



Elsys AG
Mellingerstrasse 12
CH – 5443 Niederrohrdorf
www.elsys.ch

TPC5 und TPS5 Datei Spezifikation

(TransAS 3)

1.5

Change Log:

Version 1.5 08.10.2012	Kleine Korrekturen
Version 1.4 18.01.2011	Neues Attribut: „deviceName“ in der Gruppe „channels“. Organisation von berechneten Kurven Organisation von Spektren (TPS5-Format)

Inhaltsverzeichnis

1	Grundlagen	3
2	Datenorganisation	4
2.1	Gemessene Daten	4
2.1.1	Root Gruppe.....	4
2.1.2	Gruppe measurements	4
2.1.3	Gruppe channels	4
2.1.4	Gruppe blocks	5
2.1.5	DataSets	5
2.2	Berechnete Daten	6
2.2.1	Gruppe channels	6
2.3	Spektrumdaten (TPS5)	7
2.3.1	Gruppe channels	7
2.3.2	Gruppe blocks	8
2.3.3	DataSets	8
3	Messwerte Interpretation	9
3.1	Skalierung der Messwerte.....	9
3.2	Aufteilung Signal- / Marker-Bits	9
4	Zusatzinformationen zur Messung	9
4.1	Dataset „autosequence“	9
4.2	Dataset „formulaeditor“	9
4.3	Dataset „hw-settings“	9
4.4	Dataset „layout“	9
4.5	Dataset „settings“	9
5	Programmierung.....	11
5.1	Anleitung zur Verwendung der Bibliothek	11
6	Hilfsmittel	12

1 Grundlagen

Die TPC5 sowie die TPS5 Dateien basieren auf dem HDF5 Datenformat, welches von der HDF Group an der Universität von Illinois entwickelt wurde. Die Spezifikation von HDF5 sowie der Source Code verschiedenerer Libraries sind frei verfügbar.

Siehe : <http://www.hdfgroup.org/HDF5/index.html>

In den TPC5 Dateien werden HDF5 Gruppen und HDF5 Datasets verwendet um die Messkurven zu organisieren und abzuspeichern. Aufnahmeinformationen wie zum Beispiel die Aufnahmezeit werden in Attributen gespeichert, welche den Gruppen zugeordnet werden können.

Die Messkurven werden mit verschiedenen Datenreduktionen abgespeichert. Dies ermöglicht eine beschleunigte Anzeige in TransAS bei grossen Aufnahmen.

Seit der TransAS Version 3.1.5.46 ist es möglich die Hardware Einstellungen, das Layout, die Formeln aus dem Formeleditor sowie die benutzte Autosequenz in die TPC5 Datei mit abzuspeichern.

Das Dateiformat HDF5 bietet ebenfalls die Möglichkeit die einzelnen Datasets mit gzip zu komprimieren. Diese Option ist in TransAS ein- und ausschaltbar.

Weitere Angaben zu den hier aufgeführten Begriffen befinden sich in der TpcAccess API Dokumentation.

2 Datenorganisation

2.1 Gemessene Daten

Abbildung 1 zeigt die interne Datenorganisation einer TPC5 Datei. Die Ordnersymbole stellen HD5 Gruppen, die Tabellensymbole stellen HDF5 Datasets dar. Ordner bei denen ein „A“ erscheint, wurden Attributeigenschaften definiert. Ordner mit numerischen Namen repräsentieren je eine Messung, einen Kanal oder einen Block. Sie werden bei 1 beginnend durchnummeriert.

Zusatzinformationen zur Messung werden als Textsymbole dargestellt.

