此 SEG2 数据的读取格式如下：

文件描述块：

第 1－2 字节为一个十六进制的正数 553A，它表示该文件的存贮方式是高字节在前，低字节在后。

第 6－8 字节为无符号整型量，表示道数（此数据为 201 道）；

第 33 个字节开始为道指针块，它存放的是一系列的道指针数据，每个道指针为 4 个字节，该每一个指针指向该道描述块的起始位置，它的长度为 4*道数（此数据道数为 201），这个必须读出，我们把它可以放在一个数组 Y1(I)里面，这个数据是每一道的起始地址（这个地址包括道描述块和数据块，故在读数据时要在此地址的基础上加上道描述块的大小）。接下来就是一系列的字符串它包括了数据采集的时间等信息，在每个字符串前是一个两个字节的整型量，它的数值表示后面字符串的长度（字节数），当这个整型量为 0 时表示字符串结束。

道描述块：。

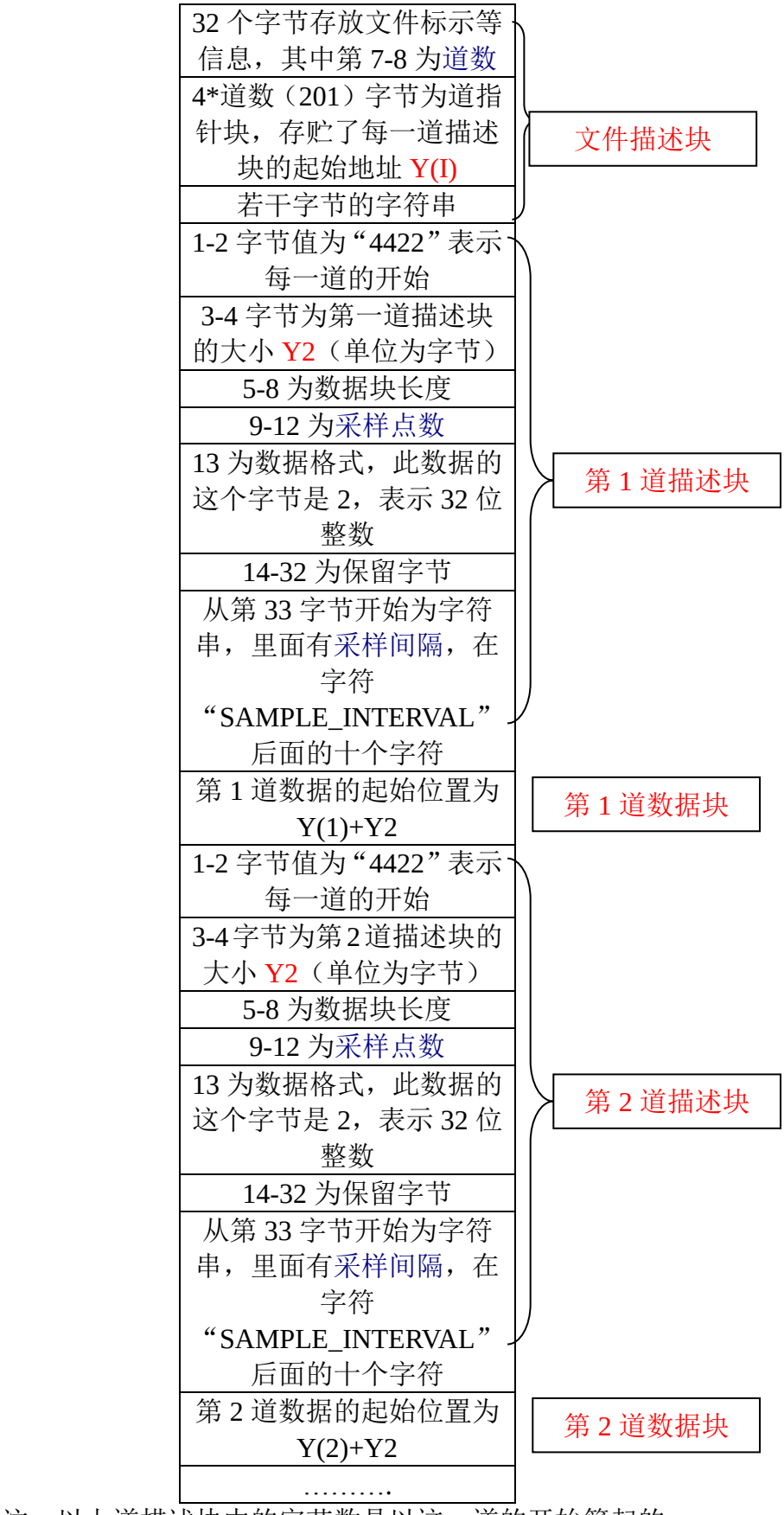
道描述块是以一个两个字节的无符号整型量开始，其值为 4422。如果把它作为每一道描述块的 1－2 字节，那么第 3－4 为道描述块的大小，这个数值要读出来。在后面读数时定位用，把这个值存在 Y2 里面。接下来 5－8 为数据块的长度，第 9－12 为采样点数（此数据为 1024），第 13 个为一无符号整型数，表示数据格式，此数据的这个数为 2，表示 32 为整数。第 14－32 为保留字节。从第 33 开始存贮一系列字符串，这个字符串的读法和前面文件块的读法一致。其中就有我们所需要的采样间隔。采样间隔数据前面有一“SAMPLE_INTERVAL”字符串。我们只有找到这个字符串，读其后面的十个字节的字符再转换为数字形式就可以了。

