StoreX Function Library II

Middleware for StoreX Integration

Erstellt		Version	
Geprüft		Projekt Nummer	
Freigegeben		Bemerkungen	

Table of Content

1 Introduction

1.1 Definitions, Acronyms and Abbreviations

BCR	Barcode Reader
CU	Control Unit
GUI	Graphical User Interface
HW	Hardware
SW	Software
FW	Firmware
MPC	Micro Processor Controller
PLC	Programmable Logic Controller
DLL	Dynamic Link Library
PRD	Product Requirements Document
RoMA	Robotic Manipulator Arm
SCR	Software Change Request
SOP	Standard Operating Procedure
TC	Test Case
OS	Operation System
IDE	Integrated Development Environment

1.2 References

STX User Manual
STX Remote Operation
STX_V7_Doc 6

2 Requirements

2.1 Specifications

There are the following specifications related to the STX_lib.dll driver-file.

The STX_lib.dll is a communication library that provides an interaction between custom software and devices of the StoreX family using the serial communication interface RS-232.

The STX_lib.dll library contains all necessary functions of Environment, Plate Operations, Error Diagnostic and periphery control. All parameters and returning values of each function are set of characters. The STX_lib.dll library doesn't contain any COM-objects and doesn't require registration in the Operation System.

2.2 Assumptions / Limitations

For using the STX_lib driver is necessary to have:

Operation System of Microsoft Windows family (Windows 95, 98, 2000, Xp)
Serial Communication Port
Programming language (IDE) which support DLL extension

3 General Interfacing Aspects

3.1 Low Level Interfacing

The low level interfacing is the most direct way to communicate with the system. The low level interfacing will offer the most complete and powerful tool for integration. However, this method is the most complex. When interfacing on this level profound knowledge of then interface and of the host operating system is a prerequisite.

For all most commonly used integration application LiCONiC offers high level integration tools such as ActiveX libraries or DLL libraries or even stand-alone applications that may be run in parallel the user's interface.

We highly recommend to make use of high level interfaces whenever possible. Our high level interfaces offer functions that will allow implementation of low-level accesses. This allows the implementation of dedicated supplementary functions when needed.

3.2 Legacy Interfacing

String Based

Strictly Polling

Ideal for simple Integration platforms (e.g. Intermediate Interface Hardware)

3.3 General Information about Inputs parameters

All input parameters and output results are sets of characters.

All functions have at least one parameter: number of a serial port.

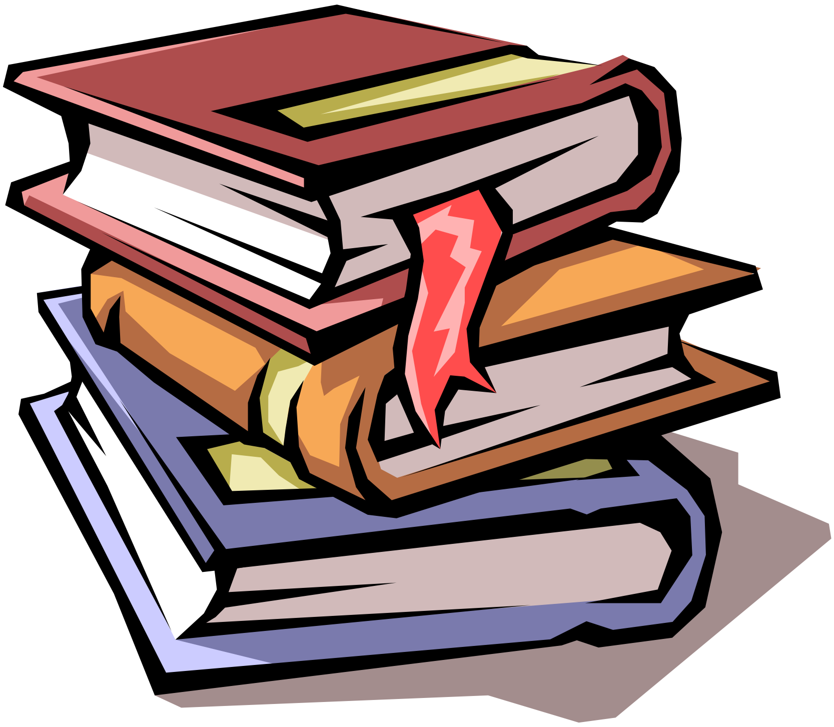
For correct using the functions, clients software have to use the following types of Strings:

IDE	String Type
MS Visual C++	char*
MS Visual Basic	String
MS C#	String
Borland C++ Builder	char*
Borland Delphi	PChar

Action	Command String (Prameter)	Response String (Result)
calling a dll's function: *STXReadBackStr*	*RD DM200*	*'0'* or *'1'*

Definition of procedures or functions are specified with calling convention directive - *"stdcall"*.

4 Command Set



4.1 Initialisation Functions

4.1.1 STXConnect(PChar PortNumber)

Sets the name of serial port and Opens Serial Communication for the device.

Parameter: PortNumber – Number of a Serial Port (e.g. “1”)

Return values (PChar):

```
"1" - Serial Port is successfully opened.

"0" - Error of opening Serial Port.

"2" - Error of opening Serial Port (serial Port is already opened).
```

Examples:

4.1.1.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXConnect Lib "STX_lib" (ByVal pNbr As
String)
As String
```

'Execution:

```
Dim STXId As String

STXId = "1"

If STXConnect(STXId) = "1" Then

  LabelStatus.Text = "Serial Port is opened"

Else

  LabelStatus.Text = "Error"

End If
```

4.1.1.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXConnect(pNbr String);
```

//Execution:

```
if (STXConnect ("1") != "1")
{
  LabelStatus.Text = "Error of opening Serial Port!";
}
```

4.1.1.3 Borland Delphi 6

//Declaration:

```
function STXConnect(PName : PChar) : PChar; stdcall; external
'STX_lib.dll';
```

//Execution:

```
var
connect : integer;
begin

  connect := StrToInt(STXConnect(‘3’));

  case connect of
    0: ShowMessage('Error of opening Serial Port');
    1: ShowMessage('Serial Port is successfully opened');
    2: ShowMessage('Error of opening Serial Port');
  end;

end;
```

4.1.1.4 Borland C++Builder

//Declaration:

```
//Prototypes *.h file
HINSTANCE dllInstance;
char* pNbr;

typedef char* (__stdcall * Connect)(char* PortNumber);
Connect STXConnect;


//Initialization (constructor) *.cpp file

dllInstance = LoadLibrary("STX_init.dll");

if(dllInstance)
{
  STXConnect = (Connect)GetProcAddress(dllInstance, "STXConnect");
}
else
ShowMessage("Error of Library loading!");
```

//Execution:

```
pNbr = "3"; //COM3

if(String(STXConnect(pNbr)) == "1")
{
  if(String(STXOpenCommunication(pNbr)) == "1")
  {
    if (String(STXInitialize(pNbr)) == "1")
    {
       Caption = "StoreX S/N: "+String(STXSerialNumber(pNbr));
    }
    else
    {
       Application->MessageBox("Initialization Error!", "Warning!",
       MB_OK+MB_ICONERROR);
    }
  }
  else
  {
    Application->MessageBox("Communication Error!", "Warning!",
    MB_OK+MB_ICONERROR);
  }
}
else
{
  Application->MessageBox("Failed to open the Serial Port!",
  "Warning!", MB_OK+MB_ICONERROR);
}
```

4.1.2 STXOpenCommunication(PChar PortNumber)

Sends Communication Request command ("CR") to the device.

Parameter: PortNumber – Number of a Serial Port.

Return values (PChar):

```
"0" - No communication.

"1" - Communication is opened.

"2" - Communication Error.
```

Examples:

4.1.2.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXOpenCommunication Lib "STX_lib" (ByVal
pNbr As String) As String
```

'Execution:

```
If STXOpenCommunication(STXId) = "1" Then

  LabelStatus.Text = "Communication is opened"

End If
```

4.1.2.2 Visual C# .NET

//Declaration:

```
using System.Runtime.InteropServices;

[DllImport("STX_lib")]
private extern static String STXOpenCommunication(String pNbr);
```

//Execution:

```
if (STXOpenCommunication(STXId) == "1")
{
  STXInitialize(STXId);
{
```

4.1.2.3 Borland Delphi 6

//Declaration:

```
function STXOpenCommunication(PName : String): PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
var
  result : integer;
begin

  result := StrToInt(STXOpenCommunication(pNbr));

  case result of
  0: ShowMessage('No communication!');
  1: ShowMessage('Communication is opened!');
  2: ShowMessage('Communication Error!');
  end;

end;
```

4.1.2.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * OpenCommunication)(char* PNbr);
OpenCommunication   STXOpenCommunication;


//Initialization (constructor) *.cpp file

STXOpenCommunication = (OpenCommunication)GetProcAddress(dllInstance,
"STXOpenCommunication");
```

//Execution:

```
STXOpenCommunication(PNbr);
```

4.1.3 STXInitialize(PChar PortNumber)

Initializes the StoreX (opesn the PLC connection, initialiszes the handling, reads StoreX system constants).

Parameter: PortNumber – Number of a Serial Port.

Return values (PChar):

```
1 - Device is successfully initialised.

0 - Error of initialisation. User must poll the Ready flag to
    determine whether the command has been completed.
```

Examples:

4.1.3.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXInitialize Lib "STX_lib" (ByVal pNbr As
String) As String
```

'Execution:

```
If STXConnect(STXId) = "1" Then

  If STXOpenCommunication(STXId) = "1" Then

        If STXInitialize(STXId) = "1" Then

        Text = "STX S/N-" + STXSerialNumber(STXId)

        End If

     End If

End If
```

4.1.3.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXInitialize(String pNbr);
```

//Execution:

```
if (STXConnect(textBox1.Text) == "1")
{

  STXId = textBox1.Text;

  if (STXOpenCommunication(STXId) == "1")
  {

    if (STXInitialize(STXId) != "1")
    {

       labelStatus.Text = "Initialisation Error!";

    }
    else
    {

       labelStatus.Text = "Device Is Initialised!";
    }

  }
  else
  {

     labelStatus.Text = "Communication Error!";

  }

}
else
{

  labelStatus.Text = "Error of opening Serial Port!";

}
```

4.1.3.3 Borland Delphi 6

//Declaration:

```
function STXInitialize(PName : String): PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
  if STXConnect(pNbr) = '1' then
  begin
    if STXOpenCommunication(pNbr) = '1' then
    begin
      if STXInitialize(pNbr) = '1' then
      begin
        // Device is successfully initialised
      end
      else
      begin
        Application.MessageBox('Initialization Error!', 'Warning!',
        MB_OK+MB_ICONERROR);
      end;
  end
    else
    begin
       Application.MessageBox('Communication Error!', 'Warning!',
       MB_OK+MB_ICONERROR);
    end;
  end
  else
  begin
    Application.MessageBox('Failed to open Serial Port!', 'Warning!',
    MB_OK+MB_ICONERROR);
  end;
```

4.1.3.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * Initialize)(char* PNbr);
Initialize STXInitialize;


//Initialization (constructor) *.cpp file

STXInitialize = (Initialize)GetProcAddress(dllInstance,
"STXInitialize");
```

//Execution:

```
if(String(STXConnect(pNbr)) == "1")
{
  if(String(STXOpenCommunication(pNbr)) == "1")
  {
    if (String(STXInitialize(pNbr)) == "1")
    {
       // Device is successfully initialised..
       …
       …

    }
    else
    {
       Application->MessageBox("Initialization Error!", "Warning!",
       MB_OK+MB_ICONERROR);
    }
  }
  else
  {
    Application->MessageBox("Communication Error!", "Warning!",
    MB_OK+MB_ICONERROR);
  }
}
else
{
  Application->MessageBox("Failed to open the Serial Port!",
  "Warning!", MB_OK+MB_ICONERROR);
}
```

4.1.4 STXReset(PChar PortNumber)

Reset the StoreX after the error. Puts the StoreX in the idle state.

Parameter: PortNumber – Number of a Serial Port.

The user should call the STXReset method after any error of the machine. The user should call STXInitialize again to continue operations, or press manually the "Reset" button of the machine.

Examples:

4.1.4.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXReset Lib "STX_lib" (ByVal pNbr As String)
```

'Execution:

```
STXReset(STXId)
```

4.1.4.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXReset(String pNbr);
```

//Execution:

```
STXReset(STXId);
```

4.1.4.3 Borland Delphi 6

//Declaration:

```
procedure STXReset(PName : String); stdcall;  external 'STX_lib.dll';
```

//Execution:

```
STXReset(pNbr);
```

4.1.4.4 *Borland C++Builder*

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * Reset)(char* PNbr);
Reset  STXReset;


//Initialization (constructor) *.cpp file

STXReset = (Reset)GetProcAddress(dllInstance, "STXReset");
```

//Execution:

```
STXReset(PNbr);
```

4.1.5 STXDisconnect(PChar PortNumber)

Closes serial communication through the active com port. This function also closes Serial communication for Barcode Reader.

Parameter: PortNumber – Number of a Serial Port.

Examples:

4.1.5.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXDisconnect Lib "STX_lib" (ByVal pNbr As
String)
```

'Execution:

```
STXDisconnect(STXId)
```

4.1.5.2 Visual C# .NET

//Declaration:

```
using System.Runtime.InteropServices;

[DllImport("STX_lib")]
private extern static void STXDisconnect(String pNbr);
```

//Execution:

```
STXDisconnect(STXId);
```

4.1.5.3 Borland Delphi 6

//Declaration:

```
procedure STXDisconnect(PName : String); stdcall; external
'STX_lib.dll';
```

//Execution:

```
STXDisconnect(pNbr);
```

4.1.5.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * Disconnect)(char* PNbr);
Disconnect STXDisconnect;


//Initialization (constructor) *.cpp file

STXDisconnect = (Disconnect)GetProcAddress(dllInstance,
"STXDisconnect");
```

//Execution:

```
STXDisconnect(PNbr);
```

4.1.6 STXSerial Number(PChar PortNumber)

Returns serial number of the device.

Parameter: PortNumber – Number of a Serial Port.

Examples:

4.1.6.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXSerialNumber Lib "STX_lib" (ByVal pNbr As
String) As String
```

'Execution:

```
Text = "StoreX S/N-" + STXSerialNumber(STXId)
```

4.1.6.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXSerialNumber(String pNbr);
```

//Execution:

```
Text = " StoreX S/N-"+ STXSerialNumber(STXId);
```

4.1.6.3 Borland Delphi 6

//Declaration:

```
function STXSerialNumber(PName : String) : PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
Caption := 'StoreX S/N: '+ STXSerialNumber(pNbr);
```

4.1.6.4 Borland C++Builder

//Declaration:

```
//Prototypes *.h file


typedef char* (__stdcall * SerialNumber)(char* PNbr);
SerialNumber STXSerialNumber;


//Initialization (constructor) *.cpp file

STXSerialNumber = (SerialNumber)GetProcAddress(dllInstance,
"STXSerialNumber");
```

//Execution:

```
Caption = "StoreX S/N: "+String(STXSerialNumber(pNbr));
```

4.1.7 STXIsInitialized(PChar PortNumber)

Checks whether the device has been initialised.

