SonosFIFO 4.0 – Zentrale Sonos-Sprachansagen

Überblick

SonosFIFO verwaltet MP3-Sprachansagen in ioBroker zentral. Die Automationen (z. B. Blockly) lösen lediglich einen Profilnamen aus. Die Verwaltung kümmert sich um Lautsprecher, MP3, Lautstärke, optionale Ankündigung, Priorität, Nachtmodus und Warteschlange.

- **Priorität:** Alarm > Warnung > Info. Bei gleicher Priorität gilt die Eingangsreihenfolge (FIFO).
- **Keine Unterbrechung:** Eine laufende Ansage inklusive Ankündigungston wird vollständig zu Ende gespielt.
- **Nachtmodus:** Info, Warnung und nicht freigegebene Alarme bleiben stumm. Ausschließlich entsprechend konfigurierte kritische Alarmprofile dürfen nachts spielen.
- **Warteschlange:** Aufträge werden vor der Annahmebestätigung dauerhaft in ioBroker gespeichert.
- **Bedienung:** Per Profil-Taster im Objektbaum oder per Blockly-Nachricht. Schnelle doppelte Tastendrücke werden entprellt.
- **Zentrale Profile:** Neue Sprachansagen werden ausschließlich im separaten Profilskript eingetragen.

System und Speicherorte

Komponente	Konfiguration / Speicherort
ioBroker	JavaScript-Adapter `javascript.0`
Sonos-HTTP-API	`http://192.168.10.11:5005`
Sonos-HTTP-API Dienst	`sonosapi.service`
Installationsverzeichnis der API	`/home/iobroker/node-sonos-http-api-current/`
Zentrale MP3-Ablage (ioBroker-Upload)	`/opt/iobroker/iobroker-data/files/0_userdata.0/clips/`
Symbolischer Link der Sonos-API	`/home/iobroker/node-sonos-http-api-current/static/clips`
Alle Zustände	`0_userdata.0.Sprachansagen`

Wichtig: Der ioBroker-Ordner ist der einzige MP3-Speicherort. Neue MP3-Dateien über die ioBroker-Oberfläche nach `0_userdata.0 / clips` hochladen. Nicht zusätzlich in das Sonos-Installationsverzeichnis kopieren.

Der folgende symbolische Link verbindet beide Systeme:

```
/home/iobroker/node-sonos-http-api-current/static/clips
  -> /opt/iobroker/iobroker-data/files/0_userdata.0/clips
```

Link und Webzugriff prüfen:

```
ls -ld /home/iobroker/node-sonos-http-api-current/static/clips
readlink -f /home/iobroker/node-sonos-http-api-current/static/clips
curl -I http://127.0.0.1:5005/clips/gong.mp3
sudo systemctl status sonosapi --no-pager
```

Erwartet: `readlink -f` zeigt den ioBroker-MP3-Ordner; der HTTP-Aufruf liefert `200 OK`.

Neuinstallation des Links (nur falls noch nicht vorhanden): Das vorhandene Verzeichnis vorher sichern und sicherstellen, dass die MP3-Dateien im ioBroker-Ordner liegen. Niemals ein befülltes `static/clips` blind löschen oder überschreiben.

```
SRC=/opt/iobroker/iobroker-data/files/0_userdata.0/clips
DST=/home/iobroker/node-sonos-http-api-current/static/clips
# Nur ausführen, wenn DST kein existierendes Verzeichnis/Link mehr ist:
sudo ln -s "$SRC" "$DST"
```

Installation und Inbetriebnahme

Voraussetzung

- Sonos-HTTP-API läuft auf Port 5005, Sonos-Zone `Flur` ist erreichbar.
- Die Dateien `gong.mp3`, `waschmaschine_fertig.mp3`, `Windalarm.mp3` und `Wasseralarm_HWR.mp3` liegen im zentralen ioBroker-Clip-Ordner.
- Im JavaScript-Adapter kann auf diesen Ordner lesend zugegriffen werden.
- Es darf **nur ein** SonosFIFO-Controller aktiv sein. Bestehende direkte `/clip/`-Aufrufe können sonst die Wiedergabe überlagern.

