micromed

SystemPlus EVOLUTION

TYPE “4” - EEG FILE STRUCTURE DESCRIPTION

Version: 1.04
Dated: 19/12/2016

micromed

Via Giotto 2 - 31021
Mogliano Veneto (TV) - Italy
tel. +39.041.5937000
fax. +39.041.5937011
e-mail micromed@micromed.eu
web: www.micromed.eu

WARNING

This document is strictly confidential and must not be distributed without permission. Contact Micromed to have permissions or any other information.

INTRODUCTION

This is a description of the Micromed EEG file of Type 4, that is files generated by SystemPlus EVOLUTION. Four older versions do exist:

- The first two are the Type 0 or 1, which refers to System 1 files.
- The third is the Type 2, which refers to System 2 files.
- The fourth is the Type 3, which refers to System98 files, first edition.

Any Micromed EEG file contains a data file structure that can be easily interpreted. Data file is broken down into two main sections: the ***header, which*** contains the patient and setup data and a variable length ***trailer*** that contains the digitised trace data. The sequence of this data can be interpreted with the header block. Files have the capability to store up to **256** individual channels of signal data, the number stored in each file may vary and this information may be found in the header. To find the file of the relative trace that you are examining, use the "Properties" function provided by the System 98 software.

HEADER OF TYPE “4” FILES - SUMMARY

Offset	Label	Data Type	Description	Values
0 = 00h	**Title**	char[32]	Micromed Descriptive String	
32 = 20h	**Laboratory**	char[32]	Laboratory Descriptive Strings	‘max.31 characters’,0x00
64 = 40h	**Patient_Data**	Micromed_New_Patient_Data	Patient Data	
128 = 80h	**Date**	Micromed_New_Date_Type	Recording Date	[dd,mm,yy] 1998 = 98
131 = 83h	**Time**	Micromed_New_Time_Type	Recording Time	[hh,mm,ss]
134 = 86h	**Acquisition_Unit**	unsigned short int	Acquisition Equipment	See constant values
136 = 88h	**Filetype**	unsigned short int	Type of File	See constant values
138 = 8ah	**Data_Start_Offset**	unsigned long int	Address of First Signal Data	
142 = 8eh	**Num_Chan**	unsigned short int	Number of Memorised Channels	1..128
144 = 90h	**Multiplexer**	unsigned short int	Distance Between Channel Data	
146 = 92h	**Rate_Min**	unsigned short int	Minimum Sampling Frequency	64,128,256,512,1024Hz...
148 = 94h	**Bytes**	unsigned short int	Number of Bytes for each sample	1 ⇒ 1 byte 2 ⇒ 2 bytes 4 ⇒ 4 bytes
150 = 96h	**Compression**	unsigned short int	Type of Data Compression	0 ⇒ No Compression
152 = 98h	**Montages**	unsigned short int	Number of Specific Montages	0..30
154 = 9ah	**Dvideo_Begin**	unsigned long int	Starting Sample of Digital Video	
159 = 9Eh	**MPEG_Delay**	unsigned short int	Number of frames per hour of de-synchronization in MPEG acq.	
160 = A0h	**Reserved_1**	unsigned char[15]		
175 = AFh	**Header_Type**	unsigned char	Header Type	0,1 ⇒ “System 1” File 2 ⇒ “System 2” File 3,4 ⇒ “System 98” File
176 = B0h	**Code_Area**	Micromed_New_Descriptor	Descriptor of storage CODE	‘ORDER’, Pos, Len
192 = C0h	**Electrode_Area**	Micromed_New_Descriptor	Descr. of ELECTRODE	‘LABCOD’, Pos, Len
208 = D0h	**Note_Area**	Micromed_New_Descriptor	Descr. of operator NOTE	‘NOTE’, Pos, Len
224 = E0h	**Flag_Area**	Micromed_New_Descriptor	Descr. of FLAG	‘FLAGS’, Pos, Len
240 = F0h	**Segment_Area**	Micromed_New_Descriptor	Descr. of REDUCTION	‘TRONCA’, Pos, Len
256 = 100h	**B_Impedance_Area**	Micromed_New_Descriptor	Descr. of BEGIN IMPED.	‘IMPED_B’, Pos, Len
272 = 110h	**E_Impedance_Area**	Micromed_New_Descriptor	Descr. of ENDING IMPED.	‘IMPED_E’, Pos, Len
288 = 120h	**Montage_Area**	Micromed_New_Descriptor	Descr. of SPECIFIC MONTAGES	‘MONTAGE’, Pos, Len
304 = 130h	**Compression_Area**	Micromed_New_Descriptor	Descr. of COMPRESSION	‘COMPRESS’, Pos, Len
320 = 140h	**Average_Area**	Micromed_New_Descriptor	Descr. Of AVERAGE	‘AVERAGE’, Pos, Len
336 = 150h	**History_Area**	Micromed_New_Descriptor	Descr. of HISTORY	‘HISTORY’, Pos, Len
352 = 160h	**Dvideo_Area**	Micromed_New_Descriptor	Descr. of DVIDEO	‘DVIDEO’, Pos, Len
368 = 170h	**EventA_Area**	Micromed_New_Descriptor	Descr. of EVENT A	‘EVENT A’, Pos, Len
384 = 180h	**EventB_Area**	Micromed_New_Descriptor	Descr. of EVENT B	‘EVENT B’, Pos, Len
400 = 190h	**Trigger_Area**	Micromed_New_Descriptor	Descr. of TRIGGER	‘TRIGGER’, Pos, Len
416 = 1A0h	**Brain_Img_Area**	Micromed_New_Descriptor	Descr. of BRAIN IMG	‘BRAINIMG’, Pos, Len
432 = 1B0h	**Reserved_2**	unsigned char [208]		
640 = 280h				
	Code	Micromed_New_Code [MAX_CAN]	List of the Recorded Channels	
	Electrode	Micromed_New_Electrode [MAX_LAB]	Description of Each Channel	
	Note	Micromed_New_Annotation [MAX_NOTE]	Operator Notes	
	Flag	Micromed_New_Marker_Pair [MAX_FLAG]	Pointers to Flags	
	Segment	Micromed_New_Segment [MAX_SEGM]	Reduction Description	
	B_Impedance	Micromed_New_Impedance [MAX_CAN]	Impedance at Beginning of Recor	
	E_Impedance	Micromed_New_Impedance [MAX_CAN]	Impedance at End of Recording	
	Montage	Micromed_New_Montage [MAX_MONT]	Specific Montages	
	Compression	Micromed_New_Compression	Data Compression	
	Average	Micromed_New_Average	Off-Line Average	
	History_Sample	Micromed_New_Sample [MAX_SAMPLE]	“As Recorded” Pointers	
	History	Micromed_New_Montage [MAX_HISTORY]	“As Recorded” Montages	
	Dvideo	Micromed_New_Dvideo [MAX_FILE]	Pointers to linked Digital Video	
	EventA	Micromed_New_Event	Event A Descriptor	
	EventB	Micromed_New_Event	Event B Descriptor	
	Trigger	Micromed_New_Trigger [MAX_TRIGGER]	Digital Triggers	
	Brain_Img	JPEG Image [MAX_LENGHT]	Brain Image	

