Module documentation

(MOD)

(TINF18C, SWE I Praxisprojekt 2019/2020)

Modul

Library

Project: *DD2AML Converter*

Customer: *Rentschler & Ewertz*
Rotebühlplatz 41
70178 Stuttgart

Supplier: by Nora Baitinger and Antonia Wermerskirch - Team 3
(Nora Baitinger, Antonia Wermerskirch, Lara Mack, Bastiane Storz)
Rotebühlplatz 41
70178 Stuttgart

Version	Date	Author	Comment
0.1	07.09.2018		created
0.1	13.04.2020	Nora Baitinger	Filled with information
1.0	04.05.2020	Nora Baitinger	Added analysis and design
1.0	10.05.2020	Nora Baitinger	Added Appendix and module tests
1.0	13.05.2020	Antonia Wermerskirch	Added conversion rules

1. Content

2. Scope

The Module Documentation (MOD) describes the architecture, the interfaces and the main features of the module. It also describes the module/component test including the results. It can also serve as a programming or integration manual for the module. If there are some risks related to the module itself, they shall be noted and commented within this document.

3. Definitions

AML	Automation Markup Language
AML DD	AML Device Description
AMLX	AML Package
CSP+	Control and Communication System Profile
IODD	Input/Output Device Description
PN	Profinet
SAS	System Architecture Specification
SRS	System Requirements Specification

4. Figures

5. Module Requirements

5.1. User View

This Module is the most important module because it contains the logic for converting the input file to an AML file. It also includes the AML Packer that generates an AMLX with the AML file and all its dependencies. This module contains the submodules “Converter” and “AML Packager” both described in an additional module documentation.

So from the users perspective, the module is supposed to:

1. Contain the important functionalities to convert a DD file and package it to the AMLX
2. Provide the logging interface for the conversion
3. Contain the necessary data about the DD-file formats and the AML formats in both CAEX Versions
4. Contain the ruleset used for the conversion
5. Provide a stand-alone library for developers

The Library module can be used as a stand-alone library. This means that this module can be separated from the rest of the project and it will work on its own. The purpose is that developers can implement the library in their own projects.

5.2. Requirements

The Library implements the requirements LF50 and part of LF80.

The LF50 requirement states the Library shall be used as a stand-alone library for developers. So, it needs to start the conversion and return either the generated AML as string or all dependencies within the AMLX package. Therefore the needed components (“Converter” and AML Packager”) must be coordinated and the output files shall be provided to the user. But most of all the library must contain the conversion rules.

According to LF80 the system should include error handling. The Library shall give details about problems in form of helpful error messages both to the user and any other developers working with this software.

5.3. Module Context

The Library is the main engine of the whole system. So all the other modules work together with the Library module directly or indirectly.

External systems can use the APIs provided to access the library directly.
The AML file constructed by the “Converter” and its resources are passed to the module “AML packager” to build the AMLX package.
The Library also uses the module “Logger” to create and write log files.

To communicate with the user, the Library uses its submodules. For instance, the “Converter” provides the necessary information either to the “Graphical User Interface” or the “Command Line Interface”. These procedures are shown and described in additional module documentations.

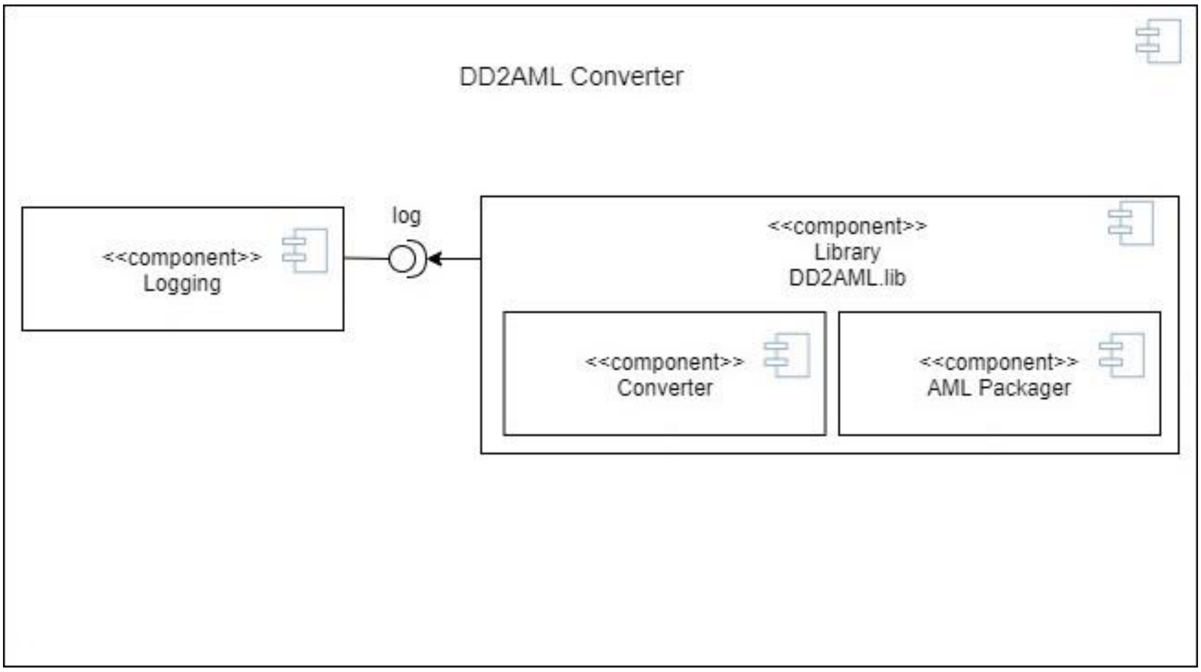

Figure 1 - Library in the module context

6. Analysis

The Library module needs to be able to convert a given DD-file into the corresponding AML file and generate the AML Package.