Parameter: PortNumber – Number of a Serial Port.

Return values (PChar):

```
"1" - Device is initialised.

"0","2","3" - Device is not initialised.
```

Examples:

4.1.7.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXIsInitialize Lib "STX_lib" (ByVal pNbr As
String) As String
```

'Execution:

```
If STXIsInitialized(STXId) = "1" Then

        MsgBox(''Device is initialised")

Else

        MsgBox("Device is not initialised ")

End If
```

4.1.7.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXIsInitialized(String pNbr);
```

//Execution:

```
if(STXIsInitialized(STXId) == "1")

{

        MessageBox.Show("Device is initialised");

}

else

{

        MessageBox.Show("Device is not initialised");

}
```

4.1.7.3 Borland Delphi 6

//Declaration:

```
function STXIsInitialized(PName : String): PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
if STXIsInitialized(pNbr) = '1' then
  ShowMessage('Device is initialised!')

else
  ShowMessage('Device is not initialised!');
```

4.1.7.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * IsInitialized)(char* PNbr);
IsInitialized STXIsInitialized;


//Initialization (constructor) *.cpp file

STXIsInitialized = (IsInitialized)GetProcAddress(dllInstance,
"STXIsInitialized");
```

//Execution:

```
if(String(STXIsInitialized(PNbr)) == "1")
  ShowMessage("Device is initialised!");
else
  ShowMessage("Device is not initialised!");
```

4.1.8 STXReadBackStr(PChar PortNumber, Message)

This is a generic function which allows the user to send a complete message in native syntax to the device (e.g. "ST 1901").

Parameter: PortNumber – Number of a Serial Port.

Return values (PChar):

Returns the raw string received back from the device (e.g. "OK").

Examples:

4.1.8.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadBackStr Lib "STX_lib" (ByVal pNbr As
String, ByVal Command  As String) As String
```

'Execution:

```
TxtBoxReadCommand.Text = STXReadBackStr(STXId, xtBoxSendCommand.Text)
```

4.1.8.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadBackStr(String pNbr, String
SendCommand);
```

//Execution:

```
TxtBoxReadCommand.Text = STXReadBackStr(STXId,
                                        TxtBoxSendCommand.Text);
```

4.1.8.3 Borland Delphi 6

//Declaration:

```
function STXReadBackStr(Pname, SendCmd : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
EditMessage.Text := STXReadBackStr(pNbr, 'ST 1900');
```

4.1.8.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadBackStr)(char* PNbr, char*
SendCommand);
ReadBackStr STXReadBackStr;


//Initialization (constructor) *.cpp file

STXReadBackStr = (ReadBackStr)GetProcAddress(dllInstance,
"STXReadBackStr");
```

//Execution:

```
EditReply->Text = String(STXReadBackStr(PNbr, EditCommand->
Text.c_str()));
```

4.2 Status Functions

4.2.1 STXReadReadyFlag(PChar PortNumber)

Reads the Ready Flag on the System.

Parameter: PortNumber - Number of a Serial Port.

Return values (PChar):

```
"1" - System is ready.

"0" - System is not ready.
```

Examples:

4.2.1.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadReadyFlag Lib "STX_lib" (ByVal pNbr
As String) As String
```

'Execution:

```
If STXReadReadyFlag(STXId) = "1" Then

  LbStatReady.Text = "Ready"

Else

  LbStatReady.Text = "Not Ready"

End If
```

4.2.1.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]

private extern static String STXReadReadyFlag(String pNbr);
```

//Execution:

```
if (STXReadReadyFlag(STXId) == "1")
{

  lbReadyFlag.Text = "Ready";

}
else
{

  lbReadyFlag.Text = "Not Ready";
}
```

4.2.1.3 Borland Delphi 6

//Declaration:

```
function STXReadReadyFlag(Pname: PChar) : PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
if STXReadReadyFlag(pNbr) = '1' then
  LbStatReady.Caption := 'Ready'

else
  LbStatReady.Caption := 'Not Ready';
```

4.2.1.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * OpenCommunication)(char* PNbr);
OpenCommunication   STXOpenCommunication;


//Initialization (constructor) *.cpp file

STXOpenCommunication = (OpenCommunication)GetProcAddress(dllInstance,
"STXOpenCommunication");
```

//Execution:

```
STXOpenCommunication(PNbr);
```

4.2.2 STXReadErrorFlag(PChar PortNumber)

Reads the Error Flag on the System.

Parameter: PortNumber - Number of a Serial Port.

Return values (PChar):

```
"1" - There is error.
"0" - No Errors.
```

Examples:

4.2.2.1 *Visual Basic .NET (Visual Basic 6.0)*

'Declaration:

```
Private Declare Function STXReadErrorFlag Lib "STX_lib" (ByVal pNbr
As String) As String
```

'Execution:

```
If STXReadErrorFlag(STXId) = "0" Then

        LbStatError.ForeColor = System.Drawing.SystemColors.HotTrack
        LbStatError.Text = "No Error"

Else
        LbStatError.ForeColor = System.Drawing.Color.Red
        LbStatError.Text = "Error"

End If
```

4.2.2.2 *Visual C# .NET*

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadErrorFlag(String pNbr);
```

//Execution:

```
if(STXReadErrorFlag(STXId) == "0")
{

     lbErrorFlag.ForeColor = System.Drawing.SystemColors.HotTrack;
     lbErrorFlag.Text = "No Error";
}
else
{

     lbErrorFlag.ForeColor = System.Drawing.Color.Red;
     lbErrorFlag.Text = "Error";
}
```

4.2.2.3

4.2.2.4 Borland Delphi 6

//Declaration:

```
function STXReadErrorFlag(PName: PChar): PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
if STXReadErrorFlag(pNbr) = '1' then
  LbStatError.Caption := 'Error!'

else
LbStatError.Caption := 'No Error!';
```

4.2.2.5 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadErrorFlag)(char* PNbr);
ReadErrorFlag STXReadErrorFlag;


//Initialization (constructor) *.cpp file

STXReadErrorFlag = (ReadErrorFlag)GetProcAddress(dllInstance,
"STXReadErrorFlag");
```

//Execution:

```
if(String(STXReadErrorFlag(PNbr)) == "1")
  LbError->Caption = "Error!";
else
  LbError->Caption = "No Error!";
```

4.2.3 STXReadErrorCode(PChar PortNumber)

Interrogates Error Flag and Error code of System.

Parameter: PortNumber - Number of a Serial Port.

Return values (PChar):

"0" - No Errors.
If returns the System status code then see descriptions of error codes.

Examples:

4.2.3.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadErrorCode Lib "STX_lib" (ByVal pNbr
As String) As String
```

'Execution:

```
Dim STXError As String

STXError = STXReadErrorCode(STXId)

If STXError = "0" Then

     LbErrorCode.ForeColor = System.Drawing.SystemColors.HotTrack
     LbErrorCode.Text = "0"  ' No Error

Else

     LbErrorCode.ForeColor = System.Drawing.Color.Red
     LbErrorCode.Text = STXError

End If
```

4.2.3.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadErrorCode(String pNbr);
```

//Execution:

```
String STXError;

STXError = STXReadErrorCode(STXId);

if(STXError == "0")
{

     lbErrorCode.ForeColor = System.Drawing.SystemColors.HotTrack;
     lbErrorCode.Text = "0";

}
else
{

     lbErrorCode.ForeColor = System.Drawing.Color.Red;
     lbErrorCode.Text = STXError;

}
```

4.2.3.3 Borland Delphi 6

//Declaration:

```
function STXReadErrorCode(PName: PChar): PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
if STXReadErrorFlag(pNbr) = '1' then
  LbErrorCode.Caption := STXReadErrorCode(pNbr)

else
  LbErrorCode.Caption := 'No Error!';
```

4.2.3.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadErrorCode)(char* PNbr);
ReadErrorCode STXReadErrorCode;


//Initialization (constructor) *.cpp file

STXReadErrorCode = (ReadErrorCode)GetProcAddress(dllInstance,
"STXReadErrorCode");
```

//Execution:

```
String ErrorCode;

ErrorCode = String(STXReadErrorCode(PNbr));

if(ErrorCode == "0")
  LbErrorCode->Caption = "No Error!";
else
  LbErrorCode->Caption = ErrorCode;
```

4.2.4 STXReadStatus(PChar PortNumber)

Interrogates Data Memory location 200 to retrieve the error code.

Parameter: PortNumber - Number of a Serial Port.

Return values (PChar):

```
"0" - No Errors.

"xxxxx" - Error Code.
```

Examples:

4.2.4.1 *Visual Basic .NET (Visual Basic 6.0)*

'Declaration:

```
Private Declare Function STXReadStatus Lib "STX_lib" (ByVal pNbr As
String) As String
```

'Execution:

```
Dim STXStat As String

STXStat = STXReadStatus(STXId)

If STXStat = "0" Then

     LbStatus.ForeColor = System.Drawing.SystemColors.HotTrack
     LbStatus.Text = "No Error"

Else

     LbStatus.ForeColor = System.Drawing.Color.Red
     LbStatus.Text = "Error-" + STXStat

End If
```

4.2.4.2 *Visual C# .NET*

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadStatus(String pNbr);
```

//Execution:

```
String STXStat;

STXStat = STXReadStatus(STXId);

if(STXStat == "0")
{

     lbStatus.ForeColor = System.Drawing.SystemColors.HotTrack;
     lbStatus.Text = "No Error";

}
else
{

     lbStatus.ForeColor = System.Drawing.Color.Red;
     lbStatus.Text = "Error-" + STXStat;
}
```

4.2.4.3 Borland Delphi 6

//Declaration:

```
function STXReadStatus(PName: PChar): PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
var
  STXStatus : String;
begin

  STXStatus := STXReadStatus(pNbr);

  if STXStatus = '0' then
    LbStatus.Caption := 'No Error!'
  else
    LbStatus.Caption := STXStatus;

end;
```

4.2.4.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadStatus)(char* PNbr);
ReadStatus STXReadStatus;


//Initialization (constructor) *.cpp file

STXReadStatus = (ReadStatus)GetProcAddress(dllInstance,
"STXReadStatus");
```

//Execution:

```
String Status;

Status = String(STXReadStatus(PNbr));

if(Status == "0")
  LbStatus->Caption = "No Error!";
else
  LbStatus->Caption = "Error - "+Status;
```

4.2.5 STXReadActualSlotPosition(PChar PortNumber)

Returns current slot position, as a string of characters.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.2.5.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadActualSlotPosition Lib "STX_lib"
(ByVal pNbr As String) As String
```

'Execution:

```
Dim SlotPosition As Integer
SlotPosition = Int(STXReadActualSlotPosition(STXId))
```

4.2.5.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadActualSlotPosition(String pNbr);
```

//Execution:

```
int SlotPos;
SlotPos = System.Convert.ToInt16(STXReadActualSlotPosition(STXId));
```

4.2.5.3 Borland Delphi 6

//Declaration:

```
function STXReadActualSlotPosition(PName: PChar): PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
var
  SlotPos : Integer;
begin

  SlotPos := StrToInt(STXReadActualSlotPosition('1'));

end;
```

4.2.5.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadActualSlotPosition)(char* PNbr);
ReadActualSlotPosition STXReadActualSlotPosition;


//Initialization (constructor) *.cpp file

STXReadActualSlotPosition =
(ReadActualSlotPosition)GetProcAddress(dllInstance,
"STXReadActualSlotPosition");
```

//Execution:

```
int SlotPos;

SlotPos = StrToInt(String(STXReadActualSlotPosition(PNbr)));
```

4.2.6 STXReadUserDoorFlag(PChar PortNumber)

Reads whether door is opened.

Parameter: PortNumber - Number of a Serial Port.

Return values (PChar):

```
"1" - User's door is opened.
"0" - User's door is closed.
```

Examples:

4.2.6.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadUserDoorFlag Lib "STX_lib" (ByVal
pNbr As String) As String
```

'Execution:

```
If STXReadUserDoorFlag(STXId) = "0" Then

        LbStatDoor.Text = "Door is closed"

Else
        LbStatDoor.Text = "Door is opened"

End If
```

4.2.6.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadUserDoorFlag(String pNbr);
```

//Execution:

```
if (STXReadUserDoorFlag(STXId) == "0")
{

  lbStatDoor.Text = "Closed";

}
else
{

  lbStatDoor.Text = "Opened";
}
```

4.2.6.3 Borland Delphi 6

//Declaration:

```
function STXReadUserDoorFlag(PName: PChar): PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
if STXReadUserDoorFlag(pNbr) = '1' then
  lbStatDoor.Caption := 'Door is Opened!'

else
  lbStatDoor.Caption := 'Door is Closed!';
```

4.2.6.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadUserDoorFlag)(char* PNbr);
ReadUserDoorFlag STXReadUserDoorFlag;


//Initialization (constructor) *.cpp file

STXReadUserDoorFlag = (ReadUserDoorFlag)GetProcAddress(dllInstance,
"STXReadUserDoorFlag");
```

//Execution:

```
if(String(STXReadUserDoorFlag(PNbr)) == "1")
  LbDoorStatus->Caption = "Door is Opened!";
else
  LbDoorStatus->Caption = "Door is Closed!";
```

4.3 Setting Functions

4.3.1 STXSetBCRZOffset(PChar PortNumber, BCRZ)

Sets Barcode Reader Z-Lift Read Position offset.