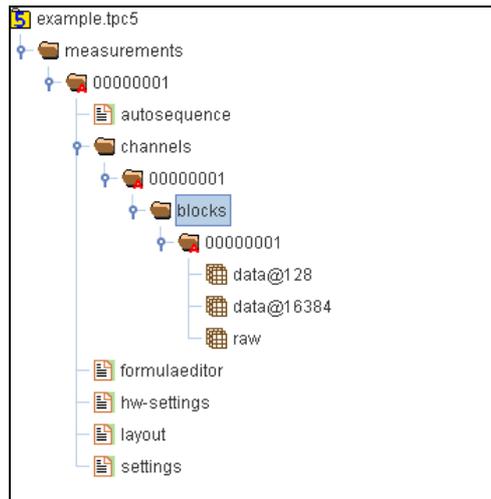


Abbildung 1 - Datenorganisation

2.1.1 Root Gruppe

Die oberste Ebene im TPC5 File ist die Root Gruppe „/“ für welche ebenfalls Attribute gesetzt werden.

Attribute Name	Attribute Wert	HDF5 Datentyp
filetype	TransAsData	H5T_C_S1, H5T_VARIABLE
format	x	H5T_STD_I32LE
compatible-format	x	H5T_STD_I32LE
Compression	X	H5T_STD_I32LE
creator	TransAS xxxxxx	H5T_C_S1, H5T_VARIABLE

- format: Definiert die Version des Datenformates
- compatible-format: Definiert zu welcher Version das aktuelle Format kompatibel ist
- Den Wert des Attributes *creator* kann frei gewählt werden.

2.1.2 Gruppe measurements

Diese Gruppe repräsentiert eine Messung. Im Moment ist es nicht möglich mehrere Messungen in einem File zu speichern, daher erscheint immer nur eine Untergruppe von „measurements“ mit dem Namen „00000001“. Für diese Untergruppe werden folgende Attribute gesetzt.

Attribute Name	Attribute Wert	HDF5 Datentyp
name	M1	H5T_C_S1, H5T_VARIABLE

2.1.3 Gruppe channels

In dieser Gruppe werden pro vorhandenen Kanal eine Kanalgruppe in der Form „xxxxxxx“ erstellt. Die Gruppennamen werden von 1 beginnend durchnummeriert. Wenn bei einer Messung nicht alle Kanäle gespeichert werden, ist die Nummerierung trotzdem durchgehend. Sollten mehrere Boards vorhanden sein, beginnt die Nummerierung bei Board A und ist über alle Boards durchgehend.

Bei den einzelnen Kanalgruppen werden folgende Attribute gesetzt.

Attribute Name	Attribute Wert	HDF5 Datentyp
analogMask		H5T_STD_I32LE
markerMask		H5T_STD_I32LE
binToVoltFactor		H5T_IEEE_F64LE
binToVoltConstant		H5T_IEEE_F64LE
name	Kanal Name	H5T_C_S1, H5T_VARIABLE
deviceName	Device Name	H5T_C_S1, H5T_VARIABLE
physicalUnit	X	H5T_C_S1, H5T_VARIABLE
voltToPhysicalConstant		H5T_IEEE_F64LE
voltToPhysicalFactor		H5T_IEEE_F64LE
markerNames	Name1;Name2;	H5T_C_S1, H5T_VARIABLE
rangeMin	x	H5T_IEEE_F64LE
rangeMax	x	H5T_IEEE_F64LE
ChannelName	Physikalischer Kanal Name	H5T_STD_I32LE
ChannelType	None, FFT, AVG	H5T_C_S1, H5T_VARIABLE
UniqueInputID	Reserved	H5T_C_S1, H5T_VARIABLE
color	ARGB	H5T_STD_I32LE

- rangeMin, rangeMax: Min, Max Werte des Eingangsverstärkers (Skaliert in physikalische Einheit)

Weitere Informationen über die Verwendung dieser Attribute siehe: 3.1 Skalierung der Messwerte.

2.1.4 Gruppe blocks

In dieser Gruppe werden pro vorhandenen Block in der Messung eine Blockgruppe in der Form „xxxxxxx“ erstellt. Die Blockgruppennamen werden von 1 beginnend durchnummeriert. Wenn bei einer Messung nicht alle Blöcke gespeichert werden, ist die Nummerierung trotzdem durchgehend. Es ist immer mindestens ein Block vorhanden (Scope Mode, Continuous Mode).

