

MakeRom

Aim

It generates the ROM content files for production, with the checksum information required by **TestRom** procedure of LibSys (read **SamVS-C System Library.pdf** documentation)

Input

It takes as input a description text file, which sets the file(s) to use, what destination address, ...

The description text file has the following content:

Directives

\$output <i>NameFile</i>	It generates several files, named <i>NameFile00.bin</i> <i>NameFile01.bin</i> , ...
\$devicesize 0x200000	Set the ROM device size in words: 0x200000 for a 32Mbits, by default 0x200000
\$pattern 0x600D	Set the word value to use for gap area, by default value 0xFFFF.
\$target 2553	Default: 3816 for SAM3816. Specify the target name: 3516 (SAM3516), 3716 (SAM3716), 3816 (SAM3816), 3703 (SAM3703), 2553 (SAM2553), 3103 (SAM3103), 3303 (SAM3303), 3108 (SAM3108), 3308 (SAM3308), ...

Common entry

<i>Namefile</i>	<i>AbsoluteAddress</i>	<i>OffsetInFile</i>	<i>Length</i>
-----------------	------------------------	---------------------	---------------

<i>OffsetInFile</i>	<i>(optional)</i>	the word offset from the file <i>NameFile</i> . By default 0,	
<i>Length</i>	<i>(optional)</i>	the word length from the file <i>NameFile</i> . By default the size of the file or (size of file – <i>offsetInFile</i>) if <i>OffsetInFile</i> defined.	

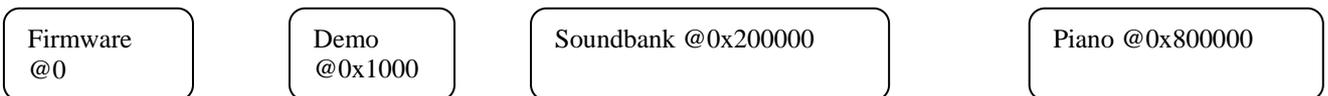
It takes, at word offset *OffsetInFile*, *Length* words data from the file *NameFile*, and it maps them at the destination address *AbsoluteAddress*.



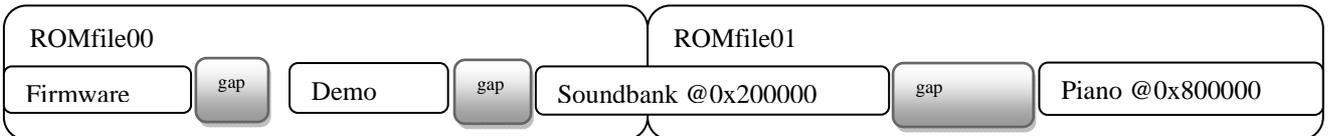
Output

Button 'Generate ROM files' creates the .bin file(s) and a .log file (in case of errors, remarks).

From the following input description,



it generates:



'Offset'

By default, it generates the file from address 0x0000. The generated files are to be used with external memory programming device, that writes entire memory chip.

With the parameter 'Offset', the first generated file (named ROMfile00z.bin) will start at specified address: If need to directly program the memory board, with ProgSam.

Remark:

- MakeRom Version 1.03 does not support overlapping files: each entry definition must **not** overlap other entry definition. It reports this situation as error.
- The low part (the first 0x4000 words) of the firmware (i.e. for the file set at address 0x0000) is not copied in the output file (mandatory for a secured firmware): it is replaced by a gap area (using the value defined by \$pattern).
- In case of target 2553, 3103, 3303, 3108, 3308, these 0x4000 first datas of firmware file are kept, and a firmware checksum is computed, at word location 'end of firmware file -1' (the last word).