



noForth website

## RAM & ROM in noForth

### In (Flash)ROM: CHERE ALIGN and the comma-words

! C! MOVE cannot be used with a ROM destination. Use , C, M, instead.

, ( x -- ) stores x in flashROM at CHERE and then updates CHERE to the address just after the stored number.

M, ( addr len -- ) moves a string to CHERE and then updates CHERE.

```
CREATE GR  
S" Hello!" DUP C, M, ALIGN
```

► When CREATE is used without a does-part, the created word will put the address of its 'ROM-body' on the stack:

```
GR COUNT TYPE
```

### In RAM: HERE and ALLOT

ALLOT ( n -- ) reserves n bytes at the RAM address HERE and then updates HERE to the address just after the allotted space.

```
: VARIABLE ( -- ) CREATE 1 CELLS ALLOT ;
```

► When ALLOT is used after CREATE and nothing is yet compiled in the created word, ALLOT will install an indirection ( HERE , ) and the default action of the created word will be to put the address of the allotted space on the stack.

A DOES> may follow, but then its code must explicitly fetch HERE from the ROM-body. In fact the above definition is shorthand for

```
: VARIABLE ( -- )  
CREATE HERE , 1 CELLS ALLOT  
DOES> @ ;
```

After a power off/on the content of RAM is unpredictable. Allotted RAM in your program should be initialized at run-time and not while the program is being compiled.

For this reason in noForth the definition of a value does not take a numerical argument:

```
VALUE TEMPERATURE
```

If you do not like this, redefine VALUE as

```
: VALUE ( x 'name' -- ) here value ! ;
```