

Resources

COLLABORATORS

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

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Resources	1
1.1	Resources	1
1.2	TMP:Modula-2/Resources.def	2

Chapter 1

Resources

1.1 Resources

Prozeduren

AbleICR	AddICRVector	AllocMiscResource
AllocPotBits	AllocUnit	FreeMiscResource
FreePotBits	FreeUnit	GetUnit
GetUnitID	GiveUnit	↔
ObtainBattSemaphore		
ReadBattClock	ReadBattMem	ReadUnitID
ReleaseBattSemaphore	RemICRVector	ResetBattClock
SetICR	WriteBattClock	WriteBattMem
WritePotgo		

Konstanten

amiga	battclockName	↔
battmemAmigaAmnesiaAddr		
battmemAmigaAmnesiaLen	battmemName	↔
battmemScsiHostIdAddr		
battmemScsiHostIdLen	battmemScsiLunsAddr	battmemScsiLunsLen
battmemScsiSyncXferAddr	battmemScsiSyncXferLen	↔
battmemScsiTimeoutAddr		
battmemScsiTimeoutLen	battmemSharedAmnesiaAddr	↔
battmemSharedAmnesiaLen		
ciaaName	ciabName	diskName
drt150rpm	drt37422D2S	dskDmaOff
empty	fsrName	mathIEEEName
miscName	potgoName	

Typ-Deklarationen

CiaResourcePtr	DiscResource	↔
DiscResourceFlagSet		
DiscResourceFlags	DiscResourcePtr	DiscResourceUnit
DiscResourceUnitPtr	FileSysEntry	FileSysResource
MathIEEEResource	MathIEEEResourceFlagSet	↔
MathIEEEResourceFlags		
MathIEEEResourcePtr	MiscResource	MiscResourcePtr
PotgoBitSet	PotgoBits	ResourceTypes

