MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

MONTAGEM E TESTES DE UM SISTEMA MICROCONTROLADO PARA AQUISIÇÃO DE DADOS DO RADIOTELESCÓPIO GEM

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA

(PIBIC/CNPq/INPE)

Eduardo Fonseca Rodrigues (ITA, Bolsista PIBIC/CNPq)
E-mail: fonseca@h8.ita.br

Dr. Carlos Alexandre Wuensche (DAS/INPE, Orientador)
E-mail: alex@das.inpe.br

COLABORADORES

Mauro Prado (DAS/INPE)
Alan Cassiano (LAC/CTE/INPE)

Maio de 2004

Orientador: Carlos Alexandre Wuensche

Título:
Montagem e testes de um sistema microcontrolado para aquisição de dados do radiotelescópio GEM

São José dos Campos - SP

1 – Indice

2 – Resumo

Esse trabalho tem o objetivo de realizar o projeto, a montagem e testes de um sistema microcontrolado para aquisição de dados do radiotelescópio GEM.

O projeto realizado utilizou um microcontrolador da família 8051 para realizar as tarefas necessárias para obter os dados de maneira conveninente do radiotelescópio GEM.

A sequência de procedimentos a ser realizados foi elaborada para que houvesse total compatibilidade com o sistema de aquisição antigo, com mesmos intervalos de tempo gastos com as tarefas.

A sequência de procedimentos realizados para enviar os dados recebidos é basicamente a seguinte: Primeiramente um canal é escolhido pelo multiplexador analógico para realizar a conversão analógico para digital(o conversor apresenta saída serial), uma sequência de bits é formada na RAM, juntamente com bits para verificação de erros para ser finalmente enviada quando o frame de transmissão for finalizado.

Os códigos dos programas necessários para o funcionamento do sistema, gravados no microcontrolador estão indicados no apêndice.

3 – Introdução

3.1 - Objetivo

O objetivo dessa bolsa é acompanhar o projeto de um sistema de aquisição de dados microcontrolado a ser utilizado no projeto GEM (Mapeamento da Emissão Galática, do inglês Galaxy Emission Mapping). O microcontrolador utilizado é o 8051.

Embora já exista um sistema de aquisição de dados para tal fim, resolveu-se implementar um sistema mais moderno, de pequenas dimensões e com uma documentação mais completa que o anterior.

3.2 – Projeto GEM

O projeto GEM tem como objetivo determinar a distribuição espacial e a intensidade absoluta na faixa de rádio e microondas da radiação emitida pela Via Láctea. A antena do GEM tem 9.5m de diâmetro e já passou por: Antártica (1991/92), Bishop, USA (1993/94), Villa de Leyva, Colômbia (1995) e Tenerife, Espanha – (1995/97). Com a antena localizada atualmente em Cachoeira Paulista (SP), o projeto GEM tem o intuito de mapear o céu em 5 freqüências (408 MHz, 1465 MHz, 2300 MHz, 5 GHz e 10 GHz) para descobrir o quanto a radiação emitida por nossa galáxia influi nas medidas da radiação cósmica de fundo.

4 – Procedimento

Inicialmente estudou-se um pouco sobre cosmologia para entender o que era a radiação cósmica de fundo em microondas e qual era objeivo do projeto GEM no contexto de cosmologia. Concluída essa fase, iniciou-se o aprendizado de sistemas microcontrolados, com ênfase no microcontrolador 8051.

Também estudou-se um pouco sobre filtros digitais e analógicos. A utilização de filtros nos dados recebidos por esse sistema é muito importante, pois os dados são contaminados por diversos fatores que modificam sensivelmente o sinal de interesse.

Filtros analógicos têm a vantagem de serem relativamente fáceis de se projetar. No entanto, a existência da faixa de passagem, isto é, valores intermediários entre os que devem ser filtrados e os que devem ser aceitos, pode ser inconveniente, conforme a aplicação. Filtros digitais têm a vantagem de possuírem faixa de passagem menor, diminuindo o inconveniente citado anteriormente.

Notou-se ainda que a utilização de filtros passa-baixa, passa-faixa, passa-alta ou rejeita-faixa pode não ser a mais conveniente, dependendo da aplicação. Muitas vezes dados importantes são retirados ao realizar uma filtragem de uma grande faixa de valores. Dessa forma, também analisou-se a possibilidade de trabalhar com filtros digitais mais elaborados, que evitassem perdas de dados importantes.

Após entender como o sistema antigo de aquisição de dados funcionava, o projeto do sistema foi iniciado por uma equipe composta de engenheiros, (engenheiros e tecnólogos são denominações equivalentes nesse contexto), técnicos e estagiários do INPE, adaptando os subsistemas do experimento antigo ao novo projeto e pensando na viabilidade de inclusão de novas partes com o objetivo de melhorar o desempenho do sistema.

Partindo do projeto original, foram feitas várias correções até obter-se um esquema razoável. O tempo gasto em cada tarefa no sistema antigo foi analisado e uma temporização semelhante foi implementada no sistema novo para manter compatibilidade. Além disso, um diagrama de estados foi esboçado para indicar que tarefas poderiam ser realizadas, em caso de imprevistos, buscando, novamente, a compatibilidade com o sistema anterior.

Com o esquema elétrico praticamente finalizado, iniciou-se a confecção dos programas para que o sistema funcione convenientemente. Levando em consideração as características de cada componente utilizado, como as memórias, conversor analógico-digital e multiplexador analógico, as constantes de tempo, a temporização foi realizada visando uma operação sequencial. Seguindo o diagrama de estados elaborado, os códigos foram escritos para que o sistema funcionasse perfeitamente.

5 – Sistema de Aquisição de dados

5.1 – Requisitos do sistema

O sistema a ser projetado deve ter 16 canais de entrada analógicos provenientes da antena responsável pela recepção dos sinais, receber os dados de cada canal de forma ordenada e trata-los, de maneira a enviar os dados para um computador através de comunicação serial, com bits de controle como o start bit, stop bit e bit de paridade paraminimizar a possibilidade de perda dos dados coletados.

Cada canal deve ser selecionado individualmente de maneira ordenada até ser realizada a leitura dos 16 canais. Isso pode ser realizado por um multiplexador. São necessários 4 bits de endereço para selecionar cada canal. Antes de entrarem no multiplexador, as entradas devem ser amplificadas.

Escolhido o canal, um sinal analógico deve aparecer na saída do multiplexador e ser novamente amplificadoe depois convertido para uma forma digital. A saída do conversor deve ser apresentada na forma serial, bit a bit.