Bei den einzelnen Blockgruppen werden folgende Attribute gesetzt.

Attribute Name	Attribute Wert	Typ
sampleRateHertz		H5T_IEEE_F64LE
startTime	YYYY-MM-DDThh:mm:ss.pppppppp	H5T_C_S1, H5T_VARIABLE
triggerSample		H5T_STD_I64LE
triggerTimeSeconds		H5T_IEEE_F64LE
relativeDivisor		H5T_STD_I32LE

- startTime ist die Zeit bei welcher die Messung gestartet wurde (Start in TransAS).
- triggerSample ist die Nummer des Samples bei welchem die Messung getriggert hat.
- triggerTimeSeconds ist die Zeitspanne in s zwischen der Startzeit der Messung und dem Triggerzeitpunkt
- relativeDivisor: Datenreduktionsfaktor zwischen den einzelnen Datasets.

2.1.5 DataSets

Der Aufbau und das Format der Datasets hängen davon ab ob die Daten aus physikalischen Messungen stammen oder durch den Formeleditor berechnet wurden. Physikalische Messdaten sind jeweils im UINT16 Format abgespeichert und enthalten auch die Markerbits. Berechnete Kurven sind in 32Bit Floatingpoint Format abgespeichert und schon in der physikalischen Einheit skaliert.

Im Datasets „raw“ werden die physikalischen Messdaten im UINT16 Format abgespeichert. In den Datasets „data@xxx“ werden die Hüllkurven in verschiedenen Datenreduktionsraten abgespeichert. Die Hüllkurve wird berechnet indem aus X Samples der minimal und maximal Wert ermittelt wird. Dieses Werte Paar wird im Dataset abgespeichert. Der Wert nach dem @ bezeichnet die Anzahl Samples aus welchem ein Min/Max Wertepaar ermittelt wird. Somit ist die Anzahl der Wertepaare in einem Hüllkurven Dataset gleich Blocksize / X. Je nach Blockgröße werden mehr oder weniger an Hüllkurven Datasets erstellt. Für X wird ein Vielfaches von 128 verwendet bis die Anzahl Samples 1 ist. Bei

Continuous oder ECR Aufnahmen ist die Blocklänge kein vielfaches von 128. Hier wird das letzte Min/Max Paar aus den restlichen Samples berechnet die bei einer Teilung durch 128 (und vielfache davon) anfallen.

Beispiel physikalische Messkurve:

Data Set Name	Beschreibung	Grösse
data@128	Min und Max Werte von 128 Samples	2x Anzahl Samples / 128
data@16384	Min und Max Werte von 16384 Samples	2x Anzahl Samples / 16384
data@2097152	Min und Max Werte von 2 ²¹ Samples	2x Anzahl Samples / 2 ²¹
data@268435456	Min und Max Werte von 2 ²⁸ Samples	2x Anzahl Samples / 2 ²⁸
raw	Raw Daten als 16 bit unsigned integer	Anzahl Samples

Beispiel berechnete Messkurve:

Data Set Name	Beschreibung	Grösse
data	Berechnete Volt Werte	Anzahl Samples
data@128	Min und Max Werte von 128 Samples	2x Anzahl Samples / 128
data@16384	Min und Max Werte von 16384 Samples	2x Anzahl Samples / 16384

2.2 Berechnete Daten

Werden in TransAS Kurven, welche mit Hilfe des Formeleditors berechnet wurden, in eine TPC5 Datei abgespeichert, ist die Organisation der Daten in der Datei ein wenig anders. Es werden im Folgenden nur noch die Unterschiede zur oben beschriebenen Struktur erläutert.

Die Abbildung 2 zeigt die Datenstruktur einer berechneten Kurve. Im Gegensatz zu einer gemessenen Kurve ist kein Raw Datensets vorhanden. Zudem sind die Datensets bereits in der physikalischen Einheit skaliert und als Fließkommazahl abgespeichert.

