Vital File Format

Introduction

- The vital file is a file format used in vital recorder for saving vital signs and waveforms
  - It is a single gzip compressed binary file starts with (1F 8B 08 00)
  - It contains a header and a body part
  - Multi-byte variables are encoded using littlen-endian
  - All data structures are arranged in 1-byte units.
    - Caution! Some compilers use 4 byte alignment implicitly.[1]

Data format

- String data format
  - Strings are encoded in UTF8
  - It contains 4 byte length (without length itself) followed by character array of specified length
  - Caution! It does not include the following NULL(\0) character
- Time data format
  - Ttimes are stored in UTC
  - UTC can be converted to the local time using time zone bias (offset) in the header part.
    - UTC = local time + tzbias[2]
  - All time data are represented as the number of seconds since 1970/01/01 00:00:00 UTC as a double data format
  - This is compatible with Unix timestamps and convenient for addition and subtraction.
  - Changing the time to __int64 makes the speed slower because it require conversion in every operation.
  - The decimal point is used to indicate the precision below seconds. (for example 0.1 means 100 milliseconds)
  - Because the unix timestamp of year 2020 is 1,606,780,800, 31 bits are enough to express integer part of it. IEEE 754 double data type has 52 bit for fraction part, so 21 digits can be used for the representation of sub second time. 2 ^ 20 is about 1 million, so it has about 1 usec resolution.

Header

- The total length of the header is 10 + headerlen.
- The vital file does not include the number or length of tracks in the header, you must parse the file to get the number of tracks or records. This limitation is because vital recorder can not know when recording will stop and you can add tracks at any time while vital files are storing.

---

[1] For visual studio, use the following code
#pragma pack (push, 1)

struct TRKINFO {... code ...};

#pragma pack (pop)

[2] time_t t = (time_t)(dt - tzbias * 60); // unixtime -> localtime
tm* ptm = gmtime(&t); // local time (in seconds) → year, month, day, hour, minute, second

Name	Type	Length	Description
sign	BYTE[4]	4	“VITA”
format_ver	DWORD	4	File format version, currently 3
headerlen	WORD	2	header length after this, currently 10
tzbias[3]	short	2	time zone bias (the difference, in minutes, between UTC time and local time)
inst_id	DWORD	4	The instance ID of the program to be issued when the program starts. It is necessary to verify that the location of several vital files are continuously recorded in the same vital recorder execute. If different vital files have the same inst_id, you can use the same track id, even though there is a probability of 1/4.2 billion for collision.
prog_ver	DWORD	4	vital recorder version

Body

- Body is a sequence of packets as specified below.
- Since the length of a packet is not constant, there is no way to directly index a specific packet without reading all packets. However, if you ignore the unwanted packet type and pass it using datalen field, you can read the information you want fairly quickly. (For example, when you want to extract only some track)

Packet structure

Name	Type	Length	Description
type	BYTE	1	SAVE_DEVINFO = 9 SAVE_TRKINFO = 0 SAVE_REC = 1 SAVE_CMD = 6
datalen	DWORD	4	length of **data** (**exclude** type and datalen itself) If you meet the type you dont know, please ignore it and skip the datalen bytes.
data		datalen	The contents of data depends on the type and are described below.

DEVINFO

- Structure for device information
  - It must be saved before the devid is used.
  - If devid is 0, it indicates the vital recorder itself (for example, filter-generated tracks, event markers)
- A new DEVINFO can appear for the same device id (devid) in a file. In this case, the new value should overwrites the previous value.

[3] TIME_ZONE_INFORMATION tzi;
GetTimeZoneInformation(&tzi);
return (short) tzi.Bias; // UTC = local time + bias // -540 for Korea

- devid and trkid are meaningful only within the same file.
- Even if the program instance is different, the Vital Recorder is implemented with the same devid value if the same kind of equipment is connected to the same physical port.
    - For example, if two BIS devices are connected to COM5 and COM6, the previous devid remains the same even if the program is terminated and then re-executed. (save the devid to the registry)
    - If you intentionally terminate the program and then switch between the two BIS devices and run the program again, the devids may cross each other.

