XML-Struktur einer Firmware-Update-Definition

Bei manchen BiDiB-Knoten reicht die bisherige Aufteilung in eine Firmware- und EEPROM-Datei (000- bzw. 001.hex) wegen der vielfältigen Konfigurationsmöglichkeiten dieses Knotens nicht mehr aus. Zusätzlich wird noch zwischen Installation und Update unterschieden.

Aus diesem Grunde wurde für das fehlerfreie Aufspielen der Firmware auf einen BiDiB-Knoten eine Beschreibung erdacht. Mit ihrer Hilfe können Konfigurationsprogramme wie BiDiB-Monitor und BiDiB-Wizard passende Auswahlmenüs bereitstellen. Dadurch kann ein Anwender verschiedene Vorgänge, wie Neuinstallation oder Firmware-Update, komfortable und vor Allem verwechslungsfrei durchführen.

Das nachfolgende Beispiel zeigt die XML-Struktur welche vom Wizard ausgewertet werden kann. Das XML-File hat gemäss Konvention den Namen **firmware.xml** und wird im Zip-File neben den Hex-Files gespeichert.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="firmware_de.xsl"?>

<Firmware xsi:schemaLocation="http://www.bidib.org/schema/firmware
firmware.xsd" xmlns:firmware="http://www.bidib.org/schema/firmware"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.bidib.org/schema/firmware">

   <!-- Version -->
   <Version Version="0.1" Lastupdate="20140411" Author="BiDiB.org" Pid="201"
Vendor="013"
      Description="Firmware Definition for StepControl !!!Minimum-
Beispiel!!!"/>

   <!-- The firmware definition -->
   <FirmwareDefinition>
      <!-- Nodes are defined here ... -->
   </FirmwareDefinition>
```

Im `<FirmwareDefinition>`-Tag wird die Struktur der Firmware angegeben. Dabei werden unterschiedliche Node-Typen unterstützt:

- DeviceNode: Äusserer Node-Typ der die Firmware-Files eines Node zusammenfasst. Hier sollten
  - die Hersteller Id (`VID`),
  - die erweiterte Hersteller Id (`EVID`) falls notwendig (z.B. 258 für *OpenDCC*),
  - die Produkt Id (`PID`) sowie optional
  - ein Kommentar angegeben werden.
- FirmwareNode: Node-Typ der ein Firmware-File (hex) definiert. Hier muss die Destination des Hex-File, der Text sowie der Filename des HEX-File innerhalb des Archiv (Zip-File) angegeben werden.
- SimpleNode: Dieser Node-Typ kann verwendet werden um eine oder mehrere zusätzliche Abstraktions-Ebenen zu definieren.

Das nachfolgende Beispiel zeigt eine minimale Definition des <FirmwareDefinition>-Tag:

```
   <FirmwareDefinition>
      <Node xsi:type="DeviceNode" Comment="Kommentar" VID="13" EVID="258"
PID="132">
         <Nodetext Lang="de-DE" Text="Beispiel Firmware"/>
         <Nodetext Lang="en-EN" Text="Sample Firmware"/>
         <Node xsi:type="FirmwareNode" DestinationNumber="0" >
         ...
         </Node>
         ...
      </Node>
   </FirmwareDefinition>
```

Mit dem Attribute `DestinationNumber` wird der Zielspeicherbereich auf dem Knoten festgelegt (0 = Flash, 1 = EEPROM.). Details dazu sind auf bidib.org zu finden.

CV-Definitionsdatei

Es besteht auch die Möglichkeit eine CV-Definition anzugeben, welche dann im *firmware.zip* mitgeliefert werden muss:

