The ISHNE Holter Standard Output File Format

Fabio Badilini, Ph.D., for the ISHNE Standard Output Format Task Force

From the Hôpital Lariboisière, Paris, France

Introduction

Standard Output Format for Digital Holter Data is a single file structurally organized in a header followed by a (larger) data block containing all stored electrocardiographic (ECG) digital samples. The format of this file is the outcome of several meetings with different manufacturers and it aims to maximize the ratio between simplicity and flexibility.

A freeware viewer of files in Standard Output Format will be made available by the end of Summer 1998 for those who may be interested.

CONVENTIONS ON DATA TYPES

In this document, the following data type conventions will be adopted:

short int → 2 bytes
long int → 4 bytes
unsigned int → 2 bytes
char → 1 byte

All string are null-terminated (last character is the ASCII value 0).

MAGIC NUMBER

A "magic number," which will consist of a predefined string of characters, will be inserted at the very beginning of the file. This will allow for a quick verification that the file referenced is indeed in ISHNE format. The magic number will consist of the string of the eight characters ISHNE1.0.

CHECKSUM

The two bytes following the magic number will be a CRC-CCITT checksum calculated over the complete header (fixed-length and variable-length blocks, see next Section). The listing of the algorithm for calculation of this checksum is provided in the Appendix B.

HEADER

The header will start at the 11th byte of the file, i.e., immediately following the checksum. It will consist of a fixed-length block (512 bytes) and a variable-length block reserved for freestyle general comments. The fixed-length block will come first and one of its fields will indicate the size and offset of the variable-length block. The variable-length block will consist simply of a stream of ASCII (extended set of 256 characters) characters that any user or manufacturer will use according to his needs.

The main goal of the header is to provide all necessary information on the associated ECG file. Beat annotations are not considered (they may be the target of a future task force). Consequently, issues such as paced versus nonpaced beats, exact localization of noise, or lead-intermittent regions are not involved. Nonetheless, the information on whether or not a pacemaker is present or whether or not a lead is noisy will be given.

HEADER SIZE

The global size of the header will depend upon the size of the variable-length block (General comment). Sizes of this block will be specified at the beginning of the fixed-length (512 bytes) block.

ECG Block Size

The total size of the ECG will depend on the number of leads and on the sampling rate (see description in ECG raw data paragraph). Size (in number of samples) and offset (in bytes) of this block

Address for reprints: Fabio Badilini, Ph.D., Hôpital Lariboisière, Service du Pr. Coumel, 2 rue Ambroise Paré, 75010 Paris, France. Fax: 33-1-49-95-84-39; E-mail: badilini@aol.com

will be specified in the header block. The actual size in bytes will be twice the size in samples multiplied by number of leads (one ECG sample will be stored in two bytes).

Lead Specification

According to several discussions, a maximum of 12 leads will be stored. The specification of these leads will consist of a sequence of int values according to the Table 1.

Table 1.

Lead	Code
Unknown	0
Generic bipolar	1
X bipolar	2
Y bipolar	3
Z bipolar	4
I to a VF (6 standard limb leads)	5 to 10
V1 to V6 (precordial leads)	11 to 16
ES	17
AS	18
AI	19

More lead information will eventually be put into a General comment field. In the fixed-length block, all 12 leads will have to be specified in the same order they appear in the ECG block and will be set to −9 for the nonpresent leads. For instance, the specification sequence in a file containing X Y Z lead will be:

2 3 4 −9 −9 −9 −9 −9 −9 −9 −9 −9.

Lead Quality

A simple lead-quality score will be incorporated for EACH lead stored (Table 2).

Table 2.

0	Unknown (unrated)
1	Good quality permanently
2	Intermittent noise < 10% of total length
3	Frequent noise (> 10%)
4	Lead disconnection (< 10%)
5	Lead disconnection (> 10%)

As it was with lead specification, lead quality score will have to be specified in the order they appear in the ECG, and −9 will be the setting for the nonpresent leads. For instance, the quality score sequence in a file containing permanent good quality X Y Z leads will be:

1 1 1 −9 −9 −9 −9 −9 −9 −9 −9 −9.