Constant Definition:

MAX_CAN	***256***	***MAX_NOTE***	***200***	***MAX_SAMPLE***	***128***	***MAX_FILE***	***1024***
MAX_LAB	***640***	***MAX_FLAG***	***100***	***MAX_HISTORY***	***30***	***MAX_TRIGGER***	***8192***

MAX_MONT 30 MAX_SEGM 100 MAX_EVENT 100 MAX_LENGTH 200KB

MICROMED FILE HEADER TYPE 4

The following is the structure of the data file. The first column contains the effective address in the file.

0 TITLE Declaration: ***char Title [32];***

This descriptive string specifies that the file is a "MICROMED Brain-quick file". It's been introduced to let the user identify a Micromed Brain-Quick file simply typing at the DOS prompt "TYPE PAZxx.TRC" where xx is the number of the selected file. The number of possible ASCII characters is 30, with a 0x00,0x1A used as terminators of the string.

32 LABORATORY Declaration: ***char Laboratory [32];***

This descriptive string identifies the laboratory where the acquisition has been made. Its value obviously is specified by the user and is not known prior. The number of possible ASCII characters is 31, with a 0x00 used as terminator of the string.

64 PATIENT DATA Declaration: ***Micromed_New_Patiant_Data Patient_Data;***

This is data pertaining to patient identification. The structure of this record is the following:

Offset	Label	Data Type	Description
64	**Surname**	char[22]	Surname, without any terminator
86	**Name**	char[20]	First Name, without any terminator
106	**Month**	unsigned char	Month of birth
107	**Day**	unsigned char	Day of birth
108	**Year**	unsigned char	Year of birth, where 1900 =0, then the year 1998 =98
109	**Reserved**	unsigned char[19]	Internally used by the Brain-Quick software. They are not meaningful for the user and can't be used.

128 RECORDING DATE Declaration: ***Micromed_New_Date_Type Date;***

Stores the date of file creation, and therefore the recording date. The structure of this record is the following:

Offset	Label	Data Type	Description
0	**Day**	unsigned char	Day
1	**Month**	unsigned char	Month
2	**Year**	unsigned char	Year, where 1900 =0, then the year 1998 =98

Please note that this format support any year until 2155.

131 RECORDING TIME Declaration: ***Micromed_New_Time_Type Time;***

Stores the time of file creation, and therefore the recording time. The structure of this record is the following:

Offset	Label	Data Type	Description
0	**Hour**	unsigned char	Hours
1	**Min**	unsigned char	Minutes
2	**Sec**	unsigned char	Seconds

134 ACQUISITON EQUIPMENT Declaration: ***unsigned short int Acquisition_Unit;***

This byte specifies the source of the data with the following values:

Value	Description
0	BQ124 - 24 channels headbox, Internal Interface
1	MS40 - Holter recorder
2	BQ132S - 32 channels headbox, Internal Interface
6	BQ124 - 24 channels headbox, BQ CARD Interface
7	SAM32 - 32 channels headbox, BQ CARD Interface
8	SAM25 - 25 channels headbox, BQ CARD Interface
9	BQ132S R - 32 channels reverse headbox, Internal Interface
10	SAM32 R - 32 channels reverse headbox, BQ CARD Interface
11	SAM25 R - 25 channels reverse headbox, BQ CARD Interface
12	SAM32 - 32 channels headbox, Internal Interface
13	SAM25 - 25 channels headbox, Internal Interface
14	SAM32 R - 32 channels reverse headbox, Internal Interface
15	SAM25 R - 25 channels reverse headbox, Internal Interface
16	SD - 32 channels headbox with jackbox, SD CARD Interface – PCI Internal Interface
17	SD128 - 128 channels headbox, SD CARD Interface – PCI Internal Interface
18	SD96 - 96 channels headbox, SD CARD Interface – PCI Internal Interface
19	SD64 - 64 channels headbox, SD CARD Interface – PCI Internal Interface
20	SD128c - 128 channels headbox with jackbox, SD CARD Interface – PCI Internal Interface
21	SD64c - 64 channels headbox with jackbox, SD CARD Interface – PCI Internal Interface
22	BQ132S - 32 channels headbox, PCI Internal Interface
23	BQ132S R - 32 channels reverse headbox, PCI Internal Interface

136 FILE TYPE Declaration: ***unsigned short int Filetype;***

This variable indicates the type of data that has been stored.
C.R. = Common Reference, poly = polygraphy

Value	Label	Description
40	**C128**	C.R., 128 EEG (headbox SD128 only)
42	**C84P**	C.R., 84 EEG, 44 poly (headbox SD128 only)
44	**C84**	C.R., 84 EEG, 4 reference signals (named MKR,MKRB,MKRC,MKRD) (headbox SD128 only)
46	**C96**	C.R., 96 EEG (headbox SD128 – SD96 – BQ123S(r))
48	**C63P**	C.R., 63 EEG, 33 poly
50	**C63**	C.R., 63 EEG, 3 reference signals (named MKR,MKRB,MKRC)
52	**C64**	C.R., 64 EEG
54	**C42P**	C.R., 42 EEG, 22 poly
56	**C42**	C.R., 42 EEG, 2 reference signals (named MKR,MKRB)
58	**C32**	C.R., 32 EEG
60	**C21P**	C.R., 21 EEG, 11 poly
62	**C21**	C.R., 21 EEG, 1 reference signal (named MKR)
64	**C19P**	C.R., 19 EEG, variable poly
66	**C19**	C.R., 19 EEG, 1 reference signal (named MKR)
68	**C12**	C.R., 12 EEG
70	**C8P**	C.R., 8 EEG, variable poly
72	**C8**	C.R., 8 EEG
74	**CFRE**	C.R., variable EEG, variable poly
76	**C28P**	C.R., 28 EEG (21 standard, 7 poly transformed to EEG channels), 4 poly – headbox BQ132S(r) only
78	**C24P**	C.R., 24 EEG (21 standard, 3 poly transformed to EEG channels), 8 poly – headbox SAM32(r) only
80	**C25P**	C.R., 25 EEG (21 standard, 4 poly transformed to EEG channels), 7 poly – headbox SD with headbox JB 21P
82	**C27P**	C.R., 27 EEG (21 standard, 6 poly transformed to EEG channels), 5 poly – headbox SD with headbox JB 21P
100	**C26P**	C.R., 26 EEG, 6 poly (headbox SD, SD64c, SD128c with headbox JB Mini)
101	**C16P**	C.R., 16 EEG, 16 poly (headbox SD with headbox JB M12)
102	**C12P**	C.R., 12 EEG, 20 poly (headbox SD with headbox JB M12)

103	**32P**	32 poly (headbox SD, SD64c, SD128c with headbox JB Bip)
120	**C48P**	C.R., 48 EEG, 16 poly (headbox SD64)
121	**C56P**	C.R., 56 EEG, 8 poly (headbox SD64)
122	**C24P**	C.R., 24 EEG, 8 poly (headbox SD64)
140	**C52P**	C.R., 52 EEG, 12 poly (headbox SD64c, SD128c with 2 headboxes JB Mini)
141	**64P**	64 poly (headbox SD64c, SD128c with 2 headboxes JB Bip)
160	**C88P**	C.R., 88 EEG, 8 poly (headbox SD96)
161	**C80P**	C.R., 80 EEG, 16 poly (headbox SD96)
162	**C72P**	C.R., 72 EEG, 24 poly (headbox SD96)
180	**C120P**	C.R., 120 EEG, 8 poly (headbox SD128)
181	**C112P**	C.R., 112 EEG, 16 poly (headbox SD128)
182	**C104P**	C.R., 104 EEG, 24 poly (headbox SD128)
183	**C96P**	C.R., 96 EEG, 32 poly (headbox SD128)
200	**C122P**	C.R., 122 EEG, 6 poly (headbox SD128c with 4 headboxes JB Mini)
201	**C116P**	C.R., 116 EEG, 12 poly (headbox SD128c with 4 headboxes JB Mini)
202	**C110P**	C.R., 110 EEG, 18 poly (headbox SD128c with 4 headboxes JB Mini)
203	**C104P**	C.R., 104 EEG, 24 poly (headbox SD128c with 4 headboxes JB Mini)
204	**128P**	128 poly (headbox SD128c with 4 headboxes JB Bip)
205	**96P**	96 poly (headbox SD128c with 3 headboxes JB Bip)

138 ADDRESS OF SIGNAL DATA Declaration: ***unsigned long int Data_Start_Offset;***

The content of this variable is a pointer to the location of the first digitised signal data. This will indicate at which address in the file you will find the first data for channel 1. The next data in the string is the data for channel 2, the next data is the first data for channel 3 etc., etc. After the first data for all channels recorded have been stored (see **Num_Chan** to see how many channels have been stored and the description of each signal to see the correlation between the data and the particular electrode) then the next series of bytes will be the second data for each progressive channel and the sequence continues until the end of the file has been reached. The file can be image as structured in records which size is equivalent to the value of **Multiplexer** i.e. if recording 32 channels with two bytes per channel all at the same sample frequency, then the record size will be 64 bytes long.

