Ciências Espaciais Atmosféricas
Divisão de Aeronomia
Ionosfera
Relatório Final
(Período jul/99 – jun/00)

Bolsista

Maria Eugenia Carvalho Pontedeiro

Orientador

prof º Dr. Polinaya Muralikrishna

Título

ESTUDOS DA IONOSFERA GLOBAL A PARTIR DE EXPERIMENTOS LANÇADOS A BORDO DE FOGUETES E SATÉLITES

São José dos Campos
Junho,2.000

AGRADECIMENTO

Agradeço ao **INPE**, ao **Dr. Polinaya Muralikrishna**, pelo incentivo do trabalho na área de Geofísica Espacial.

Ao CNPq pela bolsa de iniciação científica.

A DEUS. A meus pais e ao Carlos Eduardo, que estão sempre me apoiando.

Aos colegas de trabalho, Goreti, Lúcia, Maureni, Luciana, Dalli, Vanderli, Alcione, Vivian, que me ajudam diariamente no trabalho.

ÍNDICE

LISTA DE FIGURAS

1. INTRODUÇÃO

O trabalho apresentado neste relatório é baseado nos resultados obtidos durante os testes dos experimentos PLASMEX realizados nos laboratórios do INPE através do software LABVIEW, e nos preparativos para análise dos dados científicos esperados dos lançamentos de SACI-1 e SACI-2.

Também incluído neste relatório é o trabalho de desenvolvimento de software para a análise de dados obtidos dos experimentos lançados a bordo de foguetes com o objetivo de obter informações sobre a distribuição espectral das irregularidades de plasma nas diferentes regiões da ionosfera.

O software CURFIT (Dr. Delano, 1.997) para análise espectral baseado no método de entropia máxima, também foi estudado neste período.

1.1 A IONOSFERA

A ionosfera é a região da atmosfera terrestre que se encontra parcialmente ionizada por radiação solar e está localizada aproximadamente entre 50 à 1.000 km acima da crosta terrestre. Esta tem a propriedade condutiva (condutora de eletricidade), possuindo íons e elétrons em quantidade suficiente para influenciar na propagação de ondas eletromagnéticas tais como as ondas de radio. Na presença do campo magnético terrestre o plasma ionosférico é sujeito a vários processos dinâmicos e eletrodinâmicos. Os mecanismos, conhecidos, bem como desconhecidos, de instabilidade de plasma geram irregularidades de plasma tais como "bolhas de plasma". Esta camada também recebe o nome de "plasma ionosférico neutro", pois o número de íons e elétrons são iguais. É formada pela irradiação solar e tende a diminuir de densidade, tornando o plasma mais rarefeito durante a noite, devido a recombinação de seus íons e elétrons.

Os fenômenos: espalhamento-F (Spread-F) e o eletrojato equatorial são manifestações dos processos físicos e eletrodinâmicos que ocorrem na ionosfera equatorial. Estudamos esta camada, a ionosfera, para:

- Entender o conjunto de processos dinâmicos e eletrodinâmicos observados nessa camada;

- Entender os efeitos que atingem as telecomunicações;
- Estudar a física de plasmas e os fenômenos de instabilidade de plasma;
- Estudar os processos quânticos que ocorrem em átomos e moléculas atmosféricas excitadas;
- Estudar a espectroscopia óptica;

Ocorrem fenômenos únicos no globo terrestre, da ionosfera na região brasileira. As regras seguidas aqui são peculiares por causa da sazonalidade ser distinta das de outros setores longitudinais. Os radioamadores operam numa faixa de frequência bastante vulnerável à interferência das bolhas.

1.2 A INSTABILIDADE DO PLASMA

As irregularidades de plasma são produzidas pela instabilidade do plasma ionosférico, estas que afetam a comunicação de ondas de rádio. As comunicações via satélites, também, sofrem altíssimas interferências ionosféricas na região equatorial. Como exemplo das manifestações destas irregularidades temos as bolhas de plasma, Spread-F, Sporadic – E.

1.2.1 Spread-F

O Spread-F, compreende-se em regiões do espaço de dimensões extraordinariamente grandes, é um fenômeno gerado pelas bolhas ionosféricas. Estas bolhas, que são regiões de baixa densidade eletrônica e alinhadas ao campo magnético terrestre, são formadas em baixas latitudes e se desenvolvem na ionosfera noturna variando na frequência de ocorrência de acordo com a estação do ano e da longitude do setor equatorial. Em 1976, utilizando técnicas ótica, o Observatório Óptico da Divisão de Aeronomia do INPE, localizado em Cachoeira Paulista-SP, detectou pela primeira vez a ocorrência de bolhas ionosféricas sobre o continente sul americano.

1.2.2 Eletrojato Equatorial

O Eletrojato Equatorial é uma corrente elétrica espacial que está situado no equador magnético terrestre, aproximadamente 100-110 km de altura e alguns graus de latitude de largura.

Nesta região, o plasma ionosférico é altamente instável provocando vários tipos de irregularidades.

Este fenômeno ocorre durante o dia, quando há um crescimento da condutividade elétrica como resultados de ventos neutros, e da geometria horizontal do campo magnético nesta região.

1.3 EQUIPAMENTO PARA COLETA DE DADOS

A Divisão de Aeronomia do INPE vem usando alguns equipamentos para coleta de dados sobre o plasma ionosférico. Estes equipamentos tem como objetivo principal a compreensão dos mecanismos físicos da geração, desenvolvimento e deterioramento em geral do plasma na região ionosférica global e na região ionosférica acima do Brasil em particular.

1.3.1 Ionossonda

Este equipamento mede parâmetros ionosféricos. É um sistema transmissor-receptor que emite pulsos de energia eletromagnética em frequência variável, geralmente de 1 a 5 MHZ. O sinal, que é emitido quase sempre na vertical, é refletido pela ionosfera. Cada frequência é refletida de uma região da ionosfera onde a frequência do plasma é igual à frequência transmitida. Assim medido o tempo decorrido entre a transmissão e a recepção de uma frequência, pode-se obter o perfil da densidade eletrônica.

1.3.2 Digissonda

É um equipamento digital, preciso e moderno, que funciona com o mesmo princípio da ionossonda.

1.3.3 Receptor de GPS

O GPS (Global Positioning System) é um sistema de navegação usado para determinar a posição e a velocidade de um objeto, fixo ou móvel, este estando sobre ou próximo da superfície terrestre, usando sinais de 24 satélites em órbita na Terra.

1.3.4 Carga útil

Cargas úteis são experimentos científicos ou técnicos que são lançados a bordo de foguetes, balões ou satélites.

Quando expostos ao plasma ionosférico os sensores destes experimentos podem medir parâmetros como os campos elétricos e a densidade eletrônica do local. Várias cargas úteis já foram lançados a bordo de foguetes brasileiros e estrangeiros (pela divisão DAE). Recentemente foi lançada a carga útil PLASMEX a bordo do satélite SACI-1 e SACI-2

1.3.5 Polarímetro

O polarímetro é um equipamento transmitido por satélite geoestacionário, este atravessa as camadas ionosféricas após receber um sinal polarizado linear. O estudo da variação integral da densidade eletrônica das camadas, e das perturbações (variações irregulares da densidade) causando espalhamento do sinal enviado pelo satélite, dá-se pela medida da fase e o ângulo de amplitude que o sinal permite.

2. MICROSATÉLITE SACI-1

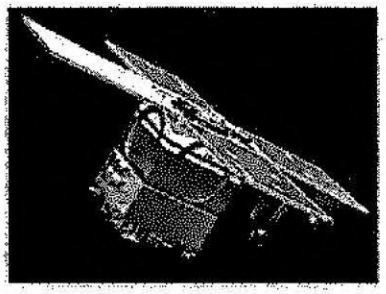

Figura 1 - SACI 1 aberto

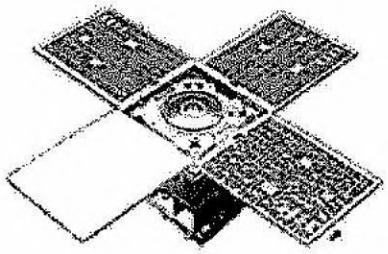

Figura 2 - SACI 1 aberto

O primeiro microsatélite científico Brasileiro SACI-1, (Satélite de Aplicações Científicas –1) foi lançado a partir do Centro Chinês de Lançamento de Taiyuan, no dia 14 de outubro de 1999, simultaneamente com o Satélite Sino-Brasileiro de Recursos Terrestres(CBERS-1). O SACI era um micro-satélite científico de órbita polar quase circular, com ele estavam quatro cargas úteis: MAGNEX, FOTOEX, ORCAS, PLASMEX.Iria monitorar a atmosfera e a ionosfera superior em escala global, mas apesar de ter entrado na orbita prevista, o SACI-1, infelizmente, não conseguiu mandar para a

estação terrestre os dados dos experimentos a bordo. O principal objetivo era desenvolver um microsatélite científico brasileiro e munir a comunidade científica brasileira, para que com isso:

- As pesquisas científicas tivessem um custo moderado;
- A cooperação nacional para a ciência do espaço e tecnologia aumentasse;
- O desenvolvimento de espaços científicos fosse promovido, além do INPE, no Brasil;
- A distância fosse reduzida e as faltas deixadas pelos grandes programas fossem preenchidas.