Therefore, the module contains two other modules, the “Converter” and the “AML Packager”, as well as other important information needed for the conversion process.

The Library module is a collection of different modules and files that are important to complete the conversion. The purpose of this module is to collect the needed program code and files to deliver the requested stand-alone library.

The essential element of the Library is the ruleset for the conversion.
Some questions remain open:

- How flexible is the ruleset?
- Is the ruleset easily extendable?
- Can all of the provided information in the given DD-file converted to AML?

7. Design

Based on the fact, that this module contains a lot of different files and program code, it is important to understand the file structure and coherences of the DD2AML.Lib folder.

Structure of the DD2AML.Lib

First of all, this part of the program can be found under TINF18C_Team_3_DD2AML-Converter/SOURCE/src/DD2Aml.Lib/. This folder contains all the necessary information to convert and log the process of converting a DD-file into the corresponding AML file.

Below the important folders and files are listed:

- **Logging/**
  This folder contains the logging service which is responsible for the log levels during the conversion process.
- **Models/**
  This folder contains the classes representing the different file formats: AML, GSD, IODD and CSP+. For each of the requested CAEX Versions of the AML file, a different AML class is needed. The folder also contains the xsd files that were the foundation for the format classes.
- **Properties/**
  The AssemblyInfo.cs is the content of this folder.
- **AML2/**
  In this folder the most important information is saved. It contains the ruleset for the conversion of the DD-files into the AML CAEX Version 2.15. So, there are three xml-files for each of the given DD-file formats (GSD, IODD and CSP+).
- **AML3/**
  Like the folder above, this folder contains the ruleset for the conversion into the AML CAEX Version 3.0. Therefore, there are also three different xml-files each representing one of the formats.
- **AMLPackager.cs**
  This is the class responsible for the creation of the AML Package. It is described explicitly in the module documentation “AML Packager”.
- **Converter.cs**
  This class file is one of the files that ensures the conversion and is described in the additional module documentation “Converter”.
- **Util.cs**
  The Util.cs class is the second part of the “Converter” described in its module documentation as well. It is a helper class created to maintain clarity and comprehensibility.

7.1. Risks

This module is the main part of the entire system, so the risk is the most important to be handled. It contains the “Converter” that is responsible for the conversion and the “AML Packager” module whose result is needed to return to the user. These modules are described separately, so their risks and how to handle them, can be read in the mentioned module documentations.

However, this module contains the ruleset for the conversion and the descriptions of how a valid IODD or CSP+ must look like.

So these rules must be appropriate and cover all the possible tags that a DD-file can include. For this it is very important to have correct and all the possible information about the given device description formats IODD and CSP+. Furthermore it is necessary to develop these rules very carefully and test their correctness most accurately.

8. Implementation

The implementation of the two modules “Converter” and “AML Packager” is described in sperate module documentations. These will contain the AMLPackager.cs, the Converter.cs and the Util.cs. All conversion rules can be found in the folder SOURCE/src/DD2Aml.Lib/AML2 for CAEX 2.15 or in the folder SOURCE/src/DD2Aml.Lib/AML3 for CAEX 3.0.

Below the ruleset for IODD and CSP+ will be explained as well as the models that represent a valid IODD and CSP+ file.

8.1. IODD

8.1.1. ProfileHeader

The IODD <ProfileHeader> contains fixed information to identity the IODD as IODD

In IODD:

```
<ProfileHeader>
  <ProfileIdentification>IO Device Profile</ProfileIdentification>
  <ProfileRevision>1.1</ProfileRevision>
  <ProfileName>Device Profile for IO Devices</ProfileName>
  <ProfileSource>IO-Link Consortium</ProfileSource>
  <ProfileClassID>Device</ProfileClassID>
  <ISO15745Reference>
    <ISO15745Part>1</ISO15745Part>
    <ISO15745Edition>1</ISO15745Edition>
    <ProfileTechnology>IODD</ProfileTechnology>
  </ISO15745Reference>
</ProfileHeader>
```

Instead of storing the IODD information there, the element is replaced by an <AdditionalInformation> element, which contains data about the converter, which was used to create the AML file. When converting to AML, the profile header is differentiated between CAEX 2.15 and CAEX 3.0.