```
   <FirmwareDefinition>
      <Node xsi:type="DeviceNode" Comment="Kommentar" VID="13" EVID="258"
PID="xyz">
         <Nodetext Lang="de-DE" Text="Beispiel Firmware"/>
         <Nodetext Lang="en-EN" Text="Sample Firmware"/>
         <Node xsi:type="FirmwareNode" DestinationNumber="0" >
         ...
         </Node>
         ...
         <CvFilename>BiDiBCV-13-xyz.xml</CvFilename>
      </Node>
   </FirmwareDefinition>
```

Ab dem Wizard-1.9 können CV-Dateien direkt in den Wizard importiert werden, falls sie in der `FirmwareDefinition` hinterlegt sind.

Ab dem Wizard-1.9.1 werden die CV-Dateien mit der Version versehen importiert. Das nachfolgende Beispiel würde die CV-Datei `BiDiBCV-13-201-0.1.xml` erzeugen:

```
   <Version Version="0.1" Lastupdate="20140411" Author="BiDiB.org" Pid="201"
Vendor="013"
      Description="Firmware Definition for StepControl !!!Minimum-
Beispiel!!!"/>
   <FirmwareDefinition>
```

```
      <Node xsi:type="DeviceNode" Comment="Kommentar" VID="13" EVID="258"
PID="201">
         <Nodetext Lang="de-DE" Text="Beispiel Firmware"/>
         <Nodetext Lang="en-EN" Text="Sample Firmware"/>
         <Node xsi:type="FirmwareNode" DestinationNumber="0" >
         ...
         </Node>
         ...
         <CvFilename>BiDiBCV-13-201.xml</CvFilename>
      </Node>
   </FirmwareDefinition>
```

Default-Labels

Die Default-Labels können im `firmware.xml` angegeben werden. Sie müssen an der letzten Stelle der Daten innerhalb der `FirmwareDefinition` eingefügt werden. Beim Laden des Firmware-Archiv erscheint dann im Wizard der Hinweis, ob die Default-Labels importiert werden sollen.

```
<Firmware ...>
   ...
   <FirmwareDefinition Version="2.06.00" Status="beta" ProtocolVersion="0.7"
RequiredMinVersion="2.01.00" >
      ...
      <!-- other parts -->
      <CvFilename>BiDiBCV-13-201.xml</CvFilename>

      <DefaultLabels>
         <DefaultLabelsFile Lang="de-DE" Filename="bidib-default-
names-13-138-de.xml" />
         <DefaultLabelsFile Lang="en-EN" Filename="bidib-default-
names-13-138-en.xml" />
      </DefaultLabels>

   </FirmwareDefinition>
</Firmware>
```


Alle referenzierten Dateien wie die *DefaultLabelsFile*s müssen im Firmware-ZIP enthalten sein.

Version, Status und Node-Images

Ab dem Wizard-1.12.2 werden die <`FirmwareDefinition`> in den CV-Dateien mit den Attributen `Version` und `Status` versehen importiert. Wenn `Version` angegeben ist, wird auf das Parsen der Filenamen zur Ermittlung der Version verzichtet. *Optional* können noch die Protokollversion `ProtocolVersion` sowie die aktuelle minimal Version `RequiredMinVersion` angegeben werden.

Folgende Werte sind als `Status` definiert:

- beta
- stable
- mandatory

Als weitere Änderung muss der CvFilename jetzt unter dem DeviceNode angegeben werden.

Neu besteht auch die Möglichkeit ein Bild der Baugruppe im Firmware-ZIP mitzuliefern. Dazu muss das Element NodeImages wie im Beispiel unten unter dem DeviceNode angegeben werden.

```
   <Version Version="0.1" Lastupdate="20190920" Author="BiDiB.org" Pid="132"
Vendor="013"
      Description="Firmware Definition for IF-2"/>

   <FirmwareDefinition Version="2.06.00" Status="beta" ProtocolVersion="0.7"
RequiredMinVersion="2.01.00" >
      <Node xsi:type="DeviceNode" Comment="BiDiB-IF2" VID="013" EVID="258"
PID="132">
         <Nodetext Lang="de-DE" Text="Beispiel Firmware"/>
         <Nodetext Lang="en-EN" Text="Sample Firmware"/>
         <Node xsi:type="FirmwareNode" DestinationNumber="0" >
         ...
         </Node>
         ...
         <!--
         <CvFilename>BiDiBCV-13-132.xml</CvFilename>

