

# UIDEP Definition Version 2.2

*Universal Immission Data Exchange Protocol*

Universelles Kommunikationssystem für Luftgüte-Messnetzkomponenten wie Zentralen, Stationsrechner, Messgeräte u.a.m.

Autoren: Ing. Erich Kitzmüller, Ambient Informtion Processing GmbH, und Dipl. Ing. Alfred Jakob

10.04.2020

## Inhaltsverzeichnis

Präambel.....	3
1. Funktionsübersicht.....	4
2. Protokollstruktur.....	5
3. Datenübertragung.....	11
3.1. Von einem Messgerät bereitgestellte Webdienste (URL) für die Abfrage von Messwerten.....	11
3.2. Von einem Messgerät bereitgestellte Webdienste (URL) für die Steuerung des Betriebsmodus (z.B. Einleiten einer Funktionskontrolle).....	13
3.3. Von einem Messgerät bereitgestellte Webdienste (URL) für die Konfigurierung.....	14
3.4. Von einem MSR bereitgestellte Webdienste (URL) für die Abfrage von Messwerten.....	15
Die Erläuterungen von 3.1 gelten hier analog.....	15
3.5. Von einem Messstellenrechner bereitgestellte Webdienste (URL) für die Konfigurierung.....	16
Die Erläuterungen von 3.3 gelten hier analog.....	16
3.6. Von einem Messstellenrechner bereitgestellte Webdienste (URL) für die Übertragung von Dokumenten.....	17
3.7. Von der Messnetzzentrale bereitgestellte Webdienste (URL) für Eigenmeldungen.....	17
3.8. Abfrage der Fähigkeiten eines Geräts.....	18
3.9. Abfrage der Zeit eines Geräts.....	20
4. Liste der Deskriptoren.....	21

## Änderungshistorie

Version	Datum	Änderung
2.0	25.7.2019	Komplette Neugestaltung der Spezifikation
2.1	21.11.2019	Empfehlungen für Betriebsmodi, Betriebsstati etc.
2.2	10.4.2020	Darstellung des Betriebsmodus bei Geräten, die pro Komponente unterschiedliche Betriebsmodi erlauben; Betriebsmodus „G“

## Präambel

Bislang ist als herstellerübergreifendes Protokoll für die Kommunikation zwischen Messgeräten und Messstationsrechnern das Bayern-Hessen-Protokoll üblich, das sowohl mit seriellen Schnittstellen als auch im LAN (TCP, UDP) verwendet werden kann. Dieses aus der Zeit der seriellen Übertragung stammende Protokoll hat jedoch erhebliche Einschränkungen, was die Zahl der übertragbaren Komponenten und Stati betrifft. Dies ist insofern ein Problem, als für die Qualitätssicherung in der Zentrale nicht nur die Messwerte für die zu messenden Schadstoffe bzw. meteorologischen Parameter benötigt werden, sondern darüber hinaus auch die Werte der internen Sensoren des Messgeräts, die z.B. Druck, Durchfluss und Temperatur der Baugruppen eines Messgeräts überwachen.

Mit **UIDEP** soll ein moderner Standard geschaffen werden, der in Bezug auf die übertragbaren Komponenten und Stati keine derartigen Beschränkungen einbringt und auf eine aktuelle technische Basis – HTTP, JSON, REST aufsetzt, um eine einfache Implementierbarkeit zu ermöglichen. Dieses Protokoll deckt die Kommunikation zwischen Messgerät und Messstellenrechner (oder Messgerät und Messnetzzentrale) auch andere Formen der Kommunikation ab, insbesondere den Datenaustausch zwischen Messnetzzentrale und interessierten Empfängern.

**UIDEP** bezeichnet somit gleichermaßen:

- eine JSON-Struktur für die Darstellung von Messwerten und anderen Informationen, die in einem Messnetz übertragen werden
- die Übertragung dieser JSON-Struktur per HTTP als Netzwerkprotokoll (zusätzlich gibt es auch einen UDP-Broadcast für die automatische Erkennung von Geräten im LAN)
- diese JSON-Struktur als Dateiformat für die Datenübertragung per Datei (z.B. von Messnetz zu Messnetz)

## 1. Funktionsübersicht

Um das altbewährte Bayern-Hessen-Protokoll vollständig ablösen zu können, ist es klarerweise erforderlich, alle darin beschriebenen Funktionen auch in UIDEP abbilden zu können. Darüberhinaus wurde UIDEP um einige neue Funktionen erweitert, insbesondere für die automatische Erkennung von UIDEP-fähigen Geräten im LAN.

Funktionen für die Kommunikation zwischen Messgerät und Messstellenrechner:

- Abfrage der aktuellen Messwerte und Stati
- Änderung des Betriebsmodus des Messgeräts (insbesondere zur Steuerung des Kalibrierablaufs)
- Setzen von Konfigurationseinstellungen des Messgeräts
- Synchronisation der Uhrzeit
- Erkennung von UIDEP-fähigen Messgeräten im LAN der Messstation
- Abfrage der Fähigkeiten eines Geräts

Funktionen für die Kommunikation zwischen Messstellenrechner (MSR) und Messnetzzentrale:

- Abfrage der Messwerte und aggregierten Werte (z.B. Halbstundenmittelwerte)
- Abfrage der Kalibrierdaten (Ergebnisse der automatischen Funktionskontrolle)
- Steuerung der Messgeräte über den MSR
- Konfiguration der Messgeräte über den MSR
- Konfiguration des Messstellenrechners
- Übertragung von Dateien vom MSR zur Zentrale
- Senden von Eigenmeldungen vom MSR zur Zentralen
- Synchronisation der Uhrzeit

Funktionen für die Kommunikation zwischen Messnetzzentrale und beliebigen Datenempfängern (z.B. andere Messnetzzentralen, Internetdienste usw.)

