

IoT / DIY / Haustechnik

ioBroker

Installation:

- OS: Ubuntu 22 inst.
- IP konfigurieren
- CURL installieren: `sudo apt install curl`
- NodeJS inst: `curl -sLf https://deb.nodesource.com/setup_16.x | sudo -E bash -`
`sudo apt-get install -y nodejs`
- IOBroker mit Curl installieren: `curl -sLf https://iobroker.net/install.sh | bash -`
- IOBroker starten mit [http://\[ip-adresse\]:8081](http://[ip-adresse]:8081)
- Auge: Discovery Adapter, Netzwerksan

Aktualisieren:

- `sudo apt-get update && sudo apt-get dist-upgrade`
- `iobroker update`
- `iobroker upgrade`
- `iobroker stop && iobroker fix -` -> ioBroker Installationsfixer
- `iob nodejs-update` - Node.js Update

ioBroker CLI Kommandos:

<https://iobroker.readthedocs.io/de/latest/basics/cli.html?highlight=json>

Backup bzw. Programm umziehen mit Backitup

- Backitup Adapter auf alter und neuer Installation installieren
- History Datenpfad konfigurieren: `/opt/iobroker/iobroker-data/history`
- je nach Installation min. ioBroker und History Backup auswählen und auf altem und neuem PC einmal ausführen
- `/opt/iobroker/backups/` - aktuellstes Backup auf neuen PC an gleiche Stelle übertragen
- auf neuem PC Backup (lokal) wiederherstellen (dauert eine Weile)
- im Idealfall gleiche IP wie Quellgerät nutzen
- alle Instanzen starten

Der Umzug ist fertig.

- Backup: NAS oder Cloud konfigurieren

Grundsätzliches

- Variablen sind temporär im Speicher. Sie lassen sich nur im selben Script nutzen.
- Datenpunkte liegen im Objektbaum und können übergreifend genutzt werden. Sie bleiben auch beim ioBroker-Neustart erhalten.
Eigene User-Datenpunkte liegen in bzw. unter `"0_userdata/0/"`.
Zu Datenpunkten lassen sich Einheit, Min., Max und Steps definieren.
- Werte werden vom Programm im Normalfall ohne Bestätigung

Seite 1 / 5

(c) 2024 Uwe Kernchen <news@uwe-kernchen.de> | 2024-04-18 21:08

URL: <https://uwe-kernchen.de/phpmyfaq/index.php?action=faq&cat=6&id=408&artlang=de>
(C) <https://uwe-kernchen.de>

IoT / DIY / Haustechnik

(bestaetigt=false) gesetzt. (rot)

Das Gerät erkennt am unbestätigten Wert, dass es eine Aktion ausführen soll

Nach der Aktion bestätigt das Gerät den Wert. (grün)

Blockly:

- * Adapter steuern: "steuere ID..." (setzt Ack=false)
- * Datenpunkt aktualisieren: "aktualisiere ID.."
- Datenpunkt JSON: hilfreich ist [JSON Online-Viewer](#)

VIS Objekte in andere Views kopieren

- funktioniert nicht über das GUI-Menue, dann werden die Objekte immer in der Quell-View eingefügt
- mit Strg-C / Strg-V funktioniert es korrekt

Berechnungen mit Bindings in VIS

HTML-Widget: Datenpunkt einer Variable zuweisen, dann rechnen.