         <DefaultLabels>
             <DefaultLabelsFile Lang="de-DE" Filename="bidib-default-
names-13-138-de.xml" />
             <DefaultLabelsFile Lang="en-EN" Filename="bidib-default-
names-13-138-en.xml" />
         </DefaultLabels>
         -->

         <NodeImages>
             <Image>bidib-13-132.png</Image>
         </NodeImages>
      </Node>
   </FirmwareDefinition>
```

Alle referenzierten Dateien wie *NodeImage* oder *CV-Defintions-File* müssen im Firmware-ZIP enthalten sein.

Update Only

Um mit einem FW-Update nur die Firmware ohne Löschen der Einstellungen (Port-Konfiguration,

Makros, etc.) zu aktualisieren, wird oft die „Update Only“ Option angeboten. In einem solchen Fall wird das Attribut `IsUpdate=„true“` verwendet, wodurch die Tools erkennen, dass es sich um eine „Update Only“ Option handelt.

```
<Node xsi:type="DeviceNode" Comment="BiDiB-IF2 UPDATE" IsUpdate="true"
VID="013" PID="132" EVID="258">
```

In diesem Fall prüfen die Tools, ob die gleiche ProduktID und Major-Version verwendet wird, bevor diese Option im Firmware-Update-Dialog angeboten wird.

Changelog Informationen

