

Installation Gaszähler Sensor GZ1 USB

Version: 1.2 vom 22.04.2023


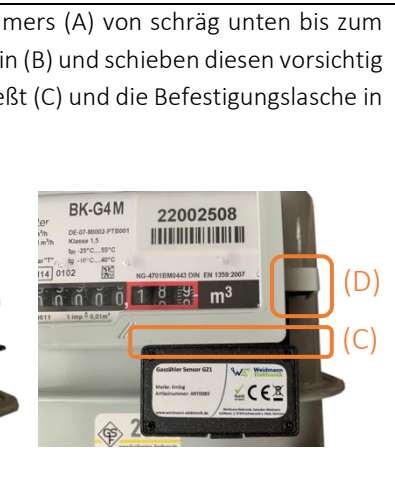

1. Installation Gaszähler Sensor

ACHTUNG:

Diese Installationsanleitung gilt ausschließlich für Elster/Honeywell Balgengaszähler des Typs BK-G2,5 bis BK-G100!

VORBEREITUNG:

Zur Installation des Gaszähler Sensor GZ1 USB an Ihren Gaszähler benötigen Sie lediglich einen Kreuzschraubendreher (empfohlene Größe PZ2)

Schritt 1:	Entfernen Sie die am Sensor GZ1 befestigte Schraube und Mutter (empfohlener Kreuzschraubendreher PZ2).	
Schritt 2:	Führen Sie den Sensor GZ1 mit der Spitze des Impulsaufnehmers (A) von schräg unten bis zum Anschlag in die Aussparung des Impulsegeber Ihres Gaszähler ein (B) und schieben diesen vorsichtig nach oben, bis der Sensor bündig mit der Aussparung abschließt (C) und die Befestigungslasche in die seitliche Befestigungslasche Ihres Zähler einrastet (D).	
Schritt 3:	Fixieren Sie den Sensor abschließend mit der zuvor entfernten Schraube und Mutter.	

<p>Schritt 4:</p>	<p>Verbinden Sie nun den Sensor GZ1 mit der Gas Box mittels des mitgelieferten Rj10 Kabels.</p>	 
<p>Schritt 5:</p>	<p>Verbinden Sie die Gas Box über USB mit Ihrem Raspberry Pi oder Emlog Gerät. Die Box startet nach einer kurzen Wartezeit.</p> <p>Optional: Sie können die Gas Box noch zusätzlich mit einem externen MicroUSB Netzteil mit Strom versorgen. Dazu können Sie jedes handelsübliche Handy Ladeteil verwenden.</p>	
<p>Schritt 6:</p>	<p>Konfigurieren Sie über die Taste 3.0.0 die Impulse pro m^3 (Standard 100 Imp/m^3). Die Zahl finden Sie auf Ihrem Zähler mit z.B.:</p> <p>1 Imp = 0.01m^3 (= 100 Imp/m^3) 1 Imp = 0.1m^3 (= 10 Imp/m^3)</p> <p>Drücken Sie mehrmals die 3.0.0 Taste, bis der gewünschte Wert im Display angezeigt wird. Warten Sie jetzt ca. 5 Sekunden. Der eingestellte Wert wird übernommen und gespeichert.</p>	

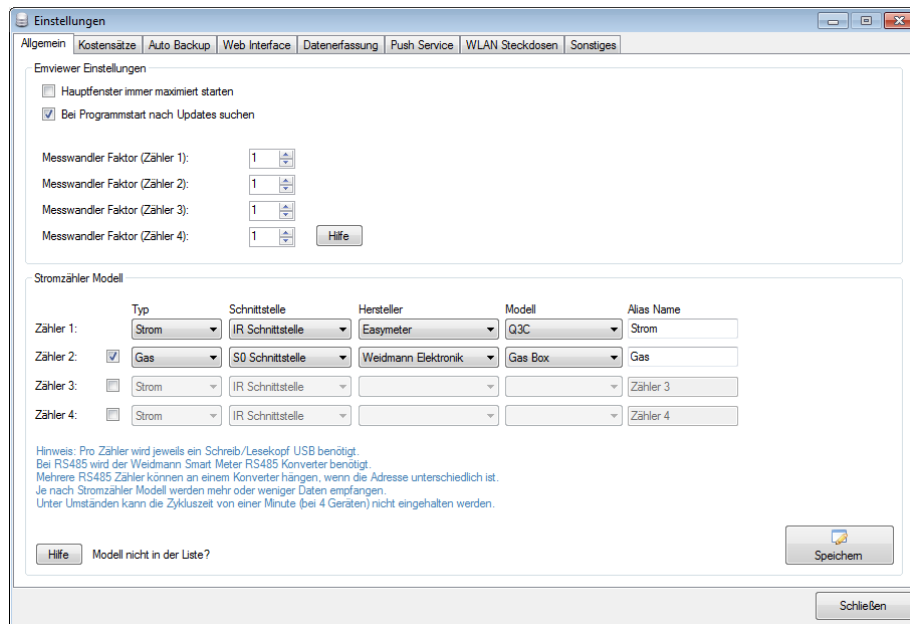
Schritt 7: Verbinden Sie nun die Gas Box mit Ihrem Emlog Datenlogger über USB und konfigurieren diesen in der Emviewer Software wie folgt:

Typ: Gas

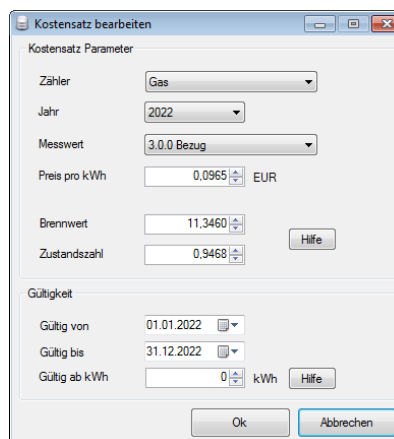
Schnittstelle: S0 Schnittstelle

Hersteller: Weidmann Elektronik

Modell: Gas Box



Erstellen Sie in der Emviewer Software einen neuen Kostensatz für den Gaszähler und Messwert 3.0.0. Hinterlegen Sie den Brennwert und die Zustandszahl Z.