142 NUMBER OF STORED CHANNELS Declaration: ***unsigned short int Num_Chan;***

This variable indicate how many channels of data are being stored, this value may vary from 1 to 128.

144 MULTIPLEXER Declaration: ***unsigned short int Multiplexer;***

This variable indicate the distance in bytes between a sample (or a block of samples) of a channel and the following sample (or block of samples) of the same channel.

146 MINIMUM SAMPLING RATE Declaration: ***unsigned short int Rate_Min;***

This variable indicates the minimum sampling frequency used to sample the channels stored. If same channels are sampled at a greater sampling frequency, in the description of each of these channels it will be found that a block of data is stored subsequently, and this number is 2 if the sampling frequency of the channel is 2 times the minimum, 4 if the sampling frequency of the channel is 4 times the minimum and so on. If all channels are sampled at the same frequency, the minimum sampling frequency will be the sampling frequency of each channel (and in the descriptor of each channel the length of data block will be 1).

148 NUMBER OF BYTES Declaration: ***unsigned short int Bytes;***

The contents of this location indicate the number of bytes used for each data in the file. This value is 1 to indicate 1 byte used (data converted with 8 bits or less), or 2 to indicate 2 bytes used (data converted with 16 bits or less). The number of bits used by each channel must be read in the description of each channel as it can be different one each other.

150 COMPRESSION Declaration: ***unsigned short int Compression;***

The contents of this location indicate if EEG data are compressed in the trailer or not. Value of 0 means no compression, value of 1 means compression, for which details can be found in the appropriate descriptor.

152 NUMBER OF MONTAGES STORED Declaration: ***unsigned short int Montages;***

The contents of this location indicate the presence and the exact number of particular montages stored for this trace. These montages are to be intended as a particular configuration of visualisation parameters, to be used only to display the EEG data in a specific way. These montages give no specification on the acquisition configuration, they are used only for the convenience of the physicians.

154 DIGITAL VIDEO START SAMPPLE Declaration: ***unsigned long int Dvideo_Begin;***

The contents of this location indicate the sample of the EEG data at which Digital Video recording has begun. This parameter, in combination with other contained in the Digital Video file is used to synchronise the DVEEG.

158 MPEG DIG. VIDEO SYNCHRONIZATION Declaration: *unsigned short int* ***MPEG_Delay***

160 .....reserved..... Declaration: ***unsigned char Reserved_1[15];***

This area will be used only for further development of Micromed Brain-Quick and cannot be used by external applications. Values that are found inside are not to be specified.

175 HEADER TYPE Declaration: ***unsigned char Header_Type;***

This byte represents the type of header. This parameter is used to distinguish this type of header from older Micromed header type.

value	description
0,1	Micromed “System 1” Header type
2	Micromed “System 2” Header Type
3	Micromed “System98” Header Type
4	Micromed “System98” Header Type

The first operation to do before reading all data is to verify if the header type is 4, otherwise descriptions of the header could be different

176 DESCRIPTOR OF CODE Declaration: ***Micromed_New_Descriptor Code_Area;***

This is the pointer to the area that contains the code of the electrodes stored in the file. Codes are written in the order than data are stored in the file. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string ‘ORDER’
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of Code Area
12	**Length**	unsigned long int	Length, in bytes, of Code Area

192 DESCRIPTOR OF ELECTRODE Declaration: ***Micromed_New_Descriptor Electrode_Area;***

This is the pointer to the area, which contains the description of the electrode conversion parameters. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string ‘LABCOD’
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of Electrode Area
12	**Length**	unsigned long int	Length, in bytes, of Electrode Area

208 DESCRIPTOR OF NOTE Declaration: ***Micromed_New_Descriptor Note_Area;***

This is the pointer to the area, which contains the operator notes. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string ‘NOTE’
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of Note Area
12	**Length**	unsigned long int	Length, in bytes, of Note Area

224 DESCRIPTOR OF FLAG Declaration: ***Micromed_New_Descriptor Flag_Area;***

This is the pointer to the area, which contains the flags inserted by the user. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string 'FLAGS'
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of Flag Area
12	**Length**	unsigned long int	Length, in bytes, of Flag Area

240 DESCRIPTOR OF REDUCTION Declaration: ***Micromed_New_Descriptor Segment_Area;***

This is the pointer to the area, which contains the description of the reduction made to the file. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string 'TRONCA'
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of Reduction Area
12	**Length**	unsigned long int	Length, in bytes, of Reduction Area