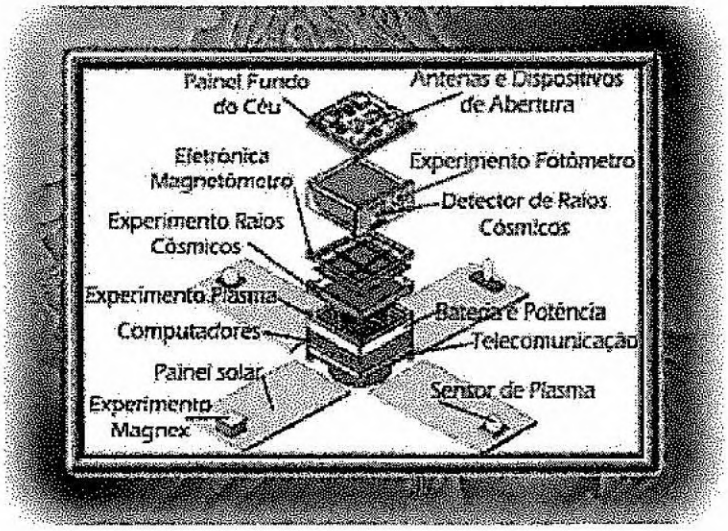

Figura 3 - partes do SACI 1

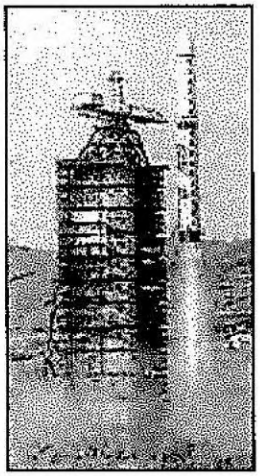

Figura 4 - lançamento do SACI 1

Figura 5 - Órbita do SACI 1

O PLASMEX foi um dos experimentos selecionado pela Academia Brasileira de Ciências para ser lançamento a bordo do SACI-1.

O principal objetivo cientifico do conjunto dos experimentos PLASMEX, era o estudo das bolhas. O SACI oferecia perfeitas condições para obter dados sobre essas bolhas, pois o desenvolvimento das bolhas de plasma atinge sua intensidade máxima aproximadamente às 22:00hs e a órbita sincrônica do satélite em relação ao sol era localizada, também, à aproximadamente 22:00hs.

2.1 PLASMEX

Os experimentos PLASMEX eram propostos para investigar o fenômeno das bolhas ionosféricas, ou depleções ionosféricas. As bolhas de plasma associadas com a turbulência influenciam em sistemas de aplicações espaciais diversos.

Era pretendido fazer medidas de densidade de plasma, densidade de elétrons, distribuição espectral das irregularidades do plasma e temperatura cinética dos elétrons usando os seguintes experimentos:

- Sonda de Capacitância em Alta Frequência (HFC), por medir a densidade de plasma;
- Sonda de Langmuir (LP), por medir o perfil de densidade de elétrons e a distribuição espectral das irregularidades do plasma;
- Sonda de Temperatura de Elétron (ETP), por medir a temperatura cinética dos elétrons ionosféricos.

2.1.1 HFC

O HFC (Sonda de Capacitância de Alta Frequência) era um sensor esférico, associado a um oscilador para medir, com alta precisão, a distribuição da densidade do plasma. Estava montado atrás de um painel solar de maneira que reduziria a interferência de outros experimentos e do potencial flutuante do satélite. A frequência do oscilador estava sendo medida através de um sistema de contador de pulso, na taxa de 12.5 medidas por segundo, e assim transmitiria 3 palavras digitais de 8 bit (HFC0, HFC1, HFC2). O HFC0 continha o LSB (menos significativo); HFC1 e HFC2 continha o MSB (mais

significativo) e o HFC3 (variando de 0 à 5 V), um sinal analógico iria monitorar a performance do experimento.

Estes dados digitais seriam adquiridos e armazenados na memória do microcontrolador PLASMEX, que transmitiria para o OBC (On Bord Computer) por uma porta serial (RS-422 serial output).

2.1.2 LP

O LP (Sonda Langmuir Probe) consistia em um sensor metálico esférico, para medir a densidade eletrônica e a distribuição espectral de irregularidades de plasma. Este experimento precisava de um sensor esférico, este atrás de um dos painéis solares, dividido em dois módulos, de maneira a reduzir a interferência de outros experimentos, e do potencial flutuante do satélite.Era usado um sistema de amplificador para medir a corrente do sensor LP, que era medido na taxa de 12.5 exemplos/segundos e a AC na taxa de 1600 exemplos/segundos, sendo transmitidos por palavras digitais de 8 bit, LPSDC, LPSAC, LPINT, LPMGB. Os dados adquiridos junto com os de outros experimentos PLASMEX iam para a memória local do microcontrolador PLASMEX para que assim fosse transmitido ao OBC. O LP estando no modo processamento de bordo, o LPSAC estava sujeito a ser transmitido para o OBC com um FFT a bordo e os dados espectrais que seriam guardados na memória para a transmissão.

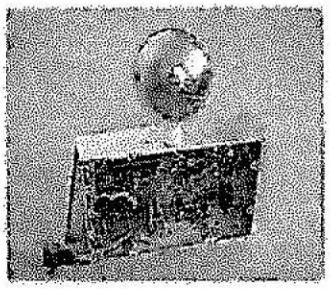

Figura 6-LP(Sonda Langmuir Probe)

2.1.3 ETP

O ETP (Electron Temperature Probe), iria medir a temperatura cinética dos elétrons. Este equipamento continha dois sensores semicirculares e a unidade eletrônica. Estes sensores estavam montados em uma das extremidades de um dos painéis solares de maneira que reduziria a interferência de outros experimentos, e do potencial flutuante do satélite. Um sinal RF de 30 Khz era aplicado a um dos sensores e um sistema de amplificadores era usado para medir na curva IV do sensor (corrente X voltagem). A temperatura eletrônica poderia ser estimada a partir destas variações. A partir do sinal recebido do segundo sensor o potencial flutuante do espaço poderia ser estimado. A taxa de aquisição era 12.5 exemplos/segundos, estes que seriam transmitidos em palavras de 8 bit ETP0 e ETP1.

3. SACI-2

A bordo de um foguete Brasileiro VLS (Veículo Lançador de Satélite) do Centro de Lançamento de Alcantara-MA iria ser lançado o SACI-2 mas, infelizmente , o VLS não conseguiu colocar o SACI-2 na orbita prevista, resultando a perda total dos dados científicos. O SACI-2, com uma orbita de baixa inclinação, deveria fornecer dados científicos na região equatorial da ionosfera e também cumprir a missão de coleta de dados meteorológicos junto ao SCD-2 e também CBERS – China Brasil Earth Resource Satélite, de órbita polar. Este, que era um satélite bastante versátil, foi adequado rapidamente à missão de coleta de dados e assim garantiria a continuidade dessa fase da MECB a um custo muito baixo. Ele possibilitaria ainda uma cobertura temporária para eventual falha de um satélite da Missão de Coleta de Dados. A capacitação tecnológica que vinha da simplificação das técnicas empregadas, criava produtos de alta tecnologia a partir de elementos industriais disponíveis. Isto se traduz por uma maior participação das indústrias convencionais locais em missões espaciais e aumentaria o nível de nacionalização e independência. No projeto dos microsatélites científicos, subsistemas e equipamentos são desenvolvidos com pouco investimento. Tais equipamentos, após a comprovação efetiva durante os vôos, tornam-se produtos confiáveis e de baixo custo para novas missões.

Início do desenvolvimento: maio 1998.

Término da construção: setembro 1999.

Lançamento: Novembro 1999.

Total: 17 meses.

O SACI-2 também levou a bordo o conjunto de experimentos PLASMEX para estudos do plasma ionosférico devido a sua grande importância para a comunidade científica brasileira, permitindo a ampliação a base de dados para análise e diversificando as condições de observação. Além disso houve um aumento da participação brasileira nos experimentos, sendo que cinco dos seis sensores foram fabricados no Brasil. SACI-2 entraria numa órbita equatorial, com 17,5 graus de inclinação, o número de passagens com visada para estações em Alcântara e Natal iria ser duas vezes maior do que o de visadas aproveitáveis por uma estação localizada em São José dos Campos.