Devem ser formados 2 bytes com os bits apresentados na saída do conversor para serem escritos na memória e depois de todos os 16 canais serem lidos, os dados contidos na memória devem ser enviados na forma serial para o computador.

5.2 – Solução proposta

A solução encontrada para realizar todas essas tarefas de forma ordenada foi utilizar um microcontrolador (8051) para gerenciar todas as tarefas a serem realizadas. Em conjunto com uma memória de programa, onde toda a programação do sistema é colocada de forma a seguir o diagrama de estados, e uma memória de dados, utilizada para guardar os dados obtidos após a leitura de todos os canais, é possível que o sistema realize as ações necessárias e armazene dados conforme necessário .

Um multiplexador analógico (MV1606) e um conversor analógico digital (ADCS5102) para realizar a seleção dos canais e a conversão dos sinais. Também foi utilizado um watchdog (DS1232LP). Ele verifica se o sistema ultrapassou um tempo limite para realizar uma operação e caso isso aconteça, o sistema é reiniciado.

Para realizar a seleção de memória e de operações de leitura e escrita, é utilizado um demultiplexador (74LS138), latches com buffers tri-state (74HC573 e 74HC244). Além disso, são utilizadas portas lógicas OR e NAND, resistores, capacitores, amplificadores

operacionais, cristais, reguladores de tensão e fontes de tensão para geração das constantes de tempo necessárias, limitação de corrente, amplificação dos sinais e fornecimento da energia necessária para o funcionamento do sistema.

6 – Projeto do sistema

6.1 – Microcontrolador

Utiliza-se um microcontrolador 8031. Na figura 1, uma parte do diagrama do circuito é mostrada. Os pinos XTAL1 e XTAL2 são conectados a capacitores e um cristal de forma a termos um oscilador de frequência 12 MHz.

Os terminais da porta 0 estão sendo usados para enviar o byte menos significativo de endereçamento da memória e um byte de dados, a seguir. E realizada uma demultiplexação para separar esses sinais através de um latch. O pino ALE (Adress Latch Enable), habilita o latch no instante em que o sinal de endereço está sendo enviado e guarda esse valor na sua saída. Depois desse momento, o sinal de dados é enviado. A demultiplexação está representada na figura 2.

Os terminais da porta 2 estão sendo usados para enviar o byte mais significativo de endereçamento da memória. Não é necessário realizar a demultiplexação.

Os terminais da porta 3, P3.0(RXD) e P3.1(TXD), são usados para trasmissão serial de dados. Os terminais P3.2(INT0#) e P3.3(INT1#), são usados para pedidos de interrupção externos ou como bit de controle para os temporizadores. P3.4(T0) e P3.5(T1) são entradas externas para os temporizadores.. O pino P3.4(WR#) é um sinalizador de escrita e P3.5(RD#) é um sinalizador de leitura na memória de dados externa.

O pino PSEN# é utilizado para habilitar a memória externa, buscando instruções ou operandos na memória externa. EA# fica em nível lógico baixo para a CPU trabalhar apenas com a memória de programa externa. Caso contrário, trabalha apenas com memória de programa interna.

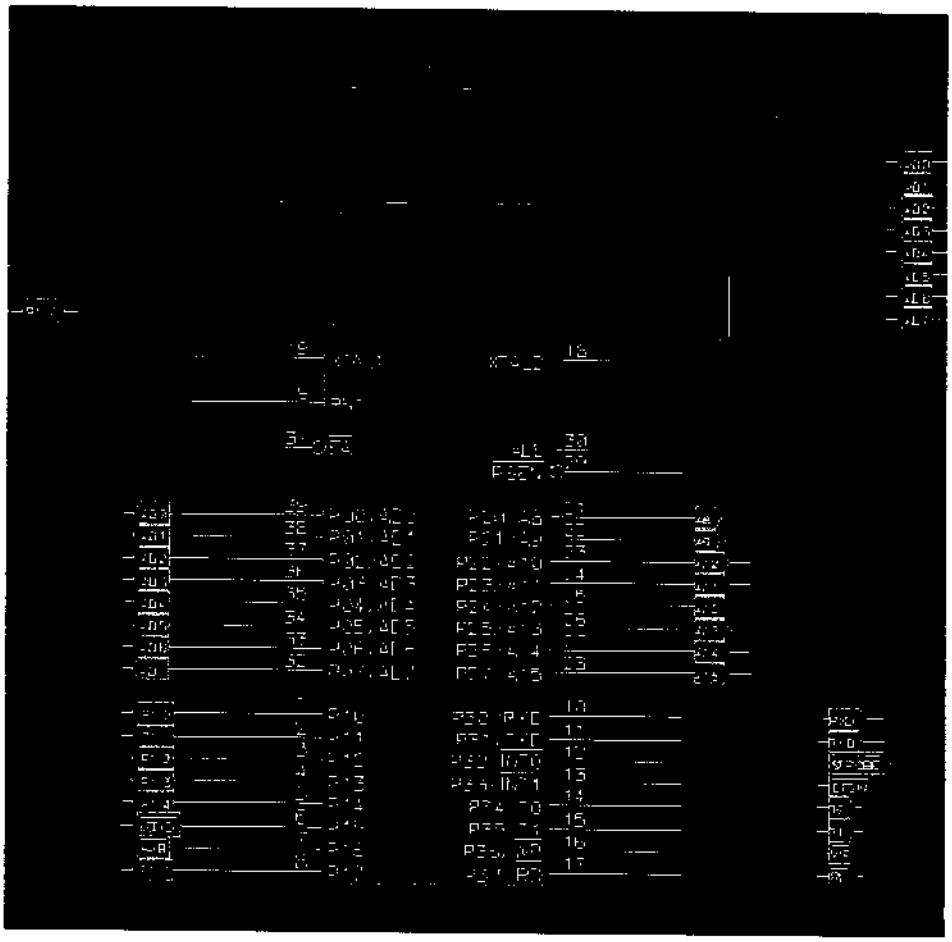


Figura 1 – Esquema do microcontrolador

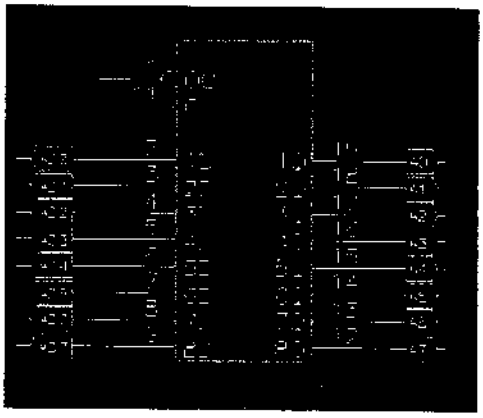


Figura 2 – Demultiplexação de endereço e dados(ALE conectado a C)

6.2 – Memória externa

E utilizada uma RAM externa (6264) e uma EPROM (2764), ambas de 64kb, para utilização da memória de programa externa e para armazenar uma maior quantidade de dados. Na figura 3, um esquema para acessar a RAM externa é mostrado. Para acessar esses elementos, PSEN# deve estar em nível alto.

