# autodoc

| COLLABORATORS | | | |
|---|---|---|---|
| | *TITLE* :<br><br>autodoc | | |
| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
| WRITTEN BY | | December 18, 2022 | |

| REVISION HISTORY | | | |
|---|---|---|---|
| NUMBER | DATE | DESCRIPTION | NAME |
| | | | |

# Contents

# Chapter 1

# autodoc

## 1.1 Autodoc

```
                  (¯`·._.·’ Help Guide ’·._.·’¯)

  exec.library quick autodocs
  Adjusted for use with Smgmw
  Converted using QACD
  114 functions
Content:


                  AbortIO()

                  AddDevice()

                  AddHead()

                  AddIntServer()

                  AddLibrary()

                  AddMemHandler()

                  AddMemList()

                  AddPort()

                  AddResource()

                  AddSemaphore()

                  AddTail()

                  AddTask()

                  Alert()

                  AllocAbs()

                  Allocate()
```

AllocEntry()

AllocMem()

AllocPooled()

AllocSignal()

AllocTrap()

AllocVec()

AttemptSemaphore()

AttemptSemaphoreShared()

AvailMem()

CacheClearE()

CacheClearU()

CacheControl()

CachePostDMA()

CachePreDMA()

Cause()

CheckIO()

CloseDevice()

CloseLibrary()

ColdReboot()

CopyMem()

CopyMemQuick()

CreateIORequest()

CreateMsgPort()

CreatePool()

Deallocate()

Debug()

DeleteIORequest()

DeleteMsgPort()

DeletePool()

Disable()

DoIO()

Enable()

Enqueue()

FindName()

FindPort()

FindResident()

FindSemaphore()

FindTask()

Forbid()

FreeEntry()

FreeMem()

FreePooled()

FreeSignal()

FreeTrap()

FreeVec()

GetCC()

GetMsg()

InitCode()

InitResident()

InitSemaphore()

InitStruct()

Insert()

MakeFunctions()

MakeLibrary()

ObtainQuickVector()

ObtainSemaphore()

ObtainSemaphoreList()

```
ObtainSemaphoreShared()

OldOpenLibrary()

OpenDevice()

OpenLibrary()

OpenResource()

Permit()

Procure()

PutMsg()

RawDoFmt()

ReleaseSemaphore()

ReleaseSemaphoreList()

RemDevice()

RemHead()

RemIntServer()

RemLibrary()

RemMemHandler()

Remove()

RemPort()

RemResource()

RemSemaphore()

RemTail()

RemTask()

ReplyMsg()

SendIO()

SetExcept()

SetFunction()

SetIntVector()

SetSignal()
```

```
                SetSR()

                SetTaskPri()

                Signal()

                StackSwap()

                SumKickData()

                SumLibrary()

                SuperState()

                Supervisor()

                TypeOfMem()

                UserState()

                Vacate()

                Wait()

                WaitIO()

                WaitPort()
```

## 1.2 AbortIO()

```
NAME     : AbortIO
FUNCTION : attempt to abort an in-progress I/O request
SYNOPSIS : AbortIO(iORequest)
            -480       A1
```

## 1.3 AddDevice()

```
NAME     : AddDevice
FUNCTION : add a device to the system
SYNOPSIS : AddDevice(device)
              -432      A1
```

## 1.4 AddHead()

```
NAME     : AddHead
FUNCTION : insert node at the head of a list
SYNOPSIS : AddHead(list,node)
```

```
            -240    A0   A1
```

## 1.5 AddIntServer()

```
NAME     : AddIntServer
FUNCTION : add an interrupt server to a system server chain
SYNOPSIS : AddIntServer(intNum,interrupt)
               -168       D0       A1
```

## 1.6 AddLibrary()

```
NAME     : AddLibrary
FUNCTION : add a library to the system
SYNOPSIS : AddLibrary(library)
              -396       A1
```

## 1.7 AddMemHandler()

```
NAME     : AddMemHandler
FUNCTION : Add a low memory handler to exec
SYNOPSIS : AddMemHandler(memHandler)
               -774          A1
```

## 1.8 AddMemList()

```
NAME     : AddMemList
FUNCTION : add memory to the system free pool
SYNOPSIS : AddMemList(size,attributes,pri,base,name )
             -618      D0     D1       D2  A0   A1
```