- Abfrage der aggregierten Werte

## 2. Protokollstruktur

Die Struktur der übertragenen Daten ist in mehreren Hierarchieebenen aufgebaut; abhängig von Datenquelle und -Ziel werden die nicht benötigten Hierarchieebenen weggelassen.

Die Hierarchieebenen laufen:

- Organisation (Messnetzbetreiber)
- Station
- Messgerät
- Sensor/Komponente/Kanal
- Wert

Beispiel für JSON-Daten<sup>1</sup>, die ein MSR von einem Messgerät erhält:

```
{
  "Device": "Horiba APNA/370",
  "SN": "12345678",
  "Components": [
    {
      "ID": "178",
      "Value": 6.543
    },
    {
      "ID": "179",
      "Value": 2.345,
      "ErrSts": ["F"]
    }
  ]
}
```

---

<sup>1</sup>Leerzeichen und Zeilenumbrüche sind im JSON-Format optional, hier zwecks Lesbarkeit eingefügt

Wenn der MSR später die Daten an die Zentrale schickt, kommt einerseits eine weitere Hierarchieebene hinzu, andererseits werden die bekannten Ebenen mit zusätzlichen Feldern ergänzt:

```
{
  "Station": "AIP-Teststation",
  "Time": "2015-05-19T11:30:00+01:00",
  "AvgTime": 1800,
  "Devices": [
    {
      "Device": "Horiba APNA/370",
      "SN": "12345678",
      "Components": [
        {
          "Component": "NO2",
          "ID": "178",
          "Unit": "ppb",
          "Value": 6.543,
          "MinValue": 0.112,
          "MaxValue": 11.778,
          "StdDev": 1.44,
          "Valid": true
        },
        {
          "Component": "NO",
          "ID": "179",
          "Unit": "ppb",
          "Value": 2.345,
          "MinValue": 0.012,
          "MaxValue": 4.778,
          "StdDev": 0.74,
          "Valid": true
        }
      ]
    },
    {
      "Device": "Horiba APOA/370",
      "SN": "12345679",
      "Components": [
        {
          "Component": "O3",
          "ID": "211",
          "Unit": "ppb",
          "Value": 16.543,
          "MinValue": 3.112,
          "MaxValue": 31.778,
          "StdDev": 2.44,
          "OpSts": ["N"],
          "Valid": true
        }
      ]
    }
  ]
}
```

Folgende Regeln gelten:

- I. Jede Quelle liefert nur die Felder, deren korrekten Inhalt sie kennt; alle anderen Felder werden ganz weggelassen. z.B. liefert ein Messgerät ohne eingebaute Uhr keine Information über den Messzeitpunkt.
- II. Felder mit leerem Inhalt dürfen und sollen weggelassen werden
- III. Die Reihenfolge der Felder innerhalb eines Objekts spielt keine Rolle, es ist keine bestimmte Reihenfolge vorgeschrieben und jede mögliche Reihenfolge ist gleich zu interpretieren.
- IV. Jedes Feld ist logisch genau einer Hierarchie-Ebene zugeordnet.
- V. Felder von Objekten niedriger Hierarchie-Ebenen dürfen in der Darstellung in höhere Hierarchie-Ebenen hinaufgezogen werden, sie gelten dann für alle zugehörigen Objekte der untergeordneten Hierarchie-Ebenen. Beispielsweise ist der *Messzeitpunkt* auf der Ebene *Wert* angesiedelt, kann aber auch in einer höheren Ebene, z.B. „Station“ angegeben werden; das hat zur Folge, dass für alle *Werte* unterhalb der Station dieser *Messzeitpunkt* gilt. Damit werden unnötige Wiederholungen vermieden.
- VI. Wenn es in einer Hierarchie-Ebene nur genau ein Objekt gibt (z.B. nur eine Komponente in einem Gerät), so darf dieses in der JSON-Darstellung mit dem darüberliegenden Objekt verschmelzen (Verflachung)

## Beispiel für Regel I:

Falsch:

```
{
  "Device": "ACME Sensor XYZ",
  "SN": "12345678",
  "Components": [
    {
      "ID": "178",
      "Value": 6.543,
      "Time": ""
    }
  ]
}
```

Richtig:

```
{
  "Device": "Horiba APNA/370",
  "SN": "12345678",
  "Components": [
    {
      "ID": "178",
      "Value": 6.543
    }
  ]
}
```

## Beispiel für Regel II:

Darstellungsvariante 1 (zulässig aber nicht empfohlen):

```
{
  "Device": "ACME Sensor XYZ",
  "SN": "12345678",
  "Components": [
    {
      "ID": "178",
      "Value": 6.543,
      "ErrSts": [],
      "OpSts": []
    }
  ]
}
```

Darstellungsvariante 2 – Bedeutung identisch mit Darstellungsvariante 1:

```
{
  "Device": "Horiba APNA/370",
  "SN": "12345678",
  "Components": [
    {
      "ID": "178",
      "Value": 6.543
    }
  ]
}
```