(Der Editor zeigt die Ergebnisse nicht an, erst der View.)

```
{var1:fronius.1.meter.2.PowerReal_P_Sum;var1/1000} kW
```

math.Runden mit (2) Nachkommastellen:

```
{wert:Eigene.0.Timer;parseInt(wert).toFixed(2)}
```

Wenn-Dann Abfrage:

```
{wert:Eigene.0.Binding; wert > 5 ? "der Wert ist größer als 5":""}
```

Gesetzt oder leer - Abfrage:

```
{wert:Eigene.0.Binding; wert ? "gesetzt" : "leer"}
```

Wahr-Falsch Abfrage:

```
{wert:Eigene.0.Binding; wert === "true" ? "wahr" : "falsch"}
```

If-When-Else Abfrage:

```
{wert:Eigene.0.Binding; wert > 5 ? "der Wert ist größer als 5" : "der Wert ist kleiner als 5"}
```

If-AND-If ELSE Abfrage

```
{wert:Eigene.0.Binding; wert > 5 && wert < 10 ? "der Wert ist größer als 5 oder kleiner als 10":"der Wert liegt nicht im definierten Bereich"}
```

Farben (rot/grün) bedingungsabhängig setzen

(Bedingung in das Color-Attribut des Widgets setzen)

```
{wert:Eigene.0.Binding; wert > 5 ? "#ff0000" : "#00ff00"}
```

Ausgabe Teilstring (Start, Anzahl)

```
{wert:Eigene.0.Binding; wert.substring(0,4)}
```

Vollständige Liste: <https://github.com/ioBroker/ioBroker.vis#bindings-of-objects>

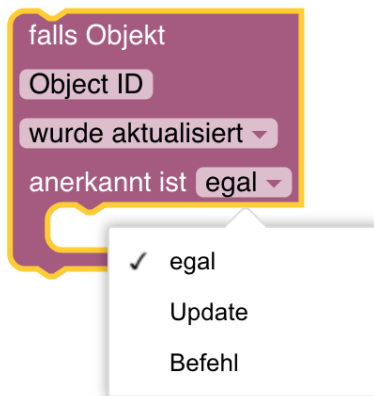
Quellen:

<https://www.machs-smart.de/iobroker-bindings-farben-berechnungen-abfragen/>

IoT / DIY / Haustechnik

Arbeiten mit Blockly:

- **Scripte** werden sofort 1x beim Script-Start ausgeführt -> Start über Trigger
- **Funktionen** können Logikblöcke zusammen fassen
- **steuere** schreibt den neuen Wert mit Bestätigt: false
- **aktualisiere** schreibt den neuen Wert mit Bestätigt: true
 1. Ein neuer Workflow wird angestoßen, indem ein Datenpunkt mit Bestätigt = false geschrieben wird
 2. Diese Änderung registriert dann ein Adapter, welcher diesen Datenpunkt "überwacht" (subscribe)
 3. Der Adapter macht daraufhin seine Arbeit und
 4. schreibt danach den Datenpunkt erneut(!!!) / bestätigt den Wert mit einem Update Bestätigt: true
- "anerkannt ist" meint dieses "Bestätigt"-Flag:



Update = Bestätigte Änderung
(z.B. durch "aktualisiere")

Befehl = Unbestätigte Änderung
(z.B. durch "steuere")

- Trigger mehrfach aufrufen startet ihn auch mehrfach! **Stop Trigger** in Schleifen nicht vergessen.
- **Adapter reagieren nur auf unbestätigte Änderungen und bestätigen diese.**
Bestätigte Reaktion des Adapters abfragen: Objekt= wurde aktualisiert, anerkannt ist= Update
- **Nicht überwachte eigene Datenpunkte (ohne Adapter) müssen per "Bestätigter Änderung" gesetzt werden.**
Eigene Datenpunkte per Script bestätigen: TRIGGER falls Objekt= wurde geändert, anerkannt ist= Befehl, Aktion= Aktualisiere Objekt mit "Wert" mit Verzögerung 1sek.