## 1.9 AddPort()

```
NAME     : AddPort
FUNCTION : add a public message port to the system
SYNOPSIS : AddPort(port)
            -354    A1
```

## 1.10 AddResource()

```
NAME     : AddResource
FUNCTION : add a resource to the system
SYNOPSIS : AddResource(resource)
              -486        A1
```

## 1.11 AddSemaphore()

```
NAME     : AddSemaphore
FUNCTION : initialize then add a signal semaphore to the system
SYNOPSIS : AddSemaphore(signalSemaphore)
              -600            A1
```

## 1.12 AddTail()

```
NAME     : AddTail
FUNCTION : append node to tail of a list
SYNOPSIS : AddTail(list,node)
            -246    A0   A1
```

## 1.13 AddTask()

```
NAME     : AddTask
FUNCTION : add a task to the system
SYNOPSIS : AddTask(task,initialPC,finalPC)
            -282    A1     A2       A3
```

## 1.14 Alert()

```
NAME     : Alert
FUNCTION : alert the user of an error
SYNOPSIS : Alert(alertNum)
            -108    D7
```

## 1.15 AllocAbs()

```
NAME     : AllocAbs
FUNCTION : allocate at a given location
SYNOPSIS : memoryBlock=AllocAbs(byteSize,location)
               D0        -204      D0       A1
```

## 1.16 Allocate()

```
NAME     : Allocate
FUNCTION : allocate a block of memory
SYNOPSIS : memoryBlock=Allocate(memHeader,byteSize)
               D0        -186      A0        D0
```

## 1.17 AllocEntry()

```
NAME     : AllocEntry
FUNCTION : allocate many regions of memory
SYNOPSIS : memList=AllocEntry(memList)
             D0       -222      A0
```

## 1.18 AllocMem()

```
NAME     : AllocMem
FUNCTION : allocate memory given certain requirements
SYNOPSIS : memoryBlock=AllocMem(byteSize,attributes)
               D0        -198      D0       D1
```

## 1.19 AllocPooled()

```
NAME     : AllocPooled
FUNCTION : Allocate memory with the pool manager
SYNOPSIS : memory=AllocPooled(poolHeader,memSize)
             d0      -708         a0       d0
```

## 1.20 AllocSignal()

```
NAME     : AllocSignal
FUNCTION : allocate a signal bit
SYNOPSIS : signalNum=AllocSignal(signalNum)
              D0        -330        D0
```

## 1.21 AllocTrap()

```
NAME     : AllocTrap
FUNCTION : allocate a processor trap vector
SYNOPSIS : trapNum=AllocTrap(trapNum)
             D0      -342      D0
```

## 1.22 AllocVec()

```
NAME     : AllocVec
FUNCTION : allocate memory and keep track of the size
SYNOPSIS : memoryBlock=AllocVec(byteSize,attributes)
               D0        -684      D0        D1
```

## 1.23 AttemptSemaphore()

```
NAME     : AttemptSemaphore
FUNCTION : try to obtain without blocking
SYNOPSIS : success=AttemptSemaphore(signalSemaphore)
              D0        -576              A0
```

## 1.24 AttemptSemaphoreShared()

```
NAME     : AttemptSemaphoreShared
FUNCTION : try to obtain without blocking
SYNOPSIS : success=AttemptSemaphoreShared(signalSemaphore)
             D0           -678                 A0
```

## 1.25 AvailMem()

```
NAME     : AvailMem
FUNCTION : memory available given certain requirements
SYNOPSIS : size=AvailMem(attributes)
            D0    -216      D1
```

## 1.26 CacheClearE()

```
NAME     : CacheClearE
FUNCTION : Cache clearing with extended control
SYNOPSIS : CacheClearE(address,length,caches)
              -642       a0      d0     d1
```

## 1.27 CacheClearU()

```
NAME     : CacheClearU
FUNCTION : User callable simple cache clearing
SYNOPSIS : CacheClearU()
              -636
```

## 1.28 CacheControl()

```
NAME     : CacheControl
FUNCTION : Instruction & data cache control
SYNOPSIS : oldBits=CacheControl(cacheBits,cacheMask)
             D0        -648         D0        D1
```

## 1.29 CachePostDMA()

```
NAME     : CachePostDMA
FUNCTION : Take actions after to hardware DMA
SYNOPSIS : CachePostDMA(vaddress,&length,flags)
               -768       a0       a1     d0
```