Parameter:

```
PortNumber - Number of a Serial Port.

BCRZ - value of Z-Lift Position
```

Examples:

4.3.1.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXSetBCRZOffset Lib "STX_lib" (ByVal pNbr As
String, ByVal BCRZOfst As String)
```

'Execution:

```
STXSetBCRZOffset(STXId, TextBoxSetBCRZOffset.Text)
```

4.3.1.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXSetBCRZOffset(String pNbr, String
BCRZOffset);
```

//Execution:

```
STXSetBCRZOffset(STXId, textBoxBCRZOffset.Text);
```

4.3.1.3 Borland Delphi 6

//Declaration:

```
procedure STXSetBCRZOffset(Pname, BCRZValue: PChar); stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXSetBCRZOffset(pNbr, EditBCRZ.Text);
```

4.3.1.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * SetBCRZOffset)(char* PNbr, char* ZOffset);
SetBCRZOffset STXSetBCRZOffset;


//Initialization (constructor) *.cpp file

STXSetBCRZOffset = (SetBCRZOffset)GetProcAddress(dllInstance,
"STXSetBCRZOffset");
```

//Execution:

```
STXSetBCRZOffset(PNbr, EditBCRZ->Text.c_str());
```

4.3.2 STXReadMaxSlot(PChar PortNumber)

Returns default number of slots.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.3.2.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadMaxSlot Lib "STX_lib" (ByVal pNbr As
String) As String
```

'Execution:

```
Dim MaxSlot As Integer
Dim MaxLevel As Integer

If STXInitialize(STXId) = "1" Then

     MaxSlot = Int(STXReadMaxSlot(STXId))
     MaxLevel = Int(STXReadMaxLevel(STXId))

End If
```

4.3.2.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadMaxSlot(String pNbr);
```

//Execution:

```
int MaxSlot;
MaxSlot = System.Convert.ToInt16(STXReadMaxSlot(STXId));
```

4.3.2.3 Borland Delphi 6

//Declaration:

```
function STXReadMaxSlot(Pname: PChar) : PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
var
  MaxSlot : Integer;
begin

  MaxSlot := StrToInt(STXReadMaxSlot(pNbr));

end;
```

4.3.2.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadMaxSlot)(char* PNbr);
ReadMaxSlot STXReadMaxSlot;


//Initialization (constructor) *.cpp file

STXReadMaxSlot = (ReadMaxSlot)GetProcAddress(dllInstance,
"STXReadMaxSlot");
```

//Execution:

```
LbMaxSlot->Caption = String(STXReadMaxSlot(PNbr));
```

4.3.3 STXReadMaxLevel(PChar PortNumber)

Returns default number of levels.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.3.3.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadMaxLevel Lib "STX_lib" (ByVal pNbr As
String) As String
```

'Execution:

```
Dim MaxSlot As Integer
Dim MaxLevel As Integer

If STXInitialize(STXId) = "1" Then

     MaxSlot = Int(STXReadMaxSlot(STXId))
     MaxLevel = Int(STXReadMaxLevel(STXId))

End If
```

4.3.3.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadMaxLevel (String pNbr);
```

//Execution:

```
int MaxLevel;
MaxLevel = System.Convert.ToInt16(STXReadMaxLevel(STXId));
```

4.3.3.3 Borland Delphi 6

//Declaration:

```
function STXReadMaxLevel(Pname: PChar) : PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
var
  MaxLevel : Integer;
begin
  MaxLevel := StrToInt(STXReadMaxLevel(pNbr));
end;
```

4.3.3.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadMaxLevel)(char* PNbr);
ReadMaxLevel STXReadMaxLevel;


//Initialization (constructor) *.cpp file

STXReadMaxLevel = (ReadMaxLevel)GetProcAddress(dllInstance,
"STXReadMaxLevel");
```

//Execution:

```
LbMaxLevel->Caption = String(STXReadMaxLevel(PNbr));
```

4.3.4 STXReadBCRZOffset(PChar PortNumber)

Reads BCR z-lift Read Position offset.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.3.4.1 *Visual Basic .NET (Visual Basic 6.0)*

'Declaration:

```
Private Declare Function STXReadBCRZOffset Lib "STX_lib" (ByVal pNbr
As String) As String
```

'Execution:

```
Dim BCRZOffset As Integer

BCRZOffset = Int(STXReadBCRZOffset(STXId))
```

4.3.4.2 *Visual C# .NET*

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadBCRZOffset(String pNbr);
```

//Execution:

```
int BCRZOffset;
BCRZOffset = System.Convert.ToInt16(STXReadBCRZOffset(STXId));
```

4.3.4.3 *Borland Delphi 6*

//Declaration:

```
function STXReadBCRZOffset(Pname: PChar): PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
LbBCRz.Caption := STXReadBCRZOffset(pNbr);
```

4.3.4.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file

typedef char* (__stdcall * ReadBCRZOffset)(char* PNbr);
ReadBCRZOffset STXReadBCRZOffset;


//Initialization (constructor) *.cpp file

STXReadBCRZOffset = (ReadBCRZOffset)GetProcAddress(dllInstance,
"STXReadBCRZOffset");
```

//Execution:

```
LbBCRZ->Caption = String(STXReadBCRZOffset(PNbr));
```

4.3.5 STXReadHandlerZPitch(PChar PortNumber)

Returns value of handler Z pitch.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.3.5.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadHandlerZPitch Lib "STX_lib" (ByVal
pNbr As String) As String
```

'Execution:

```
Dim HZPitch As Integer

HZPitch = Int(STXReadHandlerZPitch(STXId))
```

4.3.5.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]

private extern static String STXReadHandlerZPitch(String pNbr);
```

//Execution:

```
int HandZPitch;

HandZPitch = System.Convert.ToInt16(STXReadHandlerZPitch(STXId));
```

4.3.5.3 Borland Delphi 6

//Declaration:

```
function STXReadHandlerZPitch(Pname: PChar): PChar; stdcall; external
'STX_lib.dll';
```

//Execution:

```
var
  ZPitch : Integer;
begin

  ZPitch := StrToInt(STXReadHandlerZPitch(pNbr));

end;
```

4.3.5.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadHandlerZPitch)(char* PNbr);
ReadHandlerZPitch STXReadHandlerZPitch;


//Initialization (constructor) *.cpp file

STXReadHandlerZPitch = (ReadHandlerZPitch)GetProcAddress(dllInstance,
"STXReadHandlerZPitch");
```

//Execution:

```
LbHandlZPitch->Caption = String(STXReadHandlerZPitch(PNbr));
```

4.3.6 STXSetHandlerZPitch(PChar PortNumber, ZPitch)

Sets Handler z-Pitch, as a string of characters.

Parameter:

PortNumber - Number of a Serial Port.
ZPitch - value of Z-Pich

Examples:

4.3.6.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXSetHandlerZPitch Lib "STX_lib" (ByVal pNbr As
String, ByVal HandZ As String)
```

'Execution:

```
STXSetHandlerZPitch (STXId, TextBoxSetHandZ.Text)
```

4.3.6.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXSetHandlerZPitch(String pNbr, String
HandZPitch);
```

//Execution:

```
STXSetHandlerZPitch(STXId, textBoxZPitch.Text);
```

4.3.6.3 Borland Delphi 6

//Declaration:

```
procedure STXSetHandlerZPitch(Pname, BCRZValue : PChar); stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXSetHandlerZPitch('3','788');
```

4.3.6.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file

typedef void (__stdcall * SetHandlerZPitch)(char* PNbr, char*
ZPitch);
SetHandlerZPitch STXSetHandlerZPitch;

//Initialization (constructor) *.cpp file

STXSetHandlerZPitch = (SetHandlerZPitch)GetProcAddress(dllInstance,
"STXSetHandlerZPitch");
```

//Execution:

```
//STXSetHandlerZPitch(PNbr, EditHandlZ->Text.c_str());

int ZPitch;

ZPitch = 900;

STXSetHandlerZPitch(PNbr, IntToStr(ZPitch).c_str());
```

4.3.7 STXSetCassetteConfig(Pchar PortNumber, Cassette, MaxLevel, PitchMax)

Defines pitch value and number of levels for one cassette. The specified pitch will be set automatically during plate operations with specified cassette/slot. The STXSetCassetteConfig function saves values of MaxLevel and Pitch to internal array in the dll.

Parameters:

`PortNumber - Number of a Serial Port.`

`Cassette - Number of Cassette.`

`MaxLevel - Value of maximum Level.`

`PitchMax - Value of z-Pitch.`

Examples:

4.3.7.1 *Visual Basic .NET (Visual Basic 6.0)*

'Declaration:

```
Private Declare Sub STXSetCassetteConfig Lib "STX_lib" (ByVal pNbr As
String, ByVal Cassette As String, ByVal MaxLevel As String, ByVal
PitchMax As String)
```

'Execution:

```
Dim i As Integer


For i = 1 To 15
  STXSetCassetteConfig(STXId, Str(i), "8", "2158")
Next i


For i = 16 To 20
  STXSetCassetteConfig(STXId, Str(i), "22", "788")
Next i


For i = 21 To 35
  STXSetCassetteConfig(STXId, Str(i), "10", "2158")
Next i


For i = 36 To 40
STXSetCassetteConfig(STXId, Str(i), "28", "788")
Next I
```

4.3.7.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXSetCassetteConfig(String pNbr, String
Cassette, String MaxLevel, String PitchMax);
```

//Execution:

```
for(int i=1; i<=10; i++)
{

     STXSetCassetteConfig(STXId, System.Convert.ToString(i), "8",
     "2158");

}
for(int i=11; i<=20; i++)
{

     STXSetCassetteConfig(STXId, System.Convert.ToString(i), "22",
     "788");
}
```

4.3.7.3 Borland Delphi 6

//Declaration:

```
procedure STXSetCassetteConfig(Pname, Cassette, MaxLevelSet, PitchMax
: PChar); stdcall;  external 'STX_lib.dll';
```

//Execution:

```
var
  i : Integer;
begin

  for i := 1 to 5 do
  begin
    STXSetCassetteConfig(pNbr, PChar(IntToStr(i)), '8', '2158');
  end;

  for i := 6 to 10 do
  begin
    STXSetCassetteConfig(pNbr, PChar(IntToStr(i)), '22', '788');
  end;

end;
```

4.3.7.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * SetCassetteConfig)(char* PNbr, char*
Cassette, char* MaxLevelSet, char* PitchMax);
SetCassetteConfig STXSetCassetteConfig;


//Initialization (constructor) *.cpp file

STXSetCassetteConfig = (SetCassetteConfig)GetProcAddress(dllInstance,
"STXSetCassetteConfig");
```

//Execution:

```
String MaxLevel;
String ZPitch;
int Slot;

for(int i = 1; i<6; i++)
{

  MaxLevel = "8";
  ZPitch = "2158";
  Slot = i;
  STXSetCassetteConfig(PNbr, IntToStr(Slot).c_str(),
  MaxLevel.c_str(), ZPitch.c_str());
}

for(int i = 6; i<10; i++)
{
  MaxLevel = "22";
  ZPitch = "788";
  Slot = i;
  STXSetCassetteConfig(PNbr, IntToStr(Slot).c_str(),
  MaxLevel.c_str(), ZPitch.c_str());
}
```

4.3.8 STXReadCassetteConfig(Pchar PortNumber, Cassette)

Returns defined values of Maximum Level and Z-Pitch of Cassette. If values of Maximum Level and Z-Pitch were not assigned with the function STXSetCassetteConfig, by default returns - ''0,0'.

Function STXReadCassetteConfig returns values as string of characters, separated by coma. Values of Maximum Level and Z-Pitch returns from internal array of the dll.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.3.8.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadCassetteConfig Lib "STX_lib" (ByVal
pNbr As String, ByVal Stack As String) As String
```

'Execution:

```
Dim CassetteConf As String

CassetteConfig = STXReadCassetteConfig(STXId, "5")
```

4.3.8.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadCassetteConfig(String pNbr,
String Cassette);
```

//Execution:

```
String CassetteConfig = STXReadCassetteConfig(STXId,
textBoxCConf.Text);
```

4.3.8.3 Borland Delphi 6

//Declaration:

```
function STXReadCassetteConfig(pNbr, Cassette : PChar) : PChar;
stdcall; external 'STX_lib.dll';
```

//Execution:

```
var
  MaxLevel : Integer;
  ZPitch : Integer;
  Cassette : PChar;

begin

Cassette := '3';

MaxLevel := StrToInt(Copy(STXReadCassetteConfig(pNbr,Cassette), 0,
pos(',', STXReadCassetteConfig(pNbr, Cassette))-1));

ZPitch := StrToInt(Copy(STXReadCassetteConfig(pNbr,Cassette),
pos(',',
STXReadCassetteConfig(pNbr,Cassette)) + 1,
Length(STXReadCassetteConfig(pNbr,Cassette)) - pos(',',
STXReadCassetteConfig(pNbr,Cassette))));

end;
```

4.3.8.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadCassetteConfig)(char* PNbr, char*
Cassette);
ReadCassetteConfig STXReadCassetteConfig;


//Initialization (constructor) *.cpp file

STXReadCassetteConfig =
(ReadCassetteConfig)GetProcAddress(dllInstance,
"STXReadCassetteConfig");
```

//Execution:

```
LbSlotParam->Caption = String(STXReadCassetteConfig(PNbr,
EditSlotParam->Text.c_str()));
```

4.3.9 STXSetPlateDetector(PChar PortNumber, PChar {0.1})

Sets whether plate detector is present as value of global variable of the dll. This value is necessary for correct use those functions which use the plate detector (e.g. STXIsPlateAtLocation, STXReadBarcode, STXImportPlate, STXExpotrPlate, STXMovePlate). By default this value is "0", plate detector is not present.

Parameters:

`PortNumber - Number of a Serial Port.`

`"0" - Plate detector is not present.`

`"1" - Plate detector is present.`

Examples:

4.3.9.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXSetPlateDetector Lib "STX_lib" (ByVal pNbr As
String, ByVal IsDetector As String)
```

'Execution:

```
STXDevicePath("1")
STXConnect(STXId)
STXOpenCommunication(STXId)

If STXInitialize = "1" Then

     STXSetPlateDetector(STXId, "1")
     STXSetPlateTStatDetector(STXId, "1")
     STXSetPlateTStatDetector2(STXId, "0")
     STXSetPlateShovelDetector(STXId, "1")


End If
```

4.3.9.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXSetPlateDetector(String pNbr, String
PlateDetect);
```

//Execution:

```
STXSetPlateDetector(STXId, "1");


If (STXIsPlateAtLocation(STXId, "5", "1") =="1")
{

 …..

}
else
{

 …..

}
```

4.3.9.3 Borland Delphi 6

//Declaration:

```
procedure STXSetPlateDetector(pNbr, val : PChar); stdcall;  external
'STX_lib.dll';
```

//Execution:

```
STXSetPlateDetector(pNbr, '1');
```

4.3.9.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * SetPlateDetector)(char* PNbr, char* param);
SetPlateDetector STXSetPlateDetector;


//Initialization (constructor) *.cpp file

STXSetPlateDetector = (SetPlateDetector)GetProcAddress(dllInstance,
"STXSetPlateDetector");
```

//Execution:

```
STXSetPlateDetector(PNbr, "1");
```

4.3.10 STXSetPlateShovelDetector(PChar PortNumber, PChar {0.1})

Sets whether Shovel Plate Detector is present, as value of global variable of the dll. Sets whether Shovel Plate Detector is present.This value is necessary for correct use those functions which use the Plate Shovel Detector (STXReadShovelDetector). By default this value is "0", Plate Shovel Detector is not present.

Parameters:

```
PortNumber - Number of a Serial Port.

"0" - Plate Shovel Detector is not present.

