Smart Home

Alles rund ums intelligente Heim

- Fhem
  - FHEM Installation
  - Eltako mit Fhem verbinden
  - FHEM Eltako Konfig Übersicht
  - Fhem Config
  - Modul erstellen
- HomeAssistant
  - HomeAssistant Installieren in Proxmox
- iobroker
  - ioBroker Schulung
- HomeAssistant
- ioBroker
- N8N

Fhem

FHEM Installation

https://wiki.fhem.de/wiki/Erste_Schritte_in_FHEM

Back to the roots

Mit Fhem hat alles begonnen. Dann wollen wir mal zurück zu den basics.

> Tutorial

Installation auf Raspberry

Raspberry ist eingerichtet, Zugriff über SSH

Nötige Programme installieren

```
sudo apt-get -f install && sudo apt-get update && sudo apt-get -y install perl libdevice-serialport-perl libio-socket-ssl-perl libwww-perl libxml-simple-perl libdbd-sqlite3-perl libtext-diff-perl libxml-simple-perl libcgi-pm-perl libjson-perl sqlite3
```

Aktuelles Fhem laden

```
wget http://fhem.de/fhem-6.3.deb
```

Hier natürlich darauf achten, dass die aktuelle Version verwendet wird.

Aktuelleste Version hier zu finden

Aktuelle Fhem installieren

```
sudo dpkg -i fhem-5.8.deb
```

Berechtigungen setzen

```
cd /opt && sudo chmod -R a+w fhem && sudo usermod -a -G tty pi && sudo usermod -a -G tty fhem
```

Nun ist Fhem über die IP-Adresse und den Port 8083 erreichbar

Fhem Update

```
update


shutdown restart
```

Fhem Schreibschutz aufheben

Wenn du vertraut mit der FHEM Konfigurationsdatei bist und weisst was du tust, kannst du den Schreibschutz aufheben indem du das Attribute *editConfig auf 1 setzt.*

Am einfachsten kannst du das editConfig Attribut auf 1 setzen wenn du diesen Befehl

```
attr WEB editConfig 1
```

oben in das Befehls-Textfeld ein gibst und ENTER drückst.

Eltako mit Fhem verbinden

Links zu Doku

https://wiki.fhem.de/wiki/EnOcean_Starter_Guide

Benötigte Hardware

- Raspberry Pi mit FHEM
- Eltako FAM14
- Eltako FGW14
- Eltako Aktoren z.B. FSR14

Konfiguration

FGW14 einbinden

```
# Konfiguration der seriellen Schnittstelle:
#
# Drehschalter des FGW14 auf Pos 6 = 57600 Baud
#
define TCM_ESP2 TCM ESP2 /dev/ttyUSB@57600
attr TCM_ESP2 comType RS485
attr TCM_ESP2 alias Eltako FGW14
attr TCM_ESP2 verbose 3
```

Startet man FHEM nach dem Speichern der Konfigurationsänderungen neu sollte man das fhem.log kontrollieren.
Dort findet man nun folgende Einträge:

```
2014.04.02 08:43:08 3: Opening TCM120 device /dev/ttyS3
2014.04.02 08:43:08 3: Setting TCM120 baudrate to 57600
2014.04.02 08:43:08 3: TCM120 device opened
2014.04.02 08:43:08 1: get baseID: Bogus answer received:
000000001520C5A55A8B0550000000000000023012
```

Die dritte Zeile signalisiert, dass erfolgreich eine Verbindung zum FGW14 hergestellt werden konnte. Anschliessend versucht FHEM die BaseID des vermeintlichen TCM120 Moduls abzufragen
Da es sich bei einem FGW14 Modul um kein TCM120 Funkmodul mit eigener BaseId handelt kommt es zu obiger "Bogus answer received" Fehlermeldung.
Diese ist nicht weiter tragisch und kann einfach ignoriert werden.