Gas wird in kWh abgerechnet. Um die gezählten Kubikmeter in kWh umrechnen zu können, wird der Brennwert und die Zustandszahl benötigt. Sie finden beide Zahlen auf Ihrer letzten Gasrechnung oder können diese bei Ihrem Gas Versorger erfragen. Sollten auf Ihrer Gasrechnung mehrere Werte für den Brennwert vorhanden sein, können Sie einen Durchschnittswert errechnen. Oft stellt der Versorger bereits Durchschnittswerte für beide Zahlen auf deren Webseite zur Verfügung. Emlog unterstützt die Gas Box ab Version 3.35. Falls die Auswahl nicht vorhanden ist, updaten Sie die Emlog Software auf Version 3.35

Schritt 8: Weitere Informationen:

USB-Kabel

Falls das USB-Kabel der Gas Box von der Länge her nicht ausreichend sein sollte, können Sie das USB-Kabel mit einer aktiven Verlängerung auf bis zu 20m verlängern. USB 2.0 hat eine maximale Kabellänge von 5m. Bei einem aktiven USB-Kabel werden die Signale zusätzlich verstärkt. Dadurch ist eine Kabellänge >5m möglich.

Sie finden eine passende Verlängerung (5m, 10m, 15m und 20m) z.B. auf Amazon:

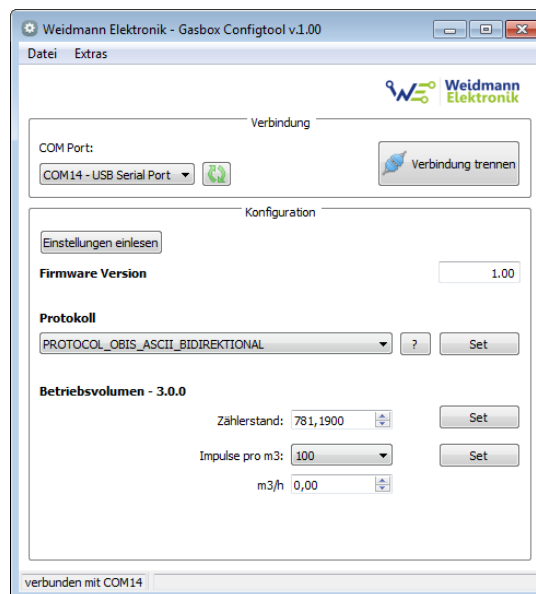
Hersteller: deleyCON USB 2.0 Verlängerungskabel Repeater-Kabel

[Amazon Link](#)

Eine Funk Variante der Gas Box zukünftig verfügbar sein, falls auch eine USB-Verlängerung nicht machbar ist.

Zählerstand


Die Gas Box kennt nicht Ihren Gaszähler Zählerstand. Aus diesem Grund beginnt die Gas Box bei 0.000m³ zu zählen. Im Download Bereich auf www.weidmann-elektronik.de finden Sie das „Gas Box Configtool“ für Windows. Verbinden Sie die Gas Box mit Ihrem Pc per USB und starten Sie das Configtool. Mit diesem Programm können Sie manuell den Gaszähler Zählerstand setzen. Danach ist der Zählerstand der Gas Box identisch mit Ihrem Gaszähler.



Wichtig: Sollte Sie bereits die Gas Box am Emlog Datenlogger angeschlossen haben, wirkt sich die Änderung des Zählerstandes auf den Emlog Tageswert aus. Ggf. müssen Sie dann in Emlog den Tageswert korrigieren. Eine Anleitung finden Sie im Download Bereich im Dokument „Emlog - Korrektur fehlerhafter Datensätze“.

Schritt 9: Bei der Verwendung von iobroker, konfigurieren Sie den smartmeter Adapter mit folgenden Einstellungen:

Instanzeinstellungen: smartmeter.0



Smartmeter Adapter Settings

General settings

Data request interval s

Data transfer ▼

Data protocol ▼

Language for datapoint names ▼

Data transfer settings

Serial device name ▼

Serial device baudrate baud

Serial device DataBits ▼

Serial device StopBits ▼

Serial device parity ▼

Serial-Response timeout s

Data protocol settings

D0: Number of WakeUp-Characters

D0: Device address

D0: SignOn-Message command

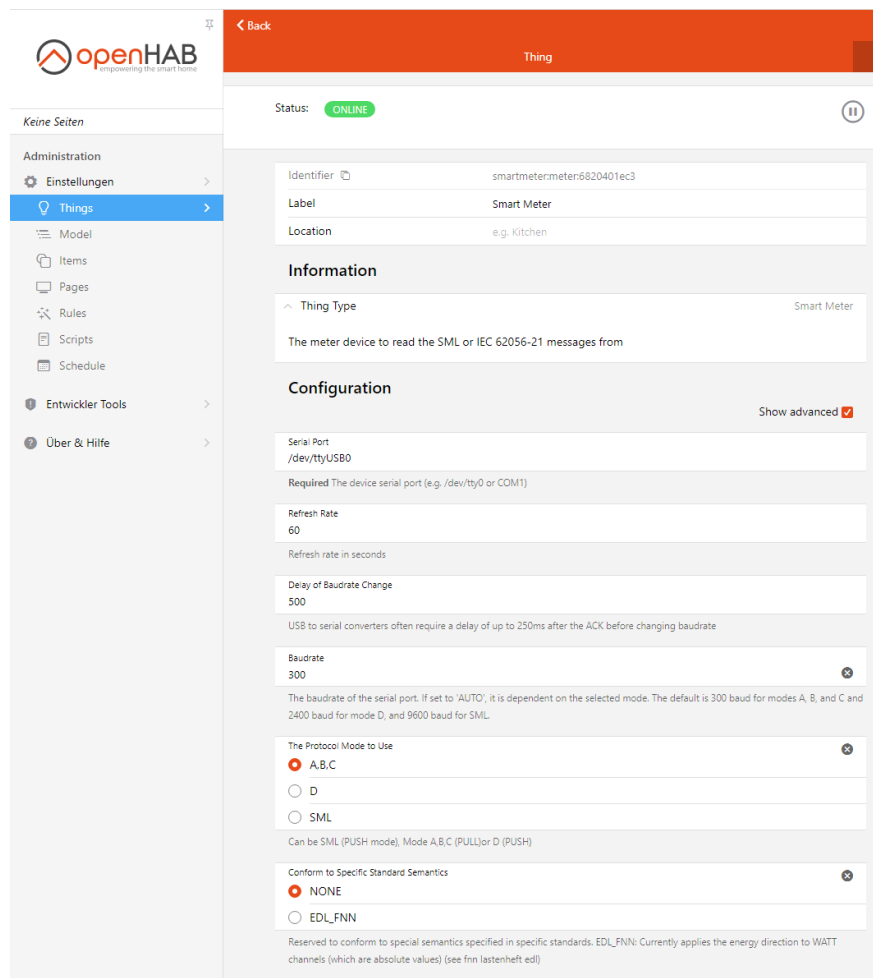
D0: Mode overwrite ▼

D0: Baudrate changeover overwrite baud

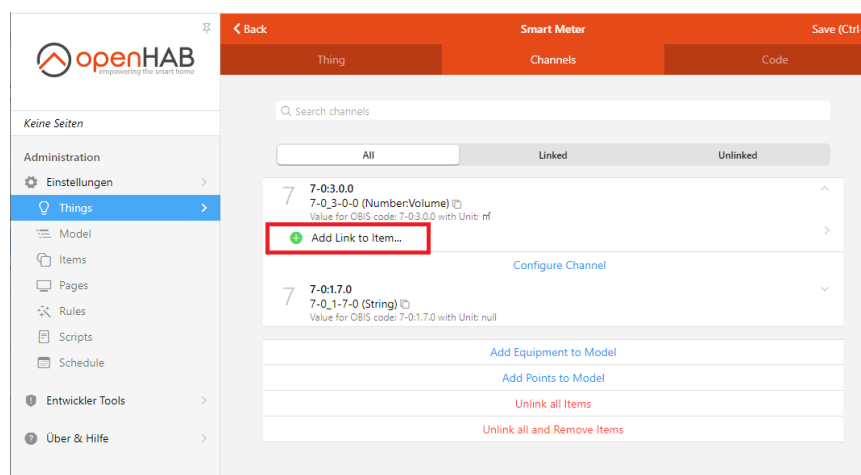
Delay between multiple SignOn-Messages ms

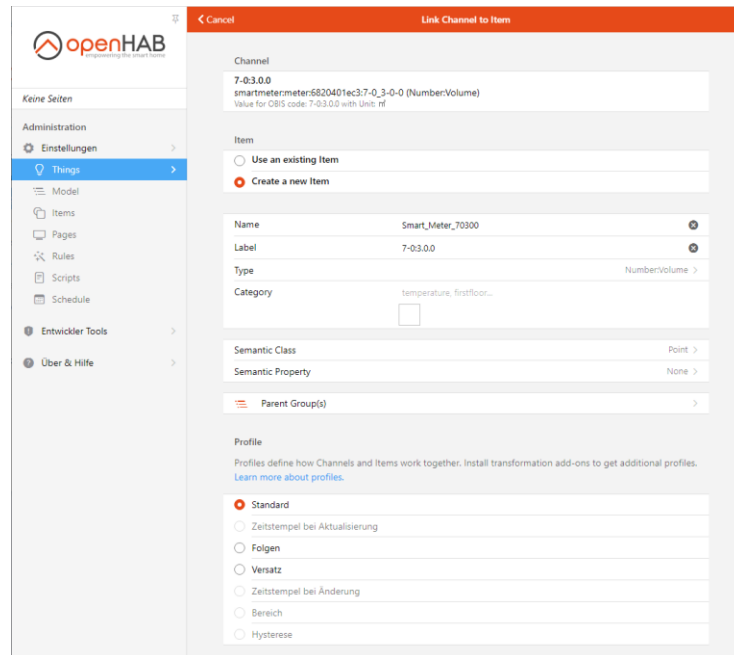
D0: Fallback OBIS-Medium ▼

Schritt 10: Bei der Verwendung von OpenHAB, konfigurieren Sie das smartmeter Plugin mit folgenden Einstellungen:

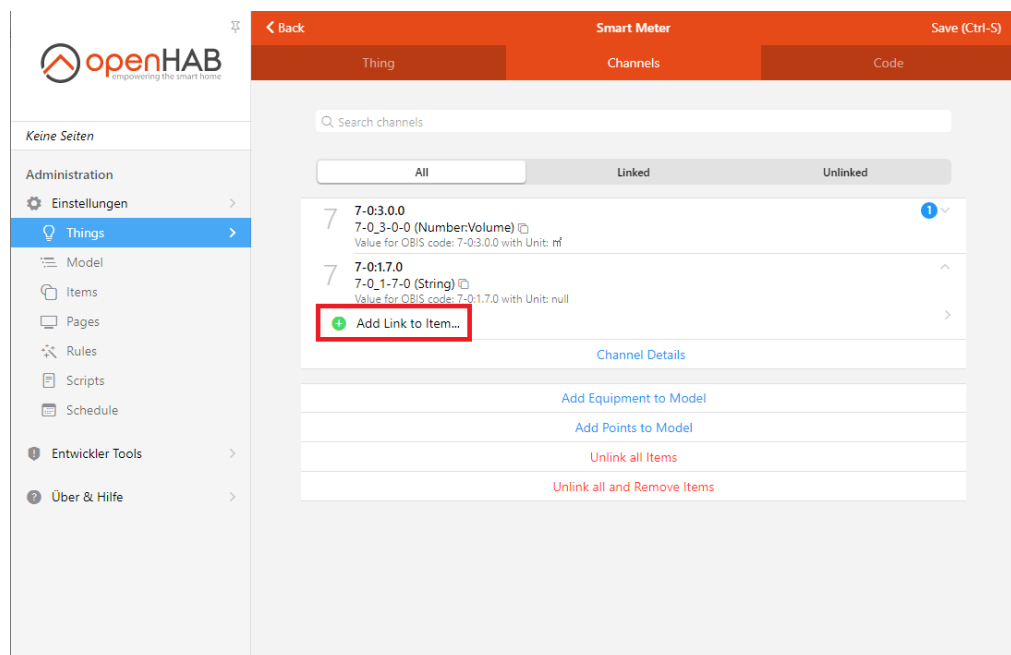


Unter Channels einen Link zu einem neuen Item erstellen für 3.0.0

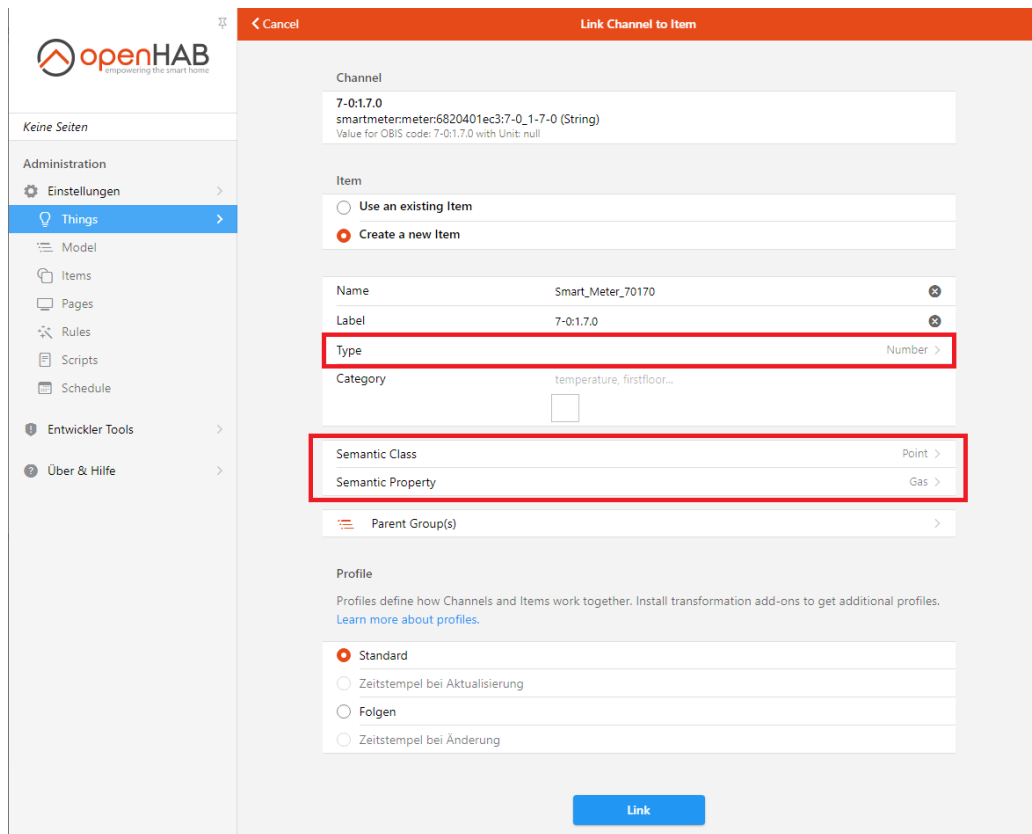




Unter Channels einen Link zu einem neuen Item erstellen für 1.7.0

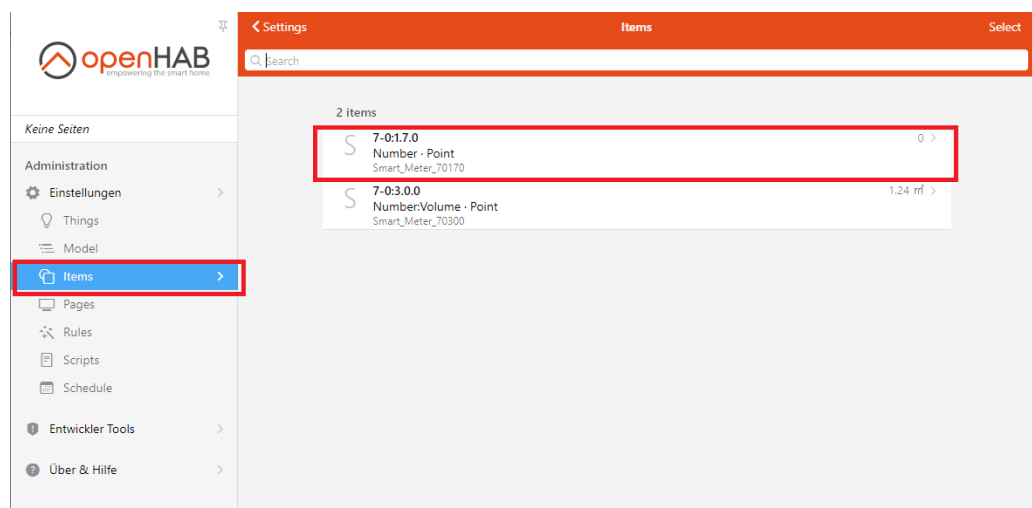


Type: Number setzen und Semantic Property: Gas



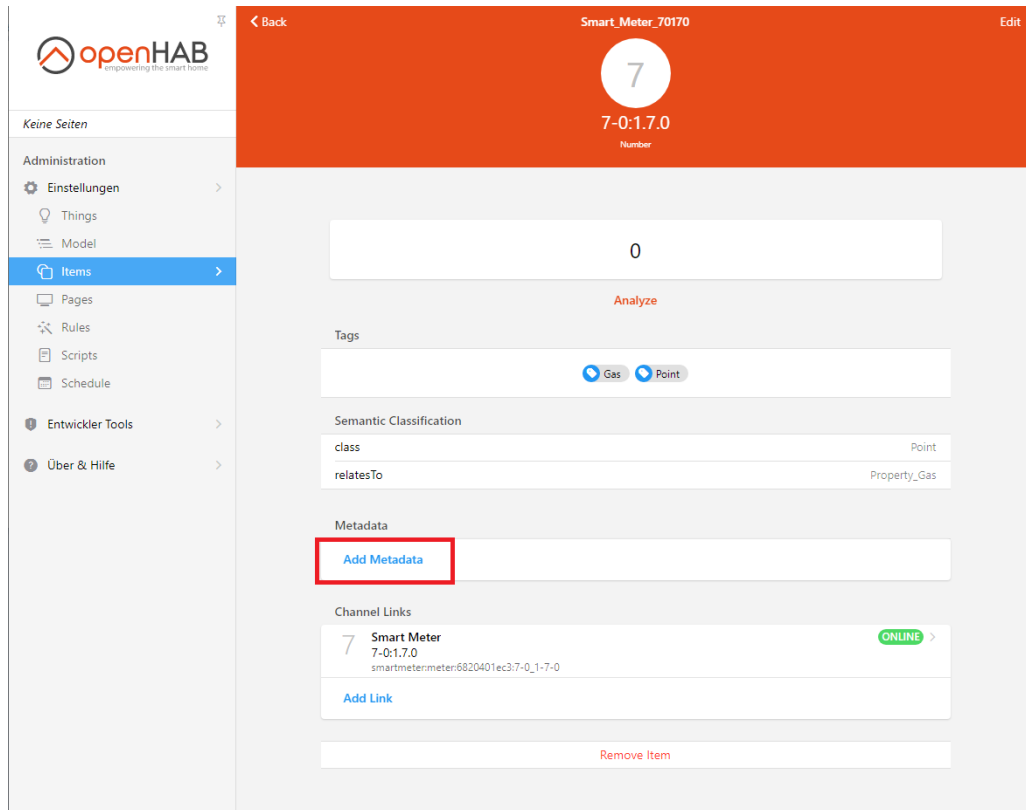
The screenshot shows the 'Link Channel to Item' configuration window in openHAB. The channel is '7-0:1.7.0 smartmeter:meter:6820401ec3:7-0_1-7-0 (String)'. The item is being created with the name 'Smart_Meter_70170' and label '7-0:1.7.0'. The 'Type' is set to 'Number' and the 'Semantic Property' is set to 'Gas'. The 'Semantic Class' is set to 'Point'. The 'Profile' is set to 'Standard'. A 'Link' button is at the bottom.

Wechsel zu Items und 7-0:1.7.0 auswählen.