"1" - Plate Shovel Detector is present.
```

Examples:

4.3.10.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXSetPlateShovelDetector Lib "STX_lib" (ByVal
pNbr As String, ByVal IsDetector As String)
```

'Execution:

```
STXDevicePath(STXId, "1")
STXConnect(STXId)
STXOpenCommunication(STXId)

If STXInitialize(STXId) = "1" Then

        STXSetPlateDetector(STXId, "1")
        STXSetPlateTStatDetector(STXId, "1")
        STXSetPlateTStatDetector2(STXId, "0")
        STXSetPlateShovelDetector(STXId, "1")

End If
```

4.3.10.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXSetPlateShovelDetector (String pNbr,
String PlateDetect);
```

//Execution:

```
STXSetPlateShovelDetector (STXId, "1");
```

4.3.10.3 Borland Delphi 6

//Declaration:

```
procedure STXSetPlateShovelDetector(pNbr, val : PChar); stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXSetPlateShovelDetector(pNbr, '1');
```

4.3.10.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * SetPlateShovelDetector)(char* PNbr, char*
param);
SetPlateShovelDetector STXSetPlateShovelDetector;


//Initialization (constructor) *.cpp file

STXSetPlateShovelDetector =
(SetPlateShovelDetector)GetProcAddress(dllInstance,
"STXSetPlateShovelDetector");
```

//Execution:

```
STXSetPlateShovelDetector(pNbr, '1');
```

4.3.11 STXSetPlateTStatDetector(PChar PortNumber, PChar {0.1})

Sets whether plate detector is present on the Transfer Station, as value of global variable of the dll. This value is necessary for correct use those functions which use the plate detector on the Transfer Station (STXImportPlate, STXExpotrPlate, STXReadTransferStationDetector). By default this value is "0", plate detector is not present on the Transfer Station.

Parameters:

`PortNumber - Number of a Serial Port.`

`"0" - Plate detector is not present on the Transfer Station.`

`"1" - Plate detector is present on the Transfer Station.`

Examples:

4.3.11.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXSetPlateTStatDetector Lib "STX_lib" (ByVal
pNbr As String, ByVal IsDetector As String)
```

'Execution:

```
STXDevicePath("1")
STXConnect(STXId)
STXOpenCommunication(STXId)

If STXInitialize = "1" Then

        STXSetPlateDetector(STXId, "1")
        STXSetPlateTStatDetector(STXId, "1")
        STXSetPlateTStatDetector2(STXId, "0")
        STXSetPlateShovelDetector(STXId, "1")

End If
```

4.3.11.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXSetPlateTStatDetector(String pNbr,
String PlateDetect);
```

//Execution:

```
STXSetPlateTStatDetector (STXId, "0");
```

4.3.11.3 Borland Delphi 6

//Declaration:

```
procedure STXSetPlateTStatDetector(pNbr, val : PChar); stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXSetPlateTStatDetector(pNbr, '1');
```

4.3.11.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * SetPlateTStatDetector)(char* PNbr, char*
param);
SetPlateTStatDetector STXSetPlateTStatDetector;


//Initialization (constructor) *.cpp file

STXReadTransferStationDetector =
(ReadTransferStationDetector)GetProcAddress(dllInstance,
"STXReadTransferStationDetector");
```

//Execution:

```
STXSetPlateTStatDetector(pNbr, '1');
```

4.3.12 STXSetPlateTStatDetector2(PChar PortNumber, PChar {0.1})

Sets whether plate detector is present on the second Transfer Station, as value of global variable of the dll. This value is necessary for correct use those functions which use the plate detector on the second Transfer Station (STXReadTransferStation2Detector). By default this value is "0", plate detector is not present on the second Transfer Station.

Parameters:

```
PortNumber - Number of a Serial Port.

"0" - Plate detector is not present on the second Transfer Station.

"1" - Plate detector is present on the second Transfer Station.
```

Examples:

4.3.12.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXSetPlateTStatDetector2 Lib "STX_lib" (ByVal
pNbr As String, ByVal IsDetector As String)
```

'Execution:

```
STXDevicePath("1")
STXConnect(STXId)
STXOpenCommunication(STXId)

If STXInitialize = "1" Then
        STXSetPlateDetector(STXId, "1")
        STXSetPlateTStatDetector(STXId, "1")
        STXSetPlateTStatDetector2(STXId, "0")
        STXSetPlateShovelDetector(STXId, "1")

End If
```

4.3.12.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXSetPlateTStatDetector2(String pNbr,
String  PlateDetect);
```

//Execution:

```
STXSetPlateTStatDetector2 (STXId, "0");
```

4.3.12.3 Borland Delphi 6

//Declaration:

```
procedure STXSetPlateTStatDetector2(pNbr, val : PChar); stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXSetPlateTStatDetector2(pNbr, '1');
```

4.3.12.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * SetPlateTStatDetector2)(char* PNbr, char*
param);
SetPlateTStatDetector2 STXSetPlateTStatDetector2


//Initialization (constructor) *.cpp file

STXReadTransferStationDetector2 =
(ReadTransferStationDetector2)GetProcAddress(dllInstance,
"STXReadTransferStationDetector2");
```

//Execution:

```
STXSetPlateTStatDetector2(pNbr, '0');
```

4.3.13 STXReadConfFile(PChar PortNumber, FileName)

Reads StoreX configuration file. Such file is a text file with extra initialization values. File can have any extension.

Parameter: PortNumber - Number of a Serial Port.

Example of StoreX configuration file:

```
...
DM22 10500
DM23 1925
DM80 100
...
T48 50
...
```

Parameters:

`"FileName" – Name of the configuration file.`

Return values (PChar):

```
"1" - The file has successfully opened and has been read

"0" - Device is not initialised.

"2" - Error in reading the file.
```

Examples:

4.3.13.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadConfFile Lib "STX_lib" (ByVal pNbr As
String, ByVal Fname As String) As String
```

'Execution:

```
'STXReadConfFile(STXId, "C:\file_name.LCF")

Private Sub OpenFileDialog1_FileOk(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles OpenFileDialog1.FileOk

STXReadConfFile(STXId, OpenFileDialog1.FileName)

End Sub
```

4.3.13.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadConfFile(String pNbr, String
FileName);
```

//Execution:

```
//STXReadConfFile(STXId, "C:\file_name.LCF");

STXReadConfFile(STXId, openFileDialog1.FileName);
```

4.3.13.3 Borland Delphi 6

//Declaration:

```
function STXReadConfFile(pNbr, FName : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXReadConfFile (pNbr, 'config_file.txt');
```

4.3.13.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadConfFile)(char* PNbr, char* FileName);
ReadConfFile STXReadConfFile;


//Initialization (constructor) *.cpp file

STXReadConfFile = (ReadConfFile)GetProcAddress(dllInstance,
"STXReadConfFile");
```

//Execution:

```
if(OpenDialog1->Execute())
{
  STXReadConfFile(PNbr, OpenDialog1->FileName.c_str());
}
```

4.4 Environmental Functions

4.4.1 STXReadActualTemperature(PChar PortNumber)

Returns value of temperature in °C, as a string of characters.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.4.1.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadActualTemperature Lib "STX_lib"
(ByVal pNbr As String) As String
```

'Execution:

```
Dim ActualTemperature As Single

ActualTemperature = CSng(STXReadActualTemperature(STXId))
```

4.4.1.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadActualTemperature(String pNbr);
```

//Execution:

```
float ActualTemp;

ActualTemp =
            System.Convert.ToSingle(STXReadActualTemperature(STXId));
```

4.4.1.3 Borland Delphi 6

//Declaration:

```
function STXReadActualTemperature(pNbr : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
var
   ActualTemperature : Real;
begin

   ActualTemperature := StrToFloat(STXReadActualTemperature(pNbr));

end;
```

4.4.1.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadActualTemperature)(char* PNbr);
ReadActualTemperature STXReadActualTemperature;


//Initialization (constructor) *.cpp file

STXReadActualTemperature =
(ReadActualTemperature)GetProcAddress(dllInstance,
"STXReadActualTemperature");
```

//Execution:

```
LbTemperature->Caption = String(STXReadActualTemperature(PNbr));
```

4.4.2 STXReadActualHumidity(PChar PortNumber)

Returns value of relative humidity in percent, as a string of characters.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.4.2.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadActualHumidity Lib "STX_lib" (ByVal
pNbr As String) As String
```

'Execution:

```
Dim ActualHumidity As Single

ActualHumidity = CSng(STXReadActualHumidity(STXId))
```

4.4.2.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadActualHumidity(String pNbr);
```

//Execution:

```
float ActualHum;

ActualHum = System.Convert.ToSingle(STXReadActualHumidity(STXId));
```

4.4.2.3 Borland Delphi 6

//Declaration:

```
function STXReadActualHumidity(pNbr : PChar): PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
var
   ActualHumidity : Real;
begin

   ActualHumidity := StrToFloat(STXReadActualHumidity(pNbr));

end;
```

4.4.2.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadActualHumidity)(char* PNbr);
ReadActualHumidity STXReadActualHumidity;


//Initialization (constructor) *.cpp file

STXReadActualHumidity =
(ReadActualHumidity)GetProcAddress(dllInstance,
"STXReadActualHumidity");
```

//Execution:

```
LbHumidity->Caption = String(STXReadActualHumidity(PNbr));
```

4.4.3 STXReadActualCO2(Pchar PortNumber)

Returns the CO2 concentration in percent, as a string of characters.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.4.3.1 *Visual Basic .NET (Visual Basic 6.0)*

'Declaration:

```
Private Declare Function STXReadActualCO2 Lib "STX_lib" (ByVal pNbr
As String) As String
```

'Execution:

```
Dim ActualCO2 As Single

ActualCO2 = CSng(STXReadActualCO2(STXId))
```

4.4.3.2 *Visual C# .NET*

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadActualCO2(String pNbr);
```

//Execution:

```
float ActualCO2;
ActualCO2 = System.Convert.ToSingle(STXReadActualCO2(STXId));
```

4.4.3.3 *Borland Delphi 6*

//Declaration:

```
function STXReadActualCO2(pNbr : PChar): PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
var
  ActualCO2 : Real;
begin

  ActualCO2 := StrToFloat(STXReadActualCO2('2'));

end;
```

4.4.3.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadActualCO2)(char* PNbr);
ReadActualCO2 STXReadActualCO2;


//Initialization (constructor) *.cpp file

STXReadActualCO2 = (ReadActualCO2)GetProcAddress(dllInstance,
"STXReadActualCO2");
```

//Execution:

```
LbCO2->Caption = String(STXReadActualCO2(PNbr));
```

4.4.4 STXReadActualN2(PChar PortNumber)

Returns the N2 concentration in percent, as a string of characters.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.4.4.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadActualN2 Lib "STX_lib" (ByVal pNbr As
String) As String
```

'Execution:

```
Dim ActualN2 As Single

ActualN2 = CSng(STXReadActualN2(STXId))
```

4.4.4.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadActualN2(String pNbr);
```

//Execution:

```
float ActualN2;
ActualN2 = System.Convert.ToSingle(STXReadActualN2(STXId));
```

4.4.4.3 Borland Delphi 6

//Declaration:

```
function STXReadActualN2(pNbr : PChar): PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
var
  ActualN2 : Real;
begin

  ActualN2 := StrToFloat(STXReadActualN2(pNbr));

end;
```

4.4.4.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadActualN2)(char* PNbr);
ReadActualN2 STXReadActualN2;


//Initialization (constructor) *.cpp file

STXReadActualN2 = (ReadActualN2)GetProcAddress(dllInstance,
"STXReadActualN2");
```

//Execution:

```
LbN2->Caption = String(STXReadActualN2(PNbr));
```

4.4.5 STXWriteSetTemperature(PChar PortNumber, PChar Temperature)

Sets the target temperature in °C, as a string of characters.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.4.5.1 *Visual Basic .NET (Visual Basic 6.0)*

'Declaration:

```
Private Declare Sub STXWriteSetTemperature Lib "STX_lib" (ByVal pNbr
As String, ByVal SetTemp As String)
```

'Execution:

```
Dim WriteTemp As Single

WriteTemp = 36

STXWriteSetTemperature(STXId, Str(WriteTemp))

'STXWriteSetTemperature(STXId, TextBoxSetTemp.Text)
```

4.4.5.2 *Visual C# .NET*

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXWriteSetTemperature(String pNbr,
String SetTemp);
```

//Execution:

```
float SetTemp = 36;
STXWriteSetTemperature(System.Convert.ToString(STXId, SetTemp));

//STXWriteSetTemperature(STXId, textBoxTemp.Text);
```

4.4.5.3 Borland Delphi 6

//Declaration:

```
procedure STXWriteSetTemperature(pNbr, T_value : PChar); stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXWriteSetTemperature('1', PChar(EditTemt.Text));
```

4.4.5.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * WriteSetTemperature)(char* PNbr, char*
param);
WriteSetTemperature STXWriteSetTemperature;


//Initialization (constructor) *.cpp file

STXWriteSetTemperature =
(WriteSetTemperature)GetProcAddress(dllInstance,
"STXWriteSetTemperature");
```

//Execution:

```
STXWriteSetTemperature(PNbr, EditSetTemp->Text.c_str());
```

4.4.6 STXWriteSetHumidity(PChar PortNumber, Humidity)

Sets the target relative humidity in percent, as a string of characters.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.4.6.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXWriteSetHumidity Lib "STX_lib" (ByVal pNbr As
String, ByVal SetHumidity As String)
```

'Execution:

```
Dim WriteHumidity As Single

WriteHumidity = 90

STXWriteSetHumidity(STXId, Str(WriteHumidity))
```

4.4.6.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXWriteSetHumidity(String pNbr, String
SetHum);
```

//Execution:

```
STXWriteSetHumidity(STXId, textBoxHum.Text);
```

4.4.6.3 Borland Delphi 6

//Declaration:

```
procedure STXWriteSetHumidity (pNbr, H_value : PChar); stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXWriteSetHumidity(pNbr, PChar(EditHum.Text));
```

4.4.6.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * WriteSetHumidity)(char* PNbr, char* param);
WriteSetHumidity STXWriteSetHumidity;


//Initialization (constructor) *.cpp file

STXWriteSetHumidity = (WriteSetHumidity)GetProcAddress(dllInstance,
"STXWriteSetHumidity");
```

//Execution:

```
STXWriteSetHumidity(PNbr, EditSetHum->Text.c_str());
```

4.4.7 STXWriteSetCO2(PChar PortNumber, CO2)

Sets the target CO2 concentration in percent, as a string of characters.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.4.7.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXWriteSetCO2 Lib "STX_lib" (ByVal pNbr As
String, ByVal SetCO2 As String)
```

'Execution:

```
STXWriteSetCO2(STXId, TextBoxSetSO2.Text)
```

4.4.7.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXWriteSetCO2(String pNbr, String
SetCO2);
```

//Execution:

```
STXWriteSetCO2(STXId, textBoxCO2.Text);
```

4.4.7.3 Borland Delphi 6

//Declaration:

```
procedure STXWriteSetCO2(pNbr, C_value : PChar); stdcall;  external
'STX_lib.dll';
```

//Execution:

```
STXWriteSetCO2(pNbr, PChar(EditCO2.Text));
```

4.4.7.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * WriteSetCO2)(char* PNbr, char* param);
WriteSetCO2 STXWriteSetCO2;


//Initialization (constructor) *.cpp file

STXWriteSetCO2 = (WriteSetCO2)GetProcAddress(dllInstance,
"STXWriteSetCO2");
```

//Execution:

```
STXWriteSetCO2(PNbr, EditSetCO2->Text.c_str());
```

4.4.8 STXWriteSetN2(PChar PortNumber, N2)

Sets the target N2 concentration in percent, as a string of characters.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.4.8.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXWriteSetN2 Lib "STX_lib" (ByVal pNbr As
String, ByVal SetN2 As String)
```

'Execution:

```
STXWriteSetN2 (STXId, TextBoxSetN2.Text)
```

4.4.8.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXWriteSetN2(String pNbr, String SetN2);
```

//Execution:

```
STXWriteSetN2(STXId, textBoxN2.Text);
```

4.4.8.3 Borland Delphi 6

//Declaration:

```
procedure STXWriteSetN2 (pNbr, N_value : PChar); stdcall;  external
'STX_lib.dll';
```

//Execution:

```
STXWriteSetN2(pNbr, PChar(EditN2.Text));
```

4.4.8.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * WriteSetN2)(char* PNbr, char* param);
WriteSetN2 STXWriteSetN2;


//Initialization (constructor) *.cpp file

STXWriteSetN2 = (WriteSetN2)GetProcAddress(dllInstance,
"STXWriteSetN2");
```

//Execution:

```
STXWriteSetN2(PNbr, EditSetN2->Text.c_str());
```

4.4.9 STXReadSetTemperature(PChar PortNumber)

Returns the target temperature in °C, as a string of characters.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.4.9.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadSetTemperature Lib "STX_lib" (ByVal
pNbr As String) As String
```

'Execution:

```
Dim Temperature As Single

Temperature = 40

If Temperature < CSng(STXReadSetTemperature(STXId)) then

        Alarm()

End If
```

4.4.9.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadSetTemperature(String pNbr);
```

//Execution:

```
float SetTemp;
SetTemp = System.Convert.ToSingle(STXReadSetTemperature(STXId));
```

4.4.9.3 Borland Delphi 6

//Declaration:

```
function STXReadSetTemperature(pNbr : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
var
  SetTemperature : Real;
begin

  SetTemperature := StrToFloat(STXReadSetTemperature('1'));

end;
```

4.4.9.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadSetTemperature)(char* PNbr);
ReadSetTemperature STXReadSetTemperature;


//Initialization (constructor) *.cpp file

STXReadSetTemperature =
(ReadSetTemperature)GetProcAddress(dllInstance,
"STXReadSetTemperature");
```

//Execution:

```
LbRTemperature->Caption = String(STXReadSetTemperature(PNbr));
```

4.4.10 STXReadSetHumidity(PChar PortNumber)

Returns the target humidity in percent, as a string of characters.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.4.10.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadSetHumidity Lib "STX_lib" (ByVal pNbr
As String) As String
```

'Execution:

```
Dim SetHumidity As Single

SetHumidity = Sng(STXReadSetHumidity(STXId))
```

4.4.10.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadSetHumidity(String pNbr);
```

//Execution:

```
float SetHum;

SetHum = System.Convert.ToSingle(STXReadSetHumidity(STXId));
```

4.4.10.3 Borland Delphi 6

//Declaration:

```
function STXReadSetHumidity(pNbr : PChar) : PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
var
  SetHumidity : Real;

begin

  SetHumidity := StrToFloat(STXReadSetHumidity(pNbr));

end;
```

4.4.10.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadSetHumidity)(char* PNbr);
ReadSetHumidity STXReadSetHumidity;


//Initialization (constructor) *.cpp file

STXReadSetHumidity = (ReadSetHumidity)GetProcAddress(dllInstance,
"STXReadSetHumidity");
```

//Execution:

```
LbRHumidity->Caption = String(STXReadSetHumidity(PNbr));
```

4.4.11 STXReadSetCO2(PChar PortNumber)

Returns the target CO2 concentration in percent, as a string of characters.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.4.11.1 *Visual Basic .NET (Visual Basic 6.0)*

'Declaration:

```
Private Declare Function STXReadSetCO2 Lib "STX_lib" (ByVal pNbr As
String) As String
```

'Execution:

```
Dim SetCO2 As Single

SetCO2 = Sng(STXReadSetCO2(STXId))
```

4.4.11.2 *Visual C# .NET*

//Declaration:

```
[DllImport("STX_lib")]

private extern static String STXReadSetCO2(String pNbr);
```

//Execution:

```
float SetCO2;

SetCO2 = System.Convert.ToSingle(STXReadSetCO2(STXId));
```

4.4.11.3 Borland Delphi 6

//Declaration:

```
function STXReadSetCO2(pNbr : PChar): PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
var
  SetCO2 : Real;
begin

  SetCO2 := StrToFloat(STXReadSetCO2(pNbr));

end;
```

4.4.11.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadSetCO2)(char* PNbr);
ReadSetCO2 STXReadSetCO2;


//Initialization (constructor) *.cpp file

STXReadSetCO2 = (ReadSetCO2)GetProcAddress(dllInstance,
"STXReadSetCO2");
```

//Execution:

```
LbRCO2->Caption = String(STXReadSetCO2(PNbr));
```

4.4.12 STXReadSetN2(PChar PortNumber)

Returns the target N2 concentration in percent, as a string of characters.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.4.12.1 *Visual Basic .NET (Visual Basic 6.0)*

'Declaration:

```
Private Declare Function STXReadSetN2 Lib "STX_lib" (ByVal pNbr As
String) As String
```

'Execution:

```
MsgBox(STXReadSetN2(STXId))
```

4.4.12.2 *Visual C# .NET*

//Declaration:

```
[DllImport("STX_lib")]

private extern static String STXReadSetN2(String pNbr);
```

//Execution:

```
float SetN2;

SetN2 = System.Convert.ToSingle(STXReadSetN2(STXId));
```

4.4.12.3 Borland Delphi 6

//Declaration:

```
function STXReadSetN2(pNbr : PChar) : PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
var
  SetN2 : Real;
begin

  SetN2 := StrToFloat(STXReadSetN2(pNbr));

end;
```

4.4.12.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadSetN2)(char* PNbr);
ReadSetN2 STXReadSetN2;


//Initialization (constructor) *.cpp file

STXReadSetN2 = (ReadSetN2)GetProcAddress(dllInstance,
"STXReadSetN2");
```

//Execution:

```
LbRN2->Caption = String(STXReadSetN2(PNbr));
```

4.5 Plate Handling Functions

4.5.1 STXIsOperationRunning(PChar PortNumber)

Checks whether previous long operation is running. The user can check whether long operations like STXLoadPlate, STXUnloadPlate, STXSetPlate, STXGetPlate, STXPickPlate or STXPlacePlate are running by means of usage this method.

Parameter: PortNumber - Number of a Serial Port.

Return values (PChar):

```
"1" - Previous long operation is still running

"0" - Previous long operation is completed or cancelled.
```

Examples:

4.5.1.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXIsOperationRunning Lib "STX_lib" (ByVal
pNbr As String) As String
```

'Execution:

```
if STXIsOperationRunning(STXId) = "1" then

        STXWaitForCompletion(STXId, "30000")

Else

        STXLoadPlate(STXId, TextBoxSlot.Text, TextBoxLevel.Text)

End If
```

4.5.1.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXIsOperationRunning(String pNbr);
```

<u>//Execution:</u>

```
if (STXIsOperationRunning(STXId) == "1")

{

      MessageBox.Show("Previous operation is still running!");

}
```

4.5.1.3 Borland Delphi 6

<u>//Declaration:</u>

```
function STXIsOperationRunning(pNbr : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

<u>//Execution:</u>

```
if STXIsOperationRunning(pNbr) = '0' then

STXLoadPlate(pNbr, '2','2');
```

4.5.1.4 Borland C++Builder

<u>//Declaration:</u>

```
//Prototype *.h file


typedef char* (__stdcall * IsOperationRunning)(char* PNbr);
IsOperationRunning STXIsOperationRunning;


//Initialization (constructor) *.cpp file

STXIsOperationRunning =
(IsOperationRunning)GetProcAddress(dllInstance,
"STXIsOperationRunning");
```

<u>//Execution:</u>

```
int Slot, Level;

Slot = 1;
Level = 1;

if (STXIsOperationRunning(pNbr) == "0")
{
  STXLoadPlate(pNbr, IntToStr(Slot).c_str(),
  IntToStr(Level).c_str());
}
```

4.5.2 STXWaitForCompletion(PChar PortNumber, mSecDelay)

Waits for completion of a long operation (STXLoadPlate, STXUnloadPlate, STXSetPlate, etc.). If long operation will be finished earlier than set time: mSecDelay, function is interrupted. This method can be used to organize the sequence of long handling operations like STXLoadPlate, STXUnloadPlate, etc.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.5.2.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXWaitForCompletion Lib "STX_lib" (ByVal pNbr As
String, ByVal Waiting As String)
```

'Execution:

```
STXUnLoadPlate(STXId, TextBoxSlot1.Text, TextBoxLevel1.Text);

STXWaitForCompletion(STXId, "15000");

STXLoadPlate(STXId, TextBoxSlot2.Text, TextBoxLevel2.Text);
```

4.5.2.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXWaitForCompletion(String pNbr, String
DalayMS);
```

//Execution:

```
STXUnloadPlate(STXId, textBoxSlot.Text, textBoxLevel.Text);

STXWaitForCompletion(STXId, "15000");

STXLoadPlate(STXId, textBoxSlot.Text, textBoxLevel.Text);
```

4.5.2.3 Borland Delphi 6

//Declaration:

```
procedure STXWaitForCompletion(pNbr, Delay : PChar); stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXUnloadPlate(pNbr, '1', '2');

STXWaitForCompletion(pNbr, '20000');

STXUnloadPlate(pNbr, '1', '3');

STXWaitForCompletion(pNbr, '20000');

STXUnloadPlate(pNbr, '1', '4');
```

4.5.2.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * WaitForCompletion)(char* PNbr, char*
param);
WaitForCompletion STXWaitForCompletion;


//Initialization (constructor) *.cpp file

STXWaitForCompletion = (WaitForCompletion)GetProcAddress(dllInstance,
"STXWaitForCompletion");
```

//Execution:

```
STXWaitForCompletion(pNbr, "20000");
STXUnloadPlate(pNbr,"2", "4");
STXWaitForCompletion(pNbr,"20000");
STXUnloadPlate(pNbr,"2", "5");
```

4.5.3 STXWaitForOperation(PChar PortNumber)

Waits for completion of a long operation (STXLoadPlate, STXUnloadPlate, STXSetPlate, etc.) and returns a result. This method can be used to organize the sequence of long handling operations like STXLoadPlate, STXUnloadPlate, etc…

Parameter: PortNumber - Number of a Serial Port.

Return values (PChar):

```
"1" - System is ready for next operation.

"0" - System is not ready due to an Error.
```

Examples:

4.5.3.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXWaitForOperation Lib "STX_lib" (ByVal
pNbr As String) As String
```

'Execution:

```
STXUnLoadPlate(STXId, TextBoxSlot1.Text, TextBoxLevel1.Text)

If STXWaitForOperation(STXId) = "1" then

        STXLoadPlate(STXId, TextBoxSlot2.Text, TextBoxLevel2.Text)

Else

        If STXReadErrorFlag(STXId) ="1" then

              MsgBox("Error - "+STXReadErrorCode(STXId))

End If
```

4.5.3.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXWaitForOperation(String pNbr);
```

//Execution:

```
STXUnloadPlate(STXId, textBoxSlot.Text, textBoxLevel.Text);

if (STXWaitForOperation(STXId) == "1")
{

        STXLoadPlate(STXId, textBoxSlot.Text, textBoxLevel.Text);

}
else
{

        Alarm();

}
```

4.5.3.3 Borland Delphi 6

//Declaration:

```
function STXWaitForOperation(pNbr : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXUnloadPlate(pNbr, '1', '2');

STXWaitForOperation(pNbr);

STXUnloadPlate(pNbr, '1', '3');

STXWaitForOperation(pNbr);

STXUnloadPlate(pNbr, '1', '4');
```

4.5.3.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * WaitForOperation)(char* PNbr);
WaitForOperation STXWaitForOperation;


//Initialization (constructor) *.cpp file

STXWaitForOperation = (WaitForOperation)GetProcAddress(dllInstance,
"STXWaitForOperation");
```

//Execution:

```
int Slot = 1;

for(int i = 1; i<10; i++)
{
if(String(STXWaitForOperation(pNbr)) == "1")
{
  STXLoadPlate(pNbr, IntToStr(Slot).c_str(), IntToStr(i).c_str());
}
}
```

4.5.4 STXLoadPlate(PChar PortNumber, Slot, Level)

Loads a plate from the Transfer Station to specified Slot and Level.

Parameters:

```
PortNumber - Number of a Serial Port.

Slot - plate slot position.

Level - plate level position.
```

Return values (PChar):

```
"1" - STXLoadPlate operation is started.

"0" - Device is not initialised.

"2" - A wrong value of slot or level.

"3" - User's door is opened.

"4" - Previous long operation is not finished.
```

Examples:

4.5.4.1 Visual Basic .NET (Visual Basic 6.0)

<u>'Declaration:</u>

```
Private Declare Function STXLoadPlate Lib "STX_lib" (ByVal pNbr As
String, ByVal Slot As String, ByVal Level As String) As String
```

<u>'Execution:</u>

```
STXLoadPlate(STXId, "1","1")

STXWaitForOperation(STXId)

If STXLoadPlate(STXId, "1","2") = "1" then

     STXWaitForOperation(STXId)
     STXLoadPlate(STXId, "3", "3")

End If
```

4.5.4.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXLoadPlate(String pNbr, String Slot,
String Level);
```

//Execution:

```
if (STXLoadPlate(STXId, textBoxSlot.Text, textBoxLevel.Text) == "1")
{

  STXWaitForOperation(STXId);

  STXLoadPlate(STXId, textBoxSlot1.Text, textBoxLevel1.Text);

}
else
{

  Alarm();
}
```

4.5.4.3 Borland Delphi 6

//Declaration:

```
function STXLoadPlate(pNbr, Slot, Level : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
var
  i : integer;
begin

  for i := 1 to 10 do
  begin
    if STXLoadPlate(pNbr, '5',PChar(IntToStr(i))) = '1' then
    STXWaitForOperation(pNbr);
  end;

end;
```

4.5.4.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * LoadPlate)(char* PNbr, char* param1, char*
param2);
LoadPlate STXLoadPlate;


//Initialization (constructor) *.cpp file

STXLoadPlate = (LoadPlate)GetProcAddress(dllInstance,
"STXLoadPlate");
```

//Execution:

```
// STXLoadPlate(pNbr, EditSlot->Text.c_str(), EditLevel-
>Text.c_str());
STXLoadPlate(pNbr, "2", "2");
```

4.5.5 STXUnloadPlate(PChar PortNumber, Slot, Level)

Unloads a plate from the specified Slot and Level to the Transfer Station

Parameters:

```
PortNumber - Number of a Serial Port.
Slot - plate slot position.
Level - plate level position.
```

Return values (PChar):

```
"1" - STXUnloadPlate is started.
"0" - Device is not initialised.
"2" - A wrong value of slot or level.
"3" - User's door is opened.
"4" - Previous long operation is not finished.
```

Examples:

4.5.5.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXUnloadPlate Lib "STX_lib" (ByVal pNbr As
String, ByVal Slot As String, ByVal Level As String) As String
```

'Execution:

```
STXUnloadPlate(STXId, "1","1")
```

4.5.5.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXUnloadPlate(String pNbr, String Slot,
 String Level);
```

//Execution:

```
STXUnloadPlate(STXId, textBoxSlot.Text, textBoxLevel.Text);


if (STXWaitForOperation(STXId) == "1")
{

  MessageBox.Show("Plate is unloaded!");

}
```

4.5.5.3 Borland Delphi 6

//Declaration:

```
function STXUnloadPlate (pNbr, Slot, Level : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
var
i, ,j : integer;
begin

j := 1;

  for i := 1 to 10 do
  begin
    if STXUnloadPlate (pNbr, PChar(IntToStr(j)),PChar(IntToStr(i)))
= '1' then
    STXWaitForOperation(pNbr);
  end;

end;
```

4.5.5.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * UnloadPlate)(char* PNbr, char* param1,
char* param2);
UnloadPlate STXUnloadPlate;


//Initialization (constructor) *.cpp file

STXUnloadPlate = (UnloadPlate)GetProcAddress(dllInstance,
"STXUnloadPlate");
```

//Execution:

```
STXUnloadPlate(pNbr, "3", "4");
```

4.5.6 STXSetPlate(PChar PortNumber)

Removes a plate from the shovel to the transfer station.

Parameter: PortNumber - Number of a Serial Port.

Return values (PChar):

```
"1" - STXSetPlate operation is started.