## Beispiel für Regel III:

Darstellungsvariante 1:

```
{
  "Device": "ACME Sensor XYZ",
  "SN": "12345678",
  "Components": [
    {
      "ID": "178",
      "Value": 6.543,
      "ErrSts": ["A", "B"],
      "OpSts": ["M"]
    }
  ]
}
```

Darstellungsvariante 2 – Bedeutung identisch mit Darstellungsvariante 1:

```
{
  "SN": "12345678",
  "Components": [
    {
      "Value": 6.543,
      "OpSts": ["M"],
      "ErrSts": ["B", "A"],
      "ID": "178"
    }
  ],
  "Device": "ACME Sensor XYZ"
}
```

## Beispiel für Regel V:

Darstellungsvariante 1:

```
{
  "Device": "ACME Sensor XYZ",
  "SN": "12345678",
  "Components": [
    {
      "ID": "178",
      "Value": 6.543,
      "Time": "2015-05-19T11:30:00.000+01:00",
      "ErrSts": ["A", "B"],
      "OpSts": ["M"]
    },
    {
      "ID": "179",
      "Value": 1.345,
      "Time": "2015-05-19T11:30:00.000+01:00",
      "ErrSts": ["A", "B"],
      "OpSts": ["M"]
    }
  ]
}
```

Darstellungsvariante 2 – Bedeutung identisch mit Darstellungsvariante 1:

```
{
  "Device": "ACME Sensor XYZ",
  "SN": "12345678",
  "ErrSts": ["A", "B"],
  "Time": "2015-05-19T11:30:00.000+01:00",
  "OpSts": ["S"],
  "Components": [
    {
      "ID": "178",
      "Value": 6.543
    },
    {
      "ID": "179",
      "Value": 1.345
    }
  ]
}
```

## Beispiel für Regel VI:

Darstellungsvariante 1:

```
{
  "Device": "ACME Sensor XYZ",
  "SN": "12345678",
  "Components": [
    {
      "ID": "178",
      "Value": 6.543,
      "Time": "2015-05-19T11:30:00.000+01:00",
      "ErrSts": ["A", "B"],
      "OpSts": ["M"]
    }
  ]
}
```

Darstellungsvariante 2 – Bedeutung identisch mit Darstellungsvariante 1:

```
{
  "Device": "ACME Sensor XYZ",
  "SN": "12345678",
  "ID": "178",
  "Value": 6.543,
  "Time": "2015-05-19T11:30:00.000+01:00",
  "ErrSts": ["A", "B"],
  "OpSts": ["M"]
}
```

### 3. Datenübertragung

Die von den Geräten angebotenen Webdienste, mit deren Hilfe Daten nach UIDEP ausgetauscht werden können, sind REST-ful (REST = Representational State Transfer) konzipiert.

Das bedeutet:

- Die Dienste sind zustandslos, das Service muss sich keine Kommunikationsdaten über einzelne Anfragen hinweg merken
- Die angebotenen Dienste stehen in einer verzeichnisartigen Struktur zur Verfügung
- http-Requests GET, POST, PUT, DELETE werden verwendet, um Daten abzufragen, anzulegen, abzuändern bzw. zu löschen
- Vermittler irgendwo im Netzwerk (firewall, gateways, u.ä.) haben keinen die Schnittstelle verändernden Einfluß

#### **Abgrenzung: Authentifizierung Verschlüsselung und Rechtevergabe**

Diese Aspekte der Kommunikation sind kein verbindlicher Teil der Spezifikation. Falls derartige Mechanismen benötigt werden, wird folgendes empfohlen: HTTP-BASIC-Authentifizierung für die Übergabe von Benutzername und Passwort, alternativ dazu Übergabe von Benutzername und Passwort als HTTP-Parameter als Parameter in der URL. Für eine wirksame Verschlüsselung der Kommunikation kann UIDEP auch über HTTPS übertragen werden; Client- und Server-Zertifikate können dabei verwendet werden, um die Kommunikationspartner sicher zu authentifizieren. Eine allfällige Beschränkung der Zugriffsmöglichkeiten pro Benutzer bzw. Rolle ist Vereinbarungssache und nicht in der UIDEP-Protokollspezifikation geregelt.

Hinweis: die in den folgenden Beschreibungen angegebenen Pfade, z.B. „/values/simple“, sind relativ zum Basis-Pfad des UIDEP-Webservice des jeweiligen Geräts angegeben. Der Basis-Pfad kann mit dem UIDEP-Geräteerkennungsprotokoll ermittelt werden und wird im Feld DevURL geliefert.

#### **3.1. Von einem Messgerät bereitgestellte Webdienste (URL) für die Abfrage von Messwerten**

Erlaubte http-Methode: GET

Primäre Messdaten: /values/simple

Messdaten mit allen Zusatzinfos: /values/complex

Ohne weitere Parameter werden die Daten aller Komponenten des Messgeräts für die zuletzt durchgeführte Messung geliefert.