Die zwei Skripte

ioBroker-Skript	Inhalt
`script.js.common.SonosFIFO_Profile`	Datei `SonosFIFO_Profile.js`; Kategorien und sämtliche MP3-Profile.
`script.js.common.SonosFIFO`	Datei `SonosFIFO_Controller.js`; Wiedergabe, Prioritäten, Nachtmodus, Status, Persistenz und Buttons.

1. In ioBroker unter **Skripte** die zwei JavaScript-Skripte anlegen (bzw. bestehende Skripte aktualisieren).
2. **Zuerst** das Profilskript aktivieren. Log: `[SonosFIFO Profile] 4 Profile veroeffentlicht`.
3. **Danach** den Controller aktivieren. Log: `[SonosFIFO] Profile geladen: 4` und `[SonosFIFO] v4.0 bereit`.
4. Unter `0_userdata.0.Sprachansagen.Status` müssen `Konfigurationsfehler` und `LetzterFehler` leer sein. Der Nachtmodus muss einen plausiblen Wert zeigen.
5. Unter `0_userdata.0.Sprachansagen.Ausloeser` erscheinen die Profil-Taster `WaschmaschineFertig`, `Gong`, `Windalarm` und `WasseralarmHWR`.

Aktualisierung einer bestehenden Installation

1. Laufende Ansage beenden lassen. Vor dem Update sollten `Status.Beschaeftigt=false` und `Status.WartendGesamt=0` gelten.
2. Die beiden bisherigen Skripte sowie `Intern.SpeicherJSON` sichern/exportieren.
3. Bestehenden Controller stoppen; **keine zweite Controller-Instanz** anlegen.
4. Vorhandenes Profilskript mit `SonosFIFO_Profile.js` aktualisieren, starten und Veröffentlichung im Log prüfen.
5. Vorhandenen Controller mit `SonosFIFO_Controller.js` aktualisieren und starten.
6. Einstellungen und Warteschlangen-Speicher werden beibehalten; bei Bedarf gespeicherte Werte unter `Einstellungen` kontrollieren.
7. Umbenannte/entfernte Profile können alte Auslöser-Datenpunkte hinterlassen. Diese werden nicht mehr abonniert und dürfen **nach Kontrolle** manuell aus dem Objektbaum entfernt werden. Der Controller löscht niemals automatisch fremde Datenpunkte.

Das interne Speicherformat wird bewusst beibehalten, damit die Warteschlange beim Controller-Austausch nicht zurückgesetzt wird. `Intern.SpeicherJSON` nicht löschen oder manuell bearbeiten.

Profile pflegen

Einzige Konfigurationsquelle: `SonosFIFO_Profile.js`. Änderungen in den Objekten `Konfiguration.ProfileJSON` sind nicht der richtige Weg: Das Profilskript überschreibt diese beim nächsten Start.

Standardwerte je Kategorie

```
const KATEGORIEN = {
    Info:    { raum: 'Flur', lautstaerke: 70, gong: true,  gongLautstaerke:
55 },
    Warnung: { raum: 'Flur', lautstaerke: 85, gong: true,  gongLautstaerke:
70 },
    Alarm:   { raum: 'Flur', lautstaerke: 95, gong: false, gongLautstaerke:
70 }
};
```

Profil-Beispiele der ausgelieferten Konfiguration

Profilname	Kategorie	MP3	Nachtfreigabe
`WaschmaschineFertig`	Info	`waschmaschine_fertig.mp3`	Nein
`Gong`	Info	`gong.mp3`	Nein
`Windalarm`	Warnung	`Windalarm.mp3`	Nein
`WasseralarmHWR`	Alarm	`Wasseralarm_HWR.mp3`	**Ja (kritischer Alarm)**

`WasseralarmHWR` ist bewusst nachts freigegeben. Falls dieser Alarm nachts **nicht** spielen soll, im Profilskript `nachtsErlaubt: false` setzen. Die Alarm-Kategorie allein erteilt keine Nachtfreigabe.

