

# Pi Presents

**Ein interaktives Multimedia-Anzeige- und Animationswerkzeug  
für Museen, Besucherzentren und mehr...**

**Für den Raspberry PI  
von  
Ken Thompson (KenT2)**

**<http://pipresents.wordpress.com>**

Handbuch  
Deutsche Fassung  
Peter Vasen (ptv)  
<http://www.web-echo.de>  
März 2026

## Inhaltsverzeichnis

1	Einführung.....	5
1.1	Danksagung.....	7
2	Installation.....	7
3	Beispiele zum Ausprobieren.....	7
3.1	Pi Presents beenden.....	13
4	Der Pi Presents Profil-Editor.....	13
4.1	Benutzung des Editors.....	13
4.1.1	Die Programmfenster.....	13
4.1.2	Profil-Menü.....	14
4.1.3	Show Menu.....	14
4.1.4	Medialist Menu.....	15
4.1.5	Track Menu.....	15
4.1.6	OSC Menü.....	15
4.1.7	Tools Menu.....	16
4.1.8	Options Menu.....	16
4.1.9	Editor Befehlszeilen-Optionen.....	16
4.2	Profile erstellen.....	16
4.2.1	Profile portieren.....	17
4.2.2	Profile auf SD-Karte.....	17
4.2.3	Editor auf einem Windows PC.....	18
4.2.4	Profile auf USB-Stick.....	18
4.2.5	Profile für Monitore mit unterschiedlicher Pixelgröße erstellen.....	18
5	Die Komponenten in Pi Presents.....	19
5.1	Einführung.....	19
5.2	Shows.....	21
5.2.1	Mediashow und Artmediashow.....	21
5.2.1.1	Steuerungen und Befehle.....	22
5.2.1.2	Triggers.....	24
5.2.1.2.1	Trigger for Start.....	25
5.2.1.2.2	Trigger for End.....	25
5.2.1.2.3	Trigger for Next.....	25
5.2.1.3	Felder.....	25
5.2.2	Menü.....	33
5.2.2.1	Steuerungen und Befehle.....	33
5.2.2.2	Felder.....	33
5.2.3	Liveshow und Artliveshow.....	35
5.2.4	Radiobuttonshow.....	37
5.2.4.1	Steuerungen und Befehle.....	37
5.2.4.2	Felder.....	38
5.2.5	Hyperlinkshow.....	39
5.2.5.1	Steuerungen und Befehle.....	40
5.2.5.2	Touchscreens und Softbuttons.....	42
5.2.5.3	Felder.....	42
5.2.6	Start Show.....	44
5.3	Medialisten.....	45
5.4	Tracks.....	45
5.4.1	Track-Speicherorte.....	46
5.4.2	Unbenannte und benannte Tracks.....	46
5.4.3	Show Track.....	46

5.4.4 Image Track.....	47
5.4.4.1 Image Window.....	50
5.4.4.2 HTML und einfacher Text.....	51
5.4.5 MPV Video Track.....	52
5.4.5.1 MPV Window.....	54
5.4.5.2 Video Playback.....	55
5.4.6 Audio Track.....	55
5.4.7 Chrome Web Tracks.....	57
5.4.7.1 Chrome Window.....	60
5.4.7.2 Browserbefehle.....	61
5.4.7.3 Titelzeile und Scrollbalken entfernen.....	61
5.4.8 Message Tracks.....	62
5.4.9 Show Track.....	64
5.4.10 Menü Track.....	65
6 Black Box Operation.....	68
6.1 Optionen auf der Befehlszeile.....	69
6.2 Ein Profil aufrufen.....	70
6.3 Ein Heimatverzeichnis aufrufen.....	70
6.4 Pi Presents mit GPIO steuern.....	71
6.5 Bildschirmpause ausschalten.....	72
6.6 Pi Presents beim Einschalten des Pi starten.....	72
6.7 Raspberry Pi per GPIO herunterfahren.....	73
7 Tracks und Shows steuern.....	74
7.1 Track-Steuerung in einzelnen Shows.....	74
7.2 Shows öffnen und schließen.....	75
7.3 Time of Day Planer.....	76
7.4 Konkurrierende Shows.....	77
7.4.1 Steuerung mit konkurrierenden Shows.....	78
7.4.2 Beschränkungen bei konkurrierenden Shows.....	78
8 Show Control Befehle.....	78
8.1 Befehle.....	79
8.2 Ereignisse zwischen Shows austauschen.....	79
8.3 Steuerung des Monitors.....	79
8.4 Show mit Ereignissen steuern (Control on Event).....	80
8.5 Beeps.....	80
8.6 Vibes.....	81
9 Displays und Touchscreens einrichten.....	81
9.1 Displays für den Raspberry Pi einrichten.....	81
9.1.1 Betrieb ohne Displays.....	82
9.2 Display auswählen.....	82
9.3 Touchscreens einrichten.....	82
9.3.1 Hintergrundbeleuchtung steuern.....	83
10 Touchscreens und Soft Buttons.....	83
10.1 Shows mit dem Touchscreen steuern.....	83
10.2 Klickbereiche.....	84
10.3 Soft Buttons.....	84
11 Audio einrichten.....	85
12 Laufzeitbefehle.....	85
13 Dynamische Inhalte in einer Liveshow.....	86
14 Zähler.....	86
15 Animationen steuern.....	88
15.1 Zeitprobleme bei Animationen.....	89
16 Input/Output (I/O) Plugins.....	89

16.1 Ein Standard-I/O-Gerät einrichten.....	90
16.2 Eingabe/Ausgabe: Treiber einrichten.....	91
16.2.1 Verknüpfen mit GPIO per pp_gpiozeroriver.py.....	91
16.2.2 Verbindungen mit GPIO per pp_gpiodriver.py.....	94
16.2.2.1 Eine besser reagierende Alternative – pp-pigpiodriver.py.....	95
16.2.3 Mit pp_inputdevicedriver.py Fernbedienung oder Keypad verbinden.....	96
16.2.4 Serielle Verbindung mit pp_serialdriver.py einrichten.....	97
16.2.5 Verbindung über I2C-Geräten mit pp_ic2driver.py.....	97
16.2.6 Tkinter: Tasten mit kbdriver.py einrichten.....	99
16.2.7 Erweitertes Verbinden mit Tastatur per pp_kbdriver_plus.py.....	100
16.2.8 Kurze Klänge mit pp_isplaydriver.py produzieren.....	101
16.2.9 Haptische Effekte per pp_DRV2605driver.py.....	102
16.2.9.1 Loop Modes.....	102
16.2.9.2 Audio zu Vibration mit dem Raspberry Pi.....	103
16.2.10 Touch- und Klickbereiche einrichten.....	103
16.2.11 Mit einem RFID-Tag-Reader über pp_pn532driver.py verbinden.....	104
16.3 I/O-Plugins schreiben.....	105
16.3.1 Einrichtung und Registrierung eines I/O-Plugins.....	106
16.3.2 Class.....	106
16.3.3 Methoden.....	106
16.3.4 Beispiel.....	109
17 Fernbedienung mit OSC.....	109
17.1 Senden und empfangen von Meldungen über OSC.....	110
17.2 OSC-Grundlagen.....	110
17.3 OSC Konfiguration.....	112
17.4 oscremote und oscmonitor.py.....	113
18 Track Plugins.....	114
19 Remote Management.....	116
19.1 Fernsteuerung (Remote) einrichten.....	117
19.2 Mit dem Manager arbeiten.....	119
19.3 Benutzung des Web Editors.....	120
20 Email Meldungen.....	121
20.1 Email-(Alarm)Meldungen einrichten.....	121
20.2 Mit eMail-(Alarm)Meldungen arbeiten.....	122
21 Anforderungen an die Hardware.....	122
22 Pi Presents updaten.....	122
22.1 Profile updaten.....	123
22.2 Pi Presents updaten.....	123
23 Fehlerkorrektur, Statistiken, Fehlerberichte und neue Programmwünsche.....	123
23.1 Statistiken erstellen.....	123
23.2 Fehler in Profilen beseitigen.....	124
24 Lösungen und bekannte Probleme.....	125

### Copyright-Hinweis

Dieses Handbuch und die Pi Presents Software stehen unter dem Copyright von Ken Thompson

Lizenzbedingungen siehe:

<https://github.com/KenT2/pipresents-gapless/blob/master/licence.md>

Raspberry Pi ist eine Marke der Raspberry Pi Foundation:

<http://www.raspberrypi.org>

# 1 Einführung

Pi Presents ist ein Werkzeug für die Erstellung und Anwendung interaktiver Darstellungen, mit denen auch Animationen gesteuert werden können. Das Programm wurde ursprünglich für Museen und Besucherzentren entwickelt. Mittlerweile wird es auch in Krankenhäusern, Shops, Kunstinstallationen, Bibliotheken und mehr eingesetzt.

Bevor es den Raspberry Pi gab, war der Kauf oder auch der Bau eines einfachen interaktiven Audioplayers teuer. Mit seiner Kombination aus Linux, GPIO und einer leistungsfähigen GPU ist der Raspberry Pi eine ideale Blackbox für Medieninhalte. Es fehlte lediglich noch ein Programm, das auch Nichtprogrammierern ermöglicht, die Power des Raspberry Pi für ihre Zwecke zu nutzen.

Es gibt eine ganze Reihe von digitalen Anzeigeprogrammen für den Raspberry Pi, die auf Diashows beschränkt sind oder ihre Inhalte lediglich aus dem Internet beziehen können. Anders bei Pi Presents: Mit einem leicht zu bedienenden Editor können mit einem Browser oder auf dem Raspberry Pi diverse interaktive Präsentationen erstellt werden.

Pi Presents unterstützt Showtypen, die üblicherweise in Museen vorkommen. Jeder Showtyp kann nach den eigenen Bedürfnissen konfiguriert werden.

Hier einige Varianten:

- Animation oder Präsentation von Sounds, Videos oder Diashows einer Ausstellung mit Buttons, Tastatur oder GPIO-Eingaben.
- GPIO-Outputs können während einer Medienpräsentation externe Geräte wie Lampen, Animationstechnik etc. steuern.
- Eine Diashow für Besucherzentren, die Bilder, Videos, Audiotracks oder Mitteilungen enthält. Wiederholungen können in einzustellenden Intervallen oder zu bestimmten Tageszeiten laufen.
- Diashows, die vom Besucher unterbrochen werden können und dabei ein Menü mit weiteren Inhalten zeigen.
- Eine Show für Kiosks, auf denen Inhalte durch Druck auf einen von mehreren Knöpfen gestartet werden können.
- Erlaubt die Konstruktion von Hyperlinkshows, wie sie auf Touchscreens in Museen laufen.
- Erstellt „Powerpoint“ ähnliche Multimedia-Präsentationen, bei denen der Ablauf der Folien manuell durch Buttons oder eine Tastatur gesteuert werden kann.
- Basisshow mit digitalen Inhalten für ferngesteuertes Abspielen. Aus dem Netzwerk lädt Pi Presents Bilder, Videos oder Audiotracks hinzu und fügt sie in die in Schleifen laufende Basisshow ein.
- Mehr-Fenster-Anzeige: Zeigt gleichzeitig unterschiedliche Medieninhalte auf einem einzigen Bildschirm.
- Startet eine Auswahl von Shows zu verschiedenen Uhr- und Tageszeiten so wie an bestimmten Tagen im Jahr.
- Erstellt ein Netzwerk von Pi-Presents-Anwendungen, kontrolliert per Open Sound Control (OSC). Kontrolle von Pi Presents mit Remotecomputern, Tablets oder Smartphones.
- Pi Presents verwalten und Profile mit einem Browser bearbeiten. (Dafür ist ein lokales Netzwerk erforderlich). Emails empfangen, mit denen Pi Presents den Status der laufenden Anwendungen anzeigt.
- Unterstützung mehrerer Sprachen. Alle Texte für den Zuschauer können konfiguriert werden.

Einmal eingerichtet, ist Pi Presents leicht zu bedienen und erfordert kein Netzwerk:

- Shows können mit einem leicht zu bedienenden Editor vorbereitet werden.
- Die Shows funktionieren auch auf einem Modell A oder A+ Raspberry Pi (Hardwareanforderungen beachten).
- Die SD-Card des Pi muss nach der Erstinstallation nicht mehr verändert werden. Alle Medien und Einstellungen können auf einem USB-Stick gespeichert werden. Die Installation geänderter/neuer Anwendungen kann so einfach sein, wie der Anschluss eines USB-Sticks.
- Für die Audioausgabe kann man zwischen USB und Bluetooth wählen und einzelne Spuren links, rechts oder über Stereo-Lautsprecher ausgeben. Mehrere Audiospuren können gleichzeitig auf verschiedenen Lautsprechern mit jeweils einstellbarer Lautstärke ausgegeben werden.

Pi Presents läuft in der Grundeinstellung als tastaturgesteuerte Desktop-Anwendung auf dem Raspberry Pi. Doch mit ein wenig Linux-Zauber, erklärt in diesem Handbuch, kann das Programm auch als Blackbox-Anwendung, gesteuert über GPIO oder ferngesteuert mit OSC, zum Laufen gebracht werden.

Die eingebauten Blackbox-Funktionen:

- Bildschirmpause und Mauszeiger können deaktiviert werden.
- Vollbildmodus ohne Fensterrahmen.
- vollständiger Betrieb ohne Tastatur, Maus oder Buttons (Ausnahme ist evtl. ein Shutdown-Button, wenn man nicht die Korruption der SD-Karte riskieren möchte).
- Blackbox-Steuerung durch Buttons, Infrarot-Fernbedienung oder andere digitale Eingangsquellen. Sie können ihre ganz spezielle Steuerung selbst einrichten.
- Mauskompatible Touchscreens werden unterstützt, wenn Treiber für den Raspberry Pi verfügbar sind. Alternativ sind Tastatur und Maus immer möglich. In Pi Presents können in Python 3 geschriebene Treiber für andere Arten von Eingabe und Ausgabe, wie RS232 und I2C, geladen werden.
- Automatischer Start, wenn der Raspberry Pi mit Strom versorgt wird.
- sicheres Abschalten ohne Tastatur oder Maus.
- automatisches Starten und Stoppen von Shows und das Herunterfahren des Raspberry Pi zu festgelegten Tageszeiten. Der Zeitplaner kann auf Tage in der Woche oder im Monat oder auf bestimmte Tage im Jahr eingestellt werden.
- Fernsteuerung von Pi Presents-Anwendungen durch einen anderen Raspberry Pi über Open Sound Control oder durch einen anderen PC, auf dem OSC funktioniert.

## 1.1 Danksagung

Davide Rosa für Reml – Remote Interface, ein Werkzeug für die Programmierung webbasierter, grafischer Oberflächen in Python.

<https://github.com/dddomodossola/remi>

Paolo Gurisatti for tk\_html\_widgets allowing html in text fields

[https://github.com/paolo-gurisatti/tk\\_html\\_widgets](https://github.com/paolo-gurisatti/tk_html_widgets)

Izzie Walton for python-mpv-jsonipc which allows reliable communication with MPV

<https://github.com/iwalton3/python-mpv-jsonipc>

Die vielen Aktivisten im Raspberry Pi-Forum und auf Pi-Webseiten, darunter StackOverflow, die mir, ohne es zu wissen, bei der Lösung etlicher technischer Probleme geholfen und Python beigebracht haben.

Bullets von <http://www.enterprise-dashboard.com/tag/red-green-yellow-alert/>

Icons von <http://www.fatcow.com/free-icons>

Buttons von <http://www.freewebsitebuttons.com>

Raspberry Pi Foundation für die erste erschwingliche Maschine, die nun auch 4K-Video- und Audio-Tracks abspielt und GPIO anbietet.

Als mich der Raspberry Pi mit Open Source Software zusammenbrachte, war ich erstaunt über die Zeit und den Eifer, womit so viele Menschen Software für andere Menschen programmieren. So fand ich es wichtig, das gleiche mit Pi Presents zu tun. Ich bin froh darüber, denn das Feedback der User und potenziellen Nutzer hat mir sehr viele Ideen für Erweiterungen von Pi Presents eingebracht. Es erübrigt sich, zu sagen, das ich über die viele Arbeit an Pi Presents keine Zeit fand, meine eigenen geplanten Projekte anzugehen. Aber eines Tages, da bin ich ganz sicher, werde ich es tun (nun habe ich drei :-).

## 2 Installation

Die Installationsanweisungen finden sich in der Datei README.md. Pi Presents erfordert Raspberry Pi OS Trixie oder später. Es sollte sich immer um die aktuelle Version der Raspberry-Pi-Stiftung handeln. Sie muss vom Pixel Desktop installiert und betrieben werden, sowie X11 benutzen.

Dieses Handbuch geht davon aus, dass der Benutzer pi ist, dessen Verzeichnisse in *home/pi* liegen. Pi Presents läuft aber auch unter anderen Benutzern.

## 3 Beispiele zum Ausprobieren

Download der Beispiele aus dem pipresents-kms-examples Github Repository wie in der README.md Datei beschrieben.

Anmerkung: Pi Presents muss innerhalb des PIXEL-Desktop gestartet werden (startx entweder per Befehl im Terminalfenster ausführen oder „Booten in den Desktop“ in der raspi,config wählen). Tippen Sie in einem Terminalfenster im Pi Presents-Verzeichnis:

```
python3 pipresents.py -p pp_mediashow_1p5
```

In einem kleinen Fenster startet eine Show in der Endlosschleife. Das kleine Fenster ist für Entwickler gedacht. Seine Größe kann durch Änderungen in `pipresents.py` bei Zeile 45 angepasst werden. Die Anzeige kann man mit der zugehörigen Fensterdekoration fast auf Bildschirmgröße bringen. Wenn die Show jedoch ohne Rahmen und komplett auf dem Vollbildschirm erscheinen soll, ist die Kommando-Option `-f` erforderlich und `-b` für die Unterdrückung des leeren Bildschirms.

```
Python3 pipresents.py -p pp_mediashow_1p5 -f -b
```

Das Mediashow-Beispiel und alle anderen Beispiele sind für eine Bildschirmauflösung von 1920x1080 geeignet. Bei Bildschirmen mit geringerer Auflösung liegen einige Textteile und Teile größerer Bilder außerhalb des Bildschirmrandes. Pi Presents passt die Auflösung nicht automatisch an die Bildschirmgröße an. Die Textgröße und -position kann im Pi Presents Editor angepasst werden, während Bilder außerhalb von Pi Presents auf die richtige Bildschirmauflösung gebracht werden müssen.

Im `pipresents-kms-examples` repository befinden sich Beispiele für Anwendungen in Pi Presents.

Für die Ausführung der Beispielprofile kann man das Bashscript `examples.sh` starten. Dafür `./examples.sh` in ein Terminalfenster im Verzeichnis `/pipresents` tippen oder die Datei `examples.desktop` von `/pipresents` auf den Desktop kopieren. Die `examples.sh` muss ausführbar gemacht werden und möglicherweise für einen anderen Benutzer als `pi` angepasst werden..

Die Beispiele nutzen folgende Tasten:

- Cursor Vor und Zurück – bewegt sich durch Menüs
- Cursor Vor und Zurück – bewegt sich in Mediashows. Damit kann man auch zum nächsten oder vorherigen Track einer automatischen Mediashow springen. Der Cursor bewegt sich zum Ende des Menüs und beginnt dann wieder beim Ersten Eintrag – und umgekehrt.
- Return – spielt den gewählten Menüeintrag oder einen Subeintrag.
- Escape – stoppt umgehend den aktuell laufenden Track oder die laufende Show.
- Leertaste – startet und beendet bei einem Image, Video- oder Audiotrack eine Pause.
- Strg-Break – beendet Pi Presents (kann auch mit dem Icon Fenster schließen und mit Alt+F4 erreicht werden).

Alle Profilbeispiele verfügen über eine Auswahl von Mediatracks, die im Verzeichnis `/home/pi/pp_home/media` gespeichert sind. Die Profile selbst sind im Verzeichnis `/home/pi/pp_home/pp_profiles` gespeichert. In jedem Profil bestimmt die Datei `pp_showlist.json` das „Look and Feel“ jeder Show; andere Dateien regeln die Benutzung der Medien. Diese Dateien können in einem gewöhnlichen Texteditor geöffnet werden – es ist jedoch viel besser, sie im Pi Presents Editor zu bearbeiten, der im [Kapitel 4](#) beschrieben ist.

Andere Dateien in den Profilen – `screen.cfg`, `gpio.cfg`, `keys.cfg`, `osc.cfg` und die `schedule.cfg` – regeln weitere Angebote von Pi Presents.

Wie Profile funktionieren, wird im Einzelnen noch erklärt. Nicht verzweifeln, wenn man es hier noch nicht gleich verstanden hat. Am besten freut man sich einfach über Pi Presents Fähigkeiten.

Die Beispiele:

- `pp_mediashow_1p5`

Das ist das oben heruntergeladene Profil. Mediashow startet sofort, fährt automatisch fort und beginnt wieder von vorn. Mit den Pfeiltasten für Auf und Ab können Sie den laufenden Showteil überspringen. Dieses Beispiel zeigt verschiedene Tracktypen und zusätzliche Informationen, die den Basisfunktionen hinzugefügt werden können.

- **pp\_menu\_1p5**  
Dieses Beispiel zeigt alle Menüformate, die Pi Presents bietet. Die zweite Menüebene und jede Show, die aus einer anderen Show gestartet wurde, heißen Subshows. Verwenden Sie die Pfeiltasten Auf und Ab, um durch das Menü zu wandern und starten dann mit Eingabe einen Track oder ein Untermenü, escape beendet beide.
- **pp\_exhibit\_1p5**  
Dieses Profil zeigt, wie eine Mediashow (nicht nur) mit der Infrarotfernbedienung (PIR) gesteuert werden kann. Beim Start ist der Bildschirm frei. Durch das Auslösen der PIR (öffnet einen Kontakt, der mit 0 Volt an P1-11 angelegt ist) oder durch Druck auf einen Button (schließt einen Kontakt mit 0 Volt, der an P1-18 angelegt ist) wird eine Mediashow einmalig gestartet. Danach wartet Pi Presents auf einen Auslöser, der die Show wiederholt.

Das Beispiel benutzt die Datei gpio.cfg im Profil, in der die Pins P1-11 und P1-18 den symbolischen Namen PIR tragen. Dieser Name ist mit dem Startbefehl für die Mediashow verknüpft.

Das Beispiel benutzt zudem die Eingabetaste als Steuertaste. So kann man den Effekt sehen, ohne die GPIO-Pins zu benutzen. Die keys.cfg Datei im Profil überschreibt die Standarddatei keys.cfg und verknüpft die Return-Taste mit dem symbolischen Namen PIR.

ACHTUNG: Vor der Benutzung der GPIO-Pins [Kapitel 6.4](#) lesen.

- **pp\_openclose\_1p5**  
Eine einmalig laufende Mediashow mit einer anderen Steuerungsvariante. Wenn ein Track durch das Öffnen einer Box oder einer Tür gestartet und durch das Schließen gestoppt werden soll – gpio.cfg nutzt dafür die Pins 1-18.
- **pp\_onerandom\_1p5** (Neu in Version 1.3)  
Spielt beim Start einen einzelnen Track aus einer Medienliste per Zufallsauswahl.
- **pp\_magicpicture\_1p5** (Neu bearbeitet in Version 1.3.2a)  
Zeigt ein statisches Bild von einer Person in einem Bildrahmen. Wenn ein Taster gedrückt wird, bekommt das Bild Leben und wird als Video gezeigt. Diesen Effekt erreiche ich mit einer Pause gleich nach dem ersten Frame. Ich habe leider kein Video mit einer Person, so muss auch hier „Suits“ wieder herhalten.
- **pp\_interactive\_1p5**  
In diesem Beispiel sind Mediashow und Menü kombiniert. Die Mediashow läuft kontinuierlich, aber jeder Beitrag zeigt den Hinweis, dass die Eingabetaste ein Menü öffnet. Durch Druck auf Escape geht es wieder zurück zur Mediashow. Ein Track oder eine Show, die von einer Mediashow erreicht werden können, wird child genannt.
- **pp\_presentation\_1p5**  
Die MediaShow ist als manuell gesteuerte Präsentation gestaltet. Mit den Auf- und Ab-Pfeiltasten kann sie durchsucht werden. Zwar kann Pi Presents keine Powerpoint-Präsentationen abspielen, sie können jedoch als Bilder (jpeg) abgespeichert und gezeigt werden.

- **pp\_liveshow\_1p5**  
Platzieren Sie bei laufender Liveshow Videos, Audiodateien, oder Bilder im Verzeichnis /home/pi/pp\_home/pp\_live\_tracks und erleben Sie, wie sie wundersamerweise erscheinen. Eine Liveshow ist eine Mediashow ohne vorher festgelegte Medien.
- **pp\_liveshowempty\_1p5** (neu in 1.3.2a)  
Demonstriert die verbesserte Handhabung von liveshows mit einem „empty tracks“-Verzeichnis. Ein „empty-live-tracks“-Verzeichnis spielt eine „list empty“ und führt die „on empty“ und „on not empty“ Showkontrollbefehle aus.
- **pp\_radiobuttonshow\_1p5**  
Die Radiobuttonshow funktioniert ähnlich, wie der Senderwechsel in einem Autoradio. Mit Druck auf die Tasten 1-4 oder auf GPIO-Taster wird jeweils ein Track gespielt. Mit Escape geht es zurück zum Hauptbildschirm.  
Definierte „controls“ im Radiobuttonshow-Profil machen es möglich, dass Tasten und GPIO-Taster für die Auswahl eines Tracks genutzt werden können. Im Profil ist eine gpio.cfg enthalten, die vier GPIO-Pins mit Einträgen und einen Pin mit der Stopp-Funktion verknüpft.
- **pp\_hyperlinkshow\_1p5**  
Die Hyperlinkshow arbeitet mit einem Set Webseiten, die untereinander verlinkt sind und eine „zurück“ und eine „home“ Schaltfläche zur Verfügung stellen. Ganz besonders die „erzählenden“ Programme in Museen gehen mit dieser Navigationstechnik um. Diese Show ist besonders für Touchscreens geeignet. Die Schaltflächen auf dem Bildschirm sind touchsensitiv. Sie können auch, z.B für Testzwecke, mit der Maus angeklickt werden. Die Datei screen.cfg in diesem Profil definiert die Buttons und die symbolischen Namen ihres Eingabebefehls. Wenn Softkeys eingesetzt werden, kann die Show über GPIO-Pins gesteuert werden, wenn sie in der gpio.cfg Datei mit den symbolischen Namen verknüpft sind.
- **pp\_audio\_1p5**  
Diese mediashow demonstriert die Fähigkeiten des Audioplayers. Um sie vollständig nutzen zu können, sind Lautsprecher am HDMI- und Analog-Port erforderlich. Die Show spielt Audiotracks auf verschiedenen Lautsprechern ab. Sie zeigt überdies die Funktion der Laufzeitkontrollen, mit denen die Lautstärken per Tastatur (8 und 9) oder GPIO-Pins sowie stumm und laut (Tasten 1 und 2) geregelt werden können. Dafür gibt es im Profil die Dateien keys.cfg und gpio.cfg.
- **pp\_web\_1p5**  
Eine Demonstration der Webbrowser-Fähigkeiten, für die eine Internetverbindung vorhanden sein sollte. Weil die Zusammenarbeit zwischen Pi Presents und dem Browser nicht besonders gut funktioniert, dauert es sehr lange bis der Browser öffnet und wenn eine Interaktion mit Pi Presents bei geöffnetem Browserfenster erfolgen soll, muss man erst einmal in das Browserfenster klicken.
- **pp\_concurrent\_1p5**  
Diese Anwendung zeigt, wie Shows gleichzeitig abgespielt werden können. Zwei Shows laufen simultan. Die Show mit den Bildern wird über die Tastatur gesteuert, während die konkurrierende Mediashow mit Audiotracks bei abgeschalteter Steuerung kontinuierlich läuft.
- **pp\_multiwindow\_1p5**

Diese Anwendung zeigt, wie Shows gleichzeitig abgespielt werden können, wobei viele Shows Inhalte zeigen. Ein Menü und vier Mediasows laufen gleichzeitig. Das Menü muss per Tastatur gesteuert werden während die gleichzeitig laufenden Mediasows bei abgeschalteten „controls“ fortgesetzt spielen. Die Zahl der gleichzeitig laufenden Shows ist lediglich durch die Leistungsfähigkeit des Raspberry Pi begrenzt. Im Beispielprofil enthalten sind:

- background – zeigt ein Hintergrundbild
- audio – spielt Hintergrundmusik
- text – ist eine Mediasow mit wechselndem Text
- slideshow – automatische Multimedia-Diasow
- menu – interaktives Menü

Die Show ist für die Fernsteuerung/-bedienung mit OSC vorbereitet.

- pp\_multidisplay\_1p5

Diese Anwendung zeigt, wie Shows gleichzeitig abgespielt werden können, wobei viele Shows Inhalte auf unterschiedlichen Bildschirmen zeigen. Ein Menü und vier Mediasows laufen gleichzeitig. Das Menü muss per Tastatur gesteuert werden während die gleichzeitig laufenden Mediasows bei abgeschalteten „controls“ fortgesetzt spielen. Die Zahl der gleichzeitig laufenden Shows ist lediglich durch die Leistungsfähigkeit des Raspberry Pi begrenzt. Im Beispielprofil enthalten sind:

- background – zeigt ein Hintergrundbild
- audio – spielt Hintergrundmusik
- text – ist eine Mediasow mit wechselndem Text auf HDMI0
- slideshow – automatische Multimedia-Diasow auf HDMI0. Die Diasow kann durch Klick auf ein klick-sensitives Feld auf HDMI1 angehalten werden, unabhängig vom der Pausenbefehl für Menüs auf HDMI1
- menu – interaktives Menü auf HDMI1. Dem Menü auf HDMI1 sind Klickbereiche zugeordnet.
- Eine der Menü-Optionen ist ein Video-Track, der auf HDMI0 gespielt wird, wenn die Diasow gestoppt worden ist.

Die Show ist für die Fernsteuerung/-bedienung mit OSC vorbereitet [s. Kapitel 17.4](#)

- pp\_multitouch\_1p5

Gleich mit pp\_multidisplay\_1p4, aber ohne Menü auf dem Touchscreen

- pp\_touchscreen\_1p5

Zeigt ein Menü auf dem Touchscreen mit einem Video auf HDMI0

- pp\_artmediashow\_1p5

Eine Mediasow, die nahtlose Übergänge ohne das Einfrieren des letzten Videoframes. Ist in der Vielfalt begrenzt, kann für künstlerische Ambitionen aber sehr brauchbar sein.

- pp\_artliveshow\_1p5

Eine Live-Mediasow, die nahtlose Übergänge ohne das Einfrieren des letzten Videoframes. Ist in seiner Vielfalt begrenzt, kann für künstlerische Ambitionen aber sehr brauchbar sein.

- pp\_subshow\_1p5

Zeigt, wie man eine größere Mediasow mit Subshows aufteilen und damit übersichtlicher machen kann.

- pp\_timeofday\_1p5

Öffnet und schließt zwei Mediashowes zu unterschiedlichen Tageszeiten und beendet Pi Presents. Kontrolliert wird der Prozeß von schedule.json im Profil. Um diese Demonstration zu ermöglichen, welchen Tag und welche Zeit auch immer, habe ich in der schedule.json Datum und Uhrzeit simuliert, indem ich „simulated time“ auf „yes“ gestellt habe. Fertigen Sie ein Backup dieser Datei an, verändern Datum und Uhrzeit und erleben Sie den Effekt. Achtung: Öffnungszeit am Morgen nur am Weihnachtstag – wie in englischen Pubs!

- pp\_trackplugin\_1p5  
Zeigt die Arbeitsweise von Track-Plugins. Track-Plugins sind Python-3-Module mit einer definierten API, die dynamische Displays etwa mit Uhren oder Inhalten aus dem Internet liefern. Bevor die Liveshow gestartet wird, müssen die cfg. Dateien vom /media Verzeichnis in das Verzeichnis /pp\_live\_tracks kopiert werden.
- pp\_showcontrol\_1p5  
Zeigt, wie Show Control konkurrierende Shows nacheinander startet und stoppt. Die Show ist für die Fernsteuerung/-bedienung für einen anderen Pi mit OSC ([siehe Kap. 17.4](#)) vorbereitet.
- pp\_showcontrolevent\_1p5  
Zeigt die Funktion des Befehls Show Control Event, wobei eine Trackpause mit pause-on- und pause-off-Befehlen begonnen und beendet wird.
- pp\_clickarea\_1p5  
Zeigt die Funktion von Klickbereichen in allen Showtypen und den Gebrauch von Softbuttons. In der radiobuttonshow sind Klickbereiche wahlweise durch einzelne Tracks abgestellt. Softbuttons verfügt über eine keys.cfg Datei, die Taste 1 und 2 den beiden Tracks zuweist. Bei entsprechender Einrichtung sind zwei GPIO-Taster links neben dem Bildschirm befestigt, die in der gpio.cfg Datei mit den symbolischen Namen verknüpft sind.

5

- pp\_animate\_1p5  
Animationen werden über den Pin P1-11 mit On am Beginn und mit Off am Ende des Tracks gesteuert. Für die Animationssteuerung gibt es im Profil eine gpio.cfg Datei. ACHTUNG: Stellen Sie zuvor sicher, dass P1-11 ungefährdet als Ausgang genutzt werden kann.
- pp\_shutdown\_1p5  
Zeigt, wie Show Control für das Herunterfahren des Pi genutzt wird. Eine entsprechende Meldung wird angezeigt. Wenn die Ab(wärts)-Pfeiltaste gedrückt wird, fährt ein so gestarteter Track den Pi sofort herunter.
- pp\_videoplout\_1p5  
Zeigt drei Showtypen, die als Videoabspieler funktionieren, wie man sie in Theatern und TV-Studios benutzt. ([siehe Kap. 5.4.5.2](#)).
- pp\_counters\_1p5  
enthält die Befehle zur Kontrolle von Zählern und nutzt den Plugin-Track, der die Werte eines Zähler aufnehmen und darstellen kann.
- pp\_quiz\_1p5  
demonstriert die Benutzung von Zählern und zeigt die Einbindung eines Quiz in eine Hyperlinkshow.

- `pp_osc_1p5`  
demonstriert die Show-Kontrollbefehle für die Steuerung via OSC
- `pp_beep_1p5`  
Zeigt Beep-Show-Befehl, Animationsbefehl und Show Control bei Events
- `pp_html_1p5`  
Zeigt HTML-Textfelder

### 3.1 Pi Presents beenden

Pi Presents wird mit mit Strg-Pause, Alt+F4 oder Click auf das Schließen-Symbol abgebrochen. Der Abbruch zielt darauf ab, Pi Presents auch im unsicheren Status zu beenden. Andere Möglichkeiten, Pi Presents zu beenden oder den Raspberry Pi herunterzufahren sind in [Kapitel 7](#) beschrieben

Einige Tastaturen haben keine Pause-Taste. In diesem Fall kann eine andere Taste für das Beenden von Pi Presents eingerichtet werden, indem in der `keys.cfg` Datei eine andere Taste mit dem symbolischen Namen `pp_terminate` verknüpft wird ([siehe Kap. 16.2.6](#)).

## 4 Der Pi Presents Profil-Editor

Der webbasierte Editor läuft in einem Browser und kann deshalb auch von einem fernen Computer bedient werden. Der Gebrauch des Webeditors per Fernsteuerung ist im [Kapitel 19.3](#) beschrieben.

Mit dem `pp_editor.py` können Profile erstellt und bearbeitet werden. Auf dem Pi lautet der Befehl:

```
python3 pp_web_editor.py
```

Es öffnet sich der Chromebrowser, der die grafische Oberfläche präsentiert.

Anmerkung: Ein anderer Benutzer als `pi` muss zunächst die Benutzerpfade in `pipresents/pp_config/editor.cfg` eintragen.

Wenn Sie den Editor benutzen, ist es erforderlich, Show-Referenzen, Track-Referenzen und Dateinamen einzurichten. Es ist ratsam, Namen nicht mit `pp-` oder `pp_` zu beginnen, um Konflikte mit Namen zu vermeiden, die Pi Presents benutzt.

### 4.1 Benutzung des Editors

#### 4.1.1 Die Programmfenster

Wählen Sie eines der Beispielprofile mit dem Menübefehl `Profile > Open`. Im Verzeichnisbrowser auf den Text des Beispielprofils (z.B. `pp_mediashow_1p4`) klicken, danach OK. Anders als im alten Editor bitte nicht das Verzeichnis öffnen.

- Das obere linke Fenster zeigt die im gewählten Profil verfügbaren Shows
- Das untere linke Fenster zeigt die Medienlisten dieses Profils. Mit einem Klick wird eine

- Medienliste markiert.
- Das rechte Fenster zeigt die Tracks der ausgewählten Medienliste.

Die ausgewählten Einträge werden auf rotem Hintergrund dargestellt. Klicken Sie auf die Schaltfläche 'Edit Show' neben dem linken Fenster, um die ausgewählte Show zu bearbeiten. Die ausgewählten Tracks werden nach Klick auf die Edit-Schaltfläche neben dem rechten Fenster bearbeitet.

Änderungen werden mit Klick auf OK gespeichert, mit Cancel bricht der Vorgang ohne zu speichern ab.

Um Auswirkungen der Änderungen gleich sehen zu können, sollten Sie pipresents.py und pp\_web\_editor.py nebeneinander in zwei Terminalfenstern öffnen. Pipresents.py zeigt die Änderungen, allerdings erst, wenn es neu gestartet wurde.

Wenn Sie die Beispiele bearbeiten wollen, sollten Sie zuvor Kopien der Originale anfertigen, indem Sie mit dem PCManFm-Dateimananger aus dem Verzeichnis

```
/home/pi/pp_home/pp_profiles
```

den gewünschten Ordner kopieren.

#### 4.1.2 Profil-Menü

**Profile>Open** öffnet ein Verzeichnisfenster, in dem Profile zur Bearbeitung ausgewählt werden können. Es handelt sich dabei um Profile im home+offset Verzeichnis, das im Menü Options>Edit Menu eingetragen ist.

**Profile>Validate** überprüft das Profil. Schrift- und Farbfelder werden nicht überprüft. Wenn die .json-Dateien lediglich mit einem Texteditor bearbeitet werden, geschieht das auf eigenes Risiko!

**Profile>Copy To** – Kopiert ein Profil

**Profile>Delete** - löscht ein Profil

**Profile>New From Template** - zeigt eine Liste von Vorlagen für alle Beispielshows. Die Vorlagen verfügen über Beispiel-Tracks die auch unbearbeitet laufen. Einigen fehlen allerdings die Konfigurationsdateien. Vorlagen dafür können in

```
/home/pi/pipresents/pp_config
```

und

```
/home/pi/pipresents/pp_ressources/pp_templates
```

kopiert werden.

#### 4.1.3 Show Menu

**Show>Add** - Fügt eine neue Show und eine Medialiste mit dem gleichen Namen hinzu. Der Showtyp einer bereits vorhandenen Show kann nicht geändert werden.

**Show>Copy To** – Kopiert die Show und erstellt eine leere Medialiste für diese Show.

**Show>Edit** – Dupliziert den Edit Show Button.

**Show>Delete** – Löscht die Show aus dem Profil, rückgängig nicht möglich.

**Show>View Backup** – zeigt die Parameter der Show und die zugehörigen Tracks aus dem Backup vom letzten Update des Profilthemas.

