# Example Memory Map

## Some values in memory:

|  |  |  |
| --- | --- | --- |
| Address | value | identified by |
| 0x1000 |  |  |
| 0x1004 |  |  |
| 0x1008 |  |  |
| 0x100C |  |  |
| 0x1010 |  |  |
| 0x1014 |  |  |

## Addresses here are just for the purposes of discussion

### The actual physical or virtual address of anything depends on your execution environment

# Example Memory Map (4-byte rows)

## Declaration:

char a = 12;

## What it might look like in memory:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Address | 0 | 1 | 2 | 3 |
| 0x1000 |  |  |  |  |
| 0x1004 |  |  |  |  |
| 0x1008 |  |  |  |  |
| 0x100C |  |  |  |  |
| 0x1010 |  |  |  |  |
| 0x1014 |  |  |  |  |

## Addresses here are just for the purposes of discussion

### The actual physical or virtual address of anything depends on your execution environment

# Example Memory Map (8-byte rows)

## Declaration:

char a = 12;

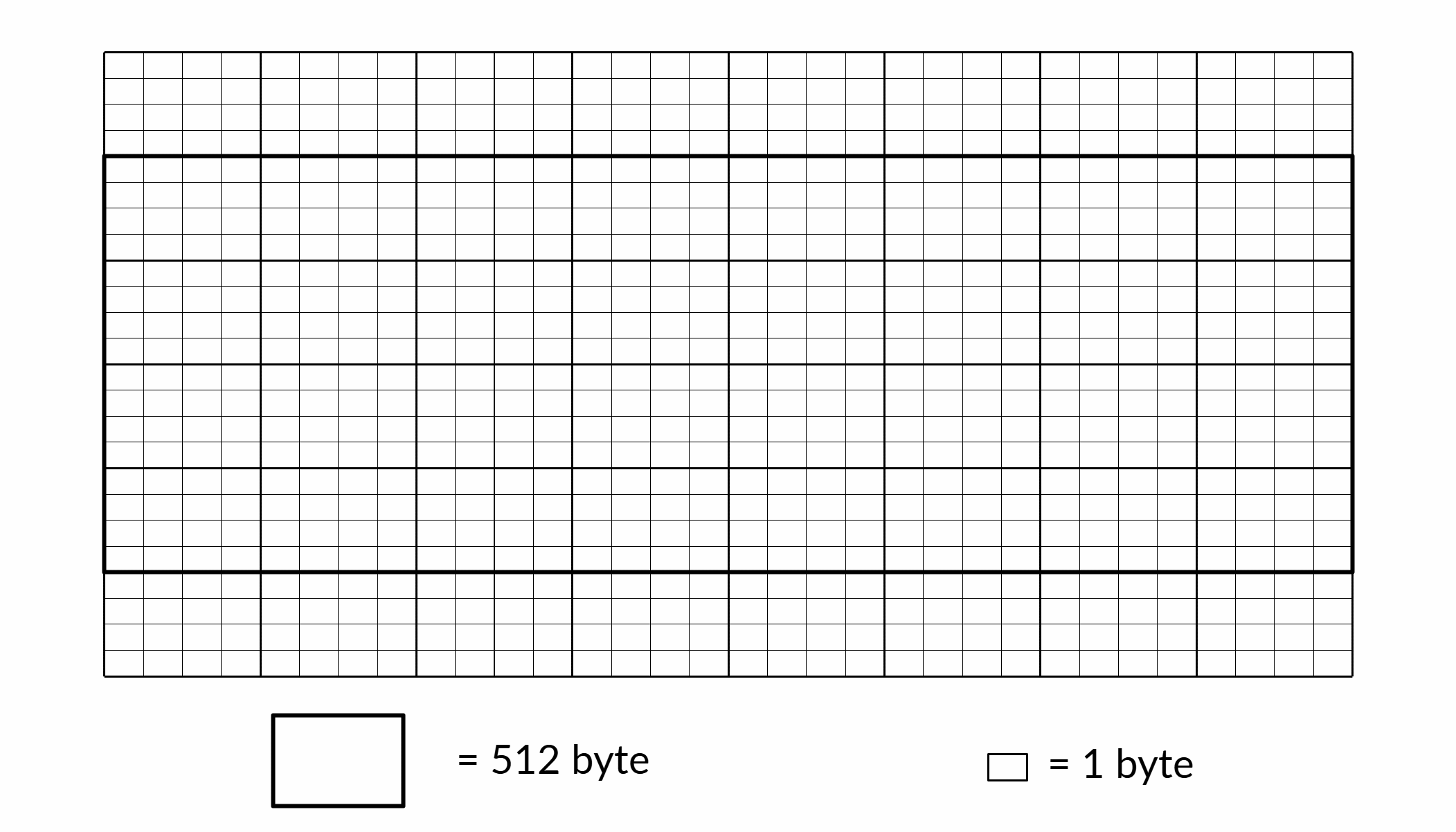
## What it might look like in memory:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0x1000 |  |  |  |  |  |  |  |  |
| 0x1008 |  |  |  |  |  |  |  |  |
| 0x1010 |  |  |  |  |  |  |  |  |
| 0x1018 |  |  |  |  |  |  |  |  |
| 0x1020 |  |  |  |  |  |  |  |  |
| 0x1028 |  |  |  |  |  |  |  |  |

## Addresses here are just for the purposes of discussion

### The actual physical or virtual address of anything depends on your execution environment

* + - 1. Memory Map (32-byte rows)



## Larger chunks of memory are often called "blocks"

* + Example - Many disk drives store and transfer 512-byte blocks

#### This doesn't seem like much in 2022 - block sizes of 1024, 2048 and 4096 bytes are also common

* + - 1. Memory Map - The Big View

## Each Operating System (on hosted implementations), and each MCU (on freestanding implementations) has it's own preferred way to to organize the address space.

* + A 32-bit x86 Linux process
  + Freestanding STM32 implementation
  + etc.

## This is usually reflected in the output of your development tools

* + The compiler/linker produce executable code that can be placed in the correct regions of memory
* When discussing large chunks of memory, memory maps are often shown with higher addresses at the top