bsa4ft.dll

Contents

Files

1. bsa4ft.dll – DLL file.
2. bsa4ft.lib – LIB file (for program developers only).
3. BsaFunctionsT.h – C++ header file (for program developers only).
4. bsa4ft.dll.pdf – current info file.

Files can be downloaded at http://www.anbinderis.lt/tomas/bsa4ft_dll/

How to use bsa4ft.dll

1. Add bsa4ft.lib and BsaFunctionsT.h to your project.
2. Copy bsa4ft.dll to your working directory.
3. Include BsaFunctionsT.h in your code:

```
#include "BsaFunctionsT.h"
```

4. Set COM port:

```
DataCommPortT comm_port;
comm_port.nCom = 0;
comm_port.dwBaudRate = 9600;
```

5. Create *CBsaFunctionsT* object:

```
CBsaFunctionsT BsaFunctionsT;
```

6. Call DLL functions (*read_ECT* example):

```
CStringArray result_str_array;
CByteArray result_byte_array;

if (BsaFunctionsT.read_ECT(&comm_port, &result_str_array, &result_byte_array) == TRUE)
{
   CFile file_bin;
   if (file_bin.Open("data_bin.txt", CFile::modeCreate|CFile::modeWrite))
   {
      file_bin.Write(result_byte_array.GetData(), result_byte_array.GetSize());
      file_bin.Close();
   }
   else { ... }

   CStdioFile file;
   if (file.Open("data_text.txt", CFile::modeCreate|CFile::modeWrite))
   {
      CString buffer;
      buffer.Format("In binary buffer lines=%d\n", result_byte_array.GetSize());
      file.WriteString(buffer);
      buffer.Format("In text buffer lines=%d\n", result_str_array.GetSize());
      file.WriteString(buffer);
      for (int z = 0; z < result_str_array.GetSize(); z++)
            file.WriteString(result_str_array.GetAt(z));
      file.Close();
   }
   else { ... }
}
else printf("read_ECT() error: [%s]\n", get_error_text(comm_port.nError));
```

Error codes

Error codes and messages (C++ code):

```
char* get_error_text(BYTE error_num)
{
   switch (error_num)
   {
      // errors as in „bulgarian“ bsa4f.dll:
      case 0:   return "No errors!";
      case 1:   return "No connection between the PC and BSA-4F!";
      case 2:   return "No response!";
      case 3:   return "Error in accepting!";
      case 4:   return "Error in delivering!";
      case 5:   return "Error in COM port!";
      case 6:   return "Invalid reply!";
      case 7:   return "Wrong control sum!";
      case 10:  return "Wrong nCom value in DataCommPortT structure!";
      case 11:  return "Wrong dwBaudRate value in DataCommPortT structure!";
      // new errors:
      case 101: return "Wrong data format!";
      case 102: return "Internal DLL error!";
      case 103: return "Wrong ECT meaningful bytes control sum!";

      default:  return "Unexpected error!";
   }

   return "Unexpected error!";
}
```

Structure *DataCommPortT*

COM port settings structure.

```
struct DataCommPortT {
   DWORD dwBaudRate;
   BYTE nCom;
   BYTE nError;
};
```

- *dwBaudRate* – 110, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 56000, 57600, 15200, 128000, 256000;
- *nCom* – COM port number (0 – COM1, 1 – COM2);
- *nError* – returned error code (see „Error codes“ section);

Class *CBsaFunctionsT*

function *read_ECT*

Function reads Electronic Control Tape (ECT). ECT size is taken from ECT header.
On success returns TRUE. Else get error from *nError* in *DataCommPortT* structure.

```
BOOL read_ECT(DataCommPortT *pDataPort, CStringArray* pStringArr, CByteArray *pBufferEKL);
```

- *pDataPort* – pointer to COM port structure;
- *pStringArr* – pointer to return buffer for text data;
- *pBufferEKL* – pointer to return buffer for binary data;

2010.09.05