Neues Profil hinzufügen:

```
TrocknerFertig: {
    prioritaet: 'Info',
    datei: 'trockner_fertig.mp3',
    lautstaerke: 75,
    gong: true
},
```

1. MP3 über ioBroker hochladen; exakt auf Groß-/Kleinschreibung achten.
2. Profil in `PROFILE` ergänzen. Erlaubt sind Profilnamen mit Buchstabe am Anfang, danach nur Buchstaben A–Z, Ziffern und Unterstriche.
3. Zulässige Profilparameter: `prioritaet` (Info/Warnung/Alarm), `datei`, optional `raum`, `lautstaerke` (0–100), `gong` (wahr/falsch), `gongLautstaerke` (0–100), `ankuendigungDatei` (MP3 nur für dieses Profil), `nachtsErlaubt` (ausschließlich bei Alarm relevant).
4. Profilskript neu starten. Der Controller übernimmt die Konfiguration automatisch und erstellt den passenden Taster. **Kein Controller-Neustart nötig.**

Ankündigungstöne

Der Profilparameter `gong: true` bedeutet: Vor der eigentlichen Sprachansage einen Ankündigungston spielen. **Dieser Ton muss nicht `gong.mp3` heißen.**

Einstellung unter `0_userdata.0.Sprachansagen.Einstellungen`	**Standarddatei**	**Verwendung**
`AnkuendigungInfo`	`gong.mp3`	Ankündigung für Info-Profile
`AnkuendigungWarnung`	`gong.mp3`	Ankündigung für Warnungsprofile
`AnkuendigungAlarm`	`gong.mp3`	Ankündigung für Alarmprofile (sofern dort `gong: true`)

Beispiel: `AnkuendigungWarnung = warnung.mp3`. Die Datei muss vorher im ioBroker-Clip-Ordner liegen. Ein Profil kann mit dem Parameter `ankuendigungDatei` (z. B. `sonderton.mp3`) eine abweichende MP3 auswählen. Die Ankündigungslautstärke ist separat einstellbar.

Ansagen auslösen

1. Profil-Taster im ioBroker-Objektbaum

Datenpunkt `0_userdata.0.Sprachansagen.Ausloeser.<Profilname>` auf `true` setzen. Der Controller stellt ihn automatisch auf `false` zurück. Eingehende Tastendrücke werden unabhängig von ihrem ACK-Flag erkannt.

Um versehentliche Doppelklicks zu vermeiden, werden weitere Tastendrücke **desselben Profils** innerhalb von `Einstellungen.TasterSperreMs` (Standard 1500 ms) ignoriert. Jeder zulässige Tastendruck führt zu einem neuen Auftrag, auch wenn bereits eine Ansage läuft.

2. Blockly: empfohlene Ansteuerung

Blockly-Block **„Nachricht an anderes Skript“**:

- JavaScript-Instanz: `javascript.0`
- Skript: `script.js.common.SonosFIFO`
- Nachrichtenname: `ansage`
- Daten: Text `WaschmaschineFertig`

Individuelle Lautstärken, ohne das Profil zu verändern:

- `WaschmaschineFertig|75` → Ansage mit 75 %
- `WaschmaschineFertig|75|45` → Ansage 75 %, Ankündigung 45 %

Die Antwort auf `messageTo` (optional „Mit Ergebnissen“) meldet, ob ein Auftrag **dauerhaft angenommen** oder z. B. wegen Nachtmodus verworfen wurde. Sie meldet **nicht**, dass die MP3 bereits zu Ende abgespielt wurde.

3. Weitere Schnittstellen

- `Eingang.Profil`: einen Profilnamen oder ein `Profil|Lautstärke|Gonglautstärke`-Kommando schreiben.
- `Eingang.AnfrageJSON`: Parameterobjekt als JSON schreiben, z. B. `{„profil“:„WaschmaschineFertig“,„lautstaerke“:75}`.
- Direkte `messageTo`-Parameterobjekte sind ebenfalls möglich, z. B. `{profil:'WaschmaschineFertig', lautstaerke:75, raum:'Flur'}`.

Für sicherheitskritische Automationen **messageTo mit Rückmeldung verwenden**, nicht einen kurzen Tasterimpuls. Anfragen, die während eines Controller-Ausfalls gar nicht ankommen, können nicht gespeichert werden. Doppelte Alarmierung über einen unabhängigen Weg empfohlen.