In AML (CAEX 2.15):

```
<AdditionalInformation>
  <WriterHeader>
  <WriterName>DHBW Stuttgart TINF18C</WriterName>
  <WriterID>D331B5B6-C3FA-4D3C-8F6D-93F6F0215EF4</WriterID>
  <WriterVendor>DD2AML TINF18C</WriterVendor>
  <WriterVersion>1.0.0</WriterVersion>
  <WriterRelease>1.0.0</WriterRelease>
  <LastWritingDateTime></LastWritingDateTime>
  <WriterProjectTitle>DD2AML Converter</WriterProjectTitle>
  <WriterProjectID>DD2AML Converter</WriterProjectID>
  </WriterHeader>
</AdditionalInformation>
```

In AML (CAEX 3.0):

```
<AdditionalInformation>
```

```
  <SuperiorStandardVersion>AutomationML 2.1</SuperiorStandardVersion>
  <SourceDocumentInformation
    OriginName="DHBW Stuttgart TINF18C"
    OriginID="Test"
    OriginVendor="DD2AML TINF18C"
    OriginVendorURL="https://github.com/WAntonia/TINF18C_Team_3_DD2AML-Converter"
    OriginVersion="1.0.0"
    OriginRelease="1.0.0"
    OriginProjectTitle="DD2AML Converter"
    OriginProjectID="DD2AML Converter">
  </SourceDocumentInformation>
</AdditionalInformation>
```

8.1.2. ProfileBody

The part <ProfileBody> contains the description of identity and functionality of the device.

In IODD:

```
<ProfileBody>
      …
      …
</ProfileBody>
```

The <ProfileBody> translates in the AML Root file to <SystemUnitClassLib>:

In AML:

```
<CAEXFile.SystemUnitClassLib>
  <SystemUnitClassLib CAEXObject.Name="ComponentSystemUnitClassLib">
  </SystemUnitClassLib>
</CAEXFile.SystemUnitClassLib>
```

8.1.3. DeviceIdentity

<DeviceIdentity> contains all information about the vendor. For example, VendorName, VendorId and VendorLogo.

In IODD:

```
<DeviceIdentity vendorId = "888" vendorName ="Balluff" deviceId = "329494">
  <VendorText textId = "TI_VendorText"/>
  <VendorUrl textId = "TI_VendorUrl"/>
  <VendorLogo name = "Balluff-logo.png"/>
  <DeviceName textId = "TI_DeviceName"/>
  <DeviceFamily textId = "TI_DeviceFamily"/>
  <DeviceVariantCollection>
    <DeviceVariant productId = "BNI00CL" deviceSymbol = "Balluff-BNI_IOL_355_S02_Z013-pic.png" deviceIcon = "Balluff-BNI_IOL_355_S02_Z013-icon.png">
      <Name textId = "TI_Device1_Name"/>
      <Description textId = "TD_Device1_Desc"/>
    </DeviceVariant>
  </DeviceVariantCollection>
</DeviceIdentity>
```

In AML:

```
<InternalElement CAEXObject.Name="DeviceIdentity" CAEXObject.ID="GUID">
  <SystemUnitClassType.Attribute>
    <Attribute CAEXObject.Name="VendorId" Attribute.AttributeDataType="xs:integer">
      <Attribute.Value>888</Attribute.Value>
    </Attribute>
    <Attribute CAEXObject.Name="VendorName" Attribute.AttributeDataType="xs:string">
      <Attribute.Value>Balluff</Attribute.Value>
    </Attribute>
    <Attribute CAEXObject.Name="DeviceId" Attribute.AttributeDataType="xs:integer">
      <Attribute.Value>329494</Attribute.Value>
    </Attribute>
    <Attribute CAEXObject.Name="VendorText" Attribute.AttributeDataType="xs:string">
      <Attribute.Value>innovating automation</Attribute.Value>
    </Attribute>
    <Attribute CAEXObject.Name="VendorUrl" Attribute.AttributeDataType="xs:anyURI">
      <Attribute.Value>http://www.balluff.com/</Attribute.Value>
    </Attribute>
    <Attribute CAEXObject.Name="DeviceFamily" Attribute.AttributeDataType="xs:string">
      <Attribute.Value>BNI - Sensor/Actor hub metal</Attribute.Value>
    </Attribute>
    <Attribute CAEXObject.Name="DeviceName" Attribute.AttributeDataType="xs:string">
      <Attribute.Value>BNI IOL-355-S02-Z013</Attribute.Value>
    </Attribute>
  </SystemUnitClassType.Attribute>
</InternalElement>
```

8.1.4. VendorLogo

<VendorLogo> contains the name of the vendor logo file.

In IODD:

```
<VendorLogo name = "Balluff-logo.png"/>
```

It is translated into AML by the following substitution.

In AML:

```
<InternalElement CAEXObject.Name="ManufacturerIcon" CAEXObject.ID="GUID">
  <SystemUnitClassType.SupportedRoleClass>
```

```
    <SupportedRoleClass SupportedRoleClass.RefRoleClassPath="AutomationMLComponentStandardRCL/ManufacturerIcon" />
  </SystemUnitClassType.SupportedRoleClass>

  <SystemUnitClassType.ExternalInterface>
    <ExternalInterface CAEXObject.Name="ExternalDataConnector" CAEXObject.ID="GUID" ExternalInterface.RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector">
      <ExternalInterface.Attribute>
        <Attribute CAEXObject.Name="refURI" Attribute.AttributeDataType="xs:anyURI">
          <Attribute.Value> Balluff-logo.png </Attribute.Value>
        </Attribute>
      </ExternalInterface.Attribute>
    </ExternalInterface>

  </SystemUnitClassType.ExternalInterface>
</InternalElement>
```

8.1.5. DeviceVariantCollection

Within the <DeviceIdentity> element there is a <DeviceVariantCollection> element. The element contains device information like name, description and device symbol.