```
<Firmware ...>
   ...
   <FirmwareDefinition Version="2.06.00" Status="beta" ProtocolVersion="0.7"
RequiredMinVersion="2.01.00" >
      ...
      <!-- other parts -->
      ...
      <Changelog>changelog.json</Changelog>
   </FirmwareDefinition>
</Firmware>
```

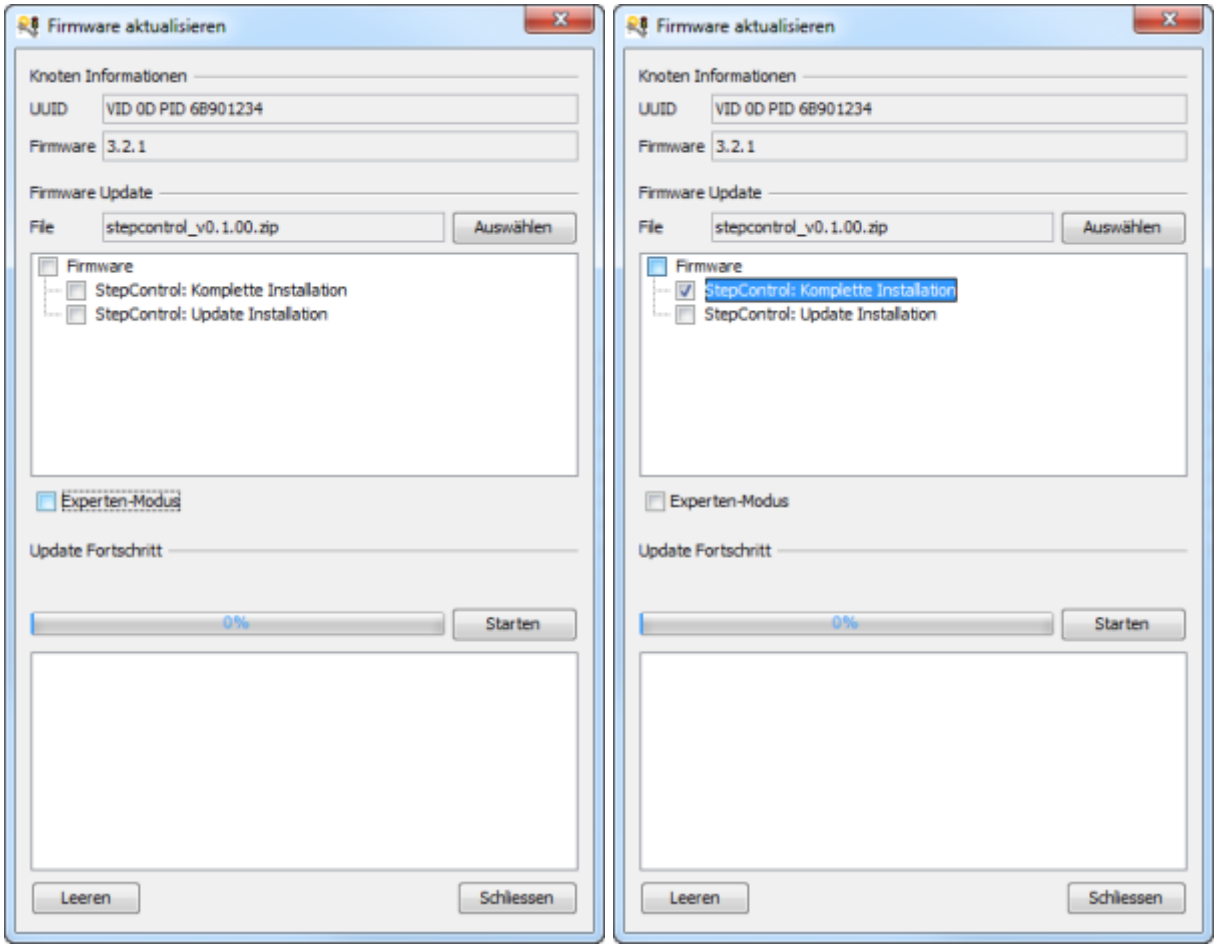

Die refenrenzierte json Datei muss Firmware-ZIP enthalten sein.

Die json Datei muss dabei dem folgenden Schema entsprechen.

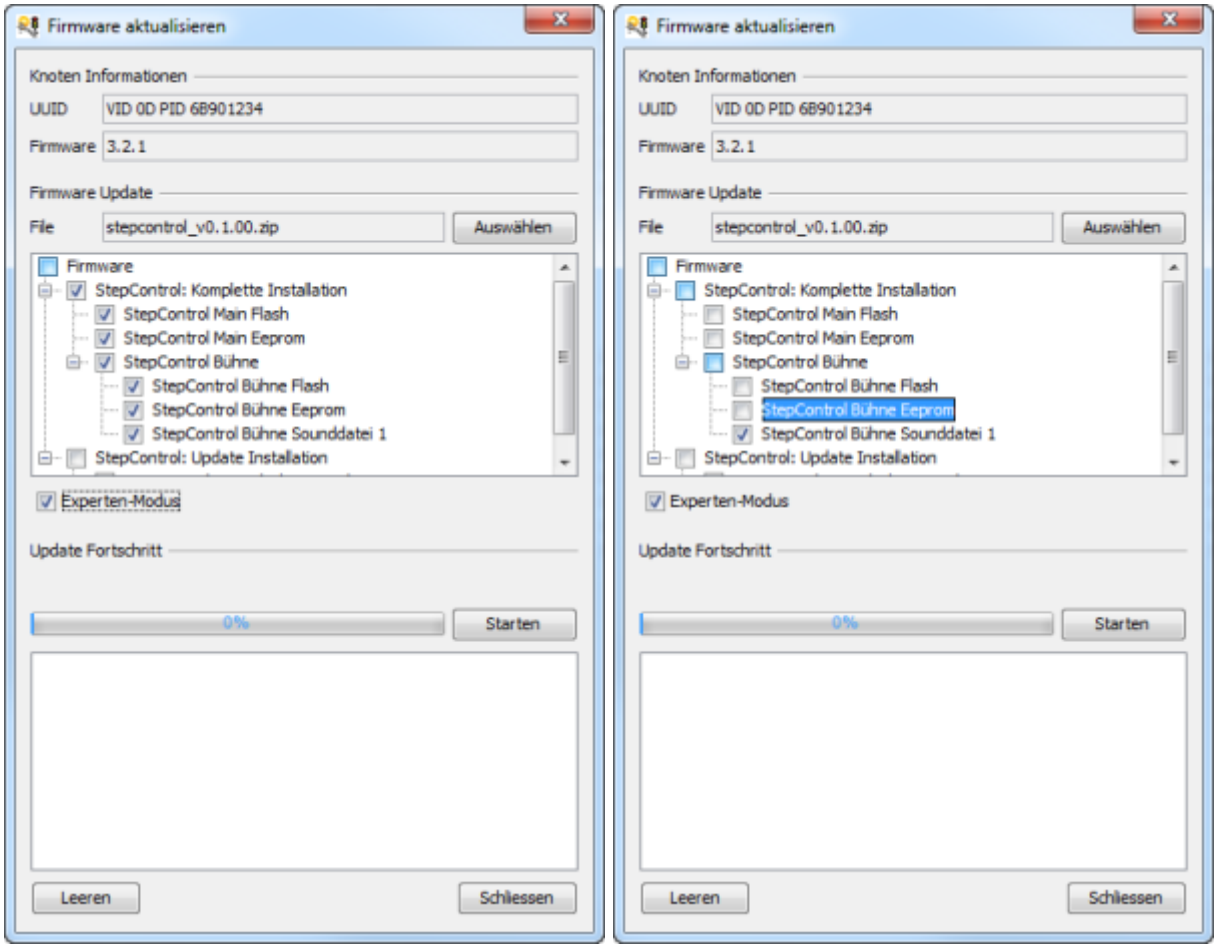
```
{
    "product": "",
    "owner": "",
    "versions": [
        {
            "version": "1.04.05",
            "releaseDate": "20210630",
            "changelog": [
                {
                    "lang": "de",
                    "description": "..."
                },
                ... weitere Übersetzungen
            ]
        },
        ... weitere Versionen
    ]
}
```

Darstellung im Wizard

Nach der Auswahl des Zip-File wird die Struktur im Wizard analysiert und in einem Tree angezeigt. Der Benutzer kann anschliessend die Firmware-Pakete auswählen welche übertragen werden sollen.



Im **Experten-Modus** hat der Anwender Zugriff auf die einzelnen Firmware-Files und kann dadurch einzelne Files übertragen.



Durch Klick auf den Starten-Button werden die selektierten Firmware-Files auf den Knoten übertragen.

Legacy-Support

Falls kein firmware.xml vorhanden ist, kann auch ein einzelnes Hex-File ausgewählt werden. Der Wizard versucht dann anhand bisheriger Konvention die Destination zu ermitteln (*.000.hex > Flash, *.001.hex > Eeprom) und erzeugt dynamisch ein entsprechendes firmware.xml welches für die Darstellung im Tree benutzt wird.

Fiktives Beispiel für die StepControl

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="firmware_de.xsl"?>

<Firmware xsi:schemaLocation="http://www.bidib.org/schema/firmware
firmware.xsd" xmlns:firmware="http://www.bidib.org/schema/firmware"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.bidib.org/schema/firmware">
   <Version Version="0.1" Lastupdate="20140411" Author="BiDiB.org" Pid="201"
Vendor="013"
      Description="Firmware Definition for StepControl !!!Minimum-
Beispiel!!!"/>
   <FirmwareDefinition Version="1.04.00" Status="stable"
```

```
ProtocolVersion="0.7" RequiredMinVersion="1.01.00">
      <Node xsi:type="DeviceNode" VID="013" EVID="258" PID="201" >