## 1.30 CachePreDMA()

```
NAME     : CachePreDMA
FUNCTION : Take actions prior to hardware DMA
SYNOPSIS : paddress=CachePreDMA(vaddress,&length,flags)
             d0         -762      a0       a1     d0
```

## 1.31 Cause()

```
NAME     : Cause
FUNCTION : cause a software interrupt
SYNOPSIS : Cause(interrupt)
            -180    A1
```

## 1.32 CheckIO()

```
NAME     : CheckIO
FUNCTION : get the status of an IORequest
SYNOPSIS : result=CheckIO(iORequest)
             D0     -468      A1
```

## 1.33 CloseDevice()

```
NAME     : CloseDevice
FUNCTION : conclude access to a device
SYNOPSIS : CloseDevice(iORequest)
               -450       A1
```

## 1.34 CloseLibrary()

```
NAME     : CloseLibrary
FUNCTION : conclude access to a library
SYNOPSIS : CloseLibrary(library)
                -414        A1
```

## 1.35 ColdReboot()

```
NAME     : ColdReboot
FUNCTION : reboot the Amiga
SYNOPSIS : ColdReboot()
              -726
```

## 1.36 CopyMem()

```
NAME     : CopyMem
FUNCTION : general purpose memory copy function
SYNOPSIS : CopyMem(source,dest,size )
             -624    A0    A1   D0
```

## 1.37 CopyMemQuick()

```
NAME     : CopyMemQuick
FUNCTION : optimized memory copy function
SYNOPSIS : CopyMemQuick(source,dest,size )
               -630       A0    A1   D0
```

## 1.38 CreateIORequest()

```
NAME     : CreateIORequest
FUNCTION : create an IORequest structure
SYNOPSIS : ioReq=CreateIORequest(ioReplyPort,size)
                      -654           A0      D0
```

## 1.39 CreateMsgPort()

```
NAME     : CreateMsgPort
FUNCTION : Allocate and initialize a new message port
SYNOPSIS : CreateMsgPort()
               -666
```

## 1.40 CreatePool()

```
NAME     : CreatePool
FUNCTION : Generate a private memory pool header
SYNOPSIS : newPool=CreatePool(memFlags,puddleSize,threshSize)
             a0      -696        d0        d1         d2
```

## 1.41 Deallocate()

```
NAME     : Deallocate
FUNCTION : deallocate a block of memory
SYNOPSIS : Deallocate(memHeader,memoryBlock,byteSize)
              -192       A0         A1         D0
```

## 1.42 Debug()

```
NAME     : Debug
FUNCTION : run the system debugger
SYNOPSIS : Debug(flags)
            -114   D0
```

## 1.43 DeleteIORequest()

```
NAME     : DeleteIORequest
FUNCTION : Free a request made by CreateIORequest
SYNOPSIS : DeleteIORequest(ioReq)
                -660         A0
```

## 1.44 DeleteMsgPort()

```
NAME     : DeleteMsgPort
FUNCTION : Free a message port created by CreateMsgPort
SYNOPSIS : DeleteMsgPort(msgPort)
                -672       A0
```

## 1.45 DeletePool()

```
NAME     : DeletePool
FUNCTION : Drain an entire memory pool
SYNOPSIS : DeletePool(poolHeader)
              -702        A0
```

## 1.46 Disable()

```
NAME     : Disable
FUNCTION : disable interrupt processing
SYNOPSIS : Disable()
            -120
```

## 1.47 DoIO()

```
NAME     : DoIO
FUNCTION : perform an I/O command and wait for completion
SYNOPSIS : error=DoIO(iORequest)
            D0   -456     A1
```

## 1.48 Enable()

```
NAME     : Enable
FUNCTION : permit system interrupts to resume
SYNOPSIS : Enable()
            -126
```

## 1.49 Enqueue()

```
NAME     : Enqueue
FUNCTION : insert or append node to a system queue
SYNOPSIS : Enqueue(list,node)
            -270    A0   A1
```

## 1.50 FindName()

```
NAME     : FindName
FUNCTION : find a system list node with a given name
SYNOPSIS : node=FindName(start,name)
            D0     -276    A0    A1
```

## 1.51 FindPort()

```
NAME     : FindPort
FUNCTION : find a given system message port
SYNOPSIS : port=FindPort(name)
            D0    -390    A1
```