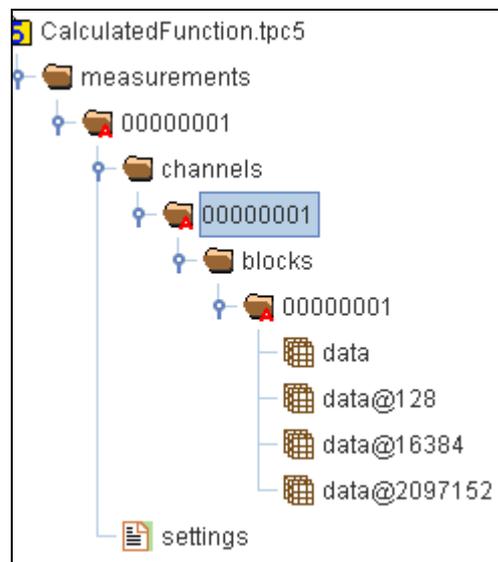


Abbildung 2

2.2.1 Gruppe channels

Bei den einzelnen Kanalgruppen werden folgende Attribute gesetzt.

Attribute Name	Attribute Wert	HDF5 Datentyp
name	Kanal Name	H5T_C_S1, H5T_VARIABLE
deviceName	Device Name	H5T_C_S1, H5T_VARIABLE
physicalUnit		H5T_C_S1, H5T_VARIABLE

markerNames	Name1;Name2;	H5T_C_S1, H5T_VARIABLE
rangeMin	x	H5T_IEEE_F64LE
rangeMax	x	H5T_IEEE_F64LE

- Berechnete Kurven sind bereits skaliert, daher sind binToVoltFactor, binToVoltConstant 0, voltToPhysicalFactor und voltToPhysicalConstant nicht vorhanden.

2.3 Spektrumdaten (TPS5)

TransAS ermöglicht es, berechnete Spektren in eine TPS5 Datei zu speichern. Die TPS5 ist ebenfalls eine HDF5 Datei mit ähnlichem Aufbau wie bei der TPC5 Datei. Die Abbildung 3 zeigt die Struktur der Datei. Bis zur Ebene „channels“ ist sie identisch mit der TPC5 Datei. Die „channel“ Attribute sowie die Datensätze und „blocks“ unterscheiden sich jedoch von der TPC5 Datei.

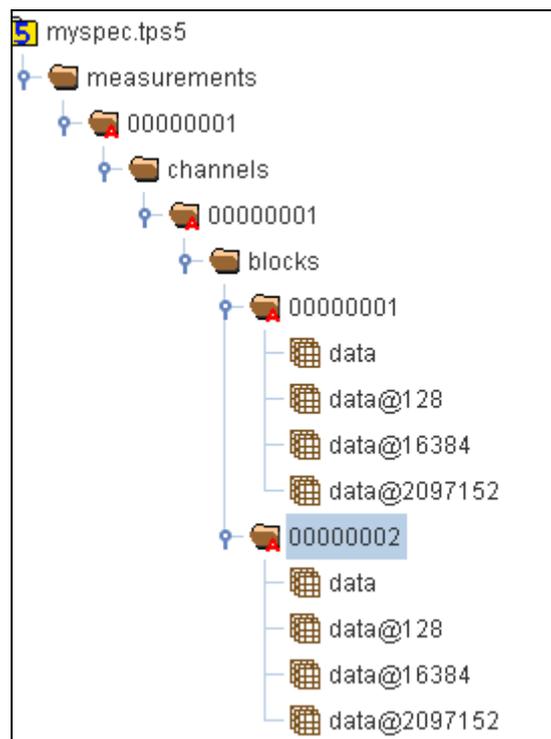


Abbildung 3

2.3.1 Gruppe channels

In dieser Gruppe werden pro vorhandenen Kanal eine Kanalgruppe in der Form „xxxxxxx“ erstellt. Die Gruppennamen werden von 1 beginnend durchnummeriert. Wenn bei einer Messung nicht alle Kanäle gespeichert werden, ist die Nummerierung trotzdem durchgehend. Sollten mehrere Boards vorhanden sein, beginnt die Nummerierung bei Board A und ist über alle Boards durchgehend.

