Blaster Interna – A documentation of internal format of the data stucture, creation and configuration of the sample sets of the ElMod ThinkTank Blaster sound module

Jegliche Haftung und Ansprüche jeglicher Art durch den Nutzer oder Dritte, welche aus der Nutzung dieser Information entstehen könnten, sind ausgeschlossen. ElMod übernimmt ausdrücklich keine Garantie für die Vollständigkeit oder Nutzbarkeit dieser Information und behält sich stets vor, jederzeit Veränderungen ohne vorherige Ankündigung zu machen.

We do not ensure any technical support for modifying the data of the Blaster module! Nevertheless we will give assistance with pleasure, when having some time. So please don't nail us down for a fast and detailed answer.

Table of content

ElMod Dipl.-Inf.(FH) Th. Kusch & Jürgen K. Huber GbR

info@el-mod.de   http://www.el-mod.de

1. Basic information

With the ThinkTank Blaster sound module a user with the usual knowledge of using a personal computer is able to create his own sample sets or to modify the existing ones.

The requierments are:

– A SD card reader for a personal computer
– Knowledge about file management (copying, editing, renaming..)
– Software converting tool and/or a wave editor(ex. the free available editor Audacity®)
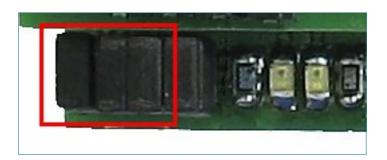– much time and patience

We also plan to release a Windows(tm) tool for easier handling of the Blaster data.

2. Directory tree

The ThinkTank Blaster module is equiped with a reader for Secure Digital memory card. On this card up to 8 complete sample sets, which are independently of each other can be saved. A sample set can only access its own samples which are saved in its directory.

Every sample set is saved in a unique directory. The name of the directory corresponds to the position of the three most left jumpers on the Blaster PCB. Thus the directory name always consits of three binary digits ('0' and '1').

A sample set is activated when the jumper position equals the name of the directory of the desired sample set.

Example: a new SD card delivered with a ThinkTank Blaster always includes a sample set which is active when the three relevant jumpers are set. The data of the sample set is stored in a directory named '111'. Mostly there is also an alternative sample set which is active when the first jumper (the most left) is removed. This set can be found in the directory '011'.

Consequently the Blaster distinguishes between up to 8 sample sets, which directory names are: '000', '001', '010', '011', '100', '101', '110' and '111'

The fourth jumper is reserved for further use and has no functionality yet.

If the jumper setting doesn't correspond to an existent sample set, the Blaster reports an error with a special blinking pattern of the first LED (see chapter 'error messages').

3. Sample sets

Every sample set consists of three file types:

- A configuration file named 'blaster.ini'
- Several samples in wave-format
- An executable file (for Windows(tm)) fchecker.exe

3.1 Blaster.ini

The file Blaster.ini defines all parameters of a sample set and assigns specific events to every defined sample file.

This file is an ASCII text file and can be viewed and edited with any text editor.

There are some restrictions which have to be considered when changing the content of the file:

- There may not be any blank characters in front of a new entry.
- Between every parameter a single space has to be placed.
- After the last entry a carriage return (new line) has to be inserted

The content of the Blaster.ini is divided into two parts:

- Configuration
- Sample definition

3.1.1 Configuration

Every configuration entry starts with the string 'CFG'. An identifier and one to three parameters follow.

In the current firmware version two identifiers are defined:

- 'M' for configuration of the speed steps of the main engine
- 'C' for configuration of the chain squeaks

Configuration of the speed steps

Syntax: CFG M s*tart*

Values range:
start: 0 – 127

Suggestive values:
start: 30 – 60

This entry defines the voltage level of the main engines on which the sample of the idle motor isn't played anymore (because the tank starts to move).

The level depends on the condition of the soil, tank type, type and shape of the motors, gearboxes and chains.

Example:

CFG M 50

Meaning: the sample of an idle motor will be played up voltage level 50.

It is recommended that the value is chosen below the point the tank starts moving. The effect is, that the motor starts to goose before the tank starts moving.

Configuration of chain squeak

Syntax: CF *start t1 t2*

Values range:
start: 0 – 127
t1: 50-5000
t2: 50-5000

Suggestive values:
start: 30 – 60
t1: 500-1000
t2: 50-200

The first parameter defines the voltage level of the main engines on which the squeak sounds are played (similar to the parameter of the 'CFG M' entry).

The second parameter defines the delay between the squeak samples on slowest speed in mili seconds and the third parameter defines the delay between the squeaks on full speed (also in mili seconds).

Example:

CFG C 60 2000 100

Meaning: the chains start to squeak on voltage level 60. The delay between the squeaks are 2000ms (2 seconds) when driving very slow and 100ms (0,1second) when driving with maximum velocity. The delay between both speed levels is interpolated accordingly.

