TRANSPUTER MEMORY ALLOCATION

Memory is considered to be an array of type INT,
with each address being considered a subscript in that array.

*A variable, channel, timer or an array may be placed at an
absolute location in memory, using the* PLACE *statement.
The format:*

```
PLACE <item> AT <address> :
```

EXAMPLES:

```
PLACE keyboard AT link3in :

[80]INT buffer:
PLACE buffer AT #0400 :

CHAN OF REAL32 coordinates :
PLACE coordinates AT 3 :
```

*APPEAL TO COMMON SENSE: Place your items in suitable
memory locations, as defined by hardware design. In particular:*

1. *Timers should be placed at locations designed
    to act as timers,*

2. *Channels should be placed at locations designed
    to act as channels,*

3. *Arrays must not be placed so that the components
    of an array overlap other allocations.*

TRANSPUTER MEMORY MAP

In OCCAM 2, all memory is allocated at compile time.
No run-time mechanism exists for allocating memory from
the heap. Consequently, OCCAM 2 does not allow for recursive
procedure calls, or for dynamical sizing of data structures.

Workspace: *amount of memory needed to accommodate*
the code generated by the compiler.

Part of physical memory is located in the (fast) on-chip RAM.
To make effective use of it, OCCAM 2 compilers provide a
separate vector space *option, to contain arrays declared within*
a compilation unit.

```
                         ◄──── Top of memory
┌───────────────────┐
│                   │
│   Free memory     │
│                   │
├───────────────────┤
│                   │
│   Vector space    │
│                   │
├───────────────────┤
│                   │
│      Code         │
│                   │
├───────────────────┤
│  Scalar workspace │
│                   │        Bottom of memory
└───────────────────┘  ◄──── (MemStart)
```