
30
40     TERMINAL 20              ; fica no comando TERMINAL por 20 segundos
50     LIMPAR DISPLAYLAY ; limpa o DISPLAY
60     DISPLAY(1)(Fim do Processo) ; apresenta "Fim do Processo"

```

Exemplo 44 - Chamando Rotinas

Agora iremos chamar aprender a chamar rotinas com a AutoEasy. As rotinas diminuem o espaço da memória de programa, pois podem ser utilizadas em comum por vários processos no programa.

```

10     ; Programa exemplo para demonstrar
20     ; a utilização das chamadas de programa
40     ; Autor: Cerne Tecnologia
30     ; Data: 29/01/2005
50
60     etiqueta1 para limpa_DISPLAYlay ; define etiqueta1
70     etiqueta2 para mostra_tela     ; define etiqueta2
80     etiqueta3 para transmite_dados ; define etiqueta3
90
100    chamar mostra_tela              ; chama rotina
110    chamar transmite_dados          ; chama rotina
120    chamar limpa_DISPLAYlay         ; chama rotina
130    tempo_ms(1000)                 ; espera 1 segundo
140    chamar mostra_tela              ; chama rotina
150    chamar transmite_dados          ; chama rotina
160    chamar limpa_DISPLAYlay         ; chama rotina

```

```

170 FIM ; fim
180
190 mostra_tela
200     DISPLAY(1)(Teste do CTUDO)
210     RETORNAR ; retorno de sub-rotina

230 transmite_dados
240     TRANSMITIR(Teste do CTUDO)
250     RETORNAR ; retorno de sub-rotina

270 limpa_DISPLAYlay
280     tempo_ms(1000)
290     LIMPAR DISPLAY
320     RETORNAR ; retorno de sub-rotina

```

Exemplo 45 - Apresentando o conteúdo de uma variável em outras bases

A PCL pode apresentar e transmitir o conteúdo de uma variável em: binário, decimal, octal e hexadecimal. Vamos acompanhar neste exemplo como isto é possível

```

10 ; Programa para apresentar
20 ; o conteúdo da variável em outras bases
30 ; Autor: Cerne Tecnologia
40 ; Data: 29/01/2005
50
60 a%=100 ; carrega variável com 100
70 DISPLAY(1)(Valor em decimal) ; apresenta mensagem
80 DISPLAY(2)(a%) ; apresenta conteúdo em decimal
90 tempo_ms(2000) ; aguarda 2 segundos
100 LIMPAR DISPLAY ; limpa o DISPLAYlay
110 DISPLAY(1)(ValOU em binario) ; apresenta mensagem
120 DISPLAY(2)(bin$(a%)) ; apresenta conteúdo em binário
130 tempo_ms(2000) ; aguarda 2 segundos
140 LIMPAR DISPLAY ; limpa o DISPLAY
150 DISPLAY(1)(Valor em Hexadecimal) ; apresenta mensagem
160 DISPLAY(2)(hex$(a%)) ; apresenta conteúdo em hexa
170 tempo_ms(2000) ; aguarda 2 segundos
180 LIMPAR DISPLAY ; limpa o DISPLAYlay
190 DISPLAY(1)(ValOU em octal) ; apresenta mensagem
200 DISPLAY(2)(oct$(a%)) ; apresenta conteúdo em octal
210 tempo_ms(2000) ; aguarda 2 segundos

```

Exemplo 46 - Atribuindo um número atômico, massa e eletrosfera á uma variável inteira

Neste exemplo iremos atribuir as variáveis a%, b% e c% o número atômico, a massa e a eletrosfera do elemento químico Hidrogênio.

```

10 ; Programa de atribuição de número atômico,
20 ; massa e eletrosfera a uma variável
30 ; Autor: Cerne Tecnologia
40 ; Data: 29/01/2005
50
60 a%=z(H) ; atribui a a% o Z de H
70 b%=mass(H) ; atribui a massa de H
80 c%=elet(H) ; atribui a eletrosfera de H
90 LIMPAR DISPLAYLAY ; limpa o DISPLAY
100 DISPLAY(1)( Hidrogenio ) ; apresenta mensagem
110 tempo_ms(2000) ; espera 2 segundos
120
180 LIMPAR DISPLAYLAY ; limpa o DISPLAY

```