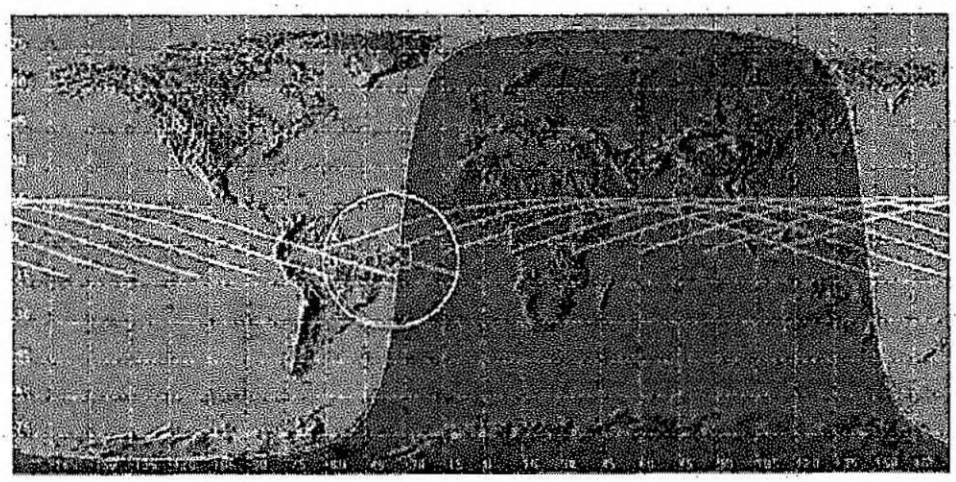

Figura 7 - Órbita equatorial do saci 2

4. LabVIEW

4.1 INTRODUÇÃO

Um software chamado LabVIEW foi usado para a visualização dos dados recolhidos pelos experimentos PLASMEX. LabVIEW é uma linguagem de programação gráfica (em forma de diagrama de bloco), foi desenvolvida pela NATIONAL INSTRUMENTS CORPORATION. Esta linguagem possui ferramentas para o desenvolvimento de sistemas como: C, BASIC, PASCAL, entre outras. "Virtual instruments" é o nome que recebe os programas em LabVIEW, isto porque as bibliotecas de funções e ferramentas de desenvolvimento são para aquisição de dados e simulação de instrumentos de controle. VIs são uma espécie de interface com o usuário, código fonte equivalente a troca de parâmetros entre as VIs de nível mais alto.

FRONT PANEL (painel frontal) é uma interface interativa com o usuário, isto porque ela simula um painel físico de um instrumento. Este contém botões, gráficos, entre outros controles e ainda indicadores que permitem a inserção ou a visualização de dados. BLOCK DIAGRAM(diagrama de bloco) mandam instruções para as VIs, seria a solução do problema. Este é o código fonte das VIs.

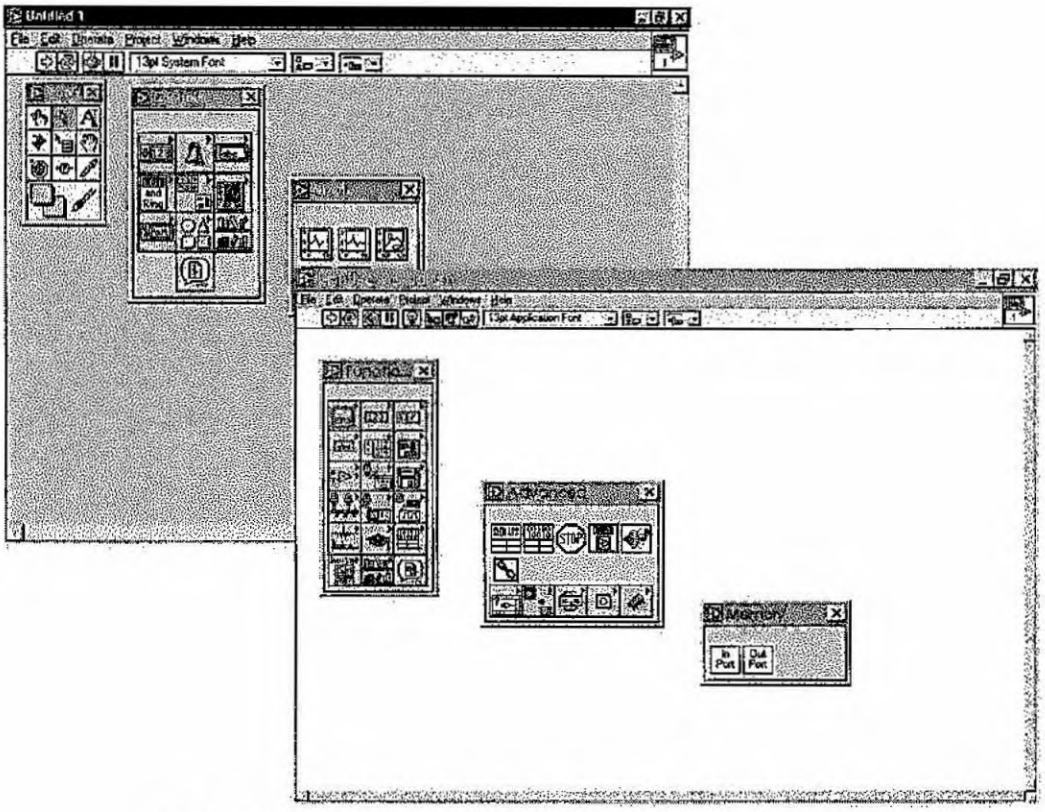

Figura 8 - FRONT PANEL (cinza) BLOCK DIAGRAM (branco)

4.2 *Programa de visualização dos Dados Adquiridos pelo Experimento PLASMEX do Microsatélite SACI- 1*

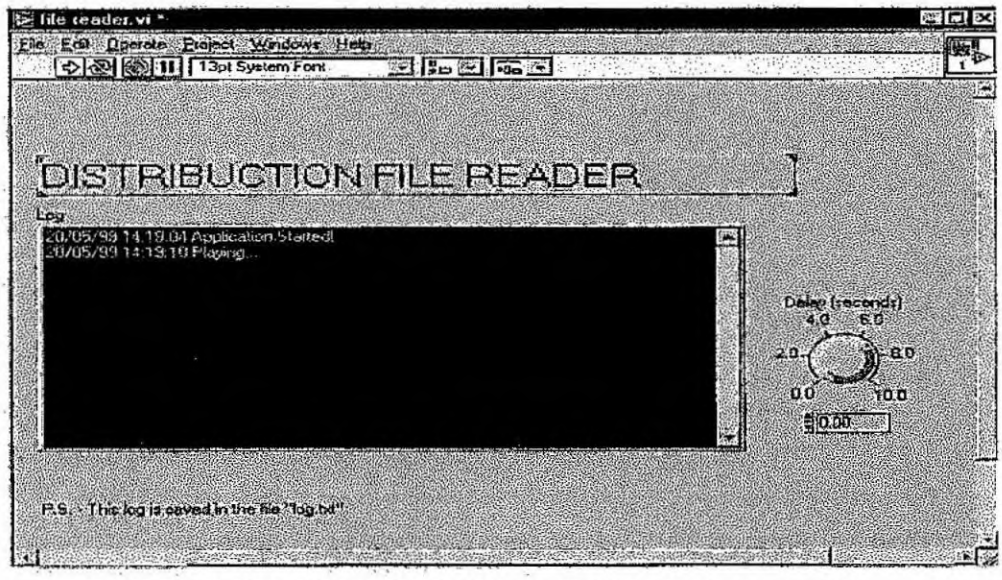

Figura 9 - FILE READER.vi VI principal (frente do programa)

4.2.1 PROCEDIMENTO

Para chegar até este programa, utilizamos o seguinte procedimento:

- Instalação completa do software LabVIEW 4.0 e do programa PLASMEX ;
- Na biblioteca PLASMEX.llb, está VI principal FILE READER.vi(figura 9);