In IODD:

```
<DeviceVariantCollection>
<DeviceVariant productId = "BNI00CL" deviceSymbol = "Balluff-BNI_IOL_355_S02_Z013-pic.png" deviceIcon = "Balluff-BNI_IOL_355_S02_Z013-icon.png">
<Name textId = "TI_Device1_Name"/>
<Description textId = "TD_Device1_Desc"/>
</DeviceVariant>
</DeviceVariantCollection>
```

In AML:

```
<InternalElement CAEXObject.Name="DeviceVariantCollection" CAEXObject.ID="GUID">
<InternalElement CAEXObject.Name="BNI IOL-355-02-Z013" CAEXObject.ID="GUID">

  <SystemUnitClassType.SupportedRoleClass>
    <SupportedRoleClass SupportedRoleClass.RefRoleClassPath="DD2AML.Lib/"/>
  </SystemUnitClassType.SupportedRoleClass>

  <SystemUnitClassType.Attribute>
    <Attribute CAEXObject.Name="productId" Attribute.AttributeDataType="xs:string">
      <Attribute.Value>BNI00CL</Attribute.Value>
    </Attribute>
    <Attribute CAEXObject.Name="Description" Attribute.AttributeDataType="xs:string">
      <Attribute.Value> Sensor- / Aktorhub mit galvanischer Trennung </Attribute.Value>
    </Attribute>
  </SystemUnitClassType.Attribute>

  <SystemUnitClassType.InternalElement>
    <InternalElement CAEXObject.Name="deviceSymbol" CAEXObject.ID="GUID">
      <InternalElement.RoleRequirements InternalElement.RoleRequirements.RefBaseRoleClassPath="AutomationMLComponentStandardRCL/ComponentPicture"/>

      <SystemUnitClassType.ExternalInterface>
        <ExternalInterface CAEXObject.Name="ExternalDataReference" CAEXObject.ID="GUID" ExternalInterface.RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector">
          <ExternalInterface.Attribute>
            <Attribute CAEXObject.Name="refURI" Attribute.AttributeDataType="xs:anyURI">
              <Attribute.Value> Balluff-BNI_IOL_355_S02_Z013-pic.png </Attribute.Value>
            </Attribute>
          </ExternalInterface.Attribute>
        </ExternalInterface>
      </SystemUnitClassType.ExternalInterface>
    </InternalElement>

    <InternalElement CAEXObject.Name="deviceIcon" CAEXObject.ID="GUID">
      <InternalElement.RoleRequirements InternalElement.RoleRequirements.RefBaseRoleClassPath="AutomationMLComponentStandardRCL/ComponentIcon"/>

      <SystemUnitClassType.ExternalInterface>
        <ExternalInterface CAEXObject.Name="ExternalDataReference" CAEXObject.ID="GUID" RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector">
          <ExternalInterface.Attribute>
            <Attribute CAEXObject.Name="refURI" Attribute.AttributeDataType="xs:anyURI">
              <Attribute.Value> Balluff-BNI_IOL_355_S02_Z013-icon.png </Attribute.Value>
            </Attribute>
          </ExternalInterface.Attribute>
        </ExternalInterface>
      </SystemUnitClassType.ExternalInterface>
    </InternalElement>
  </SystemUnitClassType.InternalElement>
</InternalElement>

</InternalElement>
```

8.1.6. Features

The transformation rules for supported Features are described here:

In IODD:

```
<Features blockParameter="true" dataStorage="true">
<SupportedAccessLocks parameter="true" dataStorage="true" localUserInterface="false" localParameterization="false"  />
</Features>
```

In AML:

```
<InternalElement CAEXObject.Name="Features" CAEXObject.ID="Features">

  <SystemUnitClassType.Attribute>
    <Attribute CAEXObject.Name="blockParameter" Attribute.AttributeDataType="xs:boolean">
      <Attribute.Value>true</Attribute.Value>
    </Attribute>
    <Attribute CAEXObject.Name="dataStorage" Attribute.AttributeDataType="xs:boolean">
      <Attribute.Value>true</Attribute.Value>
    </Attribute>
  </SystemUnitClassType.Attribute>

  <SystemUnitClassType.InternalElement>
    <InternalElement CAEXObject.Name="SupportedAccessLocks" CAEXObject.ID="GUID">
      <SystemUnitClassType.Attribute>
        <Attribute CAEXObject.Name="parameter" Attribute.AttributeDataType="xs:boolean">
          <Attribute.Value>true</Attribute.Value>
        </Attribute>
        <Attribute CAEXObject.Name="dataStorage" Attribute.AttributeDataType="xs:boolean">
          <Attribute.Value> true </Attribute.Value>
        </Attribute>
        <Attribute CAEXObject.Name="localUserInterface" Attribute.AttributeDataType="xs:boolean">
          <Attribute.Value>false</Attribute.Value>
        </Attribute>
        <Attribute CAEXObject.Name="localParameterization" Attribute.AttributeDataType="xs:boolean">
          <Attribute.Value> false </Attribute.Value>
        </Attribute>
      </SystemUnitClassType.Attribute>
    </InternalElement>
  </SystemUnitClassType.InternalElement>
</InternalElement>
```

8.1.7. VariableCollection

The part <VariableCollection> contains all device parameters. It is described in more detail within the following rules.

