

Kurzanleitung

„Integration des AMIS-Lesekopfs in FHEM“

Autor: Alex Bartels

Erstellt: 2023-04-15

Voraussetzungen

Als Voraussetzung nehme ich eine laufende FHEM-Umgebung mit aktiviertem MQTT2-Server an. Zusätzlich gehe ich davon aus, dass der AMIS-Leser bereits in das lokale Netzwerk integriert wurde.

Bei mir läuft FHEM auf einem Raspberry PI 4.

Einstellungen AMIS-Leser

Im Webinterface des AMIS-Lesers wechselt man über das Hamburger-Menü in den Bereich „MQTT“. Dort ist der MQTT Broker, Port und die Client ID anzugeben.

Hier:

MQTT Broker: 192.168.2.101

MQTT Port: 1883

MQTT Client ID: AMIS_LESER

Dies sind die IP-Adresse des FHEM-Servers und der Port des MQTT2-Servers in FHEM.

The screenshot shows the MQTT configuration page in the AMIS-Reader web interface. The page title is "MQTT" with the subtitle "Einen MQTT broker im lokalen Netzwerk definieren." The settings are as follows:

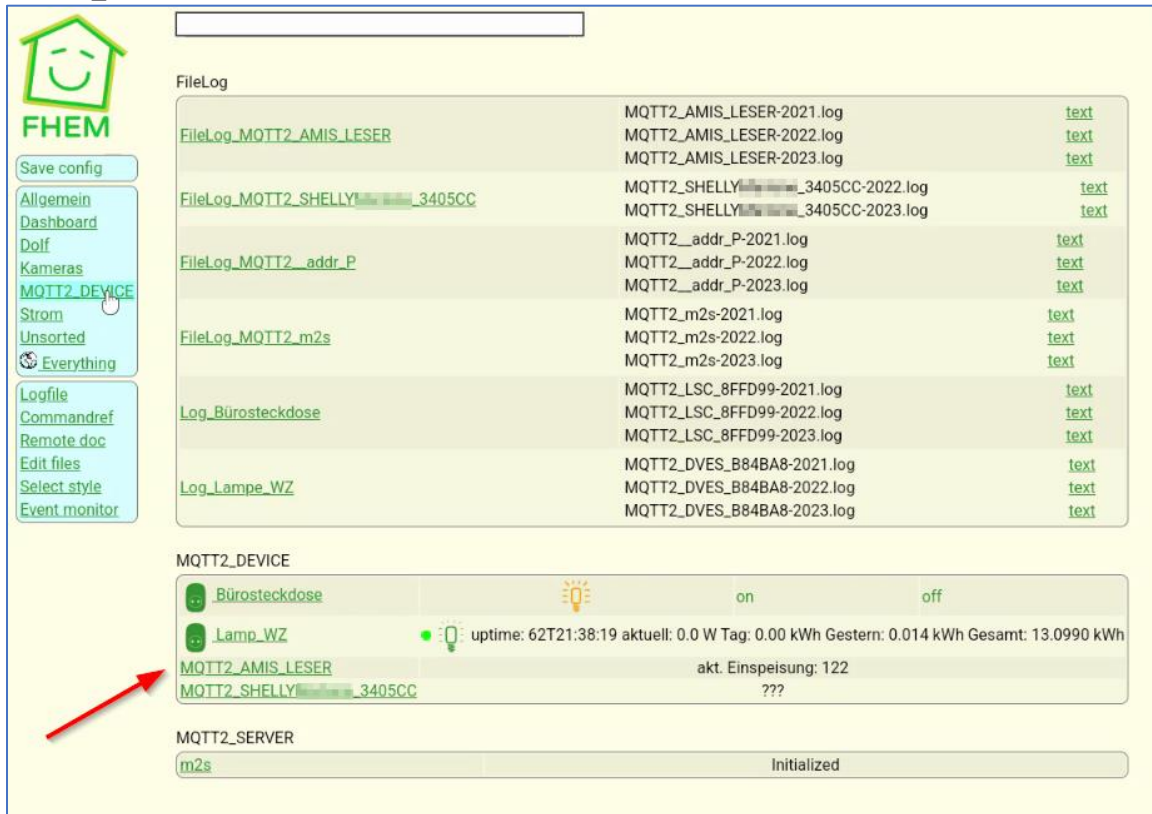
- MQTT aktiv: JA
- MQTT Broker: 192.168.2.101
- MQTT Port: 1883
- MQTT User: Frei lassen wenn kein Benutzer definiert ist
- MQTT Passwort: Frei lassen wenn kein Passwort definiert ist
- MQTT Client ID: AMIS_LESER
- MQTT QoS: 0: At most once
- MQTT Retain: JA
- MQTT Keep Alive: 60 sec
- MQTT publish: amis/out
- MQTT Last Will: (empty)