Bei den einzelnen Kanalgruppen werden folgende Attribute gesetzt.

Attribute Name	Attribute Wert	HDF5 Datentyp
name	Kanal Name	H5T_C_S1, H5T_VARIABLE
physicalUnit	X	H5T_C_S1, H5T_VARIABLE
ChannelName	Physical channel name	H5T_STD_I32LE
ChannelType	None, FFT, AVG	H5T_C_S1, H5T_VARIABLE
UniqueInputID	Reserved	H5T_C_S1, H5T_VARIABLE
color	ARGB	H5T_STD_I32LE
rangeMin	x	H5T_IEEE_F64LE
rangeMax	x	H5T_IEEE_F64LE
deviceName	Device Name	H5T_C_S1, H5T_VARIABLE
markerNames	Name1;Name2	H5T_C_S1, H5T_VARIABLE

2.3.2 Gruppe blocks

Die Bezeichnung „blocks“ wurde aus der TPC5 Datei übernommen, hat jedoch nichts mit den Blöcken einer Messung zu tun sondern dient lediglich dazu die gleiche Strukturtiefe für TPC5 und TPS5 Dateien zu erhalten.

In dieser Gruppe wird für jede Kurve eine Gruppe mit dem Namen „00000001“ und eine mit „00000002“ erstellt. Die Gruppe „00000001“ enthält die Datensets mit den Amplituden Werten des Spektrums, die Gruppe „00000002“ enthält die Datensets mit den Phase Werten des Spektrums.

Bei den einzelnen Blockgruppen werden folgende Attribute gesetzt.

Attribute Name	Attribute Wert	Typ
sampleRateHertz		H5T_IEEE_F64LE
startTime	YYYY-MM-DDThh:mm:ss.pppppppp	H5T_C_S1, H5T_VARIABLE
stopTriggerSample		H5T_STD_I64LE
timeBaseSource	Internal/External	H5T_C_S1, H5T_VARIABLE
triggerSample		H5T_STD_I64LE
triggerTimeSeconds		H5T_IEEE_F64LE
relativeDivisor		H5T_STD_I32LE

- startTime ist die Zeit bei welcher die Messung gestartet wurde (Start in TransAS).
- triggerSample ist die Nummer des Samples bei welchem die Messung getriggert hat.
- triggerTimeSeconds ist die Zeitspanne in s zwischen der Startzeit der Messung und dem Triggerzeitpunkt
- relativeDivisor: Datenreduktionsfaktor zwischen den einzelnen Datasets.

2.3.3 DataSets

Die Spektrum DataSets werden wie die berechneten Kurven einer TPC5 Datei als 32 Bit Fließkommazahl abgespeichert. Ebenfalls werden Hüllkurven DataSets in die Datei mit abgespeichert.

3 Messwerte Interpretation

3.1 Skalierung der Messwerte

Die Messwerte aus gemessenen Kurven sind in einem binären Word-Format abgelegt. Diese Werte können mit folgender Formel in Volt skaliert werden:

$$\text{Signal [V]} = \text{Signal [Binär]} * \text{binToVoltFactor} + \text{binToVoltConstant}$$

Um die Volt-Werte in physikalische Werte zu skalieren muss folgende Formel angewendet werden:

$$\text{Signal [Phys.]} = \text{Signal [V]} * \text{voltToPhysicalFactor} + \text{voltToPhysicalConstant}$$

3.2 Aufteilung Signal- / Marker-Bits

Je nach AD-Wandler Einstellung wird nur ein Teil des 16-Bit Daten-Wortes für das Analogsignal benötigt. Die restlichen Bits werden als Marker verwendet. Die Signalbits sind immer linksbündig im 16-Bit Wort.