8.1.8. StdVariableRef

The part <VariableRef> is a part of the <VariableCollection>. It contains all references to variables with their default values.

In IODD:

```
<VariableCollection>
  <StdVariableRef id = "V_DirectParameters_1"/>
  <StdVariableRef id = "V_DirectParameters_2"/>
  <StdVariableRef id = "V_SystemCommand">
    <StdSingleValueRef value = "130"/>
  </StdVariableRef>
  <StdVariableRef id ="V_DeviceAccessLocks"/>
  <StdVariableRef id = "V_VendorName" defaultValue="Balluff"/>
  <StdVariableRef id = "V_VendorText" defaultValue="innovating automation"/>
  <StdVariableRef id = "V_ProductName" defaultValue="BNI IOL-355-S02-Z013"/>
  <StdVariableRef id = "V_ProductID" defaultValue="BNI00CL"/>
  <StdVariableRef id = "V_ProductText" defaultValue="Sensor- / Aktor-hub mit galvanischer Trennung"/>
  <StdVariableRef id = "V_SerialNumber" />
  <StdVariableRef id = "V_HardwareRevision" />
  <StdVariableRef id = "V_FirmwareRevision" />
  <StdVariableRef id = "V_ApplicationSpecificTag" defaultValue = "********************************" fixedLengthRestriction = "32"/>
  …
</VariableCollection>
```

In AML:

```
<InternalElement CAEXObject.Name="StdVariableRef" CAEXObject.ID="GUID">
  <SystemUnitClassType.InternalElement>
    <InternalElement CAEXObject.Name=" V_DirectParameters_1" CAEXObject.ID="GUID">
      <InternalElement.Attribute>
        <Attribute CAEXObject.Name="defaultValue" Attribute.AttributeDataType="xs:string">
          <Attribute.Value></Attribute.Value>
        </Attribute>
      </InternalElement.Attribute>
    </InternalElement>
    <InternalElement CAEXObject.Name=" V_DirectParameters_2" CAEXObject.ID="GUID">
      <InternalElement.Attribute>
        <Attribute CAEXObject.Name="defaultValue" Attribute.AttributeDataType="xs:string">
          <Attribute.Value></Attribute.Value>
        </Attribute>
      </InternalElement.Attribute>
    </InternalElement>
     …
     …
  </SystemUnitClassType.InternalElement>
</InternalElement>
```

8.1.9. Variable

The part <Variable> also belongs to the <VariableCollection> and contains all information about the IODD parameters.

In IODD:

```
<Variable id = "V_InputInversion" index = "64" accessRights = "rw">
  <Datatype xsi:type = "RecordT" bitLength = "8" subindexAccessSupported = "true">
    <RecordItem subindex = "1" bitOffset = "0">
      <SimpleDatatype xsi:type = "BooleanT">
        <SingleValue value = "false">
          <Name textId = "TI_Not_Inverted"/>
        </SingleValue>
        <SingleValue value = "true">
          <Name textId = "TI_Inverted"/>
        </SingleValue>
      </SimpleDatatype>
      <Name textId = "TI_InversionPort0Pin4"/>
    </RecordItem>
   …
</Variable>
```

