

NAME**rgblink** — linker script file format**DESCRIPTION**

The linker script is an external file that allows the user to specify the order of sections at link time and in a centralized manner.

A linker script consists of a series of bank declarations, each optionally followed by a list of section names (in double quotes) or directives. All reserved keywords (bank types and directive names) are case-insensitive; all section names are case-sensitive.

Any line can contain a comment starting with ‘;’ that ends at the end of the line.

```
; This line is a comment
ROMX $F                ; start a bank
    "Some functions"   ; a section name
    ALIGN 8            ; a directive
    "Some array"

WRAMX 2                ; start another bank
    org $d123          ; another directive
    "Some variables"
```

Numbers can be in decimal or hexadecimal format (the prefix is ‘\$’). It is an error if any section name or directive is found before setting a bank.

Files can be included by using the **INCLUDE** keyword, followed by a string with the path of the file that has to be included.

The possible bank types are: **ROM0**, **ROMX**, **VRAM**, **SRAM**, **WRAM0**, **WRAMX**, **OAM** and **HRAM**. Unless there is a single bank, which can occur with types **ROMX**, **VRAM**, **SRAM** and **WRAMX**, it is mandatory to specify a bank number after the type.

Section names in double quotes support the same character escape sequences as strings in `rgbasm(5)`, specifically ‘\’, ‘\’’, ‘\n’, ‘\r’, and ‘\t’. Other backslash escape sequences in `rgbasm(5)` are only relevant to assembly code and do not apply in section names.

When a new bank statement is found, sections found after it will be placed right from the beginning of that bank. If the linker script switches to a different bank and then comes back to a previous one, it will continue from the last address that was used.

The only three directives are **ORG**, **ALIGN**, and **DS**:

- **ORG** *addr* sets the address in which new sections will be placed to *addr*. It can not be lower than the current address.
- **ALIGN** *addr* or **ALIGN** *addr*, *offset* will increase the address until it is aligned to the specified boundary (it tries to set to *offset* the number of bits specified by *align*: for example, **ALIGN 8** will align to \$100, and **ALIGN 8, 10** will align to \$10A).
- **DS** will increase the address by the specified non-negative amount.

Note: The bank, alignment, address and type of sections can be specified both in the source code and in the linker script. For a section to be able to be placed with the linker script, the bank, address and alignment must be left unassigned in the source code or be compatible with what is specified in the linker script. For example, **ALIGN[8]** in the source code is compatible with **ORG \$F00** in the linker script.

SEE ALSO

`rgbasm(1)`, `rgbasm(5)`, `rgblink(1)`, `rgbfix(1)`, `rgbgfx(1)`, `gbz80(7)`, `rgbds(5)`, `rgbds(7)`

HISTORY

`rgblink(1)` was originally written by Carsten Sørensen as part of the ASMotor package, and was later repackaged in RGBDS by Justin Lloyd. It is now maintained by a number of contributors at <https://github.com/gbdev/rgbds>.