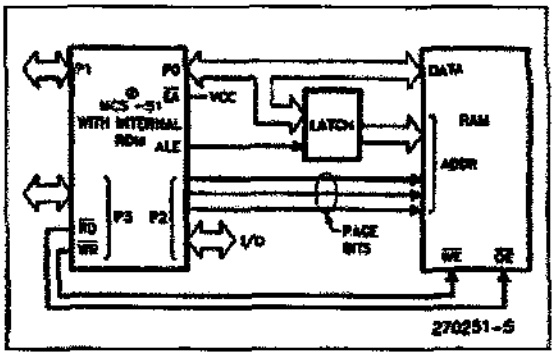
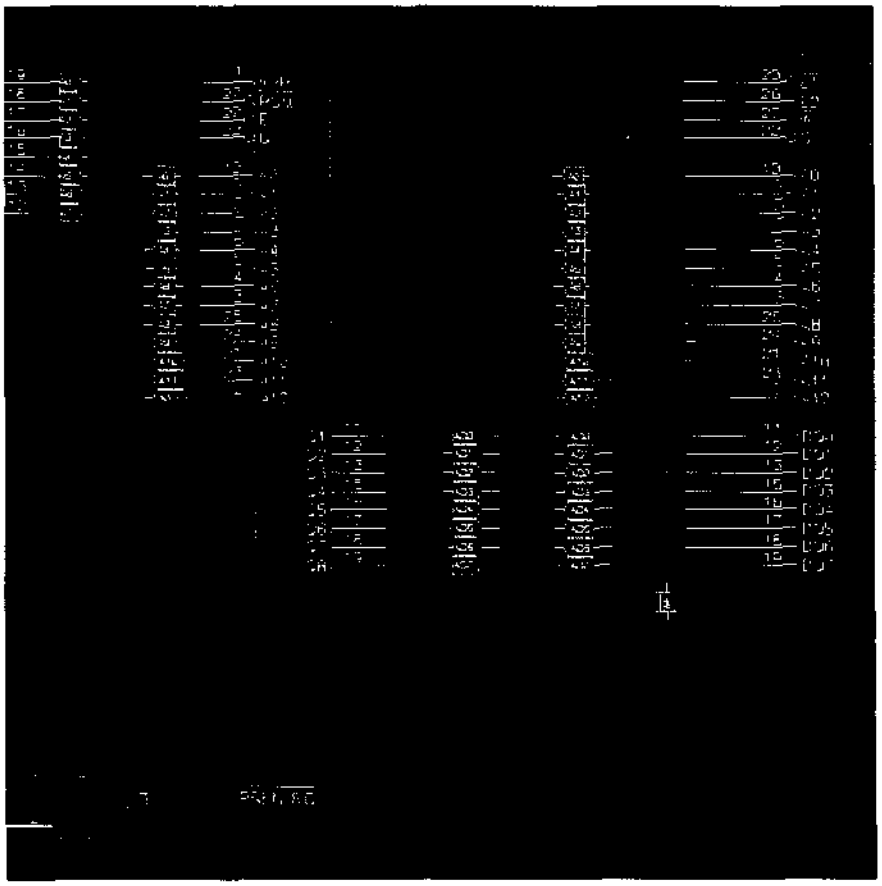


Figura 3 – Esquema para acessar a RAM externa

Na figura 4, mostra-se como estão conectados esses elementos.



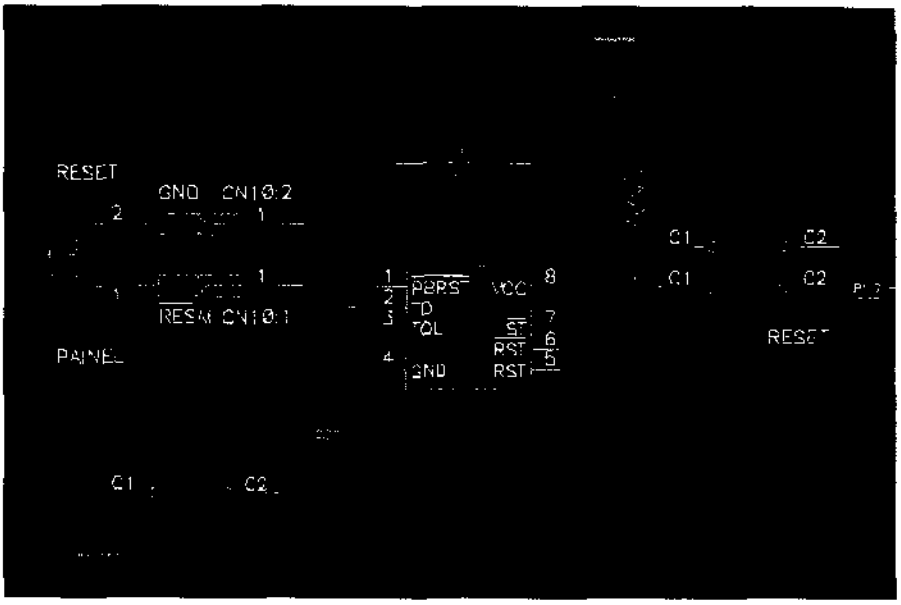


Figura 6 – Esquema do Watchdog

6.5 – Multiplexador analógico

O sistema tem 16 canais de entrada, que deverão ser multiplexados, de forma que cada canal apareça sequelcialmente na saída do multiplexador.