#### 4.1.4 Medialist Menu

**Medialist>Add** – Fügt eine neue Medialiste hinzu. Die Endung .json wird angefügt, falls nicht angegeben.

**Medialist>Copy To** -Kopiert die Medialiste

**Medialist>Delete** – Löscht die Medialiste aus dem Profil, rückgängig nicht möglich.

#### 4.1.5 Track Menu

##### Track>Add File

Fügt einen oder mehrere Trackeinträge mit Mediadateien an das Ende der Medialiste an. Das können Bilder, Videos oder Audiotracks sein. Der Editor erkennt automatisch den Speicherort und den Tracktyp. Sollte eines von beiden nicht erkannt werden oder die Dateierweiterung ist verschwunden, so fügen Sie mit Track>New einen leeren Track des gewünschten Typs hinzu.

Eine Liste der unterstützten Dateierweiterungen findet sich in den ersten Zeilen der pp\_definitions.py Quelldatei. Bitte einen Fehlerreport senden, wenn die Aufstellung unvollständig ist.

##### Track>Add Dir

Wie Add from File, fügt jedoch alle erlaubten Dateien aus dem Verzeichnis der Medialiste hinzu. Eine Liste der unterstützten Dateierweiterungen für die Auswahl von Tracks und Tracktypen findet sich in den ersten Zeilen der pp\_definitions.py Quelldatei. Bitte einen Fehlerreport senden, wenn die Aufstellung unvollständig ist.

##### Track>New

Fügt einen leeren Track des gewählten Typs am Ende der Medialiste hinzu.

##### Track>Edit

Dupliziert den Edit Button

##### Track>Copy

Kopiert einen Track

##### Track>Delete

Löscht den Track aus der Medialiste, rückgängig nicht möglich

#### 4.1.6 OSC Menü

##### OSC>Create

Erstellt eine leere osc.cfg Datei im Profil im Verzeichnis pp\_io\_plugins

### OSC>-Edit

Ändert die osc.cfg Datei

## 4.1.7 Tools Menu

### Tools>Update All

Versionsupdate für alle Profile im aktuellen Heimatverzeichnis und Offset-Verzeichnis ([siehe Kap. 22](#))

## 4.1.8 Options Menu

### Options>Edit

Bearbeiten der Editoreinstellungen

- **Pi Presents Heimatverzeichnis**  
Diese wichtige Einstellung muss das Verzeichnis nennen, in welchem alle Profilverzeichnisse und die zugehörigen Medien aufbewahrt sind. Standard ist /home/pi/pp\_home. Wenn ein alternativer Speicherort gewählt werden soll, ist sicherzustellen, dass das Verzeichnis /pp\_profiles in dem nun gewählten /pp\_home erstellt wird.
- **Einstieg Medienverzeichnis**  
Nur eine Hilfe, für den Suchanfang für 'Add Track' und 'Add from Dir'.
- **Speicherort für aktuelle Profile**  
Diese erweiterte Option erlaubt es, Profile auf Verzeichnisse unterhalb von /pp\_home/pp\_profiles zu verteilen und ermöglicht eine Verschiebung von /1p4\_examples in ein Standardverzeichnis /pp\_home/pp\_profiles/1p4\_examples.

## 4.1.9 Editor Befehlszeilen-Optionen

One of:	Wenn keine Option aufgeführt ist, gilt -l als angenommen
-l -local	local – der Editor-Server läuft und ein Browser ist auf dem RPI geöffnet (IP=127.0.0.1)
-r -remote	remote – der Editor-Server startet mit der IP-Adresse des RPI (z.B. 192.168.1.67). Er kann mit einem Browser von einem fernen Computer unter der Adresse <a href="http://192.168.1.67:8082">http://192.168.1.67:8082</a> erreicht werden
-n -native	native – der Editor läuft ohne Browser auf dem RPI. GT5 und pyview müssen installiert sein (was ich versäumt habe)

## 4.2 Profile erstellen

Die Anwendungsdaten liegen im Verzeichnis /pp\_home. Der Ordner /pp\_home muss ein Verzeichnis /pp\_profiles enthalten, in dem die Profile gespeichert sind.

Ein Profil ist ein Verzeichnis, das die Informationen enthält, die man benötigt, um eine einzelne Anwendung in Pi Presents zu konfigurieren. Es muss eine pp\_showlist.json Datei und eine oder

mehrere medialist (.json) Dateien enthalten. Die showlist.json enthält mehrere Bereiche. Einen 'start' show Bereich und mehrere Bereiche, in denen die vom Benutzer erstellten Shows definiert werden.

Jeder Bereich in der pp\_showlist.json bestimmt das 'Look and Feel' einer Show und regelt die Verknüpfungen. Medialist-(.json)-Dateien bestimmen den Inhalt jeder Show und einige Gestaltungsinformationen für die Tracks.

Um Profile transportabel zu machen, sind alle Medien am besten entweder im Verzeichnis /pp\_home oder alternativ innerhalb eines Profiles aufgehoben. In beiden Fällen empfiehlt es sich, Medien in einem Unterverzeichnis zu speichern, die Namenskonvention lautet media. Derart gespeicherte Medien referenzieren den relativen Pfad in einem Profil, z.B. +/media/myimage.jpg oder @/media/myimage.jpg.

Der Standardspeicherort für /pp\_home ist /home/pi; das muss durch Befehlszeilenoptionen überschrieben werden, wenn ein USB-Stick benutzt wird

Ein Profil kann auch andere Konfigurationsdateien enthalten, wie beispielsweise eine schedule.json und das optionale Verzeichnis /pp\_io\_config für Input/Output-Konfigurationsdateien wie gpio.cfg. Außerdem kann eine Textdatei eingefügt werden, die das Profil beschreibt.

#### 4.2.1 Profile portieren

Profile und ihre Medien können zwischen verschiedenen Laufwerken hin und her bewegt werden, ohne dabei die Verknüpfung zwischen Shows und ihren Media-Tracks zu verlieren. Das ist wichtig, wenn ein Profil auf der SD-Karte eines Raspberry Pi eingerichtet wurde, um es anschließend auf einem USB-Stick im Echtbetrieb einzusetzen.

Um diese Portabilität zu erreichen, sollten die Tracks im /pp\_home Verzeichnis, vorzugsweise in Unterverzeichnissen oder alternativ in einem Profil mit dem per Konvention benannten Verzeichnis ,media', gespeichert werden. Wenn Tracks so gespeichert werden, bevor das Profil erstellt ist, wird der Profil-Editor automatisch die richtige Verknüpfung erstellen.

- Wenn die Datei unter dem Homeverzeichnis (.../pp\_home) gespeichert wird, wie das im Menü unter Options>Edit voreingestellt ist, wird die Dateiverknüpfung bei relativem Pfad etwa so aussehen: '+/track\_to\_play.mp4'.
- Wenn sich der Track im Profil befindet, sieht die Referenz so aus: ',@/media/track\_to\_play.mp4'
- Andernfalls sieht die Verknüpfung bei absolutem Pfad folgendermaßen aus: '/home/pi/mymedia/track\_to\_play.mp4'.

Absolute Pfade machen Sinn, wenn damit beispielsweise Internetadressen wie [http://www.mysite.com/track\\_to\\_play.mp4](http://www.mysite.com/track_to_play.mp4). angegeben werden.

#### 4.2.2 Profile auf SD-Karte

Die SD-Karte für Entwicklung und Abspielen zu nutzen ist sehr sinnvoll, wenn Anwendungen auf dem Raspberry Pi erstellt werden. Um die SD-Karte für Entwicklung und Abspielen von Profilen zu nutzen

- erstellen Sie ein Verzeichnis `/pp_home` im Heimatverzeichnis des Pi-Benutzers und erstellen darin ein Verzeichnis `pp_profiles`. Sie können auch ein Verzeichnis in `/home/pi/pp_home` einrichten, etwa unter dem Namen `media`, um darin Mediendateien abzulegen.
- Tragen Sie in den Editor-Einstellungen für das Pi Presents-Daten-Heimatverzeichnis `/home/pi/pp_home` und für das Startverzeichnis von Medien `/home/pi/pp_home/media` ein.
- kopieren Sie die Mediendateien in das `/media` Verzeichnis.
- Erstellen Sie mit dem Editor ein neues Profil, nennen es `myprofile` und bearbeiten es.
- tippen Sie, um das Profil zu starten:

```
python pipresents.py -p myprofile
```

in einem Terminalfenster, das Sie zuvor im `pipresents` Verzeichnis geöffnet haben.

Wenn Pi Presents durch eine Desktopverknüpfung oder von einer Autostart-Datei gestartet wird, ist es am besten, den vollständigen Pfad von `pipresents` und auch den vollständigen Pfad des Daten-Heimatverzeichnisses wie folgt anzugeben:

```
python3 /home/pi/pipresents/pipresents.py -o /home/pi -p myprofile
```

### 4.2.3 Editor auf einem Windows PC

Bitte die Remote-Steuerung des Editors benutzen. Der Editor läuft auf dem Pi, aber die grafische Oberfläche erscheint im Browser auf dem PC.

### 4.2.4 Profile auf USB-Stick

Um Profile von einem USB-Stick zu starten kopieren Sie das `'pp_home'`-Verzeichnis von seinem Ort oder dem Pi auf die oberste Ebene eines USB-Sticks.

Stecken Sie den USB-Stick in den Pi und starten Sie Pi Presents mit der `-o` Option für `/media/STICKNAME`. `STICKNAME` ist der Laufwerksname.

### 4.2.5 Profile für Monitore mit unterschiedlicher Pixelgröße erstellen

Die Option, beliebige Displaygrößen in `/pipresents/pp_config/pp_display.cfg` zu wählen, hilft bei der Entwicklung neuer Anwendungen auf einem Arbeitsmonitor, der sich in der Pixelzahl vom Zielmonitor unterscheidet. Diese Option läßt Pi Presents die gewünschten Pixeleinstellungen statt der von Raspberry Pi OS erkannten Bildschirmgröße aufrufen.

- Alle Layoutoperationen, wie zentrieren und die Bildgrößenanpassung, erhalten die gewählten Zielgrößen.
- Wenn der `fullscreen`-modus nicht eingerichtet ist, zeigt ein gelber Rahmen die Größe des Zielbildschirms an.
- Wenn `fullscreen` eingeschaltet ist, zeigt der Bildschirm die Ziel-Bildgröße und zwar vom

oberen linken Bildschirmrand. Wenn die Ziel-Bildgröße die Bildgröße des Arbeitsmonitors übersteigt, verschwindet der überschießende Teil der Anzeige.

## 5 Die Komponenten in Pi Presents

### 5.1 Einführung

#### Shows und Tracks

Das Pi Presents Werkzeug besteht aus zwei Grundelementen – Shows und Tracks

Eine Show zeigt Tracks; die zu spielenden Tracks sind in einer Medialiste enthalten, die der Show zugeordnet ist. Stellen Sie sich die Medialiste als eine erweiterte Playliste vor.

Das Werkzeug bietet aktuell sieben Showtypen, sie eignen sich jeweils besonders gut für unterschiedliche Zwecke.

- **Mediashow** – spielt eine Sequenz von Tracks, in der Regel automatisch, allerdings kann das Abspielen auch manuell kontrolliert werden. Übergänge zwischen Tracks sind lückenlos, weil Videos am Ende und das letzte Bild der Show eingefroren werden bis der nächste Track startet.
- **Artmediashow** – spielt eine Reihe Tracks, wie die Mediashow, gänzlich lückenlos. Der nächste Track wird geladen, während der vorherige Track spielt. Infolgedessen müssen ablaufende Tracks am Ende nicht einfrieren. Die Artmediashow hat keine Steuerungselemente und erlaubt weder Childtracks noch Subshows.
- **Hyperlinkshow** – benutzt Touchscreen-Funktionalität, wie sie in vielen Museen vorkommt.
- **Radiobuttonshow** – Eine einfache Kioskshow mit einem Navigationsbildschirm. Druck auf einen von mehreren Buttons spielt einen Track.
- **Menu** – Ähnlich der Radiobuttonshow, aber die Trackauswahl erfolgt mit Durchsuchen eines Menüs mit Pfeiltasten oder einzelnen Buttons.
- **Liveshow** – Wie eine Mediashow, jedoch mit dynamisch eingespieltem Inhalt von außen.
- **Artliveshow** – zeigt von außen eingespielte Inhalte, wie die Liveshow, und zeigt sie ohne Unterbrechung. Der nächste Track wird geladen, während der aktuelle Track gespielt wird. Infolgedessen müssen ablaufende Tracks am Ende nicht einfrieren. Die Artmediashow hat keine Steuerungselemente und erlaubt weder Childtracks noch Subshows.

Das Tool Pi Presents hat Player für fünf Tracktypen, wobei jeder Player einen anderen Medientyp abspielt. Alle Player erlauben bei Bedarf das Abspielen von Primärmedien in einem Fenster mit einfachem Farbhintergrund, Bildern und Texten; Player gestatten die Steuerung von Animationen, spielen in Python 3 geschriebene Track-Plugins und erlauben Start und Stopp bei anderen konkurrierenden Shows.

- **mpv** – spielt Videos mit dem MPV\_Player.
- **Audio** – spielt Audiotracks mit dem mplayer.
- **Bilder** – zeigt Bilder in vielen Formaten (.jpg etc.) (benutzt Python Imaging Library).
- **Textmeldungen** – eine schnelle Möglichkeit, Texte anzuzeigen.

- chrome – benutzt einen eingebetteten Webbrowser (benutzt chrome).

## Subshows und Child-Tracks

In Pi Presents kann eine Show ein Track seiner Elternshow sein; sie heißt Subshow. Subshows dienen unterschiedlichen Zwecken, wie:

- Aufteilung einer längeren Mediashow in Einzelsegmente. Jedes Segment ist eine Show. Eine Elternshow führt die Segmentshows in ihrer Medialiste auf.
- Ein Menü oder eine Radiobuttonshow können Mediashow als Intro oder als Einzeltracks haben.
- Menüs mit mehreren Ebenen.

Subshows können sich beliebig nach unten verzweigen, eine Grenze bildet dabei möglicherweise nur die Verwirrung der Besucher, wenn sie mit zu vielen Ebenen arbeiten müssen. Subshows sind Tracks und tauchen als solche auch in der Medialiste als Show Tracks auf.

Childtracks sind Tracks, die in besonderer Weise von Mediashow und Liveshow benutzt werden. Wenn ein Childtrack im Profil angelegt ist, kann er während des Abspielens eines Tracks in der Elternshow gestartet werden. Nach Ablauf des Childtracks setzt sich die Elternshow mit ihrem nächsten Track fort. Child Tracks können, wie jeder Track, Shows sein, die manchmal als Child Shows referenzieren.

## Die „Eier-Uhr“

In allen Shows sind die Übergänge lückenlos (keine schwarzen „Löcher“ zwischen den Tracks). Das wird dadurch erreicht, dass Videos und Bilder am Ende des Tracks eingefroren werden, während der nächste Track noch geladen wird. Bei manueller Fortsetzung gibt es eine Verzögerung von bis zu 2 Sekunden, nachdem eine Taste gedrückt wurde. Mit der „Eier-Uhr“ kann in dieser Verzögerungspause Text eingeblendet werden, solange der nächste Track geladen wird. Dieser Text liegt über dem Track-Bild und Inhalt, Format und Position können im Show-Profil konfiguriert werden.

## Konkurrierende Shows

Pi Presents kann zwei oder mehr Shows gleichzeitig abspielen ([siehe Kapitel 7.4](#)). Die konkurrierenden Shows laufen parallel. Das macht Vieles möglich, wie

- einen Audio-Hintergrund für eine Diashow. Gebrauch von zwei Mediashow, eine mit Diashow und die andere mit den Audiotracks. Bei Letzterem muss die Steuerung abgestellt sein, wenn es eine Nutzer-Interaktion bei Ersterem gibt.
- Die Aufteilung des Bildschirms in mehrere Bereiche (Show-Leinwände), die jeweils eine andere Show zeigen, etwa eine automatische Diashow in einem Sektor, zeitgesteuert von einer Show mit einem Track-Plugin in einem zweiten Bereich und in einem dritten Sektor ein vom Benutzer bedientes Menü mit Bildern. Für alle Shows, bis auf eine, könnte die interne Steuerung ausgeschaltet sein. Um erweiterte Profile zu erstellen, kann Pi Presents jeder der konkurrierenden Shows und jeder der Subshows jeweils einen eigenen Satz

Controls zuweisen.

- Wirklich sparsam wird es, wenn zwei komplett selbständige Tasks mit einem Pi laufen, - vielleicht eine Diashow im Foyer und eine Erzählstimme in einer Ausstellung, gesteuert mit einer Infrarot-Fernbedienung.

Jede der konkurrierenden Shows kann Subshows enthalten.

## 5.2 Shows

Shows besitzen Felder, in denen ihr Abspielmodus und das Aussehen festgelegt sind. Es gibt einige Felder, die Standardwerte für die Tracks in der Show, wie z.B. Dauer, Überblendung, MPV-Video, Audio und Chrome Web benutzen, wenn deren entsprechende eigene Felder leer sind.

Alle Showes haben folgende Felder

- Title – Text, der im Editor und den Zeilen einer Menu Show erscheint.
- Show Reference – Ein Name, der anderen Shows als Verweis auf diese Show dient. Erlaubt ist ein alphanumerischer String ohne Leerzeichen.
- Medialist – ist der Name einer Datei, die im Medialistenfenster erscheint. Sie muss die Endung .json haben. Jede Show muss eine Medialiste haben, in der die Tracks dieser Show benannt sind. Die selbe Medialiste kann von mehreren Shows benutzt werden.

### 5.2.1 Mediashow und Artmediashow

Stellen Sie sich eine Mediashow als Diashow vor, die verschiedene Tracktypen abspielen kann – Videos, Audiotracks, Bilder und sogar gesteuerte Animationen.

Mediashows haben Felder, in denen ihre Steuerung festgelegt wird, wie z.B. Startbefehl, Wiederholung/Einmal-Lauf und Reihenfolge.

Ein Track kann einer Mediashow so zugeordnet sein, dass dieser Track von jedem anderen Track der Show aus erreichbar ist. Das sind 'Child Tracks'. Die Child-Track-Parameter legen die Track-Referenz fest, wobei es sich um einen Track oder eine Show handeln kann. Mit einem Child-Track ist ein Hinweistext verbunden, der nur angezeigt wird, wenn das Child-Track-Feld nicht leer ist.

Das Repeat/Single-Feld bestimmt den Ablauf der Mediashows:

- repeat – Die Mediashow wiederholt sich, bis sie gestoppt wird.
- single-run – Die Mediashow läuft einmal und endet. Single-run ist in erster Linie für Subshows und mit Show Control vorgesehen.

In weiteren Feldern können Mediashows für viele Profilanwendungen eingerichtet werden. Die folgende Tabelle zeigt die Befehle und andere Feldeinträge für den allgemeinen Gebrauch von Mediashows und Liveshows.

Task	Repeat	Trigger for Start	Trigger for End	Other Fields (where non-zero)
Fortlaufende Show	repeat	start	none	
Fortlaufende Show, die die Medialiste wiederholt abspielt bis eine festgelegte Zeit vergangen ist	repeat	start	none	Show Timeout = h:m:s

und danach endet				
Fortlaufende Show, die die Medialiste wiederholt abspielt	repeat	start	none	Repeat Interval = h:m:s
Eine Medialiste per Input abspielen (früher Single Shot)	repeat	input	none	
Eine Medialist, in der jeder Track auf eine Eingabe wartet	repeat	input	none	Trigger for Next=Input
Eine Medialiste per Input abspielen. Verhindert für eine bestimmte Zeit einen Neustart, nachdem die Medialiste abgearbeitet ist	repeat	input	none	Repeat Interval h:m:s
Spielt einen per Input gestarteten Track. Wenn die Reihenfolge = shuffle wird ein Zufallstrack gezeigt	repeat	input		Track Count Limit = tracks
Startet und beendet eine laufende Medialiste, wenn ein Input durch Tastendruck oder -lösung erfolgt	repeat	input	input	In der gpio.cfg rising edge oder falling edge eines Kontaktes für die jeweilige Aktion
'Magic Picture'. Startet ein Video und pausiert es nach dem ersten Bild. Pausenende mit dem go Befehl, danach repeat	repeat	start	none	Video erfordert ‚freeze at start‘= ‚after first frame‘. ‚Freeze at end‘ = ‚no‘ Außerdem wird der ‚go‘ Befehl benötigt
Eine Mediashow als Subshow oder als Teil eines Menüs, einer Radiobuttonshow, etc. benutzen	Single-run	start	none	

Mediashows und Liveshows kann der Benutzer mit den Kommandos Up, Down and Stop kontrollieren. Ihre Verknüpfung mit symbolischen Namen von Input-Ereignissen steht in den Kontrollfeldern.

### Artmediashow

Artmediashows bieten eine gänzlich lückenlose Wiedergabe: Zwischen den Tracks gibt es keine Lücke und die Tracks frieren am Ende nicht ein, um die Ladezeit des nächsten Tracks zu überbrücken. Artmediashows haben einige Beschränkungen und verfügen nicht über die Felder und Befehle, die in der folgenden Tabelle eingeklammert sind:

- Subshows
- Child Tracks
- Start, Next und End Befehle
- Web Tracks

### 5.2.1.1 Steuerungen und Befehle

Zunächst werden Mediashows und Liveshows mit ihren Befehlen gesteuert. Standardmäßig sind diese Befehle mit einem Satz symbolischer Namen verknüpft und diese wiederum sind mit einem Satz Tasten verknüpft. Zusätzlich kann Pi Presents mit GPIO und einem nützlichen Satz Verknüpfungen gesteuert werden, indem eine entsprechend vorbereitete gpio.cfg Datei aus dem Verzeichnis /pipresents/pp\_resources/pp\_templates in den Profilordner kopiert wird.

Die Standardeinstellungen nach der Installation sind:

Befehl	Standardeinstellung GPIO-Pins siehe <a href="#">Kapitel 16.2.2</a>			Effekt
	Symbolischer Name	Taste	GPIO Pin	
up	pp-up	Cursor Up		Vorheriger Track in einer Media- oder Liveshow oder Zurück zum Menü
down	pp-down	Cursor Down		Nächster Track in einer Media- oder Liveshow oder vor zum Menü
play	pp-play	Return		Starten eines Menüeintrages oder einer Childshow
pause	pp-pause	Leertaste		Schaltet auf Pause für Tracks, die Pause unterstützen
(repeat)				Wiederholt den aktuellen Track
pause-on				Pause für Tracks, die Pause unterstützen
pause-off				Pausenende für Tracks, die Pause unterstützen
mute				Lautstärke aus für Tracks, die das unterstützen
unmute				Lautstärke an für Tracks, die das unterstützen
stop	pp-stop	Escape		Stoppt einen laufenden Track und kehrt, wenn eine Show im Ruhestatus ist und keine Top-Level-Show ist, zur Elternshow zurück Ruhestatus: <ul style="list-style-type: none"> <li>• Liveshow/Mediashow – jederzeit</li> <li>• Menu – bei der Anzeige</li> </ul> Im Ruhestatus und im Fall einer Top-Level-Show beendet 'stop' eine Single-Run-Show, geht aber zum nächsten Track einer sich wiederholenden Show.

(go)	(nicht verknüpft)	(nicht verknüpft)	(nicht verknüpft)	Pausenende bei einem Videotrack der beim Start eingefroren ist
null	(nicht verfügbar)	(nicht verfügbar)	(nicht verfügbar)	Verhindert die Steuerung mit dem selben symbolischen Namen, der in der Show festgelegt ist
exit	pp-exit	(nicht verknüpft)	(nicht verknüpft)	Beendet das Abspielen eines Tracks und kehrt zur Elternshow zurück oder, bei einer Top-Level-Show, beendet die Show
incvolume	(nicht verknüpft)	(nicht verknüpft)	(nicht verknüpft)	Nur für VLC-Videos: Erhöht Lautstärke um einen Schritt
decvolume	(nicht verknüpft)	(nicht verknüpft)	(nicht verknüpft)	Nur für VLC-Videos: Verringert Lautstärke um einen Schritt
mplay-*	(nicht verknüpft)	(nicht verknüpft)	(nicht verknüpft)	Führt einen Laufzeitbefehl für den Audioplayer aus (s. Kapitel 13)

Die Standardtasten für alle Operationen sind in /pipresents/pp\_io\_config/keys.cfg festgelegt und die GPIO-Pins in der Datei gpio.cfg. Im Kapitel 17.2 ist beschrieben, wie die Verknüpfungen geändert werden können. Die Verknüpfung der Befehle mit den symbolischen Namen ist im Kontrollfeld der Show eingetragen. Eine Änderung der Verknüpfungen ist in der Regel nicht erforderlich, sie können jedoch für individuelle Shows überschrieben werden, wie nachfolgend beschrieben.

Wenn mehrere Shows gleichzeitig laufen geht die Eingabe an alle und wird möglicherweise in jeder Show ausgeführt. Das ist unter Umständen nicht gewünscht, wenn eine Mediashow beispielsweise mit Audiotracks als Hintergrund für eine manuell gesteuerte Diashow fungiert; dabei sollen die Audiotracks nicht mit 'vor' und 'zurück' wechseln – sondern lediglich die Dias/Bilder. Also sollten die Controls für die Audiotracks abgestellt werden.

Wenn 'Disable Controls'=yes für die Show gilt, sind alle Kommandos für die Show abgestellt und sie muss automatisch oder durch Auslöser, wie im nächsten Kapitel gezeigt, kontrolliert werden. Für weitergehende Szenarien können einzelne Befehle gelöscht oder mit alternativen symbolischen Namen verknüpft werden, indem die Kontrollfelder der Show editiert werden.

Kommandos können in den Kontrollfeldern der Tracks als Ergänzung zu den Einträgen in den Kontrollfeldern der Show platziert werden. Die Controls in den Tracks verbinden sich und überschreiben die Controls der Show. Das ist in der Regel nicht nötig, kann aber beispielsweise erforderlich sein, wenn die Zuschauer ein Video ganz zu Ende sehen sollen. In diesem Fall verhindern 'null'-kommandos die Kommandos 'up', 'down' und 'stop' für den Track.

### 5.2.1.2 Triggers

Mediashows und Liveshows haben Steuerungseigenschaften (Trigger). Sie bieten andere Kontrollparadigmen an als ‚Vor‘ und ‚Zurück‘.

So wie eine Mediashow läuft oder sich wiederholt, „fragt“ die Show im „Trigger for Start“-Feld nach, wie fortzufahren ist. Nach jedem Track kann „Trigger for Next“ für eine Kontrolle des Übergangs zum nächsten Track benutzt werden. „Trigger for End“ legt das Ende des Tracks fest.

Triggers können Subshows und Childshows beigefügt werden. Triggers werden durch „Disable Controls“ nicht unterbunden.

#### 5.2.1.2.1 Trigger for Start

Trigger for Start kann folgende Werte annehmen:

- start  
Die Mediashow geht zum ersten Track
- input/input-persist  
Die Mediashow wartet vor dem Start des ersten Show-Tracks auf eine Eingabe. Der symbolische Name der Eingabe muss im 'Trigger for Start'-Parameterfeld enthalten sein. Input persist ist ein Weg, „Magic Pictures“ einzufügen. Nun verbessert mit „Freeze at Start“, nach dem ersten Frame im Video-Track.

Text, der während der Wartezeit für eine Starteingabe gezeigt werden soll, ist im 'Notices Tab' von Media- und Liveshows einzutragen.

#### 5.2.1.2.2 Trigger for End

'Trigger for End' veranlasst die Show zu enden (single-run) oder zu wiederholen (repeat). Das Feld kann folgende Werte annehmen:

- none  
Der 'end trigger' ist nicht aktiv. Die Show endet nach Durchlauf der medialist/des livetrack-Verzeichnisses.
- Input  
Der 'end trigger' ist aktiv, wenn eine Eingabe mit einem symbolischen Namen mit dem Eintrag im 'Trigger for End'-Parameterfeld übereinstimmt

#### 5.2.1.2.3 Trigger for Next

'Trigger for next'-Feld erlaubt es, einzelne Tracks in einer Mediashow mit einer Eingabe zu steuern

- None - Trigger ist nicht aktiv. Die Show startet nach Ende eines Tracks den nächsten Track
- Input – Am Ende eines Tracks wartet die Show und geht zum nächsten Track, wenn eine Eingabe mit dem symbolischen Namen im 'Trigger for Next'-Parameterfeld verbunden ist. So lange ein Track läuft ist Trigger deaktiviert.

Anzeigetext für die Wartezeit bis zum Trigger befindet sich im notice tab einer mediashow oder liveshow.

#### 5.2.1.3 Felder

Eingabefelder für Media- und Liveshow. In Klammern die Felder, die in Artmedia- und Artliveshows nicht benutzt werden.

Feld	Beispiel	Wert
<b>Show Tab</b>		
Type	(art)mediashow	Kann nicht geändert werden
Title	My First Show	Textbeschreibung der Show. Wird im Editor und in Menüs angezeigt

Show Reference	show1	Etikett mit dem die Show anderen Shows zugeordnet wird. Beliebiger Text ohne Leerzeichen
Display	HDMI0	HDMI0, HDMI1 – Das Display, auf dem die Show/Subshow läuft
Medialist	show1.json	Name der Medialist-Datei in der die Tracks der Show enthalten sind. Standardname ist gleich mit der Show Reference, er kann jedoch auch abgeändert werden
(Trigger For Start)	start	Wie sich die Mediashow nach dem Start und zum Beginn eine Wiederholung fortsetzt: <ul style="list-style-type: none"> <li>• start – fortsetzen ohne zu warten</li> <li>• input – wartet auf Eingabeaktion</li> <li>• input-persist – wartet auf Eingabe-Aktion. Input-persist zeigt den unmittelbar vorher gegangenen Track während der Wartezeit (Videos werden an ihrem Ende eingefroren)</li> </ul>
(Trigger For Start Parameter)	PIR	Wenn Trigger For Start = input oder input-persist: Ein symbolischer Name ist mit einer Eingabeaktion wie in <a href="#">Kapitel 16.2</a> beschrieben
(Trigger For Next)	continue	Wie der nächste Track angestoßen wird <ul style="list-style-type: none"> <li>• input – reagiert auf eine Eingabe-Aktion</li> <li>• continue – setzt sich ohne Warten fort</li> </ul>
(Trigger For Next Parameter)		Wenn Trigger For Next = input: Ein symbolischer Name ist mit einer Eingabeaktion verknüpft, wie in <a href="#">Kapitel 16.2</a> beschrieben
Sequence		Trackreihenfolge: <ul style="list-style-type: none"> <li>• ordered/reverse – wird in der Reihenfolge laut Medialiste gespielt (nur un Mediashows)</li> <li>• reverse – spielt ab in umgekehrter Reihenfolge der Playlist (nur Liveshows)</li> <li>• shuffle – spielt Tracks in einer Zufallsfolge Für liveshow bedeutet das eine echte Mischung und ein Track wird solange nicht wiederholt, bis alle Tracks im Live-Tracks-Verzeichnis abgespielt sind. Bei Mediashow wird der nächste Track zufällig ausgewählt.</li> </ul>
Repeat/Single	repeat	Wiederholung der Mediashow: <ul style="list-style-type: none"> <li>• repeat – wartet auf Eingabe für Start. Einmaliges Starten der Tracks in der Medialiste. Danach setzt sich die Medialiste mit Warten auf eine Eingabe für Start fort</li> <li>• single-run – Wartet auf eine Eingabeaktion für Start. Tracks werden einmal gespielt, danach endet die Show</li> </ul> Achtung: Es erfolgt kein automatisches Ende bei einer zufälligen Trackauswahl
(Trigger For End)	none	Wie das Ende einer Show gesteuert wird. Siehe

		<p>Tabelle oben für weitere Einzelheiten.</p> <ul style="list-style-type: none"> <li>• None - die Ende-Eingabe ist nicht eingestellt</li> <li>• input - endet nach einer Eingabeaktion wie in <a href="#">Kapitel 16.2</a> beschrieben</li> <li>• duration - beendet die Show nach der vorgegebenen Zeit</li> </ul>
(Trigger For End Parameter)	5:00	<p>If Trigger For End =:</p> <ul style="list-style-type: none"> <li>• input – Ein symbolischer Name ist mit einer Eingabeaktion verknüpft <a href="#">s. Kapitel 16.2</a></li> </ul>
(Track Count Limit)	1	<p>Wenn ungleich 0 endet die Show oder startet neu, wenn die gewählte Trackanzahl erreicht ist. Das ist angebracht, wenn das Tracklimit niedriger ist, als die Zahl der unbenannten Tracks in der Medialiste</p>
(Repeat Intervall)	30	<p>h:m:s</p> <p>Die Show setzt sich fort oder endet nach dem Repeatintervall oder, je nach Dauer, mit dem Ende der Medialiste. Wenn die Medialiste vor dem Repeatintervall abgespielt ist, erscheint ein leerer Bildschirm.</p> <p>Wenn das Intervall = 0 wiederholt/endet die Show, wenn die Medialiste abgearbeitet ist. Intervall = 0 funktioniert nicht bei einer Zufallsauswahl, weil diese niemals endet.</p> <p>Für Zufallsauswahl und bei Repeatintervall &gt;0 läuft die Show weiter, während eine Zufalls-Medialiste sie nicht beendet.</p>
(Show Timeout)	1:30:20	<p>h:m:s = nicht 0 beendet die Show nach der festgelegten Zeit, wenn nichts anderes die Show bereits beendet hat.</p>
Show Canvas	100 100 400 600	<p>Obere linke und untere rechte Koordinate des Fensters der Show. Show Canvas hilft, den Bildschirm aufzuteilen, wenn mehrere Shows gleichzeitig abgespielt werden sollen. Wenn das Feld nicht leer ist, steuert Show Canvas Position und Größe von Bildern und Videos bezogen auf die Bildschirmgröße. Show Canvas bestimmt den Ursprung des Bildes (oben links), es ist jedoch kein Fenster, das Bilder oder Videos beschneidet. Wenn also die untere rechte Koordinate des Bildes größer als die Show-„Leinwand“ ist, kann sie andere Show-„Leinwände“ überlappen.</p> <p>Statt der 2 x.y.-Paare kann <math>x1+y1+w*h</math> eingesetzt werden</p>

Empty List Track		Nicht für Mediashow oder Artmediashow
Escape Track		Nicht für Mediashow oder Artmediashow
<b>Child Track Tab</b>		
(Child Track)	Child-track	Nicht leer wird ein Childtrack aktiviert und seine Track-Referenz spezifiziert.
(Hint Text)	Press Play to...	Eine Textzeile
(Hint Font)	Helvetica 30 bold	Den Schriftsucher benutzen oder direkt eintragen – siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Hint Colour	Helvetica 30 bold	Den Farbsucher benutzen, um eine sechsstellige Hexadezimalziffer zu erhalten. Oder Farbnamen direkt eintragen - siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
(Hint Justify)	right	Left, right, center – Justiert Zeilen in einem Textblock
(Hint x)	500	Textposition von links. Wenn leer, wird der Text auf dem Show-Fenster zentriert
(Hint y)	900	Textposition von oben. Wenn leer, wird der Text auf dem Show-Fenster zentriert
<b>Eggtimer Tab</b>		„Eieruhr“-Text wird angezeigt, wenn der nächste Track geladen wird
Eggtimer Text	Loading...	Eine Textzeile
Eggtimer Text Font	Helvetica 30 bold	Den Schriftsucher benutzen oder direkt eintragen – siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Eggtimer Text Colour	white	Den Farbsucher benutzen, um eine sechsstellige Hexadezimalziffer zu erhalten. Oder Farbnamen direkt eintragen – siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Eggtimer Text x	100	Abstand des Textbeginns vom linken Bildschirmrand

Position		Wenn leer, wird der Text im Showfenster zentriert.
Eggtimer Text y Position	100	Abstand des Textbeginns vom oberen Bildschirmrand (Pixel). Wenn leer, wird der Text auf dem Show-Fenster zentriert
Eggtimer Justify	right	Left, right, center – Justiert Zeilen in einem Textblock
<b>Notices Tab</b>		
(Trigger Wait Text)	Waiting...	Text bei Warten auf Eingabe
Notice Text x Position		Abstand des Textbeginns vom linken Bildschirmrand (Pixel). Wenn leer, wird der Text auf dem Show-Fenster zentriert
Notice Text y Position		Abstand des Textbeginns vom oberen Bildschirmrand (Pixel). Wenn leer, wird der Text auf dem Show-Fenster zentriert
Notice Text Colour		Den Farbsucher benutzen, um eine sechstellige Hexadezimalziffer zu erhalten. Oder Farbnamen direkt eintragen – siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Notice Text Font		Den Schriftsucher benutzen oder Schriftnamen direkt eintragen – siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Notice Justify	right	Left, right, center – Justiert Zeilen in einem Textblock
<b>Show Background and Text Tab</b>		
Background Image	+/media/image.jpg	Wenn nicht leer, steht hier der Dateiname des Bildes, das für die Tracks als Hintergrund fungiert. Das Bild ist „aufgeblasen“, um die Show-Leinwand auszufüllen. Wenn das nicht gewollt ist, sollte das Bild vorher in der Größe der Show-Leinwand eingerichtet werden.
Background Colour		Den Farbsucher benutzen, um eine sechstellige Hexadezimalziffer zu erhalten. Oder Farbnamen direkt eintragen – siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> . Wenn Pi Presents startet ist die Hintergrundfarbe schwarz. Der Wert in diesem Feld wird von einem

		Show-Track benutzt, wenn das Trackfeld leer ist. Wenn die gemeinsame Hintergrundfarbe im Feld leer ist, wird die Hintergrundfarbe nicht gewechselt.
Show Text	Pictures from my holidays	Showtext überlagert alle Tracks der Show. Gezeigter Text, wenn das Feld nicht leer ist.
Show Text Location	+/media/mytextfile	Wenn dieses Feld nicht leer ist, wird der Showtext durch Text in mytextfile ersetzt
Show Text x position	100	Entfernung des Textbeginns vom linken Bildschirmrand (Pixel). Wenn das Feld frei bleibt, wird der Text im Showfenster zentriert
Show Text y position	100	Entfernung des Textbeginns vom oberen Bildschirmrand (Pixel). Wenn das Feld frei bleibt, wird der Text im Showfenster zentriert
Show Text Type	html	Plain - einfacher text html – eine Sammlung mit Text, Bildern und Formatierungen auf Grundlage von HTML. Siehe <a href="#">Kapitel 5.4.4.2</a>
Plain Text Font	Helvetica 30 bold	Den Schriftsucher benutzen oder Schriftnamen direkt eintragen – siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Plain Text Colour	white	Den Farb sucher benutzen, um eine sechsstellige Hexadezimalziffer zu erhalten. Oder Farbnamen direkt eintragen – siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> .
Plain Text Justify	right	Left, right, center – Justiert Zeilen in einem Textblock
HTML Text Hight	300	Höhe des HTML-Textfeldes
HTML Text Width	300	Breite des HTML-Textfeldes
HTML Text Background Colour	#ff0000	Hintergrundfarbe des HTML-Textfeldes
<b>Track Defaults Tab</b>		Showtracks benötigen einige Einstellungen, sofern sie nicht bereits in den Tracks selbst vorgenommen sind
Duration	10	Sekunden - Anzeigedauer, wenn der Track nicht von selbst endet. Bei Wert 0 läuft der Track fortwährend
Pause Timeout	5	Sekunden - Wenn nicht leer und größer 0 wird ein pausierender Track automatisch mit Ablauf des Timeout die Pause beenden
Track Text Font	Helvetica 30 bold	Den Schriftsucher benutzen oder Schriftnamen direkt

		eintragen – siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Track Text Colour	white	Den Farbsucher benutzen, um eine sechstellige Hexadezimalziffer zu erhalten. Oder Farbnamen direkt eintragen – siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Track Text x Position	100	Entfernung des linken Textrandes vom linken Bildschirmrand (in Pixel). Wenn leer, wird der Text im Fenster der Show zentriert
Track Text y Position	100	Entfernung des linken Textrandes vom oberen Bildschirmrand (in Pixel). Wenn leer, wird der Text im Fenster der Show zentriert.
Track Text Justify	right	Left, right, center; Ausrichtung der Zeilen eines Textblocks
Transition	cut	Cut. Übergangstyp zwischen Tracks (aktuell nicht möglich).
Image Rotation	90	Dreht das Bild.
Image Window	Original 10 100 fit fit 10 10 40 40 fit ANTIALIAS	Ein Rahmen für ein Bild, in dem das Bild gezeigt wird. Siehe <a href="#">Kapitel 5.4.4.1</a>
Audio Player Audio	local	HDMI / A/V / USB zusätzlich HDMI0 / HDMI1 für Pi4. Wenn leer, wird das Gerät in der Raspbian Startleiste benutzt. Der frühere Wert von ‚local‘ wird von Pi Presents dargestellt, taucht aber im Editor Menü nicht mehr auf.
Audio Player Speaker	stereo	left/right/stereo – Lautsprecherkanal für Tracks die vom Mplayer in der Show gespielt werden
Audio Player Volume	100	Lautstärke der Audiotracks (0>100) die vom Mplayer gespielt werden
Audio Player Options		Weitere Optionen für den Mplayer (Vorsicht ist geboten, um Abstürze zu vermeiden)
MPV Player Audio	A/V	Hdmi / A/V / USB / USB2 / bluetooth in addition hdmi0 / hdmi1 für Pi4
MPV Player Volume	100	Lautstärke des Videotracks (0→ 100)

MPV Window	showcanvas	Einblendung für das Video. Werte siehe <a href="#">Kapitel 5.4.5.1</a>
MPV Aspect Mode	warp	<p>warp/original Skalierung des Videos im eingeblendeten Fenster:</p> <p>warp – das ganze Video ist im Fenster zu sehen. Wenn nötig wird das Seitenverhältnis angepasst.</p> <p>original – das ganze Video unter Beibehaltung des Seitenverhältnisses. Es können schwarze Balken auftreten, wenn das Seitenverhältnis von Video und Fenster nicht gleich sind.</p> <p>Anmerkung: Wenn das Seitenverhältnis von Fenster und Video nicht gleich sind, stimmen x- und y-Position nicht überein, ausgenommen im Fullscreenmodus.</p>
MPV Freeze at Start	no	<p>Einfrieren des Videos beim Trackstart</p> <ul style="list-style-type: none"> <li>• no – nicht einfrieren</li> <li>• before-first-frame – Video pausiert bevor der erste Frame gezeigt wird.</li> <li>• after-first-frame – Video pausiert nachdem der erste Frame gezeigt wird</li> </ul> <p>Zum Fortsetzen nach dem Einfrieren den ‚go‘-Befehl benutzen.</p>
MPV Freeze at End	yes	<p>yes/no – Wenn yes „friert“ das Video mit dem letzten Frame ein bis der nächste Track abspielbereit ist.</p> <p>- Wenn no, endet das Video normal.</p>
MPV Video Player Options		<p>Zusätzliche Kommandozeilenbefehle für den MPV Player, jeweils durch Komma getrennt. Die Optionen haben die Form option=value. Eine Option ist der Kommandozeilenbefehl ohne den vorangestellten Bindestrich.</p> <p>Optionen und Werte sind in PiPresents nicht getestet und könnten Abstürze des MPV Players verursachen.</p>
Chrome Web Window	showcanvas 200+200+600*400	<a href="#">Siehe Kap. 5.4.7</a>
Chrome Zoom	1.0	<a href="#">Siehe Kap. 5.4.7</a>
Chrome Freeze at End	yes	<a href="#">Siehe Kap. 5.4.7</a>
Chrome Options		<a href="#">Siehe Kap. 5.4.7</a>
<b>Controls Tab</b>		
Controls	Pp-play play pp-up up	Bezeichnet die Verknüpfung zwischen symbolischen Namen der Eingabeereignisse und Befehlen siehe vorige Tabelle und <a href="#">Kapitel 6.4</a>

		Eine Verknüpfung per Zeile, jede im Format symbolischer Name Befehl  Verknüpfungen werden nicht von Subshows übernommen
Disable Controls	no	yes/no – Wenn yes werden Befehle (z.B. play, pause, up, down, stop) außer Kraft gesetzt. Dies ist eine schnelle Methode, alle Befehle in einer concurrent show zu unterbinden. Für weitergehende Befehlsauswahlen ist das Controls-Feld zu wählen.
<b>Show Control Tab</b>		
Show Control at Beginning	open audio1	<a href="#">siehe Kapitel 8</a>
Show Control at End	close audio1	<a href="#">siehe Kapitel 8</a>
Show Control on Event	Pp-pause open audio1	<a href="#">siehe Kapitel 8.4</a>
Disable Show Control on Event	no	<a href="#">siehe Kapitel 8.4</a>

## 5.2.2 Menü

Eine Menü-Show benutzt die Titel und Miniaturfotos von Tracks und Shows, um automatisch ein Menü auf dem Bildschirm zu generieren. Mit den up/down-Befehlen kann man im Menü scrollen und mit dem play-Befehl einen Track oder eine Show starten.

