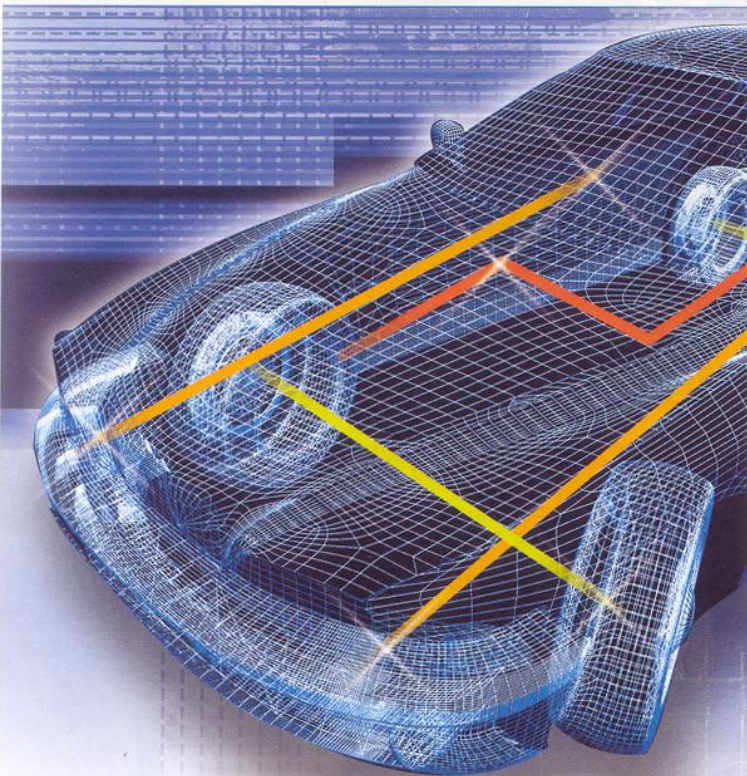


Die Netzwerke werden dabei, unter anderem, in die Teilbereiche Cluster, ECU, Channel, Gateway, Controller, Frames, Signale und Coding unterteilt (**Bild 1**).

Der FIBEX-Standard definiert die einzelnen Netzwerke und Netzwerkelemente sowie die zwischen ihnen bestehenden möglichen Referenzen. Ein Cluster im Sinne von FIBEX definiert dabei ein Netzwerk mit spezifischen Netzwerkparametern, eine ECU stellt ein elektronisches Steuergerät mit definierter Funktionalität (Functions) sowie Send- und Empfangsbotschaften (Frames) dar, die ihrerseits die Signale enthalten, die zwischen den einzelnen Steuergeräten ausgetauscht werden sollen. Wie in Bild 1 zu erkennen ist, gibt es dabei verschiedene Wege, um zum gleichen Ziel zu



kommen. So können Signale z. B. entweder über die Input/Output-Ports oder über die Signal-Instanz bzw. Signalgruppe mit dem darüber liegendem Frame referenziert werden. Für den Anwender ist es dabei nicht immer leicht den Überblick zu behalten, geschweige denn, den richtigen Weg zu finden, auf dem er die für ihn relevanten Informationen erhalten kann.

Funktionen der FIBEX Import-DLL

Die FIBEX Import-DLL ist eine 32-bit Dynamic Link Library (DLL) für Windows, die die Integration von FIBEX XML Dateien in eigene Programme wesentlich erleichtert. Mit Hilfe der FIBEX Import-DLL können FIBEX XML-Dateien gelesen und sämtliche in FIBEX definierten Bereiche (Cluster, ECUs, Frames etc.) für FlexRay, CAN, MOST und LIN extrahiert werden. Für jeden Bereich ist dabei eine eige-

ne Funktion mit definierter Parameter-Struktur vorhanden. In FIBEX definierte FlexRay Cluster-Parameter können mit Hilfe der Funktion `fxGetCHI` extrahiert und in eine Registerdefinition z.B. für einen FlexRay Communication Controller (Bosch ERay, MFR4300) umgesetzt werden.

Sind nicht die einzelnen Teilbereiche von Interesse, sondern eine komplett hierarchisch aufgebaute Baumstruktur, bestehend aus Steuergeräten, Send-, Empfangsbotschaften, Signalen und Signalelementen (Codings), steht hierfür die Funktion `fxGetTree` zur Verfügung. Mit Hilfe dieser Funktion lässt sich eine vereinfachte FIBEX-Struktur aufbauen, die für die allermeisten Fälle ausreicht.

Mit der Funktion `fxValidate` kann eine FIBEX-Datei „Offline“ validiert werden. Die dafür nötigen Schema-Dateien müssen dabei in einem definierten lokalen Arbeitsverzeichnis liegen. Sprachabhängige Einträge, z. B. alle mit „de“ (deutsch) gekennzeichneten Informationen eines bestimmten Elementes, lassen sich mit der Funktion `fxGetDescription` einfach extrahieren (**Bild 2**).

```
fxInit();
fxLoadFibexFile();
fxGetProject();
fxGetClusters();
fxGetChannels();
fxGetGateways();
fxGetECUs();
fxGetFrames();
fxGetFunctions();
fxGetSignals();
fxGetComposites();
fxGetPhysicalDimensions();
fxGetUnits();
fxGetCodings();
fxGetPortReqs();
fxGetFunctionRegs();
fxGetSignalGroups();
fxFreeMemory();
fxGetDllInfos();
fxGetErrorText();
fxValidate();
fxGetDescription();
fxGetCHI();
fxGetTree();
```

Bild 2: Funktionsaufrufe der FIBEX Import-DLL.

