

```

//
// module      : freefall.prg
// project     : freefall tower
// status      : under construction , not yet finished
// language    : xBase++ & Clipper 5.2
// author      : Rene Nootboom
// email       : r.nootboom2@chello.nl
// website     : http://members.chello.nl/r.nootboom2
// created     : 02/05/2006 12:15
// updated     : 23/09/2006 10:17
// ports       : freefall tower uses one FtCOM(30402) interface
//              and one LPT-MultiFace 72 simultaneously
// comment     : if ftcom.ini exists it is searched for entries :
//              COMPORT_WIN32 = <n> for xBase++(win32)
//              COM<n>_IO = <hex> (h3F8) for clipper(dos)
//
#ifdef __XPP__ // xbase++ compiler detected
#include „common.ch“
#else // clipper 5.2 compiler used
#include „generic.ch“
#endif
#include „inkey.ch“
#include „noot.ch“
#include „ft.ch“
#include „freefall.ch“
//
Static nComPort:=-1
//
Procedure Main()
LOCAL nkey,ckey

#ifdef __XPP__
SetMouse(FALSE)
Set Scoreboard off
#endif

CLS
If (nComPort := InitCOM()) < 1
ALARMTONE
Alert(„Main(): error initializing COM port“)
CloseApplication()
EndIf

? „this is a test for the linkmotor“
? „key F : linkmotor in freefall position“
? „key C : linkmotor in fixed(Coupled) postion“

```

```

? „press escape to quit“
Do While !(LastKey() == K_ESC)
  nkey := Inkey(0)
  ckey := Upper(Chr(nkey))
  Do Case
    Case ckey == „F“
      FreeLink()
    Case ckey == „C“
      FixLink()
  EndCase
EndDo

CloseApplication()
Return
//
Procedure CloseApplication()
  FtCCloseAll()
  Quit
Return
//
Function InitCOM()
  LOCAL cSearch,xComPort,nPort,lAskValues

#ifdef __XPP__
  LOCAL i,xResult
#endif

//
// check for entries in an optional file : ftcom.ini
//
If File(„ftcom.ini“)
  readinival(„ftcom.ini“)

#ifdef __XPP__ // WINNT & W9x family: only check for COMPORT #
  cSearch := „COMPORT_WIN32“
  xComPort := ini_value(cSearch)
  xComPort := IIf(!IS_NUM(xComPort),-1,xComPort)
#else // DOS-family : COM<1-4> and get IO-address
  xComPort := NIL
  For i:=1 to 4 // search for a line with COM<n>_IO = <hFFF>
    cSearch := „COM“ + Str(i,1,0) + „_IO“
    If !((xResult := ini_value(cSearch)) == NIL)
      xComPort := i
      lAskValues := empty(xResult)      Exit
    EndIf
  Next i

```

```
#endif
  clearinival()
EndIf

  nPort := FtCInit(xComPort,lAskValues)

Return nPort // return -1 if error
//
//
Procedure FixLink() // to attach the hoistmotor
  If !FtCSensor(nComPort,LINKED)
    FtCMotorLeft(nComPort,LINKMOTOR)
    FtCButton(nComPort,LINKED)
    FtCMotorOff(nComPort,LINKMOTOR)
  EndIf
  Return
//
//
Procedure FreeLink() // to decouple the hoistmotor from the load
  If !FtCSensor(nComPort,FREEFALL)
    FtCMotorRight(nComPort,LINKMOTOR)
    FtCButton(nComPort,FREEFALL)
    FtCMotorOff(nComPort,LINKMOTOR)
  EndIf
  Return
//
// eof freefall.prg
//
```