Eine Menüshow benötigt einen zugehörigen Menü-Track in der Medialiste, der das Layout des Menüs festlegt:

- Bildschirm, Miniaturbilder, Schaltknöpfe oder Titel in vielen Kombinationen
- Vertikales und horizontales Einrichten des Menüs
- Einrichten von Menüs mit mehreren Spalten und Zeilen
- Ändern der Abstände, Farbe und Schriften der Menüpunkte

### 5.2.2.1 Steuerungen und Befehle

Menübefehle sind identisch mit den Befehlen für Mediasows wie sie im Einzelnen in Kapitel 5.2.1.1 beschrieben sind.

### 5.2.2.2 Felder

Field	Examples	Values
<b>Show Tab</b>		<b>Wichtige Informationen</b>
Type	menu	Kann nicht verändert werden
Title		Text beschreibt die Show und wird im Editor angezeigt.
Show Reference	mymenu	Bezeichnung, unter der andere Shows sich auf diese Show beziehen. Text ohne Leerzeichen.
Display	HDMI0	HDMI0, HDMI1 – Display, auf dem die Show und etwaige Subshows laufen sollen
Show Canvas		s. Mediashow
Medialist	mymenu.json	Dateiname der Medialiste, die die Tracks für das Menü enthält.
Show Timeout	59	std:min:sek – wenn es keine Menüaktivitäten gibt schließt Pipresents die Show automatisch. 0 = kein Timeout
Track Timeout	20	std:min:sek – Wenn 0-Tracks im Menü gestartet, werden sie nach dieser Zeit gestoppt und die Kontrolle wechselt zurück ins Menü.
Menu Track	menu-track	Die Track-Referenz des Tracks, mit dem der Menü-Inhalt dargestellt wird. Der Typ muss ein Menu Track sein.
<b>Eggtimer Tab</b>		Der Text des Eggtimers wird angezeigt, während der nächste Track geladen wird.
		s. Mediashow
<b>Show-Background-and Text Tab</b>		Show Text erscheint als Overlay auf allen Tracks, im Gegensatz zu Message Tracks, die vom Menü aus geöffnet werden.
		s. Mediashow
<b>Track Defaults Tab</b>		Tracks in einer Show benötigen einige Einstellungen, wenn sie nicht schon in den Trackeinstellungen enthalten sind.
		s. Mediashow
<b>Controls Tab</b>	pp-down down	Bezeichnet die Verknüpfung zwischen symbolischen Namen der Eingabeereignisse und Befehlen Eine Verknüpfung per Zeile, jede im Format symbolischer Name Befehl Verknüpfungen werden nicht von Subshows übernommen Siehe obige Tabelle und <a href="#">Kapitel 5.2.1.1</a>
Disable Controls	no	yes/no Wenn 'yes' dann sind die Befehle deaktiviert. Dies ist eine schnelle Möglichkeit, alle Steuerungen für konkurrierende Shows zu sperren. Für weitergehende Steuerung ‚Controls‘-Feld nutzen
<b>Show Control Tab</b>		
		s. Mediashow

### 5.2.3 Liveshow und Artliveshow

Eine Liveshow ist identisch mit einer Mediashow, außer das hier die Inhalte dynamisch von den Medienverzeichnissen bereitgestellt werden, den Live-Track-Verzeichnissen. Die Liveshow ist auf Videos, Bilder, Audiodateien und die Track Plugins beschränkt. Die Tracks aus den beiden Medienverzeichnissen werden zusammen abgerufen und bei `sequence=ordered` oder `reverse` nach dem Anfangsbuchstaben des Dateinamens sortiert.

Siehe [Kapitel 5.2.1](#) für Einzelheiten von Befehlen, Steuerungen und Feldern.

Der Liveshow muss eine Medialiste zugeordnet sein. Sie dient Child Track, List Empty Track und Escape Track, alle anderen Tracks werden ignoriert.

Aktionen der Liste von leeren Tracks:

#### a. List Empty/Escape

- Wenn das Repeat Feld der Liveshow ‚Single Run‘ enthält, endet die Liveshow
- Wenn das Repeat Feld der Liveshow ‚Repeat‘ enthält, und ‚List Empty Track‘ ist leer, wird eine Fehlermeldung gezeigt.
- Wenn das Repeat Feld der Liveshow ‚Repeat‘ enthält und ‚List Empty Track‘ keine Show ist, wird die Liveshow immer dann gestartet, wenn die Liste der zu spielenden Tracks leer ist. Der Track wird einmal gespielt. Danach wird der Inhalt der Liste neu ausgewertet.
- Der List Empty Track kann eine Show sein. Wenn das so ist, muß die Show ‚single run‘ sein, oder es muß eine andere Möglichkeit zur Beendigung der Show geben, damit die Liste der Liveshow Tracks neu ausgewertet werden kann. Wenn die Liste dann leer ist, wird der Escape Track einmal gestartet. Dieser Track muss keine Show sein. Sein Zweck ist es, das Stoppen oder Abbrechen einer Liveshow zu ermöglichen, die eine Subshow einer anderen Show ist.
- Diese Methode funktioniert eingeschränkt, weil empty track/show das gleiche Showfenster hat wie die Liveshow.

#### b. Show Control

- Die Felder ‚Show Control on Empty‘ und ‚Show Control on Not Empty‘ gestatten anderen Shows, geöffnet oder geschlossen zu werden, sofern zu spielende Tracks vorhanden sind.

Es ist unwahrscheinlich, dass beide Methoden zugleich in einer Anwendung vorkommen.

### Artliveshow

Artliveshows bieten ein lückenloses Abspielen. Es gibt keine Lücke zwischen den Tracks und sie „frieren“ am Ende nicht ein, um das Laden des nächsten Tracks zu überdecken. Für Artliveshows gelten einige Beschränkungen. Felder die in der Tabelle in [Kapitel 5.2.1.3](#) in Klammern gesetzt sind, fehlen der Artliveshow.

- Start, Next und End Triggers
- Child Tracks.

Die Liveshow hat folgende Felder, zusätzlich zu den Feldern der Mediashow

Field	Examples	Values
Live Tracks Directory 1		Der vollständige Verzeichnispfad der Tracks für diese Show. Wenn leer, werden die Tracks vom <code>pp_live_tracks</code> Verzeichnis in <code>pp_home</code> gestartet
Live Tracks Directory 2		Der vollständige Verzeichnispfad der Tracks für diese Show. Wenn leer, werden die Tracks im Verzeichnis, benannt mit der Befehlszeilenoption <code>-l</code> , gestartet. Kann für Synchronisation genutzt werden (siehe unten)
(List Empty Tracks)		Keine ArtLiveshow. Trackreferenz läuft, wenn die Trackliste einer Liveshow leer ist (siehe Text).
(Escape Track)		Keine ArtLiveshow. Trackreferenz läuft nach, wenn die Trackliste einer Liveshow leer ist (siehe Text).
<b>Show control Tab</b>		
		Siehe Mediashow
Show Control on Empty	open show1	<a href="#">siehe Kapitel 7.2</a>
Show Control on Not Empty	open show2	<a href="#">siehe K7.2 Shows öffnen und schließen apitel 7.2</a>

### Synchronisation beim Hochladen von Livetracks

Standardmäßig wird Pi Presents gestoppt, wenn eine Livetrackliste aktualisiert wurde. Grund dafür ist, dass ein Track geändert wird, während er möglicherweise gespielt wird. Die Synchronisation ermöglicht es, Livelisten zu aktualisieren, während die Liveshow läuft. Das wird mit einem Cacheverzeichnis (Zwischenspeicher) auf dem Raspberry Pi bewirkt, in welches die Livelist hochgeladen wird, um von dort Pi Present mit einer Sperre neue Dateien anzukündigen. Pi Presents erkennt die Sperre und verschiebt die Dateien in die Liveliste, wenn es beim Start die Sperre erkennt und sich zwischen zwei Tracks befindet. Der Operator muss diese Sperre nutzen, um sicherzustellen, dass der Cache nicht während eines laufenden Dateitransfers geändert wird.

Um die Synchronisation zu aktivieren, muss in der Datei `pp_filelistfetcher.py` die Einstellung geändert und das Cacheverzeichnis eingerichtet werden. Weitere Hinweise gibt es in der o.a. Datei. Eine Synchronisation ist lediglich mit dem Live Tracks Directory 2 möglich

### Symbolische Links benutzen (Symlinks aus Github)

Ich habe den Ordner „available media“, der alle Medien für die Show enthält, einfach nach „live\_tracks\_cache“ verlinkt (von dort werden sie nach „live tracks“ kopiert – nach Touch auf „lock file“).

Das bewahrt vor allzu vielen großen Schreibvorgängen auf der SD-Card durch häufig sich ändernde Zugriffe bei den gespielten Medien (weil symlinks kleiner sind, als eine komplette Datei), was zu längerer Lebensdauer der SD-Card führt.

### Einsatz von Rclone (von Matt)

Rclone zusammen mit One Drive ist eine großartige Ergänzung zum Liveshow-Feature, wenn es um Netzlaufwerke und gemeinsam genutzte Laufwerke geht.

Auf Grund von kürzlich gemeldeten 403-Abstürzen versuchten wir, ein One-Drive-Verzeichnis für eine Liveshow mit UZBL zu benutzen. Allerdings gab es immer wieder Netzprobleme. Die Lösung war der Einsatz von rclone, mit dem ein autosynchroner One-Drive-Ordner mit dem Pi verknüpft wird und aus dem Pi Presents sich für die Liveshow „bedient“. Pi Presents erneuert stündlich das autosynchrone Verzeichnis und verhindert Datenverlust beim Löschen und Hinzufügen.

### 5.2.4 Radiobuttonshow

Eine Radiobuttonshow verfügt über Präsentationsmöglichkeiten, wie sie in vielen Kiosks üblich sind:

Startet mit einem Einstiegsbildschirm, der einen Text enthalten kann, der den Benutzer zum Druck auf einen Button oder einen Touch auf dem Bildschirm auffordert, wodurch ein Track gestartet wird. Beim Abspielen des Tracks startet ein Druck auf einen anderen Button einen anderen Track. Am Ende eines Tracks oder durch Druck auf Stopp erfolgt die Rückkehr zum Einstiegsbildschirm.

In Pi Presents kann der Einstiegsbildschirm ein Bild, eine Nachricht, ein Video, ein Audiotrack oder eine Show sein. Weitere Tracks werden nach Maßgabe der Play Befehle im ‚Controls‘ Feld der Radiobuttonshow gespielt, z.B.:

```
but1 play myimagetrack
but2 play myvideotrack
```

#### 5.2.4.1 Steuerungen und Befehle

Jede Steuerung hat drei durch Leerzeichen getrennte Felder:

- symbolic name – der symbolische Name der Eingabe-Aktion, die Shows oder Tracks abspielen.
- command – siehe nachfolgende Tabelle.
- track – die Track-Referenz (Speicherort) oder Showreferenz des zu spielenden Tracks oder der Show.

Command	Argument	Beschreibung
play	track-ref	Spielt den Track mit der entsprechenden Track-Referenz
return	<blank>	Zurück zum ersten Track der Show
stop	<blank>	Stoppt den aktuellen Track. Wenn die Show sich im Ruhezustand befindet und sie keine Top-Level_Show ist, endet die Show. Ruhezustand = wenn der erste Track gezeigt wird.
go	<blank>	Pausenende eines Videos ‚frozen at start‘
exit	<blank>	Beendet die Radiobuttonshow
null	<blank>	Verhindert die Steuerung mit einem gleichen symbolischen Namen, wie er in der Show definiert ist.
pause	<blank>	Bewirkt eine Pause bei Video-, Audio- und Imagetracks.
pause-on	<blank>	Pause für Tracks, die Pause unterstützen
pause-off	<blank>	Pausenende für Tracks, die Pause unterstützen
mute	<blank>	Stumm für Tracks, die mute unterstützen

unmute	<blank>	Laut für Tracks, die mute unterstützen
inc-volume	<blank>	Für MPV-Videotracks: Erhöht Lautstärke um einen Schritt
dec-volume	<blank>	Für MPV-Videotracks: Verringert Lautstärke um einen Schritt.
mplay-*	<blank>	Führt eine Laufzeitunterbrechung für den Videoplayer aus ( <a href="#">siehe Kapitel 12</a> )

Befehle können in den Kontrollfeldern der Tracks zusätzlich zu den Kontrollfeldern der Radiobuttonshow eingesetzt werden. Die Steuerungen in den Tracks werden herangezogen. Sie überschreiben den Eintrag in der Show. Das ist in der Regel nicht erforderlich, wird aber benötigt, wenn beispielsweise das Publikum den ganzen Film sehen soll. In diesem Fall verhindert der ‚null‘-Befehl, dass ein anderer Track unter diesem symbolischen Namen gestartet oder gestoppt wird.

#### 5.2.4.2 Felder

Field	Examples	Values
<b>Show Tab</b>		
Type	radiobuttonshow	Kann nicht editiert werden
Title	my Show	Beschreibt die Show, die in Editor und Menü angezeigt wird
Show Reference	myradioshow	Bezeichnung, unter der die Show von anderen Shows verwendet wird. Text ohne Leerzeichen
Display	HDMI0	HDMI0, HDMI1 – Der Bildschirm, auf dem die Show oder Subshowa angezeigt werden
Show Canvas		s. Mediashow
Medialist	mymedia.json	Name der Medialist-Datei, in der die Tracks der radiobuttonshow aufgeführt sind. Alle Tracks sollten eine Track-Referenz besitzen
First Track	myfirsttrack	Die Track-Referenz des Tracks, der den Startbildschirm der Show zeigt
Show Timeout	1:59	h:m:s, wenn es beim ersten Track keine Aktivität gibt, kehrt PiPresents automatisch zur vorherigen Show zurück. 0 steht für kein Timeout
Track Timeout	20	h:m:s, Wenn Tracks mit einer Zeitangabe von der radiobuttonshow gestartet sind, werden sie nach dieser Zeit gestoppt und die Show übernimmt wieder die Kontrolle

Field	Examples	Values
Controls in Subshows	no	yes/no no - (reguläre Radiobuttonshow Operation) startet während des Abspielens eines Tracks den ausgewählten Track. Jedoch können Tracks als Show so nicht gesteuert werden. Beispielsweise muss eine Mediashow single-run laufen. yes – Wenn der zu spielende Track ein Show-Track ist, werden Eingaben an die Show weiter gereicht. Dadurch kann die Show gesteuert werden (mit ihren eigenen Controls), aber es ist nicht möglich, einen Track zu starten, während die Show läuft.
<b>Eggtimer Tab</b>		
		s. Mediashow
<b>Show Background und Text</b>		
		s. Mediashow
<b>Track Defaults Tab</b>		
		s. Mediashow
<b>Controls Tab</b>		
Controls	myname play mytrack	Legt die Verknüpfungen zwischen symbolischen Namen von Ereignissen und Befehlen fest  Eine Verknüpfung in jeder Zeile mit dem Format:  symbolic-name command [arg]  Verknüpfungen werden von Subshows <u>nicht</u> geerbt  Befehle s. Obige Tabelle
Disable Controls	no	Yes/no Bei „Yes“ sind alle Befehle außer Kraft. Eine schnelle Möglichkeit, alle Befehle für eine zweite (concurrent) Show zu blockieren Für eine weitergehende Kontrolle stehen die Controls Felder zur Verfügung
<b>Show Control Tab</b>		
		Siehe Mediashow

### 5.2.5 Hyperlinkshow

Eine Hyperlinkshow bietet Show-Eigenschaften an, wie sie auf Touchscreens in Museen benutzt werden:

'Start mit einer Einstiegsseite mit etwas einführendem Text, Video oder Bildern so wie einer Auswahl von Displaybuttons, die es dem Benutzer ermöglichen, zu anderen Seiten zu

wechseln. Jede Seite kann über eine andere Buttonauswahl verfügen, um andere Seiten aufrufen zu können. Auf allen Seiten, außer der Startseite, erlauben zusätzliche Buttons, auf vorherige Seiten zurückzugehen oder die Startseite wieder aufzurufen.'

Alle Touchscreen-Displays, die ich untersucht habe, scheinen in dieser Weise zu funktionieren, es sei, dass es sich um Radiobuttonshows handelt.

Jede Seite im obigen Beispiel ist ein Track einer Hyperlinkshow. Jede Seite enthält ein Kontrollfeld mit Befehlen, mit denen die Bewegung zwischen Tracks und zurück ermöglicht wird.

z.B. könnte track story1b folgende Befehlsstränge enthalten:

```
next-name call story1c
alternative-name call alternative1
back-name return
home-name home
```

xxx-name ist der symbolische Name einer Eingabe. Das heißt in diesem Fall, gehe entweder vorwärts zu story1c oder zu alternative1, zurück zum vorherigen Track (möglicherweise story1a) oder zurück zum Home-Track, was eventuell heißt, zurück über story1a, ohne diesen Track anzuzeigen.

Wenn der 'call'-Befehl ausgeführt wird, merkt sich Pi Presents, über welchen Pfad der Befehl gekommen ist (zumeist ein Ordnerstapel) So kann der 'return'-Befehl eine Seite zurück gehen, indem die aktuelle Seite im Pfad gelöscht wird, vergleichbar mit verschachteltem Aufruf und Zurück (return) bei Unterprogrammen in Programmiersprachen.

Es gibt eine besondere Funktion genannt Home Track. Das Kommando ‚home‘ kann längs des gewählten Pfades zurückspringen bis zum Home Track. Es ist nicht mehr nötig, den genauen Pfad zu einem bekannten Startpunkt zu kennen.

Jeder Befehlssatz hat drei Felder, getrennt von Leerräumen:

- symbolic name – der symbolische Name der Eingabe-Aktion, die den Befehl 'command' steuert
- command - call, return, goto, jump, exit, null, repeat, pause, no-command und runtime controls.
- track - die Track-Referenz des Tracks, der gespielt werden sollen.

### 5.2.5.1 Steuerungen und Befehle

Command	Argument	Effect
call	<Track-Referenz>	spielt den Track (nach Reference) und merkt sich den Pfad
return	<blank>	return 1 ruft den Pfad auf, entfernt daraus den Track und endet bei Home Track
return	number	return n tracks ruft den Pfad auf, entfernt daraus den Track und endet bei Home Track
return	<Track-Referenz>	return - zur angegebenen Track-Referenz, entfernt die Tracks aus dem Pfad und geht über Home Track, wenn erforderlich, zurück
home	<blank>	return – folgt dem Pfad zum Home Track, vorherige

		Tracks werden aus dem Pfad entfernt
jump	<Track-Referenz>	play - Track-Referenz, löscht den Pfad zurück zum Home Track
goto	<Track-Referenz>	play - Track-Referenz, 'vergisst' den Pfad
repeat	<blank>	Wiederholt den aktuellen Track ohne ein Zeitintervall zurückzusetzen
go	<blank>	Pausenende für Tracks mit ‚frozen at start‘
null	<blank>	Inhibits – sperrt den Link mit dem gleichen symbolischen Namen, der in der Show eingetragen ist.
exit	<blank>	end – beendet die Hyperlinkshow
pause	<blank>	Bewirkt eine Pause bei Video-, Audio- und Imagetracks.
pause-on	<blank>	Pause für Tracks, die ‚pause‘ unterstützen
pause-off	<blank>	Pausenende für Tracks, die ‚pause‘ unterstützen
mute	<blank>	Still für Tracks, die ‚mute‘ unterstützen
unmute	<blank>	Laut für Tracks, die ‚mute‘ unterstützen
no-command	<blank>	Ein Klick bleibt ohne Wirkung. Wird für eine Legende von Softkeys benutzt, um einen nicht aktiven Klickbereich anzuzeigen.
inc-volume	<blank>	Für MPV-Videotracks: erhöht die Lautstärke um einen Schritt
dec-volume	<blank>	Für MPV-Videotracks: vermindert die Lautstärke um einen Schritt
mplay-*	<blank>	Ausführung eines Runtimebefehls für den Audioplayer ( <a href="#">Kapitel 12</a> )

Im Kontrollfeld der Hyperlinkshow können Befehle zusätzlich zum Links Feld der Tracks gespeichert werden. Das erspart Schreibarbeit, weil die Befehle in der Show in jedem Track genutzt werden und die Befehle der Tracks mit ihnen zusammengelegt werden. Das ist manchmal nicht erwünscht, so kann man mit dem 'null' Befehl im Track den Befehl mit dem gleichen symbolischen Namen löschen.

Der 'goto' Befehl „weiß“ nicht, wo er herkam. Er wird für besondere Umstände wie zum Beispiel timeout oder pp-onend benötigt. Er ist zudem eine Alternative zu 'call/return', um einen Zurück-Button einzufügen; zurück wird dadurch zu 'goto'. Es wäre nicht gut, im gleichen Teil einer Hyperlinkshow 'call/return' und 'goto' zu vermischen.

Wenn ein Track, wie video oder timed image, von selbst endet, muss Pi Presents einen Befehl erteilen, der angibt, was als Nächstes erfolgen soll. Das wird in Pi Presents durch eine interne Eingabe mit dem symbolischen Namen pp-onend erreicht. Das Feld Controls hat damit einen Befehl, der ausgeführt werden kann, z.B.

```
pp-onend goto mynextpage
oder
pp-onend repeat
```

Befehle mit pp-onend beschränken sich auf exit, home, return, call, goto, jump und repeat. Ein zweiter spezieller Track ist der First Track. PiPresents beginnt die Show immer hier. In den meisten Anwendungen haben Home Track und First Track die gleiche Track-Referenz, sie können dennoch unterschiedlich sein. Das ist so eingerichtet, dass First Track die Show mit einem Bild oder Video startet, um den Benutzer dazu zu verlocken, mit einem Touch auf einen Button zum

Home Track zu springen, um interaktiv eine Reihe von Tracks zu starten. Einmal verpasst würde home return oder jump den User nicht zurück zum First Track bringen, was der Track timeout jedoch kann.

Um sicher zu stellen, dass das Bild zum First Track zurückspringt, wenn der Benutzer das Bild verlässt, gibt es den Track timeout. Wenn timeout eingeschaltet ist, geht die Hyperlinkshow zum Timeout Track. Dieser Track hat die gleiche Track-Referenz wie der First Track, aber ein eigenständiger Timeout Track erlaubt zum Beispiel das Beenden, bevor es zum First Track geht.

Tracks in Hyperlinkshows können Shows sein. Sie haben ihren eigenen Satz von Befehlen inklusive stop, mit dem es zurück zur Hyperlinkshow geht.

Beim Entwickeln einer Hyperlinkshow ist es oft nützlich, den Pfad für die Webseiten zu sehen, den Pi Presents sich gemerkt hat. Dafür muss man Print Debug Path benutzen.

### 5.2.5.2 Touchscreens und Softbuttons

In Hyperlinkshows sind für jeden Track unterschiedliche Steuerungssätze erforderlich. Pi Presents bietet dafür zwei Varianten an:

- **Klickbereiche**  
Klickbereiche sind typisch für Touchscreens. Ein Klickbereich ist ein maussensitiver Bereich auf dem Display. Obwohl sie Klickbereiche heißen, lassen sie sich viel besser per Touch auf dem Bildschirm benutzen. Sie sollten deshalb besser Touchbereiche heißen. Ein Touch auf dem Bildschirm oder ein Klick mit der Maus im Klickbereich löst ein Ereignis aus, dass mit einem symbolischen Namen in der screen.cfg belegt ist. Klickbereiche können durch ein Bild oder einen Button dargestellt werden, wenn der Klickbereich eingerichtet wird. Andererseits kann ein Klickbereich auch transparent sein und als Vieleck den Umriss eines dahinter liegenden Objekts angeben.
- **Soft Buttons**  
Soft Buttons sind der Touchscreen für Arme. Sie wurden häufig in Echtzeitsteuerprogrammen eingesetzt, bevor es Touchscreens gab.

Soft Buttons sind „unmarked keys“ oder Buttons am Rand eines Bildschirms. Die Bedeutung der Buttons wird per Software festgelegt und erscheint in der Nähe der Buttons auf der Bildfläche. Diverse Buttons und ihre Zugehörigkeit können jedem Track einer Hyperlinkshow hinzugefügt werden.

Pi Presents ermöglicht beide Techniken, um Hyperlinkshows zu steuern, wie in [Kapitel 10](#) beschrieben.

### 5.2.5.3 Felder

Field	Examples	Values
<b>Show Tab</b>		
Type	hyperlinkshow	Kann nicht geändert werden
Title	My Hyperlinkshow	Text beschreibt die Show. Wird im Editor und in Menüs angezeigt

Field	Examples	Values
Show Reference	myhyperlinkshow	Eintrag, wie die Show mit anderen Shows verknüpft ist. Text ohne Leerzeichen
Display	HDMI0	HDMI0, HDMI1 – Display auf dem Shows und etwaige Subshows zu sehen sein sollen.
Medialist	mymedia.json	Dateiname der Medialiste, die die Tracks für die Hyperlinkshow enthält. Alle Tracks sollten eine Track-Referenz haben
First Track	myfirsttrack	Die Track-Referenz des Tracks, der die Einstiegsshow der Show enthält
Home Track	myhometrack	Die Track-Referenz des Tracks, der den Beginn des call/return-Pfades bezeichnet
Timeout Track		Dieser Track wird angezeigt, wenn ein Timeout geschieht. Pi Presents geht nicht sogleich zum Home-Track, weil eventuell Animationen oder konkurrierende Shows gestoppt werden sollen  Normalerweise macht dieser Track, was immer auch gefordert wird, – 'goto' zum First- oder Home-Track nach einer kurzen Zeit. Wenn es ein Audiotrack ist, kann er auf 'no sound' und eine 'zero duration' eingestellt werden
Show Timeout	60	h:m:s – wenn auf dem Home-Track nichts geschieht, beendet Pi Presents automatisch die parent show, allerdings nicht, wenn es sich um eine top-level-show handelt. 0 = kein Timeout
Track Timeout	30	h:m:s - Wenn kein Benutzerbefehl anliegt oder der Track nicht vor dem Timeout sowieso endet, erfolgt die Rückkehr zum Timeout Track. Ein Wert von 0 deaktiviert die Timeout-Funktion
Show Canvas		s. Mediashow
<b>Eggtimer Tab</b>		
		s. Mediashow
<b>ShowBackground and Text Tab</b>		
<b>Track Defaults Tab</b>		
<b>Controls Tab</b>		

Field	Examples	Values
Disable Controls	no	yes/no - yes/no Wenn 'yes' dann sind die internen Aktionen (play, pause, up, down, stop) und Laufzeitsteuerungen deaktiviert. Dies ist eine schnelle Möglichkeit, alle Steuerungen für konkurrierende Shows zu sperren. Für weitere Steuerungsauswahl steht das Controls-Feld zur Verfügung
Controls	name call mytrack	Benennt die Verknüpfung der symbolischen Namen von Eingaben und Befehlen. Ein Steuerungssatz per Zeile jeweils im Format: symbolic-name command Verknüpfungen werden von Subshows NICHT übernommen Siehe obige Tabelle mit einer Liste von Steuerungen
<b>Show Control Tabs</b>		
s. Mediashow		

### 5.2.6 Start Show

Die Startshow darf nicht gelöscht werden. Sie ist eigentlich keine Show, sondern der Teil eines Profils, die Dinge regelt, die für alle Shows gelten.

Die Start Show benennt die Shows, die laufen sollen wenn Pi Presents startet. Jede Show läuft gleichzeitig mit den anderen, Eingabe-Aktionen werden an all diese konkurrierenden Shows geleitet. Das Feld Start Shows kann leer sein, dann müssen Shows anderweitig gestartet werden.

Ob der Zeitplan eingerichtet ist oder die Planung anderer Dinge - es sind Regeln für alle Shows\_ ([siehe Kap. 7.3](#))

Kommandos für die simulierte Zeit, mit der ein Zeitplan getestet werden kann ([siehe Kap. 7.3](#))

Hintergrundfarbe, die in Shows gezeigt wird.

Field	Examples	Values
<b>Schedule Tab</b>		
Start Shows	show1 show2	Liste von Show-Referenzen, durch Leertaste getrennt. Diese Shows werden gestartet, wenn Pi Presents startet
Enable Scheduler	no	yes/no aktiviert Zeitschalter Auf no stehen lassen, wenn kein Zeitschalter gesetzt ist ( <a href="#">siehe Kap. 7.3</a> )
		Siehe <a href="#">Kapitel 7.3</a> für zusätzliche Felder
<b>Simulate Time-Tab</b>		
Simulate Time		yes/no Eine beliebige Zeit und Datum eintragen um den Zeitschalter zu testen no wenn kein Zeitschalter beabsichtigt ist ( <a href="#">siehe Kap. 7.3</a> )
		Siehe <a href="#">Kapitel 7.3</a> für zusätzliche Felder

<b>Background Tab</b>		
Background Colour		Farbe des Hintergrundes wenn keine Show läuft
<b>Counters Tab</b>		
Store Counters	no	Bei ‚yes‘ werden Counter und Werte zwischen den Läufen von Pi Presents gespeichert
Initial Values	fred=1 bill=2	Wenn Speichern aktiviert ist und die Befehlszeile die option –counters enthält, werden die Counter (Zähler) durch dieses Feld benannt.
<b>Show Tab</b>		
type	start	Muss immer „start“ sein, nicht veränderbar
title	start show	Textbeschreibung der Show, die im Editor angezeigt wird
Show Reference	start	Muss „start“ sein, nicht veränderbar

### 5.3 Medialisten

Medialisten sind vergleichbar mit Playlisten eines Mediaplayers. Sie legen fest, welche Tracks abgespielt werden sollen. Wie die Tracks abgespielt werden sollen, regeln die Shows, die mit den Medialisten verbunden sind.

Jeder Eintrag in einer Medienliste ist ein Track. Tracks können unterschiedliche Typen sein:

- image – ein Einzelbild.
- video – ein Track, der vom OMXPlayer abgespielt wird.
- audio – ein Audiotrack, der vom MPplayer abgespielt wird.
- message – zeigt Textzeilen vor einem farbigen Hintergrund, ein Hintergrundbild ist möglich.
- web – startet einen Browser für die Darstellung einer URL
- show – Shows können als Tracks angesehen werden, die sich auf vielen Ebenen verzweigen können.
- menu-track – ein spezieller Track, mit dem das Format eines Menüs festgelegt wird.

Eine Medialiste ist in der Regel mit einer einzigen Show verbunden. In Pi Presents sind sie jedoch von den anderen Show-Merkmalen getrennt, so dass die gleiche Medialiste von zwei unterschiedlichen Shows benutzt werden kann.

### 5.4 Tracks

Jeder Track-Typ besitzt Felder, die sein Abspielen beschreiben, einige dieser Anweisungen überschreiben die entsprechenden Felder in der zugehörigen Show.

Der Typ eines Tracks beschreibt seinen vorrangigen Inhalt. Vorrangiger Inhalt kann vor einem Bild oder einem einfarbigen Hintergrund stehen und enthält Textanmerkungen.

- image – ein Einzelbild. Erlaubte Bildformate sind Bilder, die von der Python Imaging Library gerendert werden können. Für optimale Darstellung sollte die Bildgröße auf etwa ein Megapixel limitiert werden.

- video-mpv – ein Track, der vom MPV Mediaplayer abgespielt wird.
- audio – ein Audio-Track, der vom MPlayer abgespielt wird. Spielbar ist jedes Format, das der MPlayer kennt. Audio-Track ist sehr flexibel, weil er über ausführliche Möglichkeiten verfügt, was ihn besonders geeignet für Animationssequenzen oder Shows macht.
- message – zeigt Textzeilen. Message kann zudem benutzt werden, um einen leeren Bildschirm zu erzeugen. Message-Tracks bieten einfache Folien-Funktionen an. Wenn es etwas mehr sein soll, kann man Impress (Libre Office) oder Powerpoint benutzen, die Präsentation als .jpg exportieren und als 'Image' anzeigen.
- chrome – zeigt eine URL in einem Browser. Die Browserintegration ist in Pi Presents nicht optimal, deshalb sollte dieser Track-Typ mit Vorsicht eingesetzt werden. Es gibt sehr viele andere browsergestützte Digitale Signagesysteme.
- show – Shows können als Tracks eingefügt werden. Erlaubt ist eine unbegrenzte Verzweigung in die Tiefe.
- menu – dient der Einrichtung eines Menü-Layouts für Menüs.

#### 5.4.1 Track-Speicherorte

Track-Speicherorte, die Dateien benennen, können relativ oder absolut sein. Mit relativen Dateinamen sind Profile portierbar. ([Siehe Kapitel 4.2.1](#))

- Das vorangestellte + Zeichen im Dateipfad bezeichnet den Track-Speicherort relativ zum /pp\_home Verzeichnis. Wenn das Pluszeichen fehlt, muss der Pfad absolut angegeben werden. Es ist ratsam, Mediadateien in /pp\_home zu speichern, wenn Profile portierbar sein sollen.
- Ein führendes @ in Dateipfaden erlaubt die Trackbenennung relativ zum jeweiligen Profilverzeichnis, z.B. @/mymedia/myfile.jpg greift auf das /mymedia-Verzeichnis im aktuellen Profil zu.
- Absolute Pfade sind angebracht, wenn Internetadressen eingetragen werden sollen, z.B.: [http://www.mysite.com/track\\_to\\_play.mp4](http://www.mysite.com/track_to_play.mp4)
- 

#### 5.4.2 Unbenannte und benannte Tracks

Einige Medialisteneinträge haben Bezeichnungen, die im Feld Track-Referenz beschrieben sind. Wenn Track-Referenz leer ist, ist der Track Bestandteil eines Menüs, einer Mediashow oder einer Liveshow – sie werden unbenannte (anonyme) Tracks genannt. Wenn das Feld nicht leer ist, kann der Track durch Befehle hinter Radiobuttonshows oder Hyperlinks oder für einen besonderen Gebrauch angesprochen werden. Besonderer Gebrauch meint:

- als Child Show einer Mediashow oder einer Liveshow.
- Als Menü-Track einer Menüshow
- als First-, Home- and Timeouttracks in Hyperlink- oder Radiobuttonshows.

