Documentation coscomekg.dll version 1.0.0

Function overview

The return value of all functions is a error indicator described as follows:
1: Successful operation
0: No Connection was assigned to device
-1: Unsupported function at this device
-2: Connection is interrupted
-3: Exception
-4: Wrong port number was used

A coscomopenPort

int coscomopenPort(int portnumber);
function coscomopenPort(portnumber : Integer) : Integer;
connects device with COM Port 1..n

B coscomclosePort

int coscomclosePort(int portnumber);
function coscomclosePort(portnumber : Integer) : Integer;
closes connection at COM Port 1..n

C coscomgetVersion

int coscomgetVersion(int portnumber;char * versionchars);
function coscomgetVersion(portnumber : Integer;versionchars : PCHAR) : Integer;
versionchars: 3 x Bytes (ASCII Chars)
Release.Version * 100

D coscomgetStatus

int coscomgetStatus(int portnumber;int *value);
function coscomgetStatus(portnumber : Integer;value : PInteger) : Integer;
value:
0 Stop
1 Run
2 Pause

E coscomgetError

int coscomgetError(int portnumber;int *value);
function coscomgetError(portnumber : Integer;value : PInteger) : Integer;
value = Error Number:
1 Oil Interval Exceeded
2 Service Interval Exceeded
21 Elevator Error (Treadmill)
30 Drive Error (Treadmill)
50 Drive Error (Treadmill)

F coscomgetDuration

int coscomgetDuration(int portnumber;int *value);
function coscomgetDuration(portnumber : Integer;value : PInteger) : Integer;
value: duration in seconds (since last start command)

G coscomgetSpeed

int coscomgetSpeed(int portnumber;double *value);
function coscomgetSpeed(portnumber : Integer;value : PDouble) : Integer;
value: actual speed in meter per seconds

H coscomgetAcceleration

int coscomgetAcceleration(int portnumber;int *value);
function coscomgetAcceleration(portnumber : Integer;value : PInteger) : Integer;
value: actual accelerationindex
0 None
1 = 131 s
2 = 66 s
3 = 33 s
4 = 16 s
5 = 8 s
6 = 5 s
7 = 3 s

The selected time relates to the duration of a speed change from standstill to maximum speed or vice versa. The exact value is calculated as follows: a = v / t = maximum speed / acceleration time. The new acceleration value is used with the next "S02" command.
Example: Acceleration index: 3 = 33 s
Maximum treadmill speed: 22 km/h = 6.11 m/s
Acceleration value: a = (6.11 / 33) m/s² = 0.185 m/s²

I coscomgetElevation

int coscomgetElevation(int portnumber;double *value);
function coscomgetElevation(portnumber : Integer;value : PDouble) : Integer;
value: actual elevation in %

J coscomgetHeartrate

int coscomgetHeartrate(int portnumber;int *value);
function coscomgetHeartrate(portnumber : Integer;value : PInteger) : Integer;
value: actual heart frequency in beats per minute

K coscomgetDistance

int coscomgetDistance(int portnumber;int *value);
function coscomgetDistance(portnumber : Integer;value : PInteger) : Integer;
value: actual distance in meter (since last start command)

L coscomgetFailSafe

int coscomgetFailSafe(int portnumber;int *value);
function coscomgetFailSafe(portnumber : Integer;value : PInteger) : Integer;
Reads communication timeout for failsafe (STOP)
value:
0: Disable Failsafe (Default) 1 - 250 1/10 seconds (= 0.1 - 25.0 seconds):

M coscomsetFailSafe

int coscomsetFailSafe(int portnumber;int value);
function coscomsetFailSafe(portnumber : Integer;value : Integer) : Integer;
Enables communication timeout for failsafe (STOP)
value:
0: Disable Failsafe (Default)
1 - 250 1/10 seconds (= 0.1 - 25.0 seconds):

N coscomsetSpeed

int coscomsetSpeed(int portnumber;value : Double);
function coscomsetSpeed(portnumber : Integer;value : Double) : Integer;
value: speed in meter per seconds

O coscomsetAcceleration

int coscomsetAcceleration(int portnumber;int value);
function coscomsetAcceleration(portnumber : Integer;value : Integer) : Integer;
value: acceleration in acceleration indices (see coscomgetAcceleration)
Only activated with new coscomsetSpeed function!