Amplitude Resolution

As pointed out by many participants, amplitude resolution has to be specified for EACH lead stored. Then, the relative field in the header will have to be an array of 12 values. As for the previous fields, resolution values will appear in the same order of ECG block, with −9 for nonpresent leads. A possible array in the scenario of three recorded leads nonequally calibrated may be:

5000 1000 2500 −9 −9 −9 −9 −9 −9 −9 −9 −9 −9

Dynamic Range

The dynamic range of each stored lead will not have to be specified, as it will be easily calculated by the amplitude resolution value. For instance, if one lead has a resolution of 1000 nV (1μV), the dynamic range will be from −32,768 to 32,767 mV. Of course, since the interval of digitized value is fixed, the dynamic range will be larger for systems with less resolution (changes of one unit will correspond to larger changes in mV).

Pacemakers

A specific field (int) will indicate the presence or absence of a pacemaker. This will be set to 0 (no pacemaker), 1 (pacemaker type not known), 2 (single chamber unipolar), 3 (dual chamber unipolar), 4 (single chamber bipolar), or 5 (dual chamber bipolar).

Fixed Block Description

Fixed block will be the sequence of data fields shown in Table 3.

ECG DATA

The International Society for Holter and Noninvasive Electrocardiology will only promote exchange of continuous ECG data. Thus, intermittent recordings will not be concerned by this format. In addition, following several discussions it has been confirmed that the issue of space is not relevant in light of continuous improvements of data storage capabilities. Thus, the need for data encapsulation or data compression has been definitely ruled out.

In order to obtain a good trade-off between simplicity and flexibility the format of the raw data (i.e., the way a single ECG sample of a specific

Table 3.

Description	Data Type	No. of Bytes
Size (in bytes) of variable length block	long int	4
Size (in samples) of ECG	long int	4
Offset of variable length block (in bytes from beginning of file, i.e., 8 + 2 + 512 = 522)	long int	4
Offset of ECG block (in bytes from beginning of file)	long int	4
Version of the file	short int	2
Subject First Name	char[40]	40
Subject Last Name	char[40]	40
Subject ID	char[20]	20
Subject Sex (0: unknown, 1: male, 2: female)	short int	2
Race (0: unknown, 1 Caucasian, 2 Black, 3 Oriental, 4–9 Reserved)	short int	2
Date of Birth (European: day, month, year)	3 short int	6
Date of recording (European)	3 short int	6
Date of creation of Output file (European)	3 short int	6
Start time (European: hour [0–23], min, sec)	3 short int	6
Number of stored leads	short int	2
Lead specification (see included Table)	12 short int	24
Lead quality (see included Table)	12 short int	24
Amplitude resolution in integer no. of nV	12 short int	24
Pacemaker code (see text for description)	short int	2
Type of recorder (either analog or digital)	char[40]	40
Sampling rate (in hertz)	short int	2
Proprietary of ECG (if any)	char[80]	80
Copyright and restriction of diffusion (if any)	char[80]	80
Reserved	char[88]	88

lead is stored) is fixed in a way "convenient" to all manufacturers. By "convenient" we intend a format allowing the lossless encoding of all present (and possibly future) Holter digital ECGs.

Format of One ECG Sample

The storage size of one ECG sample has been fixed to two bytes. Data will be stored in the signed format with digital 0 matching 0 mV; most significant bit is "dedicated" to the sign and the range of stored values covers the interval from −32,768 to +32767. Negative values will be stored in a two-complement way. All two-byte samples will be stored in little-endian form (LSB first).

	ch1, 1st sample	2 bytes
	ch2, 1st sample	
	ch3, 1st sample	
	.....	
	chn, 1st sample	
	ch1, 2nd sample	
	ch2, 2nd sample	
	ch3, 2nd sample	
	.....	
	chn, 2nd sample	
	ch1, 3rd sample	
	.....	
	.....	

Storing Order

The ECG block will follow the header block with samples stored multiplexed, as shown in Figure 1, where the number n of leads actually stored (and their identification) will be specified in the header block.

Use of a Specific Sample Value to Indicate Lead Fault

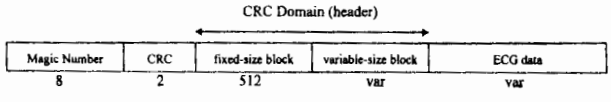
Some members of the group has proposed the use of a reserved sample value (e.g., −32,768 or 0×8,000) to indicate lead fault. This would permit us to overcome the drawbacks of AHA and MIT databases.