In AML:

```
<InternalElement CAEXObject.Name="Variable" CAEXObject.ID="GUID">
  <SystemUnitClassType.InternalElement>
    <InternalElement CAEXObject.Name="V_InputInversion" CAEXObject.ID="V_InputInversion">
      <SystemUnitClassType.Attribute>
        <Attribute CAEXObject.Name="index" Attribute.AttributeDataType="xs:integer">
          <Attribute.Value>64</Attribute.Value>
        </Attribute>
        <Attribute CAEXObject.Name="accessRights" Attribute.AttributeDataType="xs:string">
          <Attribute.Value>rw</Attribute.Value>
        </Attribute>
        <Attribute CAEXObject.Name="Name" Attribute.AttributeDataType="xs:string">
          <Attribute.Value> Input Inversion </Attribute.Value>
        </Attribute>
        <Attribute CAEXObject.Name="Description" Attribute.AttributeDataType="xs:string">
          <Attribute.Value> Inversion of the states of input pins </Attribute.Value>
        </Attribute>
        <Attribute CAEXObject.Name="xsi:type" Attribute.AttributeDataType="xs:string">
          <Attribute.Value>RecordT</Attribute.Value>
        </Attribute>
        <Attribute CAEXObject.Name="subindexAccessSupported" Attribute.AttributeDataType="xs:boolean">
          <Attribute.Value>true</Attribute.Value>
        </Attribute>
       </SystemUnitClassType.Attribute>
      <SystemUnitClassType.InternalElement>
        <InternalElement CAEXObject.Name="RecordItem" CAEXObject.ID="GUID">
          <SystemUnitClassType.InternalElement>
            <InternalElement CAEXObject.Name="Inversion Port 0 Pin 4" CAEXObject.ID="GUID">
              <SystemUnitClassType.Attribute>
                <Attribute CAEXObject.Name="subindex" Attribute.AttributeDataType="xs:integer">
                  <Attribute.Value>1</Attribute.Value>
                </Attribute>

                <Attribute CAEXObject.Name="bitOffset" Attribute.AttributeDataType="xs:integer">
                  <Attribute.Value>0</Attribute.Value>
                </Attribute>
              </SystemUnitClassType.Attribute>
              <SystemUnitClassType.InternalElement>
                <InternalElement CAEXObject.Name="SimpleDatatype" CAEXObject.ID="GUID">
                  <SystemUnitClassType.Attribute>
                   <Attribute CAEXObject.Name="Not Inverted" Attribute.AttributeDataType="xs:boolean">
                    <Attribute.Value>false</Attribute.Value>
                   </Attribute>
                   <Attribute CAEXObject.Name="$1" Attribute.AttributeDataType="xs:boolean">
                    <Attribute.Value>true</Attribute.Value>
                   </Attribute>
                  </SystemUnitClassType.Attribute>
                </InternalElement>
              </SystemUnitClassType.InternalElement>
             </InternalElement>
            </SystemUnitClassType.InternalElement>
        </InternalElement>
      </SystemUnitClassType.InternalElement>
```

```
    </InternalElement>
  </SystemUnitClassType.InternalElement>
</InternalElement>
```

8.1.10. ProcessDataCollection

The IODD part <ProcessDataCollection> contains all process data of the device.

In IODD:

```
<ProcessDataCollection>
  <ProcessData id = "V_PD">
    <ProcessDataIn id = "V_PDI" bitLength = "24">
      <Datatype xsi:type="UIntegerT" bitLength="24"/>
      <Name textId = "TI_PDI"/>
    </ProcessDataIn>
    <ProcessDataOut id = "V_PDO" bitLength = "8">
      <Datatype xsi:type="UIntegerT" bitLength="8"/>
      <Name textId = "TI_PDO"/>
    </ProcessDataOut>
  </ProcessData>
</ProcessDataCollection>
```

In AML:

```
<InternalElement CAEXObject.Name="ProcessDataCollection" CAEXObject.ID="GUID">
  <SystemUnitClassType.InternalElement>
    <InternalElement CAEXObject.Name="V_PDI" CAEXObject.ID="V_PDI">
      <SystemUnitClassType.Attribute>
        <Attribute CAEXObject.Name="bitLength" Attribute.AttributeDataType="xs:integer">
          <Attribute.Value>24</Attribute.Value>
        </Attribute>
      </SystemUnitClassType.Attribute>
        <SystemUnitClassType.InternalElement>
          <InternalElement CAEXObject.Name="Process Data Inputs" CAEXObject.ID="GUID">
            <SystemUnitClassType.Attribute>
              <Attribute CAEXObject.Name="xsi:type" Attribute.AttributeDataType="xs:string">
                <Attribute.Value>UIntergerT</Attribute.Value>
              </Attribute>
              <Attribute CAEXObject.Name="bitLength" Attribute.AttributeDataType="xs:integer">
                <Attribute.Value>24</Attribute.Value>
              </Attribute>
            </SystemUnitClassType.Attribute>
          </InternalElement>
        </SystemUnitClassType.InternalElement>
    </InternalElement>
    <InternalElement CAEXObject.Name="V_PDO" CAEXObject.ID="V_PDO">
```

```
      <SystemUnitClassType.Attribute>
        <Attribute CAEXObject.Name="bitLength" Attribute.AttributeDataType="xs:integer">
          <Attribute.Value>8</Attribute.Value>
        </Attribute>
      </SystemUnitClassType.Attribute>
        <SystemUnitClassType.InternalElement>
          <InternalElement CAEXObject.Name="Process Data Outputs" CAEXObject.ID="GUID">
            <SystemUnitClassType.Attribute>
              <Attribute CAEXObject.Name="xsi:type" Attribute.AttributeDataType="xs:string">
                <Attribute.Value> UIntegerT </Attribute.Value>
              </Attribute>
              <Attribute CAEXObject.Name="bitLength" Attribute.AttributeDataType="xs:integer">
                <Attribute.Value>8</Attribute.Value>
              </Attribute>
            </SystemUnitClassType.Attribute>
          </InternalElement>
        </SystemUnitClassType.InternalElement>
  </InternalElement>
</SystemUnitClassType.InternalElement>
```

8.1.11. ErrorTypeCollection / EventCollection

Because the two parts <ErrorTypeCollection> and <EventCollection> are very similar in structure, only one of them is discussed in detail.