Optionale Parameter bei der Abfrage:

?component= <i>id</i>	Einschränkung auf Messdaten einer Komponente
?start= <i>JJJJ-MM-TT-hh-mm-ss</i>	historische Daten ab dem Zeitpunkt liefern (inkl.)
?end= <i>JJJJ-MM-TT-hh-mm-ss</i>	historische Daten bis zum Zeitpunkt liefern (inkl.)
?avgtime= <i>n</i>	Messdaten mit angegebenem Mittelungsintervall liefern
?valid=true	Nur gültige Messdaten anfragen

Beispiel:

```
{
  "Device": "Horiba APNA/370",
  "SN": "12345678",
  "Components": [
    {
      "ID": "178",
      "Value": 6.543
    }
  ]
}
```

## Hinweise:

- /values/simple dient dazu, die Messwerte möglichst kompakt zu liefern, während /values/complex möglichst alle verfügbaren Zusatzinfos mitliefert
- es wird empfohlen, dass /values/simple nur die primären Messparameter liefert (Schadstoffe, meteorologische Parameter etc.) während /values/complex auch die Werte der internen Sensoren liefert (Durchfluss, Druck, Lampenspannung etc.)
- Betriebs- und Fehlerstati, die für die korrekte Interpretation der Messwerte unumgänglich sind, sollen auch bei /values/simple mitgeliefert werden
- "ID" ist eine eindeutige, nicht zwingend beschreibende Kennung pro Komponente. Es ist zulässig, darüber hinaus mit "Component" eine Beschreibung der Komponente zu liefern (z.B. chemische Formel des Schadstoffs). Viele Geräte, die das Bayern-Hessen-Protokoll unterstützen, erlauben eine freie Vergabe einer dreistelligen Zahl pro Komponente bzw. für die erste Komponente. Diese Zahl (als String dargestellt) ist ein möglicher Wert für "ID"; jeder andere String ist aber ebenso zulässig. Es wird empfohlen, dass der Benutzer die "ID" der von /values/simple gelieferten primären Messwerte frei konfigurieren können (ggf. als Zahlenbereich analog zu Bayern-Hessen); die zusätzlichen Kenngrößen, die von /values/complex geliefert werden, können eine voreingestellte, eindeutige, beschreibende "ID" aufweisen (z.B. „Flow4“ für den vierten Durchflusskennwert).

## Empfehlungen für Betriebs- und Fehlerstati:

Im normalen Messbetrieb sollen die Messwerte keine Betriebs- und Fehlerstati aufweisen. (Das heißt, ein Status mit der Bedeutung „Gerät ist eingeschaltet und arbeitet normal“ wird nicht empfohlen)

Stati können mehr als ein Zeichen lang sein; es wird empfohlen, dass sie nur aus Zeichen aus dem ASCII-Range 33-126 bestehen (bevorzugt Großbuchstaben, Ziffern, Kleinbuchstaben).

Folgende Vorgaben für häufig vorkommende Betriebsstati werden empfohlen:

- "W": Das Gerät ist im Wartungsmodus (üblicherweise ein vom Techniker aktivierter Schalter)
- "S": Das Gerät ist in der automatischen Funktionskontrolle, Kalibriergas liegt an
- "Z": Das Gerät ist in der automatischen Funktionskontrolle, Nullgas liegt an
- "R": Das Gerät arbeitet in einem erweiterten Messbereich (für den ersten bzw. einzigen Messbereich wird empfohlen, keinen Betriebsstatus zu liefern)

Folgende Vorgaben für häufig vorkommende Fehlerstati werden empfohlen:

- "T": ein Temperatursensor des Geräts meldet einen unzulässigen Wert
- "P": ein Drucksensor des Geräts meldet einen unzulässigen Wert
- "F": ein Durchflusssensor des Geräts meldet einen unzulässigen Wert
- "E": ein Problem mit der Stromversorgung wurde erkannt
- "L": ein Problem mit der Lampe wurde erkannt (für Geräte mit diesem Messprinzip)

Für Betriebs- und Fehlerstati, die im Gerät an mehreren Stellen auftreten können, wird empfohlen, eine Zahl hinter den oben genannten Buchstaben zu setzen, um Eindeutigkeit herzustellen.

Beispiele:

- "R3": „Range 3“ - Das Gerät arbeitet im dritten Messbereich (Betriebsstatus)
- "S2": „Span 2“ - Das Gerät ist in der automatischen Funktionskontrolle, das Kalibriergas für den zweiten Prüfpunkt liegt an (Betriebsstatus)
- "T4": „Temperature 4“ - der 4. Temperatursensor des Geräts meldet einen unzulässigen Wert

### 3.2. Von einem Messgerät bereitgestellte Webdienste (URL) für die Steuerung des Betriebsmodus (z.B. Einleiten einer Funktionskontrolle)

Erlaubte http-Methoden: GET (Betriebsmodus abfragen), PUT (Betriebsmodus ändern)  
Betriebsmodus lesen/schreiben: /mode

Hinweis: Betriebsmodi sind konzeptionell der Hierarchieebene *Komponente* zugeordnet, es ist aber zulässig, den Betriebsmodus gemäss Regel V. auf der Ebene *Messgerät* darzustellen, wenn der Modus alle Komponenten des Messgeräts gleichermaßen betrifft. (Bis UIDEP V2.1 war der Betriebsmodus konzeptionell der Hierarchieebene *Messgerät* zugeordnet)

Beispiel:

```
{
  "OperatingMode": "M"
}
```

Beispiel für ein Gerät, das einen eigenen Betriebsmodus pro Komponente erlaubt (z.B. Kalibratoren):

```
{
  "Components": [
    {"ID": "178", "OperatingMode": "Z"},
    {"ID": "179", "OperatingMode": "S"}
  ]
}
```

Bei der PUT-Methode kann die Liste auch nur einen Teil der Komponenten des Geräts umfassen; die Betriebsmodi von Komponenten, die nicht in der Liste erscheinen, bleiben dann unverändert.