Einsatz der FIBEX Import-DLL

Im Folgenden wird gezeigt, wie mit vier Funktionsaufrufen eine komplette FIBEX-Datei eingelesen und interpretiert werden kann:

Schritt 1: FIBEX Import-DLL initialisieren.

Schritt 2: FIBEX-Datei einlesen

```
// Init Dll with License-Key
IRet = fxInit(„xyyz“);
if (IRet != 0)
{
    MessageBox(L„Invalid License“);}
```

Schritt 4: Speicher freigeben

```
// Load Fibex-File
IRet = fxLoadFibexFile(cFile, READ_ALL);
if (IRet != 0)
{
    szTmp.Format(L„Error: %d“,IRet);
    MessageBox(szTmp);
}
```

```

// Get TreeRoot
fyTreeRoot *pTreeRoot = NULL;
IRet = fxGetTree(&pTreeRoot);
if (IRet != 0)
{
    szTmp.Format(L"Error: %d",IRet);
    MessageBox(szTmp);
}
//ECUs
dwNumEcus = pTreeRoot->dwNumEcus;
pEcu = pTreeRoot->pFirstEcu;
while (dwNumEcus—)
{
    //ECU Name
    szECUName = (LPWSTR)pEcu->szShortName;

    //Channels
    dwNumChannels = pEcu->dwNumChannels;
    pChannel = pEcu->pFirstChannel;
    while (dwNumChannels—)
    {
        // Channel Name
        szChannelName = (LPWSTR)pChannel->szShortName;

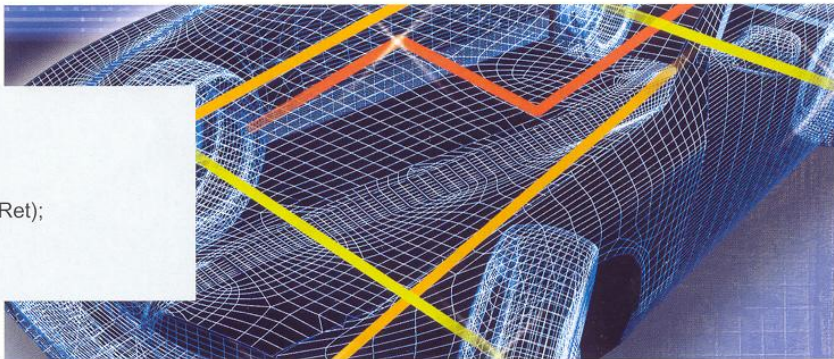
        // Tx Frames
        dwNumTxFrames = pChannel->dwNumTxFrames;
        pTxFrame = pChannel->pFirstTxFrame;
        while (dwNumTxFrames—)
        {
            // Frame Name
            szFrameName = (LPWSTR)pTxFrame->szShortName;

            // Tx Signals
            dwNumSignals = pTxFrame->dwNumSignals;
            pSignal = pTxFrame->pFirstSignal;
            while (dwNumSignals—)
            {
                // Signal Parameters
                szSignalName = (LPWSTR)pSignal->szShortName;
                szSignalUnit = (LPWSTR)pSignal->szUnit;
                dwStartBit = pSignal->dwStartBit;
                dwBitSize = pSignal->dwBitSize;
                dOffset = pSignal->dOffset;
                dFactor = pSignal->dFactor;
                // .....
                // .....
                // .....

                //Next Signal
                pSignal++;
            }
            //Next Tx Frame
            pTxFrame++;
        }
        //Next Channel
        pChannel++;
    }
    //Next ECU
    pEcu++;
}

```

```
// Free memory
IRet = fxFreeMemory();
if (IRet != 0)
{
    szTmp.Format(L"Error: %d",IRet);
    MessageBox(szTmp);
}
```



FIBEX Spezialitäten

Die FIBEX Import-DLL berücksichtigt auch viele der in FIBEX enthaltenen spezifischen Details. So werden beispielsweise nicht nur Frames, sondern auch Sub-Frames mit dazugehörigem Multiplexer eingelesen und berücksichtigt. Auch Mehrfach-Übertragungen von FlexRay Frames mit identischem Inhalt, definiert in den „FIBEX absolute Timings“, sowie herstellerspezifische FIBEX-Implementierungen berücksichtigt die FIBEX Import-DLL automatisch. So werden z.B. auch nichtreferenzierte Signalgruppen anhand eines intelligenten Suchalgorithmus den richtigen Signalen zugewiesen.

Dateien gelesen und deren Inhalt dargestellt bzw. interpretiert werden können. Mitgelieferte Beispiele in C, C++, C#, VB und VB.Net erleichtern die Integration in eigene Programme.

Dipl.-Ing. Thomas Criegee ist Geschäftsführer der CRST GmbH in 82131 Gauting.

Zusammenfassung

Der vorliegende kurze Artikel hat aufgezeigt, wie mit Hilfe der FIBEX Import-DLL auch ohne detaillierte XML-Kenntnisse und spezifische FIBEX Expertise komplexe FIBEX-