Priorität und Nachtmodus

- Die jeweils laufende Ansage wird **niemals unterbrochen**; nach ihrem Ende entscheidet `Alarm > Warnung > Info` über die nächste. Innerhalb derselben Kategorie bleibt die Reihenfolge erhalten.
- `Einstellungen.NachtmodusQuelle` enthält die Quell-Datenpunkt-ID, standardmäßig `knx.0.Zentral.Tag=0_Nacht=1.Tag=0_Nacht=1_Schlafzimmer`.
- Bei der Quelle gilt `0 = Tag` und `1 = Nacht`; auch boolesche Werte werden erkannt.
- **Nachts** werden Info-, Warnungs- und nicht ausdrücklich freigegebene Alarmaufträge beim Eingang unterdrückt; bereits wartende unzulässige Aufträge werden beim Nachtwechsel verworfen, **nicht** morgens nachgeholt.
- Eine schon laufende Ansage darf zu Ende spielen. Nur als kritisch konfigurierte Alarmprofile (`nachtsErlaubt: true`) dürfen nachts abgespielt werden.
- Ist der Nachtstatus unbekannt/ungültig, wird vorsichtshalber Nacht angenommen.
- `Einstellungen.NachtmodusTest = true` **erzwingt Nacht zu Testzwecken**; `false` aktiviert wieder die echte Quelle. Diese Einstellung niemals versehentlich aktiviert lassen.

Datenpunkte – Übersicht

Alle folgenden Namen liegen unter `0_userdata.0.Sprachansagen`.

Ausloeser / Eingang / Konfiguration

Gruppe	Datenpunkte	Zweck
`Ausloeser`	`<Profilname>`	Automatisch erzeugte Profil-Taster (boolean); einmal auf `true` setzen
`Eingang`	`Profil`, `AnfrageJSON`	Alternative Eingänge für Texte bzw. JSON-Aufträge
`Konfiguration`	`ProfileJSON`	Durch Profilskript veröffentlichte Konfiguration; nicht manuell bearbeiten

Steuerung

Datenpunkt	Funktion
`Pausiert`	`true`: nach laufender Ansage keine weiteren Aufträge starten; `false`: fortsetzen
`WarteschlangeLeeren`	Alle **wartenden** Aufträge entfernen; laufende Wiedergabe beenden lassen
`FehlerQuittieren`	Fehlerzustand aufheben; ein möglicherweise nicht vollständig gespielter aktiver Auftrag startet von vorne

Einstellungen

Datenpunkt	Bedeutung
`AnkuendigungInfo` / `AnkuendigungWarnung` / `AnkuendigungAlarm`	MP3 für die jeweilige Kategorie, Standard `gong.mp3`
`TasterSperreMs`	Entprellung der Profil-Taster, Standard `1500` ms (0–10000)
`GongPauseMs`	Pause zwischen Ankündigung und Ansage, Standard `1000` ms
`ZwischenAuftraegenMs`	Abstand zwischen Aufträgen, Standard `800` ms
`HTTPTimeoutMs`	HTTP-Timeout pro Sonos-Clip, Standard `90000` ms
`NachtmodusQuelle`	Quell-Datenpunkt-ID für Tag/Nacht
`NachtmodusTest`	`true` = Nacht erzwingen; `false` = echten Zustand lesen

Status

Datenpunkte	Bedeutung
`Zustand`, `Beschaeftigt`, `Phase`	Bereitschaft, Fehler/Pause und Fortschritt der aktiven Ansage
`Profil`, `Datei`, `Raum`, `Prioritaet`	Aktuell bearbeiteter Auftrag
`WartendGesamt`, `WartendAlarm`, `WartendWarnung`, `WartendInfo`	Anzahl wartender Aufträge, jeweils ohne aktiven Auftrag
`WarteschlangeJSON`	Nach Priorität sortierte, vorgesehene Reihenfolge

Datenpunkte	Bedeutung
ProfileJSON	Effektiv aufgelöste Profile (inkl. Kategorie-Standardwerte)
Konfigurationsfehler, LetzterFehler	Fehler bei Profilübernahme beziehungsweise Wiedergabe/Speicherung
LetzteAnsage, LetzteAnsageZeit	Letzte abgeschlossene Ansage
Nachtmodus, NachtsperreAktiv, NachtmodusQuelle	Nachtzustand und verwendete Quelle
NachtsUnterdrueckt, LetzteUnterdrueckung	Statistik verworfener Aufträge
LetzterTaster, TasterIgnoriert	Bedienung und Doppelklick-Entprellung