"0" - Device is not initialised.

"3" - User's door is opened.

"4" - Previous long operation is not finished.
```

Examples:

4.5.6.1 *Visual Basic .NET (Visual Basic 6.0)*

'Declaration:

```
Private Declare Function STXSetPlate Lib "STX_lib" (ByVal pNbr As
String) As String
```

'Execution:

```
STXSetPlate(STXId)

STXWaitForOperation(STXId)

STXSwap(STXId)
```

4.5.6.2 *Visual C# .NET*

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXSetPlate(String pNbr);
```

//Execution:

```
STXSetPlate(STXId);

if (STXWaitForOperation(STXId) == "1")
{

  STXSwap(STXId);

}
```

4.5.6.3 Borland Delphi 6

//Declaration:

```
function STXSetPlate(pNbr : PChar) : PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
STXSetPlate(pNbr);
```

4.5.6.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * SetPlate)(char* PNbr);
SetPlate STXSetPlate;


//Initialization (constructor) *.cpp file

STXSetPlate = (SetPlate)GetProcAddress(dllInstance, "STXSetPlate");
```

//Execution:

```
STXSetPlate(PNbr);
```

4.5.7 STXGetPlate(PChar PortNumber)

Gets a Plate from the Transfer Station to the shovel.

Parameter: PortNumber - Number of a Serial Port.

Return values (PChar):

```
"1" - STXSetPlate operation is  started.

"0" - Device is not initialised.

"3" - User's door is opened.

"4" - Previous long operation is not finished.
```

Examples:

4.5.7.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXGetPlate Lib "STX_lib" (ByVal pNbr As
String) As String
```

'Execution:

```
Dim Slot As Integer
Dim Level As Integer

Slot = 1

Level = 5

STXGetPlate(STXId)

STXWaitForOperation(STXId)

STXPlacePlate(STXId, Str(Slot), Str(Level))
```

4.5.7.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXGetPlate(String pNbr);
```

//Execution:

```
STXGetPlate(STXId);

if (STXWaitForOperation(STXId) == "1")
{

        STXPlacePlate(STXId, "3", "6" );

}
```

4.5.7.3 Borland Delphi 6

//Declaration:

```
function STXGetPlate(pNbr : PChar): PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
STXGetPlate(pNbr);

if STXWaitForOperation(pNbr) = '1' then
STXPlacePlate(pNbr, '5', '6');
```

4.5.7.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * GetPlate)(char* PNbr);
GetPlate STXGetPlate;


//Initialization (constructor) *.cpp file

STXGetPlate = (GetPlate)GetProcAddress(dllInstance, "STXGetPlate");
```

//Execution:

```
STXGetPlate(PNbr);
```

4.5.8 STXPickPlate(PChar PortNumber, Slot, Level)

Picks plate by shovel from the specified slot and level for further placing.

Parameters:

`PortNumber - Number of a Serial Port.`

`Slot - plate slot position.`

`Level - plate level position.`

Return values (PChar):

`"1" - STXUnloadPlate operation is started.`

`"0" - Device is not initialised.`

`"2" - A wrong value of slot or level.`

`"3" - User's door is opened.`

`"4" - Previous long operation is not finished.`

Examples:

4.5.8.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXPickPlate Lib "STX_lib" (ByVal pNbr As
String, ByVal Slot As String, ByVal Level As String) As String
```

'Execution:

```
Dim Slot As Integer
Dim Level As Integer

Slot = 1

Level = 5


STXPickPlate(STXId, Str(Slot),Str(Level))

STXWaitForOperation(STXId)

STXSetPlate(STXId)

STXPlacePlate(STXId, Str(Slot),Str(Level))
```

4.5.8.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXPickPlate(String pNbr, String Slot,
  String Level);
```

//Execution:

```
STXPickPlate(STXId, textBoxSlot.Text, textBoxLevel.Text);

if (STXWaitForOperation(STXId) == "1")
{
        STXSetPlate(STXId);

}
```

4.5.8.3 Borland Delphi 6

//Declaration:

```
function STXPickPlate(pNbr, Slot, Level : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXPickPlate (pNbr, '5', '6');

if STXWaitForOperation(pNbr) = '1' then
STXSetPlate(pNbr);
```

4.5.8.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * PickPlate)(char* PNbr, char* param1, char*
param2);
PickPlate STXPickPlate;


//Initialization (constructor) *.cpp file

STXPickPlate = (PickPlate)GetProcAddress(dllInstance,
"STXPickPlate");
```

//Execution:

```
int Slot, Level;

Slot = 1;
Level = 2;

STXPickPlate(pNbr, IntToStr(Slot).c_str(), IntToStr(Level).c_str());
STXWaitForOperation(pNbr);
STXSetPlate(pNbr);
```

4.5.9 STXPlacePlate(PChar PortNumber, Slot, Level)

Places plate from the shovel to the specified slot and level.

Parameters:

```
PortNumber - Number of a Serial Port.
Slot - plate slot position.
Level - plate level position.
```

Return values (PChar):

```
"1" - STXUnloadPlate operation is started.
"0" - Device is not initialised.
"2" - A wrong value of slot or level.
"3" - User's door is opened.
"4" - Previous long operation is not finished.
```

Examples:

4.5.9.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXPlacePlate Lib "STX_lib" (ByVal pNbr As
String, ByVal Slot As String, ByVal Level As String) As String
```

'Execution:

```
Dim Slot1 As Integer
Dim Level1 As Integer
Dim Slot2 As Integer
Dim Leve2l As Integer

Slot1 = 1

Level1 = 5

Slot2 = 2

Level2 = 8

STXPickPlate(STXId, Str(Slot1),Str(Level1))

STXWaitForOperation(STXId)

STXPlacePlate(STXId, Str(Slot2),Str(Level2))
```

4.5.9.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXPlacePlate(String pNbr, String Slot,
  String Level);
```

//Execution:

```
STXGetPlate(STXId);

STXWaitForOperation(STXId);

STXPlacePlate(STXId, System.Convert.ToString(Slot),
                          System.Convert.ToString(Level));
```

4.5.9.3 Borland Delphi 6

//Declaration:

```
function STXPlacePlate(pNbr, Slot, Level : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXGetPlate(pNbr);

if STXWaitForOperation(pNbr) = '1' then
STXPlacePlate(pNbr, '5', '6');
```

4.5.9.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * PlacePlate)(char* PNbr, char* param1,
char* param2);
PlacePlate STXPlacePlate;


//Initialization (constructor) *.cpp file

STXPlacePlate = (PlacePlate)GetProcAddress(dllInstance,
"STXPlacePlate");
```

//Execution:

```
int Slot, Level;

Slot = 1;
Level = 2;

if(String(STXGetPlate(pNbr)) == "1")
if(String(STXWaitForOperation(pNbr)) == "1")
STXPlacePlate(pNbr, IntToStr(Slot).c_str(), IntToStr(Level).c_str());
```

4.5.10 STXReadShovelDetector(PChar PortNumber)

Reads whether plate is present on the Shovel. If Plate Shovel Detector is not assigned with function STXSetPlateShovelDetector, this function returns value "0" by default.

Parameter: PortNumber - Number of a Serial Port.

Return values (PChar):

```
"1" - Plate presents on the Shovel.

"0" - Plate doesn't present on the Shovel.
```

Examples:

4.5.10.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXSetPlate Lib "STX_lib" (ByVal pNbr As
String) As String
```

'Execution:

```
STXSetPlateShovelDetector(STXId, "1")

If STXReadShovelDetector(STXId) = "1" Then

  STXSetPlate(STXId)

End If
```

4.5.10.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadShovelDetector(String pNbr);
```

//Execution:

```
STXSetPlateShovelDetector(STXId, "1");

If (STXReadShovelDetector(STXId) == "1")
{
  STXSetPlate(STXId);
}
```

4.5.10.3 Borland Delphi 6

//Declaration:

```
function STXReadShovelDetector(pNbr : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXSetPlateShovelDetector(pNbr, "1");

if STXReadShovelDetector(pNbr) = '1' then
  ShowMessage(pNbr, 'Plate presents on the Shovel')

else
  ShowMessage(pNbr, 'Plate doesn''t present on the Shovel');
```

4.5.10.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadShovelDetector)(char* PNbr);
ReadShovelDetector STXReadShovelDetector;


//Initialization (constructor) *.cpp file

STXReadShovelDetector =
(ReadShovelDetector)GetProcAddress(dllInstance,
"STXReadShovelDetector");
```

//Execution:

```
STXSetPlateShovelDetector(pNbr, "1");

if(String(STXReadShovelDetector(pNbr)) == "1")
{
  LbShovelDetect->Caption = "Plate is present!";
  STXSetPlate(pNbr);
}
else
{
  LbShovelDetect->Caption = "Plate does not present!";
}
```

4.5.11 STXIsPlateAtLocation(PChar PortNumber, Slot, Level)

Reads whether plate is present at Location (Slot, Level). If Plate Detector is not assigned with function STXSetPlateDetector, this function returns value ”0” by default.

Parameters:

`PortNumber - Number of a Serial Port.`

`Slot - plate slot position.`

`Level - plate level position.`

Return values (PChar):

`"1" - Plate is on the Location (Slot, Level).`

`"0" - Plate is not on the Location (Slot, Level).`

`"2" - Device is not initialised.`

Examples:

4.5.11.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXIsPlateAtLocation Lib "STX_lib" (ByVal
pNbr As String, ByVal Slot As String, ByVal Level As String) As
String
```

'Execution:

```
Dim Slot As Integer
Dim Level As Integer

STXSetPlateDetector(STXId, "1")

If STXIsPlateAtLocation(STXId, Str(Slot),Str(Level)) = "0" Then

        STXPlacePlate(STXId, Str(Slot),Str(Level))

End If
```

4.5.11.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXIsPlateAtLocation(String pNbr, String
Slot, String Level);
```

//Execution:

```
STXSetPlateDetector(STXId, "1");

If (STXIsPlateAtLocation(STXId, System.Convert.ToString(Slot),
                          System.Convert.ToString(Level)) == "1")
{

  STXGetPlate(STXId);

}
```

4.5.11.3 Borland Delphi 6

//Declaration:

```
function STXIsPlateAtLocation(pNbr, Slot, Level : PChar) : PChar;
stdcall;  external 'STX_lib.dll';
```

//Execution:

```
STXSetPlateDetector(pNbr, '1');

if STXIsPlateAtLocation(pNbr, PChar(EditSlot.Text),
PChar(EditLevel.Text)) = '1' then
STXUnloadPlate(pNbr, PChar(EditSlot.Text), PChar(EditLevel.Text));
```

4.5.11.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * IsPlateAtLocation)(char* PNbr, char*
param1, char* param2);
IsPlateAtLocation STXIsPlateAtLocation;


//Initialization (constructor) *.cpp file

STXIsPlateAtLocation = (IsPlateAtLocation)GetProcAddress(dllInstance,
"STXIsPlateAtLocation");
```

//Execution:

```
STXSetPlateDetector(pNbr, "1");

if(STXIsPlateAtLocation(pNbr, EditSlot->Text.c_str(),
  EditLevel->Text.c_str()) == "1")
{
  STXPickPlate(pNbr, EditSlot->Text.c_str(),
  EditLevel-> Text.c_str());
}
else
{
  LbIsLoc->Caption = "No Plate!";
}
```

4.5.12 STXImportPlate(PChar PortNumber, Slot, Level)

Loads a plate from the Transfer Station to specified Slot and Level.

This function carries out the following sub-functions in the following order:

1. Checks Plate at Transfer Station.
2. Gets plate from Transfer Station.
3. Checks that the specified location is empty.
   If it is empty then
4a. Places plate in specified location.
    Else
4b. Returns plate to Transfer Station.

To correct usage of this function the presence of Plate Detector must be assigned by means of function STXSetPlateTStatDetector and the presence of Plate Detector at Transfer Station by means of function STXSetPlateTStatDetector.

