# MemoryBank

**COLLABORATORS**

| | *TITLE* :<br><br>MemoryBank | |
|---|---|---|

| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
|---|---|---|---|
| WRITTEN BY | | July 31, 2024 | |

**REVISION HISTORY**

| NUMBER | DATE | DESCRIPTION | NAME |
|---|---|---|---|
| | | | |

# Contents

# Chapter 1

# MemoryBank

## 1.1 Memory Bank V1.00

```
Pure Basic MemoryBank library V1.00

  With this library you can allocate any number of memory banks.
  You can put them either into Chip-RAM or into Fast-RAM. It also
  includes functions to copy and fill memory. When your program ends
  the allocated memory banks will automatically be freed.

Commands summary:

   AllocateMemoryBank
   AvailableMemory
   CopyMemory
   FillMemory
   FreeMemoryBank
   InitMemoryBank
   MemoryBankAddress
   MemoryBankSize

Example:

   MemoryBank example
```

## 1.2 allocatememorybank

```
  SYNTAX
*Adresse = AllocateMemoryBank(Bank#, Size.l, MemType.l)

  COMMAND
Allocates a memory bank and returns its adress or 0, if the memory bank
couldn't be allocated. If you have already allocated a memory bank
with the same bank number, that one will be automatically freed.

Bank#: a number to identify the memory bank

Size: the size of the memory bank in bytes
```

```
MemType: with this parameter you can define additional properties
         of the memory bank:

 #MEMF_FAST  => Fast-RAM
 #MEMF_CHIP  => Chip-RAM
 #MEMF_CLEAR => Clear memory

         Just add these values or use the logical OR.

         Example: MemType = #MEMF_CHIP | #MEMF_CLEAR

         If you neither use #MEMF_FAST or #MEMF_CHIP, the
         bank will be in Fast-RAM, or in Chip-RAM, if there's
         not enough Fast-RAM available.
```

## 1.3 availablememory

```
  SYNTAX
FreeMemory.l = AvailableMemory(MemType.l)

  FUNCTION
Returns how much memory is still available.

MemType: With this parameter you can define the properties of the
         available memory.

 #MEMF_FAST => Fast-RAM
 #MEMF_CHIP => Chip-RAM

         Just add these values or use the logical OR.

         To get the amount of both Fast- and Chip-RAM, set
         MemType to 0.
```

## 1.4 copymemory

```
  SYNTAX
CopyMemory(*Source, *Dest, Size.l)

  COMMAND
Copies a memory chunk.

*Source: The source address

*Dest: The destination address

Size: The size of the memory chunk in bytes
```

## 1.5 fillmemory

```
  SYNTAX
FillMemory(*Dest, Size.l, Value.b)

  COMMAND
Fills a memory chunk with an optional Byte.

*Dest: The destination address

Size: The size of the memory chunk in bytes

Value: The value you want to fill the memory chunk with
```

## 1.6 freememorybank

```
  SYNTAX
FreeMemoryBank(Bank.l)

  COMMAND
Frees a memory bank.
```

## 1.7 initmemorybank

```
  SYNTAX
InitMemoryBank(MaxBanks.l)

  COMMAND
Allocates a memory chunk for informations about the memory banks.
This command must always be called before all other functions of
this library.
```

## 1.8 memorybankaddress

```
  SYNTAX
*MemoryBank = MemoryBankAddress(Bank#)

  FUNCTION
Returns the address of the specified memory bank.
```

## 1.9 memorybanksize

```
  SYNTAX
Size.l = MemoryBankSize(Bank#)

  FUNCTION
Returns the size of the specified memory bank in bytes.
```