         <Nodetext Lang="de-DE" Text="StepControl: Komplette Installation"/>
         <Nodetext Lang="en-EN" Text="StepControl: complete installation"/>
         <Node xsi:type="FirmwareNode" DestinationNumber="0" >
            <Nodetext Lang="de-DE" Text="StepControl Main Flash"/>
            <Nodetext Lang="en-EN" Text="StepControl Main Flash"/>
            <Filename>StepControl_v0.1.0.000.hex</Filename>
         </Node>
         <Node xsi:type="FirmwareNode" DestinationNumber="1" >
            <Nodetext Lang="de-DE" Text="StepControl Main Eeprom"/>
            <Nodetext Lang="en-EN" Text="StepControl Main Eeprom"/>
            <Filename>StepControl_v0.1.0.001.hex</Filename>
         </Node>
         <Node xsi:type="SimpleNode" >
            <Nodetext Lang="de-DE" Text="StepControl Bühne"/>
            <Nodetext Lang="en-EN" Text="StepControl Platform"/>
            <Node xsi:type="FirmwareNode" DestinationNumber="2" >
               <Nodetext Lang="de-DE" Text="StepControl Bühne Flash"/>
               <Nodetext Lang="en-EN" Text="StepControl Platform Flash"/>
               <Filename>StepControl_v0.1.0.002.hex</Filename>
            </Node>
            <Node xsi:type="FirmwareNode" DestinationNumber="3" >
               <Nodetext Lang="de-DE" Text="StepControl Bühne Eeprom"/>
               <Nodetext Lang="en-EN" Text="StepControl Platform Eeprom"/>
               <Filename>StepControl_v0.1.0.003.hex</Filename>
            </Node>
            <Node xsi:type="FirmwareNode" DestinationNumber="4" >
               <Nodetext Lang="de-DE" Text="StepControl Bühne Sounddatei
1"/>
               <Nodetext Lang="en-EN" Text="StepControl Platform Soundfile
1"/>
               <Filename>StepControl_v0.1.0.004.hex</Filename>
            </Node>
         </Node>
      </Node>
      <Node xsi:type="DeviceNode" VID="013" EVID="258" PID="201" >
         <Nodetext Lang="de-DE" Text="StepControl: Update Installation"/>
         <Nodetext Lang="en-EN" Text="StepControl: update installation"/>
         <Node xsi:type="FirmwareNode" DestinationNumber="0" >
            <Nodetext Lang="de-DE" Text="StepControl Main Flash nur
Update"/>
            <Nodetext Lang="en-EN" Text="StepControl Main Flash Update
Only"/>
            <Filename>StepControl_update_v0.1.0.000.hex</Filename>
         </Node>
      </Node>
   </FirmwareDefinition>
```

```
</Firmware>
```