The smallest possible value for the delay is 50 ms (0,05 sec).

The first parameter (the voltage level) should be just a little bit above the value on which the tank starts to move.

3.1.2 Sample definitions

All used samples are also defined and assigned to a specific event in the Blaster.ini.

The sample must be provided as a wave file (uncompressed (PCM), 8-bit, 22kHz, mono). Any other format cannot be played correctly and leads to undesired effects.

All file names of the samples must be compatible to DOS. This means no special characters except of '-' and '_', the length of the file name must not exceed 8 characters plus the extension '.wav'.

Some examples for correct names of the sample files:

ignition.wav
shot.wav
turret_2.wav

Examples for wrong file names:

turretrotation.wav   (name too long)
myshot!.wav   (special character '!')
engine   (missing the extension '.wav')

furthermore consider the limit of 80 (Blaster CPU uo tp version 1.02) or 100 (Blaster CPU version 1.03 and higher) samples per sample set. When more than 80 samples are defined, any sample above the cap will be ignored.

Each defined sample must stay at the very beginning of a new line. Next to the sample name the the assignment to a event and optional further parameters take place.

The folowing event types are defined:

− 'M' – events related to the main engine (ignition, stopping, engline idle, drive noise)

− 'C' – chain squeak samples

− 'T' – turret rotating with running main engines

− 'A' – alternative turret rotating (played when the engine is off)

− 'B' – barrel movement

− 'S' – main gun shot

− 'G' - MG shot

− 'U' – user defined samples

Event category 'M' (main engines)

Syntax: *sample.wav* M *param*

Values range:

param: ON,OFF,WARM, 0-127

This category handles the following events:

− 'ON' – ignition of a cold engine

− 'WARM' – ignition of a warm engine

− 'OFF' – motor stop

− # - motor noise on # speed step

Example 1: ignition and stopping

igni1.wav M ON
igni2.wav M ON
elstart.wav M WARM
stop.wav M OFF

In this example the following samples are defined:

igni1.wav and igni2.wav will be played when igniting the engine for the first time or after the engine was off for more than 5 minutes (cold start). The sample that will be played is determined randomly.

elstart.wav will be played when the engine was off for less than 5 minutes.

stop.wav will be played when switching the engine off.

Each of the three event types ('ON', 'WARM' and 'OFF') can be assigned to up to 10 samples. The 'WARM' type is optional.

Example 2: motor idle and driving

fs00.wav M 0

fs01.wav M 1
fs02.wav M 2
fs03.wav M 3
fs04.wav M 4
fs05.wav M 5
fs06.wav M 6
fs07.wav M 7
fs08.wav M 8
fs09.wav M 9
fs10.wav M 10

In this example 11 samples are defined. The sample 'fs00.wav M 0' corresponds to the sound of an idling motor. The other samples are played with the ascending revolutions of the engine. The sample 'fs01.wav M1' is played on motor voltage level defined with the 'CFG M' entry. The last sample is played at full driving speed.

We recommend to define at least 10 samples for engine sound. More than 20 samples don't make any significant improvement.

Event category 'C' chain squeak

Syntax: *sample.wav* C

This category defines up to 20 different samples for chain squeak that are chosen randomly when the motor voltage level reaches the value defined in the 'CFG C' entry.

Example: five samples for chain squeak

chain1.wav C
chain2.wav C
chain3.wav C
chain4.wav C
chain5.wav C

No further parameters are defined for this category

Event category 'T' turret rotation

Syntax: *sample.wav* T *param*

Values range:

param: 1 to 7

With this category up to 7 samples can be assigned to turret rotation. The samples will be played corresponding to the turret rotation speed.

Example:

turret1.wav T 1
turret2.wav T 2
turret3.wav T 3
turret4.wav T 4
turret5.wav T 5
turret6.wav T 6
turret7.wav T 7

Event category 'A' alternative turret rotation

Syntax: *sample.wav* A *param*

Values range:

param: 1 to 7

Which this optional category up to 7 samples can be assigned to turret rotation. The samples will be played corresponding to the turret rotation speed and only when the engine is off..

The format is the same as for the standard turret rotation. If this category isn't defined, the samples defined for the standard rotation will be played

Example:

notturm1.wav A 1
notturm2.wav A 2
notturm3.wav A 3
notturm4.wav A 4
notturm5.wav A 5
notturm6.wav A 6
notturm7.wav A 7

Event category 'B' barrel movement

Syntax: *sample.wav* B ON

This sample is played when moving the barrel.

Example:

barrel.wav B ON

The parameter 'ON' is obligatory.

Event category 'S' main gun shot

Syntax: *sample.wav* G *param*

Values range:

param: ON, OFF

This sample is played when making a shot with the main gun.

shot.wav S ON

The parameter 'ON' is obligatory.
If Blaster CPU version 1.03 or higher is installed, repetive firing (for example like AA-tanks) is supported. In this case the sample with parameter 'ON' defines the repeated shot and the sample with parameter 'OFF' defines the lash shot.