Buttons: Übernehmen (Save), Neustart (Restart)

Einstellungen FHEM

Grundsetup

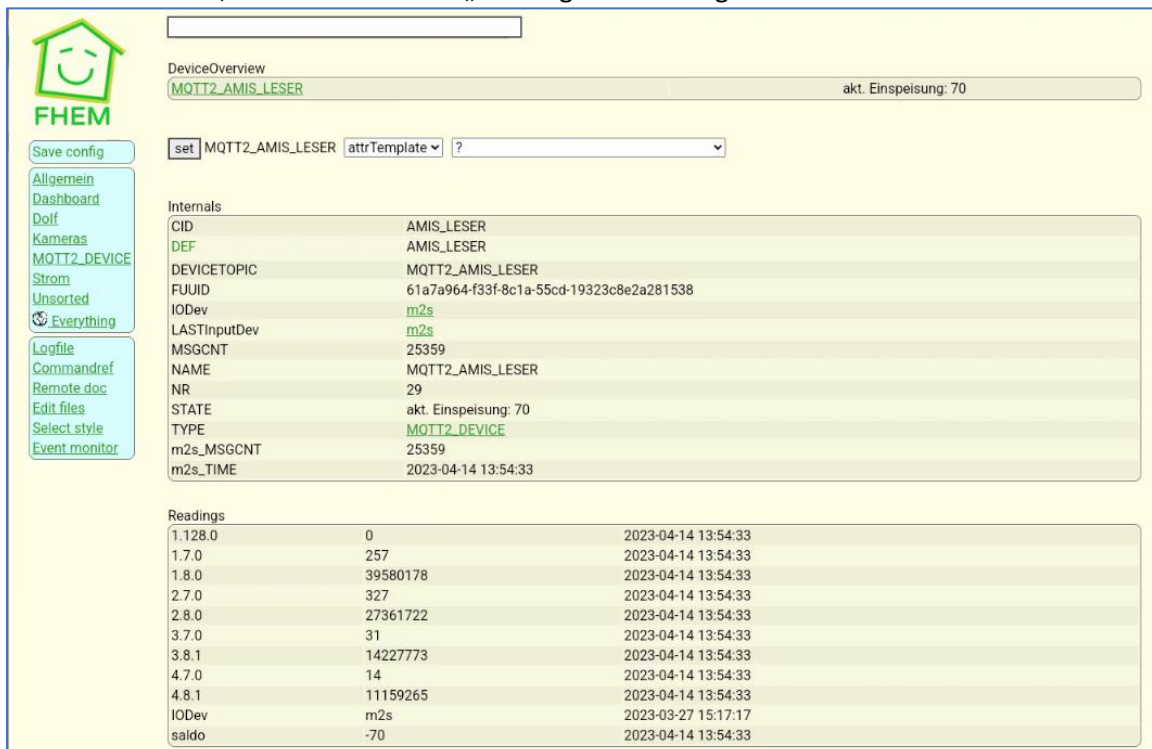
Ist der MQTT2-Server aktiv, so erscheint der AMIS-Leser automatisch nach kurzer Zeit unter den MQTT2_Devices in FHEM:



The screenshot shows the FHEM main configuration page. On the left is a navigation menu with options like 'Allgemein', 'Dashboard', 'Dolf', 'Kameras', 'MQTT2_DEVICE', 'Strom', 'Unsorted', 'Everything', 'Logfile', 'Commandref', 'Remote doc', 'Edit files', 'Select style', and 'Event monitor'. The 'MQTT2_DEVICE' menu item is highlighted with a red arrow. The main content area is divided into sections: 'FileLog' (listing various log files for MQTT2 devices), 'MQTT2_DEVICE' (showing a list of devices including 'Bürosteckdose', 'Lamp_WZ', 'MQTT2_AMIS_LESER', and 'MQTT2_SHELLY..._3405CC'), and 'MQTT2_SERVER' (showing 'm2s' as 'Initialized').