## 1.52 FindResident()

```
NAME     : FindResident
FUNCTION : find a resident module by name
SYNOPSIS : resident=FindResident(name)
               D0        -96       A1
```

## 1.53 FindSemaphore()

```
NAME     : FindSemaphore
FUNCTION : find a given system signal semaphore
SYNOPSIS : signalSemaphore=FindSemaphore(name)
                 D0            -594       A1
```

## 1.54 FindTask()

```
NAME     : FindTask
FUNCTION : find a task with the given name or find oneself
SYNOPSIS : task=FindTask(name)
            D0    -294    A1
```

## 1.55 Forbid()

```
NAME     : Forbid
FUNCTION : forbid task rescheduling.
SYNOPSIS : Forbid()
            -132
```

## 1.56 FreeEntry()

```
NAME     : FreeEntry
FUNCTION : free many regions of memory
SYNOPSIS : FreeEntry(memList)
              -228      A0
```

## 1.57 FreeMem()

```
NAME     : FreeMem
FUNCTION : deallocate with knowledge
SYNOPSIS : FreeMem(memoryBlock,byteSize)
              -210      A1         D0
```

## 1.58 FreePooled()

```
NAME     : FreePooled
FUNCTION : Free pooled memory
SYNOPSIS : FreePooled(poolHeader,memory,memSize)
               -714       a0        a1     d0
```

## 1.59 FreeSignal()

```
NAME     : FreeSignal
FUNCTION : free a signal bit
SYNOPSIS : FreeSignal(signalNum)
              -336       D0
```

## 1.60 FreeTrap()

```
NAME     : FreeTrap
FUNCTION : free a processor trap
SYNOPSIS : FreeTrap(trapNum)
             -348      D0
```

## 1.61 FreeVec()

```
NAME     : FreeVec
FUNCTION : return AllocVec memory to the system
SYNOPSIS : FreeVec(memoryBlock)
             -690      A1
```

## 1.62 GetCC()

```
NAME     : GetCC
FUNCTION : get condition codes in a 68010 compatible way
SYNOPSIS : conditions=GetCC()
               D0      -528
```

## 1.63 GetMsg()

```
NAME     : GetMsg
FUNCTION : get next message from a message port
SYNOPSIS : message=GetMsg(port)
              D0    -372   A0
```

## 1.64 InitCode()

```
NAME     : InitCode
FUNCTION : initialize resident code modules
SYNOPSIS : InitCode(startClass,version)
             -72        D0        D1
```

## 1.65 InitResident()

```
NAME     : InitResident
FUNCTION : initialize resident module
SYNOPSIS : object=InitResident(resident,segList)
             D0      -102         A1       D1
```

## 1.66 InitSemaphore()

```
NAME     : InitSemaphore
FUNCTION : initialize a signal semaphore
SYNOPSIS : InitSemaphore(signalSemaphore)
               -558            A0
```

## 1.67 InitStruct()

```
NAME     : InitStruct
FUNCTION : initialize memory from a table
SYNOPSIS : InitStruct(initTable,memory,size)
              -78         A1      A2    D0
```

## 1.68 Insert()

```
NAME     : Insert
FUNCTION : insert a node into a list
SYNOPSIS : Insert(list,node,listNode)
            -234   A0   A1     A2
```

## 1.69 MakeFunctions()

```
NAME     : MakeFunctions
FUNCTION : construct a function jump table
SYNOPSIS : tableSize=MakeFunctions(target,functionArray,funcDispBase)
               D0        -90         A0        A1           A2
```

## 1.70 MakeLibrary()

```
NAME     : MakeLibrary
FUNCTION : construct a library
SYNOPSIS : library=MakeLibrary(vectors,structure,init,dSize,segList)
              D0       -84       A0        A1     A2    D0    D1
```

## 1.71 ObtainQuickVector()

```
NAME     : ObtainQuickVector
FUNCTION : Function to obtain an install a Quick Interrupt vector
SYNOPSIS : vector=ObtainQuickVector(interruptCode)
             D0         -786             A0
```

## 1.72 ObtainSemaphore()

```
NAME     : ObtainSemaphore
FUNCTION : gain exclusive access to a semaphore
SYNOPSIS : ObtainSemaphore(signalSemaphore)
                -564              A0
```