Wenn in fhem.cfg folgendes Attribut gesetzt wurde

```
attr global autoload_undefined_devices 1
```

Dann erzeugt FHEM automatisch neue Einträge in der fhem-Konfiguration,
sobald von einem Sensor oder Aktor das erste Mal eine Nachricht auf dem RS485 Bus ausgetauscht wird.
Befindet sich der obere Drehschalter des FAM14 in Position 4 erhält FHEM über das FGW14 zyklisch Statusmeldungen von allen Aktoren am RS485 Bus.
Auf Basis dieser Statusmeldungen erzeugt FHEM bereits Defines.
Für ein FSR14-4x wird pro Kanal ein Konfiguration in der folgenden Form generiert:

```
define EnO_switch_00000007 EnOcean 00000007
attr EnO_switch_00000007 IODev TCM120
attr EnO_switch_00000007 room EnOcean
attr EnO_switch_00000007 subType switch
define FileLog_EnO_switch_00000007 FileLog ./log/EnO_switch_00000007-%Y.log EnO_switch_00000007
attr FileLog_EnO_switch_00000007 logtype text
attr FileLog_EnO_switch_00000007 room EnOcean
```

Eltako RS485-Busaktoren mit FHEM ansteuern:

Steuert man Enocean Aktoren auf dem herkömmlichen Weg über TCM120 oder TCM310 USB Adapter
über Funk an, muss FHEM eine Absender ID aus dem Adressbereich des jeweiligen TCM USB-Adapters verwenden. Diese Einschränkung gilt nicht für die Kommunikation mit den Aktoren über den RS485 Bus. D.h., FHEM darf beliebige IDs verwenden, solange diese eindeutig sind.
Möchte man mit einem FHEM-Webcmd Schalter einen Kanal eines FSR14-4x schalten, so muss die automatisch generierte Konfiguration für das FSR14-4x Relais angepasst werden.

Dazu wird dem FHEM Schalter die frei gewählte ID 00100007 zugeordnet. Um die Übersicht über vergebene IDs zu behalten, sollte man diese systematisch vergeben: z.B. FHEM-ID= Aktor-ID+0x00100000.
Nun müssen FHEM-ID und Aktor-ID einander zugeordnet werden. Dies erfolgt über ein Sub-Define innerhalb des FSR14 Defines mit dem subDef Attribut, dem man die FHEM-ID zuweist.
Im folgenden Beispiel ist nun der FHEM-Schalter mit der ID 00100007 dem Aktor mit der ID 00000007 zugeordnet.

```
#
# LAMPE
#
# B0: Lampe an
# BI: Lampe aus
#

define EnO_switch_00000007 EnOcean 00000007
attr EnO_switch_00000007 IODev TCM120
attr EnO_switch_00000007 alias LAMPE
attr EnO_switch_00000007 event-on-change-reading state,buttons,channelA,channelB
attr EnO_switch_00000007 group Beleuchtung
attr EnO_switch_00000007 gwCmd switching
attr EnO_switch_00000007 icon light_outdoor
attr EnO_switch_00000007 room Garten
attr EnO_switch_00000007 subDef 00100007
attr EnO_switch_00000007 subType gateway
```

Konfiguration der Aktoren

FSR-14

```
eep A5-38-08
gwCmd switching
subType gateway
subDef 00100004
```

FUD-14

```
eep 15-38-08
gwCmd dimming
```

subType gateway

model Eltako_TF

subDef 00100008

FSB-14 Rolladenaktor

manufID 00D

eep A5-3F-7F

subType manufProfile

model Eltako_FSB_ACK

Konfiguration Eltako mit PCT14

hier müssen die jeweiligen Aktoren mit dem dazugehörigen subDef verbunden werden.

ID ist dabei das Subdef und Funktion entweder 51 oder 31 oder dementsprechend.

Hardware-Konfiguration

Das FGW14 muss mit dem Bus verbunden sein. Und die Verbindung von Hold darf nicht vergessen werden.

FHEM Eltako Konfig Übersicht

SenderID-Tabelle

Nr.	Name EnOcean	Nahme in FHEM	Hex (Sender-ID)	Zimmer
	FGW14	FGW14		Technik
1	EnO_	Licht	00100004	Esszimmer

Fhem Config

fhem.cfg

Autocreate

```
define autocreate autocreate
```

Standardmaßig aktiv - Erstellt automatisch neue Einträge

Modul erstellen

HomeAssistant

HomeAssistant Installieren in Proxmox

HomeAssistant auf Proxmox installieren in 5 Minuten! Dank Install-Skript

22. Februar

In diesem Video zeige ich euch, wie Ihr HomeAssistant in nur 5 Minuten unter Proxmox installieren könnt, das ganze funktioniert Dank folgendem Skript:

Auf der Shell von Proxmox-Node folgenden Befehl eingeben

bash -c "$(wget -qLO - https://github.com/tteck/Proxmox/raw/main/vm/haos-vm.sh)"

> Das Skript erstellt eine neue VM. Der Speicherplatz kann ausgewählt werden. Danach ist Homeassistant über die IPAdresse und Port 8123 erreichbar

Erste Schritte in HomeAssistant (von Simon42): Home Assistant Grundlagen für Anfänger - From Zero to Hero □□♂

iobroker

Der Obergeile Alleskönner

ioBroker Schulung

https://shop.haus-automatisierung.com/customer/video/

HomeAssistant

HomeAssistant Installieren

https://y.com.sb/watch?v=1Un4zJJWUTE

ioBroker

Schulung zu iobroker

Installation ioBroker

ioBroker Installation NEU:

nur noch EIN Befehl!

1. apt update && apt upgrade -y
2. apt install curl -y
3. curl -sLf https://iobroker.net/install.sh | bash -

https://www.youtube.com/embed/l3_VQJ9sHbU

ioBroker Update

Mit der Konsole folgendes auführen:

```
iob backup
iob stop
iob update
iob upgrade self
iob start
```

Node Update

WAS IST NODE.JS UND WARUM MUSS MAN ES UPDATEN?

Node.js ist die Laufzeitumgebung der Programmiersprache JavaScript, in der ioBroker geschrieben ist. Ohne Node.js funktioniert ioBroker nicht. Node.js hast Du initial selbst installiert oder der ioBroker-Installer hat dies für dich getan.

Wie bei vielen Open-Source-Technologien üblich, entwickelt sich Node.js schnell weiter. Kleinere Updates, die die Stabilität und Sicherheit steigern oder gar neue Funktionen hinzufügen, erscheinen regelmäßig.

Node.js-Versionen mit gerader Hauptversionsnummer werden als LTS-Versionen (**L**ong **T**erm **S** upport) bezeichnet und einige Jahre gepflegt (z.B. 12.x). Jedes Jahr kommt eine neue Version ins LTS - in diesem Jahr (2021) ist das Node.js 16, welche im April veröffentlicht wurde und ab Oktober 2021 eine LTS Version wird.

Im gleichem Zug erreichen frühere LTS-Versionen ihr Lebensende (EOL, **E**nd **o**f **L**ife). So hat Node.js 8 im April 2020 den EOL-Status erhalten und bekommt damit keine Updates mehr, Nodejs 10.x wird Ende April 2021 Ihr Lebensende erreichen. Es wird also keine Sicherheits-Updates mehr geben! Node.js 12.x wird im April 2022 eol geben.

Alle Node.js-Versionen mit ungeraden Versionsnummern sind Entwicklungsversionen und sollten nicht produktiv genutzt werden.

ioBroker nutzt viele Module und Erweiterungen aus der JavaScript Open-Source Szene, und dort kommt es regelmäßig vor, dass Versionen die EOL gehen zeitnah danach auch nicht weiter unterstützt werden. Das hat im ersten Schritt keine echte Auswirkung, aber mittelfristig wird es also Adapter, und später auch den js-controller geben, der EOL Versionen von Node.js nicht mehr unterstützt.

Node.js 10 wird mit dem js-controller 3.x voll unterstützt. Ab dem js-controller 4.0 (Februar 2022) ist Node.js 10.x nicht mehr untertsützt.

AUF WELCHE NODE.JS VERSION UPDATEN?

Aktuell empfiehlt ioBroker die Nutzung von Node.js 16.x.

Folgende Adapter haben momentan Probleme mit Node.js 14:

- jeelink

Folgende Adapter haben momentan Probleme mit Node.js 16:

- jeelink?

