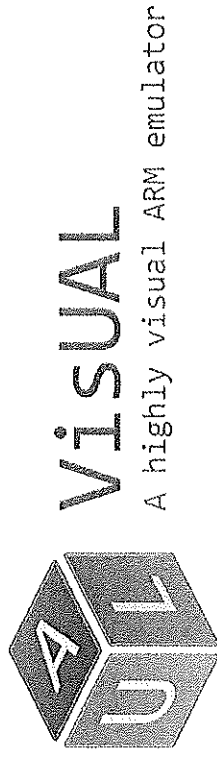


A highly visual ARM emulator

- [Home](#)
- [User Guide](#)
- [Downloads](#)
- [About](#)



Memory Map

The following diagram shows the general configuration of the VisUAL memory map:



The exact distribution of memory between resources is hardware-specific. For example, in the ARM Cortex M3, the lower 512MB of memory addresses are reserved as code memory.

Since VisUAL does not emulate external device configuration, the memory model for VisUAL is a lot simpler: the lower memory addresses are reserved for saving instructions. As a result the program counter starts with instruction 1 at address

0x0.

During execution, this instruction space memory is not available for read or write access. The addresses can still be used to manually invoke a branch, such as returning from a subroutine using the instruction `MOV PC, LR`.

Instruction Memory Configuration

By default, VisUAL sets the instruction memory size automatically to fit the available code. This scaling is done in blocks of 256 bytes. This mode enables the possibility of using the `ADR` pseudo-instruction to load symbol addresses to registers.

The instruction memory size can also be specified manually. The default manual specification is 0x10000 bytes, which allows emulation of up to 16,384 lines of code.

Memory Access Modes

VisUAL supports two memory access modes, **Open** and **Strict**. In the open access mode, all data memory addresses have read/write access. This mode is enabled by default.

In strict access mode, only data memory addresses explicitly defined using the `DCD`, `DCB` or `FILL` directives have write access. All other addresses only have read access.

These modes can be configured via the [settings panel](#).