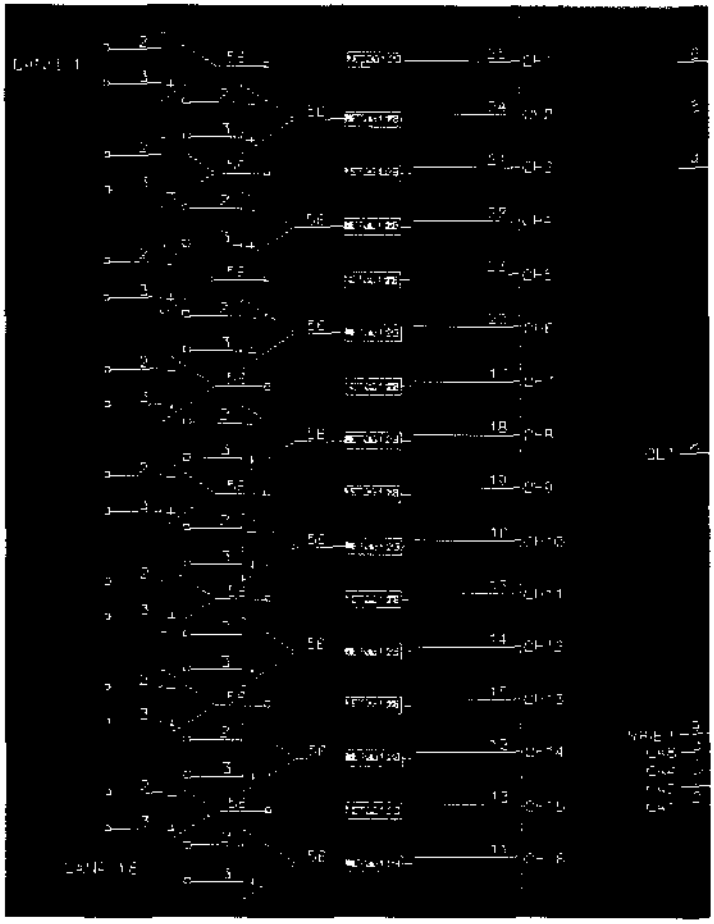


Figura 7 – Multiplexador analógico de 16 canais

6.6 – Conversor analógico-digital

Na saída do multiplexador, temos sinais analógicos. No entanto, os sinais devem se tornar digitais para que o sistema microcontrolado possa trabalhar com esses dados e depois transmitir os resultados serialmente.

O conversor analógico digital utilizado é o CS5102. Ele tem um tempo de conversão de 40μs, possui saída serial e tem 16-bits de entrada e 2 canais. Para calibrar os 16 bits com bastante precisão, esse conversor se calibra sozinho, no reset.

Quando uma transição de nível alto para baixo é realizada no pino HOLD#, o ciclo de conversão é iniciado. Nesse caso, o HOLD está conectado ao pino 1.1 do microcontrolador. Quando a conversão é finalizada, o conversor volta para o modo de rastreamento.

O pino CH1/2 escolhe entre TRK1 ou TRK2. A saída é dada em SDATA, apresentando inicialmente o bit mais significativo. No esquema apresentado, temos SCKMOD = 1 e OUTMOD = 0. Portanto, o conversor está operando no modo RBT(Registered Burst Transmission Mode). Os dados estarão disponíveis no final da conversão e na borda de descida de HOLD, a saída será apagada.

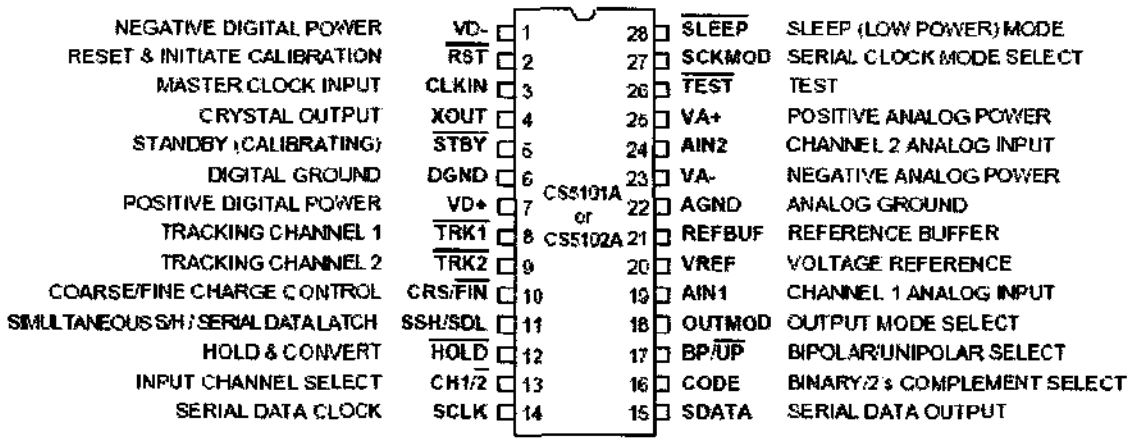


Figura 8 – Conversor Analógico/ Digital

Na figura abaixo, o diagrama de tempo do modo RBT é mostrado.

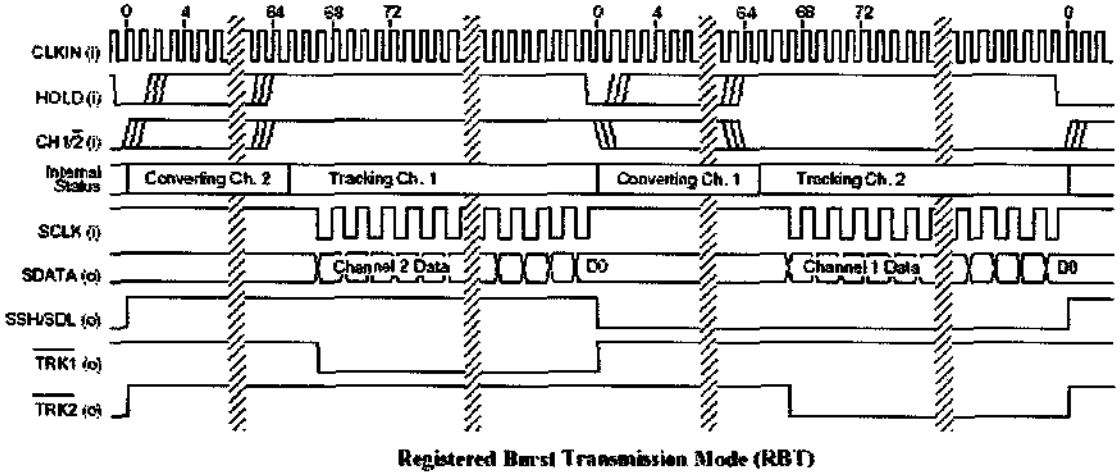


Figura 9 – Diagrama de tempo para modo RBT

7 – Operação do sistema

Inicialmente, as posições de memória interna e externa, interrupções, contadores e temporizadores são inicializadas, através do programa PADRAO3 (Apêndice F). A interface serial é configurada para operar com 9600 bps, no modo 1(um start bit, 8 bits de dados e um stop bit). Os temporizadores operam no modo 1(usa o par de registradores TH0 e TL0 ou TH1 e TL1 para efetuar a contagem, ou seja, tem-se um contador de 16 bits).