StrPtr

1.2 TMP:Modula-2/Resources.def

```
DEFINITION MODULE Resources; (*$Implementation:=FALSE*)
(* 12-May-1992/cn *)

FROM SYSTEM IMPORT ADDRESS, BPTR, LONGSET;

FROM DosD IMPORT
    BSTR , FileLockPtr ;

FROM ExecD IMPORT vectSize, Interrupt , InterruptPtr , Library , LibraryPtr ,
    List , Message , Node , TaskPtr ;

FROM Hardware IMPORT
    CiaIcrFlags , CiaIcrFlagSet , PotFlags , PotFlagSet ;

IMPORT R;

TYPE
    StrPtr =ADDRESS;

(*
    battclock resource
*)

CONST
    battclockName="battclock.resource";

PROCEDURE  ReadBattClock (base{R.A6}:ADDRESS):LONGCARD; CODE -12;

PROCEDURE  ResetBattClock (base{R.A6}:ADDRESS); CODE -6;

PROCEDURE  WriteBattClock (
    base{R.A6}:ADDRESS;
    time{R.D0}:LONGCARD); CODE -18;

(*
    battmem resource
*)

CONST
    battmemName="battmem.resource";

    battmemAmigaAmnesiaAddr=0;
    battmemAmigaAmnesiaLen=1;

    battmemScsiTimeoutAddr=1;
    battmemScsiTimeoutLen=1;

    battmemScsiLunsAddr=2;
    battmemScsiLunsLen=1;

    battmemSharedAmnesiaAddr=64;
```

```
battmemSharedAmnesiaLen=1;

battmemScsiHostIdAddr=65;
battmemScsiHostIdLen=3;

battmemScsiSyncXferAddr=68;
battmemScsiSyncXferLen=1;

PROCEDURE ObtainBattSemaphore (base{R.A6}:ADDRESS); CODE -6;

PROCEDURE ReadBattMem (
    base{R.A6}:ADDRESS;
    buffer{R.A0}:ADDRESS;
    offset{R.D0}:LONGCARD;
    length{R.D1}:LONGCARD):LONGCARD; CODE -18;

PROCEDURE ReleaseBattSemaphore (base{R.A6}:ADDRESS); CODE -12;

PROCEDURE WriteBattMem (
    base{R.A6}:ADDRESS;
    buffer{R.A0}:ADDRESS;
    offset{R.D0}:LONGCARD;
    length{R.D1}:LONGCARD):LONGCARD; CODE -24;

(*
cia resources
*)

CONST
    ciaaName="ciaa.resource";
    ciabName="ciab.resource";

TYPE
    CiaResourcePtr =ADDRESS;

PROCEDURE AbleICR (
    cia{R.A6}: CiaResourcePtr ;
    mask{R.D0}: CiaIcrFlagSet ): CiaIcrFlagSet ; CODE -18;

PROCEDURE AddICRVector (
    cia{R.A6}: CiaResourcePtr ;
    icrBit{R.D0}: CiaIcrFlags ;
    interrupt{R.A1}: InterruptPtr ): InterruptPtr ; CODE -6;

PROCEDURE RemICRVector (
    cia{R.A6}: CiaResourcePtr ;
    icrBit{R.D0}: CiaIcrFlags ;
    interrupt{R.A1}: InterruptPtr ); CODE -12;

PROCEDURE SetICR (
    cia{R.A6}: CiaResourcePtr ;
    mask{R.D0}: CiaIcrFlagSet ): CiaIcrFlagSet ; CODE -24;

(*
disk resource
*)
```

```
TYPE
  DiscResourceUnit =RECORD
    message: Message ;
    discBlock: Interrupt ;
    discSync: Interrupt ;
    index: Interrupt
  END;
  DiscResourceUnitPtr =POINTER TO DiscResourceUnit ;

  DiscResourceFlags =(
    alloc0,alloc1,alloc2,alloc3,drf4,drf5,drf6,active
  );
  DiscResourceFlagSet =SET OF DiscResourceFlags ;

  DiscResource =RECORD
    library: Library ;
    current: DiscResourceUnitPtr ;
    flags: DiscResourceFlagSet ;
    pad:SHORTCARD;
    sysLib: LibraryPtr ;
    ciaResource: LibraryPtr ;
    unitID:ARRAY [alloc0..alloc3] OF LONGCARD;
    waiting: List ;
    discBlock: Interrupt ;
    discSync: Interrupt ;
    index: Interrupt ;
    currTask: TaskPtr ;
  END;
  DiscResourcePtr =POINTER TO DiscResource ;

CONST
  diskName="disk.resource";

  dskDmaOff=4000H;

  amiga=0;
  drt37422D2S=55555555H;
  drt150rpm=0AAAAAAAH;
  empty=0FFFFFFFH;

PROCEDURE AllocUnit (
  disk{R.A6}: DiscResourcePtr ;
  unitNum{R.D0}:LONGINT):BOOLEAN; CODE -6;

PROCEDURE FreeUnit (
  disk{R.A6}: DiscResourcePtr ;
  unitNum{R.D0}:LONGINT); CODE -12;

PROCEDURE GetUnit (
  disk{R.A6}: DiscResourcePtr ;
  unitPointer{R.A1}: DiscResourceUnitPtr
): DiscResourceUnitPtr ; CODE -18;

PROCEDURE GetUnitID (
  disk{R.A6}: DiscResourcePtr ;
  unitNum{R.D0}:LONGINT):LONGCARD; CODE -30;
```

```
PROCEDURE GiveUnit (disk{R.A6}: DiscResourcePtr ); CODE -24;
```

```
PROCEDURE (*37*) ReadUnitID (
    disk{R.A6}: DiscResourcePtr ;
    unitNum{R.D0}:LONGINT):LONGCARD; CODE -36;
```

```
(*
  filesystem resource
*)
```

```
CONST
  fsrName="FileSystem.resource";
```

```
TYPE
  FileSysResource =RECORD
    node: Node ;
    creator: StrPtr ;
    fileSysEntries: List ;
  END;
```

```
FileSysEntry =RECORD
  node: Node ;
  dosType:LONGCARD;
  version:LONGCARD;
  patchFlags:LONGSET;
  type:LONGCARD;
  task: TaskPtr ;
  lock: FileLockPtr ;
  handler: BSTR ;
  stackSize:LONGCARD;
  priority:LONGINT;
  startup:BPTR;
  segList:BPTR;
  globalVec:BPTR;
  END;
```

```
(*
  mathieee resource
*)
```

```
CONST
  mathIEEEName="MathIEEE.resource";
```

```
TYPE
  MathIEEEResourceFlags =(
    dblbas,dbltrans,sglbas,sgltrans,extbas,exttrans,mirf6,mirf7,
    mirf8,mirf9,mirf10,mirf11,mirf12,mirf13,mirf14,mirf15
  );
  MathIEEEResourceFlagSet =SET OF MathIEEEResourceFlags ;
```

```
MathIEEEResource =RECORD
  node: Node ;
  flags: MathIEEEResourceFlagSet ;
  baseAddr:ADDRESS;
  dblBasInit:PROC;
  dblTransInit:PROC;
```



```
    sglBasInit:PROC;
    sglTransInit:PROC;
    extBasInit:PROC;
    extTransInit:PROC;
END;
    MathIEEEEResourcePtr =POINTER TO    MathIEEEEResource ;

(*
    misc resource
*)

CONST
    miscName="misc.resource";

TYPE
    ResourceTypes =(
        serialPort,serialBits,parallelPort,parallelBits
    );
    MiscResource =RECORD
        library:    Library ;
        allocArray: ARRAY    ResourceTypes    OF ADDRESS;
    END;
    MiscResourcePtr =POINTER TO    MiscResource ;

PROCEDURE    AllocMiscResource (
    misc{R.A6}: MiscResourcePtr ;
    unitNum{R.D0}:LONGINT;
    name{R.A1}: StrPtr ):ADDRESS; CODE -6;

PROCEDURE    FreeMiscResource (
    misc{R.A6}: MiscResourcePtr ;
    unitNum{R.D0}:LONGINT); CODE -12;

(*
    potgo resource
*)

CONST
    potgoName="potgo.resource";

TYPE
    PotgoBits = PotFlags ;
    PotgoBitSet = PotFlagSet ;

PROCEDURE    AllocPotBits (
    potgo{R.A6}:ADDRESS;
    bits{R.D0}: PotgoBitSet ): PotgoBitSet ; CODE -6;

PROCEDURE    FreePotBits (
    potgo{R.A6}:ADDRESS;
    allocated{R.D0}: PotgoBitSet ); CODE -12;

PROCEDURE    WritePotgo (
    potgo{R.A6}:ADDRESS;
    word{R.D0}: PotgoBitSet ;
    mask{R.D1}: PotgoBitSet ); CODE -18;
```

END Resources.noimp