## 1.73 ObtainSemaphoreList()

```
NAME     : ObtainSemaphoreList
FUNCTION : get a list of semaphores
SYNOPSIS : ObtainSemaphoreList(list)
                 -582           A0
```

## 1.74 ObtainSemaphoreShared()

```
NAME     : ObtainSemaphoreShared
FUNCTION : gain shared access to a semaphore
SYNOPSIS : ObtainSemaphoreShared(signalSemaphore)
                    -678               A0
```

## 1.75 OldOpenLibrary()

```
NAME     : OldOpenLibrary
FUNCTION : obsolete OpenLibrary
SYNOPSIS : library=OldOpenLibrary(libName)
              D0       -408          A1
```

## 1.76 OpenDevice()

```
NAME     : OpenDevice
FUNCTION : gain access to a device
SYNOPSIS : error=OpenDevice(devName,unitNumber,iORequest,flags)
             D0     -444       A0       D0         A1      D1
```

## 1.77 OpenLibrary()

```
NAME     : OpenLibrary
FUNCTION : gain access to a library
SYNOPSIS : library=OpenLibrary(libName, version)
              D0       -552       A1       D0
```

## 1.78 OpenResource()

```
NAME     : OpenResource
FUNCTION : gain access to a resource
SYNOPSIS : resource=OpenResource(resName)
              D0        -498        A1
```

## 1.79 Permit()

```
NAME     : Permit
FUNCTION : permit task rescheduling
SYNOPSIS : Permit()
            -138
```

## 1.80 Procure()

```
NAME     : Procure
FUNCTION : bid for a semaphore
SYNOPSIS : Procure(semaphore,bidMessage)
             -540     A0         A1
```

## 1.81 PutMsg()

```
NAME     : PutMsg
FUNCTION : put a message to a message port
SYNOPSIS : PutMsg(port,message)
            -366   A0     A1
```

## 1.82 RawDoFmt()

```
NAME     : RawDoFmt
FUNCTION : format data into a character stream
SYNOPSIS : NextData=RawDoFmt(FormatString,DataStream,PutChProc,PutChData)
              D0      -522        A0          A1        A2        A3
```

## 1.83 ReleaseSemaphore()

```
NAME     : ReleaseSemaphore
FUNCTION : make signal semaphore available to others
SYNOPSIS : ReleaseSemaphore(signalSemaphore)
                  -570             A0
```

## 1.84 ReleaseSemaphoreList()

```
NAME     : ReleaseSemaphoreList
FUNCTION : make a list of semaphores available
SYNOPSIS : ReleaseSemaphoreList(list)
                   -588           A0
```

## 1.85 RemDevice()

```
NAME     : RemDevice
FUNCTION : remove a device from the system
SYNOPSIS : RemDevice(device)
               -438     A1
```

## 1.86 RemHead()

```
NAME     : RemHead
FUNCTION : remove the head node from a list
SYNOPSIS : node=RemHead(list)
            D0    -258   A0
```

## 1.87 RemIntServer()

```
NAME     : RemIntServer
FUNCTION : remove an interrupt server from a server chain
SYNOPSIS : RemIntServer(intNum,interrupt)
               -174       D0       A1
```

## 1.88 RemLibrary()

```
NAME     : RemLibrary
FUNCTION : remove a library from the system
SYNOPSIS : RemLibrary(library)
              -402       A1
```

## 1.89 RemMemHandler()

```
NAME     : RemMemHandler
FUNCTION : Remove low memory handler from exec
SYNOPSIS : RemMemHandler(memHandler)
                -780         A1
```

## 1.90 Remove()

```
NAME     : Remove
FUNCTION : remove a node from a list
SYNOPSIS : Remove(node)
            -252   A1
```

## 1.91 RemPort()

```
NAME     : RemPort
FUNCTION : remove a message port from the system
SYNOPSIS : RemPort(port)
             -360   A1
```

## 1.92 RemResource()

```
NAME     : RemResource
FUNCTION : remove a resource from the system
SYNOPSIS : RemResource(resource)
              -492        A1
```

## 1.93 RemSemaphore()

```
NAME     : RemSemaphore
FUNCTION : remove a signal semaphore from the system
SYNOPSIS : RemSemaphore(signalSemaphore)
               -606          A1
```