```

130 DISPLAY(1) (Numero Atomico ) ; mostra mensagem
140 DISPLAY(2) (a%) ; mostra conteúdo de a%
150 tempo_ms(2000) ; espera 2 segundos
160
170 LIMPAR DISPLAYLAY ; limpa o DISPLAY
190 DISPLAY(1) (Massa) ; mostra mensagem
200 DISPLAY(2) (b%) ; mostra conteúdo de b%
210 tempo_ms(2000) ; espera 2 segundos
220
230 LIMPAR DISPLAYLAY ; limpa o DISPLAY
240 DISPLAY(1) (Eletrosfera) ; mostra mensagem
250 DISPLAY(2) (c%) ; mostra conteúdo de c%
```

Exemplo 47 - Medido de RPM Externo

Agora iremos apresentar um exemplo bem interessante, um medidor de RPM. Vamos utilizar o comando RPM para esta função.

```

10 ; Programa de Medição de
20 ; RPM Externo
30 ; Autor: Cerne Tecnologia
40 ; Data: 29/01/2005
50
60 ALETAS PADRÃO=10 ; define número de aletas
70 etiquetal para again ; define etiqueta
80 ENTRADA6 COMO RPM ; define a entrada 6 como RPM
90
100 again
110 DISPLAY(1) (RPM$) ; apresenta o RPM no DISPLAY
120 vá para again
```

Exemplo 48 - Atribuindo Seno, Cosseno, Tangente, Cossecante, Secante e Cotangente as Variáveis

Agora iremos atribuir os valores supracitados as variáveis. Observem.

```

10 ; Programa de Atribuição de
20 ; Seno, Cosseno, Tangente, Cossecante,
30 ; Secante e Cotangente as variáveis
40 ; Autor: Cerne Tecnologia
50 ; data: 29/01/2005
60
70 a@=sin(45) ; atribui seno de 45
80 b@=cos(30) ; atribui coseno de 30
90 c@=tan(10) ; atribui tangente de 10
100 d@=sec(70) ; atribui secante de 70
110 a@=cosec(80) ; atribui cosecante de 80
120 a@=cotan(15) ; atribui cotangente de 15
```

Exemplo 49 - Atribuindo Logaritmo, Logaritmo Neperiano e Exponencial as Variáveis

Agora iremos atribuir os valores supracitados as variáveis. Observem.

```

10 ; Programa de atribuição de
20 ; Exponencial, Logaritmo Neperiano e na Base 10
30
40 a@=exp(2)
```

```
50 b@=ln(2)
60 c@=log(10)
```

Exemplo 50 - Trabalhando com números randômicos

A PCL 1001 vem com um gerador de números randômicos. Vejamos um exemplo.

```
10 ; Programa para ler números
20 ; randômicos
30 ; Autor: Cerne Tecnologia
40 ; Data: 29/01/2005
50
60 a%=rEom$ ; lê o valor randômicos
70 DISPLAY(1)(a%) ; apresenta no DISPLAY
80 tempo_ms(300) ; espera 300 ms
90 PULAR PARA O INICIO ; volta para o início
```

Exemplo 51 - Voltímetro

A PCL 1001 tem um módulo de medição de tensão externa. Vamos ver como utiliza-lo.

```
10 ; Programa exemplo para demonstrar a
20 ; utilização do módulo voltímetro
30 ; Autor: Cerne Tecnologia
40 ; Data: 29/01/2005
50
110 DISPLAY(1)(CAD$(voltagem)) ; apresenta valor medido
120 PULAR PARA O INICIO
```

Exemplo 52 - Medição de Velocidade Angular

A PCL 1001 também pode medir velocidades angulares externas. Para isso, a entrada 6 deve estar configurada para o modo frequencímetro e a entrada de pulsos ligada neste. Vejamos o exemplo.

```
10 ; Programa para medição de
20 ; velocidade angular
30 ; Autor: Cerne Tecnologia
40 ; Data: 29/01/2005
50
60 ENTRADA6 COMO FREQUENCIMETRO ; entrada 6 como frequencímetro
70 etiquetal para again ; define etiquetal
80 again
90 a%=vel_ang$ ; variável recebe vel_ang$
100 DISPLAY(1)(a@) ; apresenta o conteúdo da variável
110 vá para again ; salta para again
```