Das niederwertigste Bit ist das Marker-Bit 1. Wie viele und welche Bits das Analogsignal belegt kann aus dem Attribute *analogmask* oder *markermask* ausgelesen werden. Durch eine AND Verknüpfung des binären Wertes mit der *markermask* erhält man die digitalen Marker Signale. Durch eine AND Verknüpfung des binären Wertes mit der *analogmask* erhält man das analoge Signal.

4 Zusatzinformationen zur Messung

4.1 Dataset „autosequence“

In diesem Dataset wird die verwendete Autosequenz als Text abgelegt. Wurde keine Autosequenz verwendet, ist dieses Dataset leer.

4.2 Dataset „formulaeditor“

In diesem Dataset werden die verwendeten Formeln aus dem Formeleditor als Text gespeichert. Wurden keine Formeln verwendet, ist dieses Dataset leer.

4.3 Dataset „hw-settings“

In diesem Dataset werden die Hardware Einstellungen aus dem Control Panel als XML gespeichert.

4.4 Dataset „layout“

In diesem Dataset werden die Layout Einstellungen von TransAS als XML gespeichert.

4.5 Dataset „settings“

In diesem Dataset werden die Attribute und Record Logs, welche in TransAS generiert werden, in einer XML Struktur abgelegt. Die XML Struktur sieht wie folgt aus. In diesem Beispiel wurden 2 Record Logs und 2 Attribute gesetzt.

```
<?xml version="1.0" encoding="utf-16"?>
<Tpc5MeasurementInternalData xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xmlns:xsd="http://www.w3.org/2001/XMLSchema" Version="1.0"
xmlns="http://www.elsys.ch/schema/transas/tpc5filedata-1.x.xsd">
  <RecordingLog Origin="">
    <RecordingLogEntries>
      <RecordingLogEntry>
        <Time>2008-09-25T17:11:36</Time>
        <Text>RecordLog1</Text>
      </RecordingLogEntry>
      <RecordingLogEntry>
        <Time>2008-09-25T17:11:36.05</Time>
        <Text>RecordLog1</Text>
      </RecordingLogEntry>
    </RecordingLogEntries>
  </RecordingLog>
</Tpc5MeasurementInternalData>
```

```
</RecordingLogEntries>
<RecordingLogAttributes>
  <RecordingLogAttribute>
    <Name>MusterAttribut1</Name>
    <Text>Dies ist ein Muster Attribute</Text>
  </RecordingLogAttribute>
  <RecordingLogAttribute>
    <Name>MusterAttribute2</Name>
    <Text>Dies ist ein weiteres Attribute</Text>
  </RecordingLogAttribute>
</RecordingLogAttributes>
</RecordingLog>
</Tpc5MeasurementInternalData>
```

5 Programmierung

Auf der oben genannten Website stehen mehrere Bibliotheken zur Verfügung mit welchen HDF5/ TPC5 Files erstellt oder ausgelesen werden können. Die auf der Website verfügbare High Level (HL) Library kann nicht verwendet werden da nicht alle benötigten Einstellungen implementiert sind.