A porta 1 é inicializada da seguinte forma:

P1.0 = 0 - SDATA - SAIDA SERIAL DO CONVERSOR
P1.1 = 1 - CONVERT - INICIO DE CONVERSAO
P1.2 = 0 - SCLK - CLOCK EXTERNO PARA SERIALIZAÇÃO DOS DADOS
P1.3 = 0 - TRK1 - INDICA FIM DE CONVERSAO

P1.4 = 0 - STBY - INICIALIZACAO DO CONVERSOR
P1.5 = 0
P1.6 = 0
P1.7 = 1 - RESET WATCH DOG

Depois disso, o temporizador é programado e tem sua contagem iniciada, de maneira a realizar operações no mesmo tempo realizado pelo sistema antigo, embora seja possível realizar operações mais rapidamente.

A seguir, a conversão é realizada. O resultado da conversão é recebido bit a bit e os bytes mais e menos significativos são montados pelo microcontrolador, na memória RAM, e uma moldura é gerada, com parâmetros indicado começo e final do frame para realizar a transmissão serial dos dados. Os programas necessários para a operação desejada do sistema estão indicados nos apêndices.

8 – Conclusão

Essa bolsa permitiu o contato com a área de cosmologia e em especial, a radiação cósmica de fundo, que supõe-se estar relacionada intimamente com as origens do universo. O conhecimento adquirido com sistemas digitais é extenso, pois envolve toda a concepção de um sistema, seu diagrama de estados e sua programação.

O sistema foi projetado, os programas necessários para o funcionamento do sistema foram escritos, mas o sistema se encontrava em fase de revisão e encaminhamento para produção de placa de circuito impresso para montagem final do sistema e testes de bancada.na linguagem

Depois de montado o circuito na placa de circuito impresso, o sistema será configurado e testado e finalmente, com algumas adaptações, deverá substituir o sistema antigo.

9 – Bibliografia

1 – Aplicações práticas do microcontrolador 8051 – Vidal Pereira Silva Júnior
2 – Introdução à cosmologia – Laerte Sodré Junior
3 – The Scientist and Engineer´s guide to Digital Signal Processing – Steven W.Smith
4 – Datasheet do Conversor A/D CS5102A e do watchdog utilizado

10 – Apêndice

10.1 - Apêndice A – rotina_timer01

```
;*************************************************************************
;****************** INTERRUPÇÃO DE TIME0 - LEADC ***********************
;*************************************************************************

        ORG   2100H                 ;ENDEREÇO DE INICIO DE TIMER0
```

```
;************************************************************
;************************************************************
;*********PROGRAMAR O TIMER E INICIAR SUA CONTAGEM************
;************************************************************

        PUSH A
        PUSH PSW
        PUSH DPTR



        MOV  B,#09H


L2:     MOV  TL0,#F1H           ;TEMPORIZACAO DA REPETICAO DO FRAME
        MOV  TH0,#0CH           ;TEMPO 560 mS F=12MHZ
                                ;T=83,3nS

        SETB TR0                ;DISPARA TIMER 0.

        JNB  TF0,$              ;AGUARDA ESTOURO DO CONTADOR
TCON.5=1

        CLR  TF0
        CLR  TR0

        DJNZ B,L2

        SETB ET0                ;HABILITA A INTERRUPCAO DO TIMER 0
QUANDO EA=1



;************************************************************************
;************************************************************************

        MOV  DPTR,#4000H            ;APONTA PARA ENDEREÇO DO CANAL 1
        MOV  22H,#10H           ;POSICAO 22H E USADA COMO PONTEIRO
PARA OS 16 CANAIS.
                                ;VALOR 10H SIGNIFICA CANAL 1 HABILITADO
(INHIBIT = 1)
```

```
        MOV  A,22H
        MOVX      @DPTR,A                ;ESCREVENDO VALOR 10H NO
BARRAMENTO.


MENOR_16CH:INC 28H                       ;INICIALIZADO COM #FFH

        CLR  P1.1                ;INICIO DE CONVERSÃO PINO HOLD DE 1
PARA 0.
                                 ;NOTA: MENOR TEMPO 1CLOCK + 20nS E
MAXIMO TEMPO DE 2CLOCKS

        SETB P1.1                ;RESTAURA PINO HOLD PARA PROXIMA
CONVERSÃO


TESTE:  JNB  P1.3,TESTE          ;GARANTIR FINAL DE CONVERSÃO
EVITANDO ERRO DE LEITURA
                                 ;DO PINO TRK1 NO CASO DE LEITURA RAPIDA.

TESTE2:  JB  P1.3,TESTE2         ;NOTA 28 DO MANUAL CS5102A PAGINA
7.

;*********************************************************************
;*********************************************************************


        MOV  DPTR,#4000H         ;SELECAO DO PROXIMO CANAL DO
MUX.
        INC  22H                 ;PARA ESTABILIZAÇÃO.

        MOV  A,22H
        MOVX      @DPTR,A                ;SELECIONA O SEGUNDO
CANANL PARA ESTABILIZACAO

;*********************************************************************
;*********************************************************************
```

```
        ACALL   CANALMOL        ;SUB ROTINA QUE APONTA E GUARDA
OS VALORES DOS CANAIS
                                ;NA MOLDURA PARA TRANSMISSAO.

        INc     2DH             ;CONTA CANAIS ATE #0FH


        MOV  A,28H              ;CONTA CANAIS ATE #0FH

        CJNE  A,#0FH,MENOR_16CH



;*************************************************************************
;*************************************************************************
;******************CONTAGEM DO FRAME LSB E
MSB************************
;*************************************************************************
;*************************************************************************


        MOV  A,29H
        CJNE  A,#FFH,LS1
        INC   29H
        MOV  DPTR,#3202H
        MOV  A,29H
        MOVX    @DPTR,A
        SJMP  LS2


LS1:    INC   29H
        MOV  DPTR,#3202H
        MOV  A,29H
        MOVX    @DPTR,A
        SJMP  TRANST

;*************************************************************************
;*************************************************************************


LS2:    MOV  A,2AH
        CJNE  A,#FFH,MS1
        INC   2AH
        MOV  DPTR,#3203H
        MOV  A,2AH
```

```
        MOVX    @DPTR,A
        SJMP MS2


MS1:  INC   2AH
        MOV  DPTR,#3203H
        MOV  A,2AH
        MOVX    @DPTR,A
        SJMP TRANST

;*********************************************************************
;*********************************************************************


MS2:  MOV  A,2BH
        CJNE  A,#FFH,MS3
        MOV  DPTR,#3204H
        MOV  A,2BH
        MOVX    @DPTR,A
        SJMP MS4

MS3:  INC   2BH
        MOV  DPTR,#3204H
        MOV  A,2BH
        MOVX    @DPTR,A
        SJMP TRANST


;*********************************************************************
;*********************************************************************


MS4:  MOV  A,2CH
        CJNE  A,#FFH,MS5
        MOV  DPTR,#3205H
        MOV  A,2CH
        MOVX    @DPTR,A
        SJMP MS6

MS5:  INC   2CH
        MOV  DPTR,#3205H
```

```
        MOV  A,2CH
        MOVX        @DPTR,A
        SJMP  TRANST



;*********************************************************************
;*********************************************************************

MS6:  SJMP  REINICIA


REINICIA:MOV      29H,#00H          ;CONTADOR DE LSB
         MOV 2AH,#00H             ;CONTADOR DE LSB
         MOV 2BH,#00H             ;CONTADOR DE MSB
         MOV 2CH,#00H             ;CONTADOR DE MSB


        SJMP  RETORNA

;*********************************************************************
;*********************************************************************

TRANST:     ACALL       TX


;*********************************************************************
;*********************************************************************
;*****************************WATCHDOG**********************************
*
;****************SUPERVISIONA                    CICLO                    DE
AQUISIÇÃO**********************
;*********************************************************************


        CLR   P1.7          ;STROBE APLICADO NO PINO ST (STROBE INPUT)
        SETB  P1.7          ;DO SUPERVISOR MIN 20nS DE LARGURA E MAX
TDmin.
```