In IODD:

```
<EventCollection>
  <StdEventRef code = "20753"/>
  <StdEventRef code = "20754"/>
  <StdEventRef code = "30480"/>
</EventCollection>
```

In AML:

```
<InternalElement CAEXObject.Name="EventCollection" CAEXObject.ID="GUID">
  <SystemUnitClassType.Attribute>
    <Attribute CAEXObject.Name="20753" Attribute.AttributeDataType="xs:integer">
      <Attribute.Value></Attribute.Value>
    </Attribute>
    <Attribute CAEXObject.Name="20754" Attribute.AttributeDataType="xs:integer">
      <Attribute.Value></Attribute.Value>
    </Attribute>
    <Attribute CAEXObject.Name="30480" Attribute.AttributeDataType="xs:integer">
      <Attribute.Value></Attribute.Value>
    </Attribute>
  </SystemUnitClassType.Attribute>
</InternalElement>
```

8.1.12. MenuCollection

The <MenuCollection>, which is located in the part <UserInterface>, is similar to the structure of the <EventCollection>, which is why it is not listed further. The only difference is that the <MenuCollection> has an attribute for "accessRightRestriction".

8.1.13. CommNetworkProfile

This part describes the NetworkProfile of the IODD device.

In IODD:

```
<CommNetworkProfile xsi:type = "IOLinkCommNetworkProfileT" iolinkRevision = "V1.1">
  <TransportLayers>
    <PhysicalLayer bitrate = "COM3" minCycleTime = "400" sioSupported = "false" mSequenceCapability = "11">
      <Connection xsi:type = "M12-4ConnectionT">
        <ProductRef productId = "BNI00CL"/>
        <Wire1 function = "L+"/>
        <Wire2 function = "Other"/>
        <Wire3 function = "L-"/>
        <Wire4 function = "C/Q"/>
      </Connection>
    </PhysicalLayer>
  </TransportLayers>
</CommNetworkProfile>
```

In AML:

```
<InternalElement CAEXObject.Name="IO-Link Device Port" CAEXObject.ID="GUID">
  <SystemUnitClassType.Attribute>
    <Attribute CAEXObject.Name="minCycleTime" Attribute.AttributeDataType="xs:integer">
      <Attribute.Value>400</Attribute.Value>
    </Attribute>
    <Attribute CAEXObject.Name="sioSupported" Attribute.AttributeDataType="xs:boolean">
      <Attribute.Value>false</Attribute.Value>
    </Attribute>
    <Attribute CAEXObject.Name="bitrate" Attribute.AttributeDataType="xs:string">
```

```
      <Attribute.Value>COM3</Attribute.Value>
    </Attribute>
    <Attribute CAEXObject.Name="mSequenceCapability" Attribute.AttributeDataType="xs:integer">
      <Attribute.Value>11</Attribute.Value>
    </Attribute>
  </SystemUnitClassType.Attribute>
  <SystemUnitClassType.ExternalInterface>
    <ExternalInterface CAEXObject.Name="IO-Link Device Port" CAEXObject.ID="GUID" RefBaseClassPath="physicalEndPoint/IOLD">
      <ExternalInterface.Attribute>
        <Attribute CAEXObject.Name="Socket" Attribute.AttributeDataType="xsi:type">
          <Attribute.Value> BNI00CL </Attribute.Value>
        </Attribute>
      </ExternalInterface.Attribute>
    </ExternalInterface>
  </SystemUnitClassType.ExternalInterface>
      <SystemUnitClassType.InternalElement>
        <InternalElement CAEXObject.Name="Wire1" CAEXObject.ID="GUID">
          <SystemUnitClassType.ExternalInterface>
            <ExternalInterface CAEXObject.Name="Pin1" CAEXObject.ID="GUID" RefBaseClassPath="physicalEndPoint/wire">
              <ExternalInterface.Attribute>
                <Attribute CAEXObject.Name="function" Attribute.AttributeDataType="xsi:string">
                  <Attribute.Value>L+</Attribute.Value>
                </Attribute>
                <Attribute CAEXObject.Name="color" Attribute.AttributeDataType="xsi:string">
                  <Attribute.Value></Attribute.Value>
                </Attribute>
              </ExternalInterface.Attribute>
            </ExternalInterface>
          </SystemUnitClassType.ExternalInterface>
        </InternalElement>

        <InternalElement CAEXObject.Name="Wire2" CAEXObject.ID="GUID">
          <SystemUnitClassType.ExternalInterface>
            <ExternalInterface CAEXObject.Name="Pin2" CAEXObject.ID="GUID" RefBaseClassPath="physicalEndPoint/wire">
              <ExternalInterface.Attribute>
                <Attribute CAEXObject.Name="function" Attribute.AttributeDataType="xsi:string">
                  <Attribute.Value>Other</Attribute.Value>
                </Attribute>
                <Attribute CAEXObject.Name="color" Attribute.AttributeDataType="xsi:string">
                  <Attribute.Value></Attribute.Value>
                </Attribute>
              </ExternalInterface.Attribute>
            </ExternalInterface>
          </SystemUnitClassType.ExternalInterface>
        </InternalElement>

        <InternalElement CAEXObject.Name="Wire3" CAEXObject.ID="GUID">
          <SystemUnitClassType.ExternalInterface>
            <ExternalInterface CAEXObject.Name="Pin3" CAEXObject.ID="GUID" RefBaseClassPath="physicalEndPoint/wire">
              <ExternalInterface.Attribute>
                <Attribute CAEXObject.Name="function" Attribute.AttributeDataType="xsi:string">
                  <Attribute.Value>L-</Attribute.Value>
                </Attribute>
                <Attribute CAEXObject.Name="color" Attribute.AttributeDataType="xsi:string">
                  <Attribute.Value></Attribute.Value>
                </Attribute>
              </ExternalInterface.Attribute>
            </ExternalInterface>
          </SystemUnitClassType.ExternalInterface>
        </InternalElement>

        <InternalElement CAEXObject.Name="Wire4" CAEXObject.ID="GUID">
          <SystemUnitClassType.ExternalInterface>
            <ExternalInterface CAEXObject.Name="Pin4" CAEXObject.ID="GUID" RefBaseClassPath="physicalEndPoint/wire">
              <ExternalInterface.Attribute>
                <Attribute CAEXObject.Name="function" Attribute.AttributeDataType="xsi:string">
                  <Attribute.Value>C/Q</Attribute.Value>
                </Attribute>
                <Attribute CAEXObject.Name="color" Attribute.AttributeDataType="xsi:string">
                  <Attribute.Value></Attribute.Value>
                </Attribute>
              </ExternalInterface.Attribute>
            </ExternalInterface>
          </SystemUnitClassType.ExternalInterface>
        </InternalElement>
      </SystemUnitClassType.InternalElement>
</InternalElement>
```