数据块：

每一道数据的起始地址为前面所读出的 Y1(I)＋Y2.接下来就开始读这一道的数据。

读完一道数据后接着从第二道的道描述块开始读下一道数据。

图示如下：



注：以上道描述块中的字节数是以这一道的开始算起的。

以下是程序：

```
f1 = fopen(fname,"rb");
        fread(&blockid,2,1,f1); //十六进制的正数 553A
        fread(&revnum,2,1,f1);//SEG-2 的版本号
        fread(&pointerbytecount,2,1,f1);//道指针子块的大小
        fread(&numtrace,2,1,f1);//道数
        fread(&stringtermcount,1,1,f1);
        fread(&stringterm1,1,1,f1);
        fread(&stringterm2,1,1,f1);
        fread(&linetermcount,1,1,f1);
        fread(&lineterm1,1,1,f1);
        fread(&lineterm2,1,1,f1);
        //
        fread(&reserved,1,18,f1);//以上共 32 个字节

        fread(tracepointers,4,numtrace,f1);//201*4=804 字节，存放道指针数据
         fread(&stringlength,2,1,f1);//开始读文件描述中的字符串，stringlength 为
                                                每一个字符串的长度。
        while(0 != stringlength)
        {
            fread(string1,1,stringlength-2,f1);//采集参数：数据采集日期和时间等
            keycheck();
            fread(&stringlength,2,1,f1);//?
        }
        for(j=0; j<numtrace; j++)//开始从第一道开始循环读描述块和数据块
        {
          fseek(f1,tracepointers[j],SEEK_SET);  // 每一道的开始（包括描述块和
                                                        数据块）
            fread(&blockid,2,1,f1);//道描述开始标示符 4422
            fread(&blockleng,2,1,f1);//3-4 字节为道的描述块长度
            fread(&datalength,4,1,f1);//5-8 字节为数据块的长度
            fread(&numsamples,4,1,f1);//采样点数 9-12
/*
    1 = 16 bit fixed. (INT)
    2 = 32 bit fixed. (LONG)
    3 = 20 bit packed floating.
    4 = 32 bit ieee floating point.(FLOAT)
    5 = 64 bit ieee floating point.(DOUBLE)
*/
            fread(&datatype,1,1,f1);//无符号整型，用来指明该道数据的存储格式
（此数据为 2，即 32 位整数）

            fread(reserved,1,19,f1); 保留字节
            fread(&stringlength,2,1,f1); /开始读道描述中的字符串，stringlength 为
```

每一个字符串的长度。

```
while(0 != stringlength)
{
    fread(string1,1,stringlength-2,f1);//采集参数信息，其中包括采样率
    keycheck();
    fread(&stringlength,2,1,f1);
}
printf("\n");
fseek(f1,blockleng+tracepointers[j],SEEK_SET);  // 每一道数的开始，
其中 blockleng 为每一道的描述块的大小，tracepointers[j]为相应道的起始地址
switch(datatype)
{
case 1:
    {
        fread(iinbuf,2,(short)numsamples,f1);
        for(k=0; k<numsamples; k++)
            outbuf[k] = iinbuf[k];
        break;
    }
case 2:
    {
        fread(linbuf,4,(short)numsamples,f1);
        for(k=0; k<numsamples; k++)
            outbuf[k] = linbuf[k];
        fwrite(&outbuf[k],2,1,f3);
        break;
    }
case 3:
    {
        unsigned short totalbytes,subpointer;
        unsigned short expo;
        long longdat;
        totalbytes = (numsamples * 5)/2;
        fread(cinbuf,1,totalbytes,f1);
        for(k=0; k<(numsamples);    )
        {
            subpointer = (k / 4) * 5;
            expo = (unsigned short)iinbuf[subpointer++];
            for(i=0; i<4; i++)
            {
                if(0x8000 & iinbuf[subpointer])
                    longdat = 0xffff8000;
                else
                    longdat = 0;
```

```
                        longdat = longdat | ((long)iinbuf[subpointer++] <<
(0x000f & expo));
                        expo >>= 4;
                        outbuf[k++] = longdat;
                    }
                }
                break;
            }
        case 4:
            {
                fread(finbuf,4,(short)numsamples,f1);
                for(k=0; k<numsamples; k++)
                    outbuf[k] = finbuf[k];
                break;
            }
        case 5:
            {
                fread(dinbuf,8,(short)numsamples,f1);
                for(k=0; k<numsamples; k++)
                    outbuf[k] = dinbuf[k];
                break;
            }
        } // end switch
```