Parameters:

```
PortNumber - Number of a Serial Port.

Slot - plate slot position.

Level - plate level position.
```

Return values (PChar):

```
"1" - Function STXImportPlate has been completed.

"0" - Device is not initialised.

"2" - Error: there is no Plate at Transfer station.

"3" - Error: gets plate from Transfer Station.

"4" - Error: places Plate to the Stacker.

"5" - Error: specified Slot and Level position is not empty .

"6" - Previous long operation is not finished.
```

Examples:

4.5.12.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXImportPlate Lib "STX_lib" (ByVal pNbr As
String, ByVal Slot As String, ByVal Level As String) As String
```

'Execution:

```
Dim Result As String

Dim BCR As String

Result = STXImportPlate(STXId, TextBoxSlot.Text, TextBoxLevel.Text)

If Result ="1" then

  BCR  = STXReadBarcode(STXId, TextBoxSlot.Text, TextBoxLevel.Text)

End If
```

4.5.12.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXImportPlate(String pNbr, String Slot,
String Level);
```

//Execution:

```
int Slot = 1;

for(int Level = 1; Level <=
System.Convert.ToInt16(STXReadMaxLevel(STXId)); Level++)
{

      STXImportPlate(STXId, System.Convert.ToString(Slot),
                           System.Convert.ToString(Level));
}
```

4.5.12.3 Borland Delphi 6

//Declaration:

```
function STXImportPlate (pNbr, Slot, Level : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXImportPlate(pNbr, PChar(EditSlot.Text), PChar(EditLevel.Text));
```

4.5.12.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ImportPlate)(char* PNbr, char* Slot, char*
Level);
ImportPlate STXImportPlate;


//Initialization (constructor) *.cpp file

STXImportPlate = (ImportPlate)GetProcAddress(dllInstance,
"STXImportPlate");
```

//Execution:

```
STXImportPlate(pNbr, EditSlot->Text.c_str(),
EditLevel->Text.c_str());
```

4.5.13 STXExportPlate(PChar PortNumber, Slot, Level)

Unloads a plate from the specified Slot and Level to the Transfer Station.

This function carries out the following sub-functions in the following order:

1. Checks whether plate is present at specified location.
2. Gets Plate from specified location.
3. Checks that the Transfer station is empty.
   If it is empty then
4a. Places plate at Transfer Station.
   Else
4b. Returns plate to Stacker.

To correct usage of this function the presence of Plate Detector must be assigned by means of function STXSetPlateTStatDetector and the presence of Plate Detector at Transfer Station by means of function STXSetPlateTStatDetector.

Parameters:

`PortNumber - Number of a Serial Port.`

`Slot - plate slot position.`

`Level - plate level position.`

Return values (PChar):

`"1" - Function STXExportPlate has been completed.`

`"0" - Device is not initialised.`

`"2" - Error: there is no Plate at Stacker.`

`"3" - Error: gets plate from Stacker.`

`"4" - Error: places Plate to Transfer Station.`

`"5" - Error: transfer Station is not empty.`

`"6" - Previous long operation is not finished.`

Examples:

4.5.13.1 *Visual Basic .NET (Visual Basic 6.0)*

'Declaration:

```
Private Declare Function STXExportPlate Lib "STX_lib" (ByVal pNbr As
String, ByVal Slot As String, ByVal Level As String) As String
```

'Execution:

```
STXExportPlate (STXId, TextBoxSlot.Text, TextBoxLevel.Text)
```

4.5.13.2 *Visual C# .NET*

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXExportPlate(String pNbr, String Slot,
 String Level);
```

//Execution:

```
int Slot = 1;

for(int Level = 1; Level <=
System.Convert.ToInt16(STXReadMaxLevel(STXId)); Level++)
{

        STXExportPlate(STXId, System.Convert.ToString(Slot),
                           System.Convert.ToString(Level));
}
```

4.5.13.3 *Borland Delphi 6*

//Declaration:

```
function STXExportPlate (pNbr, Slot, Level : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXExportPlate (pNbr, PChar(EditSlot.Text), PChar(EditLevel.Text));
```

4.5.13.4 *Borland C++Builder*

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ExportPlate)(char* PNbr, char* Slot, char*
Level);
ExportPlate STXExportPlate;


//Initialization (constructor) *.cpp file

STXExportPlate = (ExportPlate)GetProcAddress(dllInstance,
"STXExportPlate");
```

//Execution:

```
STXExportPlate(pNbr, EditSlot->Text.c_str(),
EditLevel->Text.c_str());
```

4.5.14 STXMovePlate(PChar PortNumber, Slot1, Level1, Slot2, Level2)

Moves a plate from (Slot1, Level1) position to (Slot2, Level2) position.

This function consists of following sub-steps:

1. Checks whether a plate is present at location (Slot1, Level1).
2. Gets Plate from location (Slot1, Level1).
3. System checks whether location (Slot2; Level2) is empty.
   If it is empty then

4a. Places plate to the location (Slot2, Level2).
   Else
4b. Returns plate to the location (Slot1, Level1).

To correct usage of this function the presence of Plate Detector must be assigned by means of function STXSetPlateTStatDetector

Parameters:

```
PortNumber - Number of a Serial Port.

Slot1 - plate slot position of source.

Level1 - plate level position of source.

Slot2 - plate slot position of target.

Level3 - plate level position of target.
```

Return values (PChar):

```
"1" - Function STXMovePlate has been completed.

"0" - Device is not initialised.

"2" - Error: there is no Plate at location (Slot1, Level1).

"3" - Error: gets Plate from location (Slot1, Level1).

"4" - Error: places Plate to location (Slot2, Level2).

"5" - Error: location (Slot2, Level2) is not empty.

"6" - Previous long operation is not finished.
```

Examples:

4.5.14.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXMovePlate Lib "STX_lib" (ByVal pNbr As
String, ByVal Slot1 As String, ByVal Level1 As String, ByVal Slot2 As
String, ByVal Level2 As String) As String
```

'Execution:

```
Dim Result As String

Result = STXMovePlate (STXId, "1", "1", "2", "2")

If Result = "6" then

  STXWaitForCompletion(STXId, "30000")

Else

  STXMovePlate (STXId, "1", "1", "2", "2")

End If
```

4.5.14.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXMovePlate(String pNbr, String Slot1,
String Level1, String Slot2, String Level2);
```

//Execution:

```
if (STXMovePlate (STXId, "1", "1", "2", "2") == "1")
{
  lbStatus.Text = "Plate has been removed";
}
```

4.5.14.3 Borland Delphi 6

//Declaration:

```
function STXMovePlate(pNbr, Slot, Level : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXMovePlate(pNbr, PChar(EditSlot.Text), PChar(EditLevel.Text),
    PChar(EditSlot1.Text), PChar(EditLevel1.Text));
```

4.5.14.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * MovePlate)(char* PNbr, char* Slot1, char*
Level1, char* Slot2, char* Level2);
MovePlate STXMovePlate;


//Initialization (constructor) *.cpp file

STXMovePlate = (MovePlate)GetProcAddress(dllInstance,
"STXMovePlate");
```

//Execution:

```
STXMovePlate(pNbr, EditSlot1->Text.c_str(),
EditLevel1->Text.c_str(), EditSlot2->Text.c_str(),
EditLevel2->Text.c_str());
```

4.6 Shaker Functions

4.6.1 STXWriteShakerSpeed(Pchar PortNumber, Speed)

Writes shaker speed settings value.

Parameters:

`PortNumber - Number of a Serial Port.`

`Speed - Shaker speed (range 1...50), PChar (e.g. “15”)`

Examples:

4.6.1.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXWriteShackerSpeed Lib "STX_lib" (ByVal pNbr As
String, ByVal SetSpeed As String)
```

'Execution:

```
STXWriteShackerSpeed(STXId, "30")
```

4.6.1.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXWriteShakerSpeed(String pNbr, String
Speed);
```

//Execution:

```
//STXWriteShakerSpeed(STXId, "30");

STXWriteShakerSpeed(STXId, textBoxShSpeed.Text);
```

4.6.1.3 Borland Delphi 6

//Declaration:

```
procedure STXWriteShakerSpeed(pNbr, Speed : PChar); stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXWriteShakerSpeed(pNbr, '30');
```

4.6.1.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * WriteShakerSpeed)(char* PNbr, char*
ShkSpeed);
WriteShakerSpeed STXWriteShakerSpeed;


//Initialization (constructor) *.cpp file

STXWriteShakerSpeed = (WriteShakerSpeed)GetProcAddress(dllInstance,
"STXWriteShakerSpeed");
```

//Execution:

```
STXWriteShakerSpeed(pNbr, "30");
```

4.6.2 STXActivateShaker(PChar PortNumber)

Switches shaker on.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.6.2.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXActivateShaker Lib "STX_lib" (ByVal pNbr As
String)
```

'Execution:

```
STXActivateShaker(STXId)
```

4.6.2.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXActivateShaker(String pNbr);
```

//Execution:

```
STXActivateShaker(STXId);
```

4.6.2.3 Borland Delphi 6

//Declaration:

```
procedure STXActivateShaker(pNbr : PChar) ; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
STXActivateShaker(pNbr);
```

4.6.2.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * ActivateShaker)(char* PNbr);
ActivateShaker STXActivateShaker;


//Initialization (constructor) *.cpp file

STXActivateShaker = (ActivateShaker)GetProcAddress(dllInstance,
"STXActivateShaker");
```

//Execution:

```
STXActivateShaker(pNbr);
```

4.6.3 STXDeactivateShaker(PChar PortNumber)

Switches shaker off.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.6.3.1 *Visual Basic .NET (Visual Basic 6.0)*

'Declaration:

```
Private Declare Sub STXDeactivateShaker Lib "STX_lib" (ByVal pNbr As
String)
```

'Execution:

```
STXDeactivateShaker(STXId)
```

4.6.3.2 *Visual C# .NET*

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXDeactivateShaker(String pNbr);
```

//Execution:

```
STXDeactivateShaker(STXId);
```

4.6.3.3 *Borland Delphi 6*

//Declaration:

```
procedure STXDeactivateShaker(pNbr : PChar) ; stdcall; external
'STX_lib.dll';
```

//Execution:

```
STXDeactivateShaker(pNbr);
```

4.6.3.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef void (__stdcall * DeactivateShaker)(char* PNbr);
DeactivateShaker STXDeactivateShaker;


//Initialization (constructor) *.cpp file

STXDeactivateShaker = (DeactivateShaker)GetProcAddress(dllInstance,
"STXDeactivateShaker");
```

//Execution:

```
STXDeactivateShaker(pNbr);
```

4.6.4 STXReadShakerSpeed(PChar PortNumber)

Returns shaker speed value (applies on systems with shaker option only), as a string of characters.

Parameter: PortNumber - Number of a Serial Port.

Examples:

4.6.4.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadShakerSpeed Lib "STX_lib" (ByVal pNbr
As String) As String
```

'Execution:

```
Dim Speed As Integer

Speed = Int(STXReadShakerSpeed(STXId))
```

4.6.4.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadShakerSpeed(String pNbr);
```

//Execution:

```
lbShakerSpeed.Text = STXReadShakerSpeed(STXId);
```

4.6.4.3 Borland Delphi 6

//Declaration:

```
function STXReadShakerSpeed(pNbr : PChar) : PChar; stdcall;  external
'STX_lib.dll';
```

//Execution:

```
LbSpeed.Caption := STXReadShakerSpeed(pNbr);
```

4.6.4.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadShakerSpeed)(char* PNbr);
ReadShakerSpeed STXReadShakerSpeed;


//Initialization (constructor) *.cpp file

STXReadShakerSpeed = (ReadShakerSpeed)GetProcAddress(dllInstance,
"STXReadShakerSpeed");
```

//Execution:

```
LbSpeed->Caption = String(STXReadShakerSpeed(pNbr));
```

4.7 Swap Station Functions

4.7.1 STXSwap(PChar PortNumber)

Rotate the swap station on 180 degree.

Parameter: PortNumber - Number of a Serial Port.

Return values (PChar):

```
"1" - Function STXSwap has been completed.

"0" - Error.
```

Examples:

4.7.1.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXSwap Lib "STX_lib" (ByVal pNbr As String)
```

'Execution:

```
STXSwap(STXId)
```

4.7.1.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXSwap(String pNbr);
```

//Execution:

```
STXSetPlate(STXId);

if (STXWaitForOperation(STXId) == "1")
{

  STXSwap(STXId);

}
```

4.7.1.3 Borland Delphi 6

//Declaration:

```
procedure STXSwap(pNbr : PChar); stdcall;  external 'STX_lib.dll';
```

//Execution:

```
STXSwap(pNbr);
```

4.7.1.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file

typedef char* (__stdcall * Swap)(char* PNbr);
Swap STXSwap;

//Initialization (constructor) *.cpp file

STXSwap = (Swap)GetProcAddress(dllInstance, "STXSwap");
```

//Execution:

```
STXSwap(pNbr);
```

4.7.2 STXSwapBack(PChar PortNumber)

Rotate the swap station back to home position.

Parameter: PortNumber - Number of a Serial Port.

Return values (PChar):

```
"1" - Function STXSwapBack has been completed.

"0" - Error.
```

Examples:

4.7.2.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXSwapBack Lib "STX_lib" (ByVal pNbr As String)
```

'Execution:

```
STXSwapBack(STXId)
```

4.7.2.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static void STXSwapBack (String pNbr);
```

//Execution:

```
STXSwapBack (STXId);
```

4.7.2.3 Borland Delphi 6

//Declaration:

```
procedure STXSwapBack(pNbr : PChar); stdcall;  external
'STX_lib.dll';
```

//Execution:

```
STXSwapBack(pNbr : PChar);
```

4.7.2.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * SwapBack)(char* PNbr);
SwapBack STXSwapBack;


//Initialization (constructor) *.cpp file

STXSwapBack = (SwapBack)GetProcAddress(dllInstance, "STXSwapBack");
```

//Execution:

```
STXSwapBack(pNbr);
```

4.7.3 STXReadTransferStationDetector(PChar PortNumber)

Reads whether plate is present on the Transfer Station. If Plate Station Detector is not assigned with function STXSetPlateTStatDetector, this function returns value ”0” by default.

Parameter: PortNumber - Number of a Serial Port.

Return values (PChar):

```
"1" - Plate is on the Transfer Station.

"0" - Plate is not on the Transfer Station.
```

Examples:

4.7.3.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadTransferStationDetector Lib "STX_lib"
(ByVal pNbr As String) As String
```

'Execution:

```
STXSetPlateTStatDetector(STXId, "1")

If STXReadTransferStationDetector(STXId) = "1" Then

        STXGetPlate(STXId)

End If
```

4.7.3.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadTransferStationDetector(String
pNbr);
```

//Execution:

```
STXSetPlateTStatDetector(STXId, "1");

If(STXReadTransferStationDetector(STXId) == "1")
{

        STXGetPlate(STXId);

}
```

4.7.3.3 Borland Delphi 6

//Declaration:

```
function STXReadTransferStationDetector(pNbr : PChar) : PChar;
stdcall;  external 'STX_lib.dll';
```

//Execution:

```
STXSetPlateTStatDetector(pNbr, '1');

if STXReadTransferStationDetector(pNbr) = '1' then
  STXLoadPlate(pNbr,'1','1');
```

4.7.3.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadTransferStationDetector)(char* PNbr);
ReadTransferStationDetector STXReadTransferStationDetector;


//Initialization (constructor) *.cpp file

STXReadTransferStationDetector =
(ReadTransferStationDetector)GetProcAddress(dllInstance,
"STXReadTransferStationDetector");
```

//Execution:

```
if (STXReadTransferStationDetector(pNbr)) == "1"
STXLoadPlate(pNbr,"1","1");
```

4.7.4 STXReadTransferStation2Detector(PChar PortNumber)

Reads whether plate is present on the second Transfer Station. If Second Plate Station Detector is not assigned with function STXSetPlateTStatDetector2, this function returns value ”0” by default.

Parameter: PortNumber - Number of a Serial Port.