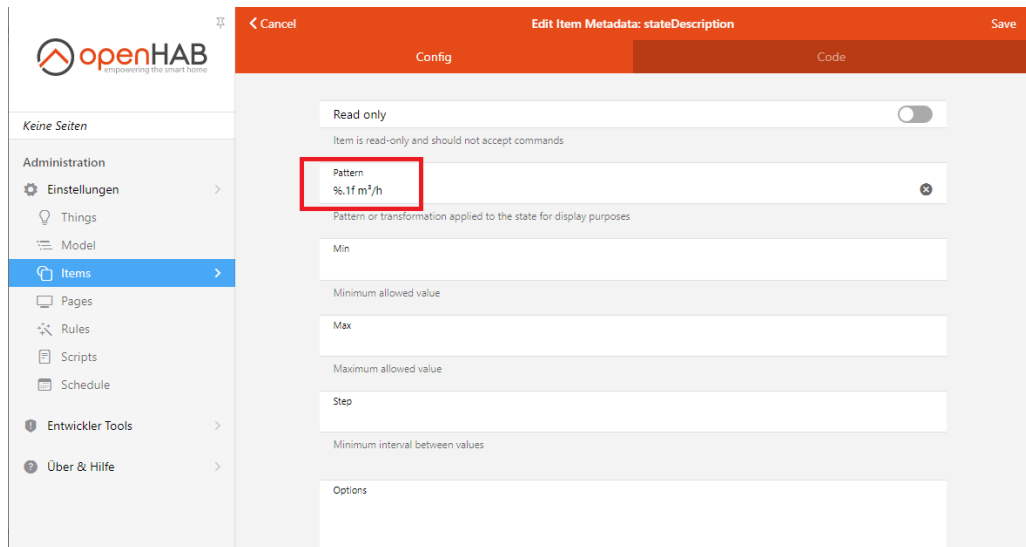
The screenshot shows the 'Items' configuration page in openHAB. The 'Items' menu item in the sidebar is highlighted. The main area shows a list of items with a search bar. The item '7-0:1.7.0 Number - Point' is selected and highlighted with a red box.

Add Metadata: state description



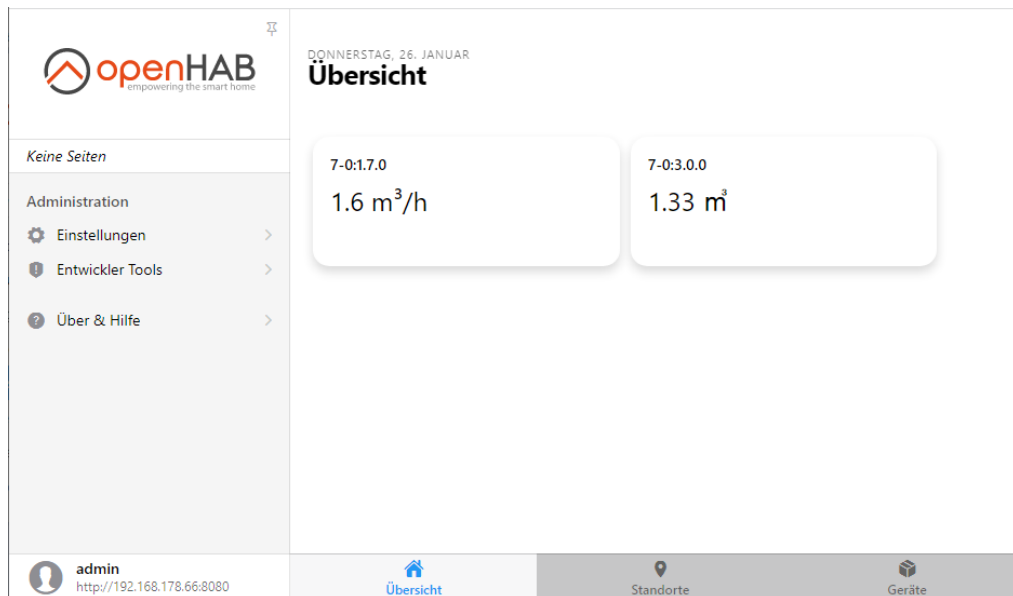
The screenshot shows the openHAB web interface for editing the state description of a smart meter. The left sidebar contains navigation options like 'Administration', 'Einstellungen', 'Items', 'Pages', 'Rules', 'Scripts', 'Schedule', 'Entwickler Tools', and 'Über & Hilfe'. The main content area is titled 'Smart_Meter_70170' and shows a current value of '7' with a unit of '7-0:1.7.0 Number'. Below this, there are sections for 'Tags' (Gas, Point), 'Semantic Classification' (class: Point, relatesTo: Property_Gas), and 'Metadata'. The 'Add Metadata' button is highlighted with a red box. At the bottom, there is a 'Channel Links' section showing the item's details and an 'Add Link' button.

Einheit setzen mit Pattern „%.1f m³/h“



The screenshot shows the 'Edit Item Metadata: stateDescription' configuration page in the openHAB web interface. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Edit Item Metadata: stateDescription' and has tabs for 'Config' and 'Code'. The 'Config' tab is active, showing a 'Read only' toggle (disabled) and a 'Pattern' field set to '%.1f m³/h', which is highlighted with a red box. Below the pattern field, there are fields for 'Min', 'Max', 'Step', and 'Options'.

Ergebnis bestaunen und dabei schimpfen, wie komplex openHAB ist.