5.1 Anleitung zur Verwendung der Bibliothek

Funktion	C API
File erstellen	<pre> hid_t plist = H5Pcreate(H5P_FILE_CREATE); hid_t aplist = H5Pcreate(H5P_FILE_ACCESS); hid_t hdfFile = H5Pcreate(„Filename.tpc5“, H5F_ACC_TRUNC, plist, aplist); H5Pclose(plist); H5Pclose(aplist); </pre>
Gruppe erstellen	<pre> hid_t linkcreatprop, groupcreatprop, groupaccessprop; linkcreatprop = H5Pcreate(H5P_LINK_CREATE); groupcreatprop = H5Pcreate(H5P_GROUP_CREATE); groupaccessprop = H5Pcreate(H5P_GROUP_ACCESS); hid_t Group = H5Gcreate2(Id, „GroupName“, linkcreatprop, groupcreatprop, groupaccessprop); H5Pclose(linkcreatprop); H5Pclose(groupcreatprop); H5Pclose(groupaccessprop); </pre>
Attribute setzen	<pre> // String Attribut hid_t dataspace = H5Screate(H5S_SCALAR); hid_t typeId = H5Tcopy(H5T_C_S1); H5Tset_size(typeId, H5T_VARIABLE); hid_t attrId = H5Acreate2(Id, keyString, typeId, dataspace, 0, 0); H5Awrite(attrId, typeId, &Value); H5Tclose(typeId); H5Aclose(attrId); H5Sclose(dataspace); // INT32 Attribut hid_t dataspace = H5Screate(H5S_SCALAR); hid_t attrId = H5Acreate2(Id, keyString, H5T_NATIVE_INT, dataspace, 0, 0); H5Awrite(attrId, H5T_NATIVE_UINT, &i32value); H5Aclose(attrId); H5Sclose(dataspace); //Float Attribut hid_t dataspace = H5Screate(H5S_SCALAR); hid_t attrId = H5Acreate2(Id, keyString, H5T_NATIVE_DOUBLE, dataspace, 0, 0); H5Awrite(attrId, H5T_NATIVE_DOUBLE, &dbvalue); H5Aclose(attrId); H5Sclose(dataspace); //INT64 Attribut hid_t dataspace = H5Screate(H5S_SCALAR); hid_t attrId = H5Acreate2(Id, keyString, H5T_NATIVE_INT64, dataspace, 0, 0); H5Awrite(attrId, H5T_NATIVE_UINT64, &i64value); H5Aclose(attrId); H5Sclose(dataspace); </pre>
Dataset erstellen	<pre> hid_t cparms, linkprop, accessprop; herr_t status; hsize_t dims[1]; dims[0] = 0; hsize_t maxdims[1] = {H5S_UNLIMITED}; hsize_t chunk_dims[1] = {0x400}; hid_t dataspace; /* Create the data space with unlimited dimensions. */ dataspace = H5Screate_simple(1, dims, maxdims); /* Modify dataset creation properties, i.e. enable chunking */ cparms = H5Pcreate(H5P_DATASET_CREATE); status = H5Pset_chunk(cparms, 1, chunk_dims); /* Create Link Creation Property List */ linkprop = H5Pcreate(H5P_LINK_CREATE); /* Create Dataset Access Property List */ accessprop = H5Pcreate(H5P_DATASET_ACCESS); </pre>

	<pre> /* Create a new dataset within the file using cparms creation properties. */ hid_t DataSetID = H5Dcreate2 (GroupID,"raw", H5T_NATIVE_UINT16, dataspace, linkprop, cparms, accessprop); H5Pclose(accessprop); H5Pclose(linkprop); H5Pclose(cparms); H5Sclose(dataspace); </pre>
Dataset schreiben	<pre> hid_t filespace,dataspace; hsize_t inputdims[1],size[1]; herr_t status; size[0] = uiNbOfSamples; inputdims[0] = uiNbOfSamples ; status = H5Dset_extent(DataSetID, size); /* Select a hyperslab */ filespace = H5Dget_space (DataSetID); hsize_t offset[1]; offset[0] = m_i64Size; status = H5Sselect_hyperslab (filespace, H5S_SELECT_SET, offset, NULL, inputdims, NULL); /* Define memory space */ dataspace = H5Screate_simple (1,inputdims , NULL); /* Write the data to the hyperslab */ status = H5Dwrite (m_DataSetID, H5T_NATIVE_UINT16, dataspace, filespace, H5P_DEFAULT, data); H5Sclose(dataspace); H5Sclose(filespace); </pre>

Anmerkung: Die File und Gruppen Close Befehle sind hier nicht aufgeführt sind aber zwingend notwendig. Werden sie nicht ausgeführt, bleibt das File für andere Anwendungen gesperrt und der verwendete Speicher der Library wird nicht freigegeben.

6 Hilfsmittel

Auf der oben genannten Website findet man das Tool „HDFView“. Mit diesem Tool können TPC5 und TPS5 Dateien geöffnet werden. Dieses Tool ist sehr hilfreich um den Aufbau der Datei zu verstehen.