**Node.js 16.x wird auch vom js-controller 3.3 grundsätzlich unterstützt, aber nur mit npm 6! npm 7 bzw 8 sind mit dem js-controller 4.0 nutzbar.*

UPDATE VORBEREITEN

Node.js Version prüfen

Bevor man beginnt, sollte man in der Befehlszeile mit dem Befehl

```
node -v
```

überprüfen, welche Version von Node.js gerade installiert ist. Eine gute Idee ist es, diese Versionsangabe auch mit der Node.js-Version im Übersichts-Fenster des ioBroker-Admins für diesen Host zu vergleichen. Sollten sich die Versionen unterscheiden, sind mehrere Node.js-Varianten installiert, was zu Problemen führen kann. **Diese Probleme müssen VOR dem Update dann behoben werden!** Anleitung zB unter https://forum.iobroker.net/topic/35090/howto-nodejs-installation-und-upgrades-unter-debian/2

Betriebssystem prüfen

Dann auch prüfen was man für ein Betriebssystem hat. Vor allem im Raspi Umfeld sind gern auch älterer Systeme auf basis von "Debian jessie" oder "Debian wheezy" im Einsatz. Für die gibt es nichts was höher ist als Nodejs 10, da steht dann ggf auch ein Betriebssystemupdate an, was wir hier aber nicht behandeln können.

Unterstützte Linux Distributionen sind unter https://github.com/nodesource/distributions#debian-and-ubuntu-based-distributions aufgelistet.
Unter Debian und Ubuntu gibt es mit `lsb_release -a` eine Ausgabe was man aktuell nutzt.

js-controller Version prüfen

Weiterhin bitte prüfen welche js-controller Version Installiert ist (ebenfalls auf dem Host-Tab im Admin einsehbar).
Bei Versionen VOR js-controller 3.x, wenn möglich bitte zuerst den js-controller aktualisieren. Am besten auf mindestens die 3.2! Hierzu gibt es extra Threads im Forum wie z.B.

https://forum.iobroker.net/topic/42385/js-controller-3-2-jetzt-im-stable bzw

https://forum.iobroker.net/topic/52886/js-controller-4-0-x-jetzt-für-alle-user-im-stable

Adapter aktualisieren

Damit es nach dem Update zu keinen Inkompatibilitäten oder Probleme kommt, sollte man alle Adapter prüfen und aktualisieren. Vor allem Adapter mit nativen Bestandteilen, wie alles mit Serialport oder Bluetooth können Probleme bereiten. Hier am besten die Adapter-Readme's per Admin oder im GitHub prüfen, ob neue Versionen zur Verfügung stehen die die geplante Node.js Version explizit erst unterstützen.

Bei Updates wo es größere Versionssprünge bei npm gibt (zb Node.js 14->16 updated npm von 6.x auf 8.x) kann es sehr hilfreich sein wenn man schaut ob Adapter die von GitHub installiert wurden inzwischen in der gleichen version auf auf npm liegen und dann ggf von dort nochmals installieren oder updaten. Im Admin werden Adapter die per GitHub installiert wurden gesondert mit einem GitHub Symbol angezeigt. Das hilft auch im Vorfeld Probleme zu vermeiden.

Wenn man diesen Schritt nicht durchführt kann es zu unnötigen Problemen beim update der Adapter kommen!

Backup erstellen

Zuerst muss natürlich unbedingt ein Backup erstellt werden. Dazu kann z.B. der BackItUp-Adapter genutzt oder der Kommandozeilenbefehl

```
cd /opt/iobroker
iobroker backup
```

ausgeführt werden. Das Backup sollte aktuell sein, damit möglichst keine Daten verloren gehen.

NODE.JS UPDATEN

Für Windows-Systeme kann ich leider gerade nichts genaues sagen, wir schauen das wir das noch ergänzen. Aufruf an die Community: Wer Schritte hat gern als eigener Post oder hier einbringen □□ Danke

Einen Post aus der Community gab es dazu: https://forum.iobroker.net/post/624003

Linux-Systeme

ioBroker stoppen
Zuerst ioBroker stoppen, damit Updates keine Nebeneffekte oder Abstürze verursachen.

```
iobroker stop
```

Bitte anschließend im Webbrowser prüfen, dass der ioBroker-Admin danach wirklich nicht mehr läuft. Sollte er weiterhin aufrufbar sein, dann den Rechner neu starten und nochmals „iobroker stop“ ausführen und erneut testen. Für die Techniker unter uns: Man kann auch mit einem Tool wie "top" prüfen, ob noch Prozesse existieren, die mit "io." beginnen. Die dann am besten mit einem beherzten "sudo kill -9 <ProzessID>" zwangsbeenden.