Intern

Intern.SpeicherJSON enthält die dauerhaft gespeicherte Warteschlange einschließlich ggf. aktivem Auftrag, Pause-/Fehlerstatus und Zählern. **Nicht manuell ändern oder löschen.**

Bedienung und Wartung

- **MP3 hinzufügen:** Über ioBroker nach 0_userdata.0 / clips hochladen; Dateinamen exakt in Profil eintragen.
- **Profil ändern/anlegen:** Nur SonosFIFO_Profile bearbeiten und neu starten.
- **Ankündigung für eine Kategorie wechseln:** Entsprechenden Datenpunkt unter Einstellungen ändern; keine Skriptänderung nötig.
- **Warteschlange stoppen:** Steuerung.Pausiert = true; laufende Ansage wird noch beendet.
- **Warteschlange leeren:** Steuerung.WarteschlangeLeeren = true; nur wartende Aufträge werden entfernt.
- **Wiedergabefehler:** Status.LetzterFehler und Sonos-API prüfen, danach Steuerung.FehlerQuittieren auslösen.
- **Kontrolle:** Status.WarteschlangeJSON, Status.WartendGesamt und die ioBroker-Logmeldungen verwenden.
- **Nach Sonos-API-Update:** Symbolischen Link und HTTP-Abruf erneut prüfen.
- **Sicherung:** Beide Skripte und den Datenpunkt Intern.SpeicherJSON sichern. ioBroker-/Proxmox-Backup einschließen.

Kurzer Funktionstest

Achtung: Je nach Profil können Warnungen/Alarme laut abgespielt werden (Kategorie-Standards 85/95 %). Zu Testzwecken kann aus Blockly vorübergehend eine niedrigere Lautstärke übergeben werden, z. B. Windalarm|35.

1. Status.Konfigurationsfehler und Status.LetzterFehler müssen leer sein; Steuerung.Pausiert=false, Nachtstatus beachten.
2. Ausloeser.Gong auf true setzen. Erwartung: gong.mp3 auf Flur, automatische Rückstellung auf false.
3. Für Priorität: tagsüber Steuerung.Pausiert=true, anschließend per Blockly

`WaschmaschineFertig`, `Windalarm` und `WasseralarmHWR` einreihen. `WarteschlangeJSON` muss Alarm vor Warnung vor Info anzeigen.
4. `Steuerung.Pausiert=false`: Alle Aufträge werden in Prioritätsreihenfolge vollständig abgearbeitet.
5. Für Nacht: `Einstellungen.NachtmodusTest=true` (nur bewusst zum Testen!). Info/Warnung werden unterdrückt; `WasseralarmHWR` ist freigegeben. Danach `NachtmodusTest=false`.
6. Zum Schluss `Status.WartendGesamt=0` und `Status.Zustand=Bereit` kontrollieren.

Sicherheits- und Zuverlässigkeitshinweise

- **Speicherung:** Ein angenommener Auftrag wird vor der Bestätigung gespeichert. Nach unerwartetem Neustart können offene Aufträge erneut abgespielt werden; eine Doppelansage ist möglich.
- **Ausfall des Controllers:** Nicht zugestellte Taster-/Blockly-Auslöser während des Ausfalls können nicht nachträglich rekonstruiert werden. Für kritische Alarmmeldungen zusätzliche unabhängige Alarmwege vorsehen.
- **Nachtmodus:** Absichtlich unterdrückte Aufträge werden verworfen und nicht nachträglich abgespielt.
- **Sonos-Wiedergabe:** Nicht parallel an denselben Lautsprechern direkte Sonos-`clip`-Aufrufe oder einen zweiten FIFO-Controller verwenden.
- **Absturz-Wiederanlauf:** Nach einem vorzeitig beendeten aktiven Auftrag wartet der Controller 30 Sekunden und reiht den Auftrag erneut ein; vor erneuter Wiedergabe auf Sonos-Status achten.
- **Sicherheit:** Sonos-Sprachansagen ersetzen weder eine Sirene noch Alarmmeldungen per Handy für sicherheitskritische Ereignisse.

From:
https://wiki.mahlen.eu/ - **Smart-Home Wiki**

Permanent link:
https://wiki.mahlen.eu/doku.php?id=iobroker:iobroker_scripte:sprachansagen_tts

Last update: **12.10.2026**