Es gelten die folgenden Empfehlungen für häufig vorkommende Betriebsmodi. Generell wird empfohlen, dass Betriebsmodi und allfällig durch sie verursachten Betriebsstati bei den Messwerten zusammenpassende Kennungen aufweisen.

- "M": „Measuring“ - normaler Messbetrieb
- "S": „Span“ - automatische Funktionskontrolle, Kalibriergas wird eingeleitet
- "Z": „Zero“ - automatische Funktionskontrolle, Nullgas wird eingeleitet
- "G": „Gasphasentitration“ - Kalibrierung, Kalibriergas wird durch GPT hergestellt

Analog zu den Betriebs- und Fehlerstati wird auch hier empfohlen, eine Zahl hinter das Zeichen zu setzen, wenn es mehrere Ausprägungen eines Betriebsmodus gibt, z.B.

- "S1": Das Kalibriergas für den ersten Prüfpunkt wird eingeleitet
- "S2": Das Kalibriergas für den zweiten Prüfpunkt wird eingeleitet
- "R1": Erster Messbereich aktiv
- "R2": Zweiter Messbereich aktiv

### 3.3. Von einem Messgerät bereitgestellte Webdienste (URL) für die Konfigurierung

Erlaubte http-Methoden: GET (Konfiguration abfragen), PUT (Konfiguration ändern)  
Konfiguration lesen/schreiben: /configuration

Beispiel:

```
{
  "DevConfiguration": [
    {
      "SettingName": "foo",
      "DataType": "string",
      "MaxLength": 10,
      "CurrentSetting": "bar"
    },
    {
      "SettingName": "xyz",
      "DataType": "integer",
      "MinSetting": 0,
      "MaxSetting": 100,
      "CurrentSetting": 50
    }
  ]
}
```

Folgende Ausprägungen für "DataType" sind empfohlen:

- "string" - Zeichenkette; die maximal erlaubte Länge wird in "MaxLength" angegeben
- "integer" - Ganzzahl; der erlaubte Bereich wird in "MinSetting" und "MaxSetting" angegeben
- "float" - Gleitkommazahl; der erlaubte Bereich wird in "MinSetting" und "MaxSetting" angegeben
- "boolean" - logischer Wert; kann die Werte *true* und *false* annehmen

Konfigurationsdaten mit einem anderen als den in der Liste angeführten "DataType" sind nur mit einem herstellerspezifischen Konfigurationsprogramm bearbeitbar.

Für Geräte, bei denen keine Konfigurationsmöglichkeiten über UIDEP vorgesehen sind, wird für GET folgende Antwort empfohlen (Regel II von Seite 7 wird in diesem Fall bewusst ignoriert):

```
{
  "DevConfiguration": []
}
```

Beim Ändern der Konfiguration mit PUT sind nur jene Einstellungen mitzugeben, die auch tatsächlich zu ändern sind; die Felder "DataType", "MaxLength", "MinSetting" und "MaxSetting" können mitgegeben oder weggelassen werden; das Gerät soll und darf diese Felder ignorieren.

### 3.4. Von einem MSR bereitgestellte Webdienste (URL ) für die Abfrage von Messwerten

Erlaubte http-Methoden: GET

Primäre Messdaten von allen Geräten: /values/simple

Messdaten mit allen Zusatzinfos: /values/complex

Kalibrierwerte: /functioncheck

Ohne weitere Parameter werden die Daten aller angeschlossener Messgeräte für die zuletzt durchgeführte Messung mit dem primären Mittelungsintervall (i.d.R. HMW) geliefert.

Optionale Parameter bei der Abfrage:

?device=seriennummer	Einschränkung auf Messdaten eines Geräts
?start=JJJJ-MM-TT-hh-mm-ss	Messdaten ab dem angegebenen Zeitpunkt liefern (inkl.)
?end=JJJJ-MM-TT-hh-mm-ss	Messdaten bis zum angegebenen Zeitpunkt liefern (inkl.)
?avgtime=n	Messdaten mit angegebenem Mittelungsintervall liefern

Beispiel:

```
{
  "Station": "AIP-Teststation",
  "Time": "2015-05-19T11:30:00.000+01:00",
  "AvgTime": 1800,
  "AmsType": "GEMI IOX",
  "Version": "3.02",
  "Devices": [
    {
      "Device": "Horiba APNA/370",
      "SN": "12345678",
      "Components": [
        {
          "Component": "NO2",
          "ID": "178",
          "Unit": "ppb",
          "Value": 6.543,
          "MinValue": 0.112,
          "MaxValue": 11.778,
          "StdDev": 1.44,
          "Valid": true,
          "ChkTime": "2015-05-19T11:30:00.000+01:00",
          "Refs": [
            {
              "RefNo": 0,
              "NominalValue": 0.0,
              "CheckValue": 0.2
            },
            {
              "RefNo": 1,
              "NominalValue": 120.0,
              "CheckValue": 118.85
            },
            {
              "RefNo": 2,
              "NominalValue": 200.0,
              "CheckValue": 201.7
            }
          ]
        }
      ]
    }
  ]
}
```

Die Erläuterungen von 3.1 gelten hier analog.