Die Felddefinitionen für jeden Typ von Medialisten-Tracks folgen im nächsten Kapitel.

### 5.4.3 Show Track

Die Tracks einer Show können selbst Shows sein. Sie heißen sub-shows. Showtracks erlauben das Einfügen von Sub-Shows in die Medialiste. Das Feld 'Show To Run' benennt die Show-Referenz der Show.

### 5.4.4 Image Track

Image Tracks werden von der Python Imaging Library gerendert. (Siehe empfohlene Bildgröße in den Hardwareanforderungen). Image Tracks pausieren mit dem Pause-Befehl. Bilder können in einem Fenster erscheinen, können gedreht werden, sie können ein Bild oder eine Farbe als Hintergrund benutzen und können von einem Text überlagert sein. Text überlagert einzelne Tracks, Show Text einzelne Shows.

Field	Example	Values
<b>Track Tab</b>		
Type	image	
Title		Wird in einem Menü und im Editor angezeigt
Track Referenz		Kennzeichnung eines Tracks – leer, wenn der Text innerhalb eines Menüs oder einer Mediashow läuft. Nicht leer, wenn der Track mit einem symbolischen Namen gesteuert wird oder ein spezieller Track ist
Location	+/media/ sarename.gif	Der Dateiname des Tracks, der auch leer sein kann
Thumbnail	+/media/ sarename_tn.jpg	Dateiname eines Thumbnails in einer Menüshow
Duration	5	Sekunden – in Schritten von 0.1 secs. Wenn '0' wird das Bild angezeigt und durch eine Benutzereingabe beendet. Wenn leer, wird der Wert in der Eltern Show benutzt
Transition	cut	cut. (not used).
Rotation	90	90 Grad. Wenn leer, wird der Wert in der Eltern Show benutzt
Image Window	original 10 100 fit fit 1010 40 40 fit ANTIALIAS	Ein Sichtfenster, in dem ein Bild gezeigt wird. Wenn leer, wird der Wert in der Eltern Show benutzt, ( <a href="#">siehe Kap. 5.4.4.1</a> )
Background Image	+/media/ image.jpg	Der Dateiname des Bildes, das als Hintergrund für das Hauptbild genutzt wird. Wenn leer, wird der Wert in der Eltern Show genutzt
Display Show Background Image	yes	yes,no yes – das Hintergrundbild der Elternshow wird angezeigt, wenn der Track 'Background Image' leer ist. no – das Hintergrundbild der Elternshow wird nicht angezeigt
Background Colour		Schriftfarbe im Auswahlfenster wählen, erzeugt eine sechsstellige Hexadezimalzahl, oder Farbnamen direkt eintragen Siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> Wenn leer, wird der Wert der Elternshow benutzt
Plugin Configuration File	+/media/krt- time.cfg	Datei enthält den Namen und die Parameter des Track-Plugins
Pause Timeout	5	Sekunden - wenn nicht leer und größer 0, wird ein pausierender Track automatisch nach dem Timeout wieder gestartet

Field	Example	Values
<b>Control Tabs</b>		
Controls	myname null	Verknüpfung symbolischer Namen mit Befehlen. Die für einen Track hier festgelegten Verknüpfungen überschreiben die Befehle der zugehörigen Show. ( <a href="#">siehe Kapitel 8.1</a> )
<b>Text Tab</b>		
Track Text	Picture of Taj Mahal	Wenn nicht leer, wird der mit dem Bild angezeigte Text in den Vordergrund auf das Bild gestellt.
Track Text Location	+/media/ mytextfile	Wenn nicht leer, hat die Datei mit dem Track-Text Vorrang vor dem Track Text Feld.
Track Text x Position	100	Linker Textabstand vom linken Bildschirmrand (Pixel). Wenn leer, wird der Text im Showfenster zentriert
Track Text Type	html	Plain - einfacher Text html – eine Zusammenfassung von Text, Bildern und Formatierungen in HTML-Fassung ( <a href="#">siehe Kap. 5.4.4.2</a> ).
Plain Text Font	Helvetica 30 bold	Eine Schrift mit der Schriftauswahl wählen oder Schrift direkt eintragen Siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Plain Text Colour	white	Schriftfarbe im Auswahlfenster wählen, erzeugt eine sechsstellige Hexadezimalzahl, oder Farbnamen direkt eintragen. Siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Plain Text Justify	right	Left, right, center – Richtet Zeilen im Textblock aus
HTML Text Height	300	Höhe des HTML-Textfeldes
HTML Text Width	300	Breite des HTML-Textfeldes
HTML Background Colour	##f0000	Hintergrundfarbe des HTML-Textfeldes
Display Show Text	yes	Yes/no Erlaubt oder sperrt die Anzeige von Showtext in diesem Track.
Pause Text	Paused....	Gezeigter Text während der Pause.
Pause Text Font	Helvetica 30 bold	Eine Schrift mit der Schriftauswahl wählen oder Schrift direkt eintragen  Siehe: <a href="https://web.archive.org/web/20200315112731/">https://web.archive.org/web/20200315112731/</a>

Field	Example	Values
		<a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Pause Text Colour	white	Schriftfarbe im Auswahlfenster wählen, erzeugt eine sechsstellige Hexadezimalzahl, oder Farbnamen direkt eintragen. Siehe: <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Pause Text x Position	100	Entfernung des linken Textendes vom linken Bildschirmrand (Pixel). Wenn leer, wird der Text im Showfenster zentriert.
Pause Text y Position	100	Entfernung des oberen Textendes vom oberen Bildschirmrand (Pixel). Wenn leer, wird der Text im Showfenster zentriert.
Pause Text Justify	right	Left, right, center – Richtet Zeilen im Textblock aus
<b>Show Control Tab</b>		
Show Control at Beginning	open audio1	<a href="#">(siehe Kapitel 7.2)</a>
Show Control at End	close audio1	<a href="#">(siehe Kapitel 7.2)</a>
<b>Animation Tab</b>		
Animation at Beginning	1 out1 state on 2 out1 state off 1 out2 state on 6 out3 state on	<a href="#">(siehe Kapitel 15)</a>
Clear Animation	no	yes/no - <a href="#">(siehe Kapitel 15)</a>
Animation at End	0 out2 state off	<a href="#">(siehe Kapitel 15)</a>

#### 5.4.4.1 Image Window

Image Window (Bildfenster) steuert die Positionierung eines Bildes auf der Bildschirmfläche. Es enthält eine Reihe von Feldern, die durch Leerräume getrennt sind. Wenn Image Window leer ist, wird der Wert der Show benutzt.

Das erste Feld enthält original, shrink, fit oder warp

- original  
Das Bild wird in seiner Originalgröße gezeigt. Dieser Befehl kennt zwei Argumente, die x,y, Koordinaten der oberen linken Ecke des angezeigten Bildes. Wenn die Argumente fehlen, wird das Bild zentriert auf dem Bildschirm platziert.

z.B. original, original 100 100

- fit

Das Bild wird im voreingestellten Fenster eingepasst und behält dabei sein Seitenverhältnis. Das Bild wird wie gewünscht geschrumpft oder gestreckt. Der Befehl verfügt über vier Argumente, die x,y, - Koordinaten des angezeigten Bildes: die obere linke und die untere rechte Ecke. Wenn keine Argumente gesetzt sind, wird das Bild auf der Bildschirmfläche zentriert.

z.B. fit, fit 100 100 1000 500. Statt 2x,y,-Paaren kann x1+y1+w\*h eingesetzt werden

- shrink  
gleich mit fit, Ausnahme: das Bild wird nicht vergrößert, wenn es kleiner als das Fenster ist.

z.B. shrink, shrink 100 100 1000 500. Statt 2x,y,-Paaren kann x1+y1+w\*h eingesetzt werden

- warp  
Das Bild erscheint eingepasst in das eingerichtete Fenster ohne sein Seitenverhältnis beizubehalten. Das Bild wird, wie gewünscht, geschrumpft oder vergrößert. Der Befehl verfügt über vier Argumente: die x,y, - Koordinaten der oberen linken und der unteren rechten Ecke des angezeigten Fotos. Wenn keine Argumente gesetzt sind, wird das Bild auf volle Bildschirmgröße gestreckt.

z.B. warp, warp 100 100 1000 500, warp 100 100 1000 500 BICUBIC  
Statt 2x,y,-Paaren kann x1+y1+w\*h eingesetzt werden

Für fit, shrink und warp kann ein optionaler Filter als fünftes Argument gesetzt werden, eines von NEAREST, BILINEAR, BICUBIC, ANTIALIAS siehe:

<https://web.archive.org/web/20200315112731/http://effbot.org/imagingbook/image.htm>

oder

<http://effbot.org/imagingbook/image.htm>

Miniaturbild und Vergrößerungs-Kapitel. Wenn kein Argument gesetzt ist, wird NEAREST benutzt.

e.g.: fit 100 100 1000 500 BICUBIC.

Statt 2x.y,-Paaren kann x1+y1+w\*h eingesetzt werden

#### 5.4.4.2 HTML und einfacher Text

Message-Text, Track-Text und Show-Text können einfacher Text oder HTML sein.

Plain – Textfarbe, Schriftart und Ausrichtung können gewählt werden. Die Overlay-Texte haben einen durchsichtigen Hintergrund.

Html – Text ist Bestandteil von HTML. Er steht in einem rechteckigen Fenster mit undurchsichtigem Hintergrund, der nur von Videos überlagert werden kann. Höhe, Breite und Hintergrund des Textfeldes können eingestellt werden.

HTML-Text erfordert tk\_html\_widgets, um den Text darstellen zu können:

[https://github.com/paolo-gurisatti/tk\\_html\\_widgets](https://github.com/paolo-gurisatti/tk_html_widgets)

Alle unterstützten Features auf der o.a. Website. Die Beispiel-HTML dieser Seite befindet sich im

Beispiel pp\_html\_1p4 und im Media-Verzeichnis.

Eine leicht veränderte Version des Codes von tk\_html\_widgets ist in Pi Presents enthalten. Ein Installation von Widgets ist deshalb nicht erforderlich.

### 5.4.5 MPV Video Track

Ein Track für den MPV Media Player. MPV ist in erster Linie ein Kommandozeilen Videoplayer. Pi Presents nutzt die Python MPV library zur Steuerung von MPV. Pi Presents spielt jedes Video ab, das vom MPV-Player abgespielt werden kann. Video Tracks können mit dem Pause-Kommando angehalten werden. Videos können in einem Fenster erscheinen. Sie können überdies ein Bild oder eine Farbe als Hintergrund haben. Track Text sind mit Tracks verknüpft, Show Texte mit Shows. Das Video selbst kann nicht mit Text belegt werden, weil MPV das nicht zulässt. MPV benutzt nicht die GPU, deshalb benötigt die CPU für das Abspielen von HD-Videos (1080p) einen hohen Anteil Rechenkapazität, auf einem P4 typischerweise 75 Prozent.

Ein MPV-Videotrack kann Audiotracks abspielen. Freeze-at-End sollte auf no eingestellt werden.

MPV-Videotrack kann keine Streams abspielen (url://...)

Field	Example	Values
<b>Track Tab</b>		
Type	mpx	
Title	The Film	Wird im Menü und im Editor angezeigt.
Track-Reference		Kennzeichnung des Tracks – leer, wenn der Track in einem Menü oder einer Mediashow enthalten ist. Nicht leer, wenn der Track mit einem symbolischen Namen gesteuert wird oder ein spezieller Track ist.
Location	+/myvideos/ film.mp4	Der Dateiname des Tracks. Wenn er einen Doppelpunkt (:) mit mehr als einem Buchstaben davor enthält, wird er als URL behandelt. Ansonsten wird er als Dateipfad herangezogen.
Thumbnail	+/media/ film_tn.jpg	Der gewünschte Dateiname eines Thumbnails, welches in einer Menüshow gezeigt wird.
Freeze at Start	before-first-frame	Friert Video beim Trackstart ein <ul style="list-style-type: none"> <li>no – friert nicht ein</li> <li>before-first-frame – pausiert das Video bevor der erste Frame gezeigt wird</li> <li>after-first-frame – pausiert das Video nach dem ersten Frame</li> </ul> Mit dem Befehl ‚go‘ nach dem Einfrieren fortsetzen Wenn leer, gilt der Wert der Elternshow.
Freeze at End	yes	yes/no – wenn yes, wird das letzte Videobild gezeigt (das Video pausiert kurz vor seinem Ende). Wenn no, endet das Video ganz normal

		Wenn das Feld leer ist, wird der Wert in der Show übernommen
MPV Player Audio	default	Hdmi/ A/V / USB / USB2 / bluetooth, zusätzlich /hdmi0 /hdmi1 beim Pi 4. Wenn leer, wird der Wert der Elternshow genutzt.)
MPV Player Volume	100	Lautstärke des Videotracks (0 -> 100 dB). Wenn leer, wird der Wert der Elternshow genutzt
MPV Player Max Volume	100	Maximale Videolautstärke. VLC könnte verzerren, Wenn das Eingangssignal sehr hoch ist und RTC (Real Time Control) die Lautstärke zu hoch treibt. Die Lautstärke kann hier limitiert werden.
MPV Player Options	–	Zusätzliche Kommandozeilenoption für den MPV Player – durch Komma getrennt. Eine Option hat die Form: option=value. Option ist die Befehlszeilenoption ohne den vorangestellten Bindestrich. Optionen und Values sind mit Pipresents nicht getestet und könnten zu Abstürzen führen. Wenn leer, wird der Wert der Elternshow übernommen.
MPV Window	Showcanvas 100+100+400*300  showcanvas 400*300  showcanvas  display	Ein Sichtfenster für das Video. Siehe Werte <a href="#">Kapitel 5.4.5.1</a> Wenn leer, werden die Werte der Elternshow genutzt
MPV Aspect Mode	original	warp/original Wie das Video im MPV Fenster angezeigt wird:  warp – richtet das Video das gesamte Videofenster ausfüllend ein und ändert, falls erforderlich, das Seitenverhältnis des Videos.  original – richtet das Video im Videofenster unter Beibehaltung des Seitenverhältnisses ein. Wenn das Seitenverhältnis von Video und Videofenster nicht übereinstimmt, erscheinen schwarze Balken.  Anmerkung: Wenn die Video und Fenster im Seitenverhältnis nicht übereinstimmen, stimmen die x- und y-Positionen des Videos nicht mehr, weil die Werte im Fenster zentriert werden. Ausgenommen im Fullscreenmodus.  iWenn leer, werden die Werte der Elternshow genutzt.
Background Colour	red	Schriftfarbe im Auswahlfenster wählen, erzeugt

		eine sechsstellige Hexadezimalzahl, oder Farbnamen direkt eintragen. Siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> or <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> Wenn leer, wird der Wert der Elternshow benutzt
Background Image	+/images/ back.jpg	Der Dateiname eines optionalen Bildes, das als Hintergrund gezeigt wird, kann leer bleiben. Video, Showtext und Track werden über dem Bild angezeigt. Wenn leer, wird der Wert der Elternshow benutzt
Display Show Background Image	yes	yes/no yes – das Hintergrundbild der Elternshow wird angezeigt, wenn der Track 'Background Image' leer ist. no – das Hintergrundbild der Elternshow wird nicht angezeigt.
Plugin Configuration File	+/media/ weather_ny.cfg	Datei enthält den Namen und die Parameter des Plugins
Pause Timeout	5	Sekunden - wenn nicht leer und größer 0, wird ein pausierender Track automatisch nach dem Timeout wieder gestartet
<b>Controls Tab</b>		
Controls	myname null	Verknüpfung symbolischer Namen mit Befehlen. Die für einen Track hier festgelegten Verknüpfungen überschreiben die Befehle der zugehörigen Show ( <a href="#">siehe Kap. 7.1</a> )
<b>Text Tab</b>		
		s. Image Track. Hat keinen Pausentext
<b>Show Control Tab</b>		
		s. Image Track
<b>Animation Tab</b>		
		s. Image Track

#### 5.4.5.1 MPV Window

Ein Darstellungsfenster für das Video.

Werte: display, showcanvas, mit Optionen...

...<x>+<y>+<B>\*<H>

oder

...<Breite>\*<Höhe>

- showcanvas – MPV-Fenster bezieht sich auf das Videofenster
- display – MPV-Fenster bezieht sich auf den Bildschirm

Wenn das Feld optional dimensions nicht gewählt ist, wird das Video zentriert und hat die Größe von showcanvas/display.

Das optional Feld bestimmt die Position und Größe des MPV-Fensters relativ zur gewählten Basis. Wenn x,y nicht bestimmt wurden, ist das Fenster in showcanvas/display zentriert.

Width,height – Fenstergröße in Pixels

x,y – der Ort in Pixel von der oberen linken Fensterecke unter Berücksichtigung der oberen linken Ecke von showcanvas/display.

Wenn leer, gelten die Einstellungen der Elternshow.

### 5.4.5.2 Video Playout

Pi Presents kann als Video/Audio-Ausgabesystem in Teatern oder Fernsehstudios genutzt werden. Dabei kommt es darauf an, das gewünschte Video zu laden und vor dem ersten Frame zu pausieren. Ein Klick auf g (Go-Befehl) startet das Video sofort.

Im Beispiel pp\_videoplout\_1p5 stellt die Mastershow auf hdmi1 per Show Control Kommandos Videos bereit für die Slave-Radiobuttonshow auf hdmi0. Die Mastershow kann ein Menü oder eine Mediashow statt der Radiobuttonshow sein. Audios können abgespielt werden, voreingestellt mit dem MPV Videotrack.

### 5.4.6 Audio Track

Audiotracks werden vom MPlayer abgespielt. Pi Presents sollte alle Audiotracks spielen, die der Mplayer abspielen kann. Der Ton kann über HDMI oder analog und jeweils über einen der analogen Lautsprecher ausgegeben werden. Audiotracks können mit dem Pause-Befehl angehalten werden.

Audiotracks kann ein Bildschirm zugeordnet werden. Der Bildschirm kann eine Farbe oder ein Bild als Hintergrund besitzen. Tracktext liegt jeweils über einem Track, während Showtext über einer Show gezeigt wird.

Ein Audiotrack hat erweiterte Laufzeit- und Bildschirmsteuerungen, die es erlauben, mit einem 'Dummytrack' Animationen zu steuern oder konkurrierende Shows abzuspielen. Er kann zero duration' und run enthalten ohne Medien abzuspielen.

Field	Example	Values
<b>Track Tab</b>		
Type	audio	
Title	The Music	Wird im Menü und im Editor angezeigt
Track Reference		Kennzeichnung des Tracks – leer, wenn der Track in einem Menü oder einer Mediashow enthalten ist. Nicht leer, wenn der Track mit einem symbolischen Namen gesteuert wird oder ein spezieller Track ist

Thumbnail	+/media/ sarename_tn.jpg	Der Dateiname eines Thumbnailbildes wird in einer Menüshow gezeigt
Location	+/tracks/ music.mp3	Der Dateiname des Tracks. Die Location ist beliebig. Wenn leer, erfolgt die Zeitsteuerung im Duration Feld, das nicht leer sein darf.
Duration	5	<p>Sekunden in Schritten von 0.1 secs.</p> <p>Wenn 'Duration' leer ist, endet der Track, wenn die Audiodatei zu Ende ist oder gestoppt wurde. Falls kein Audiotrack vorhanden ist, muss der Benutzer den Track stoppen. Letzteres ist erforderlich, wenn der Track als Show oder Animation Control benutzt wird.</p> <p>Wenn ,0' dann sollte Location leer sein, kein Audiotrack wird abgespielt. Animation und Show Control werden ausgeführt. Zero Duration wird vom Track für Show- oder Animationssteuerung benötigt.</p> <p>Wenn Duration größer ist als ,0' endet der Track zur angegebenen Zeit. Wenn eine Datei mit Location eingerichtet wurde, kann das Abspielen des Tracks zu früh abgebrochen werden oder es gibt eine Tonpause nach der Audiodatei. Das gilt in erster Linie für Animation und Show Control.</p> <p>Bei (art)Liveshows ist duration immer blank.</p>
Audio Player Speaker	left	left/right/stereo – Wenn leer, wird der Wert in der Parents show benutzt.
Audio Player Audio	hdmi	<p>hdmi / A/V / USB zusätzlich hdmi/0 / hdmi1 für den Pi4 Wenn leer, wird der Wert in der Parents show benutzt.</p> <p>Die früheren Werte „local“ und „alsa“ berücksichtigt Pi Presents zwar, sie erscheinen aber nicht mehr im Dropdown-Menü des Editors.</p>
Audio Player Volume	100	Volume of audio track (0 ->100 dB) – Wenn leer, wird der Wert der Elternshow benutzt.
Audio Player Options		Zusätzliche Befehlszeilen-Kommandos für den mplayer. Wenn leer, wird der Wert in der Elternshow genutzt.
Background Colour	red	<p>Schriftfarbe im Auswahlfenster wählen, erzeugt eine sechsstellige Hexadezimalzahl, oder Farbnamen direkt eintragen.</p> <p>Siehe:  <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>  oder  <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a></p>

		Wenn leer wird der Wert in der Elternshow benutzt.
Background Image	+/images/ back.jpg	Der Dateiname eines optionalen Bildes, das als Hintergrund gezeigt wird. Video, Showtext und Track werden über dem Bild angezeigt. Wenn leer, wird der Wert der Elternshow benutzt.
Display Show Background Image	yes	yes,no yes – das Hintergrundbild der Elternshow wird angezeigt, wenn im Track 'Background Image' leer ist. no – das Hintergrundbild der Elternshow wird nicht angezeigt.
Plugin Configuration File	+/media/ weather_ny.cfg	Datei enthält den Namen und die Parameter des Plugins
Pause Timeout	5	Sekunden - wenn nicht leer und größer 0, wird ein pausierender Track automatisch nach dem Timeout wieder gestartet
<b>Controls Tab</b>		
Controls	Mynsme null	Verknüpfung symbolischer Namen mit Befehlen. Die für einen Track hier festgelegten Verknüpfungen überschreiben die Befehle der zugehörigen Show. ( <a href="#">siehe Kap. 7.1</a> )
<b>Text Tab</b>		
		Siehe Image Track
<b>Show Control Tab</b>		
		Siehe Image Track
<b>Animation Tab</b>		
		Siehe Image Track

### 5.4.7 Chrome Web Tracks

Web Tracks werden vom Chromium Browser dargestellt (in Pi Presents als Chrome bezeichnet). Chromium wird mit Raspberry Pi OS installiert.

Um den Browser in Pi Presents nutzen zu können, müssen einige Packages installiert sein.

Websites können aus dem Internet oder vom lokalen Dateisystem abgerufen werden. Sie können in einem Fenster oder im Vollbildmodus laufen. Als Hintergrund ist ein Bild oder eine Farbe möglich und sie können Track Text und Show Text zeigen. Track Text liegt über den einzelnen Tracks, Show Text über der Show.

Die Integration von Chrome in Pi Presents ist nicht so gelungen, wie die Zusammenarbeit mit den

anderen Playern. Am besten funktioniert Chrome auf nicht interaktiven Bildschirmen:

- Der Browser öffnet sich, wenn ein Web Track gestartet wird. Das kann einige Sekunden dauern und ist zudem nicht besonders sicher im Zusammenspiel mit Pi Presents. Für eine Reihe von Websites verfügt der Browser in Pi Presents über ein Browser-Befehlsscript, das die Darstellung einer Abfolge von Websites mit einstellbaren Abspielschleifen möglich macht.
- Im Gegensatz zu den anderen Playern fokussiert der Browser auf das Browserfenster, nicht auf das Fenster von Pi Presents. Tastatureingaben werden an Chrome gesandt und nicht an Pi Presents, bis das Pi Presents-Fenster angeklickt wird. GPIO-Eingaben sind wirkungslos.
- Die Taskleiste beeinflusst die Position des Browserfensters. Um das Browserfenster korrekt einzupassen, muss die Taskleiste in den Einstellungen>Erweitert ausgeblendet werden.
- Das Browserfenster hat Scrollbalken und eine Titelleiste. ([siehe Kap. 5.4.7.3](#)).

Field	Example	Values
<b>Track Tab</b>		
Type	chrome	
Title		Wird im Menü und im Editor angezeigt.
Track Reference		Kennzeichnung des Tracks – leer, wenn der Track in einem Menü oder einer Mediashow enthalten ist. Nicht leer, wenn der Track mit einem symbolischen Namen gesteuert wird oder ein spezieller Track ist.
Location	<a href="http://www.google.co.uk">http:// www.google.co.uk</a>  +/media/ mywebpage.html	Der Dateiname des Tracks: <ul style="list-style-type: none"> <li>• eine Internetadresse (url)</li> <li>• ein voller Pfadname zu einer Datei</li> <li>• ein relativer Pfad zu einer unter pp_home gespeicherten Datei</li> </ul>
Thumbnail	+/media/ sarename_tn.jpg	Der Dateiname eines Thumbnailbildes, wie er in einer Menüshow gezeigt wird.
Duration	5	Sekunden in Schritten von 0.1 secs. Wenn '0' wird die Website gespielt, bis sie durch Benutzerbefehl abgebrochen wird. Wenn leer, wird der Wert in der Elternshow benutzt.
Freeze at End		Bei yes verbleibt der Web Track auch nach der eingestellten Dauer, bis der folgende Track geladen ist. Das verhindert schwarze Lücken zwischen den Tracks. Bei no endet der Track mit der eingestellten Dauer. Wenn das Feld leer bleibt, gilt der Wert der Elternshow.
Chrome Window	kiosk	Ein Darstellungsfenster, in dem die Website

Field	Example	Values
	showcanvas 0+0+400*800  display 400*800	dargestellt wird. Wenn leer, wird der Wert der Elternshow benutzt. <a href="#">Siehe Kapitel 5.4.7.1</a> , um Werte zu sehen.
Chrome Zoom	0,5 02,0	Ein positiver Wert. Der Inhalt des Browserfensters wird um diesen Betrag erhöht oder verringert. Wenn leer, wird der Wert der Elternshow benutzt.
Chrome Options	-incognito	Flags, die dem Browserbefehl (mit Leerzeichen) angehängt werden können. Hier eine Liste: <a href="https://peter.sh/experiments/chromium-command-line-switches/">https://peter.sh/experiments/chromium-command-line-switches/</a> Wenn leer, wird der Wert der Elternshow benutzt.
Background Image	+/media/image.jpg	Der Dateiname eines Bildes, das als Hintergrund für die Webseite gezeigt wird. Wenn leer, wird der Wert der Elternshow benutzt.
Display Show Background Image	yes	yes/no yes – das Hintergrundbild der Elternshow wird angezeigt, wenn der Track 'Background Image' leer ist. no – das Hintergrundbild der Elternshow wird nicht angezeigt.
Background Colour		Schriftfarbe im Auswahlfenster wählen, erzeugt eine sechsstellige Hexadezimalzahl, oder Farbnamen direkt eintragen. Siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> Wenn leer, wird der Wert der Elternshow benutzt.
Plugin Configuration File	+/media/ weather_ny.cfg	Datei enthält den Namen und die Parameter des Plugins
<b>Text Tab</b>		
		Siehe Image Track. Kein Pause-Text.
<b>Browser Commands Tab</b>		
Browser Commands	wait 10 refresh	Befehle die URL etc. laden ( <a href="#">siehe Kap. 5.4.7.2</a> )
<b>Controls Tab</b>		
Controls	myname null	Verknüpfung symbolischer Namen mit Befehlen. Die für einen Track hier festgelegten Verknüpfungen überschreiben die Befehle der

Field	Example	Values
		zugehörigen Show. <a href="#">(siehe Kap. 7.1)</a>
<b>Text Tab</b>		
		Siehe image track
<b>Show Control Tab</b>		
		Siehe image track
<b>Animation Tab</b>		
		Siehe image track

### 5.4.7.1 Chrome Window

Chrome Window ist ein Darstellungsfenster, in dem Webseiten gezeigt werden.

Werte:

- kiosk, fullscreen
- display, showcanvas, with <x>+<y>+<width>\*<height>

#### **kiosk, fullscreen**

Die Webseite wird ohne Scrollbalken und Titelzeilen bei Anwendung der Chromium-Browser-Kommandozeilenbefehle -kiosk oder -start-fullscreen im Vollbild angezeigt.

Kiosk scheint dabei alle Steuerungen mit Ausnahme von ALT F4 auszuschalten, während fullscreen das nicht bewirkt.

#### **Showcanvas, display**

Die Webseite wird im Fenster in Position und Größe so eingerichtet, wie es nachfolgend beschrieben ist. Titelzeile und Scrollbalken sind vorhanden, Browserbefehle dagegen nicht (-app mode). Für die Entfernung von Titelzeile und Scrollbalken [siehe Kap. 5.4.7.3](#)

Werte: display, showcanvas, optionally

...<x>+<y>+<width>\*<height>

oder

...<width>\*<height>

- showcanvas – Chrome Fenster bezieht sich auch showcanas
- display – Chrome Fenster bezieht sich auf die Bildschirmgröße

Wenn das Feld optional deaktiviert ist, wird das Chrome Fenster zentriert und hat die Größe von showcanvas/display

Das Feld optional bestimmt Position und Größe des Chrome Fensters, bezogen auf die gewählte Basis. Wenn x,y deaktiviert sind, wird das Fenster in showcanvas/display zentriert.

width, height – Fenstermaße in Pixel.

x,y – obere linke Ecke des Fensters mit Bezug auf die obere linke Ecke von display/showcanvas

### 5.4.7.2 Browserbefehle

Das Feld für Browser Commands enthält '0' oder mehr Befehle. Jeder Befehl steht in einer eigenen Zeile. Einige Befehle haben ein Argument, durch einen Leerraum vom Befehl getrennt.

Wenn Browserbefehle eingesetzt werden, ist die Laufzeit des Tracks von der Ausführungsdauer Befehle und dem Feld duration im Schowprofil abhängig.

Pi Presents arbeitet die Liste der Browserbefehle ab. Wenn dort keine Schleife gesetzt wurde, endet Pi Presents. Ist eine Schleife, so kehrt Pi Presents zum entsprechenden Befehl ein- oder mehrmals zurück

Command	Beschreibung
load <arg>	Laden der Webseite mit <arg>. Argumente können sein: <ul style="list-style-type: none"> <li>• eine Internet URL</li> <li>• ein absoluter Pfad zu einer Datei</li> <li>• ein relativer Pfad zu einer Datei im Verzeichnis pp_home + oder das Profil</li> </ul>
refresh	Erneuerung der aktuell geladenen Webseite
wait <arg>	Wartet <arg> in Sekunden
Loop <arg>	Ein einzelner Schleifen-Befehl ist zulässig. Seine Ausführung wird nach dem letzten Befehl eines laufenden Scripts erfolgen. Das optionale ganzzahlige Argument bestimmt die Anzahl der Schleifen. Wenn leer, läuft die Schleife fortlaufend.

**Beispiel:** Wiederholung von zwei Webseiten. Schleife wird fortgesetzt bis die Trackdauer erreicht ist.

```
wait 20
load www.google.co.uk
wait 20
load www.museumoftechnology.org.uk
```

**Beispiel:** Zeigt zwei Webseiten, wechselt dann zum nächsten Track.

```
wait 20
load www.google.co.uk
wait 20
load www.museumoftechnology.org.uk
wait 20
exit
```

Sicherstellen, dass im Duration Feld mindestens 60 steht

**Beispiel:** Lädt die geladene Webseite im 20-Sekunden-Rhythmus fünf Mal neu.

```
Wait 20
loop5
refresh
wait 20
```

Sicherstellen, dass im Duration Feld mindestens 120 steht

### 5.4.7.3 Titelzeile und Scrollbalken entfernen

Im Kiosk- und Fullscreenmodus zeigt das Chrome Fenster die Webseite ohne Titelzeile und Scrollbalken.

Die Einstellungen showcanvas und display für das Chrome Fenster zeigen Webseiten über den Chromium Application Modus (--app). Standardmäßig werden dabei Titelzeile und Scrollbalken angezeigt. Um sie zu entfernen, ist folgende Dtei zu kopieren:

```
/etc/xdg/openbox/lxde-pi-rc.xml
```

zu

```
/home/pi/.config/openbox
```

füge in derDatei lxde-pi-rc.xml direkt vor dem Eintrag </applications> Folgendes hinzu:

```
<application class="Chromium-browser" role="pop-up">
<decor>no</decor>
</application>
```

Das Verzeichnis /home/pi/.config/openbox muss eingerrchtet werden. Um die Änderung wirksam werden zu lassen, muss der Pi neu gebootet werden.

Scrollbalken tauchen auf, wenn der Inhalt einer Webseite größer ist als ihr Fenster. Das kann verhindert werden, indem das Fenster vergrößert wird. Alternativ kann im zoom Feld des Profiles die Seitengröße angepasst werden. Scrollbalken sind dann nicht mehr erforderlich.

### 5.4.8 Message Tracks

Message Tracks zeigen Text wahlweise vor einem farbigen Hintergrund oder einem Bild. Sie benötigen keine Mediadatei zur Einrichtung, weil der Text im Message-Textfeld enthalten ist.

Field	Example	Values
<b>Track Tab</b>		
Type	message	
Title	A Message	Wird im Menü und im Editor angezeigt.
Track-Reference		Kennzeichnung des Tracks – leer, wenn der Track in einem Menü oder einer Mediashow enthalten ist. Nicht leer, wenn der Track mit einem symbolischen Namen gesteuert wird oder ein spezieller Track ist.
Thumbnail	+/media/sarename_tn.jpg	Der Dateiname eines Thumbnailbildes, der in einer Menüshow angezeigt wird.
Duration	5	Sekunden in Schritten von 0.1 secs. Wenn '0' wird die Message gezeigt, bis sie durch Benutzerbefehl abgebrochen wird. Wenn leer, wird der Wert in der Elternshow benutzt.
Background Colour	red	Schriftfarbe im Auswahlfenster wählen, erzeugt eine sechsstellige Hexadezimalzahl, oder Farbnamen direkt eintragen. Siehe: <a href="https://web.archive.org/web/20200315112731/">https://web.archive.org/web/20200315112731/</a>

		<a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> Wenn leer, wird der Wert der Elternshow benutzt.
Background Image	+/images/ back.jpg	Dateiname eines Bildes, das an Stelle eines einfachen Hintergrundes angezeigt wird. Die Message wird am oberen Bildrand angezeigt. Wenn leer, wird der Wert der Elternshow benutzt.
Display Show Background Image	yes	yes/no yes – das Hintergrundbild der Elternshow wird angezeigt, wenn das Trackfeld 'Background Image' leer ist. no – Das Hintergrundbild der Elternshow wird nicht angezeigt.
Plugin Configuration File	+/media/ weather_ny.cfg	Datei enthält den Namen und die Parameter des Plugins.
<b>Message Tab</b>		
Message Text	Welcome	<u>Angezeigter Text</u>
Message Text Location	+/me4dia/ mytextfile	Wenn nicht leer, hat der Text dieser Datei Vorrang vor dem Message-Text-Feld. Dieses Feld oder der Text in der Datei kann durch ein Plugin nicht verändert werden
Message Text x Position	100	Abstand des linken Randes des Textes vom linken Bildschirmrand in Pixel. Wenn leer, wird der Text zentriert dargestellt.
Message Text y Position	100	Abstand des oberen Textrandes vom oberen Bildschirmrand in Pixel. Wenn leer, wird der Text zentriert dargestellt.
Message Text Type	html	Plain – unformatierter Text html – ein Rahmen mit Text, Bildern und Formatierungen, in HTML gesetzt ( <u>s. Kap. 5.4.4.2</u> )
Plain Text Font	Helvetica 30 bold	Mit dem Schriftenmanager auswählen oder eine Schrift eintragen, siehe <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Plain Text Colour	white	Schriftfarbe im Auswahlfenster wählen, erzeugt eine sechsstellige Hexadezimalzahl, oder Farbnamen direkt eintragen. Siehe: <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Plain Text6 Justify	left	left/center/right – Textposition.
HTML Text Height	300	Höhe des HTML-Textfeldes

HTML Text Width	300	Breitet des HTML-Textfeldes
HTML Text Background Colour	##ff0000	Hintergrundfarbe des HTML-Textfeldes
<b>Controls Tab</b>		
Controls	Myname play track34	Verknüpfung symbolischer Namen mit Befehlen. Die für einen Track hier festgelegten Verknüpfungen überschreiben die Befehle der zugehörigen Show. ( <a href="#">siehe Kap. 7.1</a> )
<b>Text Tab</b>		
		Siehe Image Track. Kein Pausentext.
<b>Show Control Tab</b>		
		Siehe Image Track
<b>Animation Tab</b>		
		Siehe Image Track

### 5.4.9 Show Track

Eine Show kann ein Track in einer anderen Show sein. Möglich sind:

- Eine Mediashow, zusammengestellt aus kleineren Mediashows
- Ein Menü von Mediashows
- Menüs mit Untermenüs
- Eine Liveshow gestartet von einer Mediashow, die den Startbildschirm liefert

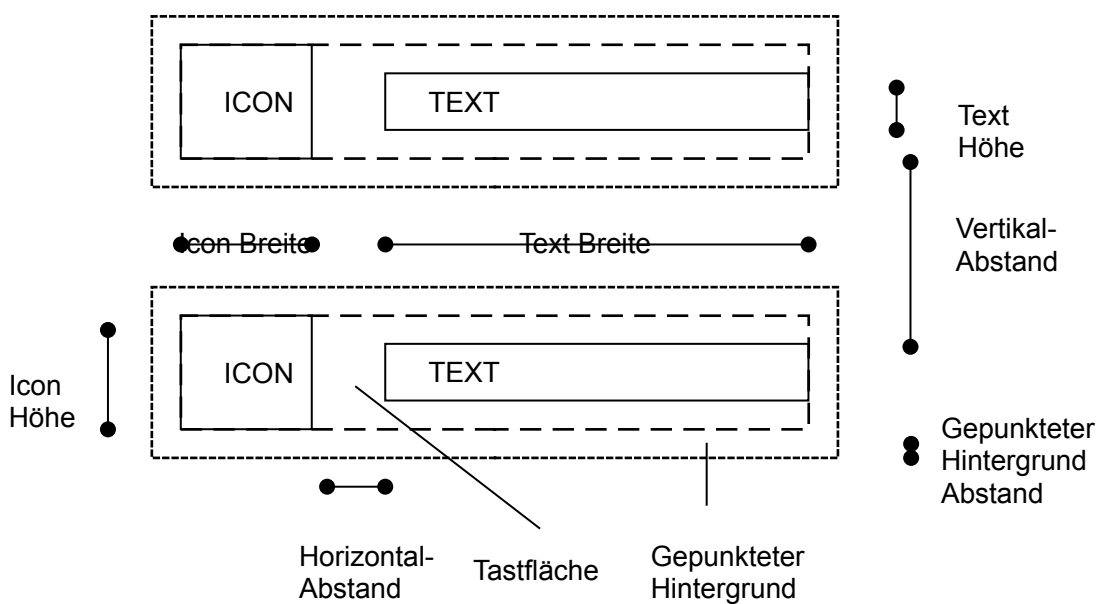
Field	Example	Values
Type	show	
Title	My Other Show	Wird im Menü und im Editor angezeigt.
Track-Reference		Kennzeichnung des Tracks – leer, wenn der Track in einem Menü oder einer Mediashow enthalten ist. Nicht leer, wenn der Track mit einem symbolischen Namen gesteuert wird oder ein spezieller Track ist.
Show to Run	myothershow	Show-Referenz der Show, die laufen soll.
Thumbnail	+ /media/ sarenam_ tn.jpg	Der Dateiname eines Miniaturbildes, das in einer Menüshow angezeigt wird.