Node.js updaten
Jetzt aktualisiert man Node.js auf die gewünschte neue Version.

Unter Linux reicht es, dazu den Nodesource-Installationsbefehl für das jeweilige Betriebssystem auszuführen. Verschiedene Varianten (auch Root und Nicht-Root) sind unter

https://github.com/nodesource/distributions#debinstall gelistet.

Zum Beispiel lauten die Befehle für einen Raspberry Pi der ein Debian bzw. Raspbian-Image verwendet wie folgt, wenn man *nicht* als root-User (z.B. richtig mit dem User "pi") angemeldet ist:

```
curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
sudo apt install -y nodejs
```

Für Node.js 16 einfach in der URL oben anstelle der 14 eine 16 reinschreiben.

Für macOS gibt einen Installer auf https://nodejs.org/en/download/ , den man einfach ausführt.

Ob die Aktualisierung geklappt hat, kann man wieder mit dem Befehl

```
node -v
```

überprüfen.

Auch die npm Version sollte mit

```
npm -v
```

geprüft werden. Mit einem js-controller <4 sicherstellen das idealerweise keine 7.x/8.x von npm installiert ist!

ioBroker fixer ausführen
Da die Installation von Node.js einige Einstellungen am System verändert haben kann, ist es jetzt ratsam, den ioBroker-Installationsfixer aufzurufen. Das geschieht mit dem Befehl

```
iobroker fix
```

Er stellt unter anderem die für den Betrieb von ioBroker notwendigen Sicherheitseinstellungen wieder her und prüft und korrigiert alle Berechtigungen. Das kann einen Augenblick dauern, bitte Geduld haben.

ERSTER IOBROKER NEUSTART NACH UPDATE

Einige genutzte JavaScript Module haben binäre Teile, welche bei einem Node.js Update nicht mehr kompatibel sind und neu erstellt werden müssen.

Automatische Rebuilds

ioBroker versucht seit dem js-controller 3.0 automatisch die Adapter zu erkennen die nicht starten weil Sie aktualisiert werden müssen. Dies funktioniert so das die typischen Fehlermeldungen erkannt werden und ioBroker dann die Aktualisierung versucht.

js-controller 3.x
Zuerst wird ein "rebuild" des betroffenen Adapters ausgeführt, falls das nicht hilft werden die Adapter-Abhängigkeiten aktualisiert.

js-controller 4.0
Zuerst wird versucht alle Adapter zu rebuilden, falls das nicht hilft wird versucht zielgerichtet die relevanten Module neu zu bauen.

Daher kann es sein das der Adapter mehrfach ersucht wird neu zu starten. **Hier bitte UNBEDINGT Geduld haben!** Erst wenn der Adapter dauerhaft rot bleibt und auch im Log steht das der Rebuild nicht geklappt hat aktiv werden!

Bei einigen Adaptern (zB iot die optionale native Abhängigkeiten haben) funktioniert die automatische Erkennung nicht und das rebuild muss manuell angestoßen werden. Dies kann dadurch erkannt werden das der Adapter "Rot" bleibt und nicht startet oder einzelne Funktionen nicht gehen und das als Fehler im Log steht. Dann sollte das Log geprüft werden (neben Admin stehen Logfiles auch unter /opt/iobroker/log/... zur Verfügung.

Manuelle Rebuilds

Hier zu gibt es `iobroker rebuild`, bzw die Kommandos die im Log angezeigt werden falls der Automatische Rebuild nicht funktioniert.

Sonderfälle (z.B. Serialport)

Leider gibt es Sonderfälle, wo auch die obigen Optionen das Rebuild nicht erledigen, einer davon ist Serialport.