P coscomsetElevation

int coscomsetElevation(int portnumber;double value);
function coscomsetElevation(portnumber : Integer;value : Double) : Integer;
value: elevation in %

Q coscomsetSpeedAccel

int coscomsetSpeedAccel(int portnumber;double value;int accelerationindex);
function coscomsetSpeedAccel(portnumber : Integer;value : Double;accelerationindex : Integer) : Integer;
values: speed in meter per seconds
acceleration in acceleration indices (see coscomgetAcceleration)

R coscomKeyPressStart

int coscomKeyPressStart(int portnumber;int how);
function coscomKeyPressStart(portnumber : Integer;how : Integer) : Integer;
how:
1: key pressed
0: key released

S coscomKeyPressStop

int coscomKeyPressStop(int portnumber;int how);
function coscomKeyPressStop(portnumber : Integer;how : Integer) : Integer;
how:
1: key pressed
0: key released

T coscomKeyPressPlus

int coscomKeyPressPlus(int portnumber;int how);
function coscomKeyPressPlus(portnumber : Integer;how : Integer) : Integer;
how:
1: key pressed
0: key released

U coscomKeyPressMinus

int coscomKeyPressMinus(int portnumber;int how);
function coscomKeyPressMinus(portnumber : Integer;how : Integer) : Integer;
how:
1: key pressed
0: key released

V coscomKeyPressUp

int coscomKeyPressUp(int portnumber;int how);
function coscomKeyPressUp(portnumber : Integer;how : Integer) : Integer;
how:
1: key pressed
0: key released

W coscomKeyPressDown

int coscomKeyPressDown(int portnumber;int how);
function coscomKeyPressDown(portnumber : Integer;how : Integer) : Integer;
how:
1: key pressed
0: key released

X coscomsetProtocol

int coscomsetProtocol(int portnumber;int how);
function coscomsetProtocol(portnumber : Integer;how : Integer) : Integer;
Activates communication protocol.
how:
1 On
2 Off

Y coscomgetProtocol

int coscomgetProtocol(int portnumber;int *how);
function coscomgetProtocol(portnumber : Integer;how : PInteger) : Integer;
Activates communication protocol.
how:
1 On
2 Off

Z How to use coscomekg.dll

Delphi

Make sure that the coscomekg.dll is in your application path.
Add coscomdll.pas to your project.
Sample code open communication:

```
procedure TTestcoscomdllForm.ActionOpenDeviceExecute(Sender: TObject);
var
  aportnumber : Integer;
begin
    closeDevice;
    try
      aportnumber := StrToInt(Edit1.Text);
    except
        exit;
    end;
    if coscomopenPort(aportnumber) > 0 then
      FPortNumber := aportnumber;
end;
```

Sample code close communication:

```
procedure TTestcoscomdllForm.closeDevice;
begin
    if (FPortNumber > 0) and (coscomclosePort(FPortNumber) > 0) then
      FPortNumber := 0;
end;
```

Sample code read status

```
procedure TTestcoscomdllForm.ActionGetStatusExecute(Sender: TObject);
var
  myvalue,arowindex : Integer;
begin
    arowindex := 1;
    FLastError := coscomgetStatus(FPortNumber,@myvalue);
    if FLastError > 0 then
    begin
        with StringGrid1 do
        begin
            Cells[0,FixedRows + arowindex] := 'Status';
```

```
            Cells[1,FixedRows + arowindex] := Format('%d',[myvalue]);
        end;
    end
    else
    begin
        with StringGrid1 do
        begin
            Cells[0,FixedRows + arowindex] := 'Fail Safe';
            case FLastError of
            0: Cells[1,FixedRows + arowindex] := 'Connection Off';
            -1: Cells[1,FixedRows + arowindex] := 'Function unsupported';
            -2: Cells[1,FixedRows + arowindex] := 'Connection interrupted';
            -3: Cells[1,FixedRows + arowindex] := 'Exception';
            -4: Cells[1,FixedRows + arowindex] := 'wrong port number';
            end;
        end;
    end;
end;
```

Sample code set speed

```
procedure TTestcoscomdllForm.ActionGetSpeedExecute(Sender: TObject);
var
  myvalue : Double;
begin
    if coscomgetSpeed(FPortNumber,@myvalue) > 0 then
    begin
        with StringGrid1 do
        begin
            Cells[0,FixedRows + 2] := 'Speed';
            Cells[1,FixedRows + 2] := FormatFloat('0.0',myvalue);
        end;
    end
    else
    begin
        with StringGrid1 do
        begin
            Cells[0,FixedRows + 2] := 'Speed';
            Cells[1,FixedRows + 2] := 'Error';
        end;
    end;
end;
```