Return values (PChar):

```
"1" - Plate is on the Transfer Station.

"0" - Plate is not on the Transfer Station.
```

Examples:

4.7.4.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadTransferStation2Detector Lib
"STX_lib" (ByVal pNbr As String) As String
```

'Execution:

```
STXSetPlateTStatDetector2(STXId, "1")

If STXReadTransferStation2Detector(STXId) = "0" Then

        STXSwapBack(STXId)

        STXSetPlate(STXId)

End If
```

4.7.4.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadTransferStation2Detector(String
pNbr);
```

//Execution:

```
STXSetPlateTStatDetector2 (STXId"1");

If(STXReadTransferStation2Detector(STXId) == "1")
{
        STXSwapBack(STXId);

        STXSetPlate(STXId);
}
```

4.7.4.3 Borland Delphi 6

//Declaration:

```
function STXReadTransferStation2Detector(pNbr : PChar) : PChar;
stdcall;  external 'STX_lib.dll';
```

//Execution:

```
STXSetPlateTStatDetector2(pNbr, '1');

if STXReadTransferStation2Detector(pNbr) = '1' then

begin

  STXSwapBack(pNbr)
  STXLoadPlate(pNbr, '1','1');

end;
```

4.7.4.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadTransferStation2Detector)(char* PNbr);
ReadTransferStation2Detector STXReadTransferStation2Detector;


//Initialization (constructor) *.cpp file

STXReadTransferStation2Detector =
(ReadTransferStation2Detector)GetProcAddress(dllInstance,
"STXReadTransferStation2Detector");
```

//Execution:

```
if (STXReadTransferStation2Detector(pNbr)) == "1"
STXLoadPlate(pNbr,"1","1");
```

4.8 Barcode Reader Functions

4.8.1 STXBCRPath(PChar PortNumber, PortName)

Sets the name of Serial Port for the Barcode Reader.

Parameter: PortName – Name of Serial Port (e.g. “1”)

Return values (PChar):

```
"1" - Serial Port is successfully opened.

"0" - Error of opening Serial Port.

"2" - Error: wrong name of the Serial Port.

"3" - Error: of opening Serial Port (serial Port is already opened).
```

Examples:

4.8.1.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXBCRPath Lib "STX_lib" (ByVal pNbr As String,
ByVal BCRPortNAme As String)
```

'Execution:

```
'STXBCRPath (STXId, "3")

STXBCRPath (STXId, TextBoxBCRPort.Text)
```

4.8.1.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXBCRPath(String pNbr, String BCRPath);
```

//Execution:

```
//STXBCRPath (STXId, "3");

STXBCRPath (STXId, textBoxBCRPort.Text);
```

4.8.1.3 Borland Delphi 6

//Declaration:

```
function STXBCRPath(pNbr, BCRPortName : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
STXBCRPath(pNbr, '2');
```

4.8.1.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * BCRPath)(char* STXId, char* bPath);
BCRPath STXBCRPath;


//Initialization (constructor) *.cpp file

STXBCRPath = (BCRPath)GetProcAddress(dllInstance, "STXBCRPath");
```

//Execution:

```
STXBCRPath(pNbr, "2"); //COM2
```

4.8.2 STXReadBarcode(PChar PortNumber, Slot, Level)

Reads the barcode of a plate at specified location. Returns the barcode as string of characters.

Parameters:

```
PortNumber - Number of a Serial Port.

Slot - plate slot position.

Level - plate level position.
```

Return values (PChar):

```
"BCRError" - Barcode reader is not initialised.

"InitError" - StoreX is not initialized.

"No Plate" - There is no Plate at the specified position
(If Plate Detector is assigned with function STXSetPlateDetector).

"No Barcode" - There is no Barcode on the Plate.
```

Examples:

4.8.2.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Function STXReadBarcode Lib "STX_lib" (ByVal pNbr As
String, ByVal Slot String, ByVal Level As String) As String
```

'Execution:

```
Dim BarCode As String

STXSetPlateDetector(STXId, "1")

STXBCRPath (STXId, "3")

BarCode = STXReadBarcode(STXId, TextBoxSlotPos.Text,
TextBoxLevelPos.Text)
```

4.8.2.2 Visual C# .NET

//Declaration:

```
[DllImport("STX_lib")]
private extern static String STXReadBarcode(String pNbr, String Slot,
 String Level);
```

//Execution:

```
String  BarCode = STXReadBarcode(STXId, TextBoxSlotPos.Text,
TextBoxLevelPos.Text);
```

4.8.2.3 Borland Delphi 6

//Declaration:

```
function STXReadBarcode(pNbr, Slot, Level : PChar) : PChar; stdcall;
external 'STX_lib.dll';
```

//Execution:

```
var
  BCRResult : String;
begin

  BCRResult := STXReadBarcode(pNbr, PChar(EditSlot.Text),
PChar(EditLevel.Text));

end;
```

4.8.2.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * ReadBarcode)(char* PNbr, char* Slot, char*
Level);
ReadBarcode STXReadBarcode;


//Initialization (constructor) *.cpp file

STXReadBarcode = (ReadBarcode)GetProcAddress(dllInstance,
"STXReadBarcode");
```

//Execution:

```
String BC = String(STXReadBarcode(pNbr, "1", "1"));
```

4.8.3 STXInventorySlot (PChar PortNumber, Slot, MinLevel, MaxLevel, FileName, PPD, BCR)

Implements inventory of a defined Slot from MinLevel position to MaxLevel position, result saves to the file – InvFileName. If name of file is not assigned, it will be generated automatically. The name of file consists of a Serial number of device, date (e.g. 3298_A6010101.inv, where last two digits is a number of the file).

Parameters:

```
PortNumber - Number of a Serial Port.

InvFileName - name of file for saving results of inventory.

PPD – {0,1} sets whether Plate Present Detector will be
used for Inventory.
1 - Inventory with Plate Presents Detector.
2 - Inventory without Plate Presents Detector.

BCR - {0,1} sets whether Barcode Reader will be used for
Inventory.
1 - Inventory with Barcode Reader.
2 - Inventory without Barcode Reader.
```

Return values (PChar):

```
"1" - STXInventorySlot operation is completed.

"0" - Device is not initialised.

"2" - Previous long operation is not finished.

"3" - Error save result to file.
```

A result of the STXInventoryCassets is a file which consists of five columns separated by coma. A first column is a number of Cassette, a second column is a value of Level, a fourth - value of Plate Present Detector (1 - plate is present; 0 - plate is not present or Plate Present Detector is off), a fifth column is value of a Barcode (<null> - No Barcode or Barcode Reader is switched off).

For example:

```
...
1,10,0,1,<null>
1,11,0,0,<null>
1,12,0,1,0484856654
...
```

Examples:

4.8.3.1 *Visual Basic .NET (Visual Basic 6.0)*

'Declaration:

```
Private Declare Sub STXInventorySlot Lib "STX_lib" (ByVal pNbr As
String, ByVal invSlot As String, ByVal invMinLevel As String, ByVal
invMaxLevel As String, ByVal F_NAme As String, ByVal PPD As String,
ByVal PPD As String)
```

'Execution:

```
STXInventorySlot(STXId, "1", "1", "22", "Inventory.txt", "0", "1")
```

4.8.3.2 *Visual C# .NET*

//Declaration:

```
using System.Runtime.InteropServices;

[DllImport("STX_lib")]
private extern static String STXInventorySlot(String pNbr, String
Slot, String MinLevel, String MaxLevel, String FileName, String PPD,
String BCR);
```

//Execution:

```
STXInventorySlot(STXId, "1", "1", "10", "", "0", "1");
```

4.8.3.3 *Borland Delphi 6*

//Declaration:

```
function STXInventorySlot(pNbr, Slot, MinLevel, MaxLevel, Fname, PPD,
BCR : PChar) : PChar; stdcall; external 'STX_lib.dll';
```

//Execution:

```
var
  BCR : String;
  PPD : String;
begin

  BCR := '0'; // Inventory without BCR
  PPD := '1'; // Inventory with Plate Presents Detector

  // Iventory of Slot: 1, Levels: from 1 to 22
  STXInventorySlot(pNbr, '1', '1', '22', 'Inventory_file.txt',
PChar(PPD), PChar(BCR));

end;
```

4.8.3.4 Borland C++Builder

//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * InventorySlot)(char* PNbr, char* Slot,
char* MinLevel, char* MaxLevel, char* FileName, char* PPD, char*
BCR);
InventorySlot STXInventorySlot;


//Initialization (constructor) *.cpp file

STXInventorySlot = (InventorySlot)GetProcAddress(dllInstance,
"STXInventorySlot");
```

//Execution:

```
STXInventorySlot(pNbr, "1", "1", "22", "", "1", "1");
```

4.8.4 STXInventoryCassettes(PChar PortNumber, FileName, PPD, BCR)

Implements inventory of entire unit, result saves to the file – InvFileName. If name of file is not assigned then it will be generated automatically. The name of file consists of a Serial number of device, date (e.g. "3298_A6010101.inv", where last two digits is a number of the file).

Parameters:

Parameter:

```
PortNumber - Number of a Serial Port.

FileName - name of file for saving results of inventory.

PPD – {0,1} sets whether Plate Present Detector  will be
used for Inventory.
1 - Inventory with Plate Presents Detector.
2 - Inventory without Plate Presents Detector.

BCR – {0,1} sets whether Barcode Reader will be used for
Inventory.
1 - Inventory with Barcode Reader.
2 - Inventory without Barcode Reader.
```

Return values (PChar):

```
"1" - STXInventoryCassets operation is started.

"0" - Device is not initialised.

"2" - Previous long operation is not finished.
```

A result of the STXInventoryCassets is a file which consists of five columns separated by coma. A first column is a number of Cassette, a second column is a value of Level, a fourth - value of Plate Present Detector (1 - plate is present; 0 - plate is not present or Plate Present Detector is off), a fifth column is value of a Barcode (<null> - No Barcode or Barcode Reader is switched off).

For example:

```
...
1,1,0,1,<null>
1,2,0,0,<null>
1,3,0,1,0484856654
...
```

Examples:

4.8.4.1 Visual Basic .NET (Visual Basic 6.0)

'Declaration:

```
Private Declare Sub STXInventoryCassets Lib "STX_lib" (ByVal pNbr As
String, ByVal F_Name As String, ByVal PPD As String, ByVal PPD As
String)
```

'Execution:

```
Dim PPD As String
Dim BCR As String

If CheckBox1.Checked Then
  PPD = "1"

Else

  PPD = "0"

End If


If CheckBox2.Checked Then
  BCR = "1"

Else

  BCR = "0"

End If

STXInventoryCassets(STXId, "", PPD, BCR)
```

4.8.4.2 Visual C# .NET

//Declaration:

```
using System.Runtime.InteropServices;

[DllImport("STX_lib")]
private extern static String STXInventoryCassets(String pNbr, String
FileName, String PPD, String BCR);
```

//Execution:

```
STXInventoryCassets(STXId, "", "0", "1");
```

4.8.4.3 Borland Delphi 6

//Declaration:

```
function STXInventoryCassets(pNbr, Fname, PPD, BCR : PChar) : PChar;
stdcall; external 'STX_lib.dll';
```

//Execution:

```
var
  BCR : String;
  PPD : String;
begin

BCR := '1'; // Inventory with BCR
PPD := '1'; // Inventory with Plate Presents Detector

STXInventoryCassets(pNbr, 'Inventory_file.txt', PChar(PPD),
PChar(BCR));

end;
```

4.8.4.4 Borland C++Builder

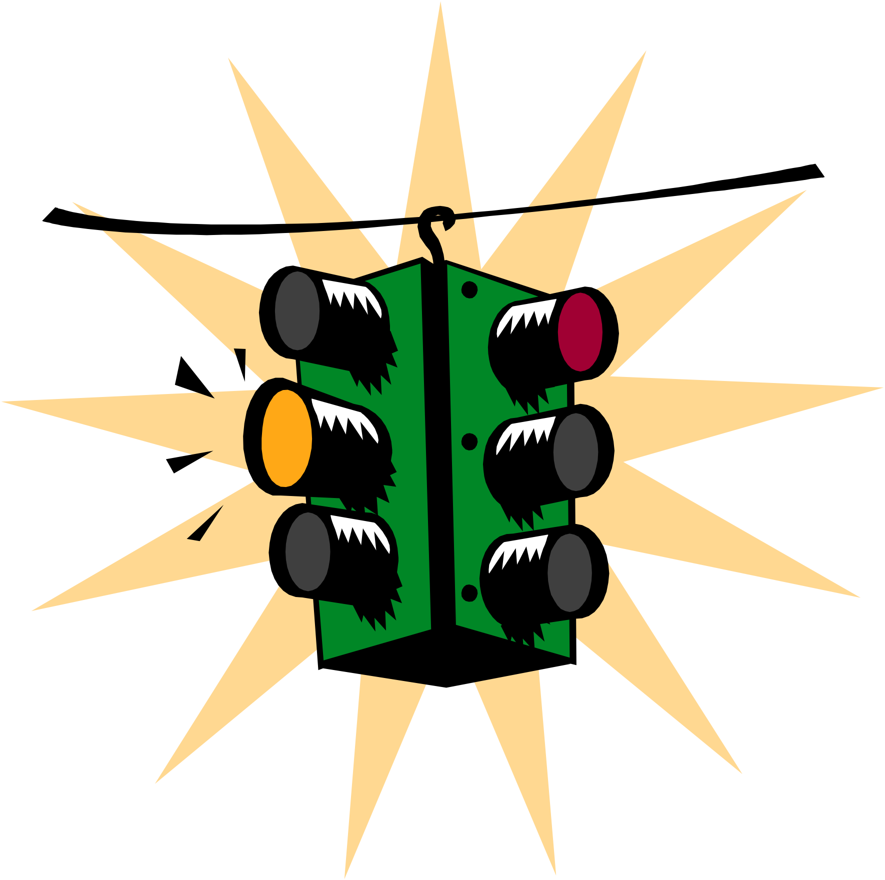
//Declaration:

```
//Prototype *.h file


typedef char* (__stdcall * InventoryCassets)(char* PNbr, char*
FileName, char* PPD, char* BCR);
InventoryCassets STXInventoryCassets;


//Initialization (constructor) *.cpp file

STXInventoryCassets = (InventoryCassets)GetProcAddress(dllInstance,
"STXInventoryCassets");
```

//Execution:

```
STXInventoryCassets(pNbr, "", "1", "0");
```

5 Special Applications Hints



5 Special Applications Hints

5.1 Borland Familiy Laguages (Delphi, C++ Builder)

The STX_lib.dll completely compatible with Borland programming languages: C++ Builber and Delphi.

Examples of using this languages you can find in this document, see above. For safe and reliable using the STX_lib library we recommend all version of this languages.

5.2 Skripted Type Language Applications

For using STX_lib by Interpreted languages such as VBScript, JScript and so on, client software have to use 1.0.0.2 version of DLL and special ActiveX-bridge for it. The ActiveX-bridge has the same names of functions, the same numbers and types of parameters and returning values as STX_lib.dll.

5.3 .NET Based Applications

5.3.1 Microsoft Visual Basic

The STX_lib.dll was tested with Microsoft Visual Basic .NET (Microsoft .NET Framework 1.1) ,(as well Microsoft Visual Basic 6.

Examples of using this languages you can find in this document, see above.

5.3.2 Microsoft Visual C#

The STX_lib.dll was tested with Microsoft Visual C# (Microsoft .NET Framework 1.1).

Examples of using this languages you can find in this document, see above.

5.4 Java Based Applications

This version of STX_lib.dll was not tested with JVM from Sun. Examples or special edition of STX_lib.dll for JVM we will release in future.