;NOTA: FINAL DE TRANSMISSÃO DO FRAME COM 42 BYTES.
;SISTEMA PROGRAMADO PARA GERAR UM INTERRUPÇÃO DE TIMER 0 = 0,56SEGUNDOS.
;RETOMANDO TODO CICLO DE AQUISIÇÃO.
;O TEMPO ENTRE A TRANSMISSÃO DO FRAME E O TIMER 0 SERA DISPONIVEL PARA O PESQUISADOR
;PROPOR NOVOS MODOS DE OPERAÇÃO.

```
;*********************************************************************
;*********************************************************************
;*********************************************************************
;*********************************************************************



RETORNA:POP     DPTR
        POP   PSW
        POP   A


        RETI
```

10.2 - Apêndice B – rotina_ le_canal1

```
;*********************************************************************
;*********************************************************************
;*******LE O PORT P1.0 RESPONSAVEL PELA LEITURA DOS BITS DO ADC********
;*********************************************************************
;*********************************************************************

LE_CANAL:PUSH     A
         PUSH PSW
         PUSH DPTR


PROXBIT:JNB  P1.0,BIT_IN0        ;TESTA SE O BIT LIDO DO ADC E 0 OU 1
                                 ;SE P1.0 = 0 VAI PARA BIT_IN0
                                 ;SE P1.0 = 1 SEGUE O PROGRAMA

        SETB  23H.0

        SJMP  BIT_IN1
```

```
BIT_IN0:CLR   23H.0

BIT_IN1:MOV   26H,24H                 ;26H-ENDERECO QUE ARMAZENA MSB
        MOV   25H,23H                 ;25H-ENDERECO QUE ARMAZENA LSB

        ACALL SHIFT_BIT               ;SUB ROTINA QUE ROTACIONA O BIT LIDO EM P1.0
                                      ;ATRAVES DO CARRY.

        INC   27H                     ;INCREMENTA CONTADOR DE BITS ATE 0FH.
        MOV   A,27H
        CJNE  A,#0FH,MENOR_16BIT;COMPARA E DESVIA SE O ACUMULADOR FOR
DIFERENTE DO DADO.

        MOV   27H,#00H                ;PREPARA CONTADOR PARA PROXIMO CANAL.


        POP   DPTR
        POP   PSW
        POP   A

        RET                           ;RETORNA DO FINAL DA MONTAGEM DE UM CANAL.

;*********************************************************************
;*********************************************************************

MENOR_16BIT:SETB  P1.3                ;GEROU SCLK ATE 5MHZ
            CLR   P1.3
;*********************************************************************
;*********************************************************************


        JMP   PROXBIT                 ;LE O PROXIMO BIT.
```

10.3 - Apêndice C – rotina _tx1

```
;*********************************************************************
;*********************************************************************
;**********************TRANSMISSÃO DO FRAME*****************************
;*********************************************************************
;*********************************************************************
```

```
TX:     PUSH  A
        PUSH  PSW
        PUSH  DPTR
        PUSH  B

?????? MOV    B,#2CH          ;CONTADOR DA MOLDURA MAIS 2BYTES - 44 BYTES.


        MOV   DPTR,#3200H  ;ENDEREÇO INICIAL DA MOLDURA - EBH
DESVIO:       MOVX  A,@DPTR



        MOV   SBUF,A                ;TRANSMITE BYTE

        INC   DPTR

        DJNZ  B,DESVIO      ;QUANDO B FOR 0 SIGNIFICA QUE TRANSMITIU TODO O
FRAME.



        POP   B
        POP   DPTR
        POP   PSW
        POP   A


        RET
```

Apêndice D – rotina_shift_bit1

```
;*********************************************************************
;
;****************************** SHIFT_BIT ****************************
;
;*********************************************************************
;

SHIFT_BIT:PUSH      A                       ;SALVA REGISTROS NA PILHA
        PUSH PSW


        MOV  C,23H.0                        ;O BIT P1.0 E COLOCADO NO CARRY ATRAVES
DA POSIÇÃO 23H.0.


        MOV  A,23H                  ;ROTACIONA LSB COM CARRY
        RLC  A
        MOV  23H,A


        MOV  A,24H                  ;ROTACIONA MSB COM CARRY
        RLC  A
        MOV  24H,A

        POP  PSW                    ;RETAURA REGISTROS NA PILHA
```

```
        POP  A

        RET                     ;RETORNA
```