### 5.4.10 Menü Track

Eine Menüshow benötigt einen in der Medialiste zugeordneten Menü Track, der das Menülayout regelt. Seine Track-Referenz sollte im Feld Menu Track der Menü Show erscheinen.

Die Menüformatierungen sind vielfältig:

- Display, Miniaturbilder, Knopfsymbole und Überschriften in vielen Kombinationen.
- Vertikale oder horizontale Anordnung des Menüs.
- Mehrspaltige und mehrzeilige Menüs.
- Änderung von Trennungen, Farben und Schriften der Menüeinträge.



#### Beschreibung der Menü-Geometrie

Field	Example	Use
<b>Track Tab</b>		
Type	menu	
Title	menu-track	In einem Menü und dem Editor angezeigt
Track Reference	menu-track	Eine Kennung für den Track.
Background Colour		Siehe Mediashow – Hintergrundfarbe des Menüs.
Background Image	+/media/image.jpg	Falls nicht leer steht hier der Name des Bildes, das als Hintergrund für das Menü dient.
Display Show Background Image	yes	yes/no yes – das Hintergrundbild der Elternshow wird angezeigt, wenn das Feld Background Image im Track leer ist. no – das Hintergrundbild der Elternshow wird nicht angezeigt.
Entry Font	Helvetica 30 bold	Den Schriftwähler benutzen oder direkt eintragen. Siehe :

		<a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Entry Colour	white	Den Farbenwähler benutzen der eine sechstellige Hexadezimalziffer ausgibt oder direkt eintragen: Siehe : <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Selected Entry Colour	red	Farbe des gewählten Menüeintrags.
Plugin Configuration File	+media/krt-time.cfg	Datei, die den Namen und die Parameter des Track Plugins enthält.
<b>Geometry Tab</b>		
Menu Window		fullscreen, ein oder zwei Paare x,y-Koordinaten, getrennt durch Leertaste  <ul style="list-style-type: none"> <li>• fullscreen – das Menü wird als Vollbild dargestellt</li> <li>• einzelne x,y-Koordinate – bezeichnet die ober linke Ecke des Menüs</li> <li>• zwei x,y-Koordinaten – das zweite Koordinatenpaar bezeichnet die untere erchte Ecke eines Rahmens für das Menü. Statt 2x,y,-Paaren kann x1+y1+w*h eingesetzt werden</li> </ul>
Direction	vertical	Vertical, horizontal – die Anordnung des Menüs und die Durchlaufrichtung
Rows	5	Wenn Direction = vertical ist das die Anzahl der Menüzeilen.
Columns	1	Wenn Direction = horizontal ist das die Anzahl der Menüspalten.
Icon Mode		Thumbnail, bullet, none <ul style="list-style-type: none"> <li>• thumbnail – wenn der Track ein Miniaturbild hat, wird es angezeigt, oder wird, wenn es ein Image Track ist, auf Icongröße angepasst, oder eine Miniatur wird je nach Typ des Tracks gezeigt.</li> <li>• bullet – das Icon, das im Feld Bullet benannt ist, wird gezeigt.</li> <li>• None – das Menü besteht nur aus Text.</li> </ul>
Text Mode		None, right, below <ul style="list-style-type: none"> <li>• right – der Text wird rechts vom Icon gezeigt.</li> <li>• Below – der Text wird unterhalb des Icons</li> </ul>

		gezeigt.
Bullet	+/media/bullet.jpg	Der Dateiname des Bullet mit absolutem oder relativem Pfad
Icon Width	100	Breite des Icons. Bilder werden in die vorgesehene Größe eingepasst.
Icon Height	100	Höhe des Icons. Bilder werden in die vorgesehene Größe eingepasst.
Horizontal Padding	10	Bei Textmode „right“ der horizontale Abstand zwischen Icon und Text
Vertical Padding	10	Bei Textmode „below“ der vertikale Abstand zwischen Icon und Text
Text Width	200	Textweite. Der Text wird eingepasst
Text Height	50	Texthöhe. Hilft bei der Positionierung der Einträge, die Texthöhe passt sich jedoch nicht an. Sie muss per Schriftgröße eingepasst werden
Horizontal Separation	20	Abstand zwischen dem unteren Rand eines Menüeintrages und dem oberen Rand des Nächsten.
Vertical Separation	20	Abstand zwischen dem rechten Rand eines Menüeintrages und dem linken Rand des Nächsten.
Stipple Background		yes/no – Zeigt einen gestrichelten, rechteckigen Rahmen um den Menüeintrag.
Stipple Background Padding		Der Abstand, den die gestrichelte Linie zum Menütext einhält.
Guidelines		never, auto, always Wenn das Menü durch zwei x,y-Paare gekennzeichnet ist, werden farbige Rechtecke auf dem Bildschirm angezeigt, die beim Entwurf des Menüs helfen. <ul style="list-style-type: none"> <li>• Weiße Rechtecke stehen für den Menüeintrag.</li> <li>• Ein blaues Rechteck zeigt die Ausbreitung des Menüs.</li> <li>• Das rote Rechteck ist die Box, wie sie im Menu Window eingetragen ist.</li> </ul> Auto zeigt die Rechtecke nur, wenn der Rahmen der Box durch das Menü überschritten wird.
<b>Text Tab</b>		
Track Text	This Menu	Mehrzeiliger Text
Track Text Font	Helvetica 30 bold	Den Schriftenwähler benutzen oder direkt eintragen. Siehe : <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Track Text Colour	white	Den Farbenwähler benutzen der eine sechstellige Hexadezimalziffer ausgibt oder direkt eintragen: Siehe : <a href="https://web.archive.org/web/20200315112731/">https://web.archive.org/web/20200315112731/</a>

		<a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Track Text x	100	Abstand des Textbeginns vom linken Bildschirmrand (Pixel). Wenn leer, wird die Message auf dem Bildschirm zentriert.
Track Text y	800	Abstand des oberen Textrandes vom oberen Bildschirmrand des Bildschirms (Pixel). Wenn leer, wird die Message auf dem Bildschirm zentriert.
Track Text Justify	right	Left, right, center – richtet Zeilen im Textblock aus
Hint Text	To play, press return	Menüs haben keine Ableger. Hint ist ein vielseitiges Feld, das für Hinweise und Instruktionen genutzt werden kann.
Hint Font	Helvetica 30 bold	Den Schriftenwähler benutzen oder direkt eintragen. Siehe : <a href="https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm">https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a> oder <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Hint Colour	white	Den Farbenwähler benutzen, der eine sechstellige Hexadezimalziffer ausgibt oder direkt eintragen: Siehe : <a href="http://effbot.org/tkinterbook/tkinter-widget-styling.htm">http://effbot.org/tkinterbook/tkinter-widget-styling.htm</a>
Hint x	100	Abstand des Textbeginns vom linken Bildschirmrand (Pixel). Wenn leer, wird die Message auf dem Bildschirm zentriert.
Hint y	800	Abstand des oberen Textrandes vom oberen Bildschirmrand Bildschirms (Pixel). Wenn leer, wird die Message auf dem Bildschirm zentriert.
Hint Justify	right	Left, right, center – richtet Zeilen im Textblock aus
<b>Controls Tab</b>		
Controls	Myname null	Verknüpfung symbolischer Namen mit Befehlen. Die für einen Track hier festgelegten Verknüpfungen überschreiben die Befehle der zugehörigen Show. ( <a href="#">siehe Kap 7.1</a> )
<b>Show Control Tab</b>		
		Siehe Image Track
<b>Animation Tab</b>		
		Siehe Image Track

## 6 Black Box Operation

Es gibt eine Menge einzurichten um Pi Presents in ein bildschirmfüllendes, automatisch startendes und GPIO-kontrolliertes Werkzeug zu verwandeln. Sie werden nicht alles brauchen.

## 6.1 Optionen auf der Befehlszeile

`python3 pipresents.py -h` zeigt die Optionen auf der Befehlszeile.

Options	
<code>-p --profile &lt;arg&gt;</code>	Name des benutzten Profils. z.B. <code>pp_mediashow_1p4</code> Wenn nicht eingetragen, erscheint eine Fehlermeldung
<code>-b --noblack</code>	Verhindert Bildschirmpause
<code>-f --fullscreen</code>	Pi Presents läuft bildschirmfüllend
<code>-o --home &lt;arg&gt;</code>	Speicherort des Pi Presents Homeverzeichnis  z. B. <code>/media/pi/USBStick</code> oder <code>/home/pi/my_data</code>  Wenn diese Option nicht gewählt wurde, ist das Benutzer-Homeverzeichnis Standardeinstellung
<code>-d --debug</code>	Schlimme Systemfehler und Profilprobleme geben einen Alarm auf dem Bildschirm aus. Nach Bestätigung fährt Pi Presents herunter, weil es sich nicht fortsetzen kann. <ul style="list-style-type: none"> <li>• Profilfehler sind Probleme in den Profilen</li> <li>• Fatale Fehler sind Probleme in der Programmsoftware von Pi Presents, die eine Begleiterscheinung von nicht entdeckten Profilfehlern sein können</li> </ul> Der <code>pp_editor</code> erkennt viele Profilfehler mit seiner Validierung. Bitte Fehler melden, die nicht vom Editor erkannt werden.  Protokollausgaben (logoutput) werden im Terminal angezeigt, wenn Pi Presents darin gestartet wird. Zudem wird alles in  <code>/pipresents/pp_log/pp_log.txt</code>  protokolliert.  Wenn die <code>-d</code> -Option nicht gesetzt ist, zeigt das Protokoll lediglich fatale Fehler, Profilfehler und Warnungen. Bei gesetzter <code>-d</code> -Option enthält das Protokoll die Pi Presents-Aktionen, hilfreich bei der Fehlersuche in Profilen. Warnungen betreffen zwischenzeitlich auftretende Probleme, die Pi Presents lösen kann. Fast alle werden durch Speicherabstürze des <code>omxplayer</code> verursacht.
<code>-d --debug &lt;arg&gt;</code>	Die Verwendung von <code>-d</code> mit Argumenten bietet eine genauere Kontrolle beim Logging ( <a href="#">siehe Kap. 23</a> )
<code>-l --liveshow &lt;arg&gt;</code>	Liveshow Tracks werden immer aus dem Verzeichnis <code>pp_live_tracks</code> im Heimatverzeichnis abgeholt, wie es in der <code>-o</code> -Option als Standard festgelegt ist.  <code>/home/pi/pp_home/pp_live_tracks</code>

	Wenn die -l-Option einen Verzeichnispfad aufweist, ist das eine zweite Quelle für Liveshow Tracks.  z.B. /home/pi/mylivettracks
--counters	Wenn counter storage (Zählwerkspeicherung) aktiviert ist werden Zählwerke von der Start Show (Counters Tab) initiiert.
-manager	Damit kann der Pi-Presents-Manager Terminalmeldungen unterbinden. Nicht für Benutzer gedacht.
-n --nonetwork	Wenn die Time-of-Day-Planung eingerichtet ist und diese Option ausgelassen wird, wird PiPresents bis zu 5 Sekunden auf das LAN (Netzwerk) warten. Wenn die Option ohne Argument gewählt ist, wartet PiPresents nicht. Das ist sinnvoll, wenn eine RTC (RealTimeClock) vorhanden ist und NTP-Time (Network Time Protocoll) nicht benötigt wird. Besonders wenn eine gewisse Zeit gewartet werden soll, ist --nonetwork für das Intervall von n Sekunden einzustellen
--nounclutter	Nicht den Mauszeiger mit unclutter verbergen. Hinweis: Unclutter wird nicht mit Pi Presents beendet. Diese Option ist deshalb nur nach einem Neustart angesagt.
--vibes	Startet den DRV2605 Haptic Player
--backlight	Startet den Hintergrundbeleuchtungscontroller für Touchscreens

## 6.2 Ein Profil aufrufen

Die --profile (-p) Befehlszeilenoption bezeichnet das aufzurufende Profil. Es ist der Name des Profilverzeichnis im Ordner pp\_profiles. Wenn diese Option nicht gewählt ist, erfolgt eine Fehlermeldung.

Das Präfix pp\_ kennzeichnet die vom Entwickler stammenden Dateien. Es hat ansonsten keine besondere Bedeutung.

## 6.3 Ein Heimatverzeichnis aufrufen

Die --home (-o) Befehlszeilenoption legt das Heimatverzeichnis für die Pi Presents Daten fest. Ist sie nicht gewählt, wird standardmäßig das Homeverzeichnis des Benutzers festgelegt.

Um Profile und Medien bereitzustellen, können USB-Sticks eingesetzt werden. Sie erscheinen als Mountpoints im Verzeichnis /media

z.B. /media/pi/KINGSTON

Pi OS erkennt USB-Sticks automatisch, wenn sie beim Einschalten vorhanden sind oder später eingesteckt werden. Wenn Pi Presents mit dem Einschalten des Computers gestartet wird dauert es bis zu 10 Sekunden, bis das USB-Laufwerk erkannt ist. Pi Presents toleriert das.

Pi OS erzeugt den Mountpoint mit dem Namen des Sticks. Wenn der USB-Stick auf einem Windowsrechner eingerichtet wurde, erscheint der Laufwerksname in Großbuchstaben, wie das bei Windows üblich ist.

## 6.4 Pi Presents mit GPIO steuern

Die GPIO-Steuerung ist aktiv, wenn eine gpio.cfg Datei im Verzeichnis /pp\_io\_config eines Profiles vorhanden ist.

sudo ist nicht erforderlich und darf auch nicht benutzt werden

Wenn die gpio.config Datei aus /pipresents/pp\_resources/pp\_templates in ein Profilverzeichnis kopiert wurde, sind die Pins des P1-Connectors an die folgenden symbolischen Namen gebunden. Diese Namen sind in den Beispielen für Mediashows, Menüs und besondere Steuerungen eingesetzt. Ihre Verknüpfung kann, wie in [Kapitel 16.2.2](#) beschrieben, geändert werden.

Pin of P1-Connector	Symbolischer Name	Befehl
P1-15	pp-down	down
P1-16	pp-up	up
P1-18	pp-play	play
P1-22	pp-pause	pause
P1-7	pp-stop	stop
P1-11	PIR	In einigen Beispielen als Startbefehl für eine Mediashow
P1-12	pp-shutdownnow	In der gpio.cfg so eingestellt, dass eine Taste 5 Sekunden lang gedrückt werden muss. Dadurch wird Pi Presents beendet und der Pi wird heruntergefahren.

Wenn diese Konfiguration benutzt wird, sind die Anschlüsse als Flanken auslösende Eingaben mit internen pull-up-Widerständen geschaltet und erfordern folgende Gerätecharakteristik:

- Drucktaster sollten mechanisch sein, drücken für Aktionen (in der Regel offen) können und den jeweiligen GPIO-Pin mit 0 Volt verbinden. Der Kontakt schließt bei Knopfdruck und bewirkt eine Auslösung mit fallender Flanke.
- Den PIR-Eintrag habe ich nach den Vorgaben von Burglar Alarms (UK) eingerichtet. Die Burglar-Geräte haben normalerweise geschlossene Relaiskontakte, die mit dem GPIO-Pin und 0 Volt verbunden sind. Der Relaiskontakt öffnet, wenn eine Bewegung erkannt wird. Der PIR-Eintrag ist also als Auslöser ansteigender Flanke konfiguriert und weil es ein Relaiskontakt ist, wurde der pull-up Widerstand eingerichtet, um Störsignale und damit eine falsche Auslösung zu vermeiden.

Eingaben können von normal offen zu normal geschlossen und umgekehrt durch Flankenänderung geändert werden. ([siehe Kap. 16.2.2](#))

Ein 330-Ohm-Widerstand für Taster und PIR wird empfohlen, um den Pi zu schützen, falls unabsichtlich Eingaben als Ausgaben benutzt werden.

GPIO-Pin ----- 330R ----- contact ----- 0 Volt

Achtung: Niemals einen GPIO-Pin mit dem 5-Volt-Pin verbinde – das würde Ihren Pi fritieren.

Die softwareseitige Kontakt-Entprellung ist auf eine geringe Empfindlichkeit eingestellt. Bei Problemen mit der Kontakt-Prellung kann die Empfindlichkeit für den jeweiligen Pin in der `gpio.config` geändert werden. ([siehe Kap. 16.2.2](#))

## 6.5 Bildschirmpause ausschalten

Um Bildschirmpausen abstellen zu können, muss zunächst `xset` installiert werden, das ein Teil der `x11-server utilities` ist. Raspbian bringt `x11-server utilities` bereits mit. Kontrolle mit:

```
sudo apt-get install x11-server-utils
```

Mit der Befehlszeilenoption `--noblink` wird die Bildschirmpause verhindert.

## 6.6 Pi Presents beim Einschalten des Pi starten

Das funktioniert nur, wenn in der `raspi-config` „boot to desktop“ eingerichtet ist. Das Verzeichnis `.config` ist bereits vorhanden. Es könnte sein, das „versteckte Dateien anzeigen“ im Dateimanager aktiviert werden muss, um dieses Verzeichnis sichtbar zu machen.

In Trixie existiert die Autostart-Datei

```
/home/pi/.config/lxsession/rpd-x/autostart
```

noch nicht. Die Datei oder ihr Inhalt muss von

```
/etc/xdg/lxsession/rpd-x/autostart
```

kopiert werden.

Danach muss

```
home/pi/.config/lxsession/rpd-x/autostart
```

wie folgt editiert werden

```
/home/pi/pipresents/autostart.sh
```

Achtung: Diese Datei ist kein Standard `.sh` Skript. Das führende `@` ist nicht erforderlich es bewirkt, starte neu bei Fehler.

Nun in die Datei

```
/home/pi/pipresents/autostart.sh
```

den Befehl zum Start von Pi Presents eingeben und danach die Berechtigung zum Ausführen der Datei erteilen (Rechtsklick<Eigenschaften<Zugriffsrechte)

## 6.7 Raspberry Pi per GPIO herunterfahren

Wenn eine gpio.cfg eingerichtet wurde (siehe nächstes Kapitel) den Shutdown-Button länger als 5 Sekunden drücken. Pi Presents wird danach sauber beendet und der Raspberry Pi fährt sicher herunter.

Der Pi kann unmittelbar auch auf anderem Weg beendet werden ([siehe Kap.7](#)).

Pi Presents beenden/neu starten und Raspberry Pi herunterfahren

Es gibt viele Wege, Pi Presents zu verlassen und den Pi durch Pi Presents herunterzufahren.

- Eine Taste, einen Pin oder einen Klickbereich mit dem internen symbolischen Namen pp-shutdownnow verknüpfen ([Siehe Kapitel 16.2](#)). Eine Eingabe mit diesem symbolischen Namen veranlasst den Pi zum sofortigen Herunterfahren Für GPIO kann der zugehörige Pin so konfiguriert werden, dass das Ereignis so lange nicht ausgelöst wird, wie der Pin für eine definierte Zeit gedrückt wird.
- Mit dem Show Control-Befehl shutdownnow wird der Pi innerhalb einer laufenden Show unmittelbar heruntergefahren ([Siehe Kapitel 7.2](#)).
- Mit dem Time-of-Day-Planer.
- Von Ferne über einen OSC-Befehl.

Es gibt viele Wege, Pi Presents zu verlassen:

- Eine Taste, einen GPIO-Pin oder einen Klickbereich mit dem intern festgelegten symbolischen Namen pp-exitpipresents verknüpfen (Siehe unten).
- Mit dem Show-Control-Befehl 'exitpipresents' innerhalb einer Show Pi Presents beenden.
- Mit dem Zeitplaner (Time of Day).
- Von Ferne über einen OSC-Befehl.
- Beim Einrichten mit Ctrl+Break, Alt+F4 oder dem Schließen-Icon im Pi Presents-Fenster des Terminals.

### Pi Presents Neustart

Es gibt eine Möglichkeit Pi Presents neu zu starten:

- Verknüpfen einer Taste, eines GPIO-Pins oder einen Klickbereich mit dem intern festgelegten symbolischen Namen pp.restartpipresents (siehe weiter unten)

Das funktioniert wie hier dargestellt:

<https://blog.petrzemek.net/2014/03/23/restarting-a-python-script-within-itself>

Hinweis: Auf geöffnete externe Dateien achten!

Symbolischer Name	Aktion
pp-exitpipresents	Beendet Pi Presents nach Schließen aller Shows.
pp-shutdownnow	Beendet Pi Presents und fährt den Raspberry Pi sicher herunter.
pp-restartpipresents	Startet Pi Presents neu nachdem alle Shows beendet sind. Der Restart-Befehl benutzt die originalen Kommandozeilenbefehle.

## 7 Tracks und Shows steuern

### 7.1 Track-Steuerung in einzelnen Shows

Die Steuerung einer einzelnen Show benötigt Befehle wie up, down, play, stop, call, und return, um die Erscheinungsweise einzelner Tracks zu regeln. Es erfordert mehrere Schritte, diese Befehle mit den Eingabegeräten wie Tasten und GPIO-Buttons zu verbinden. Es ist ein zweistufiger Verknüpfungs- oder Zuordnungsprozess.

- In Konfigurationsdateien, verbunden mit einem bestimmten Eingabegerät (z.B. keys.cfg, gpio.cfg) sind die physikalischen Tasten mit symbolischen Namen verknüpft, z.B. bei gpio `falling_name=myname`.
- Im Controls-Feld von Shows und Tracks wird der symbolische Name mit einem Befehl verknüpft, z.B. `myname up`.

Um mehr in technische Detail zu gehen: In Pi Presents sind die eingesetzten physikalischen Geräte unabhängig von den Kernprozessen in Shows und Tracks. Die physikalischen Geräte haben Treiber für den physikalischen Input und generieren Eingabeereignisse, die durch symbolische Namen gekennzeichnet sind. Die Shows und Tracks nehmen die Inputereignisse entgegen und wandeln sie in die Befehle für die Steuerung der Show um.

Diese beiden Zuordnungen (Verknüpfungen) finden statt in:

- Konfigurationsdateien, die mit den Treibern zusammenarbeiten, z.B. gpio.cfg. Diese Datei legt fest, welche Eingaben Input-Ereignisse bewirken und benennen deren symbolische Namen.
- Das Feld Controls in Shows und Tracks. Dieses Feld bringt symbolische Namen der Eingabeereignisse mit den Befehlen, Laufzeitkontrollen und Steuerungen, wie sie in Shows und Tracks benutzt werden, zusammen.

Jeder Showtyp hat einen Satz von Befehlen, wie sie in den Kapiteln zur Beschreibung der Shows dargestellt sind. Mediashow und Menüs haben Befehle wie play, pause, up und stop. Hyperlinkshows haben Befehle wie call und return.

Neben den Kommandos haben Mediashow und Liveshow Auslöser, die in diversen Shows alternativ zu den Befehlen eingesetzt werden. Sie werden hauptsächlich per gpio genutzt, um

einfache Aktionen mit Buttons oder PIR auszulösen.

Alle Shows haben Laufzeitkontrollen, die dem Benutzer bei laufenden Tracks Kontrollen der Lautstärke etc. erlauben.

Für Ausgaben ist ein einfaches System eingerichtet. Animationsbefehle enthalten symbolische Namen. Ausgabegerätetreiber benutzen die Konfigurationsdateien, wie `gpio.cfg`, um die symbolischen Namen in physikalische Ausgaben zu wandeln.

## Controls und Subshows

Eine Show kann Subshows besitzen. Eingabeereignisse werden von der Show an die Subshow weitergereicht, so dass sie die aktuelle Show und deren Tracks auf einer unteren Ebene steuern.

Eine Show auf der obersten Ebene startet mit dem Beginn von Pi Presents nach Maßgabe der Befehle in der Show Control, per Eieruhr oder von Ferne per OSC; es gibt hier keine Elternshow. In Version 1.2 wurden die Verknüpfungen im Feld Controls der Top-Level-Show von den Subshows übernommen. In Version 1.3 ist das nicht mehr der Fall: Jede Show muss einen eigenen Satz Befehlsverknüpfungen haben.

## 7.2 Shows öffnen und schließen

Es gibt vier Methoden, Shows zu starten und zu beenden:

- Start von einer oder mehrerer Shows durch Eintrag der Show-Referenzen im Feld Start Shows der Startshow. Sie werden gestartet, wenn Pi Presents startet.
- Starten oder Beenden der Show mit dem Zeitplaner (Time of Day).
- Mit einem Befehl Show Control im Feld Show Control eines Tracks eine Show starten oder beenden

```
open myothershow
close myothershow
```

- Starten oder Beenden einer Show aus der Ferne über das Open Sound Protocol (OSC).

Mit einer dieser Methoden geöffnete Shows können mit einer anderen beendet werden.

Nur eine Instanz einer Show kann in einer Zeit laufen. Bei Versuchen, eine Show zu starten, die bereits läuft, werden die Kommandos ignoriert.

Pi Presents kann auch ohne den Start einer Show begonnen werden. Genauso kann Pi Presents, auch wenn alle Shows beendet sind, mit einem dunklen Bildschirm weiter laufen, um einer Fernbedienung oder einer geplanten Timerschaltung zu erlauben, weitere Shows starten.

Befehle für die Kontrolle der Show sind im Feld Show Control der Tracks eingetragen.

Command	Aktion
open <show-ref>	Öffnet die bezeichnete Show. Der Befehl wird ignoriert, wenn die Show bereits läuft.
close <show-ref>	Beendet die bezeichnete Show. Der Befehl wird ignoriert, wenn die Show gar nicht läuft.

closeall	Beendet alle Shows. Pi Presents läuft weiter
Openexclusive <show-ref>	Beendet alle Shows(auch die Show, von der der Befehl kommt) und started die benannte Show

### 7.3 Time of Day Planer

Der Zeitplaner öffnet und beendet Shows zu einer festgelegten Zeit/Datum. Er kann zudem jeden anderen Show Control-Befehl zu einer festgelegten Zeit/Datum auslösen. Der Planer wird durch ein Profil aktiviert und gesteuert, das den Schedule Tab in der Start Show und in jeder anderen Show benutzt.

Der Schedule Tab in der Start Show macht die Aktivierung des Time of Day Schedulers möglich. Zudem steuert dieser Schedule Tab alle gemeinsamen Funktionen von Shows indem er die pp-core-section in der legacy.json Datei ersetzt.

Jede andere Show eines Profiles besitzt einen Schedule Tab, der den Zeitplan für diese Show umsetzt.

Jede hat 4 Felder: jeden Tag, Wochentag, Monatstag und ein bestimmter Tag. Jeder hat 0 oder mehr Startkriterien (Tag) mit einer zugehörigen Terminliste. Im Beispielprofil pp\_timeofday\_1p4 liegt ein Muster vor. Ziel dabei ist es, die Tipperei für jeden Tag im Jahr zu vermeiden. Statt dessen wird Jedentag als Basis benutzt. Ausnahmen dazu werden entweder als Wochentag(e), Monatstag(e) oder besondere(r) Tag(e) hinzugefügt.

Pi Presents legt für den aktuellen Tag beim Start und um jede Mitternacht eine Taskliste an. Für jede Show berücksichtigt Pi Presents jedes Feld in der folgenden Anordnung und jeden Eintrag mit Tag/Uhrzeit und vergleicht das aktuelle Datum mit den Startkriterien:

- everyday - läuft immer (die Zeile ,day' muss das Wort ,everyday' enthalten)
- weekday – läuft, wenn der aktuelle Wochentag in der Liste der Wochentage enthalten ist.
- monthday – läuft, wenn der aktuelle Monatstag in der Nummernliste enthalten ist.
- specialday – läuft, wenn der aktuelle Tag in der Datumsliste enthalten ist.

Wenn ein Termin läuft, merkt sich Pi Presents die Zeiten und auch die zugeordneten Befehle ,open' und ,close' werden festgehalten. Wenn ein späterer Zeileneintrag für Tag/Uhrzeit oder Feldeintrag zutrifft, überschreibt er die vorherigen Termine der Show. Eine Anordnung von Tag/Uhrzeit-Zeilen mit einer Zeile für den Tag aber einer leeren Liste der Uhrzeiten kann dazu dienen, Zeiten eines früheren Ereignisses zu löschen. Wenn alle Termine abgearbeitet sind, wird der Plan für den aktuellen Tag nach den Zeiten sortiert.

Zeiten können hh:mm:ss: oder hh:mm bei eingestelltem 24-Stundenrhythmus sein. Die Befehle lauten ,open' und ,close'. Sie sind verknüpft mit der Show-Referenz.

Wenn Pi Presents mitten am Tag gestartet wurde, überprüft das Programm die Zeittafel für die tägliche Aufgabenliste für jede Show (nicht aber die Start Show) und startet unmittelbar die Show, die zur aktuellen Zeit an der Reihe war. Catchup kann in einigen Fällen problematisch sein, weil ein spät gestarteter Task einen danach gestarteten Task überlappen kann oder selbst gekürzt wird; deshalb kann catchup für einzelne Shows deaktiviert werden.

Bei der Start Show ist es anders: Mit diesem Eintrag wird Pi Presents zeitlich gesteuert. Das Verfahren ist aber gleich und ein Zeitplan für den aktuellen Tag wird aufgestellt. Die Befehle, die nur in dieser Section erlaubt sind, sind alle Showbefehle mit Parametern. (Man kann ,open' und

,close' Showbefehle verwenden, wenn man in dieser Sektion die Show-Referenz statt einer Einzelshow angibt. Für die Start-Show existiert catchup nicht, alle vorherigen und aktuellen Tasks dieses Tages werden ignoriert.

Achtung!

- Bei einer Benutzung des Time of Day Planers sollten keine Shows in das Startshow-Feld Schedule Tab eingefügt und auch keine show controls für das Öffnen oder Schließen von Shows eingesetzt werden. Sie werden von catchup nicht berücksichtigt, was zu unerwarteten Verhaltensweisen führen kann.
- Kommandos in dem Schedule Tab der Start Show werden von catchup nicht berücksichtigt.
- Der Zeitplaner sollte auf die richtige Zeitzone eingestellt sein. Vorsicht bei Shows, die Mitternacht überschreiten. Ich habe einen Wechsel zur Sommerzeit nicht weiter untersucht.

Speziell für die Überprüfung des Planers gibt es eine Log-Funktion. Darin werden die Zeiten und Ereignisse aufgeführt. ([s. Kapitel 6.1](#))

Den Planer mit einer Echtzeituhr zu testen ist schwierig, deshalb kann der Simulate-Time Tab in der Start Show genutzt werden, um eine simulierte Zeit für einen Test des Zeitplaners (und nur des Planers) zu wählen. Wenn aktiviert, muss die gewünschte Zeit für den Teststart eingegeben werden. Das Datum muss nicht explizit ausgeführt werden, weil die Treffer-Kriterien unabhängig voneinander sind.

Der Planer benötigt time of day beim Start von PiPresents. Dazu wartet Pi Presents bis zu 5 Sekunden auf das LAN und erwartet ein Zeitsignal von einem erreichbaren NTP-Server. Das erlaubt, sich mit dem WLAN zu verbinden. Die Wartezeit kann mit der Befehlszeilenoption -nonetwork gesteuert werden.

## 7.4 Konkurrierende Shows

Pi Presents kann zwei oder mehr Shows gleichzeitig abspielen. Die Shows erscheinen parallel. Alle konkurrierenden Shows benutzen zwar die gleiche Bildschirmfläche, aber das Feld Show Canvas im Showprofil hilft, die Shows auf verschiedenen Bereichen des Bildschirms zu verteilen. Die Leistungsfähigkeit des Raspberry Pi und des Betriebssystems Pi OS setzen dem Gebrauch von konkurrierenden Shows aber gewisse Grenzen. Hier einige Beispiele für konkurrierende Shows:

- Ein Hintergrund-Audio-Track für eine Diashow. Dabei werden zwei Mediashows eingerichtet, eine mit einer manuell gesteuerten Diashow und die andere mit den Audio-Tracks. Für die Audio-Tracks sollten die Controls mit disable controls deaktiviert sein, wenn Benutzerinteraktionen mit der Diashow vorgesehen sind.
- Einen einzelnen Bildschirm für mehrere unabhängige Shows nutzen, etwa eine Diashow links, ein benutzergesteuertes Menü rechts und ein Infoband mit Zeit und Datum am oberen Bildschirmrand.
- Besonders sparsam ist es, zwei ganz unterschiedliche Prozesse auf dem gleichen Pi laufen zu lassen, beispielsweise eine Diashow im Besucherempfangsraum mit Childshow-Fähigkeiten, und eine 'dummy' in der Museumsausstellung, die etwas erklärt, gesteuert von einem Bewegungsmelder, wobei eine single shot mediashow zum Einsatz kommt.

### 7.4.1 Steuerung mit konkurrierenden Shows

Pi Presents kann zwei oder mehrere Shows gleichzeitig laufen lassen. Diese konkurrierenden Shows beginnen gleichzeitig zu laufen und Eingaben gehen an alle konkurrierenden Shows. Das ist korrekt, aber wenn es nicht vorsichtig gesteuert wird, drohen unerwünschte Effekte. Die meisten normalen Situationen können jedoch mit Disable Controls gemeistert werden.

Zum Beispiel eine manuell gesteuerte Mediashow, die Bilder parallel zu einer Mediashow mit Hintergrundmusik zeigt; die Benutzung von up und down bei der Betrachtung von Bildern soll nicht die Audiotracks unter- oder abbrechen. Die Lösung heißt: Disable Controls=yes in der Audioshow.

Kompliziertere Szenarios bewältigt man durch editieren oder löschen der Verknüpfungen der Controls in einer bestimmten Show. Mit dem Feld Controls einer Show kann ein Befehl mit einem anderen symbolischen Namen für jede Show verknüpft und so mit einer anderen Eingabe gesteuert werden. Alternativ können einzelne Befehle dieser Show gelöscht werden.

### 7.4.2 Beschränkungen bei konkurrierenden Shows

Konkurrierende Shows unterliegen einigen Beschränkungen, die von Linux, Mplayer und MPV verursacht werden.

- Wenn zwei Shows überlagernde Bilder oder Texte anzeigen, überdeckt das letzte Objekt in der Liste ältere Objekte.
- Video-Tracks überdecken immer Bilder und Texte.
- Wenn zwei Videos sich überlagern wird es möglicherweise ein Flackern geben.
- HD-Videos könnten ruckeln, wenn ein Audio-Track gespielt wird.
- Animations-Ausgaben verzögern sich wenn sie gleichzeitig mit einem Audio-Track oder einem Bild gestartet werden. Das Ausgabeereignis geht nicht verloren, aber es kommt mit Verspätung.

## 8 Show Control Befehle

Show Control Befehle wurden bisher für das Öffnen und Schließen von Shows durch andere Shows verwandt, wie der Name verrät. Sie wurden um verschiedene Funktionen erweitert, wie, nachfolgend aufgeführte.

Show Control Befehle können an diversen Stellen wirken:

- am Beginne und Ende eines Tracks
- am Beginn und Ende einer Show
- When eine Live-Liste einer Liveshow leer wird oder nicht leer ist
- Von Version 1.4.1a an mit einer Eingabe mit symbolischen Namen ([siehe Kap. 8.4](#)).

## 8.1 Befehle

Befehl	Funktion	Kapitel
open	Startet eine Show	<a href="#">8.1</a>
close	Beendet eine Show	<a href="#">8.1</a>
openexclusive	Schließt alle Shows und startet dann eine neue Show	<a href="#">8.1</a>
closeall	Beendet alle Shows	<a href="#">8.1</a>
exitpipresents	Beendet Pi Presents	
shutdownnow	Beendet Pi Presents und fährt den RPi herunter	
reboot	Beendet Pi Presents und startet den Rpi neu	
beep	Spielt eine kleine Audiodatei	<a href="#">8.5</a>
vibe	Erzeugt ein fühlbare Vibration	<a href="#">8.6</a>
monitor	Steuert den Monitor mit vcgencmd Befehlen	<a href="#">8.3</a>
cec	Steuert den Monitor mit cec Befehlen	<a href="#">8.3</a>
Osc, OSC	Startet einen OSC-Befehl	<a href="#">17</a>
event	Erstellt ein Eingabeereignis	<a href="#">8.2</a>
counter	Steuert Zählwerke	<a href="#">14</a>
backlight	Steuert Hintergrundbeleuchtung des PiTouch	<a href="#">9.3.1</a>
animate	Die Argumente können Daten vom Animation Befehl sein, mit Ausnahme des führenden Löschparameters. Für Tracks ist das Feld Animation flexibler.	<a href="#">15</a>

## 8.2 Ereignisse zwischen Shows austauschen

Der Befehl Event show control ermöglicht es einer Show, Ereignis-Eingaben für alle anderen laufenden Shows zu senden. Wie bei den von Benutzern erzeugten Ereignissen in [Kapitel 7.1](#) erhalten Tracks und Shows auch diese Eingaben und setzen sie in Kontrollbefehle um.

Befehl	Funktion
event <symbolic name>	Sendet ein Eingabeereignis mit einem symbolischen Namen an alle laufenden Shows

## 8.3 Steuerung des Monitors

Show-Kontrollbefehle können für die Steuerung des Monitors genutzt werden. Es werden zwei alternative Protokolle unterstützt.

Befehl	Aktion
monitor on <monitor-name>	Sendet den vcgencmd display_power 1 Befehl an den Monitor. Das erlaubt die Videoanzeige. Monitor-name kann sein DSI0, HDMI, HDMI0, HDMI1.
monitor off <monitor-name>	Sendet den vcgencmd display_power 0 Befehl an den Monitor. Das unterbindet die Videoanzeige. Monitor-name kann sein DSI0, HDMI, HDMI0,

	HDMI1.
cec standby <device>	Sendet den Befehl ,echo „standby <device>“   cec-client-s' zum HDMI-Ausgang – siehe unten Das aus wählende <device> ist die Gerätenummer mit 0 als Standard.
cec on <device>	Sendet den Befehl ,echo „on <device>“   cec-client-s' zum HDMI-Ausgang – siehe unten. Das aus wählende <device> ist die Gerätenummer mit 0 als Standard.
cec scan	Sendet den Befehl ,echo scan   cec-client-s d 1' zum HDMI-Ausgang. Im Terminal erscheint eine Liste von Geräten am CEC-Bus für die Fehlersuche – siehe unten.
cec as	Sendet den Befehl ,echo „as“   cec-client s' zum HDMI-Ausgang. Bestimmt den CEC-Adapter des Pi als aktiv.

Es könnte erforderlich sein, cec-utils zu installieren – sudo apt install cec-utils

Auf einem Pi4 funktioniert CEC aktuell nur am HDMI0-Ausgang.

## 8.4 Show mit Ereignissen steuern (Control on Event)

Das Feld Show Control on Event ermöglicht die Verknüpfung von Show Control Befehlen mit symbolischen Namen von Eingabeereignissen. Die Verknüpfung erfolgt ausschließlich für diese Show und ihre Media Tracks, also nicht für andere Shows, die als Show Tracks eingebunden sind. Das Feld Show Control enthält einen symbolischen Namen vor dem Show Control Befehl z.B.

```
pp-pause open myshow
```

Wenn ein Eingabeereignis geschieht, wird es mit der Liste der Show Control Befehle abgeglichen und die darin übereinstimmenden Befehle werden ausgeführt. Das Eingabeereignis wird dann für Track Befehle herangezogen.