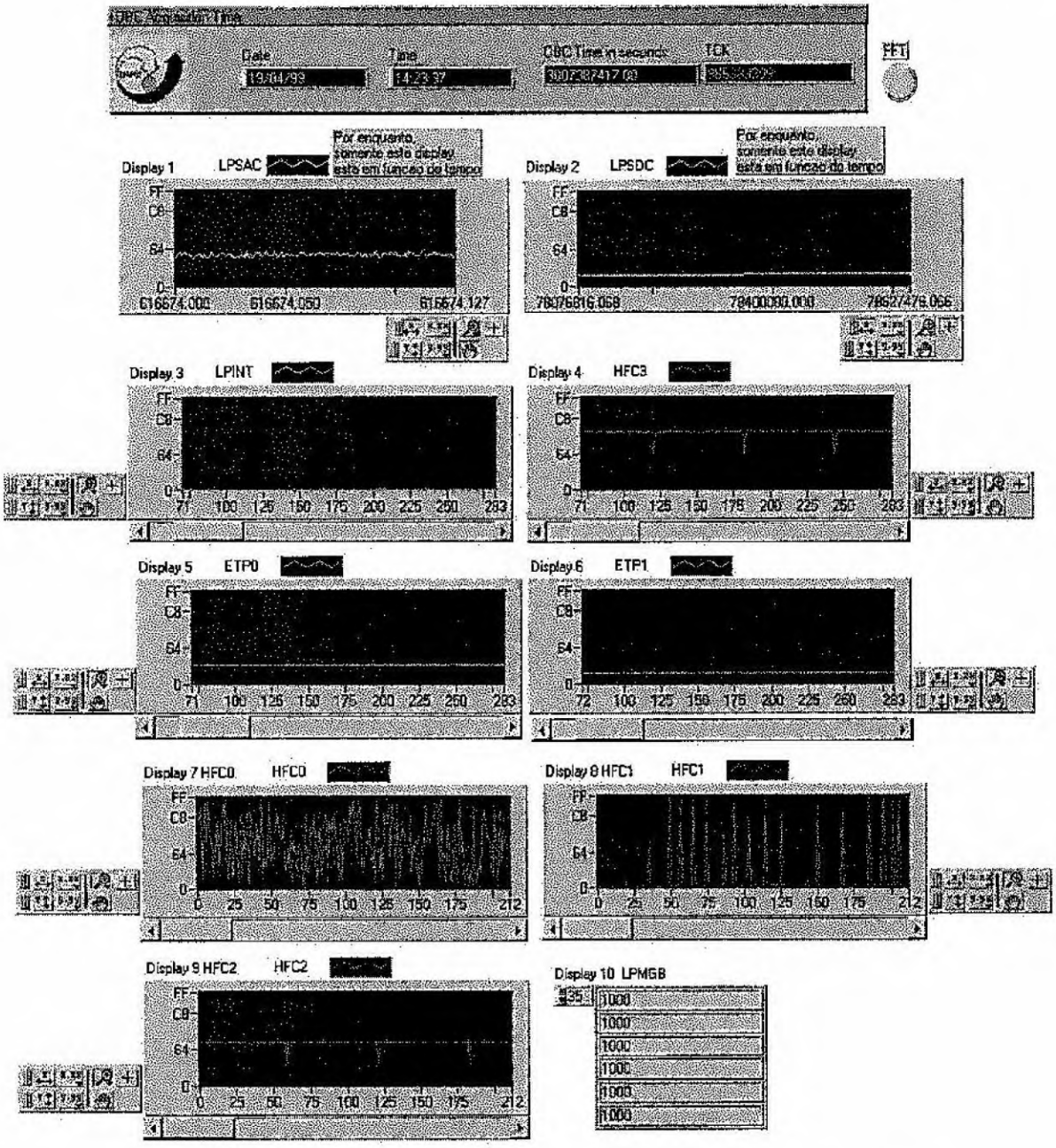

Figura 10 - subprograma BREAK FRAME FOR PLASMEX WITH TIME.vi

4.2.2 EXECUÇÃO

Esta VI chama as outras VIs, por isso VI principal.. Executando este programa ele nos mostrará o subprograma BREAK FRAME FOR PLASMEX WITH TIME.vi, uma tela com vários display (figura 3). Este programa lia os dados do experimento PLASMEX (LPSAC, LPSDC, LPINT, ETP0, ETP1, HFC0,HFC1, HFC2,HFC3) e os coloca, cada um, em um display, para que nesses sejam observados os valores dos experimentos. Os dados, que vinham do microsatélite, erão anteriormente separados, assim o programa os lia separadamente.

4.2.3 GRÁFICOS

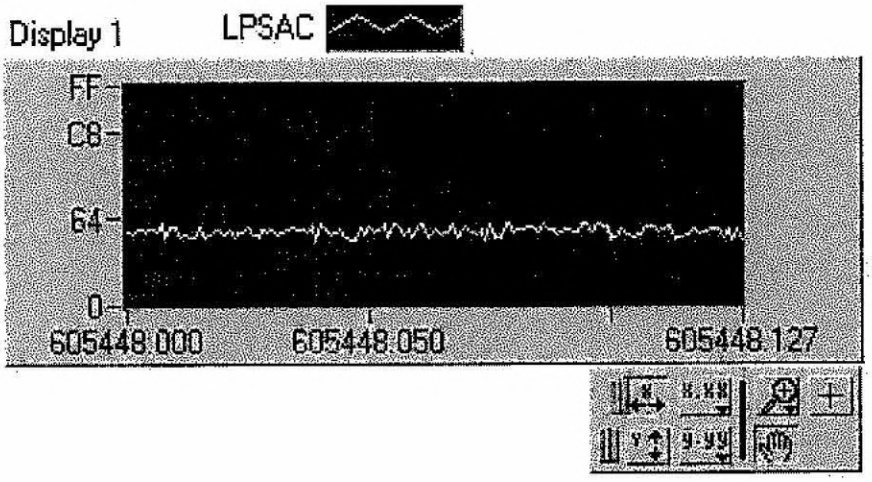

Figura 11 - Amostra de LPSAC

O canal LPSAC representa o sinal flutuante do sensor de "Sonda de Langmuir" (Langmuir Probe – LP). Pela análise deste sinal podemos obter informações sobre a distribuição espectral das irregularidades de plasma ionosférico.

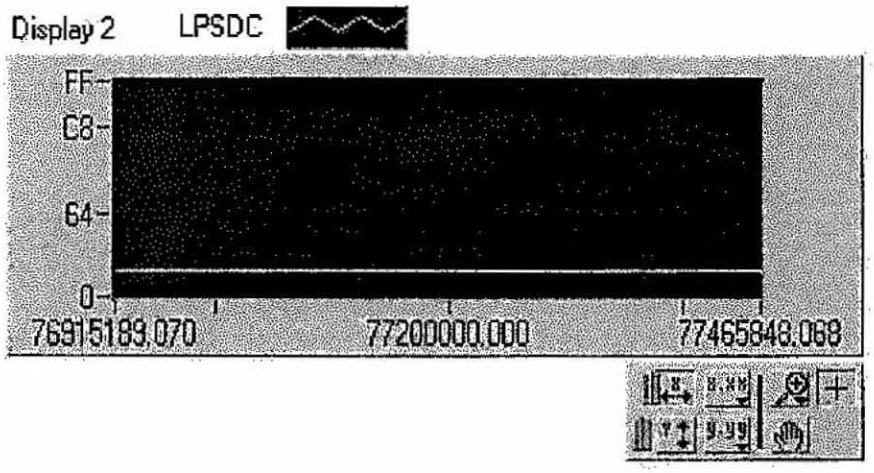

Figura 12 - Amostra do LPSDC

Este canal mede a corrente recolhida pelo sensor de LP (Sonda de Langmuir), e representa a variação com a altura, da densidade eletrônica.

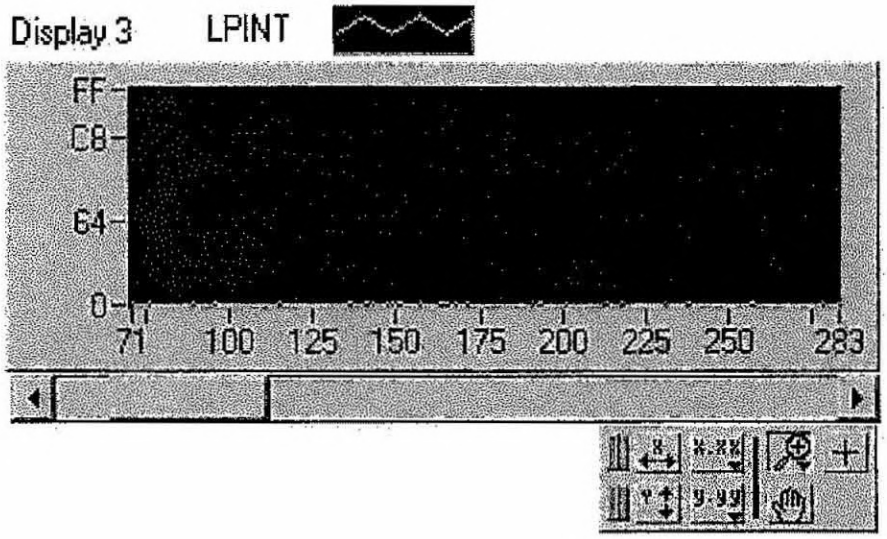

Figura 13 - Amostra de LPINT

Este é um canal integral, representa o sinal do sensor de LP incluindo as flutuações e as variações lentas. Na ausência do plasma, o nível é praticamente zero.

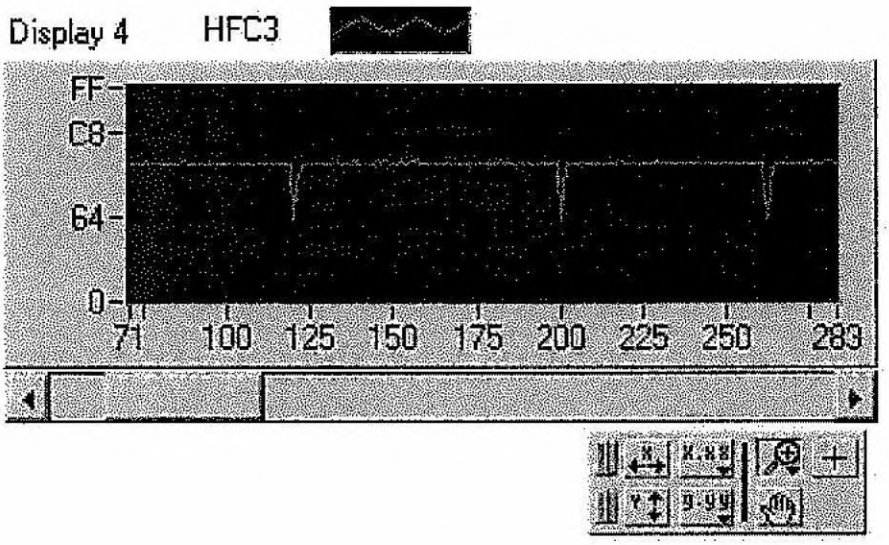

Figura 14 - amostra do HFC 3

O HFC3,um sinal analógico que está entre 0 e 5 V e monitora a performance do experimento HFC. Este sinal é convertido em uma palavra digital de 8 bits em uma taxa de 12.5 exemplos por segundo.