256 DESCRIPTOR OF BEGIN IMPEDANCE Declaration: ***Micromed_New_Descriptor B_Impedance_Area;***

This is the pointer to the area which contains the description of the impedance measured at the beginning of the acquisition. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string 'IMPED_B'
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of Begin Impedance Area
12	**Length**	unsigned long int	Length, in bytes, of Begin Impedance Area

272 DESCRIPTOR OF END IMPEDANCE Declaration: ***Micromed_New_Descriptor E_Impedance_Area;***

This is the pointer to the area, which contains the description of the impedance measured at the end of the acquisition. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string 'IMPED_E'
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of End Impedance Area
12	**Length**	unsigned long int	Length, in bytes, of End Impedance Area

288 DESCRIPTOR OF MONTAGES Declaration: ***Micromed_New_Descriptor Montage_Area;***

This is the pointer to the area, which contains the description of the specific montages stored for this trace. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string 'MONTAGE'
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of Montage Area
12	**Length**	unsigned long int	Length, in bytes, of Montage Area

304 DESCRIPTOR OF COMPRESSION Declaration: ***Micromed_New_Descriptor Compression_Area;***

This is the pointer to the area, which contains the description of the compression used to store data. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string 'COMPRESS'
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of Compression Area
12	**Length**	unsigned long int	Length, in bytes, of Compression Area

320 DESCRIPTOR OF AVERAGE Declaration: ***Micromed_New_Average Average_Area;***

When present, this is the pointer to the area, which contains the information of an Off-Line Average process, which has been done, on the described traces. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string 'AVERAGE'
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of Average Area
12	**Length**	unsigned long int	Length, in bytes, of Average Area

336 DESCRIPTOR OF HISTORY Declaration: ***Micromed_New_Descriptor History_Area;***

This is the pointer to the area which contains the information necessary to perform the "As Recorder" review of the trace. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string 'HISTORY'
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of History Area
12	**Length**	unsigned long int	Length, in bytes, of History Area

352 DESCRIPTOR OF DVIDEO Declaration: ***Micromed_New_Descriptor Dvideo_Area;***

This is the pointer to the area which contains the information necessary to play all Digital Video linked to the EEG file. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string 'DVIDEO'
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of Dvideo Area
12	**Length**	unsigned long int	Length, in bytes, of Dvideo Area

368 DESCRIPTOR OF EVENT A Declaration: ***Micromed_New_Descriptor EventA_Area;***

This is the pointer to the area, which contains the pointers to the event of type "A" marked by the user. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string 'EVENT A'
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of Event A Area
12	**Length**	unsigned long int	Length, in bytes, of Event A Area

384 DESCRIPTOR OF EVENT B Declaration: ***Micromed_New_Descriptor EventB_Area;***

This is the pointer to the area, which contains the pointers to the event of type "B" marked by the user. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string 'EVENT B'
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of Event B Area
12	**Length**	unsigned long int	Length, in bytes, of Event B Area

384 DESCRIPTOR OF TRIGGER Declaration: ***Micromed_New_Descriptor Trigger_Area;***

This is the pointer to the area which contains the pointers to the digital triggers acquired during recording. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string 'TRIGGER'
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of Trigger Area
12	**Length**	unsigned long int	Length, in bytes, of Trigger Area

416 DESCRIPTOR OF BRAIN IMAGE Declaration: ***Micromed_New_Descriptor Brain_Img_Area;***

This is the pointer to the area which contains the pointers to the Brain image saved in connect. The descriptor has the following structure:

Offset	Label	Data Type	Description
0	**Name**	char[8]	fixed string 'BRAINIMG'
8	**Start_Offset**	unsigned long int	Start Offset, in bytes, of Brain Img Area
12	**Length**	unsigned long int	Length, in bytes, of Brain Img Area

432 RESERVED AREA Declaration: ***unsigned char Reserved_2[208];***

This is an area reserved by Micromed for further development. It's not to be used for other purposes.

The following is the description of the structure of each information area that can be encountered in the header.

- **ORDER: Order of storage of the electrodes**

This is an array, which contains the coded list of the electrodes stored in the file. Each byte specifies the code of signal entered in the amplifier, which has made the measurement. The number of channels stored can be found at location **142-NUMBER OF CHANNELS STORED.** The code refers to the description of the electrodes described in the next paragraph, i.e. 1 = Fp1, 2 = Fp2 and so on. If, for example, you find a Byte with the value **0x01**, it means that the electrode is **Fp1-G2**, as the second record of **ELECTRODE** (position 1, location 128) specifies **Fp1-G2.**

Declaration: ***unsigned short int Code [MAX_CAN];***

- **ELECTRODE: Description of electrodes**

This is a list of electrodes that could be used by the system, and contains all the information needed about the electrode signal conversion. To see which of these electrodes are stored, see previous area, **ORDER**, which specifies the order and the number of electrodes used. Number 0 means the first electrode of this area (that is always G2), number 1 means the second electrode of this area (located 128 bytes after the previous). Every cell of **ELECTRODE** is a record of 128 bytes with the following structure:

Declaration: ***Micromed_New_Electrode Electrode [MAX_LAB];***

offset	Label	Type	Description	Typical Values
0	**Status**	unsigned char	Status of Electrode for acquisition	0 = not acquired 1 = acquired
1	**Type**	unsigned char	Reference and Type of Electrode	bit0 (LSB) is for Reference: • 0 = Referred to G2 • 1 = Bipolar if bit1=1 then type=Marker if bit2=1 then type=Oxym. if bit3=1 then type=16DC if bit4=1 then type=bip2eeg
2	**Positive_Input_Label**	char[6]	Label of Positive Electrode Input	'label',0
8	**Negative_Input_Label**	char[6]	Label of Negative Electrode Input	'label',0
14	**Logic_Minimum**	long int	Logic Minimum of Signal	
18	**Logic_Maximum**	long int	Logic Maximum of Signal	
22	**Logic_Ground**	long int	Logic Value of Ground	LOGIC_MIN≤value value≤LOGIC_MAX
26	**Physic_Minimum**	long int	Physical Minimum of Signal	
30	**Physic_Maximum**	long int	Physical Maximum of Signal	
34	**Measurement_Unit**	unsigned short int	Measurement Unit	-1 = nVolt 100 = % 0 = μVolt 101 = bpm 1 = mVolt 102 = Adim. 2 = Volt
36	**Prefiltering_HiPass_Limit**	unsigned short int	HighPass Prefilitering	value in Hz ∗ 1000
38	**Prefiltering_HiPass_Type**	unsigned short int	HighPass Filter Type	
40	**Prefiltering_LowPass_Limit**	unsigned short int	LowPass Prefiltering	value in Hz
42	**Prefiltering_LowPass_Type**	unsigned short int	LowPass Filter Type	
44	**Rate_Coefficient**	unsigned short int	Sampling Rate Coefficient	1 ⇒ min. Sampling Rate 2 ⇒ 2∗min. Sampling Rate 4 ⇒ 4∗min. Sampling Rate
46	**Position**	unsigned short int	Reserved	
48	**Latitudine**	float	Map's Latitudo (radius=1)	
52	**Longitudine**	float	Map's Longitudo (radius=1)	
56	**presentInMap**	unsigned char	Presence of electrode in map	0 = electrode not present 1 = electrode is present
57	**isInAvg**	unsigned char	Used for AVG calculation	0 = electrode is not used 1 = electrode is used
58	**Description**	char[32]	Electrode Description String	'max. 31 characters',0
90	**x**	float	x coordinate for 3D map analysis	
94	**y**	float	y coordinate for 3D map analysis	
98	**z**	float	z coordinate for 3D map analysis	
102	**Coordinate_Type**	unsigned short int	Type of coordinate used for 3D	0 = polar map (Lat. & Lon.)

			map analysis	1 = XYZ 3D-system
104	**Free**	unsigned char[24]	reserved for future use	

- **NOTE: Operator Annotation**

These are the operator notes that are written during acquisition or subsequent review of the patient signals. The structure of this area is very simple: Each record contains a pointer to the instant of the note insertion and a string containing the annotation text. This area has the following structure:

Declaration: ***Micromed_New_Annotation Note [MAX_NOTE];***

offset	Label	Data Type	Description
0	**Sample**	unsigned long Int	The time in samples, from the start of signal recording, at which the note has been recorded. Used to synchronise the placement of the note with respect to the signal data with which it is associated.
4	**Comment**	char[40]	The actual ASCII text of the note to a maximum of 40 characters eventually filled with spaces. No terminators are used here.

WARNING:
1. A time of 0000 means that there are no more notes, so a note in sample 0000 can't be placed.
2. All notes are in ascending order of time.

- **FLAG: Pointers to selection flags**

A trace may contain portions of data, which the user has highlighted with the FLAGs as an area of interest or for further elaboration. Most flags come in pairs, one start and one end limit. The user may place up to MAX_FLAG pairs of flags within any trace and all flags are in ascending order (decided on the Begin field). A single flag may exist if a start was marked but no end is marked. The flag, start or end is formed of 4 bytes (long int) that is the numerical value of the record number or sample number of the marked data. Therefore, the time position of the flag is dependent on the sampling rate in order to calculate the flag's exact position in time within the trace. Any value of 0000 means no more flags, so if the first value is 0000 then no user set flags are present. (Note also that a flag in position 0000 can't be placed!). The area is an array of record with the following structure:

Declaration: ***Micromed_New_Marker_Pair Flag [MAX_FLAG];***

Offset	Label	Data Type	Description
0	**Begin**	Long Int	Starting sample of selected segment
4	**End**	Long Int	Ending sample of selected segment

Example: a value of 2560 in a trace file sampled at 256 Hz, means that a flag has been set by the user at sample 2560 or at 2560/256 = 10 sec. from the beginning of the trace.