Dort kann ein Log zB (auch nach allen Rebuild Versuchen) wie folgt aussehen

```
host.SmartHomeCenter | 2020-05-10 09:28:01.788 | error | Caught by controller[0]: }
host.SmartHomeCenter | 2020-05-10 09:28:01.788 | error | Caught by controller[0]: ]
host.SmartHomeCenter | 2020-05-10 09:28:01.788 | error | Caught by controller[0]:
'/opt/iobroker/node_modules/serialport/compiled/12.16.3/linux/arm/serialport.node'
host.SmartHomeCenter | 2020-05-10 09:28:01.787 | error | Caught by controller[0]:
'/opt/iobroker/node_modules/serialport/build/default/serialport.node',
host.SmartHomeCenter | 2020-05-10 09:28:01.787 | error | Caught by controller[0]:
'/opt/iobroker/node_modules/serialport/Release/serialport.node',
host.SmartHomeCenter | 2020-05-10 09:28:01.787 | error | Caught by controller[0]:
'/opt/iobroker/node_modules/serialport/out/Release/serialport.node',
```

host.SmartHomeCenter | 2020-05-10 09:28:01.787 | error | Caught by controller[0]:
'/opt/iobroker/node_modules/serialport/Debug/serialport.node',
host.SmartHomeCenter | 2020-05-10 09:28:01.787 | error | Caught by controller[0]:
'/opt/iobroker/node_modules/serialport/out/Debug/serialport.node',
host.SmartHomeCenter | 2020-05-10 09:28:01.786 | error | Caught by controller[0]:
'/opt/iobroker/node_modules/serialport/build/Release/serialport.node',
host.SmartHomeCenter | 2020-05-10 09:28:01.786 | error | Caught by controller[0]:
'/opt/iobroker/node_modules/serialport/build/Debug/serialport.node',
host.SmartHomeCenter | 2020-05-10 09:28:01.786 | error | Caught by controller[0]:
'/opt/iobroker/node_modules/serialport/build/serialport.node',
host.SmartHomeCenter | 2020-05-10 09:28:01.786 | error | Caught by controller[0]: tries: [
host.SmartHomeCenter | 2020-05-10 09:28:01.786 | error | Caught by controller[0]: at Module._compile
(internal/modules/cjs/loader.js:1133:30) {
host.SmartHomeCenter | 2020-05-10 09:28:01.785 | error | Caught by controller[0]: at Object.
(/opt/iobroker/node_modules/serialport/lib/bindings/auto-detect.js:16:22)
host.SmartHomeCenter | 2020-05-10 09:28:01.785 | error | Caught by controller[0]: at require
(internal/modules/cjs/helpers.js:77:18)
host.SmartHomeCenter | 2020-05-10 09:28:01.785 | error | Caught by controller[0]: at Module.require
(internal/modules/cjs/loader.js:1019:19)
host.SmartHomeCenter | 2020-05-10 09:28:01.785 | error | Caught by controller[0]: at Function.Module._load
(internal/modules/cjs/loader.js:877:14)
host.SmartHomeCenter | 2020-05-10 09:28:01.785 | error | Caught by controller[0]: at Module.load
(internal/modules/cjs/loader.js:977:32)
host.SmartHomeCenter | 2020-05-10 09:28:01.784 | error | Caught by controller[0]: at
Object.Module._extensions..js (internal/modules/cjs/loader.js:1153:10)
host.SmartHomeCenter | 2020-05-10 09:28:01.784 | error | Caught by controller[0]: at Module._compile
(internal/modules/cjs/loader.js:1133:30)
host.SmartHomeCenter | 2020-05-10 09:28:01.784 | error | Caught by controller[0]: at Object.
(/opt/iobroker/node_modules/serialport/lib/bindings/linux.js:2:36)
host.SmartHomeCenter | 2020-05-10 09:28:01.784 | error | Caught by controller[0]: at bindings
(/opt/iobroker/node_modules/serialport/node_modules/bindings/bindings.js:93:9)
host.SmartHomeCenter | 2020-05-10 09:28:01.783 | error | Caught by controller[0]: →
/opt/iobroker/node_modules/serialport/compiled/12.16.3/linux/arm/serialport.node
host.SmartHomeCenter | 2020-05-10 09:28:01.783 | error | Caught by controller[0]: →
/opt/iobroker/node_modules/serialport/build/default/serialport.node
host.SmartHomeCenter | 2020-05-10 09:28:01.783 | error | Caught by controller[0]: →
/opt/iobroker/node_modules/serialport/Release/serialport.node
host.SmartHomeCenter | 2020-05-10 09:28:01.783 | error | Caught by controller[0]: →
/opt/iobroker/node_modules/serialport/out/Release/serialport.node
host.SmartHomeCenter | 2020-05-10 09:28:01.782 | error | Caught by controller[0]: →

```
/opt/iobroker/node_modules/serialport/Debug/serialport.node
host.SmartHomeCenter | 2020-05-10 09:28:01.782 | error | Caught by controller[0]: →
/opt/iobroker/node_modules/serialport/out/Debug/serialport.node
host.SmartHomeCenter | 2020-05-10 09:28:01.782 | error | Caught by controller[0]: →
/opt/iobroker/node_modules/serialport/build/Release/serialport.node
host.SmartHomeCenter | 2020-05-10 09:28:01.782 | error | Caught by controller[0]: →
/opt/iobroker/node_modules/serialport/build/Debug/serialport.node
host.SmartHomeCenter | 2020-05-10 09:28:01.781 | error | Caught by controller[0]: →
/opt/iobroker/node_modules/serialport/build/serialport.node
host.SmartHomeCenter | 2020-05-10 09:28:01.781 | error | Caught by controller[0]: Error: Could not locate the
bindings file. Tried:
host.SmartHomeCenter | 2020-05-10 09:28:01.781 | error | Caught by controller[0]: ^
host.SmartHomeCenter | 2020-05-10 09:28:01.780 | error | Caught by controller[0]: throw err
host.SmartHomeCenter | 2020-05-10 09:28:01.780 | error | Caught by controller[0]:
/opt/iobroker/node_modules/serialport/node_modules/bindings/bindings.js:96
```