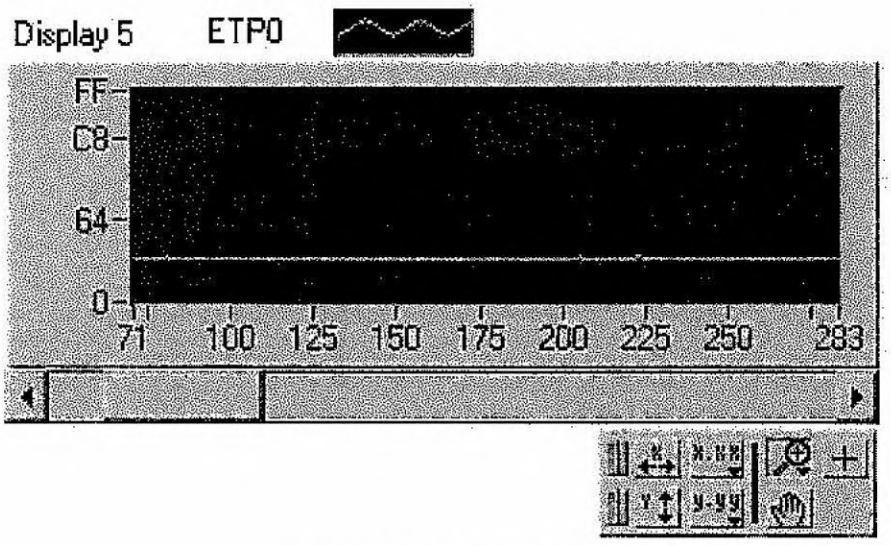

Figura 15 - Amostra do ETP 0

Este canal representa um sinal do experimento "Sonda de temperatura Eletrônica" (ETP) que poderá ser usado para determinar a variação da temperatura cinética dos elétrons de plasma ionosférico com a altura .

Parâmetro estimado: Te (temperatura eletrônica).

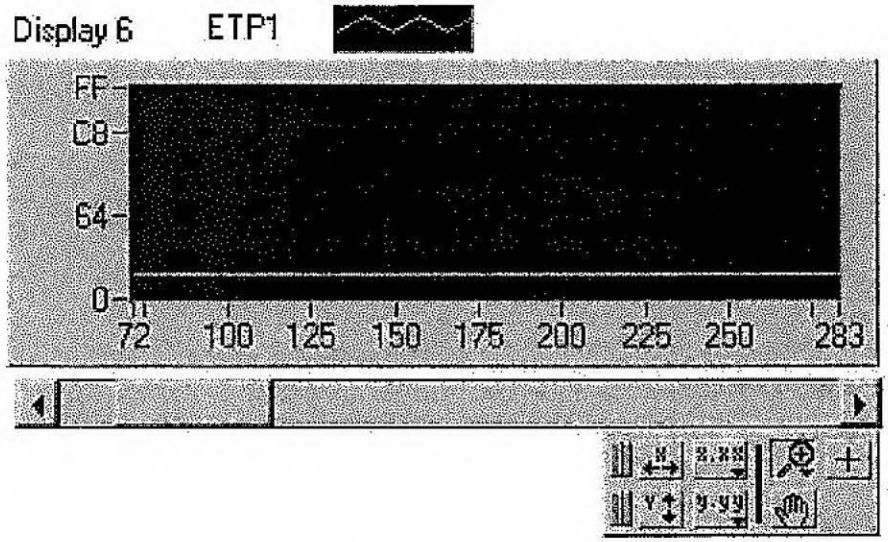

Figura 16 - Amostra do ETP 1

As flutuações no potencial flutuante do sensor do experimento ETP, é medido por este canal.

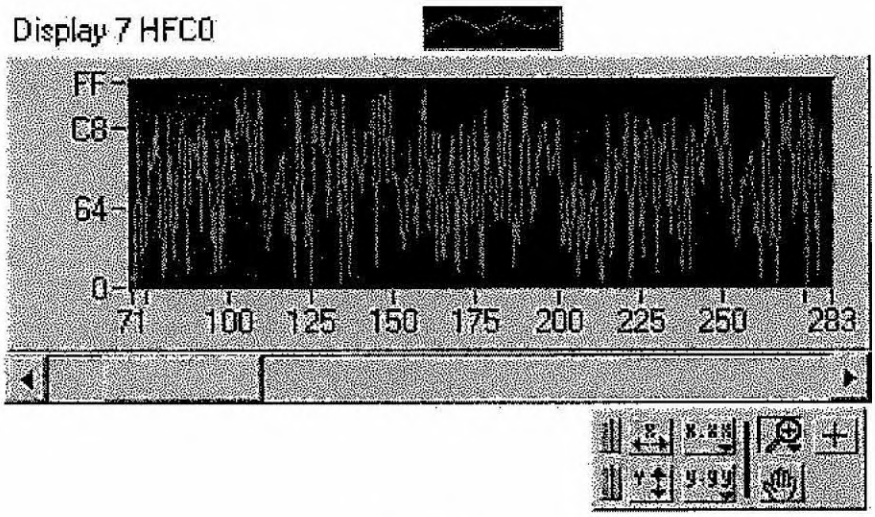

Figura 17 - Amostra do HFC 0

O HFC 0 é um canal digital de 8 bits, que junto com o HFC 1 e HFC2 representam três canais contendo informações sobre a frequência de oscilação do experimento HFC (High Frequency Capacitance Probe). Sendo que este canal é o menos significativo.

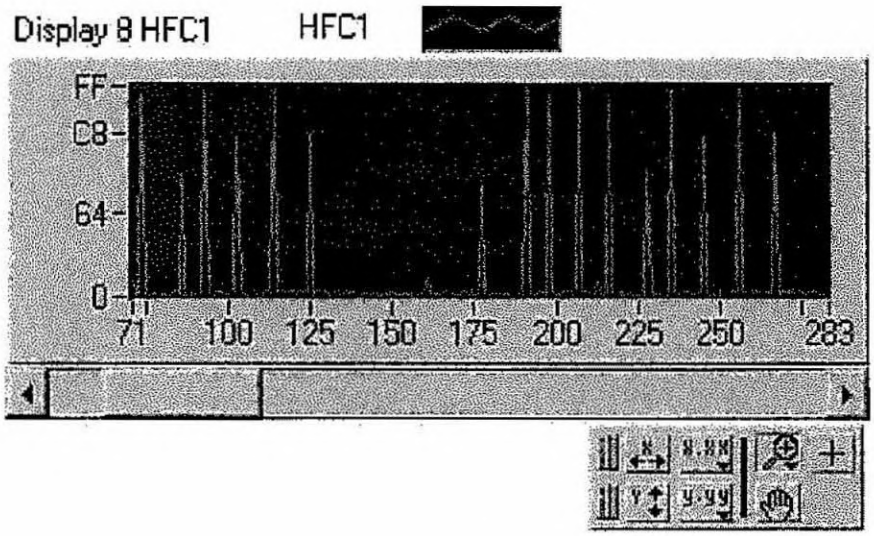

Figura 18 - Amostra do HFC 1

HFC 1 é o segundo canal digital com informações da frequência de oscilação do experimento HFC . Este canal ,também, tem 8 bits sendo que este são de significado médio.

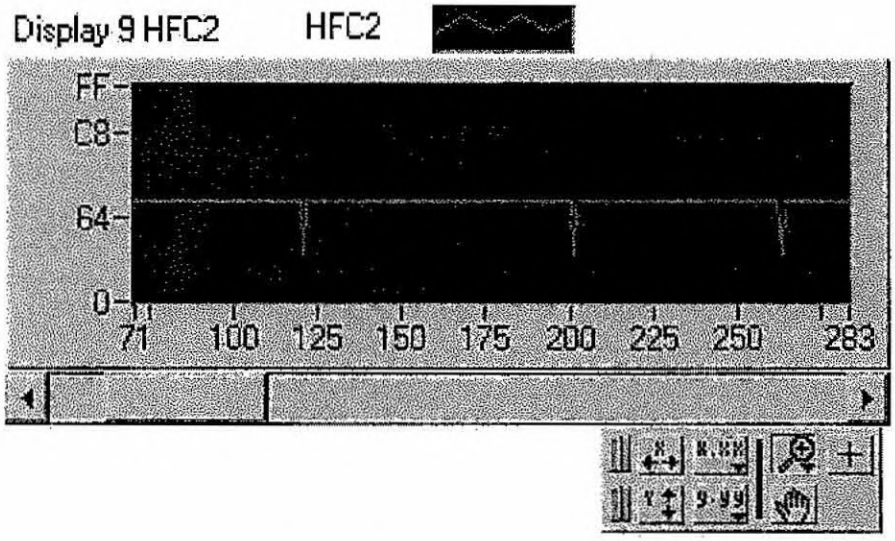

Figura 19 - Amostra do HFC 2

Com o HFC 2 finalizamos os três canais digitais de informações sobre a frequência de oscilação do experimento HFC. Este canal contém os 4 bits mais significativos, e mais 4 bits que contém informações do monitor de operação.