8.1.14. ExternalTextCollection

The part <ExternalTextCollection> contains all text references and their actual value. Its structure is similar to the already mentioned collections, so it is not listed again.

8.2. CSP+

Since all CSP+ information in the device description file are very similar in structure, only the names and the number of attributes vary per element. Therefore, the translation rule is only presented as an example for one element. A <DeviceInfoMember> is used as an example.

In CSP+

```xml
<p:device label="DeviceSection">
    <p:comment>
      <p:item>Device Section</p:item>
    </p:comment>
    <p:deviceInfo label="DeviceInformation">
      <p:comment>
        <p:item>Device Information</p:item>
      </p:comment>
      <p:deviceInfoMember label="VendorName">
        <p:label2>
          <p:item>VendorName</p:item>
        </p:label2>
        <p:category>
          <p:item>COMMON</p:item>
        </p:category>
        <p:name>
          <p:item>Vendor name</p:item>
        </p:name>
        <p:datatype>
          <p:item>STRING_U(64)</p:item>
        </p:datatype>
        <p:data>
          <p:item>BALLUFF</p:item>
        </p:data>
      </p:deviceInfoMember>
        …
        …
    </p:deviceInfo>
  </p:device>
```

In AML

```xml
<InternalElement CAEXObject.Name="Device Information" CAEXObject.ID="GUID">
  <SystemUnitClassType.InternalElement>
    <InternalElement CAEXObject.Name= "Vendor name" CAEXObject.ID="GUID">
      <SystemUnitClassType.Attribute>
        <Attribute CAEXObject.Name="Datatype" Attribute.AttributeDataType="xs:string">
            <Attribute.Value>STRING_U(64)</Attribute.Value>
          </Attribute>
          <Attribute CAEXObject.Name="Value" Attribute.AttributeDataType="xs:string">
            <Attribute.Value> BALLUFF </Attribute.Value>
          </Attribute>
        </SystemUnitClassType.Attribute>
    </InternalElement>
  </SystemUnitClassType.InternalElement>
</InternalElement>
```

9. Module Test

This Library module contains the rules for the conversion to AML. These can’t be tested without any additional code for the actual conversion process. Therefore, no Unit Tests or any other tests are possible for this module.

The system tests that cover the whole functionality of this program contain tests to guarantee the correct function of this module as well. The system test plan provides more information about these tests.

10. Summary

The Library module is the most important and powerful module of all. It fulfils a lot of tasks and contains the ruleset for all the conversions.

A huge strength of this module is the extensibility to other problems related to conversion from XML based files to AML. For example it is possible to extend the conversion with other rule sets to enable the conversion from other XML based file types like ESI, SDDML, EDS, PB-GSD or XDD.

But the module has its limits: For example it is not possible to re-convert the AML file into one of the given DD-file formats or even re-convert the file back into its original format.

Another issue is the complexity of this module. On the one hand it is powerful but on the other hand, the source code is divided into many files, classes and functions which make it difficult to understand and to extend or enhance it.

The design of the library works but however it would be sensible to restructure and modularize it even further.

11. Appendix

11.1. References

[1] SRS: *https://github.com/WAntonia/TINF18C_Team_3_DD2AML-Converter/wiki/System-Requirements-Specification*

[2] STP: *https://github.com/WAntonia/TINF18C_Team_3_DD2AML-Converter/wiki/Systemtestplan*

[3] STR: *https://github.com/WAntonia/TINF18C_Team_3_DD2AML-Converter/wiki/Systemtestreport*

11.2. Code

The source code to this module can be found:
https://github.com/WAntonia/TINF18C_Team_3_DD2AML-Converter/tree/master/SOURCE/src/Dd2Aml.Lib