## 8.5 Beeps

Der Beep Befehl spielt eine kurze Audiodatei auf dem aktuellen Audioausgang. Es gibt dafür keine Abspielbefehle

```
beep <file> <device>
```

Die zu spielende Datei wird im Befehl genannt. Sie kann entweder im pp\_home-Verzeichnis oder im Profilverzeichnis liegen:

```
+/myfile.wav. - Datei liegt im Verzeichnis pp_home
```

```
&/myfile.wav. - Datei liegt im aktuellen Profilverzeichnis
```

Erlaubt sind unkomprimierte Audiodateien, spielbar mit dem Linuxbefehl PAPLAY (.wav und andere) oder, wenn mpg123 installiert ist, .mp3-Dateien

```
sudo apt install mpg123
```

Das Feld device ist optional. Wenn nicht ausgefüllt, wird das auf der Taskbar eingestellte Audiogerät benutzt. Gültige Werte sind hdmi, A/V, USB, USB2, Bluetooth

## 8.6 Vibes

Der Vibe-Befehl verursacht auf dem DVR2605 Haptic Controller eine spürbare Vibration  
Der Befehl lautet:

```
vibe ,<sequence>
```

<sequence> in der Form 16,-100,16

Damit der Vibe Show Steuerbefehl funktioniert, muss darin die Befehlszeilenoption `--vibes` enthalten, und im Verzeichnis `pp_io_config` eine zugehörige `DRV2605.cfg` Datei vorhanden sein. Lediglich die Bibliothek und der Gerätenamen werden aus der Konfigurationsdatei abgerufen.

Die Benutzung des DVR2605 ist im Kapitel 17.2.8 und in der `DVR2605.cfg` - Datei in `/pipresents/pp_resources/pp_templates` umfassend erklärt

DRV2605 kann lediglich eine Vibration zur gleichen Zeit auslösen. Eine per Show Control Befehl gestartete Sequenz, wie Show Control on Event, stoppt laufendes Audio und vibriert in der Sequenz.

## 9 Displays und Touchscreens einrichten

### 9.1 Displays für den Raspberry Pi einrichten

Der Raspberry Pi 4 hat 2 HDMI-Ausgänge (Pi 3 und ältere 1 HDMI), einen A/V-Ausgang und den offiziellen über DSI verbundenen Touchscreen. Pi Presents kann mit jeglicher Kombination der beiden HDMI-Anschlüsse mit dem Touchscreen umgehen.

Pi Presents kann den A/V-Anschluss nicht nutzen. Bei Bedarf gibt es preiswerte HDMI-zu-A/V-Adapter.

Ein einzelner Raspberry Pi mit erweitertem Desktop bedient die Fläche aller Monitore. Im Gegensatz dazu erkennt Pi Presents alle angeschlossenen aktiven Monitore und konfiguriert automatisch das System mit separaten Anzeigen von Rotation und Größe der zugehörigen Monitore. Das Feld „Display“ im Show-Eintrag dient der Auswahl des gewünschten Monitors. Ein Eintrag in der `screen.cfg` benennt Klickbereiche und deren Verwendungsart.

Um die korrekte automatische Konfiguration von Bildschirmen durch Pi presents zu erreichen ist folgendes zu beachten:

- Alle gewünschten Bildschirme vor dem Start des Raspberry Pi anschließen. Einen Wechsel der Monitore im laufenden Betrieb kann Pi OS nicht immer erkennen. Es wird empfohlen, den Pi vor einem Bildschirmwechsel herunterzufahren und nach dem Neustart im Screen Configuration Menu sicherzustellen, dass der richtige Bildschirm aktiviert wurde.
- Wenn HDMI-Bildschirme nicht benötigt werden, sollten sie ausgestöpselt werden. Es reicht nicht aus, wenn der Monitor ausgeschaltet wird, weil er weiterhin vom Pi einen geringen

Strom für die Bildschirmerkennung abfordert.

- Um die angeschlossenen Monitore beliebig zu arrangieren ist das Pi OS Screen Configuration Menu – HDMI0 (HDMI-1), HDMI1 (HDMI-2) und DSI0 (DSI-1) - geeignet: vorausgesetzt, sie überlappen sich nicht und entweder die linke oder die obere Seite sind gewählt. Die Reihenfolge ist für Pi Presents beliebig, so, wie es der Projektgestaltung am besten nützt.
- Auch die Rotation der Bildschirme wird im Pi OS Screen Configuration Menu eingestellt.

Informationen über Pi Presents Darstellung der Displaykonfigurationen gibt es mit der Option -d in der Pi Presents Befehlszeile oder dem Kommandozeilenbefehl:

```
python 3 pp_displaymanager.py
```

When Pi Presents läuft, hängt die Bildschirmdarstellung von der Option im Fullscreen-Befehl ab:

- die Option `-fullscreen (-f)` zeigt den vollen Bildschirm ohne Dekoration.
- Ohne die Option `fullscreen` wird der einzelne Monitor oder mehrere Bildschirme per HDMI0 in einstellbarem Fenster dargestellt. Die Fenstergrößen können in `/pipresent/pp_display.cfg` eingerichtet werden.

### 9.1.1 Betrieb ohne Displays

Eine PiPr-Anwendung, die nur einen Audioplayer nutzt und mit dem Boot automatisch startet, benötigt keinen Bildschirm. Wenn aber weder DSI noch HDMI angeschlossen sind, bricht Pi Presents mit einem Fehler ab.

Der einfachste Weg, dies zu umgehen ist die Benutzung des Composite Videoausgangs wie folgt:

Zunächst sollte der VNC-Server im Pi gestartet werden, um einen VNC-Viewer auf einem anderen Computer starten zu können, mit dem man den Desktop des Pi ohne angeschlossenen Monitor erreichen kann.

In der `/boot/config.txt` müssen die Zeilen `hdmi_force_hotplug=1` so wie `hdmi_group` und `hdmi_mode` auskommentiert werden. Die Zeile `hdmi_force_hotplug` gaukelt dem Pi einen Monitor vor, obwohl keiner angeschlossen ist.

## 9.2 Display auswählen

Show-Einträge eines Profiles haben ein Feld „Display“, in dem das Display für die Show und ihre Subshows bestimmt wird. Das Feld „Display“ wird nur für die Top Level Show benutzt; Subshows übernehmen das Display von der Top Level Show.

## 9.3 Touchscreens einrichten

Touchscreens sind Monitore mit einer resistiven oder kapazitiven Oberfläche, die den Ort der Berührung messen. Für den offiziellen Pi-Touchscreen laufen Monitor- und Touchdaten über den DSI Anschluss. Bei HDMI-Bildschirmen wird die Touchoberfläche zumeist über USB mit dem Pi verbunden. Die Touchfähigkeit wird mit einem Softwaretreiber hergestellt. Treiber werden nicht immer von den Herstellern bereit gestellt.

Touchscreens müssen kalibriert werden. Das erledigt PI OS automatisch, wenn die Position und

die Stellung des Monitors sich ändern

Der offizielle Pi-Touchscreen verfügt über eine Hintergrundbeleuchtung, für deren Steuerung eine Software benötigt wird (siehe Kapitel 10.3.1)

### 9.3.1 Hintergrundbeleuchtung steuern

Show Control Befehle können die Hintergrundbeleuchtung des offiziellen PiTouch steuern. Dafür muss die python library rpi-backlight installiert sein und eine udev-Ausnahmeregel gesetzt sein. Weitere Informationen gibt es unter

<https://rpi-backlight.readthedocs.io/en/latest/installation.html>

Hinweis: Der Befehl lautet ‚pip3 install rpi backlight‘

Nach der Installation funktioniert Hintergrundkontrolle das mit Linux-Befehlen. Ob die Hintergrundbeleuchtung zur Verfügung steht sollte mit den rpi-backlight Befehlen kontrolliert werden.

PI OS merkt sich den Helligkeitsstatus zwischen den Einschaltphasen. Pi Presents arbeitet an diesen Einstellungen beim Start und setzt beim Beenden die Hintergrundbeleuchtung auf On und stellt die originale Helligkeit wieder her. Während Pi Presents läuft, dienen die folgenden Befehle der Kontrolle:

backlight on backlight off	Stellt die Hintergrundbeleuchtung an oder aus
Backlight set <level>	Wählt die Werte zwischen 0 und 100
Backlight inc <delta> backlight dec <delta>	Erhöht oder verringert die Helligkeitsänderung
backlight fade <level> <time>	Verschiebt die Hintergrundhelligkeit innerhalb eines Zeitraumes in Sekunden

## 10 Touchscreens und Soft Buttons

Einrichtung des Pi für Touchscreens siehe Kapitel 10

### 10.1 Shows mit dem Touchscreen steuern

Alle Showtypen können mit Touchscreens und Soft Buttons gesteuert werden. Touchscreens oder Soft Buttons sind nahezu unverzichtbar für Hyperlinkshows, bilden eine brauchbare Alternative zu GPIO-Tasten in Radiobuttonshows und sie können den Cursor oder die GPIO-Tastensteuerung in Mediashows, Liveshows und Menüs ersetzen.

- **Klickbereiche**

Klickbereiche gehören zum Touchscreen. Ein Klickbereich auf dem Bildschirm ist maussensitiv. Obwohl sie Klickbereiche genannt werden, taugen sie am besten für Touchscreens und sollten deshalb auch Touchbereiche heißen. Beim Berühren oder Anklicken des Bereiches auf dem Bildschirm wird ein Eingabeereignis ausgelöst. Bei Bedarf ist der Klickbereich mit einem Bild verbunden, möglicherweise von einem

Button, der auftaucht, wenn der Klickbereich aktiviert ist. Andererseits kann ein Klickbereich auch transparent sein und als Vieleck eingerichtet den Umriss eines Objektes im Hintergrundbild erfassen.

- **Soft Buttons**

Soft Buttons sind der Touchscreen für Arme. Sie wurden häufig in Echtzeitanwendungen benutzt, als es noch keine Touchbildschirme gab.

Soft Buttons sind eine Reihe von unbeschrifteten Tasten oder Knöpfen an den Rändern eines Bildschirms. Die Erläuterungen für die Knöpfe sind in der Software geschaffen und befinden sich auf dem Bildschirm in der Nähe der Knöpfe, die auf dem Rand des Bildschirms platziert sind. So können leicht verschiedene Knöpfe und Legenden zu jedem Track einer Hyperlinkshow hinzugefügt werden.

Pi Presents macht es möglich, dass beide Steuerungstechniken für jede Art von Show funktionieren.

## 10.2 Klickbereiche

In seiner screen.cfg Datei ermöglicht Pi Presents die Festlegung von Klickbereichen. Das sind vieleckige Bildschirmbereiche, die touch- und maussensitiv sind. Ein Touch oder Klick bewirkt ein Eingabeereignis, das über seinen symbolischen Namen erkannt wird.

Klickbereiche können Text, farbige Hintergründe, Umrandungen und Bilder enthalten.

Eine vorhandene screen.cfg Datei im /pp\_io\_config Verzeichnis im Profilordner aktiviert die Klickbereiche und enthält alle Klickbereiche, die für die Shows und Tracks des Profils eingerichtet wurden. Welche Klickbereiche in den Shows und Tracks zur Verfügung stehen sollen, wird mit den symbolischen Namen im Feld Controls von Shows und Tracks festgelegt. Die Einrichtung von Klickbereichen wird oben beschrieben ([s. Kap. 16.2.10](#)).

## 10.3 Soft Buttons

Soft Buttons benötigen beides: einen GPIO-Taster, der mit den erforderlichen Befehlen, wie andernorts bereits gezeigt, verbunden ist und zweitens eine passive Legende auf dem Bildschirm, die das Ziel des Knopfdrucks benennt, so wie es nachfolgend beschrieben ist. Die Legende soll nur gleichzeitig mit dem zugehörigen Track erscheinen. Deshalb muss sie in der screen.cfg als Klickbereich eingerichtet werden und im Feld Controls des Tracks aber mit einem speziellen Befehl deaktiviert werden, damit nur vom Button und nicht vom Klickbereich eine Eingabe verarbeitet wird.

Hier ein Beispiel für die Einrichtung eines Soft Buttons, der die Bewegung einer Seite in einer Hyperlinkshow steuert:

- Im Feld Controls eines Tracks den symbolischen Namen eintragen, der mit dem Button verknüpft ist

```
my-button call pig-video
```

- Platzieren Sie den Button nahe des Bildschirms und richten einen Klickbereich in der `screen.cfg` mit dem symbolischen Namen `my-button-legend`. Wählen Sie die Eckpunkte des Bereiches so, dass er sich in unmittelbarer Nähe des Buttons befindet. In diesem Klickbereich wird die Legende zwar angezeigt, wie z.B. „Schau das Schwein an“, sie soll aber nicht anklickbar sein.
- Um den Klickbereich mit dem entsprechenden Track erscheinen zu lassen, aber gleichzeitig das Warten des Klickbereichs zur Antwort abzustellen, geben Sie im Trackfeld Controls den Befehl 'no command' ein:

```
My-button-legend no-command
```

## 11 Audio einrichten

Der Pi Presents Profil Editor verweist auf Audio Outputs mit dem Namen des Output-Anschlusses wie auf der Platine ausgedruckt. - `hdmi`, `hdmi0`, `hdmi1` und `A/V`. Dazu kommen eventuell zwei USB-Soundkarten und ein Bluetooth-Audio-Adapter.

Pi Present benutzt Pulseaudio. Pi Presents Namen müssen in Pulseaudio Gerätenamen konvertiert werden. Das geschieht in einer speziellen Pi Presents Konfigurationsdatei:

```
/pipresents/pp_config/pp_audio.cfg
```

Die Pulseaudio Gerätenamen unterscheiden sich in den diversen Pi-Modellen, und werden bei externen Geräten erst erkannt, wenn die Geräte angeschlossen sind. Die Konfigurationsdatei muss also für USB- und/oder Bluetooth-Geräte editiert werden, das gilt auch, wenn Audio über HDMI nicht funktioniert. Details dazu finden sich in der Datei.

Informationen zu Pi Presents Interpretation der Audiokonfiguration des Pi erhält man mit der `-d` Option auf der Pi Presents Befehlszeile oder mit dem Aufruf `python3 pp_audiomanager.py`.

Pi Presents findet heraus, ob ein Gerät angeschlossen ist, in einem Profil steckt und ob das Profil aktuell läuft. Wenn ein Touchscreen benutzt wird oder wenn keine HDMI-Displays angeschlossen sind, eventuell zu Testzwecken, könnte eine Verbindungsfehler angezeigt werden. Das A/V-Audiogerät ist immer verbunden und kann in diesem Fall genutzt werden.

## 12 Laufzeitbefehle

Mplayer verfügt über eine Anzahl von Laufzeitbefehlen, die zu finden sind in:

<https://www.mplayerhq.hu/DOCS/man/en/mplayer.1.html#INTERACTIVE%20CONTROL>

Erlaubt sind nur Operationen mit einfachem Buchstaben aus der Referenzliste. Dem einfachen Buchstaben muss `mplay-` vorangestellt werden. Mehrfachoperationen werden aktuell nicht unterstützt-

Beispiele:

```
audiovolup mplay-/  
audiovoldown mplay-*
```

in Kontrollfeldern von Shows oder Tracks fügt Lautstärkeregelung des Mplayers hinzu, `audiovolup` ist ein symbolischer Name, der mit einer Taste, einem GPIO-Pin, etc. verknüpft werden kann. Tastaturen haben von Haus aus Wiederholungsmöglichkeiten, für GPIO-Pins können sie in der `gpio.cfg` aktiviert werden.

## 13 Dynamische Inhalte in einer Liveshow

Pi Presents war eigentlich nicht für die dynamische Darstellung von Medien gedacht. Auf vielfachen Wunsch habe ich ein Feature eingefügt, mit dem Tracks abgespielt werden können, die während einer laufenden Show wechseln.

Die `New>Liveshow-Vorlage` und `New>Artliveshow` sind aktive Liveshows. Die abzuspielenden Tracks sollen im Live Tracks Directory1, standardmäßig `/home/pi/pp_home/pp_live_tracks` gespeichert werden.

Liveshows spielen Audio-, Video- und Image-Tracks. Die Mediadataotypen, die von Liveshows erkannt werden, sind in den ersten Zeilen der Datei `pp_definitions.py` aufgeführt.

Der Pi Presents Manager ([siehe. Kap. 19](#)) kann Tracks hochladen oder importieren. Als Alternative können z.B. mit Filezilla Tracks per FTP in das Live Tracks Directory1 gesendet werden.

### Erweiterte Möglichkeiten:

Mit der `-l` Befehlszeilenoption von Pi Presents ist es möglich, einen zweiten Speicherort für Live-Tracks, das Live Tracks Directory2 zu wählen. Der Ort für dieses Verzeichnis wird mit der `-l` Befehlszeilenoption festgelegt. Die Dateien in den beiden Live Track Verzeichnissen werden zusammen behandelt und nach ihrem Dateinamen sortiert.

Das Verzeichnis könnte auf einem fernen Server liegen (das habe ich nicht getestet). Alternativ könnte das gesamte Profil auf einem Remote-Datei-Server gespeichert sein.

Seit der Pi Presents Version 1.3.1 können konkurrierende Shows, wenn nötig, jeder Liveshow gestatten, auf ihr eigenes Verzeichnis zuzugreifen. Das Liveshowprofil verfügt zu diesem Zweck nun über zwei Felder: `Live Tracks Directory1` und `Live Tracks Directory2`. Wenn diese Felder nicht leer sind, wird der standardmäßige Speicherort überschrieben.

## 14 Zähler

Zähler eignen sich für Quiz-Formate, sie sind aber auch für andere Zwecke geeignet. Alle Tracktypen bieten die Möglichkeit, Zähler zu entwerfen, einzusetzen, hoch oder herunter zu zählen und zu löschen. Aktuell können Zählerwerte nur durch Track Plugins dargestellt werden. Das erfordert ein wenig Pythonwissen (oder etwas copy and paste bei den Beispielen).

Zählerbefehle sind in den Feldern „Show Control Begin“ und „Show Control End“ von Tracks und Shows enthalten. Diese Befehle haben folgendes Format:

```
counter [name] [command] [parameters]
```

Das Counter-Feld muss vorhanden sein, um Zählerbefehle von anderen Befehlen in Show-

Control\_Feldern zu unterscheiden.

Die Befehle lauten:

Kommandos	Parameter	Beispiel	Zweck
set	name value	counter fred set 0	Erzeugt einen Zähler mit dem gewählten Namen und legt seinen Wert fest. Wenn der Zähler bereits existiert ist lediglich der Wert einzutragen.
inc	name value	counter fred inc 1	Setzt Wert höher (Wert muss positiv sein)
dec	name value	counter fred dec 2	Setzt Wert herunter (Wert muss positiv sein)
delete	name	counter fred delete	Löscht den Zählernamen.

Zähler werden im Python-Verzeichnis ‚counters‘ im Python Class Countermanager gespeichert, das in der Datei pp\_countermanager.py festgelegt ist.

Es gibt zwei Beispiele für Track-Plugins, die die Art der Anzeige von Zähler demonstrieren. Sie befinden sich im Verzeichnis /pipresents/pp\_track\_plugins

- krt\_counters.py  
das Track-Plugin zeigt alle Anzeigemethoden, die CounterManager bietet
  - get-counter (name) – gibt den Wert von counter name als Aufzählung wieder
  - str\_counters() - gibt eine Aufzählung aller aktuell vorhandenen Namen/Zähler aus
  - print\_counters() - gibt alle aktuell vorhandenen Namen/Zähler in einem Terminalfenster aus

Das Beispielprofil pp\_counters\_1p4 nutzt dieses Plugin.

- krt\_quiz.py

Wird vom Beispielprofil pp\_quiz\_1p4 genutzt und bietet eine schönere Ansicht.

Standardmäßig werden Zähler und ihre Ergebnisse vernichtet, wenn Pi Presents beendet wird. Im nächsten Abschnitt wird beschrieben, wie Zähler und Ergebnisse erhalten werden können, und zwar zwischen verschiedenen Läufen von Pi Presents.

Zählergebnisse können in einer Datei zwischen Läufen von Pi Presents gespeichert werden. Sie befinden sich dann in der Datei counters.cfg im benutzten Profil. Das ist eine Textdatei. Die mit einem Editor gelesen und verändert werden kann.

Das Speichern wird aktiviert, wenn im Counters Tab in der Start Show „yes“ bei Store Counters eingetragen ist. So werden alle Zählerergebnisse in der Textdatei bei Änderungen berücksichtigt und Zähler werden mit dieser Datei geladen, wenn Pi Presents neu gestartet wird.

Um Zähler zu benennen und ihren Anfangswert festzulegen dient die Option –counters im Befehl

zum Start von Pi Presents, `starting pipresents.py`. Wenn diese Option gewählt ist, wird `counters.cfg` bestückt nach dem ‚Initial Values‘ Feld im Counters Tab. Das Format dieser Datei ist gleich dem in `python's configparser`:

```
fred = 1
bill = -1
```

Es gibt auch weitere Möglichkeiten, Zähler zu starten, wie z.B. eine spezielle Show, gesteuert durch einen versteckten Starter, oder durch den Time of Day Scheduler.

## 15 Animationen steuern

Mit allen Tracktypen können Animationen gesteuert werden. Mit den Befehlen in 'Animation at Beginning' und 'Animation at End' Feldern kann eine externe Ausgabe ein- oder ausgeschaltet werden und dabei mit dem Start oder Ende eines Tracks synchronisiert werden. Das ist auch zeitversetzt möglich.

Hier ein Beispiel für Animationsbefehle:

animate at beginning	1 out1 state on 2 out1 state off 1 out2 state on 6 out3 state on
Clear Animation	no
Animation at End	0 out2 state off

Ein Befehl hat vier oder mehr durch Leerraum getrennte Felder und wird mit einer neuen Zeile beendet:

- Verzögerung – Sekunden als positiver Wert oder 0
- Symbolic Name – Name der Ausgabe, wie er in der `gpio.cfg` Datei des Profils festgelegt ist ([siehe Kap. 16.2.2](#))
- Parametertyp
- Parameter – ein oder mehr Felder die kompatibel mit dem Parametertyp sind.

Befehle für Animationen am Beginn werden mit Beginn eines Tracks ausgeführt, Befehle für die Animation am Ende am Ende eines Tracks. Wenn die Befehle festgelegt wurden, werden sie in einer Warteschlange gespeichert um zur rechten Zeit „abgefeuert“ zu werden. Die Zeit wird durch die Pausierung eines Tracks nicht beeinflusst. Standard

Jeder Befehl richtet sich an alle aktivierten Plugins. Wenn symbolischer Name und Parametertyp mit einem der in den Plugins Eingebauten übereinstimmt, wird der Befehl mit den zur Verfügung gestellten Parametern ausgeführt.

Animationsbefehle in der Warteschleife sind auch nach Beendigung eines Tracks vorhanden und wirken sich auch auf weitere Tracks aus. So kann es passieren, dass ein Output zum falschen Zeitpunkt wirkt, wenn die Dauer des Tracks nicht begrenzt ist. Um das zu vermeiden und Outputs

in einen definierten Zustand zu bringen, sollte am Ende eines Tracks Animate clear mit „yes“ gesetzt werden. Wenn „yes“ gilt, bevor Animation am Ende ausgeführt wird, wird die Warteschlange von den Ereignissen gesäubert, die der Track zwar enthält, aber noch nicht „abgefeuert“ wurden.

## 15.1 Zeitprobleme bei Animationen

Wenn Pi Presents beim Booten automatisch gestartet wird, kann das Setzen der Uhrzeit direkt nach dem Einwählen ins Internet Probleme mit den Animationszeiten verursachen. Um das zu verhindern:

- a. Das WLAN beenden und /oder das Ethernetkabel abziehen
- b. In den Raspberry Pi Einstellungsmenü „Nicht auf das Netzwerk warten“ wählen
- c. In die PiPresents-Befehlszeile die -n Option einfügen, so dass PiPr nicht auf das Netzwerk wartet

## 16 Input/Output (I/O) Plugins

In Pi Presents sind Eingaben und Ausgaben unabhängig von den eigentlichen Kernen der Shows und Tracks. Ab Version 1.3.3 können Entwickler ihre Input/Output-Geräte einbinden, indem sie weitere I/O-plugins erstellen. Diese Plugins haben eine Standard-API, die per Konfigurationsdatei eingerichtet und gestartet werden.

Ein Input-Plugin nimmt eine Eingabe eines physikalischen Gerätes entgegen, verarbeitet sie und erstellt ein Input-Ereignis mit einem symbolischen Namen. Ein Output-Plugin nimmt einen Ausgabebefehl (z.B. out1 state on) mit einem symbolischen Namen und Parametern und erzeugt eine physikalische Ausgabe.

I/O-Plugins sind Python3-Module im Verzeichnis /pipresents/pp\_io\_plugins. Ein I/O-Plugin muss mit einer .cfg-Datei konfiguriert werden. Diese Dateien werden im Verzeichnis /pp\_io\_config innerhalb eines Profiles gespeichert. Es können mehrere .cfg-Dateien für ein Plugin sein. So kann zum Beispiel pp\_inputdevicedriver.py für jede Fernsteuerung eine .cfg-Datei haben.

Die meisten I/O-Plugins haben eine Standard-API und Standard-.cfg-Dateien. Aktuell gibt es I/O-Plugins mit Standardkonfigurationsdateien und Standard-API für:

Gerät	Konfigurationsdatei	Richtung	I/O-Plugin	Aktion
Tkinter Keyboard	keys.cfg	input	pp_kbdriver.py	Verbindet mit einem Keyboard, das die Tkinter-API benutzt.
GPIO	gpio.cfg	Input/Output	pp_gpiodriver.py	Spricht die GPIO-Pins per RPI.GPIO an.
Input Device	osmcremot.cfg	Input	pp_inputdevicedriver.py	Spricht Fernbedienungen, keypads etc. per evdev an.

Serial Device	Serialdriver.cfg	Input/Output	pp_serialdevicedriver.py	Interface mit serial devices üblicherweise USB to RS232 Konverter
I2C devices	i2c.cfg	Input/Output	pp_i2cdriver.py	Interface mit I2C-Geräten
RFID Tag Reader	Pn532driver.cfg	input	pp_pn532driver.py	Interfaces mit PN532basierten Lesegeräten
DRV2605 Haptic Controller	DRV2605.cfg	Output	pp_DRV2605driver.py	Controls ERM and LRA Haptic actuators

Es gibt zwei I/O-Plugins mit Nicht-Standard-Konfigurationsdatei und Nicht-Standard-API. Ihre Eigenschaften sind wie folgt hinterlegt:

touchscreen	screen.cfg	Input		Aktiviert touch/klick-sensitive Bereiche auf einem Touchscreen/Monitor und bestimmt das Aussehen der touchsensitiven Bereiche
Remote control using OSC	osc.cfg	Input/Output		Erlaubt Pi Presents andere Geräte zu steuern oder selbst gesteuert zu werden. Die Konfigurationsdatei wird mit dem Editor erzeugt.

## 16.1 Ein Standard-I/O-Gerät einrichten

Um ein I/O-Gerät zu aktivieren, muss die zugehörige Konfigurationsdatei in einem /pp\_io\_config Verzeichnis des Profils vorhanden sein. Beispieldateien finden sich in /pipresents/pp\_resources/pp\_templates. Sie können kopiert und editiert werden. Diese Dateien enthalten zusätzlich zu diesen Informationen weitere Hinweise für ihre Anwendung.

Beim Start durchsucht Pi Presents zunächst das /pp\_io\_config-Verzeichnis innerhalb des Profiles und nutzt eine dort vorhandene .cfg-Datei. Wenn dort keine .cfg-Datei liegt, sucht Pi Presents in /pipresents/pp\_io\_config nach einer gleichnamigen Datei.

Nach der Installation befindet sich die keys.cfg in /pipresents/pp\_io\_config, so ist sogleich das Tkinter-Keyboard aktiviert.

Die Konfigurationsdateien sind sämtlich Textdateien, die mit dem Editor Leafpad editiert werden können. Bitte die Dateien nicht innerhalb von /pp\_templates speichern; wenn sie geändert werden sollen, kopieren sie die Dateien in ein Profil, zum Beispiel in:

```
/home/pi/pp_home/pp_profiles/myprofile/pp_io_config
```

oder, wenn sie von allen Profilen genutzt werden sollen in:

`/pipresents/pp_io_config`

Beachten Sie beim Editieren dieser Dateien, dass Pi Presents die Inhalte nur teilweise überprüft. Wenn Sie die Dateien verändern wollen, sollten Sie Pi Presents zuvor in einem Terminalfenster starten, in dem etwaige Pythonfehlermeldungen angezeigt werden.

## 16.2 Eingabe/Ausgabe: Treiber einrichten

[Kapitel 7.1](#) beschreibt das Eingabesystem von Pi Presents und zeigt, wie externe physikalische Ereignisse in Eingaben mit symbolischen Namen umgewandelt werden. Die meisten Systemantworten können mit den in diesem Kapitel beschriebenen Dateien konfiguriert werden. Beim Editieren der I/O-Konfigurationsdateien sind symbolische Namen erforderlich. Es ist ratsam, Namen nicht mit `pp_` beginnen zu lassen. Das verhindert Kollisionen mit Namen, die Pi Presents benutzt.

Alle I/O-Konfigurationsdateien müssen einen Bereich mit Namen [DRIVER] enthalten. Darin müssen folgende Felder enthalten sein:

Feld	Beispiel	Erklärung
title	GPIO	Text über Aktivitäten des Plugins für Fehlerreports und Logs.
enabled	yes	yes/no. Ein I/O-Plugin wird aktiv, wenn eine <code>.cfg</code> -Datei im zugehörigen Verzeichnis liegt und <code>=yes</code> eingetragen ist. Das Feld erlaubt es, ein I/O-Plugin zu deaktivieren, ohne die Konfigurationsdatei zu entfernen.
module	pp_gpiodriver	Der Name des Pythonmoduls, das das I/O-Plugin einsetzt (ohne <code>.py</code> ). Die Datei muss im Verzeichnis <code>/pipresents/pp_io_plugins</code> liegen.  Es können sich mehrere <code>.cfg</code> -Dateien auf das gleiche Modul beziehen, beispielsweise für zwei Fernsteuerungen mit unterschiedlicher Tastenbelegung.

Die Sektion [DRIVER] kann weitere Felder enthalten, die von einem bestimmten I/O-Plugin benutzt werden.

### 16.2.1 Verknüpfen mit GPIO per `pp_gpiozeroriver.py`

ANMERKUNG: Dieser Treiber arbeitet zusammen mit der `gpiozerolibrary`:

<https://gpiozero.readthedocs.io/>

Der Treiber funktioniert auf dem Pi5 und allen anderen Pi Modellen. Er ist der bevorzugte Treiber

für GPIO.

**ACHTUNG:** Wenn Sie einen Pin als Ausgang nutzen, wobei der Ausgang mit +3,3 Volt oder Ground kurzgeschlossen wird, könnte das Ihren Pi grillen. Legen Sie zum Schutz vor jeden Ein- und Ausgang einen Serienwiderstand.

Diese Datei in

/pipresents/pp\_resources/pp\_templates/gpiozero.cfg

ist ein Beispiel und, wenn sie in den Ordner

/pp\_io\_config

innerhalb eines Profiles abgelegt ist, aktiviert sie GPIO für die Steuerung von Pi Presents. Die Einstellung der Pins entspricht der Einstellung in einigen Beispielprofilen und kann als Grundlage für eigene Profile übernommen und nach Bedarf geändert werden.

Die .cfg-Datei bildet die physischen P1-Konnektoren, GPIO Input- und Output-Pins, auf den symbolischen Namen der von Pi Presents-Profilen gebrauchten Inputs und Outputs ab. In Ihr werden auch die Input-Pins konfiguriert.

Für jeden Pin muss ein Eintrag in der Datei vorhanden sein. Ein Pin mit direction=none wird ignoriert.

## Inputs

Anmerkung: Anders als andere GPIO-Treiber benutzt gpiozero „active“ und „inactive“ statt „rising“ und „falling“. Eine Taste ist active, wenn sie gedrückt wird und inactive, wenn sie gelöst wird. Die angelegte Spannung für active und inactive wird in den pull-up oder active-state Feldern geregelt.

Wenn ein Taster zwischen input und 0 Volt (bevorzugte Verbindung) geschaltet wird, sollte das pull-up Feld auf high gesetzt sein, so dass sich der interne pull-up Widerstand zwischen input und 3,3 Volt befindet. In diesem Fall ist active, wenn der Taster bei Druck mit 0 Volt verbunden wird.

Wenn ein interner Widerstand nicht erforderlich ist sollte pull-up auf none stehen und active state auf high oder low stehen, je nach Anforderung.

Die relevanten Felder in der .cfg-Datei sind:

- pull-up – interner pull up. Werte: up/down/none.up to +3V, down to 0V. Wenn up aktiviert ist, dann 0V, wenn down aktiviert ist,3V
- active state - high/low, high bedeutet +3V=active, low bedeutet 0V=active. Muss eingestellt werden nur, und nur, wenn pull-up auf none steht.

Jede Eingabe kann Ereignisse mit symbolischen Namen auf dreifache Weise auslösen.

- Wenn der Eingabe-Pin von inactive auf active wechselt (activated-name)
- Wenn der Eingabe-Pin von active auf inactive wechselt (deactivated-name)
- Wenn ein Pin für eine Zeit lang im active state gehalten wird (held-name)

Die relevanten Felder sind:

- activated-name – Symbolischer Name des activated-input-Ereignisses. Wenn leer, ist kein Ereignis vorgesehen
- deactivated-name – Symbolischer Name des deactivated-input-Ereignisses. Wenn leer, ist kein Ereignis vorgesehen.
- Held-name – Symbolischer Name des activated-hold-Ereignisses, das nach der Haltezeit erfolgt. Wenn leer, ist kein Ereignis vorgesehen.

Das held-name-Ereignis geschieht nach hold-time-Sekunden. Wenn hold-repeat ist yes, dann wiederholt sich das held-name-Ereignis nach repeated-time-Sekunden.

Die relevanten Felder sind:

- hold-time – Zeit in Sekunden des laufenden Inputs bis das Ereignis startet, wie z.B. Taster
- hold-repeat – yes/no Wiederholt das hold-Ereignis nach repeat\_time
- repeat-time - Zeit in Sekunden die ein Input laufen muss bevor das hold-Ereignis wieder startet

Mechanische Kontakte, z.B. Taster, können für einen Zeitraum prellen/entprellen (Kontakt besteht und wird unterbrochen). Bounce-time kann diesen falschen Statuswechsel beseitigen. Ein Wert von 0.1 Sekunden ist typisch.

- bounce-time – Eingabe muss für diese Zeit in Sekunden stetig bleiben, damit der Statuswechsel erkannt wird.

Auf die Eingabe kann eine Ausgabe folgen. Wenn eine solche Verknüpfung erfolgen soll, bitte active-high für den output-pin setzen:

- linked-output – Gerätename am Ausgabe-Pin für Ausgabe, die der Eingabe folgt.

## Outputs

Anmerkung: Gegenüber anderen GPIO-Treibern nutzt gpiozero on und off als active oder inactive. Active-high bestimmt, das active 0V oder 3,3Volt ist.

Raspberry Pi initialisiert GPIO-Ausgaben nach 0V.. Es empfiehlt sich deshalb, active-high auf high zu setzen und Relais z.B. für positive logic einzurichten (was die meisten auf dem Markt Erhältlichen aber nicht sind). Die Ausgaben werden auf 0V zurückgesetzt, wenn Pi Presents beendet wird.

Die Felder sind:

- name – Symbolischer Name im Ausgabebefehl
- active-high – high/low, liefert für den high On Status +3,3V und für den low On Status 0V

Zwei Animationsbefehle nutzen den gpiozero-Treiber.

### GPIO Status Kontrolle

Das Kommando lautet:

```
delay output name parameter_type parameter_value
```

z.B. 0 LED Status on

- Der output\_name wird mit den Namensfeldern aller Pins mit direction=out abgeglichen
- Das Feld parameter\_type sollte state für die GPIO-Statuskontrolle zeigen. Der Animationsbefehl wird nur ausgeführt, wenn der output\_name und parameter\_type

zusammenpassen

- Der Parameterwert muss on oder off lauten, sonst erscheint eine Fehlermeldung.

### Blinken einer GPIO-Ausgabe

Das Kommando lautet:

```
delay output name parameter_type parameter_values
```

- Der output\_name wird mit den Namensfeldern aller Pins mit direction=out abgeglichen
- Das Feld parameter\_type sollte blink für die Blinkkontrolle zeigen. Der Animationsbefehl wird nur ausgeführt, wenn der output\_name und parameter\_type zusammenpassen

Es gibt drei Felder für die Parameterwerte: on\_time, off\_time and n

- on\_time – die Zeit in Sekunden, in denen es blinkt
- off\_time - die Zeit in Sekunden, in denen es nicht blinkt
- n – die Zahl der Blinkzyklen. Bei 0 blinkt es kontinuierlich und wird erst durch den Animationsstatus off gestoppt.

## 16.2.2 Verbindungen mit GPIO per pp\_gpiodriver.py

Anmerkung: Dieser Treiber benutzt RPI.GPIO library, die veraltet ist und nicht mit dem Pi5 zusammengeht. Stattdessen sollte gpiozerodriver benutzt werden.

**ACHTUNG:** Ein als Ausgang genutzter Pin mit der Ausgabe von 3,3V oder gegen ground könnte den Pi grillen. Zum Schutz davor bitte an jedem Eingabe- und Ausgabepin einen seriellen Widerstand verwenden.

Wie GPIO von Pi Presents standardmäßig genutzt wird zeigt die Datei gpio.cfg.

Diese Datei in /pipresents/pp\_resources/templates/gpio.cfg ist eine Vorlage, die Pi Presents für Buttons und einen Bewegungssensor einrichtet, wie im Handbuch beschrieben und von den Beispielprofilen umgesetzt wird.

Die .cfg Datei verknüpft die GPIO Pins für Eingabe und Ausgabe mit den symbolischen Namen, die in den Profilen benutzt werden. Außerdem legt sie den Eingabepin P1-12 als shutdown-Button für den Raspberry Pi fest.

Für jeden Pin muss ein Abschnitt in der Konfigurationsdatei vorhanden sein. Ein Pin mit directe=none wird bei der Ausführung ignoriert.

### Inputs

Jeder Pin kann ein Ereignis, das mit einem symbolischen Namen verknüpft ist, auf vierfache Weise auslösen:

- rising edge – Ein Ereignis mit dem symbolischen Namen im Feld ‚rising-name‘ tritt ein, wenn der Input von 0 auf 1 (0 Volt zu 3,3 Volt) wechselt

- falling edge - Ein Ereignis mit dem symbolischen Namen im Feld ‚falling-name‘ tritt ein, wenn der Input von 1 auf 0 (3,3 Volt zu 0 Volt) wechselt
- one state – Ein Ereignis mit dem symbolischen Namen in ‚one-name‘ tritt wiederholt laut ‚repeat‘ ein, solange der Input 0 beträgt (0 Volt). Erstmals tritt das Ereignis nach dem ‚repeat‘-Intervall ein. Soll es sofort eintreten, muss das rising-edge-Ereignis den gleichen symbolischen Namen erhalten
- zero state – Ein Ereignis mit dem sControls ERM and LRA Haptic actuatorsymbolischen Namen in ‚zero-name‘ tritt wiederholt laut ‚repeat‘ ein, solange der Input 1 beträgt (3,3 Volt). Erstmals tritt das Ereignis nach dem ‚repeat‘-Intervall ein. Soll es sofort eintreten, muss das falling-edge-Ereignis den gleichen symbolischen Namen erhalten

Wenn das Ereignis nicht eintreten soll, keinen symbolischen Namen eintragen.