Size of Standard Output File

For a fixed time length, this will depend on sampling frequency and number of channels stored. For example (neglecting the header block), a 24-hour ECG with three leads will be ~66 MB at 128 Hz, ~103 MB at 200 Hz, and ~206 MB at 400 Hz. A 24-hour ECG with two leads will be ~44 MB at 128 Hz, ~69 MB at 200 Hz, and ~138 MB at 400 Hz. In some instances, these files may be "redundant," i.e., the same information could have been stored in a smaller space (think of those who store samples in 8 or 10 bits encapsulated). This is somewhat the price paid to be able to store all current formats without any loss of information.

CONCLUSION

The following schema (Fig. 2) summarizes the sequential structure of the Standard Output File in



its three blocks of data preceded by a magic number and a checksum calculated over the two blocks of the header. At the bottom of each block, the size in bytes is indicated (var = variable).

POTENTIAL FUTURE MODIFICATIONS

Standard Output File Format described in this document only concerns raw ECG and is intended to facilitate data exchange and research in the field of Holter. The ambition of this effort is that this format will increase the use and the importance of long-term ambulatory recording. Possible future goals will the either modification/improvements of current format or the development of a second standard file covering annotation lists associated with ECG.

APPENDIX A: FIXED-LENGTH BLOCK EXAMPLE IN C LANGUAGE

Of course, the Standard Output file could be obtained with different computer languages and everyone will be free to decide how to organize the work in his/her context. However, we will here provide an example of a C structure to read/write the fixed-length (512 bytes) header block.

```
struct {
    long        Var_length_block_size;
    long        Sample_Size_ECG;
    long        offset_var_length_block;
    long        Offset_ECG_block;
    short int   file_Version;
    char        First_Name[40];
    char        Last_name[40];
    char        ID[20];
    short int   Sex;
    short int   Race;
    short int   Birth_Date[3];
    short int   Record_Date[3];
    short int   File_Date[3];
    short int   Start_Time[3];
    short int   nLeads;
    short int   Lead_Spec[12];
    short int   Lead_Qual[12];
    short int   Resolution[12];
    short int   Pacemaker;
    char        Recorder[40];
    short int   Sampling_Rate;
    char        Proprietary[80];
    char        Copyright[80];
    char        Reserved[88];
}
```

APPENDIX B: CHECKSUM

We will provide here a description of a CRC-CCITT checksum. This appendix has been directly extracted from the official document entitled "A standard communications protocol for computerized electrocardiography" (SCP-ECG) available in the office of the European Commitee for Standardization.

The two-byte checksum is calculated over the complete header block, starting with the first byte following the cheksum and ending with the last byte before start of ECG data block (i.e., it starts at 11th byte and stops at ECGoffset−1).

The CRC is based on CRC-CCITT (X^16 + X^12 + X^5 + 1). The CRC is a 16-bit quantity and should be preset to all 1s ($FFFF) at the start of the calculation for each block of data. Note: all operations are on bytes.

```
A = new byte
B = temp byte
CRCHI = High Byte (most significant) of the 16-bit CRC
CRCLO = Low Byte (least significant) of the 16-bit CRC

START:

  FOR A = FIRST_BYTE TO LAST_BYTE IN BLOCK DO:
    A = A XOR CRCHI
    CRCHI = A
    SHIFT A RIGHT FOUR TIMES   {ZERO FILL}
    A = A XOR CRCHI            {I J K L M N O P}
    CRCHI = CRCLO              {swap CRCHI, CRCLO}
    CRCLO = A
    ROTATE A LEFT 4 TIMES      {M N O P I J K L}
    B = A                      {temp save}
    ROTATE A LEFT ONCE         {N O P I J K L M}
    A = A AND $IF              {0 0 0 I J K L M}
    CRCHI = A XOR CRCHI
    A = B AND $F0              {M N O P 0 0 0 0}
    CRCHI = A XOR CRCHI        {CRCHI complete}
    ROTATE B LEFT ONCE         {N O P 0 0 0 0 M}
    B = B AND $E0              {N O P 0 0 0 0 0}
    CRCLO = B XOR CRCLO        {CRCLO complete}
  DOEND;
FINISH.
```

The final check on CRC is accomplished by adding or concatenating CRCHI and CRCLO at the end of data stream.