Es gibt auch andere Fehlermeldungen die aber alle auf das gleiche hinauslaufen.
Die einfachste Option ist es dann manuell im richtigen Verzeichnis neu zu bauen.
In dem Fall das Verzeichenis mit "bindings" suchen - oben ist das /opt/iobroker/node_modules/serialport/node_modules/bindings ... bei neueren Versionen kann es auch etwas wie /opt/iobroker/node_modules/serialport/node_modules/@serialport/bindings sein.
Dann in dieses Verzeichnis wechseln und `npm install --production` ausführen. Danach den Adapter nochmal sneu starten, das sollte dann tun.

Ein weiterer Fall sind Adapter mit canvas Modul (ggf echarts oder Mihome-vacuum) wo es Probleme gebe kann.

Andere Sonderfälle muss man sich im Detail ansehen. Bitte unten Posten und wir unterstützen.

WEITERE NOTFALL OPTIONEN

Im früheren Artikel unter https://forum.iobroker.net/topic/22867/how-to-node-js-für-iobroker-richtig-updaten sind noch weitere manuelle Möglichkeiten beschrieben ioBroker wieder zum laufen zu bekommen, aber diese sollten an sich nicht mehr nötig sein, gehen aber natürlich auch noch! Dieser Artikel gilt also auch weiterhin.

Jetzt viel Erfolg und gebt bitte Feedback wie git es geklappt hat oder welche Probleme Ihr habt.

N8N

Docker muss installiert sein

Docker, Docker-Compose und Portainer installieren

```
mkdir n8n
```

```
cd n8n
```

Dockercompose.yml anlegen

```
nano docker-compose.yml
```

```
services:
  n8n:
    image: n8nio/n8n:latest
    restart: always
    ports:
      - "5678:5678"
    environment:
      - DB_TYPE=postgresdb
      - DB_POSTGRESDB_HOST=postgres
      - DB_POSTGRESDB_PORT=5432
      - DB_POSTGRESDB_DATABASE=n8n
      - DB_POSTGRESDB_USER=n8n
      - DB_POSTGRESDB_PASSWORD=Bitteaendern2
      - GENERIC_TIMEZONE=Eruope/Berlin
      - N8N_SECURE_COOKIE=false
    volumes:
      - n8n_data:/home/node/.n8n
    depends_on:
      - postgres
  postgres:
    image: postgres:14
    restart: always
    environment:
```

```
      - POSTGRES_USER=n8n
      - POSTGRES_PASSWORD=Bitteaendern2
      - POSTGRES_DB=n8n
    volumes:
      - postgres_data:/var/lib/postgresql/data

volumes:
  n8n_data:
  postgres_data:
```

```
docker compose up -d
```