Die gpio.cfg ist für die Arbeit mit dem Handbuch und den Profilbeispielen für Schließer-Buttons gegen ground (0 Volt) und für PIR (Bewegungsmelder) mit normalerweise geschlossenem Relaiskontakt gegen ground voreingestellt.

#### Entprellung

Entprellung wird für jeden einzelnen Pin mit dem Schwellenwert-Parameter (threshold) festgelegt. Die Entprellung soll mechanische Kontaktprobleme (ungewollte Mehrfachkontakte) verhindern.

#### Verlinkte Outputs

Die Felder inked-output und linked invert erlauben eine direkte Verbindung mit einem gpio-input.

### Outputs

Der logische On-Status liefert 3.3Volt, der Off-Status 0 Volt.

Pi Presents richtet GPIO Outputs mit 0 Volt ein, so empfiehlt es sich, anzuschließende Relais etc. mit positiver Logik (von denen es auf dem Markt allerdings wenige gibt) zu verwenden. When Pi Presents beendet wird, werden die Outputs auf 0 Volt zurück gesetzt.

Bei Anwendung des animation-command

```
delay output_name parameter_type parameter_value
```

- Der output\_name wird mit allen Pin-Namen abgeglichen, die als direction=out definiert sind
- Das Feld parameter\_type sollte state lauten, weil dies der Paramertyp ist, den pp\_gpiodriver.py unterstützt. Der Animationsbefehl wird bei anderslautendem Status ignoriert, auch wenn die Outpu-Namen übereinstimmen.
- Der Parameterwert muss on oder off sein, andernfalls wird ein Fehler angezeigt

#### 16.2.2.1 Eine besser reagierende Alternative – pp-pigpiodriver.py

ACHTUNG: Dieser Treiber benutzt die RPI.GPIO Bibliothek, die überholt ist und nicht mit dem Raspberry Pi 5 funktioniert. Ersatzweis sollte gpiozodriver.py installiert werden.

Pp-pigpiodriver.py funktioniert wie pp-gpiodriver.py. Allerdings basiert es auf Joans pigpio (<http://abyz.me.uk/rpi/pigpio/>) statt auf GPIO.IO.

Die Inputs werden alle 5 Millisekunden per DMA statt alle 10 Sekunden abgefragt. Das bringt eine weit bessere Reaktionszeit bei kurzem Knopfdruck, besonders wenn MPV in mehreren Instanzen läuft und Streaming-Cameras anzeigt.

Die Datei pigpio.cfg in *pipresentspp\_resources/pp\_templates* ist eine Beispiel für eine Konfigurationsdatei des Treibers. Alle Felder der Pins-Sektionen sind gleich mit den Feldern in der *gpio.cfg* mit der Ausnahme, das für ‚threshold‘ nun ‚steady‘ steht.

Steady – das ist die Zeit in Millisekunden in der ein Input stabil sein muss, damit ein Statuswechsel erfolgt – siehe: <http://abyz.me.uk/rpi/pigpio/python.html> #set\_glitch\_filter.

Um den Treiber zu aktivieren muss der pigpio Dämon beim Booten mit `sudo pigpiod` gestartet werden, entweder im Terminal oder im Autostart.

### 16.2.3 Mit *pp\_inputdevicedriver.py* Fernbedienung oder Keypad verbinden

Fernbedienungen usw., die in Verbindung mit Linux Userland eine Datei in `/dev/input` und Tastenbelegungen bieten, sollten auch mit Pi Presents funktionieren, indem sie das *pp\_inputdevicedriver.py* I/O-Plugin benutzen. Die Beispieldatei *osmcremote.cfg* richtet dieses Plugin für den Gebrauch mit der SMC-Fernbedienung ein.

<https://osmc.tv/store/product/osmc-remote-control/>

Die [DRIVER]-Sektion muss folgende zusätzliche Felder enthalten:

Feld	Inhaltsbeispiel	Erklärung
device-name	HBGIC Technology Co. Ltd. USB Keyboard Mouse	Name des Input-Gerätes.  Nur Eingaben dieses Gerätes erzeugen ein Eingabeereignis. ( außer Duplicate Events, s. Unten).  Das Programm <i>input_device.py</i> zeigt den Namen des Eingabegerätes an.
key-codes	KEY_STOP.KEY_PLAYPA USE.KEY_DOWN.....	Liste der Tastenbefehle auf die Pi Presents reagiert.  <i>input_device.py</i> starten, um die Tastenbefehle der Fernbedienung zu finden.
tick-interval	50	Das Zeitintervall zwischen Auswahlen auf der Fernbedienung in Millisekunden mS.

Es gibt Bereiche für jede Taste, die den Bereichen in der *gpio.cfg* ähnlich sind. Einzelheiten siehe in:

`/pipresents/pp_templates/osmcremote.cfg`

ACHTUNG: Wenn zwei Eingabeereignisse von Fernbedienungstasten auftreten, bitte Tkinter Keyboard ([siehe Kap. 16.2.6](#)) lesen.

### 16.2.4 Serielle Verbindung mit pp\_serialdriver.py einrichten

Serielle Verbindungen wie RS232 werden häufig für die Bedienung von Projektoren benutzt. Der Raspberry Pi bringt einen eingebauten seriellen Anschluß mit, der für Gerätefehlersuche eingerichtet ist. Er ist aber schwer zu bedienen und verlangt nicht selten Stromstärken-Konverter. Besser ist ein USB auf RS232 Adapter. Diese Adapter werden unter Linux als Datei /dev/ttyUSB0 eingerichtet. Sie können die pynserial python library nutzen.

Das pp\_serialdriver.py I/O plugin steuert über die pynserial library die seriellen Geräte. Es wurde mit einem USB-Gerät an einer Win XP und einer Win 98-Maschine getestet. Die Vorlage serialdriver.cfg im Verzeichnis /pipresents/pp\_resources/pp\_templates zeigt den Gebrauch des I/O-plugins. Im Beispiel dient die Verknüpfung der Steuerung eines Projektors. Ein anderer Projektor muss neu konfiguriert werden.

Details der Konfiguration finden sich in

/pipresents/pp\_resources/pp\_templates/serialdriver.cfg

Der Treiber regelt sowohl Input als auch Output.

#### Input

Zeichen, die über den Link eintreffen, erreichen Zeichen und Zeichenketten, wie sie in der Konfigurationsdatei definiert sind. Wenn ein „Treffer“ erfolgt, wird ein Ereignis mit symbolischen Namen eintreten, wie es ebenfalls in der Konfigurationsdatei vordefiniert ist:

- Jedes eingehende Zeichen wird mit den Konfigurationsdaten abgeglichen, um ein Eingabeereignis zu generieren.
- Wenn das Ende einer Zeichenfolge erkannt ist, werden die danach erhaltenen Zeichen mit den Konfigurationsdaten abgeglichen, um ein Eingabeereignis zu generieren.

Ereignisse können durch beliebige oder spezielle Zeichen so wie durch beliebige oder besondere Folge generiert werden.

#### Output

Ausgaben über den seriellen Link bestehen aus Animationsbefehlen. Es gibt zwei Arten, diese Befehle anzuwenden, abhängig von den Feldwerten in den Konfigurationsdaten:

- preset – die Zeichen für die Ausgabe sind in den Konfigurationsdaten definiert. Die Konfigurationsdaten definieren die zu sendende Botschaft jede Kombination von Namen, Parametertyp und Parameterwert. Der Treiber übersetzt das in die zu sendende Botschaft.

Beispielsweise wird der Befehl ‚0 projector state on‘ in eine Bytesequenz 02 00 00 00 01 übersetzt.

- explicit – die Zeichen/Bytes zur Ausgabe befinden sich im Animationsbefehl. Beispiele sind:

```
0 serial-send bytes „02 00 00 00 01“
0 serial-send string „the cat sat on the mat“
```

### 16.2.5 Verbindung über I2C-Geräten mit pp\_ic2driver.py

Viele Geräte können auf dem I2C-Bus mit dem Raspberry Pi kommunizieren. pp\_i2cdriver.py unterstützt ein ganze Reihe von Geräten, die für Pi-Presents-Nutzer interessant sind. Der

Hauptzweck dieses Treibers ist es jedoch, ein Beispiel für eine Verbindung des eigenen Gerätes mit Pi Presents zu zeigen.

Um die pp\_i2cdriver.py nutzen zu können, muss I2C in den Einstellungen des Raspberry Pi aktiviert sein.

Folgende Geräte werden von pp\_i2cdriver.py unterstützt:

Device	Input/ Output	Symbolic Name	Parameter Type
Adafruit MCP472 5 DAC	Output	dac	<p>set – setzt DAC to auf einen Prozentanteil ('percentage'*** von 3.3 volts</p> <p>mirror – spiegelt den benannten ADC input (analog1,analog2,analog3)</p> <p>fade – erhöht den Output von a nach b in t Sekunden; a und b sind prozentual ('percentages'***) ausgedrückt.</p>
Pimoroni Four Letter Phat	Output	fourletter	<p>string – zeigt eine 4-Buchstaben-Zeile</p> <p>num-string – zeigt eine vierstellige Zahl mit Dezimalpunkt</p> <p>blank – leer</p> <p>mirror-voltage – zeigt die Voltzahl für die ADC-Eingabe</p> <p>mirror-percentage – zeigt den prozentualen Wert (percentage***)des ADC-Input</p> <p>countdown – zählt von den angegebenen Sekunden in Minuten und Sekunden herunter</p>
Pimoroni Scroll HD LED matrix display	Output	scrollhd	<p>scroll - scroll einzelne oder mehrere Textzeilen</p> <p>static – zeigt statischen Text</p> <p>static-high – eine hohe brightness variant of static</p> <p>blank - leer</p>

Pimoroni Automaton Phat (ADS1015 ADC)	Input		Die drei analogen Input-Kanäle des HAT werden kontinuierlich alle 100 Millisekunden ausgelesen. Die Ergebnisse werden spiegeloptisch aufbereitet und können so von Track Plugins über den <code>get_input</code> Modus des I/O-Plugin-Managers aufgenommen werden. Die Keys sind: <code>analog1-volts</code> <code>analog2-volts</code> <code>analog3-volts</code> <code>analog1-percentage</code> <code>analog2-percentage</code> <code>analog3-percentage</code> Hinweis: Die digitalen inputs und outputs so wie das Relais des HAT laufen über die GPIO-Pins, gesteuert per <code>pp_gpiodriver.py</code>
---------------------------------------	-------	--	--

\*\*Percentage – eine Zahl zwischen 0 und 100, benennt den prozentualen Wert von 3,3 Volt für den ADC-Input.

Die Datei `/pipresents/pp_resources/pp_templates/i2c.cfg` enthält Einzelheiten zu den implementierten Befehlen und Methoden.

### Vom I2C plugin genutzte Bibliotheken

Das Scroll HD und Four Letter Phats nutzen die Pimoroni-Bibliotheken. Einige Methoden sind lediglich übernommene Pimoroni-Beispiele für die Benutzung dieser Geräte. Die Bibliotheken sind im Pi-OS-Image enthalten.

Die Automation PHAT und Adafruit-Bibliotheken werden nicht benötigt, weil Pi Presents seine eigenen Bibliotheken nutzt. Sie befinden sich in `pp_i2cdevices.py`. Es könnte allerdings erforderlich sein, dass `smBus`-Modul zu installieren.

### 16.2.6 Tkinter: Tasten mit `kbdriver.py` einrichten

Pi Presents benutzt Tkinter, eine Tastatur/Maus-Schnittstelle, die mittels `pp_kbdriver.py` angesteuert und per `keys.cfg` konfiguriert wird. Diese Schnittstelle ist die Einzige, die bereits beim Start von Pi Presents aktiviert ist. Sie besitzt die Datei `/pipresents/pp_io_config/keys.cfg`.

Die [DRIVER]-Sektion der `.cfg`-Datei muss folgende zusätzliche Felder enthalten:

field	content	Erklärung
<code>bind-printing</code>	<code>yes/no</code>	Wenn <code>yes</code> , sind sämtliche Drucktasten mit den unten stehenden symbolischen Namen verbunden.

Die Tastaturtasten sind an die symbolischen Namen in der `.cfg`-Datei angebunden. Die Datei hat etliche Zeilen im Format

Bedingung = symbolischer Name

Die Bedingungen (conditions) sind in

<https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-events-and-bindings.htm>

oder

<https://effbot.org/tkinterbook/tkinter-events-and-bindings.htm>

in den Sektionen <Return>, and <Shift-Up> definiert. Bedingungen und symbolische Namen unterscheiden zwischen Groß- und Kleinschreibung.

Zusätzlich zu diesen Verknüpfungen sind die Buchstaben auf den Tasten (die den Bedingungen <Key> in der Referenz folgen) automatisch mit den symbolischen Namen pp-key-x verknüpft, wenn bind-printing = yes.

z.B. realisiert die a-Taste pp-key-a

Die automatische Anbindung einer Taste kann hinzugefügt oder überschrieben werden, wenn eine Zeile wie

```
a=pp-pause
```

eingefügt wird.

Die Standard keys.cfg in /pipresents/pp\_io\_config hat folgende Verknüpfungen:

Symbolic Name	Bound Key	Command
pp-down	Cursor Down	down
pp-up	Cursor Up	up
pp-play	Return	play
pp-pause	Spacebar	pause
pp-stop	Escape	stop
pp-terminate	CTRL-BREAK	Abort Pi Presents

## Doppelte Ereignisse

Der Tkinter-Gerätetreiber hat eine Kinderkrankheit und akzeptiert Eingaben aller Geräte, die in /dev/input aufgeführt sind. So kann ein Druck auf der Fernbedienung ein Ereignis in pp\_inputdevicedriver.py und zugleich ein Ereignis in pp\_kbdriver.py, auslösen. Es kann also von Fall zu Fall erforderlich sein, bindings in .cfg-Dateien mit bind-printing=no zu unterbinden.

Doppelte Ereignisse können erkannt werden, wenn debugging in Pi Presents mit der Befehlszeilenoption -d aktiviert wird.

### 16.2.7 Erweitertes Verbinden mit Tastatur per pp\_kbdriver\_plus.py

Dieses Plugin verbessert das pp\_kbdriver.py I/O Plugin. Es gestattet Pi Presents:

- strings zusätzlich zu einzelnen Zeichen für Ereignissteuerung verwenden
- ganze Textzeilen an Track Plugins übergeben

Die Arbeitsweise des Plugins ist fast identisch mit der des serial port drivers ([Kapitel 16.2.4](#))

Details für die Anwendung der Konfiguration sind in der Datei :

```
/pipresents/pp_resources/pp_templates/keys_plus.cfg
```

Der pp\_kbdriver\_plus.py darf nicht neben pp\_kbdriver.py eingesetzt werden. kbdriver.py kann wie folgt abgeschaltet werden:

- Eine keys.cfg dem Profil hinzufügen mit enabled=no
- driver in /pipresents/pp\_io\_config/keys.cfg deaktivieren
- keys.cfg aus /pipresents/pp\_io\_config entfernen

Das Beispiel pp\_kbdisplay\_1p4 beschreibt den Gebrauch des Drivers, der Textzeilen als Ereignisse akzeptiert und per Track Plugin darstellt.

Eingehende Zeichen werden mit Zeichen und Strings in der Konfigurationsdatei abgeglichen. Wenn ein Abgleich zutrifft, wird ein Ereignis mit einem symbolischen Namen, wie er in der Konfigurationsdatei steht, ausgelöst.

- jedes empfangene Zeichen wird mit den Konfigurationsdaten abgeglichen, um dann eine Eingabe zu generieren
- wenn das Ende einer Zeichenkette erkannt wird, werden die Zeichen nach der vorhergehenden Zeichenfolge mit den Konfigurationsdaten abgeglichen um dann ein Input-Ereignis zu generieren.

Ereignisse können von beliebigen Zeichen, besonderen Zeichen, beliebigen Zeichenketten und besonderen Zeichenketten generiert werden.

### 16.2.8 Kurze Klänge mit pp\_aplaydriver.py produzieren

Der Linux-Befehl APLAY lässt .wav und andere unkomprimierte Audiodateien spielen. Mpg123, wenn es installiert ist, spielt .mp3 Audios. Mit dem aplay I/O plugin können Töne mit Animation Kommandos gespielt werden.

Das I/O plugin wird nicht vom Beeps Show Control Befehl eingesetzt ([siehe Kap. 8.5](#))

Näheres zum Umgang mit den Konfigurationsmöglichkeiten des Plugins in der Datei:

```
/pipresents/pp_resources/pp_templates/aplay.cfg
```

Der Animationsbefehl sieht so aus:

```
0 beep 1 beep
```

1. beep1 geht zusammen mit einem Namen in der Datei aplay.cfg

2. das Parameterfeld muss beep enthalten

Akzeptiert werden unkomprimierte Audiodateien, die mit dem Linux-Kommando APLAY gespielt werden können (.wav und andere), oder .mp3-Dateien, wenn mpg123 installiert ist.

```
Sudo apt-get install mpg123
```

### 16.2.9 Haptische Effekte per pp\_DRV2605driver.py

Der DRV2605 Haptic Controller steuert ERM oder LRA Aktuatoren, um haptische Effekte zu produzieren. Dieser Treiber generiert Vibrationssequenzen und steuert den Audio-zu-Vibration-Modus des Controllers. Die Effekte werden durch Animationsbefehle oder Show Control Befehle gestartet. (s. [Kapitel 8.6](#))

Einzelheiten zu den Plugin-Konfigurationsdaten in der Datei:

```
/pipresents/pp_resources/pp_templates/DRV2605.cfg
```

Der Animationsbefehl hat folgende Form:

```
0 DRV2605 sequence-name sequence1
0 DRV2605 audio-vibe-name vibe1
```

Die Konfigurations-Beispieldatei enthält den kompletten Satz verfügbarer Parameter.

Für den Einsatz des Controllers muss I2C aktiviert sein und der Pimoroni DRV2605 Treiber muss installiert sein mit:

```
python3 -m pip install drv2605
```

Pi Presents benutzt nicht den Pimoroni Treiber, sondern einen I2C-Treiber, der mit dem DRV2605-Treiber statt einer externen Version des Pimoroni-Treibers installiert ist. Der Pimoroni-Treiber ist in:

```
/pipresents/DRV2605_lib.py.
```

Pimoroni vertreibt den DRV2605 Controller so wie die ERM und LRA Aktuatoren. Pi Presents wurde nur mit dem ERM Aktuator, nicht mit dem LRA Aktuator getestet.

Der Originalcode der Pimoroni-Bibliothek befindet sich in:

<https://github.com/pimoroni/drv2605-python>

und das DRV2605-Datenblatt ist hier:

<https://www.ti.com/lit/ds/symlink/drv2605.pdf>

#### 16.2.9.1 Loop Modes

Das TI Datenblatt erklärt nicht eindeutig, ob Open Loop oder Closed Loop Feedback Modus einzurichten ist. Deshalb kann der Loop Mode in der .cfg-Datei geschaltet werden.

Das Datenblatt wird wie folgt interpretiert:

Einstellung	Loop Mode
ERM Sequenz mit TS2200 A-Bibliothek	Open Loop
ERM Sequenz mit anderen ERM-Bibliotheken	Closed Loop
LRA Sequenz (von Pi Presents intern benutzte Bibliothek)	Closed Loop
Audio zu Vibration für ERM oder LRA	Closed Loop

### 16.2.9.2 Audio zu Vibration mit dem Raspberry Pi

Die A/V-Ausgabe am Raspberry Pi erfolgt über einen 4-Pin\_Stecker. Tip and rings ist kein Standard.

Die Audio-Ausgabe am Raspberry Pi ist über einen 1.8K Widerstand an 0 Volt angeschlossen. Es ist deshalb laut Datenblatt erforderlich, einen 1uF Kondensator zu verwenden. Pi Presents setzt AC-Koppelung. Der Input in den DRV2605 beträgt 1,8 Volt.

Beim Experimentieren lag der A/V-Output beim Abspielen des Videos „Suits“ typischerweise bei 100mV p-p. Um den Beat der Filmmusik richtig zu spüren, musste die maximale Eingangsspannung in der .cfg Datei auf 0.025 gesetzt werden. Ein günstiger USB-Audio\_Dongle liefert vergleichbare Ausgangsleistung bei geringerem Rauschen.

### 16.2.10 Touch- und Klickbereiche einrichten

Die Datei screen.cfg definiert die Bereiche des Bildschirms, die maus- und/oder berührungssensitiv sein sollen. Klickbereiche sind nicht auf Hyperlinkshows beschränkt, sie können in allen Showtypen angewandt werden.

Die Datei besteht aus mehreren Sektionen mit jeweils einem einmaligen Namen. Der Name ist beliebig, darf in der Datei aber nur einmal vorkommen.

Alle Felder jeder Sektion müssen vorhanden sein. Diese Felder werden wie folgt benutzt:

- displays – Displays, auf denen der Klickbereich eingerichtet werden soll, z. B. HDMI0, HDMI1. Der Klickbereich wird angezeigt, wenn der symbolische Name in den Controls von Show oder Track vorhanden ist.
- name – Der symbolische Name des Klickbereiches. Jeder Befehl im Controls Feld des Tracks oder der Show hat einen symbolischen Namen. Wenn ein Track in der Show läuft, werden die Klickbereiche des Controls Feldes auf dem Bildschirm angezeigt, vorausgesetzt der Displayname steht im displays Feld. Wenn ein Bereich angeklickt wird, erkennt der symbolische Name ein Input-Ereignis. Dieses Input-Ereignis wird allen Shows/Tracks auf beiden Bildschirmen angeboten. Um einen Bildschirm auswählen zu können müsste man Klickbereiche mit unterschiedlichen symbolischen Namen einrichten.
- points – das ist ein Satz von xy-Paaren, die ein Vieleck beschreiben, das den Klickbereich ausmacht. Das Vieleck wird automatisch vervollständigt, deshalb hat ein Viereck 4 (und nicht 5) xy-Paare. Es müssen mindestens 3-Punkt-Paare vorhanden sein. Weitere Einzelheiten und andere Attribute siehe

<https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/tkinter-events-and-bindings.htm>

oder

<http://effbot.org/tkinterbook/canvas.htm>

- in section `create_polygon`. Für rechteckige Bereiche können die Points folgendermaßen angegeben werden:

$x1+y1+w*h$

where `x1`, `y1` sind die Koordinaten der oberen linken Klickbereichsecke und `w`, `h` bezeichnen Breite und Höhe – alles in Pixel.

- `fill-colour`, `outline-colour`

Bestimmt das Aussehen des Vielecks. Für Transparenz muss das Feld leer sein.

- `text`, `text-font`, `text-colour`

Wenn 'text' nicht leer ist, wird der eingetragene Text im Vieleck zentriert.

- `Bild`, `Bildbreite`, `Bildhöhe`

Wenn `image` nicht leer ist, wird ein Bild im Klickbereich angezeigt. Pfade relativ zu `pp_home` (+) und zu einem Profil @ sind möglich. Alternativ muss der komplette Pfad angegeben werden. Das Bild wird angepasst in `Bildhöhe` und `-breite` und wird im Vieleck zentriert.

### 16.2.11 Mit einem RFID-Tag-Reader über `pp_pn532driver.py` verbinden

RFID-Tag-Reader lesen Daten einer Karte, zum Beispiel einer Kreditkarte. Diese Karten haben einen kleinen Mikrochip, der vom Mikrochip-Tag\_Reader mit Strom versorgt wird und dann die Karte ausliest oder auch Daten auf und von der Karte schreibt. Jede Karte hat eine einmalige Identifikationsnummer (UID). Das ist die einzige Information die Pi Presents benötigt.

Es gibt eine ganze Reihe unterschiedlicher Cardreader mit unterschiedlichen Kartenstandards. Ich habe einen Reader genommen, der auf dem Chip PN532 basiert. Er liest vorzugsweise günstige Chips aus. Die Datenträger kommen in unterschiedlichen Formaten daher – kreditkartengroß, als Schlüsselanhänger und, vermutlich sehr brauchbar für Museen, als Papiersticker.

Das `pp_pn532driver.py` Plugin benötigt die `pynfc` - und `libnfc`-Bibliotheken um sich mit dem PN532 zu verbinden. Es wurde mit der PN532 I2C Schnittstelle getestet, sollte aber auch mit `serial/USB` und `SPI` funktionieren.

Das I/O-Plugin `pp_pn532.cfg` in `/pipresents/pp_resources/pp_templates` ist ein Beispiel für den Gebrauch des I/O-Plugins. In diesem Beispiel sind zwei codes definiert, die mit symbolischen Namen von Ereignissen verknüpft sind.

Um diese tag codes zu erhalten, müssen die `nfc-tools` wie nachfolgend beschrieben installiert werden. Danach erhält man mit dem Befehl `nfc poll` eine folgende Rückmeldung wie z.B.

ISO/IEC 14443A (106 kbps) target:  
ATQA (SENS\_RES): 00 04  
UID (NFCID1): 76 35 a2 ac  
SAK (SEL\_RES): 08

Der tag code ist von der UID abgeleitet, die vermutlich eine 8-digit oder 14-digit Hexadezimalzahl ohne Leerzeichen ist.

Details zum Gebrauch der Konfigurationsdatei finden sich in der Datei:

`/pipresents/pp_resources/pp_templates/pn532.cfg`

Wenn ein Datenträger dem Reader gezeigt wird, liest er ihn aus und verknüpft ihn mit den verschiedenen Einträgen in der .cfg-Datei. Wenn ein Treffer erfolgt ist, wird ein Ereignis mit dem zugehörigen symbolischen Namen generiert. Ein Ereignis kann auch generiert werden, wenn der Datenträger entfernt wird. Die Empfindlichkeit kann über die Schwellwert-Parameter geregelt werden. Wenn eine Karte sich auf den Reader zubewegt, können diverse Ereignisse ausgelöst werden.

Das Beispiel `pp_rfidplugin_1p4` zeigt eine Radiobuttonshow, in der zwei Tags zwei Tracks steuern-

### **RFID-Tag-Reader installieren**

Ich habe diesen Bausatz gekauft:

XCSOURCE® NXP PN532 NFC RFID Module V3

Es funktioniert gut, nur die Annäherungsdistanz beschränkt sich auf wenige Zentimeter. Ich habe den I2C-Anschluss benutzt aber es sollte auch mit SPI oder UART funktionieren.

Die meisten Installationsanleitungen im Internet scheinen veraltet. Diese Instruktionen sind besser:

<https://blog.stigok.com/2017/10/12/setting-up-a-pn532-nfc-module-on-a-raspberry-pi-using-i2c.html>

Mit dem obigen Link werden `libnfc` und `nfc-tools` installiert. Sie erlauben die Erkennung der Tag-ID auf der Kommandozeile. Zusätzlich wird die `pynfc` Bibliothek benötigt. Eine Kopie im `pipresents/pynfc`-Verzeichnis, dass für eine Fehlerkorrektur geändert wurde. `Pynfc` muss nicht installiert werden, weil es bereits in Pi Presents enthalten ist. Für Interessierte: Das Original befindet sich in

<https://github.com/ikelos/pynfc>

## **16.3 I/O-Plugins schreiben**

Ein I/O-Plugin ist ein Python3-Modul das:

- für Eingaben physikalische Inputs auf dem Raspberry Pi in Ereignisse mit symbolischen Namen übersetzt
- für Ausgaben Animationsbefehle mit symbolischen Namen und Parametern in physikalische

Ausgaben übersetzt.

I/O Plugins sollten im Verzeichnis /pipresents/pp\_io\_plugins platziert werden, wo sie für alle Profile erreichbar sind.

### 16.3.1 Einrichtung und Registrierung eines I/O-Plugins

I/O-Plugins werden registriert, wenn eine I/O-Plugin-Konfigurationsdatei im pp\_io\_config Verzeichnis eines Profils vorliegt. Die Registrierung erfolgt über den pp\_iopluginmanager.py. Dieses Modul liest den Bereich [driver] in der Konfigurationsdatei aus und importiert bei enabled =yes das Modul, das in module= aufgeführt ist.

Die I/O-Konfigurationsdatei kann auch benutzt werden, um ein I/O-Pluginmodul zu erstellen. So kann ein einziges Modul von unterschiedlichen Geräten genutzt werden, z.B. unterschiedlichen Fernbedienungen. Es können weitere Konfigurationsdaten im Bereich [driver] eingesetzt, oder bei Bedarf, weitere Bereiche geschaffen werden.

Der Pfad zur Plugin-Konfigurationsdatei ist dem I/O-Modul bekannt, so dass die zusätzlichen Parameter ausgelesen werden können.

### 16.3.2. Class

Der Name der Class muss mit dem Dateinamen übereinstimmen.

Die I/O-Plugin Output-Methode könnte von mehreren Objekten abgerufen werden. Deshalb sollten die Variablen auf der Outputseite Class-Variable sein.

### 16.3.3 Methoden

#### init

```
def init (self,filename,filepath,widget pp_dir, pp_home, pp_profile, button_callback=None)
```

init () wird einmalig vom I/O-Pluginmanager beim Start von Pi Presents aufgerufen. Die Methode muss:

- Konfigurationsdaten aus dem Bereich [driver] in der I/O-Konfigurationsdatei des Profiles auslesen
- das physikalische Gerät initialisieren
- mit einer statischen Variablen anzeigen, dass die Initialisierung erfolgreich war und das Plugin aktiv ist
- bei erfolgreicher Aktivierung eine Meldung „normal“ für die Logdatei ausgeben
- bei nicht erfolgreicher Aktivierung eine Meldung „error“ für die Fehler-Meldung und die

Logdatei ausgeben.

### Arguments:

filepath – der Pfad zur I/O-Plugin-Konfigurationsdatei

filename – Teil des Pfades für Logdatei

widget – die Instanz eines Objektes, der man die Tkinter after\_and\_after\_cancel Aufrufe zuordnet

pp\_dir – Pfad zu pipresents

pp\_home – Pfad zu pp\_home

pp\_profile – Pfad zum aktuellen Profil

callback – die zu wählende Fernsteuerungsfunktion, um ein Ereignis durch einen physikalischen Input zu erzeugen. Callback hat zwei Parameter:

- symbolischer Name des Ereignisses
- eine Textzeile, die den Anstoß für das Ereignis benennt. Das Plugin liest das im Bereich [driver]

### start

```
def start(self)
```

start() wird einmalig vom I/O-Plugin beimStart von Pi Presents aufgerufen. Die Methode:

- Wenn möglich starte eine Unterbrechungsaktion
- Wenn die Eingabe erfolgt ist durchsuche die Inputs noch einmal und rufe widget.after() um einen weiteren Scan zu planen.

Eine Suche nach physikalischen Inputs blockiert nicht, weil Pi Presents die parallele Suche beherrscht

Ein Interrupt oder ein bei der Ausführung erkannter physikalischer Input löst die callback-Funktion aus:

```
self.button_callback(symbolic_name,source).
```

### get input

```
get_input (self, key, driver_ref=“)
```

get-input wird von track plugins genutzt um Werte für Inputs zu bekommen. Der Schlüssel ist ein String mit dem die angeforderte Eingabe identifiziert werden kann.

Beim Einsatz von get\_input hat sich herausgestellt, dass das key Feld allein kein besonders praktischer Weg ist, um einen bestimmten Input zu erreichen, weil Key den gleichen Namen in mehreren I/O-Treiber tragen könnte. Das optionale driver\_ref Feld erlaubt die Auswahl eines

bestimmten Treibers für `get_input` Operationen. Das beschränkt den Key auf diesen Treiber.

Um diese Operation nutzen zu können muss die I/O-Konfigurationsdatei ein Feld `driver_ref` mit einem Referenznamen haben, wenn nicht alle I/O Plugins den Key abfordern sollen.

Beispiel: Wenn der Treibername in einer I/O-Konfigurationsdatei `caldav` heißt, dann fragt `get_input` (`((<start value>, <end value>), ,caldav')` nur dieses I/O nach dem Key.

Die Methode gibt zwei Argumente zurück:

- Found – True/False
- Value – der Wert

### **terminate**

```
def terminate(self)
```

`terminate` wird vom Hauptprogramm aufgerufen, wenn Pi Presents aus irgend einem Grund schließt. Es werden alle interrupts beendet und alle Eingabezeiten gelöscht

### **is active**

ermöglicht Pi Presents zu bestimmen, ob das Plugin aktiv sein soll

```
def is_active(self)
```

meldet ob das Plugin aktiv ist (True/False)

### **handle output event**

Diese Methode verwendet Animationsbefehle in Formaten

```
name param_type param_value_1 „param_value2“
```

um physikalische Outputs zu erstellen

```
def handle_output_event (self, name, param_type, param_values, reg_time)
```

name – der symbolische Name im Animationsbefehl

param\_type – der Parametertyp im Animationsbefehl

param\_values – einer Liste der Parameter-Values im Animationsbefehl. Es können 0 oder mehr Werte sein. Wenn ein Parameter Leerstellen enthält muss er in „“ Anführungszeichen stehen.

reg\_time – der Zeitpunkt, an dem der Befehl ausgeführt werden soll. Wird nur für das Log benötigt, denn den Output zur rechten Zeit bewerkstelligt der Animation Manager.

Jeder Animationsbefehl ist zu diesem und allen anderen I/O-Plugins übermittelt. Wenn Name und Parametertyp zu den Inhalten des Plugin passen, wird der Befehl mit den gewählten Parametern ausgeführt und ‚normal‘ wieder gegeben.

Die Parameterwerte müssen auf Korrektheit geprüft werden (z.B. state is on or off). Wenn nicht erlaubte Parameterwerte vorkommen, kommt eine Fehlermeldung zustande.

Diese Funktion gibt zwei parameter aus – status, messages

status - ‚normal‘/‘error‘  
message – eine Textzeile für die Fehlermeldung und die Logdatei.

### 16.3.4 Beispiel

Komponenten:

- pp\_exampeldriver.py im Verzeichnis pp\_io\_plugin
- exampeldriver.cfg im Verzeichnis /resources/templates
- pp\_ioplugin\_1p4 Beispiel im Pi Presents examples github repository.

Das Beispiel-Plugin enthält alle Elemente eines Plugin, bewirkt jedoch keinen physikalischen In-/Output.

- Die Eingabe erstellt ein Ereignis mit dem symbolischen Namen tick in Intervallen, die in der Konfigurationsdatei exampeldriver.cfg festgelegt sind.
- Die Ausgabe zeigt die Meldung aus dem Animationsbefehl im Terminalfenster.

## 17 Fernbedienung mit OSC

Seit Version 1.3.1 kann Pi Presents andere Instanzen von Pi Presents auf anderen Computern steuern oder von anderen Instanzen gesteuert werden, sofern die Computer das Open Sound Control (OSC) Protokoll beherrschen. Das gilt auch für Smartphones und Tablets.

OSC ist ein schlankes und flexibles Protokoll, das ursprünglich für die Steuerung von Musikgeräten gedacht war. Es kann aber viel mehr und wird von vielen Multimediageräten genutzt. Für Pi Presents ist ein Master eine Einheit, die Kommandos sendet um Slaves zu kontrollieren.

- Eine Master-Einheit (OSC Client) sendet Befehle an Slaves
- Eine Slave-Einheit (OSC Server) erhält Befehle von einer Master-Einheit und handelt danach

Je nach Konfiguration kann Pi Presents als Master oder Slave agieren.

Die Einbindung in diese Version (1.3.1c) ist stark verbessert gegenüber der ersten experimentellen Version. Die Verbesserungen sind:

- Viele Slaves pro Master
- Eine bestimmte Einheit kann beides sein, Master und Slave
- Ein Slave kann von vielen Masters kontrolliert werden, Antworten gehen an den Master, der die Kommandos gesendet hat.
- Benutzt wird weiterhin das UDP-Protokoll und nicht TCP. Das macht es schnell und

kompatibel mit mehreren Fremdsystemen. Allerdings ist UDP, was man ein „unzuverlässiges“ Protokoll nennt – Sendungen kommen nicht sicher an und häufig auch in verdrehter Reihenfolge. Ob das für kleine Systeme in einem lokalen Netz eine Rolle spielt, bleibt abzuwarten.

- Befehle benutzen jetzt die Pi Presents Show-Kommando-Syntax. Sie werden intern in für OSC taugliche Sendungen konvertiert.
- Die Konfiguration wurde viel einfacher

Die Implementation unterstützt nicht:

- eine Muster erkennende Sprache, um diverse Empfänger für eine Sendung anzusprechen.
- Hoch aufgelöste Zeitangabe
- "Bündel" von Sendungen die gleichzeitig auftreten sollen

## 17.1 Senden und empfangen von Meldungen über OSC

In den Pi Presents Konventionen werden OSC-Befehle von Mastergeräten an Slavegeräte gesandt. Um einen OSC-Befehl zu senden wird das zugehörige Kommando in das Show Control Feld eines Tracks oder einer Show eingetragen. Ansonsten geschieht der Sendungsempfang automatisch.

Anders, als bei Sendungen (loopback oder Server-Info-Befehle), haben alle OSC-Befehle den gleichen Effekt, als würden sie auf der lokalen Maschine angewandt werden.

Ein OSC-Show-Control-Befehl hat drei oder mehr Felder:

- `osc` – alle OSC-Befehle beginnen mit `osc` oder `OSC` im ersten Feld
- `unit` – der OSC-Name des Gerätes, zu dem der Befehl gesandt wird. Dieser Name sollte auch im Feld „OSC Names of slaves“ in der Konfigurationsdatei stehen.
- `command` – `open`, `openexclusive`, `close`, `closeall`, `event`, `monitor`, `animate`, `exitpipresents`, `shutdownnow`, `reboot`, `send`, `loopback`, `server-info`
- `parameters` – zero oder mehr Parameterwerte

z.B. `osc unit1 open myshow`

## 17.2. OSC-Grundlagen

OSC-Befehle werden von Pi Presents in Meldungen übersetzt. Die OSC-Software erledigt eine weitere Konversion in ein wirkungsvolles Binärformat für die Übertragung. Die Meldungen haben ein OSC-Adressfeld und 0 oder mehrere Datenfelder, z.B.

```
/pipresents/unit1/core/open myshow
```

Die OSC-Adresse ist hierarchisch, z.B.:

```
/pipresents/unit1/core/open
```

Das bedeutet:

- /pipresents – die Meldung ist für Geräte bestimmt, die Pi Presents typische OSC-Meldungen verstehen. Alle anderen Geräte ignorieren diese Meldung.
- /unit1 – Meldung ist für Pi Presents unit1 und wird von allen anderen Geräten ignoriert. Unit1 muss in der Liste der slave-Geräte in der Konfigurationsdatei aufgeführt sein
- /core – die Meldung ist für den Kern von Pi Presents, nämlich dem Teil, der Shows und Tracks per Eingabeereignisse steuert.
- /open – ist der Befehl zum Öffnen einer Show. Die komplette Meldung enthält die Showreferenz als Argument.