- **REDUCTION: Reduced file description**

It is possible that the file in use is not an original file but a "reduced" file, created by extracting one or more portions from a file (i.e. using flags to cut out interesting portions) or joined together from two or more separate files. These files have the same structure as an original, so it must be possible to find the points of data which are not contiguous and also to allow a correct reconstruction of the exact time. It is possible to have up to a maximum of MAX_SEGM separate pieces in a reduced file, so this area is an array of MAX_SEGM elements with the following structure:

Declaration: ***Micromed_New_Segment Segment [MAX_SEGM];***

Offset	Label	Data Type	Description
0	**Time**	unsigned long int	Real value of the sample
4	**Sample**	unsigned long int	Starting sample of reduced segment

Using the starting sample value, the programmer may find at which sample of the current file data starts for a particular segment. The real sample in the original file of the beginning of that segment is contained in the other value, called the real value of the sample. The reduction indicators terminate when the first string of real value equal to 0000 is found or the end of this reserved area is reached. Note that all these values point to or are indicative of the sample number. The following is an example of what could be found in the reduction area:

ORIGINAL FILE

0 ..39	40..139 to be cut	140..169	170..269 to cut	270...300

REDUCED FILE

0..39	40..69	70..99

real value	starting sample
140	40
270	70

- **STARTING IMPEDANCE: Electrodes impedance at the beginning of the recording**

This area contains the measurements of the impedance of the electrodes *at the beginning of the recording*, that are stored in the file. The order is the one that is specified in the structure **CODE.** The values are expressed in KΩ but they are multiplied by 10, so effectively it is a value in hundreds of Ω. The maximum values are 200 (20 KΩ) if a 32 channel headbox has been used, and 176 (17.6KΩ) if a 24 channel headbox has been used. The records of the array have the following structure:

Declaration: ***Micromed_New_Impedance Impedance_B [MAX_CAN]);***

Offset	Label	Data Type	Description
0	**Positive**	unsigned char	Value of the impedance measured at positive Input
1	**Negative**	unsigned char	Value of the impedance measured at negative input

Each value specifies the impedance of the positive and negative input of the amplifier, and for channels referred to G2 the negative impedance will be that of G2 (if measured). If any measurement hasn't been taken, a value of 255 will be found.

- **ENDING IMPEDANCE Electrodes impedance at the end of the recording**

This area contains the measurements of the impedance of the electrodes *at the end of the recording*, The description is exactly the same than that for STARTING IMPEDANCE.

Declaration: ***Micromed_New_Impedance Impedance_E [MAX_CAN]);***

- **SPECIFIC MONTAGES: Specific visualisation montages for the trace**

This is a group of specific montages stored in the file, to be used only to specify a particular configuration of parameters to be used to display the data stored in the file. *This information is not necessary for the interpretation of the trace.* The structure of this area is the following:

Declaration: ***Micromed_New_Montage Montage [MAX_MONT];***

Offse	Label	Data Type	Description	Typical Values
0	**Lines**	unsigned short int	Number of Lines	1..128
2	**Sectors**	unsigned short int	Number of Sectors	1,2
4	**Base_Time**	unsigned short int	Length of Page	1..40 sec.
6	**Notch**	unsigned short int	Notch Filter	0..10
8	**Colour**	unsigned char[MAX_CAN_VIEW]	Line Colour	0 = Not Active 60 = 60 Hertz 50 = 50 Hertz
136	**Selection**	unsigned char[MAX_CAN_VIEW]	Selection of Lines	0 = Not Selected 1 = Selected
264	**Description**	char[64]	Montage Description	
328	**Inputs**	Micromed_New_Inputs[MAX_CAN_VIEW]	Electrodes Montage	**NonInv** = Non Inverting Input **Inv** = Inverting Input
840	**HiPass_Filter**	unsigned long int[MAX_CAN_VIEW]	High Pass Band Limit	value in Hz * 100
1352	**LowPass_Filter**	unsigned long int[MAX_CAN_VIEW]	Low Pass Band Limit	value in Hz
1864	**Reference**	unsigned long int[MAX_CAN_VIEW]	Gain Value	800μV/cm, ...

2376	**Free**	unsigned char[1720]	Reserved	

Declaration: ***Micromed_New_Inputs Inputs[MAX_CAN_VIEW];***

Offset	Label	Data Type	Description	Typical Values
0	**NonInv**	unsigned short int	Non Inverting Input	
2	**Inv**	unsigned short int	Inverting Input	

Constant Definition:

MAX_CAN_VIEW 128

- **COMPRESSION: Compression description**

This parameter would specify the kind of compression used on the digital data. Now no compression is used on any file, so this parameter is not meaningful. The structure of this area is the following:

Declaration: ***Micromed_New_Compression Compression;***

- **AVERAGE: Specific parameters for Off-Line Average process**

This is a group of specific parameters stored in the file when an Off-Line Average process has been done on the traces. These parameters are not needed to display data, but only for further process of them. *This information is not necessary for the interpretation of the trace.* The structure of this area is the following:

Declaration: ***Micromed_New_Average Average;***

Offset	Label	Data Type	Description	Typical Values
0	**Mean_Trace**	unsigned long int	Averaged Triggers	0, 1, ...
4	**Mean_File**	unsigned long int	Averaged Files	1, ...
8	**Mean_Prestim**	unsigned long int	Pre-stimuli Interval	in msec.
12	**Mean_PostStim**	unsigned long int	Post-stimuli Interval	in msec
16	**Mean_Type**	unsigned long int	Average Type	0 = Normal (By File) 1 = Weighted (By Traces)
20	**Free**	unsigned char[AVERAGE_FREE]	free	

Constant Definition:

AVERAGE FREE 108

- **HISTORY: Recording History Description**

This is the area which contains the information necessary to perform the "As Recorder" review of the trace. It means that it contains all the information of the changes occurred to the visualisation of the trace during recording. The way to memorise these data is very simple: every time a modification occur to the visualisation the instant of the modification is memorised (as the sample from the beginning of the trace) and the new visualisation configuration (the complete montage) is stored also. This space is composed of 2 main areas, that have the following structure:

Declaration: ***Micromed_New_Sample History_Sample [MAX_SAMPLE];***

Micromed_New_Sample is an *unsigned long int* that is the pointer to the sample from which the montage must be changed.