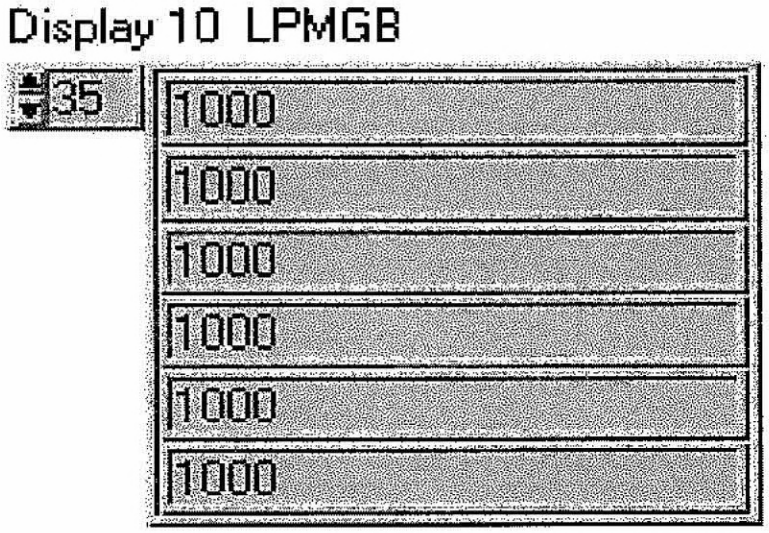

Figura 20 - Amostra do LPMGB

Palavra digital de 8 bits que representa o canal monitor de ganho dos amplificadores do experimento LP. Através dos gráficos anteriormente mostrados analisamos o desenvolvimento dos dados de testes. Rodando todo o programa e pegando os dados no começo, meio e fim conseguimos analisar o seu desenvolvimento.

4.2.4 DIAGRAMA HIERÁRQUICO

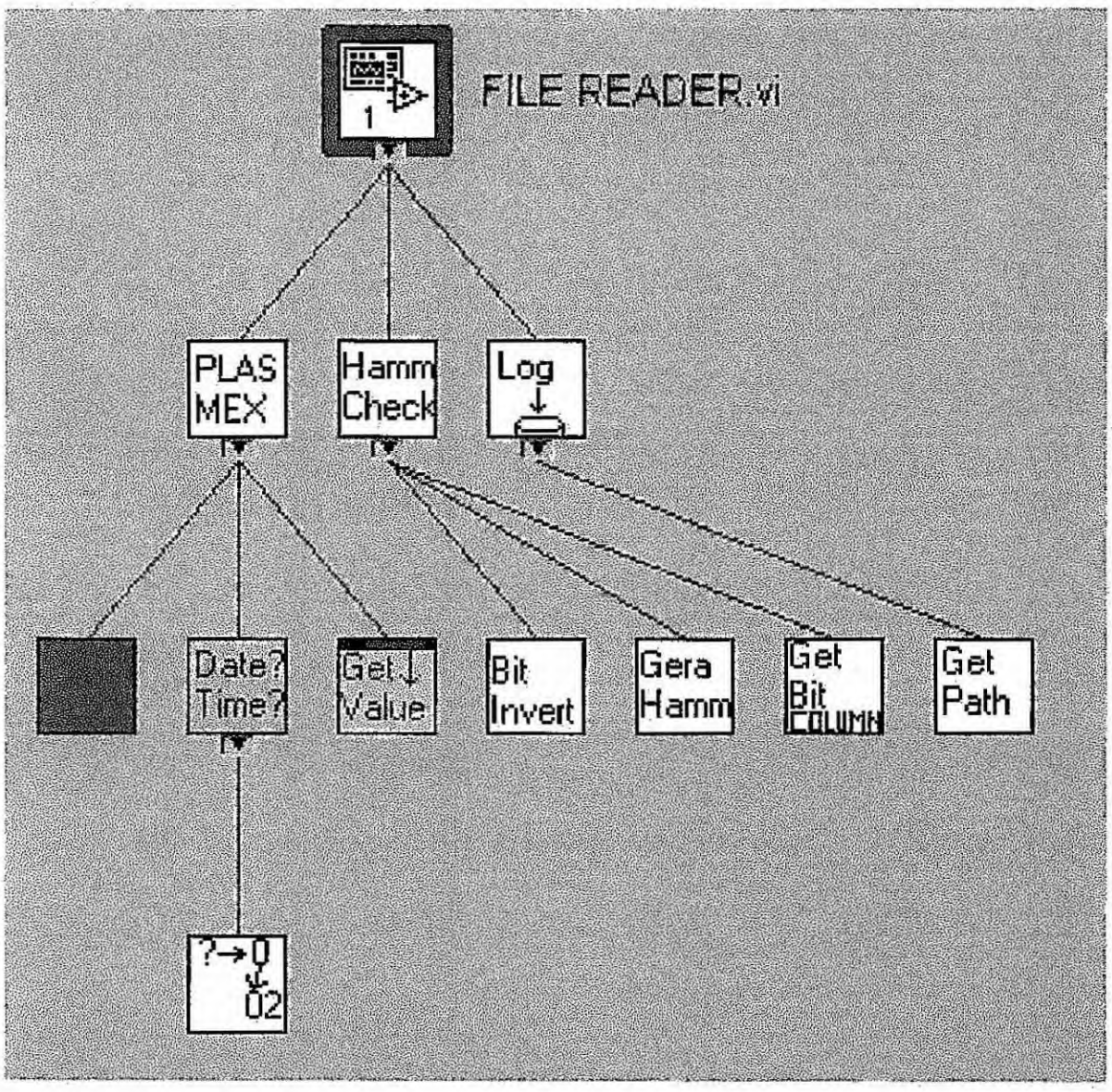

Figura 21 - diagrama hierárquico do programa FILE READER.vi

5. CURFIT

Teríamos que desenvolver um software para a análise de dados obtidos dos experimentos lançados a bordo de foguetes com o objetivo de obter informações sobre a distribuição espectral das irregularidades de plasma nas diferentes regiões da ionosfera.

O software CURFIT pode ser utilizado pois serve para fazer análise espectral baseado no método de entropia máxima. É da linguagem VISUAL BASIC e desenvolvido no INPE (Dr. Delano,1.997). Este software está sendo adaptado para a análise de dados atuais. Talvez haja a necessidade de fazer implementações, pois este programa está sendo estudado, além de sua linguagem. Por enquanto podemos observar que quando executado, o programa abre uma interface que pede um arquivo que tenha dados para serem processados. Ao mandar plotar o gráfico com os dados de entrada, este aparece sem a análise espectral. Ao cancelar, o programa aparece , na sua interface, com uma nova opção para que aí seja feita a análise espectral pela entropia máxima (figura 22). Continuando, o programa plota um novo gráfico com a análise espectral feita.

Para testar este programa fizemos um programa em C++. Este que gera um arquivo .dat com dados artificiais pela função:

$$S(t) = A_0 + A_1 * \text{sen}(2\pi f_1 t) + A_2 * \text{sen}(2\pi f_2 t) + A_3 * \text{sen}(2\pi f_3 t) + A_4 * \text{sen}(2\pi f_4 t) + A_5 * \text{sen}(2\pi f_5 t)$$

Para valores de $t = t_1, t_2, t_3, t_4, t_5, \ldots, t_n = I * \Delta t$, onde $I = 1,n$ (um inteiro), Δt é o intervalo de tempo entre duas amostras e $f_1, f_2, \ldots f_5$, são frequências arbitrariamente escolhidas. Os dados artificiais declarados nesse programa deverão estar diretamente ligados com os picos do gráfico gerado no programa que estamos testando.