Schritt 11: Bei der Verwendung von Volkszähler (vzlogger), können Sie die Gas Box integrieren. Die Werte kommen in vzlogger an. Die weitere Verarbeitung kenne ich nicht im Detail. Fragen Sie hier bitte Herrn Volkszähler für weitere Details. Ich bin absolut kein Fan der Volkszähler Software.

```
[Apr 22 21:58:01][d0] Parsed reading (OBIS code=7-0:3.0.0, value=1234.4200, unit=m3)
[Apr 22 21:58:01][d0] Parsed reading (OBIS code=7-0:1.7.0, value=0.00, unit=m3/h)
[Apr 22 21:58:01][d0] Read package with 2 tuples (vendor=SWE, baudrate=5, identification=@V1.01)
[Apr 22 21:58:01][chn0] Adding reading to queue (value=1234.42 ts=1682193481869)
[Apr 22 21:58:01][chn1] Adding reading to queue (value=0.00 ts=1682193481885)
[Apr 22 21:58:01][MAX] 1234.420000 @ 1682193481869
[Apr 22 21:58:01][push] push: { "data": [ { "uuid": "aed5b190-e144-11ed-aa45-5ff2ec105fec", "tuples":
[Apr 22 21:58:01][MAX] RESULT 1234.420000 @ 1682193481869
[Apr 22 21:58:01][mtr0] waiting 20 seconds before next reading
[Apr 22 21:58:01][chn1] ==> number of tuples: 1
[Apr 22 21:58:01][chn0] ==> number of tuples: 1
[Apr 22 21:58:01][chn1] compare: 1682193457688 1682193481885
[Apr 22 21:58:01][chn0] compare: 1682193457672 1682193481869
[Apr 22 21:58:01][chn1] JSON request body: [ [ 1682193481885, 0 ] ]
[Apr 22 21:58:01][push] CURL Request to http://127.0.0.1:5582 succeeded with code: 200
[Apr 22 21:58:01][push] push: { "data": [ { "uuid": "6c71d080-e146-11ed-a08b-67bc0afde140", "tuples":
[Apr 22 21:58:01][push] CURL Request to http://127.0.0.1:5582 succeeded with code: 200
[Apr 22 21:58:01][chn1] CURL Request succeeded with code: 200
[Apr 22 21:58:01][chn0] JSON request body: [ [ 1682193481869, 1234.4200000000001 ] ]
[Apr 22 21:58:01][chn0] CURL Request succeeded with code: 200
pi@raspberrypi:~/vzlogger $
```

vzlogger.conf Eintrag:

```
"meters": [  
  {  
    "enabled": true,  
    "allowskip": false,  
    "interval": 20,  
    "aggttime": -1,  
    "aggfixedinterval": false,  
    "channels": [  
      {  
        "uuid": "ihre uuid aed5b190-e144...",  
        "identifier": "7-0:3.0.0",  
        "api": "volkszaehler",  
        "middleware": "http://localhost/middleware.php",  
        "aggmode": "none",  
        "duplicates": 0  
      },  
      {  
        "uuid": "ihre uuid aed5b190-e144...",  
        "identifier": "7-0:1.7.0",  
        "api": "volkszaehler",  
        "middleware": "http://localhost/middleware.php",  
        "aggmode": "none",  
        "duplicates": 0  
      }  
    ],  
    "protocol": "d0",  
    "device": "/dev/ttyUSB0",  
    "pullseq": "2F3F210D0A",  
    "ackseq": "063035300d0a",  
    "baudrate": 300,  
    "baudrate_read": 9600,  
    "parity": "7e1",  
    "wait_sync": "off",  
    "read_timeout": 10,  
    "baudrate_change_delay": 1000  
  }  
]
```

2. Protokoll der Gas Box

Die Gas Box verwendet als Standard das Protokoll DIN EN 62056-21 / IEC 62056-21 mit Mode C. Dieses Protokoll verwenden auch viele Stromzähler. Dadurch ist die Gas Box leicht in Drittanbieter Software zu integrieren.

Über das Gas Box Configtool können noch zwei einfachere Protokolle konfiguriert werden. Sollten Sie selbst eine Software entwickeln, könnte das Ihnen das Einlesen der Gas Box erleichtern.

<p>Protokoll OBIS_ASCII_BIDIREKTIONAL (Default)</p> <p>Baudrate: 300 Databits: 7 Parity: Even Stopbits: 1</p> <p>Standard Protokoll nach DIN EN 62056-21, IEC-62056-21 MODE C.</p> <p>Abfrage mit /?! auf Baud 300. Danach Umstellmöglichkeit auf 300,600,1200,2400,4800, 9600 Baud. Danach folgt die Datenausgabe auf der gewünschten Baudrate.</p>	<p>Beispiel:</p> <p>Senden: /?!<CR><LF></p> <table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> <tr><td>/</td><td>?</td><td>!</td><td>$\backslash r$</td><td>$\backslash n$</td></tr> <tr><td>2F</td><td>3F</td><td>21</td><td>0D</td><td>0A</td></tr> </table> <p>Antwort der Gas Box:</p> <p>/SWE5\@V1.01$\backslash r \backslash n$</p> <p>Innerhalb von 2 Sekunden einen der 5 Befehle senden:</p> <p><ACK>000<CR><LF> (Baud 300) <ACK>010<CR><LF> (Baud 600) <ACK>020<CR><LF> (Baud 1200) <ACK>030<CR><LF> (Baud 2400) <ACK>040<CR><LF> (Baud 4800) <ACK>050<CR><LF> (Baud 9600)</p> <p>z.B. bei Baud 300 bleiben:</p> <table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr> <tr><td>□</td><td>0</td><td>0</td><td>0</td><td>$\backslash r$</td><td>$\backslash n$</td></tr> <tr><td>06</td><td>30</td><td>30</td><td>30</td><td>0D</td><td>0A</td></tr> </table> <p>Antwort der Gas Box:</p> <p>□7-0:3.0.0(0.7000*m3)$\backslash r \backslash n$ 7-0:1.7.0(0.00*m3/h)$\backslash r \backslash n$!$\backslash r \backslash n$ □`</p> <p>Erstes Zeichen: STX (HEX: 02) Vorletztes Zeichen: ETX (HEX: 03) Letztes Zeichen: Block Checksum Byte</p> <p>INFO:</p> <p><CR> = Carriage Return Zeichen ("$\backslash r$" HEX: 0D) <LF> = Line Feed Zeichen ("$\backslash n$" HEX: 0A) <ACK> = Acknowledgement Zeichen (HEX: 06)</p>	1	2	3	4	5	/	?	!	$\backslash r$	$\backslash n$	2F	3F	21	0D	0A	1	2	3	4	5	6	□	0	0	0	$\backslash r$	$\backslash n$	06	30	30	30	0D	0A
1	2	3	4	5																														
/	?	!	$\backslash r$	$\backslash n$																														
2F	3F	21	0D	0A																														
1	2	3	4	5	6																													
□	0	0	0	$\backslash r$	$\backslash n$																													
06	30	30	30	0D	0A																													