Firmware-Repo

Firmware-Pakete im ZIP-Format (mit firmware.xml) können im Firmware-Repo auf github comitted und gepushed werden. Nach einem neuen Commit läuft ein Job im Gitlab-Repository welches das Firmware-Paket scanned (auf Vorhandensein des firmware.xml) und die Meta-Daten (PID, VID, EVID, Version, Name, etc.) aus dem firmware.xml ausliest. Mit diesen Meta-Daten wird dann ein json-Dokument erzeugt, welches den aktuellen Stand im Firmware-Repo beinhaltet. Anschliessend wird das json-Dokument und alle geänderten Firmware-Pakete auf den `bidib.org` Server kopiert.

Dieses json-Dokument (http://bidib.org/node_firmware/repository.json) kann heruntergeladen werden und wird z.B. auch vom BiDiB-Monitor verwendet um die Firmware-Daten im Knoten-Firmware-Repository anzuzeigen.

Neue Firmware hinzufügen

Um ein neues Firmware-Paket hinzuzufügen, muss das Firmware-Paket im ZIP-Format in den entsprechenden Unterordner von `repo` kopiert werden.
Die Unterordner sind anhand der `VID` bezeichnet, für OpenDCC also die 13, Tams ist 62, Fichtelbahn ist 251.

Nach Commit und Push läuft der Job im Gitlab-Repo.

Bestehende Firmware überschreiben

Um ein bestehendes Firmware-Paket zu überschreiben kann man das neue Firmware-Paket einfach über das alte kopieren. Bei Merge-Konflikten einfach die Änderungen des neuen Firmware-Paket übernehmen.
Es ist nicht notwendig das alte Firmware-Paket zuerst zu löschen.

Nach Commit und Push läuft der Job im Gitlab-Repo.