Declaration: ***Micromed_New_Montage History [MAX_HISTORY];***

The description of this structure is the same of that for SPECIFIC MONTAGES.

- **DVIDEO: Pointers to Digital Video files linked to the EEG**

This area contains the pointers to the Digital Video file linked to the EEG. The records of the array have the following structure:

Declaration: ***Micromed_New_DVideo Dvideo [MAX_FILE];***

Offset	Label	Data Type	Description
0	**Start_time**	Long Int	Starting time in msec (offset from beginning of file)
4	**Length**	Long Int	Length in msec of video file
8	**Video_num**	Long Int	ID Video file
12	**Free**	Long Int	Always -1 (0xFFFFFFFF)

Constant Definition:

MAX_FILE 256

- **EVENT "A" : Pointers to events of type "A"**

A trace may contain portions of data, which the user has highlighted as they contain a particular event like Hyperventilation, Visual Stimulation or other. The user may mark up to MAX_EVENT event on a single trace, and the data structure contains all the pointers to these segments and the general description of this event. The exact structure of this area is the following:

Declaration: ***Micromed_New_Event Event_A;***

Offset	Label	Data Type	Description
0	**Description**	Long Int	Description of the kind of event
64	**Selection**	Micromed_New_Marker_Pair[MAX_EVENT]	Pointers to marked segments

Selection is an area with the following structure

Offset	Label	Data Type	Description
0	**Begin**	Long Int	Starting sample of marked segment
4	**End**	Long Int	Ending sample of marked segment

All marked segments within any trace are in ascending order (decided on the Begin field). A Begin record with no End set may exist if a start was marked but no end has been marked. The Begin or End values are the numerical record number or sample number of the marked data. Therefore, their time position is dependent on the sampling rate with the same relation already discussed for the FLAGS.

- **EVENT "B" : Pointers to events of type "B"**

The description is the same than that for EVENT "A".

Declaration: ***Micromed_New_Event Event_B;***

- **TRIGGER: Digital Triggers**

This area contains the digital triggers acquired via RS232 or inserted off-line. The records of the array have the following structure:

Declaration: ***Micromed_New_Trigger Trigger [MAX_TRIGGER];***

Offset	Label	Data Type	Description
0	**Trig_Sample**	unsigned long int	Pointer to the sample where the trigger was received
1	**Trig_Value**	unsigned short int	value obtained from RS232

Constant Definition:

MAX_TRIGGER 8192

- **BRAIN_IMG: Brain Image**

This area contains the brain image file format JPEG loaded from connect menu for the electrode position representation on the brain

Declaration: ***JPEG Image [MAX_LENGTH]***

Constant Definition:

MAX_LENGTH 200KB

GENERAL NOTE

1. The sampled data is always 1 or 2 bytes long, they are unsigned, recorded in EXCESS XXX standard, where XXX can be read for each electrode at the location **Logic_Ground**. This means that the value range from a **Logic_Minimum** (normally 0) to a **Logic_Maximum** (normally 255 or 4095) and the physical zero is at **Logic_Ground**. Let's suppose we have a convertible signal ranging from -400μV (value of **Physic_Minimum**) to 400μV (value of **Physic_Maximum**), converted with 12 bits (value of **Logic_Minimum** 0, **Logic_Maximum** 4095, **Logic_Ground** 2048). If we obtain a digital value of 4000, the real value of the sample can be obtained with the following calculation:

$$\left[\frac{\text{Digital Value} - \text{Logic\_ Ground}}{\text{Logic\_ Maximum} - \text{Logic\_ Minimum} + 1}\right] \cdot \left[\text{Physic\_ Maximum} - \text{Physic\_ Minimum}\right] = \frac{4000 - 2048}{4096} \cdot 800 = 381.25\mu v$$

2. When present, the MARKER (labelled "MKR+", code 160) is a square wave whose amplitude has to be used as a calibration, since it's amplitude is normally 100μV (from -50 to 50μV).

3. The samples are always stored referred to the common electrode (labelled "G2"), so the value of the common electrode must not be subtracted to data. Average or any other reference specified is always re-calculated by the software.

4. Differences from files of type 2 are the following:
   - ***Header*** has a value of 3 and no longer a value of 2
   - ***Note*** pointers are no longer in seconds but in samples
   - ***Data_Start_Address*** is no longer forced to be multiple of ***Multiplexer***
   - ***Dvideo_Begin*** has been added
   - ***Event_A***, ***Event_B*** and ***History*** descriptor and relative structure have been added
5. The differences from file of type 3 are the following:
   - ***Code*** is unsigned short int instead of unsigned char
   - ***Electrode*** has a different structure
   - ***Specific*** Montage has a different structure
   - ***DVideo***, ***Average***, ***Triggers***, ***BrainImg*** descriptors have been added