Bei mir ist dies das Device „MQTT2_AMIS_LESER“.

Öffnet man dieses, so sieht man unter „Readings“ alle verfügbaren Attribute:



The screenshot shows the FHEM device overview page for the 'MQTT2_AMIS_LESER' device. The page displays the device name and its current state ('akt. Einspeisung: 70'). Below this, there is a 'set' dropdown menu and a table of 'Internals' (device attributes). The 'Readings' section shows a list of device readings with their values and timestamps.

Attribute	Value	Timestamp
CID	AMIS_LESER	
DEF	AMIS_LESER	
DEVICETOPIC	MQTT2_AMIS_LESER	
FUUIID	61a7a964-f33f-8c1a-55cd-19323c8e2a281538	
IODev	m2s	
LASTInputDev	m2s	
MSGCNT	25359	
NAME	MQTT2_AMIS_LESER	
NR	29	
STATE	akt. Einspeisung: 70	
TYPE	MQTT2_DEVICE	
m2s_MSGCNT	25359	
m2s_TIME	2023-04-14 13:54:33	

Reading	Value	Timestamp
1.128.0	0	2023-04-14 13:54:33
1.7.0	257	2023-04-14 13:54:33
1.8.0	39580178	2023-04-14 13:54:33
2.7.0	327	2023-04-14 13:54:33
2.8.0	27361722	2023-04-14 13:54:33
3.7.0	31	2023-04-14 13:54:33
3.8.1	14227773	2023-04-14 13:54:33
4.7.0	14	2023-04-14 13:54:33
4.8.1	11159265	2023-04-14 13:54:33
IODev	m2s	2023-03-27 15:17:17
saldo	-70	2023-04-14 13:54:33

Ich habe noch zusätzlich das Attribut "stateFormat" ergänzt, um den aktuellen Wert kombiniert mit Text angezeigt zu bekommen:

4.8.1	11159265	2023-04-14 13:54:33
IODev	m2s	2023-03-27 15:17:17
saldo	-70	2023-04-14 13:54:33

attr MQTT2_AMIS_LESER

Attributes

readingList	AMIS_LESER:amis/out:.*{json2nameValue(\$EVENT)}	deleteattr
room	Dashboard.MQTT2_DEVICE,Strom	deleteattr
stateFormat	<pre>{if (ReadingsVal(\$name, "2.7.0", "none") - ReadingsVal(\$name, "1.7.0", "none") > 0){ "akt. Einspeisung: " . (ReadingsVal(\$name, "2.7.0", "none") - ReadingsVal(\$name, "1.7.0", "none")); } else { "akt. Verbrauch: " . (ReadingsVal(\$name, "2.7.0", "none") - ReadingsVal(\$name, "1.7.0", "none")); } }</pre>	deleteattr

Code:

```
{if (ReadingsVal($name, "2.7.0", "none") - ReadingsVal($name, "1.7.0", "none") > 0){
"akt. Einspeisung: " . (ReadingsVal($name, "2.7.0", "none") - ReadingsVal($name, "1.7.0", "none"));
} else {
"akt. Verbrauch: " . (ReadingsVal($name, "2.7.0", "none") - ReadingsVal($name, "1.7.0", "none"));
}
}
```

Damit wird folgendes in der Übersicht von FHEM angezeigt (hier inkl. des Outputs meines Kostal-Wechselrichters):

KOSTALPIKO
Kostal W: 637 - Einspeisen MPP
MQTT2_DEVICE
MQTT2_AMIS_LESER akt. Verbrauch: -214

Optional: Visualisierung in Diagrammen

Mit den jetzt vorliegenden Daten kann man zum Beispiel SVG Plots erzeugen:



Hier sieht man z.B. die Plots des Wechselrichters und des AMIS-Zählers über einen Tag verteilt.

Die Diagramm-Erstellung beschreibe ich hier nicht, dafür gibt es bereits entsprechende HOWTOs.