Name	Type	Length	Description
devid	DWORD	4	device identifier
typename	string	4+len	device type
devname	string	4+len	device name
port	string	4+len	Information that allows device types such as COM1, COM10, ETH1, and Line Input Mixer to recognize the port to which the device is connected with this information only. If the device type and port are the same, it is generally considered the same device.

TRKINFO

- Structure for track information
- TRKINFO must be appeared before the first record with the track id.
    - Otherwise, the record should be ignored.
- Tracks are separated by trkids, which are 2-byte integers.
    - The numeric value of trkid itself has no meaning and there is no continuity
    - trkids can be changed every time the program is executed.
    - The order of the tracks is not in the order of trkid size, but in the order of appearance and the CMD_ORDER command described later.
- Tracks with a devid of 0 (vital recorder itself) and name of EVENT are treated specially in the vital recorder. It is not displayed on a separate track but displayed in the event bar.

Name	Type	Length	Description
trkid	WORD	2	track id
rec_type	BYTE	1	TYPE_WAV = 1 TYPE_NUM = 2 TYPE_STR = 5

recfmt	BYTE	1	FMT_NULL = 0 // for TYPE_STR
			FMT_FLOAT = 1
			FMT_DOUBLE = 2
			FMT_CHAR = 3
			FMT_BYTE = 4
			FMT_SHORT = 5
			FMT_WORD = 6
			FMT_LONG = 7
			FMT_DWORD = 8

name	string	4+len	
unit	string	4+len	

minval	float	4	
maxval	float	4	
color	color	4	4byte ARGB format
srate	float	4	sample rate
adc_gain	double	8	measured_value = adc_offset + saved_value * adc_gain
adc_offset	double	8	measured_value = adc_offset + saved_value * adc_gain
montype	BYTE	1	Specifies the physiologic meaning of the track. MON_ECG_WAV = 1, MON_ECG_HR = 2, MON_ECG_PVC = 3, MON_IABP_WAV = 4, MON_IABP_SBP = 5, MON_IABP_DBP = 6, MON_IABP_MBP = 7, MON_PLETH_WAV = 8, MON_PLETH_HR = 9, MON_PLETH_SPO2 = 10, MON_RESP_WAV = 11, MON_RESP_RR = 12, MON_CO2_WAV = 13, MON_CO2_RR = 14, MON_CO2_CONC = 15, MON_NIBP_SBP = 16, MON_NIBP_DBP = 17, MON_NIBP_MBP = 18, MON_BT = 19, MON_CVP_WAV = 20, MON_CVP_CVP = 21,
devid	DWORD	4	Devid of the device that created the track

REC

- It is a data structure that stores time specific data (measured sample, sample list, string).

Name	Type	Length	Description
infolen	WORD	2	length of header of the record, currently 10
dt	double	8	The time at which the record's value was measured
trkid	WORD	2	track id
values	BYTE[]	datalen-infolen-2	rec_type specific values described below

- values are different among the track type (NUM, WAV, STR)
- WAV records
    - Samples that are continuously measured from the time specified by dt are stored as an array.
    - The sample rate and data type are taken from the track information.

Name	Type	Length	Description

num	DWORD	4	number of samples
vals	dat_fmt[num]	sizeof(recfmt)*num	data samples

- NUM records

Name	Type	Length	Description
val	dat_fmt	sizeof(recfmt)	

- STR records

Name	Type	Length	Description
unused	DWORD	4	not used
sval	string	4+len	

- CMD records
    - Currently 2 commands are supported.
    - Depending on the cmd, additional data may follow.
    - Unknown commands should skip over the datalen of the packet.

Name	Type	Length	Description
cmd	BYTE	1	CMD_ORDER = 5 CMD_RESET_EVENTS = 6

        - CMD_TRK_ORDER
            - Define track order
            - The following additional data are shown below.

Name	Type	Length	Description
cnt	WORD	2	number of tracks
trkids	WORD [cnt]	cnt*2	array of trkids

        - CMD_RESET_EVENTS
            - Removed all event markers prior to this command
            - No additional data