## 1.94 RemTail()

```
NAME     : RemTail
FUNCTION : remove the tail node from a list
SYNOPSIS : node=RemTail(list)
            D0    -264   A0
```

## 1.95 RemTask()

```
NAME     : RemTask
FUNCTION : remove a task from the system
SYNOPSIS : RemTask(task)
             -288   A1
```

## 1.96 ReplyMsg()

```
NAME     : ReplyMsg
FUNCTION : put a message to its reply port
SYNOPSIS : ReplyMsg(message)
             -378      A1
```

## 1.97 SendIO()

```
NAME     : SendIO
FUNCTION : initiate an I/O command
SYNOPSIS : SendIO(iORequest)
            -462      A1
```

## 1.98 SetExcept()

```
NAME     : SetExcept
FUNCTION : define certain signals to cause exceptions
SYNOPSIS : oldSignals=SetExcept(newSignals,signalMask)
               D0        -312      D0          D1
```

## 1.99 SetFunction()

```
NAME     : SetFunction
FUNCTION : change a function vector in a library
SYNOPSIS : oldFunc=SetFunction(library,funcOffset,funcEntry)
             D0       -420        A1       A0        D0
```

## 1.100 SetIntVector()

```
NAME     : SetIntVector
FUNCTION : set a new handler for a system interrupt vector
SYNOPSIS : oldInterrupt=SetIntVector(intNumber,interrupt)
               D0           -162        D0         A1
```

## 1.101 SetSignal()

```
NAME     : SetSignal
FUNCTION : define the state of this task's signals
SYNOPSIS : oldSignals=SetSignal(newSignals,signalMask)
              D0         -306      D0          D1
```

## 1.102 SetSR()

```
NAME     : SetSR
FUNCTION : get and/or set processor status register
SYNOPSIS : oldSR=SetSR(newSR, mask)
             D0   -144   D0    D1
```

## 1.103 SetTaskPri()

```
NAME     : SetTaskPri
FUNCTION : get and set the priority of a task
SYNOPSIS : oldPriority=SetTaskPri(task, priority)
               D0         -300     A1      D0
```

## 1.104 Signal()

```
NAME     : Signal
FUNCTION : signal a task
SYNOPSIS : Signal(task,signals)
            -324   A1    D0
```

## 1.105 StackSwap()

```
NAME     : StackSwap
FUNCTION : EXEC supported method of replacing task's stack
SYNOPSIS : StackSwap(newStack)
              -732      A0
```

## 1.106 SumKickData()

```
NAME     : SumKickData
FUNCTION : compute the checksum for the Kickstart delta list
SYNOPSIS : checksum=SumKickData()
              D0       -612
```

## 1.107 SumLibrary()

```
NAME     : SumLibrary
FUNCTION : compute and check the checksum on a library
SYNOPSIS : SumLibrary(library)
              -426       A1
```

## 1.108 SuperState()

```
NAME     : SuperState
FUNCTION : enter supervisor state with user stack
SYNOPSIS : oldSysStack=SuperState()
               D0         -150
```

## 1.109 Supervisor()

```
NAME     : Supervisor
FUNCTION : trap to a short supervisor mode function
SYNOPSIS : result=Supervisor(userFunc)
             Rx      -30        A5
```

## 1.110 TypeOfMem()

```
NAME     : TypeOfMem
FUNCTION : determine attributes of a given memory address
SYNOPSIS : attributes=TypeOfMem(address)
               D0        -534     A1
```

## 1.111 UserState()

```
NAME     : UserState
FUNCTION : return to user state with user stack
SYNOPSIS : UserState(sysStack)
              -156      D0
```

## 1.112 Vacate()

```
NAME     : Vacate
FUNCTION : release a bitMessage from Procure
SYNOPSIS : Vacate(semaphore,bidMessage)
            -546      A0        A1
```

## 1.113 Wait()

```
NAME     : Wait
FUNCTION : wait for one or more signals
SYNOPSIS : signals=Wait(signalSet)
             D0    -318     D0
```

## 1.114 WaitIO()

```
NAME     : WaitIO
FUNCTION : wait for completion of an I/O request
SYNOPSIS : error=WaitIO(iORequest)
            D0    -474     A1
```

## 1.115 WaitPort()

```
NAME     : WaitPort
FUNCTION : wait for a given port to be non-empty
SYNOPSIS : message=WaitPort(port)
             D0      -384    A0
```