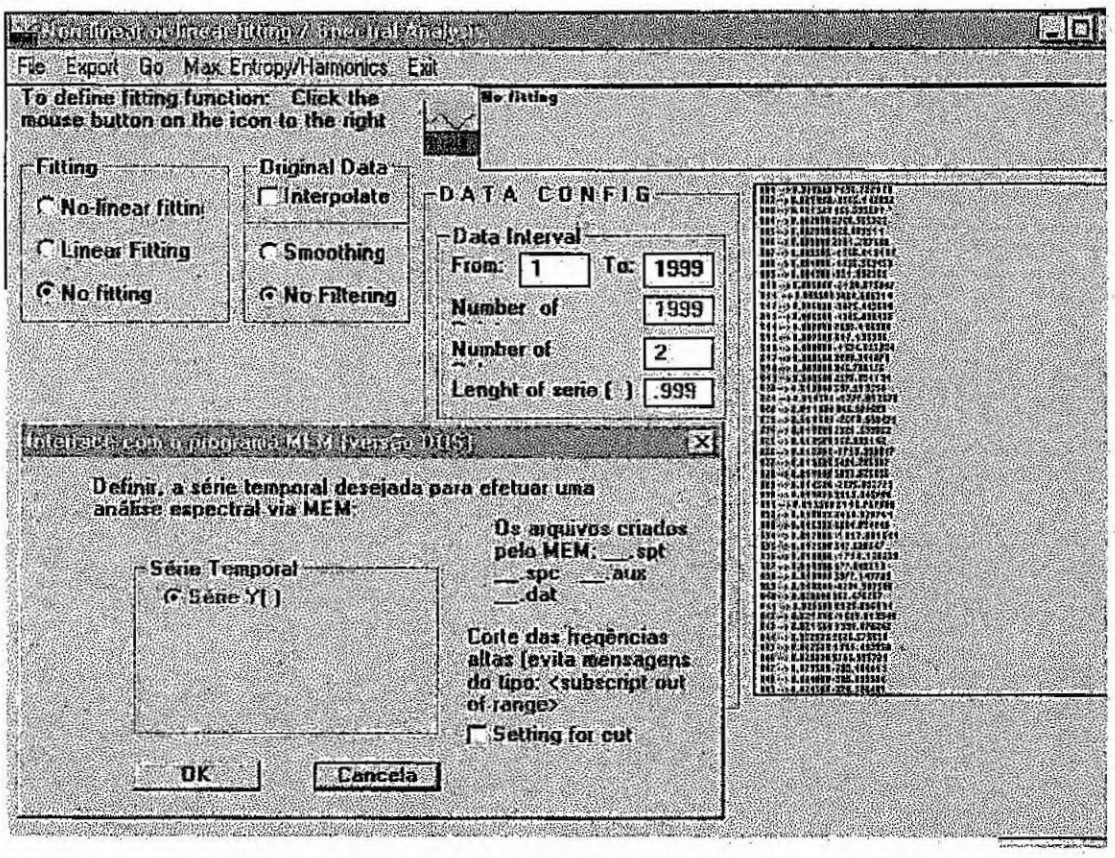

Figura 22 - Interface do CURFIT com a análise espectral pela entropia máxima sendo pedida.

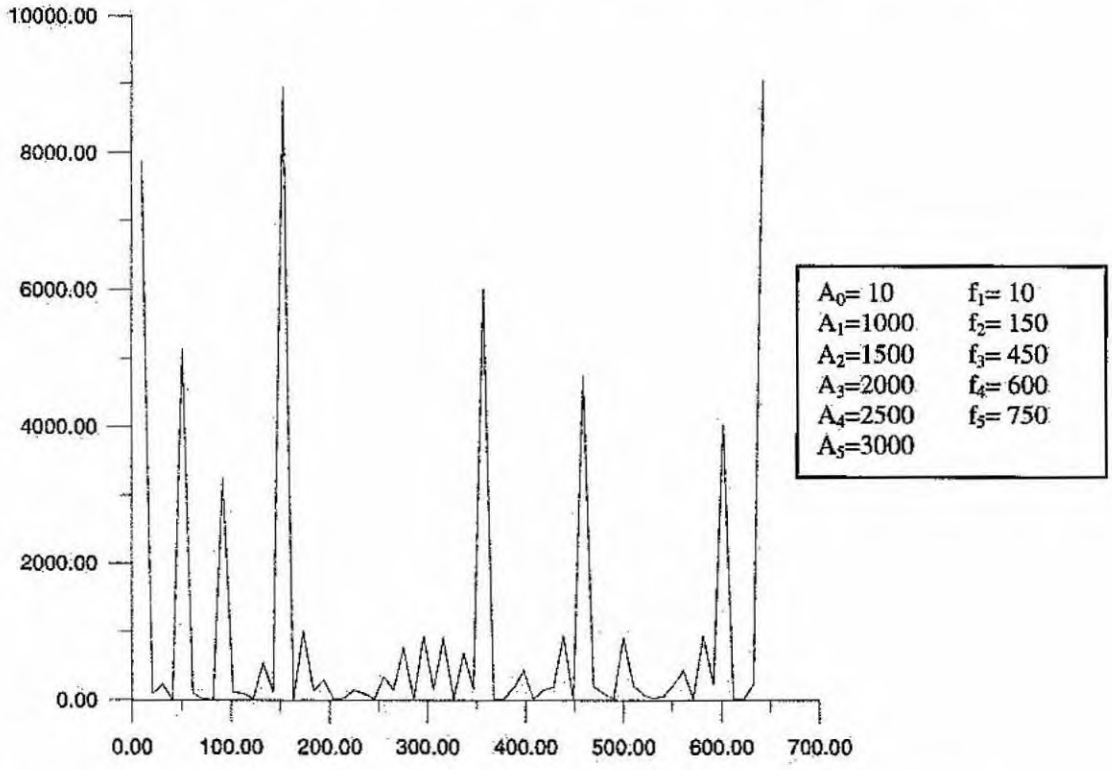

Figura 23 - Gráfico gerado pelo CURFIT após análise espectral pela entropia máxima

6. Referência bibliográfica

- ABDU, M.A.; Uma Metodologia para Obtenção do Conteúdo Eletrônico Total (TEC) através do GPS.
- SOUZA, J.R.; Estudo do conteúdo total ionosférico. Dissertação de mestrado em ciência espacial/ geofísica espacial, 1992.
- BITTENCOURT, J.A.; Física da ionosfera, relatório n. INPE- 1191- nte/111, 1978.
- VIEIRA, LEANDRO P.; Projeto e desenvolvimento dos experimentos PLASMEX e pdp para medição de parâmetros de plasma espacial.
- Dr. Delano (INPE / LASER) desenvolvedor do programa CURFIT,1.997
- ERA, J.C.P.; FRANCISCO, M.F.M.; MURALIKRISHNA, Polinaya; Interface Specification of plasma probe Payload.
- LabVIEW - Function Reference Manual- National instruments corporation, december 1993.
- LabVIEW for Windows- Tutorial- National instruments corporation, august 1993.
- LabVIEW for Windows- User Manual- National instruments corporation, december 1993.

```
if( x > 2 )
then    x = 2;
if  x < 1
        x = 1;
```

Ответ: в Си нет ключевого слова then, условия в операторах if, while должны браться в ()-скобки.

1.6. Напишите программу, печатающую ваше имя, место работы и адрес. В первом варианте программы используйте библиотечную функцию printf, а во втором - puts.

1.7. Составьте программу с использованием следующих постфиксных и префиксных операций:

```
a = b = 5
a + b
a++ + b
++a + b
--a + b
a-- + b
```

Распечатайте полученные значения и проанализируйте результат.

1.8.
Цикл for

```
for(INIT; CONDITION; INCR)
        BODY
```

```
        INIT;
repeat:
        if(CONDITION){
                BODY;
        cont:
                INCR;
                goto repeat;
        }
out:    ;
```

Цикл while

```
while(COND)
        BODY
```

```
cont:
repeat:
        if(CONDITION){
                BODY;
                goto repeat;
        }
out:    ;
```

Цикл do

```
do
        BODY
while(CONDITION)
```

```
cont:
repeat:
        BODY;
        if(CONDITION) goto repeat;
out:    ;
```

В операторах цикла внутри тела цикла BODY могут присутствовать операторы break и continue; которые означают на наших схемах следующее:

```
#define break     goto out
#define continue  goto cont
```

1.9. Составьте программу печати прямоугольного треугольника из звездочек

```
*
**
***
****
*****
```

используя цикл for. Введите переменную, значением которой является размер катета треугольника.

1.10. Напишите операторы Си, которые выдают строку длины WIDTH, в которой сначала содержится x0 символов '-', затем w символов '*', и до конца строки - вновь символы '-'. Ответ:

```
int x;
for(x=0; x < x0;     ++x) putchar('-');
for(   ; x < x0 + w; x++) putchar('*');
for(   ; x < WIDTH ; ++x) putchar('-');
putchar('\n');
```

либо

```
for(x=0; x < WIDTH; x++)
        putchar( x < x0     ? '-' :
                 x < x0 + w ? '*' :
                              '-' );
putchar('\n');
```

1.11. Напишите программу с циклами, которая рисует треугольник:

```
   *
  ***
 *****
*******
*********
```

Ответ:

```
/* Треугольник из звездочек */
#include <stdio.h>

/* Печать n символов c */
printn(c, n){
        while( --n >= 0 )
                putchar(c);
}

int lines = 10;         /* число строк треугольника */
void main(argc, argv) char *argv[];
{
        register int nline;  /* номер строки */
        register int naster; /* количество звездочек в строке */
        register int i;

        if( argc > 1 )
                lines = atoi( argv[1] );

        for( nline=0; nline < lines ; nline++ ){
                naster = 1 + 2 * nline;

                /* лидирующие пробелы */
                printn(' ', lines-1  - nline);

                /* звездочки */
                printn('*', naster);

                /* перевод строки */
                putchar( '\n' );
        }
        exit(0);        /* завершение программы */
}
```

1.12. В чем состоит ошибка?

```
main(){  /* печать фразы 10 раз */
   int i;
   while(i < 10){
       printf("%d-ый раз\n", i+1);
       i++;
   }
}
```

Ответ: автоматическая переменная i не была проинициализирована и содержит не 0, а какое-то произвольное значение. Цикл может выполниться не 10, а любое число раз (в том числе и 0 по случайности). Не забывайте инициализировать переменные, возьмите описание с инициализацией за правило!

```
   int i = 0;
```

Если бы переменная i была статической, она бы имела начальное значение 0.

