

MemoryBank

COLLABORATORS

	TITLE : MemoryBank		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		August 26, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	MemoryBank	1
1.1	Memory Bank V1.00	1
1.2	allocatememorybank	1
1.3	availablememory	2
1.4	copymemory	2
1.5	fillmemory	2
1.6	freememorybank	3
1.7	initmemorybank	3
1.8	memorybankaddress	3
1.9	memorybanksizes	3

Chapter 1

MemoryBank

1.1 Memory Bank V1.00

Pure Basic MemoryBank library V1.00

With this library you can allocate any number of sized 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 be automatically freed.

Befehlsübersicht:

- AllocateMemoryBank
- AvailableMemory
- CopyMemory
- FillMemory
- FreeMemoryBank
- InitMemoryBank
- MemoryBankAddress
- MemoryBankSize

1.2 allocatememorybank

SYNTAX

```
*Adresse = AllocateMemoryBank(Bank.1, Size.1, MemType.1)
```

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 number bank, this one will 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:

```
0      = Public memory (Fast-RAM, if possible)
1      = Fast-RAM
2      = Chip-RAM
65536 = clear memory (der allocated memory will be filled with 0s)
```

Just add the values or use the logical OR.

Example: MemType = 2 | 65536 (Chip-RAM and the memory will be filled with 0s)

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.

```
0      = Public memory (Fast-RAM, if possible)
1      = Fast-RAM
2      = Chip-RAM
```

Just add the values or use the logical OR.

Example: MemType = 1 | 2 (returns the whole amount of free memory, ie. Chip- and Fast-RAM)

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 chunk. It's not important if you've already allocated a memory bank with the number bank.

1.7 initmemorybank

SYNTAX

InitMemoryBank (MaxBanks.l)

COMMAND

Allocates a memory chunk to put informations about the memory banks there. This command must always be called before all other MemoryBank-funtions.

1.8 memorybankaddress

SYNTAX

*MemoryBank = MemoryBankAddress (Bank.l)

FUNCTION

Returns the address of the specified memory bank.

1.9 memorybanksize

SYNTAX

Size.l = MemoryBankSize (Bank.l)

FUNCTION

Returns the size of the specified memory bank in bytes.