10.4 - Apêndice E – rotina_canalmol1

```
;*********************************************************************
;
;*********************************************************************
;
;********SUB ROTINA QUE APONTA E GUARDA OS VALORES DOS CANAIS**********
;*****************NA MOLDURA PARA TRANSMISSAO************************
;*********************************************************************
;
;*********************************************************************
;


CANALMOL:PUSH    A
         PUSH PSW
         PUSH DPTR


         MOV   A,2DH            ;DEFINE POSICAO DOS CANAIS NA MOLDURA

         CJNE  A,#00H,PRO       ;COMPARO SE E CANAL 1
         MOV   DPTR,#3206H
         JMP   SAIFI

PRO:     CJNE  A,#01H,PRO1      ;COMPARO SE E CANAL 2
         MOV   DPTR,#3208H
         JMP   SAIFI

PRO1:    CJNE  A,#02H,PRO2      ;COMPARO SE E CANAL 3
         MOV   DPTR,#320AH
         JMP   SAIFI

PRO2:    CJNE  A,#03H,PRO3      ;COMPARO SE E CANAL 4
         MOV   DPTR,#320CH
         JMP   SAIFI

PRO3:    CJNE  A,#04H,PRO4      ;COMPARO SE E CANAL 5
         MOV   DPTR,#320EH
         JMP   SAIFI

PRO4:    CJNE  A,#05H,PRO5      ;COMPARO SE E CANAL 6
         MOV   DPTR,#3210H
         JMP   SAIFI

PRO5:    CJNE  A,#06H,PRO6      ;COMPARO SE E CANAL 7
         MOV   DPTR,#3212H
         JMP   SAIFI

PRO6:    CJNE  A,#07H,PRO7      ;COMPARO SE E CANAL 8
         MOV   DPTR,#3214H
         JMP   SAIFI
```

```
PRO7: CJNE  A,#08H,PRO8          ;COMPARO SE E CANAL 9
      MOV   DPTR,#3216H
      JMP   SAIFI

PRO8: CJNE  A,#09H,PRO9          ;COMPARO SE E CANAL 10
      MOV   DPTR,#3218H
      JMP   SAIFI

PRO9: CJNE  A,#0AH,PRO10         ;COMPARO SE E CANAL 11
      MOV   DPTR,#321AH
      JMP   SAIFI

PRO10:      CJNE  A,#0BH,PRO11         ;COMPARO SE E CANAL 12
      MOV   DPTR,#321CH
      JMP   SAIFI

PRO11:      CJNE  A,#0CH,PRO12         ;COMPARO SE E CANAL 13
      MOV   DPTR,#321EH
      JMP   SAIFI

PRO12:      CJNE  A,#0DH,PRO13         ;COMPARO SE E CANAL 14
      MOV   DPTR,#3220H
      JMP   SAIFI

PRO13:      CJNE  A,#0EH,PRO14         ;COMPARO SE E CANAL 15
      MOV   DPTR,#3222H
      JMP   SAIFI

PRO14:      MOV   DPTR,#3224H          ;CANAL 16

      MOV   2DH,#00H             ;ZERA POSICAO 2DH PARA NOVO CICLO DE TIMER 0.

SAIFI: ACALL LE_CANAL                  ;PREPARA MSB E LSB DE CADA CANAL NOS
ENDEREÇOS
                                       ;RESPECTIVOS 26H E 25H.


      MOV   A,26H                  ;SALVA MSB NA MOLDURA NA POSIÇÃO DEFINIDA
POR DPTR.
      MOVX  @DPTR,A


      INC   DPTR
      MOV   A,25H                  ;SALVA LSB NA MOLDURA NA POSIÇÃO DEFINIDA
POR DPTR.
      MOVX  @DPTR,A


      POP   DPTR
      POP   PSW
      POP   A
```

```
        RET

;*********************************************************************
;*********************************************************************
;*********************************************************************
;*********************************************************************
```