В данном примере было бы еще лучше использовать цикл for, в котором все операции над индексом цикла собраны в одном месте - в заголовке цикла:

```
for(i=0; i < 10; i++) printf(...);
```

1.13. Вспомогательные переменные, не несущие смысловой нагрузки (вроде счетчика повторений цикла, не используемого в самом теле цикла) принято по традиции обозначать однобуквенными именами, вроде i, j. Более того, возможны даже такие курьезы:

```
main(){
  int _ ;
  for( _ = 0; _ < 10; _++) printf("%d\n", _ );
}
```

основанные на том, что подчерк в идентификаторах - полноправная буква.

1.14. Найдите 2 ошибки в программе:

```
main(){
        int x = 12;

        printf( "x=%d\n" );
        int y;
        y = 2 * x;
        printf( "y=%d\n", y );
}
```

Комментарий: в теле функции все описания должны идти перед всеми выполняемыми операторами (кроме операторов, входящих в состав описаний с инициализацией). Очень часто после внесения правок в программу некоторые описания оказываются после выполняемых операторов. Именно поэтому рекомендуется отделять строки описания переменных от выполняемых операторов пустыми строками (в этой книге это часто не делается для экономии места).

1.15. Найдите ошибку:

```
int n;
n = 12;
main(){
        int y;
        y = n+2;
        printf( "%d\n", y );
}
```

Ответ: выполняемый оператор n=12 находится вне тела какой-либо функции. Следует внести его в main() после описания переменной y, либо переписать объявление перед main() в виде

```
int n = 12;
```

В последнем случае присваивание переменной n значения 12 выполнит компилятор еще во время компиляции программы, а не сама программа при своем запуске. Точно так же происходит со всеми статическими данными (описанными как static, либо расположенными вне всех функций); причем если их начальное значение не указано явно - то подразумевается 0 ('\0', NULL, ""). Однако нулевые значения не хранятся в скомпилированном выполняемом файле, а требуемая "чистая" память расписывается при старте программы.

1.16. По поводу описания переменной с инициализацией:

```
TYPE x = выражение;
```

является (почти) эквивалентом для

```
TYPE x;          /* описание */
x = выражение;   /* вычисление начального значения */
```

Рассмотрим пример:

```
#include <stdio.h>
extern double sqrt();   /* квадратный корень */
double x   = 1.17;
double s12 = sqrt(12.0);            /* #1 */
double y   = x * 2.0;               /* #2 */
FILE  *fp  = fopen("out.out", "w"); /* #3 */
main(){
  double ss = sqrt(25.0) + x;       /* #4 */
  ...
}
```

Строки с метками #1, #2 и #3 ошибочны. Почему?

Ответ: при инициализации статических данных (а s12, y и fp таковыми и являются, так как описаны вне какой-либо функции) выражение должно содержать только константы, поскольку оно вычисляется КОМПИЛЯТОРОМ. Поэтому ни использование значений переменных, ни вызовы функций здесь недопустимы (но можно брать адреса от переменных).

В строке #4 мы инициализируем автоматическую переменную ss, т.е. она отводится уже во время выполнения программы. Поэтому выражение для инициализации вычисляется уже не компилятором, а самой программой, что дает нам право использовать переменные, вызовы функций и.т.п., то есть выражения языка Си без ограничений.

1.17. Напишите программу, реализующую эхо-печать вводимых символов. Программа должна завершать работу при получении признака EOF. В UNIX при вводе с клавиатуры признак EOF обычно обозначается одновременным нажатием клавиш CTRL и D (CTRL чуть раньше), что в дальнейшем будет обозначаться CTRL/D; а в MS DOS - клавиш CTRL/Z. Используйте getchar() для ввода буквы и putchar() для вывода.

1.18. Напишите программу, подсчитывающую число символов поступающих со стандартного ввода. Какие достоинства и недостатки у следующей реализации:

```
#include <stdio.h>
main(){ double cnt = 0.0;
        while (getchar() != EOF) ++cnt;
        printf("%.0f\n", cnt );
}
```

Ответ: и достоинство и недостаток в том, что счетчик имеет тип double. Достоинство - можно подсчитать очень большое число символов; недостаток - операции с double обычно выполняются гораздо медленнее, чем с int и long (до десяти раз), программа будет работать дольше. В повседневных задачах вам вряд ли понадобится иметь счетчик, отличный от long cnt; (печатать его надо по формату "%ld").

1.19. Составьте программу перекодировки вводимых символов со стандартного ввода по следующему правилу:

```
        a -> b
        b -> c
        c -> d
        ...
        z -> a
другой символ -> *
```

Коды строчных латинских букв расположены подряд по возрастанию.

1.20. Составьте программу перекодировки вводимых символов со стандартного ввода по следующему правилу:

```
            B -> A
            C -> B
            ...
            Z -> Y
      другой символ -> *
```

Коды прописных латинских букв также расположены по возрастанию.

1.21. Напишите программу, печатающую номер и код введенного символа в восьмеричном и шестнадцатеричном виде. Заметьте, что если вы наберете на вводе строку символов и нажмете клавишу ENTER, то программа напечатает вам на один символ больше, чем вы набрали. Дело в том, что код клавиши ENTER, завершившей ввод строки - символ '\n' - тоже попадает в вашу программу (на экране он отображается как перевод курсора в начало следующей строки!).

1.22. Разберитесь, в чем состоит разница между символами '0' (цифра нуль) и '\0' (нулевой байт). Напечатайте

```
      printf( "%d %d %c\n", '\0', '0', '0' );
```

Поставьте опыт: что печатает программа?

```
    main(){
            int c = 060;  /* код символа '0' */
            printf( "%c %d %o\n", c, c, c);
    }
```

Почему печатается 0 48 60? Теперь напишите вместо
```
  int c = 060;
```
строчку
```
  char c = '0';
```

1.23. Что напечатает программа?

```
    #include <stdio.h>
    void main(){
            printf("ab\0cd\nxyz");
            putchar('\n');
    }
```

Запомните, что '\0' служит признаком конца строки в памяти, а '\n' - в файле. Что в строке "abcd\n" на конце неявно уже расположен нулевой байт:

```
    'a','b','c','d','\n','\0'
```

Что строка "ab\0cd\nxyz" - это

```
    'a','b','\0','c','d','\n','x','y',z','\0'
```

Что строка "abcd\0" - избыточна, поскольку будет иметь на конце два нулевых байта (что не вредно, но зачем?). Что printf печатает строку до нулевого байта, а не до закрывающей кавычки.
Программа эта напечатает ab и перевод строки.
Вопрос: чему равен sizeof("ab\0cd\nxyz")? Ответ: 10.

1.24. Напишите программу, печатающую целые числа от 0 до 100.

1.25. Напишите программу, печатающую квадраты и кубы целых чисел.

1.26. Напишите программу, печатающую сумму квадратов первых n целых чисел.

1.27. Напишите программу, которая переводит секунды в дни, часы, минуты и секунды.

1.28. Напишите программу, переводящую скорость из километров в час в метры в секундах.

1.29. Напишите программу, шифрующую текст файла путем замены значения символа (например, значение символа C заменяется на C+1 или на ^C).

1.30. Напишите программу, которая при введении с клавиатуры буквы печатает на терминале ключевое слово, начинающееся с данной буквы. Например, при введении буквы 'b' печатает "break".

1.31. Напишите программу, отгадывающую задуманное вами число в пределах от 1 до 200, пользуясь подсказкой с клавиатуры "=" (равно), "<" (меньше) и ">" (больше). Для угадывания числа используйте метод деления пополам.

1.32. Напишите программу, печатающую степени двойки

1, 2, 4, 8, ...

Заметьте, что, начиная с некоторого n, результат становится отрицательным из-за переполнения целого.

1.33. Напишите подпрограмму вычисления квадратного корня с использованием метода касательных (Ньютона):

$$x(0) = a$$

$$x(n+1) = \frac{1}{2} * \left(\frac{a}{x(n)} + x(n) \right)$$

Итерировать, пока не будет $| x(n+1) - x(n) | < 0.001$

Внимание! В данной задаче массив не нужен. Достаточно хранить текущее и предыдущее значения x и обновлять их после каждой итерации.

1.34. Напишите программу, распечатывающую простые числа до 1000.

1, 2, 3, 5, 7, 11, 13, 17, ...