Sample code key functions

```
procedure TTestcoscomdllForm.Button1MouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
    coscomKeyPressPlus(FPortNumber,1);
end;
procedure TTestcoscomdllForm.Button1MouseUp(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
```

```
begin
    coscomKeyPressPlus(FPortNumber,0);
end;
```

C/ C++

Make sure that the coscomekg.dll is in your application path.
Add coscomdll.cpp to your project.
Use multithread options to compile your project!
Make sure loadcoscomEKGDLL() and unloadcoscomEKGDLL() is invoked only once. Do not use the instance initialization or clean up functions of the application!
Sample C++ Builder:

```
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
   try
   {
      Application->Initialize();
      loadcoscomEKGDLL();
      Application->CreateForm(__classid(Tformborlandtestegk), &formborlandtestegk);
      Application->Run();
   }
   catch (Exception &exception)
   {
      Application->ShowException(&exception);
   }
   unloadcoscomEKGDLL();
   return 0;
}
```

Sample Visual C++:

```
BOOL CMfctestcoscomdllApp::InitApplication()
{
      // TODO: Speziellen Code hier einfügen und/oder Basisklasse aufrufen
      loadcoscomEKGDLL();
      return CWinApp::InitApplication();
}
void CMfctestcoscomdllApp::OnFinalRelease()
{
      // TODO: Speziellen Code hier einfügen und/oder Basisklasse aufrufen
      unloadcoscomEKGDLL();
      CWinApp::OnFinalRelease();
}
```

Sample code open communication:

```
void __fastcall Tformborlandtestegk::OpenDevice1Click(TObject *Sender)
{
  char StrComPort[20];

   int Size = Edit1->GetTextLen();
```

```
    Edit1->GetTextBuf(StrComPort,Size+1);
    int aportnumber = atoi(StrComPort);
       closeDevice();
       if (coscomopenPort(aportnumber) > 0 )
       {
             FPortNumber = aportnumber;
       }
}
Sample code close communication:
void Tformborlandtestegk::closeDevice()
{
    if ((FPortNumber > 0) && (coscomclosePort(FPortNumber) > 0))
        FPortNumber = 0;
}
Sample code read status
void __fastcall Tformborlandtestegk::ReadStatus1Click(TObject *Sender)
{
    int myValue;
    int arowindex = 1;
    StringGrid1->Cells[0][StringGrid1->FixedRows + arowindex] = "Status";
    FLastError = coscomgetStatus(FPortNumber,&myValue);
    if (FLastError > 0)
    {
       StringGrid1->Cells[1][StringGrid1->FixedRows + arowindex] = FormatFloat("0",myValue);
     }
     else
     {
       switch (FLastError) {
       case 0: StringGrid1->Cells[1][StringGrid1->FixedRows + arowindex] = "Connection Off"; break;
       case -1: StringGrid1->Cells[1][StringGrid1->FixedRows + arowindex] = "Function unsupported"; break;
       case -2: StringGrid1->Cells[1][StringGrid1->FixedRows + arowindex] = "Connection interrupted"; break;
       case -3: StringGrid1->Cells[1][StringGrid1->FixedRows + arowindex] = "Exception"; break;
       case -4: StringGrid1->Cells[1][StringGrid1->FixedRows + arowindex] = "wrong Portnumber"; break;
       }
     }

}
Sample code set speed
void __fastcall Tformborlandtestegk::SetSpeed1Click(TObject *Sender)
{
   double myValue;
    try
    {
       myValue = StrToFloat(StringGrid2->Cells[1][StringGrid2->FixedRows]);
       if (coscomsetSpeed(FPortNumber,myValue) > 0)
          StringGrid2->Cells[0][StringGrid2->FixedRows] = "Speed Ok";
       else
          StringGrid2->Cells[0][StringGrid2->FixedRows] = "Speed failed";
```

```
    }
    catch(...)
    {
          StringGrid2->Cells[0][StringGrid2->FixedRows] = "Speed failed";
    }
}
```

Sample code key functions

```
void __fastcall Tformborlandtestegk::Button1MouseDown(TObject *Sender,
      TMouseButton Button, TShiftState Shift, int X, int Y)
{
    coscomKeyPressMinus(FPortNumber,1);
}
//---------------------------------------------------------------------------
void __fastcall Tformborlandtestegk::Button1MouseUp(TObject *Sender,
      TMouseButton Button, TShiftState Shift, int X, int Y)
{
    coscomKeyPressMinus(FPortNumber,0);
}
```