Quelle:

<https://haus-automatisierung.com/software/iobroker/2021/02/02/iobroker-steuere-aktualisiere-bestaetigt.html>

ALEXA Routine steuert ioBroker

1. Routine in Alexa anlegen, Bedingung: gewünschter Steuer-Text, Aktion benutzerdefiniert: Sprachausgabe "stopp"
 2. ioBroker/Blockly Script fragt Datenpunkt "alexa/history/summary" ab (Text der Alexa-Routine) und ggf. noch "alexa/history/name" (Echo Device Name)
- Quelle: [\(statisches\) Alexa Kommando ohne Cloud \(iobroker.net\)](https://statisches.com/Alexa-Kommando-ohne-Cloud-iobroker.net/)

IoT / DIY / Haustechnik

meine **Adapter**:

- Visualisation (Community Lizenz kostenlos, wird online geprüft)
- Javascript / Blockly - Scripting
- History - Historie-Daten (JSON-basiert, Speicherdauer (1 Jahr), Verzeichnis festlegen)
oder SQL-Protokollierung - (SQL-DB: MySQL, PostgreSQL, MS-SQL oder SQLite3)
für beliebige Datenpunkte im Objektbaum aktivierbar
- Flot - zur grafischen Auswertung der History-Daten
- Fronius Inverter (fronius.0, fronius.1) - Solar Wechselrichter + SmartMeter
- Maxx! Cube (maxcube.0) - Heizkörperthermostate
- Midea und Carrier- Dimstal, Nethome klimageräte
- [Tuya](#) - WLAN Thermometer, Gosund Steckdosen
- Shelly - Shelly Adapter
- Alexa 2 - Sprachsteuerung
- OpenWeatherMap - Wetterpunkte im Objektbaum, lassen sich prima scripten, mit Windstärke und Richtung
- auch cool: <https://www.wetteronline.de/wetter-widget> konfigurieren, iFrame Code in manuellen Datenpunkt (Wert) einfügen und per Widget String(unescaped) fertig einbinden

Liste aller Adapter: <http://download.iobroker.net/list.html>

Nach meinem Empfinden Mist:

- XTerm - Shell-Befehle auf dem ioBroker Host (sinnlos: kein iobroker stop o.ä.)

"Datum-Objekt" - gibt Datum + Zeit im UNIX Timestamp-Format in ms aus.

Man kann direkt damit rechnen und vergleichen.

1 Sekunde = 1.000 ms

1 Minute = 60.000 ms

1 Stunde (60 Minuten) = 3.600.000 ms

1 Tag = 86.400.000 ms

Quellen:

- ioBroker Doku: <https://www.iobroker.net/#de/documentation>
- Erste Schritte mit Blockly: <https://forum.iobroker.net/topic/58950/erste-schritte-mit-blockly/2>
- IOBroker und Blockly Grundlagen: <https://www.smarthome-tricks.de/einfuehrung-und-grundlagen/>
- ioBroker Blockly: <https://hobbyblogging.de/iobroker-blockly>

Tests:

- Dimstal Midea Klimageräte: Adapter: Homebridge Zubehör Manager
- danach erscheint links im Menue ein neuer Menüpunkt "Homebridge" und

IoT / DIY / Haustechnik

Objekte: ham/

- Homebridge -> midea-air (<https://github.com/hillaliy/homebridge-midea-air>)
- besser geht es über Alexa / Smart-Home-Devices

Test2:

Bei der Siemens Logo8 (S7-Adapter) brauchst du nur bei DB was eintragen.

Du kannst Netzwerk Eingänge und Ausgänge in dein Logo-Programm hinzufügen.

Die Adresse die du da einstellst musst du bei den DB eintragen

z.b. Netzwerk Eingang 0.0 --> DB0.0 Typ: bool

usw.

Du kannst auch Ausgänge lesen

Q0 --> db1065.0

Q1 --> db1065.1

Typ: bool

Eingänge

E0 --> db1025.0

E1 --> db1025.1

Typ: bool

Analogwerte schreiben und lesen geht auch.

Links:

- ioBroker Forum: [Deutsch \(iobroker.net\)](https://www.iobroker.net)

Eindeutige ID: #1407

Verfasser: Uwe Kernchen

Letzte Änderung: 2024-01-08 14:16