Optional: Aktionen mittels DOIF

Natürlich kann man mit den aktuellen AMIS-Leser-Werten auch Aktionen ausführen, wie z.B. das Aktivieren/Deaktivieren einer Steckdose oder eine einfache Mail-Versendung.

Versendung einer Mail

In diesem Beispiel wird eine Mail versendet, wenn zwischen Einspeisen und Verbrauch gewechselt wird.

Ich verwende dazu aus den Readings das Attribut „saldo“. Dieses gibt den aktuellen Verbrauch bzw. Einspeisung wieder.

DeviceOverview

di_amis_mail cmd_2

set di_amis_mail checkall

Internals

CFGFN (MQTT2_AMIS_LESER.saldo) < 0) (MYHELPER_Sendmail('#####@web.de', 'FHEM: Einspeisung', 'Hurra, wir speisen ein!')) DOELSE (MYHELPER_Sendmail('#####@web.de', 'FHEM: Verbrauch', 'Mist, wir verbrauchen zu viel Strom!'))

FUJID 6439330c-f33f-8c1a-a16b-0c1ab1d0328dd6ff

MODEL FHEM

NAME di_amis_mail

NOTIFYDEV MQTT2_AMIS_LESER.global

NR 27906

NTFY_ORDER 50-di_amis_mail

STATE cmd_2

TYPE DOIF

VERSION 24905 2021-09-01 18:35:54

Readings

Device	MQTT2_AMIS_LESER	2023-04-14 14:04:33
cmd	2	2023-04-14 13:55:36
cmd_event	MQTT2_AMIS_LESER	2023-04-14 13:55:36
cmd_nr	2	2023-04-14 13:55:36
e_MQTT2_AMIS_LESER_saldo	396	2023-04-14 14:04:33
mode	enabled	2023-04-14 13:44:16
state	cmd_2	2023-04-14 13:55:36

attr di_amis_mail room

Attributes

cmdpause	900.900	deleteattr
room	Dolfl	deleteattr

Probably associated with

MQTT2_AMIS_LESER	akt. Verbrauch: -396	MQTT2_DEVICE
------------------	----------------------	--------------

MYHELPER_Sendmail ist eine Funktion, welche ich für FHEM geschrieben habe. Da kann man sicherlich auch die allg. Mail-Funktionen verwenden.

Befehl für die Anlage von DOIF:

Code:

```
define di_amis_mail DOIF ([MQTT2_AMIS_LESER:saldo] < 0)
{MYHELPER_Sendmail('YOURMAILADDRESS', 'FHEM: Einspeisung', 'Hurra, wir
speisen ein. Ertrag: ' . (-1 * [MQTT2_AMIS_LESER:saldo]) . ' Watt' )}
DOELSE {MYHELPER_Sendmail('YOURMAILADDRESS', 'FHEM: Verbrauch', 'Mist, wir
verbrauchen zu viel Strom. Verbrauch: ' . (-1 * [MQTT2_AMIS_LESER:saldo]) .
' Watt )}
```

Zusätzlich habe ich in der FHEM-Oberfläche noch das Attribut cmdpause mit dem Wert "900:900" hinzugefügt. Das bedeutet, dass erst nach 15 Min. die nächste Statusänderung eine Mail auslöst.

Steuerung einer Steckdose

Die Steckdose muss natürlich in FHEM eingebunden sein. Dies ist bei mir ebenfalls per MQTT2 erfolgt. Bei mir ist es die Steckdose mit dem Namen „MQTT2_LSC_8FFD99“.

Dann legen wir wieder ein DOIF an, um eine Steckdose zu aktivieren, wenn mehr als 1000W eingespeist werden:

Code:

```
define di_amis_steckdose1 DOIF ([MQTT2_AMIS_LESER:saldo] < 1000) (Set
MQTT2_LSC_8FFD99 on) DOELSE (Set MQTT2_LSC_8FFD99 off)
```

Hier sollte auch entweder das Attribut cmdpause oder wait verwendet werden, um das sofortige Hin- und Herschalten der Steckdose zu verhindern, falls ein größerer Verbraucher an der Steckdose angeschlossen ist.

Oder man definiert eine 2. DOIF Aktion, um die Steckdose bei Bedarf wieder zu deaktivieren.