

## **ExpansionD**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> ExpansionD		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 22, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>ExpansionD</b>	<b>1</b>
1.1	ExpansionD . . . . .	1
1.2	TMP:Modula-2/ExpansionD.def . . . . .	1

## Chapter 1

# ExpansionD

### 1.1 ExpansionD

#### Konstanten

bootTime	busWidth	chainedConfig
diagValid	errBadmem	errLastboard
errNoboard	errNoexpansion	errNomemory
errOk	expansionBase	expansionName
expansionSize	expansionSlots	memBit
memList	memMask	memSize
memoryBase	memorySize	memorySlots
never	newBoard	nibbleWide
slotMask	slotShift	slotSize
startProc	typeBit	typeMask
typeSize	z3ConfigArea	z3ConfigAreaEnd
z3ExpansionBase	z3SizeGranularity	zeroIII
zorroII		

#### Typ-Deklarationen

BootNode	BootNodePtr	ConfigDev
ConfigDevFlagSet	ConfigDevFlags	ConfigDevPtr
ConfigFlagSet	ConfigFlags	CurrentBinding
CurrentBindingPtr	DiagArea	ExpansionBase
ExpansionBaseFlagSet	ExpansionBaseFlags	ExpansionBasePtr
ExpansionControl	ExpansionRom	↔
ExpansionRomFlagSet		
ExpansionRomFlags	InterruptFlagSet	InterruptFlags
ParameterPkt	ParameterPktPtr	

### 1.2 TMP:Modula-2/ExpansionD.def

```
DEFINITION MODULE ExpansionD; (*$ Implementation:=FALSE *)
(* 10-Mar-1992/cn *)
```

```
FROM SYSTEM IMPORT ADDRESS, BITSET, LONGSET, SHORTSET;
```

```
FROM DosD IMPORT
    DeviceNodePtr , DosEnvec ;

FROM ExecD IMPORT
    ExecBasePtr , Interrupt , Library , List , Node , SignalSemaphore ;

CONST
    expansionName="expansion.library";

TYPE
    ExpansionRomFlags =(
        erf0,erf1,erf2,erf3,
        zorro3,extended,noShutup,memSpace
    );
    ExpansionRomFlagSet =SET OF  ExpansionRomFlags ;
    ExpansionRom =RECORD
        type:SHORTCARD;
        product:SHORTCARD;
        flags: ExpansionRomFlagSet ;
        reserved03:SHORTCARD;
        manufacturer:CARDINAL;
        serialNumber:LONGCARD;
        initDiagVec:CARDINAL;
        reserved0c:SHORTCARD;
        reserved0d:SHORTCARD;
        reserved0e:SHORTCARD;
        reserved0f:SHORTCARD
    END;
    InterruptFlags =(
        if0,intena,if2,reset,int2pend,int6pend,int7pend,interrupting
    );
    InterruptFlagSet =SET OF  InterruptFlags ;
    ExpansionControl =RECORD
        interrupt: InterruptFlagSet ;
        z3HighBase:SHORTCARD;
        baseAddress:SHORTCARD;
        shutup:SHORTCARD;
        reserved14:SHORTCARD;
        reserved15:SHORTCARD;
        reserved16:SHORTCARD;
        reserved17:SHORTCARD;
        reserved18:SHORTCARD;
        reserved19:SHORTCARD;
        reserved1a:SHORTCARD;
        reserved1b:SHORTCARD;
        reserved1c:SHORTCARD;
        reserved1d:SHORTCARD;
        reserved1e:SHORTCARD;
        reserved1f:SHORTCARD;
    END;

CONST
    slotSize=10000H;
    slotMask=0FFFFH;
    slotShift=16;

    expansionBase=0E80000H;
```

---

```

z3ExpansionBase=0FF000000H;
expansionSize=0800000H;
expansionSlots=8;
memoryBase=2000000H;
memorySize=8000000H;
memorySlots=128;
z3ConfigArea      =400000000H;
z3ConfigAreaEnd   =7FFFFFFFH;
z3SizeGranularity=000800000H;

```

```

typeMask=0C0H;
typeBit=6;
typeSize=2;
newBoard=0C0H;
zorroII=newBoard;
zoroIII=080H;

```

```

memMask=07H;
memBit=0;
memSize=3;
chainedConfig=3;
diagValid=4;
memList=5;

```

TYPE

```

ConfigFlags =(cf0,cf1,cf2,cf3,configTime,bindTime,byteWide,wordWide);
ConfigFlagSet =SET OF ConfigFlags ;
DiagArea =RECORD
config: ConfigFlagSet ;
flags:SHORTCARD;
size:CARDINAL;
diagPoint:CARDINAL;
bootPoint:CARDINAL;
name:CARDINAL;
reserved01:CARDINAL;
reserved02:CARDINAL
END;

```

CONST

```

busWidth= ConfigFlagSet {byteWide,wordWide};
nibbleWide= ConfigFlagSet {};
bootTime= ConfigFlagSet {configTime,bindTime};
never= ConfigFlagSet {};

```

(\* ----- \*)

TYPE

```

ConfigDevFlags =(
shutup,configMe,badMemory,processed,cdf4,cdf5,cdf6,cdf7
);
ConfigDevFlagSet =SET OF ConfigDevFlags ;
ConfigDevPtr =POINTER TO ConfigDev ;
ConfigDev =RECORD
node: Node ;
flags: ConfigDevFlagSet ;
pad:SHORTCARD;

```

```

    rom: ExpansionRom ;
    boardAddr:ADDRESS;
    boardSize:LONGCARD;
    slotAddr:CARDINAL;
    slotSize:CARDINAL;
    driver:ADDRESS;
    nextCD: ConfigDevPtr ;
    unused:ARRAY [0..3] OF LONGINT;
END;

TYPE
    CurrentBinding =RECORD
        configDev: ConfigDevPtr ;
        fileName:ADDRESS;
        productString:ADDRESS;
        toolTypes:ADDRESS;
    END;
    CurrentBindingPtr =POINTER TO CurrentBinding ;

(* ----- *)

TYPE
    BootNode =RECORD
        node: Node ;
        flags:BITSET;
        deviceNode:ADDRESS
    END;
    BootNodePtr =POINTER TO BootNode ;

    ExpansionBaseFlags =(
        clogged,shortmem,badmem,dosflag,kickback33,kickback36,silentStart
        ,startCC0
    );
    ExpansionBaseFlagSet =SET OF ExpansionBaseFlags ;
    ExpansionBase =RECORD
        libNode: Library ;
        flags: ExpansionBaseFlagSet ;
        private01:SHORTCARD;
        private02:LONGCARD;
        private03:LONGCARD;
        private04: CurrentBinding ;
        private05: List ;
        mountList: List ;
    END;
    ExpansionBasePtr =POINTER TO ExpansionBase ;

CONST
    (*
        Möglicher Wert für den Parameter flags von AddBootNode und AddDosNode,
        z.B. AddBootNode(0,LONGSET{startProc},..).
    *)
    startProc=0;

    (*
        Mögliche Rückgabewerte von ConfigBoard
    *)
    errOk=0;

```

---

```
errLastboard=40;  
errNoexpansion=41;  
errNomemory=42;  
errNoboard=43;  
errBadmem=44;
```

TYPE

```
ParameterPkt =RECORD  
  dosName:ADDRESS;  
  execName:ADDRESS;  
  unit:LONGCARD;  
  flags:LONGSET;  
  env: DosEnvec ;
```

END;

```
ParameterPktPtr =POINTER TO ParameterPkt ;
```

END ExpansionD.noimp