### 3.5. Von einem Messstellenrechner bereitgestellte Webdienste (URL) für die Konfiguration

Erlaubte http-Methoden: GET (Konfiguration abfragen), PUT (Konfiguration ändern)  
 Konfiguration lesen/schreiben: /configuration

Beispiel:

```
{
  "Station": "AIP-Teststation",
  "StationConfiguration": [
    {
      "SettingName": "foo",
      "DataType": "string",
      "MaxLength": 10,
      "CurrentSetting": "bar"
    },
    {
      "SettingName": "xyz",
      "DataType": "integer",
      "MinValue": 0,
      "MaxValue": 100,
      "CurrentSetting": 50
    }
  ],
  "Devices": [
    {
      "Device": "Horiba APNA/370",
      "SN": "12345678",
      "DevConfiguration": [
        {
          "SettingName": "bar",
          "DataType": "string",
          "MaxLength": 10,
          "CurrentSetting": "foo"
        },
        {
          "SettingName": "abc",
          "DataType": "integer",
          "MinValue": 0,
          "MaxValue": 60,
          "CurrentSetting": 20
        }
      ]
    }
  ]
}
```

Die Erläuterungen von 3.3 gelten hier analog.

### 3.6. Von einem Messstellenrechner bereitgestellte Webdienste (URL) für die Übertragung von Dokumenten

Erlaubte http-Methoden: GET (Liste abfragen, Dokument herunterladen), POST (Dokument hochladen)

Liste der Dokumente: /documents (nur GET zulässig)

Dokument herunterladen/hochladen: /documents/dateiname

Beispiel:

```
{
  "Documents": [
    "somedocument.pdf",
    "anotherdocument.zip"
  ]
}
```

### 3.7. Von der Messnetzzentrale bereitgestellte Webdienste (URL) für Eigenmeldungen

Erlaubte http-Methode: POST

Eigenmeldung schicken: /eventnotification

Beispiel:

```
{
  "Station": "AIP-Teststation",
  "Time": "2015-05-19T11:37:15.000+01:00",
  "EventType": "Invalid function check",
  "EventText": "Span gas bottle empty",
  "Device": "Horiba APNA/370",
  "SN": "12345678",
  "Components": [
    {
      "ID": "178"
    }
  ]
}
```

Folgende vordefinierten Ereignistypen ("EventType") werden empfohlen:

- "Operating Status" → Übermittlung von Betriebs- und/oder Fehlerstati
- "Limit exceedance" → Meldung aufgrund einer Grenzwertüberschreitung
- "Device change" → Meldung aufgrund eines Messgerätetausches
- "Invalid function check" → Meldung wegen einer fehlgeschlagenen Funktionskontrolle
- "Restart" → Meldung wegen eines Neustarts des Geräts (MG oder MSR)
- "System" → sonstige vom System ausgelöste Meldungen
- "Manual" → Vom Benutzer vor Ort ausgelöste Meldung
- "Document transfer" → Es sind Dokumente von der Station in die Zentrale zu übertragen

### 3.8. Abfrage der Fähigkeiten eines Geräts

Erlaubte http-Methoden: GET

Liste der Fähigkeiten: /capabilities

Beispiel:

```
{
  "DevCapabilities": {
    "Descriptors": [
      "DevURL", "DevCapabilities", "Descriptors", "ChkTime",
      "CheckValue", "Component", "Components", "CurrentSetting",
      "DataType", "DevConfiguration", "Device", "ErrSts",
      "EventText", "EventType", "HistoricalData",
      "ID", "IntSts", "LastCalibration", "MaxSetting", "MeasuredValues",
      "MinSetting", "NominalValue", "NotificationURL", "OperatingModes",
      "OperatingMode", "OpSts", "RefNo", "Refs", "Services",
      "SN", "StoragePeriod", "Time", "Unit", "Valid", "Value",
      "Version"
    ],
    "Services": [
      "values/simple", "values/complex", "configuration",
      "capabilities"
    ],
    "UIDEPVersion": "2.1"
  },
  "DevConfiguration": [
    {
      "SettingName": "SomeSetting",
      "DataType": "integer",
      "MinSetting": 0,
      "MaxSetting": 100,
      "CurrentSetting": 42
    },
    {
      "SettingName": "AnotherSetting",
      "DataType": "string",
      "MaxLength": 50,
      "CurrentSetting": "FooBar"
    }
  ],
  "LastCalibration": "2015-05-19T11:37:15.000+01:00",
  "Features": {
    "TimeSynchronized": true,
    "HistoricalData": true,
    "Aggregations": [
      {
        "AvgTime": 5,
        "StoragePeriod": 86400
      },
      {
        "AvgTime": 300,
        "StoragePeriod": 864000
      },
      {
        "AvgTime": 1800,
        "StoragePeriod": 86400000
      }
    ]
  },
  "OperatingModes": [
    {
      "OperatingMode": "M",

```

```

    "Description": "Measuring"
  },
  {
    "OperatingMode": "Z",
    "Description": "Zero Function Check"
  },
  {
    "OperatingMode": "S",
    "Description": "Span Function Check"
  },
  {
    "OperatingMode": "P",
    "Description": "Power saving mode (Standby)"
  }
],
"OperatingStatuses": [
  {
    "Status": "Z",
    "Description": "Zero function check active"
  },
  {
    "Status": "S",
    "Description": "Span function check active"
  },
  {
    "Status": "U",
    "Description": "Device in startup procedure"
  },
  {
    "Status": "D",
    "Description": "Device in shutdown procedure"
  }
]
"ErrorStatuses": [
  {
    "Status": "T",
    "Description": "Operating temperature not in acceptable range"
  },
  {
    "Status": "N",
    "Description": "No response from sensor unit"
  }
]
}

```

Die Angabe von "Descriptors" ist optional (nur noch aus Kompatibilitätsgründen mit früheren Versionen vorhanden); empfohlen wird, sie wegzulassen. Empfänger von UIDEP-Daten müssen das Vorhandensein von "Descriptors" tolerieren, den Inhalt aber nicht prüfen. Wenn "Descriptors" angegeben ist, sind alle Descriptoren anzugeben, die von diesem Gerät in UIDEP geliefert werden können.