Pi Presents Show Command	OSC Command	Use
osc unit1 open myshow	/pipresents/unit1/core/open myshow	Öffnet die referenzierte Show
osc unit1 close myshow	/ipresents/unit1/core/close myshow	Schließt die referenzierte Show
osc unit1 openexclusive myshow	/pipresents/unit1/core/openexclusive myshow	Schließt alle laufenden Shows und öffnet dann die referenzierte Show
osc unit1 closeall	/pipresents/unit1/core/closeall	Schließt alle laufenden Shows
osc unit1 event pp-stop	/pipresents/unit1/core/event pp-stop	Generiert ein Eingabeereignis mit dem referenzierten symbolischen Namen
osc unit1 exitpipresents	/pipresents/unit1/core/exitpipresents	Beendet Pi Presents
osc unit1 shutdownnow	/pipresents/unit1/core/shutdownnow	Fährt RPI herunter
osc unit1 reboot	/pipresents/unit1/core/reboot	Startet RPI neu
osc unit1 animate out1 state on	/pipresents/unit1/core/animate out1 state on	Stellt ein output pass-through system bereit. Argumente sind Angaben der Animationsbefehle ohne den führenden Verzögerungsparameter
osc unit1 monitor on	/pipresents/unit1/core/monitor on	Startet den Monitor
osc unit1 monitor off	/pipresents/unit1/core/monitor off	Schließt den Monitor
osc anyunit send /myprotocol/anything/you/like arg1 2	/myprotocol/anything/you/like arg1 2	Sendet die Meldung zu allen möglichen Geräten. Wird benötigt, um Befehle an Geräte zu senden, die nicht Pi Presents-Geräte sind. Alle Argumente werden als Zeichenkette gesendet
osc unit1 loopback	/pipresents/unit1/system/loopback	Veranlasst den Slave eine leere Rückmeldung an den Master zu senden. Diese Antwort erscheint im Terminalfenster
osc unit1 server-info	/pipresents/unit1/system/server-info	Veranlasst den Slave, Informationen über sich in einer

		server-info-reply message zu senden Diese Antwort erscheint im Terminalfenster
--	--	--

### 17.3 OSC Konfiguration

OSC wird durch das Vorhandensein einer osc.cfg Datei im Verzeichnis /pp\_io\_config innerhalb eines Profiles aktiviert und konfiguriert. Die Datei ist eine nicht standardgemäße I/O-Plugin-Konfigurationsdatei. pp\_web\_editor.py hat eine Menüoption, mit der osc.cfg Dateien im Profil erstellt und editiert werden können.

#### Dieses Gerät

Field	Example	Aktion
OSC Name of this Unit	my-pi	Der /unit Teil der Adresse dieses Gerätes. Richtet Meldung gezielt an dieses Gerät.
IP of this Unit	192.168.1.101	Normalerweise bleibt dieses Feld leer und Pi Presents erhält diesen Wert vom Netzwerk. Sollte das nicht der Fall sein, kann die IP hier benannt werden  Wenn der Pi in zwei Netzwerken arbeitet, kann es erforderlich sein, ein bevorzugtes Netzwerk in der web.cfg einzutragen (Kap. 20.1).  Die Ip für das „Lauschen“ auf Sendungen des Masters, wenn der Slave aktiviert ist und Antworten des Slave, wenn der Master aktiviert ist
<b>Fields for a Master Unit</b>		
Field	Example	Use
Master Enabled	yes	Yes/no – dieses Gerät als Master aktivieren oder deaktivieren
Listening Port for replis from slaves	9001	Port für das „Lauschen“ auf Antwort von Slave-Geräten  Slaves antworten auf dem gleichenPort, auf dem sie auch „Lauschen“, deshalb muss der „Lauschport“ des Masters der gleiche sein, wie die „Lauschports“ der Slaves
OSC Names of slaves	Slave1 slave2	Eine Liste von OSC-Namen (durch Leertasten gerennt), die als Merkhilfe für die Festlegung der IP der Slave-Geräte in den OSC-Befehlen dient.
IP of Slaves	Myhostname	Eine Liste von IP (durch Leertatsten getrennt) der

	192.168.1.102	Slave-Geräte in der gleichen Reihenfolge, wie die Slave-Namen
--	---------------	---

### Fields for a Slave Unit

Field	Example	Use
Slave Enabled	yes	Aktivieren/deaktivieren dieses Gerätes als Slave
Listening Port for commands from a Master	9001	Port, auf dem Slave auf Befehle des Master „lauscht“. Alle Slaves müssen auf dem gleichen Port auf Befehl „lauschen“, der gleich sein muss mit dem „Lauschport“ des Masters

## 17.4 oscremote und oscmonitor.py

pp\_oscremote.py und pp\_oscmonitor.py sind zwei eigenständige Programme, die ich entwickelt habe, um die OSC-Schnittstelle von Pi Presents zu testen. Beide laufen unter Windows (benötigt ein installiertes Python), Linux oder Raspbian. Sie enthalten Code, der als Basis für andere Anwendungen genutzt werden kann. Ist Irgend jemand daran interessiert, eine App für Iphone (IOS) oder Android für die Steuerung von Pi Presents zu schreiben?

oscremote sendet Befehle an Pi Presents, während oscmonitor OSC-Meldungen von Pi Presents anzeigt. Standardmäßig sind beide so eingestellt, dass sie mit dem Gerät /pipresents mit Port 9001 senden und oscmonitor auf Port 9002 Sendungen des Masters annehmen können. Beim ersten Start müssen weitere Konfigurationsfelder ergänzt werden.

Wenn beide Programmen auf unterschiedlichen Raspberry Pis laufen, kann Pi Presents als Monitor und Remote agieren (oder je ein PI). Drei Beispiele dienen als Demonstration von OSC. Sie können wie folgt aus dem Stand laufen:

- pp\_osc\_1p4 ist ein einfaches Profil, das die Hauptfeatures zeigt. Die zugehörige osc.cfg setzt Pi Presents als Master und Slave. Die Show Mediasow läuft in der Startshow und sendet eine ganze Reihe verschiedener Befehle aus den Show-Control-Feldern des Tracks . Sie können von oscmonitor angezeigt werden. Antworten zum Loopback und Server-Info-Befehle können unter Pi Present in seinem Terminalfenster angesehen werden. Das Profil benutzt benutzt oscremote zum Starten und Beenden der Show Mediashow-
- pp\_multiwindow\_1p4 verfügt über eine cfg-Datei, die es erlaubt, das Profil per oscremote.py zu steuern; es müssen alle Shows aus der Startshow entfernt werden. Danach können sie von oscremote.py mit open show und close show Befehlen gesteuert werden.
- pp\_showcontrol.py verfügt über eine osc.cfg-Datei, die es erlaubt, Ausgabebefehle aus dem Show-Control\_Feld der beiden Image tracks in die Steuerung der Audioshow zu spiegeln. Oscmonitor.py zeigt das an.

Mit einer Konfigurationsänderung können oscremote und oscmonitor paarweise laufen. Beide können zudem auf einer Maschine laufen, während Pi Presents auf einer anderen läuft.

### pp\_oscremote.py

Vor dem Gebrauch von `oscremote.py` muss im `option>edit`-Menü die Kommunikation eingerichtet werden. Standardmäßig eingerichtet ist die Kommunikation mit dem OSC-Gerät Pi Presents unter dem Hostnamen `raspberrypi` mit dem Port 9001.

Jeder Button generiert teilweise oder insgesamt eine OSC-Meldung im Feld `Message to Send`; es müssen `Show Referenzen` und `Ausgabebefehle` eingefügt werden, dann auf `Send` drücken. Gesendete Meldung und etwaige Antworten erscheinen im Feld `status`.

Die Menüoption `profile>select` ermöglicht die Auswahl eines Pi Presents Profils. Dafür muss ein Verzeichnis `/pp_home/pp_profiles` mit dem Profil eingerichtet sein und es benötigt die Befehlszeilenoption `-o` für `pp_oscremote.py` um den Pfad nach `pp_home` einzustellen (Standard ist `/home/pi`)

Die Shows dieses Profils sind im `Shows tab` aufgeführt: Eines davon auszuwählen (vor dem `open` oder `close` Button) hilft, sich die `Show Referenz` zu merken oder zu notieren, mehr nicht

### **pp\_oscmonitor.py**

Zeigt lediglich Meldungen an, die im Statusfenster aufgelaufen sind. Standardmäßig ist der Empfang von Meldungen auf Port 9002 eingestellt.

## **18 Track Plugins**

Track Plugins sind Module in Python3 Code mit einer offen gelegten Schnittstellenbeschreibung, die vom Benutzer zur Erweiterung der Darstellungseigenschaften von Pi Presents programmiert werden können. Track Plugins ermöglichen die dynamische Anzeige von Informationen wie Wetter oder Tickermeldungen. Es ist besonders für die Übernahme aus Webseiten oder von RSS-Feeds gedacht.

Track Plugins werden von Playern ausgeführt. Ihr wesentlicher Zweck ist die Darstellung dynamischer Inhalte. Das kann man auf zwei Wegen erreichen:

- durch Änderung oder Ersetzen des zu spielenden Mediums
- durch die Ausgabe dynamischer Informationen direkt auf dem Pi Presents Bildschirm

Track Plugins werden im Verzeichnis `/pipresents/pp_track_plugins` in `.py` Dateien gespeichert. Es empfiehlt sich, die Namen der Plugins mit vorangestellten eigenen Initialen zu versehen, um Konflikte mit Python-Modulen zu vermeiden (z.B. `time.py` würde mit dem Standard-Zeit-Modul kollidieren).

### **Plugins anwenden**

Wenn ein Plugin für einen Track benutzt wird, muss das 'Plugin Configuration File' Feld eines Players den Namen einer Plugin-Konfigurationsdatei enthalten, z.B.:

```
+/media/krt_image_text.cfg.
```

Relative Pfade sind erlaubt. Die vom Plugin bereit gestellten Medien müssen zum gespielten Tracktyp passen.

Die Plugin-Konfigurationsdatei kann das gleiche Plugin mit unterschiedlichen Parametern aufrufen. Sie muss mindestens enthalten:

```
[plugin]
plugin = pluginname
type = image      # erforderlich, wenn das Plugin in einer Liveshow aufgerufen wird
```

pluginname ist der Name des Python-Moduls, das das Plugin enthält (der Dateiname ohne .py).

Der Name des Plugins kann relativ angegeben werden, wenn dem Namen vorangestellt ist:

- @ - das Plugin befindet sich im Profil im Verzeichnis pp\_track\_plugins
- + - das Plugin befindet sich in pp\_home im Verzeichnis pp\_track\_plugins
- no prefix – das Plugin befindet sich in /pipresents/pp\_track\_plugins (ursprünglicher Ort)

Track Plugins im Verzeichnis /pipresents sind nicht mehr angebracht. Vorzugsweise sollten sie im Profil mit dem @ als Präfix gespeichert werden.

Die Konfigurationsdatei kann weitere Parameter angeben, die der Plugin code aus dem Verzeichnis plugin\_paramsgin\_params übernehmen kann, z.B:

```
[plugin]
plugin = krt_image_text
type = image      # erforderlich, wenn das Plugin in einer Liveshow aufgerufen wird
# optional
text = text to display
```

Der Plugincode kann den Wert von text auslesen. Mit mehreren Konfigurationsdateien, die alle das gleiche krt\_image\_text Plugin aufrufen, kann man mit jedem Track einen anderen Text anzeigen.

Man kann Plugins in Liveshows nutzen. Dafür müssen Plugins besonders programmiert sein. Um sie in einer Liveshow zu gebrauchen, müssen die Konfigurationsdateien lediglich in das Verzeichnis pp\_live\_tracks kopiert werden. Plugins in der Liveshow müssen in ihrer Konfiguration den Tracktyp angeben, damit Pi Presents weiß, welcher Tracktyp gezeigt werden soll.

## Plugins erstellen

In der Version 1.3.1 wurde die Trackplugin-Schnittstelle geändert und verbessert. Die Änderungen waren nötig geworden, weil der Aufruf von Inhalten durch Plugins das Vor-Laden von Medien für eine unterbrechungsfreie Vorführung tangiert. Schnittstellendetails finden sich in

pp\_example\_plugin.py .

Drei Beispiele für plugins:

- pp\_example\_plugin.py  
Darin befindet sich die Schnittstellenbeschreibung. Das Beispiel ist umfangreich, weil es alle Tracktypen benennt, die mit einem Plugin umgehen können, ebenso Liveshows.
- krt\_image\_text.py  
Ein Beispiel, in dem Text einem Image beigefügt wird.

- `krt_time.py`  
Zeigt die aktuelle Zeit auf dem Bildschirm.

Die Beispiele benötigen:

- Python Imaging Library (PIL).  
Das Handbuch ist hier:  
<https://web.archive.org/web/20200315112731/http://effbot.org/imagingbook/pil-index.htm>.  
oder  
<http://effbot.org/imagingbook/pil-index.htm>.
- Tkinter canvas Operationen sind dargestellt:  
<https://web.archive.org/web/20200315112731/http://effbot.org/tkinterbook/canvas.htm>  
oder  
<http://effbot.org/tkinterbook/canvas.htm>

Die Funktionsschnittstelle verändert sich relativ wenig, das hängt vom Tracktyp ab:

- `image` – das Plugin erwartet den Pfad zu einer Datei, die ein Bild enthält (z.B. `picture.jpg`). Eine Bilddatei muss wiederholt werden und für die Anzeige mit PIL und Tkinter geeignet sein.
- `audio` – das Plugin wird mit einer Audiodatei „gefüttert“. Der Audiotrack kann für das Abspielen im MPlayer ausgegeben werden. Leer ist erlaubt, dann wird kein Audio gespielt.
- `MPV Video` – das Plugin kommt mit einer Videodatei. Der Videotrack muss wiederholt werden, um im OMXPlayer abgespielt werden zu können. Das Plugin kann auch direkt in Canvas schreiben, aber achten Sie darauf, dass das Video über dem oberen Rand des Textes erscheint. Richten Sie besser die Position mit einem Videofenster ein.
- `web` – das Plugin kommt mit einer Datei, die HTML-Code enthält (was immer auch `uzbl` rendern kann). Eine HTML-Datei kann für das Abspielen in `uzbl` ausgegeben werden. Das Plugin kann auch direkt in Canvas schreiben, aber achten Sie darauf, dass der Browser über dem oberen Textrand erscheint. Richten Sie seine Position mit `Web Window` ein.
- `message` – das Plugin wird mit dem ursprünglichen Text versorgt, der geändert und zurück gegeben wird. Das Plugin kann auch direkt in Canvas schreiben.

## 19 Remote Management

Die Programme `pp_manager` und `pp_web_editor` dienen dem Remote Management von PiPresents per Webbrowser, das auf jedem Computer innerhalb des Netzwerkes läuft, in dem sich Pi Presents befindet. Beide Programme starten einen Webserver, der wiederum eine interaktive Webseite anbietet, die, bezogen auf den Editor, ziemlich ähnlich aussieht, wie die normale grafische Oberfläche des Pi Presents Editor.

Diese Programme basieren auf Remi. Remi muss man sich vorstellen als ein Framework für eine Standard Desktop-Oberfläche. Die Bibliotheken enthalten keine explizite Sicherheitsstrategien. Deshalb ist es nicht empfehlenswert, damit in unsichere Netzwerke zu gehen.

Wenn Inhalte von fremden Quellen heruntergeladen werden, sollte das System vor möglichen Javascript-Angriffen geschützt werden, bevor die Inhalte dargestellt werden.

- Manager – pp\_manager.py kann Pi Presents Profile auswählen und starten. Zudem kann der Manager Profile, Medien und Livetracks hoch- und herunterladen.

Der Manager kann außerdem den normalen Autostartmechanismus ([siehe Kap. 6.6](#)) dadurch ersetzen, dass er Profile per Remote für einen Autostart auswählt.

Die Arbeitsweise des Managers zielt auf das Hochladen von Medien und Livetracks von einem entfernten Computer auf den Pi, oder auf deren Import von einem USB-Stick. Einmal auf dem Pi, können Medien und Livetracks gelöscht oder umbenannt werden. Profile sind ähnlich hochzuladen oder zu importieren; darüber hinaus können sie erstellt und bearbeitet werden um sie dann an den Remotecomputer senden zu können.

- Web Editor – pp\_web\_editor.py ist eine Verwandter des Pi Presents Profil Editors.

## 19.1 Fernsteuerung (Remote) einrichten

Vor der Benutzung der beiden Remoteprogramme muss

- die Datei /pipresents/pp\_config/pp\_web.cfg mit einem Texteditor bearbeitet werden. Einträge im Abschnitt [manager-editable] können anschließend mit dem Manager bearbeitet werden.

Field	Example	Description
<b>[manager-editabel]</b>		
media_offset	/media	Der Ort im pp_home Verzeichnis, zu dem Medien importiert oder hochgeladen werden. So meint /media, dass Medien im Verzeichnis /home/pi/pp_home/media gespeichert werden
livetracks_offset	/pp_live_tracks	Der Ort im pp_home Verzeichnis, zu dem livetracks importiert oder hochgeladen werden. So meint /pp_live_tracks, dass Medien im Verzeichnis /home/pi/pp_home/pp_live_tracks, dem Standardverzeichnis für Livetracks, gespeichert werden
profiles_offset		Der Ort im pp_profiles Verzeichnis, zu dem Profile importiert oder hochgeladen werden. Bleibt in der Regel leer. Dieser Ort bezeichnet auch das Verzeichnis für die Profil-Auswahlliste des Managers. z.B. ein Speichern von /test_profile zielt auf /home/pi/pp_home/pp_profiles/test_profiles.
options		Wenn Pi Presents mit dem Manager gestartet wird, werden die benannten Optionen eingefügt ( <a href="#">siehe Kap. 6.1</a> ) Die Optionen -p und -o müssen nicht benannt werden, weil der Manager sie automatisch einfügt

autostart_path	/my_profile	Wenn nicht leer, startet der Manager Pi Presents im Autostart sofort bevor sein Webserver startet. Das Feld bezeichnet das für den Autostart gewünschte Profil. /my_profile startet dieses Profil in /home/pi/pp_home/pp_profiles/my_profile
autostart_options	-fb	Wenn Pi Presents per Autostart gestartet wird, werden benannte Optionen hinzugefügt ( <a href="#">siehe Kap. 6.1</a> ) Die Optionen -p und -o müssen nicht benannt werden, weil der Manager sie automatisch einfügt
<b>[manager]</b>		
home	/home/pi/pp_home	Pfad des Pi Presents Home-Verzeichnisses. Er muss in pp_home enden
import_top	/media/pi	Dieses Feld beschränkt die Dateien für folgende Transaktionen auf die Dateien in diesem Verzeichnis und darunter: <ul style="list-style-type: none"> <li>• Quelle für Import von Medien- und Livetrack-Dateien</li> <li>• Quelle für Import von Profilen</li> </ul> Aus Sicherheitsgründen beschränkt diese Option die sichtbaren Dateien auf ihre Position unterhalb von Top. Sie bestimmt zudem den Startpunkt für den Dialog zur Dateiauswahl für den Import
port	8081	Vom Editor benutzter Port. Sollte nicht geändert werden
username		Wenn Benutzername und Passwort gesetzt sind, ist für den Manager ein Login erforderlich
password		
<b>[editor]</b>		
port	8082	Der vom Editor genutzte Port. Sollte nicht geändert werden
username		Wenn Benutzername und Passwort gesetzt sind, ist für den Web Editor ein Login erforderlich
password		
<b>[network]</b>		
unit	My Pi	Diese Felder werden auch vom eMail-System genutzt Der Name erscheint im Manager-Fenster und in eMails. Nützlich, wenn mehrere Pi's im Manager verwaltet werden
force_ip		IP_Adresse des Pi, auf dem Editor oder Manager laufen. Normalerweise kann dieses Feld leer bleiben, weil die IP-Adresse des Pi automatisch festgelegt wird. Hat das Feld einen Eintrag, wird die automatische Erkennung abgeschaltet
preferred_interface		Wenn WLAN und Ethernet-LAN gleichzeitig angeschlossen sind, erhält der Pi zwei Ip-Adressen. Wenn beide Geräte aktiv sind, kann eines davon ausgewählt werden. Die Werte lauten in der Regel wlan0 oder eth0

## 19.2 Mit dem Manager arbeiten

Der Manager wird mit dem Befehl

```
python3 pp_manager.py
```

in einem Terminalfenster im pipresents-Verzeichnis gestartet. Damit startet der Managerserver. Der Manager wird per Browser angesteuert, indem die IP-Adresse und der Port eingegeben werden, z:B.

```
192.168.1.108:8081
```

Der Manager zeigt eine Liste der Profile im Profilverzeichnis.

### Buttons

- Start button – startet das ausgewählte Profil aus der Liste
- Exit button – beendet Pi Presents
- Refresh List button – Erneuert die Liste der Profile: erforderlich, wenn ein Profil vom Manager oder dem Pi gelöscht oder hinzugefügt wurde

Die Zeile über den Buttons zeigt den Programmteil und den Status von Pi Presents an. Wenn Pi Presents endet, wird der Exit-Code in Klammern angezeigt.

- 100 - Normal Exit
- 101 – Pi Presents durch Ereignis von außen geschlossen (in der Regel durch den Pi)
- 102 - Pi Presents wurde durch Laufzeitfehler geschlossen. Die Fehlermeldung kann mit dem Log heruntergeladen werden.

### Menus

- media>import- kopiert Mediendateien von einem gewählten Verzeichnis nach pp\_home in den Unterordner mit dem voreingestellten Media-Pfad (media\_offset)
- media>upload – Upload von Mediadateien per Browser nach pp\_home in den Unterordner mit dem voreingestellten Media-Pfad (media\_offset)
- media>manage – Mediadateien umbenennen oder löschen
- livetacks>import - kopiert Livetracks von einem gewählten Verzeichnis nach pp\_home in den Unterordner mit dem voreingestellten Livetracks-Pfad (livetacks\_offset)
- livetacks>Upload - Upload von Livetracks per Browser nach pp\_home in den Unterordner mit dem voreingestellten Livetracks-Pfad (livetacks\_offset)
- livetacks>manage – Livetracksdateien löschen oder umbenennen
- profile>import - kopiert Profilverzeichnis von einem gewählten Verzeichnis nach pp\_home/pp\_profiles in den Unterordner mit dem voreingestellten Profil-Pfad (profiles\_offset)
- profile>upload - Upload vom Profilverzeichnis per Browser nach pp\_home/pp\_profiles in den Unterordner mit dem voreingestellten Profilpfad (profiles\_offset). Vor dem Upload muss

das Profilverzeichnis in eine Zip-Archivdatei umgewandelt werden. Der Manager entpackt die Datei automatisch

- profile>download – Download der ausgewählten Datei per Browser. Das Profil wird als Zip-Datei heruntergeladen
- profile>manage – Profile umbenennen oder löschen
- editor>run – Startet den Web Editor. Dabei wird der Editor-Server auf dem Pi gestartet. Zugriff auf den Editor mit Klick auf den Link. Der Editor öffnet ein neues Browserfenster unter IP-Adresse und Port des Editors; z.B. 192.168.1.108:8082
- editor>exit – Beendet den Web Editor
- options>manager/ options>autostart – Bearbeiten der Manager-Konfiguration
- options>email – Bearbeiten der eMail-Alarm-Optionen ([siehe Kap. 20.1](#))
- logs>Download Log – Download der Pi Presents Logdatei
- logs>Downloads Stats – Download der Pi Presents Statistikdatei
- Pi>Reboot/Pi>Shutdown – Neustart oder Herunterfahren des RPI

### Autostart nutzen

Wenn das Feld in der Manager Pfadeinstellung (autostart\_path) nicht leer ist, und der Manager läuft, wird Pi Presents mit den eingerichteten Profilen und Optionen starten

Um den Autostart in Raspbian zu öffnen ([siehe Kap. 6.6](#)) benötigt es den Kommandozeilenbefehl:

```
usr/bin/python /home/pi/pipresents/pp_manager.py
```

an Stelle eines Befehls für den Start von Pi Presents.

Wenn das Profil gestartet ist, startet auch der Manager-Server, auf den per Browser zugegriffen werden kann. Der Manager zeigt an, dass ein Profil läuft.

Es ist beabsichtigt, das der Manager durchgehend parallel zu Pi Presents läuft. Wenn das zu viel Prozessorleistung abzieht, sollte in der letzten Zeile von pp\_manager.py editiert werden:

```
von          update_intervall=0.1 (Sek.)
zu           update_interval=0.5 (Sek.)
```

## 19.3 Benutzung des Web Editors

Wenn Pi Presents von Ferne gesteuert wird, wird der Web Editor normalerweise vom Manager gestartet und gestoppt, er kann jedoch auch unabhängig davon laufen:

- Mit dem Befehl `python3 pp_web_editor.py -r` in einem Terminalfenster im Verzeichnis pipresents wird der Web Editor Server gestartet.

Andrerseits kann der Web Editor per Browser unter IP-Adresse und Port, z.B. 192.168.1.108:8082 erreicht werden.

Der Web-Editor bietet mehr Funktionen an als der interne Editor des Pi. Es gibt ein paar kleinere Unterschiede.

- Navigation für die Auswahl von Dateien und Verzeichnissen  
Verzeichnis öffnen mit Klick auf das Icon. Verzeichnis auswählen mit Klick auf den Namen.  
Datei auswählen mit Klick auf den Namen
- Farbnamen können benutzt werden. Wenn jedoch der Farbwähler verwandt wird, wird der Farbname in den Hexadezimalcode der RGB-Farbe umgewandelt
- Der Farbwähler wird vom Internet Explorer nicht unterstützte n

- Verschiedene Browser stellen Seiten unterschiedlich dar. Der Internet Explorer scheint der Hauptübeltäter bei der abweichenden Darstellung von Seitenlayouts zu sein. Manager und Editor wurden mit Hilfe von Firefox und Chromium (aktueller Browser des Raspberry Pi) entwickelt

## 20 Email Meldungen

Pi Presents und der Manager können (Alarm)Meldungen per eMail versenden.

email_with_ip	Wird vom Manager beim Start gesendet. Die eMail enthält die IP-Adresse des Pi für die Arbeit mit dem Remote-Browser
email_at_start	Wird von Pi Presents beim Start gesendet
email_on_error	Wird gesendet, wenn Pi Presents endet, weil ein fataler oder Profilfehler aufgetreten ist. Die Meldung zeigt den Fehlertext
email_on_terminate	Wird gesendet, wenn Pi Presents endet, entweder durch Beendigung durch den Pi oder durch Auswahl von ‚exit‘ im Manager
log_on_error	(nicht eingebaut) Hängt die Logdatei an die eMail

### 21.1 Email-(Alarm)Meldungen einrichten

Vor dem Einsatz der eMail-(Alarm)Meldungen:

- Die Datei /pipresents\_config/pp\_web.cfg muss mit einem Texteditor bearbeitet werden. Die [network] Abschnittsfelder ‚preferred\_interface‘ und ‚force\_ip‘ müssen eingerichtet sein. Für den normalen Gebrauch sollten sie leer sein.
- Die Datei /pipresents/pp\_config\_pp\_email.cfg muss mit einem Texteditor bearbeitet werden. Einträge im Abschnitt [email\_editable] können danach im Manager bearbeitet werden.

Field	Example	Description
<b>[email]</b>		
server		URL des Servers, von dem die eMails gesendet werden. Diese Software wurde nur mit gmail getestet. Weil Benutzername und Passwort nicht verschlüsselt werden, sollte nicht das normale eMail-Konto benutzt werden. Ein gmail-Konto speziell für diesen Zweck einzurichten wird empfohlen. Dabei ist es erforderlich, das ‚advanced security feature‘ im gmail-Konto abzuschalten.
port		Port des Servers, von dem die eMails gesendet werden.
username		
password		

<b>[email_editable]</b>		
email_allowed		Globales Einschalten. ‚yes‘ um alle Alarmtypen freizuschalten ‚no‘ nichts freischalten
to		Liste der eMail-Empfängeradressen an die Meldungen gesendet werden. Eine Zeile pro Adresse
sender		Eine eMail-Adresse für das Feld: From; Wenn es das Feld nicht gibt oder es leer ist, wird der Benutzername eingetragen.
email_with_ip email_at_start email_on_error email_on_terminate		‚yes‘ um den eMail-Alarm zu gestatten ‚no‘ um keinen eMail-Alarm zu gestatten
log_on_error		Nicht eingebaut ‚yes‘ um Logs an eMail anzuhängen ‚no‘ um keine Logs an eMail anzuhängen

## 20.2 Mit eMail-(Alarm)Meldungen arbeiten

Wenn pp\_email\_config eingerichtet ist, werden (Alarm)Meldungen automatisch gesendet. Der editierbare Teil der pp\_email\_config kann mit dem Manager modifiziert werden.

## 21 Anforderungen an die Hardware

Pi Presents läuft sowohl auf Modell 1, Modell 2 oder Modell 3 und auch auf Model 4 des Raspberry Pi. Für eine bessere Performance bei Videos und Bildern werden jedoch Modell 2 und Modell 3 oder 4 empfohlen. Die GPIO-Pins wurden so ausgewählt, dass die Version irrelevant ist. Model 4 ist nur erforderlich, wenn zwei Displays benutzt werden sollen.

Für das Abspielen von Videos sollten eigentlich 256-MB für die GPU reserviert werden. Allerdings verursacht das auf dem Pi Zero unzureichenden CPU-Speicher, so sollte man 128 MB einstellen.

Die Bildanzeige ist mit älteren RPI langsam. Ein 1-Megapixel-Bild benötigt zum Aufbau etliche Sekunden. Ein 10-Megapixel-Bild benötigte im Test 10 Sekunden und brachte einen 256-MB-Pi zum Absturz. Bildauflösungen, größer als die Bildschirmauflösung, bringen keine bessere Bildqualität und benötigen auch auf 512-MB-Maschinen lange Ladezeiten. Ich benutze den brillianten Faststone Photo Resizer <http://www.faststone.org/FSResizerDetail.htm> um die Bildgrößen auf einem Windowsrechner zu verringern.

Für den Sound benutzen Sie HDMI oder den Kopfhörerausgang. Verstärkung und Lautstärkekontrolle können mit externer Hardware erfolgen und so die Anwendung ergänzen.

## 22 Pi Presents updaten

## 22.1 Profile updaten

Weil Pi Presents sich weiter entwickelt, werden Felder zu den Profildefinitionen hinzugefügt, andere gelöscht. Um Kontinuität sicherzustellen müssen Pi Presents, der Editor und die Profile die gleiche Version tragen. Pi Presents warnt, wenn ein Profil mit falscher Versionsnummer benutzt wird. Um das zu korrigieren, öffnen Sie dieses Profil im Editor, der das Profil und seine Felder automatisch updated. Eine Sicherungskopie der bisherigen Einträge wird in `pp_profiles.bak` gespeichert- Es gibt einige wenige Update-Schritte, die nicht automatisiert werden können. Lesen Sie die Release Notes, um mehr darüber zu erfahren.

Wenn viele Profile innerhalb eines Verzeichnisses upgedated werden sollen, steht unter der Menüoption des Editors `Tools>Update All` ein entsprechendes Werkzeug zur Verfügung.

Die Profile im `pipresents-gapless-examples` github repository sind immer auf dem Stand der neuesten Version von Pi Presents. Achtung: Eine Neuinstallation könnte eigene Profile überschreiben.

## 22.2 Pi Presents updaten

Vor einem Update sollte zur Sicherheit eine Kopie des `pipresents`-Verzeichnisses nebst zugehörigen Daten gespeichert werden.

Laden Sie Pi Presents von Github herunter und installieren Sie es, wie in der Datei `README.md` beschrieben. Daten in `/home/pi/pp_home` werden nicht angerührt. Speichern oder ändern Sie Ihre eigenen Daten nicht im `pipresents/pp_home` Verzeichnis, weil sie dort überschrieben werden könnten.

Wenn Sie Pi Presents updaten, lesen Sie bitte die Versionshinweise. Sie müssen möglicherweise die Konfigurationsdateien updaten und einige wenige Aufgaben erledigen, die nicht automatisch ablaufen.

# 23 Fehlerkorrektur, Statistiken, Fehlerberichte und neue Programmwünsche

## 23.1 Statistiken erstellen

Pi Presents zeichnet Ereignisse in der Datei `/pipresents/pp_logs/pp_stats.txt` auf. Ihr Inhalt eignet sich zur Analyse, um Statistiken über die Nutzung eines Programmes zu erstellen.

Die Datei enthält CSV und kann in eine Tabelle eingelesen werden. Das Trennzeichen ist ein Semikolon, es kann in `pp_statsrecorder.py` geändert werden.

Die Statistikfunktion wird mit dem Befehlszeilen-Kommando `-d` aktiviert, wie in [Kapitel 23.2](#) beschrieben.

Wenn die Datei `pp_stats.txt` nicht existiert, wird sie beim Start von Pi Presents erstellt und mit einer Überschriftenzeile versehen. Wenn die Datei bereits vorhanden ist, wird sie nicht überschrieben. Pi

Presents fügt nach den vorhandenen Einträgen aus vorherigen Programmläufen die neuen Einträge hinzu.

Es gibt vier verschiedene Ereigniszeilen:

- Start – bedeutet, das Pi Presents gestartet wurde. Aufgezeichnet werden Datum, Zeit und der Profilname.
- start trigger, next trigger – zeigen an, das für eine Mediashow oder eine Liveshow ein start oder next ausgelöst wurde. Datum, Zeit und Showdetails werden aufgezeichnet.
- play child – zeigt an, das Mediashow oder Liveshow ein Signal zum Abspielen einer Childshow erhalten haben. Der Inhalt ist gleich mit den nachfolgenden Ereignissen
- Andere Befehle wie play, call – Menushows, Radiobuttonshows and Hyperlinkshows zeichnen Datum und Zeit sowie Showdetails auf, wenn durch Benutzer verursachte Ereignisse auftreten.

Showdetails sind show type, show reference, show title, profile. Trackdetails sind track type, Track-Referenz, track title, location und profile.

Beispiel:

```
"Date";"Time";"Show Type";"Show Ref";"Show Title";"Command";"Track Type";"Track Ref";"Track Title";"Location", profile
```

```
"2016-02-01";"15:02:43";"";"";"";"";"start";"";"";"";"/pp_interactive_1p4";"/pp_interactive_1p4
```

```
"2016-02-01";"15:02:53";"mediashow";"mediashow";"Mediashow";"play child";"menu";"mymenu";"Menu";"";"/pp_interaktive_1p4
```

```
"2016-02-01";"15:02:58";"menu";"mymenu";"Menu";"play";"image";"";"A Stunning River Scene";"+/media/river.jpg";"/pp_interactive_1p4
```

Die Aufzeichnung von Statistiken für ein Gerät ist getrennt vom Rest der Pi Presents Daten. Sie finden sich in der Datei pp\_statsrecorder.py. Dieses Modul kann überschrieben werden, wenn ein anderer Statistikrecorder, wie InfluxDB, eingesetzt wird.

## 23.2 Fehler in Profilen beseitigen

Die Befehlszeilenoption -d listet die Operationen von Pi Presents, so dass sie auf dem Bildschirm ausgegeben und in einer Log-Datei gespeichert werden können ([Siehe Kap. 6.1](#)). Die Option -d kann mit einem Parameter die Log-Ausgabe fein steuern.

Wert (binär)	Wert (dezimal)	Log-Ausgabe
1	1	Schlimme-(System)-Fehler
10	2	Profil-Fehler
100	4	Warnungen
1000	8	Liste für Profilreparatur geeignet
10000	16	Liste für Reparatur von Pi Presents geeignet
100000	32	Liste für Reparatur in Pi Presents mit Player- und

		Shower-Instanzen
1000000	64	Anzeige von Speicherlecks um Fehler in Pi Presents zu beseitigen.
10000000	128	Schreibt Ereignisse auf – brauchbar für Statistiken in stats.txt (s. Kapitel 23.1)
1000000000	256	Liste für Reparatur von Time of Day geeignet

Ohne -d Option benutzt Pi Presents den Wert 7. Mit -d Option ohne spezifizierten Wert wird der Wert 15 angenommen.

Die Reportfunktion über Datenmüll ist immer aktiv. Das kann dazu führen, dass eine Meldung im Terminalfenster erscheint, wenn Pi Presents geschlossen ist. Die Meldung zeigt an, dass Pipresents Tracks oder Shows nicht vollständig gelöscht hat. An diesen Meldungen bin ich sehr interessiert.

Zusätzlich zu den Auflistungen kommen Reparaturmeldungen als Texte aus dem Pythoncode, die ich unabsichtlich dort hinterlassen habe. Sie tauchen als Meldungen im Terminal auf, selbst wenn die Fehlersuche abgeschaltet ist.

### Fehlersuche und Programmwünsche

Bitte benutzen Sie Github Issues Tab

<https://github.com/KenT2/pipresents-gapless/issues>

um Fehler aber auch neue Ideen für Erweiterungen zu senden.

Ich bin versessen darauf, Pi Presents zu verbessern. Ihre Erfahrungen im Echtbetrieb und Ihre Wünsche an das Programm sind für mich von unschätzbarem Wert, weil sie mir beim Feintuning der aktuellen Funktionalität im Kleinen und bei größeren Verbesserungen helfen.

## 24 Lösungen und bekannte Probleme

### Beim Autostart oder Start mit einer Desktopverknüpfung wird das Profil nicht gefunden

In diesen Fällen ist es das Beste, den absoluten Pfad von Pi Presents und Daten-Heimatverzeichnis in die Kommandos aufzunehmen z.B.

```
usr/bin/python3 /home/pi/pipresents.py -o /home/pi -p myprofile
```

Bitte achten Sie darauf, dass der Speicherort der Autostartdatei seit Wheezy (25.12.2014) geändert wurde und auch sein Inhalt in Jessie/Stretch/Buster geändert wurde.

### Pi Presents hängt im Vollbildschirm fest, wie kann ich da herauskommen?

Das liegt daran, dass Python eine Ausnahme berichtet, die auf inkorrekte Konfigurationsdaten

zurückzuführen ist. Um das zu vermeiden, benutzen Sie Validation im Editor und testen die Show, nicht im Vollbildschirm aber in einem Terminalfenster, dann geht CTRL-C, wenn CTRL-BREAK nicht funktioniert. Wenn das fehlschlägt, versuchen Sie mit Ctrl+Z das Terminalfenster zu schließen.

Wenn Pi Presents abstürzt, hinterläßt es schon einmal MPV oder mplayer laufend, was Sie mit top herausfinden können. Um diese Prozesse zu beenden:

```
killall mpv mplayer
```

Das hinterläßt möglicherweise Zombie-Prozesse. Um die zu löschen, muss das Terminalfenster geschlossen werden.

Eine Stromunterbrechung könnte das System korrumpieren, so wäre eine saubere Lösung für einen nahtlosen, vollständigen Programmschluss, von einem anderen Rechner mit SSH auf den Pi zuzugreifen, top ( top -upi) zu starten und den Pythonprozess mit einem k xxxx Befehl abzuberechnen.

### **Mein Video/Audiotrack läuft nicht in Pi Presents**

Versuchen Sie mit MPV oder MPlayer von der Befehlszeile abzuspielen.

### **Unclutter versteckt den Cursor nicht durchgehend**

Azizar berichtete das er /etc/default/unclutter, geändert hat und folgendes zugefügt wurde:

```
-grab  
-noevents
```

### **Ich habe ein Input- oder Outputgerät eingerichtet, aber es funktioniert nicht mit Pi Presents**

Es gibt zwei einzeln funktionierende GPIO-Testprogramme im pipresents Verzeichnis 'input\_test.py' und 'output\_test.py'. Damit können Sie Ihre I/O testen bevor Sie Pi Presents tadeln!