Event category 'G' MG shot

Syntax: *sample.wav* G *param*

Values range:

param: ON, OFF

Two samples have to be defined for the MG shot. A loop sample with repeated shot noise and a sample for the last shot.

Example:

mgshot.wav G ON
mglast.wav G OFF

If Blaster CPU version 1.03 or higher is installed, single fire (for example smoke grenades) is supported. In this case only the sample with parameter 'ON' has to be defined.

Event category 'L' combat unit samples (Firmware 1.04 and higher)

Syntax: *sample.wav L param*

Value range:

param: REL, HIT, DEAD, RES

These samples are played when an IR combat unit is connected and the following events occure:

REL – Reloading is done
HIT – Tank is being hit
DEAD – Tank is destroyed
RES – Tank is resurrected

Example:

reloaded.wav L REL
hit.wav L HIT
dead.wav L DEAD
resurrec.wav L RES

Event category 'U' user defined samples

Syntax: *sample.wav* U *param*

Values range:

param: 1 to 4

When using a at least 6 channel RC set, the channels 5 and 6 can be used to trigger up to four different free definable samples (for example songs or other noises).

Example:

marleen.wav U 1
milord.wav U 2
singing.wav U 3
dog.wav U 4

The files are assigned as follows:

marleen.wav	Channel 5 full deflection upward
milord.wav	Channel 5 full deflection downward
singing.wav	Channel 6 full deflection upward
dog.wav	Channel 6 full deflection downward

3.1.3 Checking the content

To assert that a sample set is correct and to get detailed informations about its content, every set is contains an executable file for personal computers running Windows(tm) operating system. This file should be copied into every new created sample set.

The sample set will work properly unless the application reports no errors.

The checing procedure includes:

− listing of all physically existent samples in the sample set directory with statement about the file size and playing time.

− Checking the correctness of the sample format and file name.

− Checking the content of the 'Blaster.ini' file

  − listing and checking of the configuration parameters ('CFG')

  − listing and checking of all assignments of a sample to an event category

  − statistical information of number of defined samples in the given categories and total number of samples.

− Checking and listing of existent but not used samples.

− A summary of all warnings and errors.

4 Error messages

The Blaster module reports different modes of operation and error states with different blinking patterns of the power LED. You can see the complete list of the patterns below.

−_____   Blaster is looking for a Booster module

−−−−_____   Connected to Booster and operational (up to Blaster CPU 1.02)

−−−−−−−−   (LED is on) Connected to Booster and operational (Blaster CPU 1.03 and newer)

−−−−−_−_   No SD card found, SD card is not formatted or faulty

−−−_−_−_   No Blaster.ini found (check the jumper settings) or faulty file system

_____−_−   Maximum limit of defined samples per sample set is reached (80 samples per sample set)

___−_−_−   A sample is defined in the Blaster.ini but cannot be found in the directory or the format of the sample is wrong. The sample will be ignored.

−_−_−_−_   Other critical error (mostly a hardware problem or damaged SD card)

5 Example of a valid blaster.ini

A valid blaster.ini file will look like this

```
CFG M 40 4
CFG C 50 1000 100
kblstart.wav M ON
elkstart.wav M WARM
stopp.wav M OFF
fs00.wav M 0
fs01.wav M 1
fs02.wav M 2
fs03.wav M 3
fs04.wav M 4
fs05.wav M 5
fs06.wav M 6
fs07.wav M 7
fs08.wav M 8
fs09.wav M 9
fs10.wav M 10
fs11.wav M 11
fs12.wav M 12
fs13.wav M 13
fs14.wav M 14
fs15.wav M 15
fs16.wav M 16
fs17.wav M 17
fs18.wav M 18
fs19.wav M 19
fs20.wav M 20
chain1.wav C
chain2.wav C
chain3.wav C
chain4.wav C
chain5.wav C
chain6.wav C
chain7.wav C
chain8.wav C
chain9.wav C
chain10.wav C
chain11.wav C
mg_end.wav G OFF
mg_loop.wav G ON
kwk_88.wav S ON
notturm1.wav A 1
notturm2.wav A 2
notturm3.wav A 3
notturm4.wav A 4
notturm5.wav A 5
notturm6.wav A 6
notturm7.wav A 7
turret7.wav T 7
turret6.wav T 6
turret5.wav T 5
turret4.wav T 4
turret3.wav T 3
turret2.wav T 2
turret1.wav T 1
barel1.wav B ON
marleen.wav U 1
milord.wav U 2
singen.wav U 3
hund.wav U 4
reloaded.wav L REL
hit.wav L HIT
dead.wav L DEAD
resurrec.wav L RES
```