### 3.9. Abfrage der Zeit eines Geräts

Erlaubte http-Methoden: GET

Aktuelle Uhrzeit: /currenttime

Beispiel:

```
{  
  "CurrentTime": "2015-11-18T10:22:05.000+01:00",  
  "TimeSynchronized": true  
}
```

Mit dieser Methode kann man überprüfen, ob die eigene Uhr mit der des Kommunikationspartners ausreichend übereinstimmt.

Als Reaktion auf ermittelte Differenzen kann man die von diesem Gerät gelieferten Messwerte

- so übernehmen, wie sie kommen (allfällige Zeitdifferenz wird ignoriert)
- zeitversetzt übernehmen (die Zeitdifferenz wird kompensiert)
- verwerfen (Messwerte werden wegen der Zeitdifferenz als unbrauchbar angesehen)

Dies betrifft nur den Fall, dass man sich historische Daten liefern lässt. Bei der Abfrage der aktuellen Messwerte kann man die Zeitangabe des Kommunikationspartners ignorieren und die eigene Uhr als Referenzzeitmesser verwenden.

## 4. Liste der Deskriptoren

Deskriptor	Typ	Beschreibung	Ebene
Active	Boolean	Aktiv-Kennzeichen	Gerät
AmstType	String	MSR-Typ	Station
AvgTime	Integer	Mittelungsintervall in Sekunden	Wert
ChkTime	String (ISO-Zeit)	Zeitpunkt einer Funktionskontrolle	FK
CheckValue	Gleitkomma	Istwert der Funktionskontrolle**	FK-Ref
Component	String	Komponente (Parameter) - Bezeichnung	Komp.
Components	Array	Liste der Komponenten	Gerät
Configurati on	Object	Konfigurationsdaten (gerätespezifisch)	Org., Station, Gerät, Komp.
CurrentSet ting	variabel	Aktueller Wert einer Konfigurationseinstellung	Konfig.
CurrentTim e	String (ISO-Zeit)	Aktuelle Zeit laut interner Uhr des Geräts	Gerät
DataType	String	Datentyp einer Konfigurationseinstellung	Konfig.
Descriptio n	String	Beschreibung eines Status oder Modus	Modus, Status
Descriptor s	Array	Liste der vom Gerät unterstützten Standard-Deskriptoren	Gerät
Device	String	Gerätetyp	Gerät
Devices	Array	Liste der Daten pro Gerät	Station
DevURL	String	URL, unter der die UIDEP-Dienste erreicht werden können	Gerät
Documents	Array<String>	Liste der verfügbaren Dokumente	Station
ErrorStatu ses	Array	Liste der möglichen Fehlerstati	Gerät
ErrSts	Array<String>	Liste der angefallenen Fehlerstati	Wert
EventText	String	Ereignistext einer Eigenmeldung	EM
EventType	String	Typ einer Eigenmeldung	EM
Features	Object	Angabe der unterstützten Features	Gerät
FunctionCh ecks	Array	Liste der Funktionskontrollen (Kalibrierungen)	Komp.
ID	String	Komponenten-ID (eindeutig, muss nicht beschreibend sein)	Komp.
IntSts	Array<String>	Liste der angefallenen internen Stati	Wert
LastCalibr ation	String (ISO-Zeit)	Datum der letzten „richtigen“ Kalibrierung	Gerät
MaxLength	Integer	Maximale Länge einer Konfig-Einstellung vom Typ String	Konf.
MaxSetting	Gleitkomma	Maximal erlaubter Wert einer Einstellung	Konf.
MaxValue	variabel	Größter Messwert im Mittelungsintervall	Wert
MeasuredVa lues	Array	Liste der Messwerte	Komp.
MinSetting	Gleitkomma	Minimal erlaubter Wert einer Einstellung	Konf.

MinValue	variabel	Kleinster Messwert im Mittelungsintervall	Wert
NominalValue	Gleitkomma	Sollwert der Funktionskontrolle**	FK-Ref
NotificationURL	String	URL der Zentrale (für Eigenmeldungen)	Org.
OperatingMode	String	Betriebsmodus („M“=Messen, „Z“=Nullen, „S“=Span, ...)	Komp.
OperatingModes	Array	Liste der möglichen Betriebsmodi	Gerät
OperatingStatues	Array	Liste der möglichen Betriebsstati	Gerät
OpSts	Array<String>	Liste der angefallenen Betriebsstati	Wert
Organisation	String	Name der Organisation	Org.
RefNo	Integer	Nummer des Referenzpunktes (0=Zero, 1=Span, ...)**	FK-Ref
Refs	Array	Liste der Referenzpunkte einer Funktionskontrolle	FK
Services	Array	Liste der unterstützten Dienste (HTTP-Aufrufe)	Gerät, Station
SN	String	Seriennummer	Gerät
Station	String	Stationsname (Schlüsselbegriff)	Station
StationURL	String	URL, unter der die UIDEP-Dienste erreicht werden können	Station
Status	String	Statuscode für Selbstbeschreibung	Status
StdDev	Gleitkomma	Standardabweichung im Mittelungsintervall	Wert
StoragePeriod	Integer	Zeitraum, in dem das Gerät zurückliegende Messwerte speichert, in Sekunden	Aggr.
Time	String (ISO-Zeit)	Zeitpunkt	Wert
TimeSynchronized	Boolean	True, wenn das Gerät über eine mit offizieller Zeit synchronisierte Echtzeituhr enthält (per NTP, Funkuhr o.ä.)	Gerät
UIDEPVersion	String	Unterstützte UIDEP-Version	Gerät
Unit	String	Einheit	Komp.
Valid	Boolean	Gültigkeitskennzeichen	Wert
Value	Gleitkomma	Messwert	Wert
Version	String	Software-Version	Station