<p>Protokoll OBIS_ASCII_BIDIREKTIONAL_FIX</p> <p>Baudrate: 9600 Databits: 7 Parity: Even Stopbits: 1</p> <p>Abfrage mit /?! auf Baud 9600. Danach folgt die Datenausgabe.</p>	<p>Beispiel:</p> <p>Senden: /?!<CR><LF></p> <table border="1" data-bbox="887 360 1106 450"> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> <tr> <td>/</td> <td>?</td> <td>!</td> <td>\r</td> <td>\n</td> </tr> <tr> <td>2F</td> <td>3F</td> <td>21</td> <td>0D</td> <td>0A</td> </tr> </table> <p>Antwort der Gas Box:</p> <pre>/SWE5\@V1.01\r\n 07-0:3.0.0(0.7000*m3)\r\n 7-0:1.7.0(0.00*m3/h)\r\n !\r\n 0`</pre> <p>Erstes Zeichen: STX (HEX: 02) Vorletztes Zeichen: ETX (HEX: 03) Letztes Zeichen: Block Checksum Byte</p> <p>INFO: <CR> = Carriage Return Zeichen (“\r” HEX: 0D) <LF> = Line Feed Zeichen (“\n” HEX: 0A) <ACK> = Acknowledgement Zeichen (HEX: 06)</p>	1	2	3	4	5	/	?	!	\r	\n	2F	3F	21	0D	0A
1	2	3	4	5												
/	?	!	\r	\n												
2F	3F	21	0D	0A												
<p>Protokoll OBIS_ASCII_UNDIREKTIONAL</p> <p>Baudrate: 9600 Databits: 8 Parity: None Stopbits: 1</p> <p>Sendet alle 4-5 Sekunden automatisch einen Datensatz auf Baud 9600.</p>	<p>Beispiel:</p> <p>Kontinuierliche Ausgabe der Gas Box:</p> <pre>/SWE5\@V1.01\r\n 07-0:3.0.0(0.7000*m3)\r\n 7-0:1.7.0(0.00*m3/h)\r\n !\r\n 0`</pre> <p>Erstes Zeichen: STX (HEX: 02) Vorletztes Zeichen: ETX (HEX: 03) Letztes Zeichen: Block Checksum Byte</p> <p>INFO: <CR> = Carriage Return Zeichen (“\r” HEX: 0D) <LF> = Line Feed Zeichen (“\n” HEX: 0A) <ACK> = Acknowledgement Zeichen (HEX: 06)</p>															

Je nach Protokoll Konfiguration nimmt die Gas Box auf der dementsprechenden Baudrate, Databits, Parity und Stopbits noch folgende Kommandos entgegen:

Beschreibung	Command	Antwort der Gas Box
Firmware Version	/?!VERSION	ACK<\r><\n> 1.01<\r><\n>
Get Meter Protocol	/?!METERPROTOCOL! 0 = PROTOCOL_OBIS_ASCII_BIDIREKTIONAL 1 = PROTOCOL_OBIS_ASCII_BIDIREKTIONAL_FIX 2 = PROTOCOL_OBIS_ASCII_UNIDIREKTIONAL	ACK<\r><\n> 1<\r><\n>
Set Meter Protocol	/?!SETMETERPROTOCOL! 0 = PROTOCOL_OBIS_ASCII_BIDIREKTIONAL 1 = PROTOCOL_OBIS_ASCII_BIDIREKTIONAL_FIX 2 = PROTOCOL_OBIS_ASCII_UNIDIREKTIONAL	ACK<\r><\n> ACK<\r><\n> 0<\r><\n>
Get 300 Imp/m3	/?!300IMP! 10, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000	ACK<\r><\n> 100<\r><\n>
Set 300 Imp/m3	/?!SET300IMP!100 10, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000	ACK<\r><\n> 100<\r><\n>
Get 300 Count m3	/?!300CNT!	ACK<\r><\n> 0.7000<\r><\n>
Set 300 Count m3	/?!SET300CNT!1234.5678 Max: 9999999.9999	ACK<\r><\n> 1234.5678<\r><\n>
Clear 300	/?!CLEAR300!	ACK<\r><\n>
Get 170 m3/h	/?!170POW!	ACK<\r><\n> 0.41<\r><\n>

3. Tasten der Gas Box



Taste A	Einstellung der Impulse pro m ³
Taste B	Aktuell keine Funktion
Taste A und B gleichzeitig für 10 Sekunden gedrückt, bis „Restart“ im Display erscheint	Neustart der Gas Box (z.B. für Firmware Update)
Taste A für 20 Sekunden gedrückt, bis „Reset“ im Display erscheint	Zählerstand 3.0.0 wird auf 0 zurückgesetzt