

Relatório de Aplicação de Microprocessadores

Termômetro

Daniel Vasconcelos Gomes n.o USP- 2959907

1- Objetivo

Inicialmente foram efetuadas medidas da saída do conversor A/D do Aes-10, variando-se a tensão de entrada. Desta forma elaboramos uma tabela aonde poderíamos associar temperatura com a leitura do conversor A/D.

Em seguida, nós elaboramos dois programas, um em *Basic* e outro em *Assembler* aonde através da leitura da temperatura no conversor A/D (com um resistor variável com a temperatura) poderíamos usando os programas ter o resultado da temperatura lida.

2- Programa em Basic.

Através de uma regressão logarítmica associamos a saída do conversor A/D à temperatura que desejávamos medir.

A primeira regressão resultou em:

$$T(x) = C1 + C2 * \log(x), \text{ onde } C1 = 224.8680 \text{ e } C2 = -44.8819$$

Posteriormente com o programa implementado corrigimos as constantes acima para:

$$C1 = 254.000 \text{ e } C2 = -50.000$$

A listagem do programa em Basic é a seguinte:

```
1 call C
10 print "A TEMPERATURA LIDA E-
20 call l
30 a=dby(21H
40 b= (log(a)/ log(2.7182
50 c= 224.868 - 44.8819
60 print& using (xx,x),
70 for k= 1 to 10
80 next k
90 goto 1
```

3- Programa em Assembler.

Inicialmente elaboramos uma tabela onde constavam lado a lado, código, em hexadecimal e temperatura. Como não tínhamos todos os pontos, foi preciso fazer algumas adaptações para podermos ler qualquer temperatura no intervalo desejado.

Os códigos de 39H até 9DH representam as temperaturas positivas . Seus valores respectivos de temperatura são: 29H e 0H. Os códigos de 9EH até FFH representam as temperaturas de -1H até -1EH. Como não podemos manipular números negativos em hexadecimal, foi necessária uma rotina que imprimia um sinal “-“ se a temperatura fosse negativa. A tabela de temperatura foi colocada numa posição de memória externa. Quando o programa lê o dado no conversor A/D, ele “pula” este número à partir da posição inicial de memória. Em seguida, detecta se a temperatura é negativa ou não, depois imprime na tela a temperatura lida, depois do número ter sido convertido em código ASCII. O algoritmo do programa em Basic e o Assembler constam no final do relatório.

A listagem do programa em Assembler esta abaixo.

ORG 5039

LJMP INICIO:

```
DB 29H, 28H, 27H, 26H, 25H, 24H, 23H, 23H, 23H, 22H, 22H, 22H, 21H, 21H, 21H, 20H, 1FH, 1FH, 1FH, 1EH,
,1EH, 1DH,1DH, 1CH,1CH,1CH, 1BH,1BH, 1AH,1AH, 19H, 19H,
18H, 18H, 18H, 17H,17H,17H, 16H,16H,15H,15H, 14H,14H,
13H, 13H, 13H, 12H,12H, 11H,11H,11H,10H,10H, 0FH, 0FH,
0FH, 0EH, 0EH ,0DH, 0DH, 0DH, 0CH, 0CH, 0CH, 0BH, 0BH,
0BH, 0AH, 0AH, 0AH, 9H,9H,9H, 8H, 8H, 8H, 7H, 7H, 7H,
6H,6H, 6H, 5H,5H,5H, 4H, 4H, 4H, 3H, 3H, 3H, 2H, 2H, 2H,
1H, 1H, 1H, 0H, 0H, 0H, 1H, 1H, 1H, 1H, 1H, 1H, 1H,
2H, 2H, 2H, 2H, 3H, 3H, 3H, 4H, 4H, 4H, 5H, 5H, 5H,
5H, 6H , 6H, 6H, 6H, 6H, 6H, 7H, 7H, 7H, 8H, 8H, 8H,
9H, 9H, 0AH, 0AH, 0BH, 0BH, 0BH, 0CH, 0CH, 0CH, 0DH,
0DH, 0DH, 0EH, 0EH, 0FH, 0FH, 0FH, 10H, 10H, 10H, 11H,
11H, 11H, 12H, 12H, 12H, 12H, 12H, 13H, 13H, 13H, 14H,
14H, 14H, 14H, 14H, 15H, 15H, 15H, 16H,16H, 16H, 17H,
17H, 17H, 18H, 18H, 18H, 18H, 19H, 19H, 19H, 1AH, 1AH,
1AH, 1AH, 1BH, 1BH, 1CH, 1CH, 1CH, 1DH, 1EH
INICIO: CALL 4100 ; LIMPA O DISPLAY DO KIT
CALL 4114 ;LE CONVERSOR A/D
MOV R0,21H ;R0<---21H
MOV DPTR,5000 ;PONTEIRO DE DADOS APONTA PARA
;5000h EXTERNO
MOV A,@R0 ;[R0]= LEITURA P/ ACC
DEC A ;DECREMENTA O ACC INCREMENTANDO
```

```

JZ    FIM            ;DPTR AT ENCONTRAR ZERO
INC   DPTR          ;QDO. ACC=0, TEMOS A TEMP.
FIM:  MOVX    R0,DPTR ;LEITURA DA TEMP NO REG, R0
      MOVX    A, DPTR ;LEITURA DA TEMP NO ACC
      MOV    R1,# 0H  ; CONDICAO DE TEMP <0
      SUBB   A,#9E    ;
      JC    COND     ;ACC<9EH
COND:  ADD    R1,#1H  ;COND= 0 T<0 COND= 1 T>0
      MOV    A, R0   ; RESTAURA O VALOR DA TEMP
      PUSH  0H      ;ARMAZENA A, R0, R1
      MOV   A,R1    ;MOVE 0--T>0 OU 1 T<0
      SUBB  A,#1H   ;0 NAO IMPRIME - 1 IMPRIME -
      JZ    CONDI   ;
      MOV   21H,'-' ;
      CALL  4104H   ;IMPRIME - SE T<0
CONDI: POP    0     ; RESTAURA A,R0,R1
      MOV   A, R0   ;ACC CONTEM LEITURA
      SWAP A       ;ROTACIONA A 4X
      ANL  A,#0F   ;ZERA PARTE + SIGNIFICATIVA
      ADD  A,30H   ;CONVERTE PARA ASCI
      MOV  21H,A   ;
      PUSH 0H     ;
      CALL 4104H   ;IMPRIME 0 1.o DIGITO
      POP  0H     ;
      MOV  A,R0   ;PROCEDIMENTO PARA
      ANL  A,#0FH  ;IMPRIMIR O 2.O DIGITO
      MOV  21H,A  ;
      CALL 4104H  ;IMPRIME NA TELA O 2.o DIGITO
      MOV  R0,#0H ;PROCEDIMENTO DE DELAY
      MOV  R1,#0H ;
      MOV  R2,#11H ;
      DJNZ R0,$   ;
      DJNZ R1,$-2 ;
      DJNZ R2,$-4 ;
      LJMP INICIO
      END

```