10.5 - Apêndice F – PADRAO3

```
        ORG   2500H

;*********************************************************************************
;*****************ORG - 2500H ÁREA DO PROGRAMA PRINCIPAL**************************
;*********************************************************************************
;*********************************************************************************
;COMENTÁRIO: NO PROGRAMA PRINCIPAL FAZ - SE TODAS AS INICIALIZAÇÕES
;DE POSIÇÕES DE MEMÓRIA INTERNA E EXTERNA, INTERRUPÇÃO, CONTADORES E
;TIMERS.
;*********************************************************************************
;*********************************************************************************


;*************************************************************
;*************************************************************
;***************INICIALIZA BAUD RATE**************************
;*************************************************************

        MOV   T2CON,#34H          ;PROGRAMA O MESMO BAUDRATE PARA RECEPÇÃO
E TRANSMISSÃO.
        MOV   SCON,#50H           ;CONTROLE DO CANAL SERIAL - MODO 1 REN=1
HABILITA RX LOGO
                                  ;QUE O START BIT SERA DETECTADO.

        MOV   RCAP2H,#FFH         ;F = 12MHZ BAUD RATE 9600BPS
        MOV   RCAP2L,#D8H
        MOV   TH2,#FFH
        MOV   TL2,#D8H


;*************************************************************
;*************************************************************
;***************INICIALIZA STACK******************************
;*************************************************************

        MOV   SP,#50H                     ;PILHA A PARTIR DA POSIÇÃO 50H DA RAM
INTERNA (51H)


;*************************************************************
;*************************************************************
;***************INICIALIZA PORT*******************************
```

```
;*************************************************************

        MOV   P1,82#H                   ;P1.0 = 0   - SDATA - SAIDA SERIAL DO
CONVERSOR
                                        ;P1.1 = 1  - CONVERT - INICIO DE CONVERSAO
                                        ;P1.2 = 0    - SCLK - CLOCK EXTERNO PARA
SERIALIZAÇÃO DOS DADOS
                                        ;P1.3 = 0  - TRK1 - INDICA FIM DE CONVERSAO
                                        ;P1.4 = 0  - STBY - INICIALIZACAO DO CONVERSOR
                                        ;P1.5 = 0
                                        ;P1.6 = 0
                                        ;P1.7 = 1 - RESET WATCH DOG


                        ;NOTA: SCLK = 0 INICIALMENTE, GARANTE A PRIMEIRA
                        ;LEITURA   DO   BIT   CONVERTIDO   QUANDO   P1.1   FOR   A
ZERO.(CONVERT)

;*************************************************************
;
;*************************************************************
;
;*************************************************************
;
;***************INICIALIZA TIMER0 E TIMER1********************
;*************************************************************
;
        MOV   TMOD,#11H             ;TIMERS TRABALHANDO NO MODO 1
                                    ;M1.X E M0.X=1

        MOV   TCON,#00H             ;ZERADO INICIALMENTE PARA GARANTIR O
                                    ;NAO ACIONAMENTO DO TIMER CADA TIMER
                                    ;SERA MANIPULADO POR SOFTWARE.


;*************************************************************
;
;*************************************************************
;
;*****************PROGRAMAR INTERRUPÇÕES**********************
;*************************************************************
;

        MOV   IE,#80H                   ;EA=1    PERMITE    ESCOLHER    QUAL
INTERRUPÇÃO SERA
                                        ;HABILITADA  SEGUNDO  OS  BITS  INDIVIDUAIS  DE
CONTROLE.


;*************************************************************
;
;*************************************************************
;
;*********PROGRAMAR O TIMER E INICIAR SUA CONTAGEM************
;*************************************************************
;
```

```
        MOV   B,#09H

L1:     MOV   TL0,#F1H      ;TEMPORIZACAO DA REPETICAO DO FRAME
        MOV   TH0,#0CH      ;TEMPO 560 mS F=12MHZ
                                ;T=83,3nS

        SETB  TR0           ;DISPARA TIMER 0.

        JNB   TF0,$         ;AGUARDA ESTOURO DO CONTADOR TCON.5=1

        CLR   TF0
        CLR   TR0

        DJNZ  B,L1

        SETB  ET0           ;HABILITA A INTERRUPCAO DO TIMER 0 QUANDO
EA=1



;*************************************************************
;*************************************************************
;*******INICIALIZA RAM INTERNA - PALAVRA DE STATUS************
;*************************************************************


        MOV   20H,#00H      ;INDICA STATUS DO BUFFER - 256 OPCOES
                            ;20H.0 -
                            ;20H.1 -
                            ;20H.2 -
                            ;20H.3 -
                            ;20H.4 -
                            ;20H.5 -
                            ;20H.6 -
                            ;20H.7 -


        MOV   21H,#00H      ;SOLUÇÃO MOMENTANEA PARA RODAR APENAS UM
MODO.

        MOV   22H,#00H      ;PONTEIRO PARA OS CANAIS (CONTA 16 CANAIS)

;PEGA BIT A BIT DO CONVERSOR ADC E MONTA OS BYTES MSB E LSB DO VALOR
CONVERTIDO
```

```
        MOV   23H,#00H          ;PONTEIRO LSB
        MOV   24H,#00H          ;PONTEIRO MSB
        MOV   25H,#00H          ;LSB
        MOV   26H,#00H          ;MSB


        MOV   27H,#00H          ;BCONT - CONTA 16 BITS DA SAIDA SDATA DO
CONVERSOR AD.
        MOV   28H,#FFH          ;CONTADOR PARA OS TESTAR SE JA LEU OS 16
CANAIS


;CONTADOR DE 32 BITS, CONTAGEM DOS FRAMES.

        MOV   29H,#00H          ;CONTADOR DE LSB
        MOV   2AH,#00H          ;CONTADOR DE LSB
        MOV   2BH,#00H          ;CONTADOR DE MSB
        MOV   2CH,#00H          ;CONTADOR DE MSB


        MOV   2DH,#00H          ;DEFINE O NUMERO DE CANAIS NA MOLDURA

;*************************************************************
;
;*************************************************************
;
;***************INICIALIZA RAM EXTERNA************************
;**ROTINA PARA ZERA 256 POSIÇÕES DE MEMORIA NA RAM EXTERNA****
;*************************************************************
;


        MOV   B,#FFH            ;COLOCAR O VALOR #FFH NO REGISTRADOR B.
        MOV   DPTR,#3200H       ;COLOCAR O VALOR #3200H NO REGISTRADOR DPTR.
        MOV   A,#00H            ;CARREGA ACUMULADOR COM VALOR #00H.

ZERA:  MOVX  @DPTR,A                ;MOVA ACUMULADOR PARA RAM EXTERNA
ENDEREÇO 16BITS.
        INC   DPTR              ;INCREMENTA PARA PROXIMA POSIÇÃO DE MEMORIA
(3201H)
        DJNZ  B,ZERA                ;DECREMENTA VALOR INICIAL DE B (#FFH)
ATE ZERAR.



;*************************************************************
;
;*************************************************************
;
;*********************MONTAGEM DO FRAME************************
;*************************************************************
;
;*************************************************************
;


        MOV   DPTR,#3200H
        MOV   A,#EBH
        MOVX  @DPTR,A               ;COLOQUEI #EBH NA POSIÇÃO 3200H
```

```
        INC     DPTR
        MOV     A,#90H
        MOVX    @DPTR,A                 ;COLOQUEI #90H NA POSIÇÃO 3201H


        ;NOTA: INICIALMENTE COLOCO APENAS EB90H INDICANDO INICIO DO FRAME.
        ;AS DEMAIS PALAVRAS SERA ORIENTADAS PARA OS NOVOS ENDERECOS QUE SE
SEGUEM A 3201H

        ;NO FUTURO IMPLEMENTAREMOS MAIS QUATRO BYTES NO FINAL DO FRAME,
CKSUM,ETC.



;*************************************************************
;
;*************************************************************
;
;**********LER BUFFER PARA SABER O MODO DE OPERACAO**************
;*************************************************************
;
;*************************************************************
;

        MOV     DPTR,#4000H     ;APONTADOR PARA AREA DE MEMORIA RESERVADA
DO BUFFER.
        MOVX    A,@DPTR                 ;TRANFERE O CONTEUDO DA POSIÇÃO 4000H
PARA ACUMULADOR.


        MOV     20H,A           ;CARREGO CONTEUDO DO BUFFER NA PALAVRA DE
STATUS.


        MOV     A,20H           ;INICIALMENTE TEREMOS APENAS O PROGRAMA
PADRAO
                                ;RODANDO A CADA INTERRUPÇÃO DO TIMER 0.

        CJNE    A,#00H,MODO01   ;FUTURAMENTE TRATAREMOS CADA MODO
IDEPENDENTEMENTE.

        JMP     MODO00

MODO01:     CJNE    A,#01H,MODO02

        JMP     MODO01



MODO00:     SETB    21H.0
```

```
VOLTA:      JB      21H.0,VOLTA  ;APOS SELEÇÃO DO MODO FICA NUM LOOP INFINITO
                                 ;SAINDO PELA INTERRUPÇÃO DO TIMER 0.
```