\*\* ... bei mehreren FK-Vorgaben bzw in Messungen als „lack of fit“-Test bestimmt RefNo welcher Messpunkt betroffen ist

## 5. Geräteerkennungsprotokoll

Das Geräteerkennungsprotokoll dient dazu, im lokalen Netz befindliche Messgeräte, welche UIDEP unterstützen, automatisch erkennen und einbinden zu können.

Der Hauptzweck ist der, dass ein MSR automatisch die Messgeräte in der Station auffinden und konfigurieren kann. Es ist aber auch denkbar, dass sich ein Messgeräte auf diese Weise mit einem anderen Messgerät verbindet, um von diesem Messwerte zum Beispiel zur Kompensation bestimmter, aufgrund von Querempfindlichkeit sich ergebender Messfehler, zu erhalten.

### **Abgrenzung: Auffinden von MSR durch die MNZ, IPv6**

Für das unten beschriebene Verfahren ist es notwendig, dass sich Sender und Empfänger der UDP-Broadcast im gleichen Netzwerksegment befinden. Dies ist im Fall von MSR und MG regelmässig der Fall (stationsinternes Netz), aber normalerweise nicht im Fall von MNZ und MSR. Durch eine Umstellung auf IPv6 würde diese Limitierung obsolet werden; die dann einzusetzenden Multicasts (statt Broadcasts) könnten alle interessierten Geräte erreichen. Daher ist unten vorsorglich beschrieben, wie eine MSR-Antwort auf eine derartige Anfrage auszusehen hat.

Dies wird durch einen UDP-Broadcast auf Port 4120 realisiert, der folgenden Inhalt (ASCII) hat:

UIDEP-DEVICE-DISCOVERY

UIDEP-Geräte, die das Erkennungsprotokoll unterstützen, reagieren darauf ihrerseits mit einer UDP-Nachricht an den Absender des Broadcasts, ebenfalls Port 4120; in dieser Nachricht werden (UTF-8 codiert) im JSON-Format die wichtigsten Daten des Geräts bekannt gegeben.

Folgende Descriptoren können dabei zur Anwendung kommen:

Device, SN, DevURL, Version, Components, ID, Component

Ein MSR, der auf eine derartige Anfrage reagiert, kann darüber hinaus auch folgende Descriptoren liefern:

Station, StationURL, Amstype, Version, DeviceList

Beispiel für ein MG, das sich meldet:

```
{
  "Device": "Horiba APNA/370",
  "SN": "12345678",
  "DevURL": "http://192.168.11.15:8080/uidep",
  "Components": [
    {
      "ID": "178",
      "Component": "N02"
    },
    {
      "ID": "179",
      "Component": "N0"
    }
  ]
}
```

Beispiel für einen MSR, der sich meldet:

```
{
  "Station": "AIP-Teststation",
  "AmsType": "Horiba IOX",
  "Version": "3.02",
  "StationURL": "http://192.168.11.2/foobar/uidep",
  "Devices": [
    {
      "Device": "Horiba APNA/370",
      "SN": "12345678",
      "DevURL": "http://192.168.11.15:8080/uidep",
      "Components": [
        {
          "ID": "178",
          "Component": "NO2"
        },
        {
          "ID": "179",
          "Component": "NO"
        }
      ]
    }
  ]
}
```

Die Antwort muss gültiges JSON (UTF-8 codiert) sein; Leerzeichen und Zeilenumbrüche sind wie üblich im obigen Beispiel der Lesbarkeit wegen enthalten, aber nicht obligatorisch.

Bitte beachten: Port 4120 ist fix für diesen Erkennungsdienst festgelegt; im Gegensatz dazu können die in den vorigen Kapiteln beschriebenen Webdienste jeden beliebigen Port verwenden – es ist nur wichtig, dass dieser im **DevURL**-Feld der Antwort auf den Geräteerkennungsbroadcast angegeben wird. (Im obigen Beispiel wäre das Port 8080)

Aus technischen Gründen ist es notwendig, eine Obergrenze für die maximale Paketgröße festzulegen, daher darf das Antwort-Paket der UDP-Identifizierung nicht größer werden als 1000 Byte. Folgende Antwort-Varianten sind gleichermassen zulässig:

- Übertragung aller Komponenten
- Übertragung nur der Komponenten, die bei values/simple geliefert werden
- Weglassen von "Components" (z.B. bei einem Partikelklassifizierer, der über 100 primäre Messkomponenten hat und damit auch damit das Limit überschreiten würde)

Empfohlen wird jene Variante, bei der ein Maximum an Komponenten übertragen wird, ohne die maximale Paketgröße zu überschreiten.