

Inhaltsverzeichnis

Einrichten des Sat->IP Clients Schnipsl	3
<i>Wiederbelebung eines Telestar Digibit R1</i>	3
<i>Kodi als Player einrichten</i>	3
<i>Kathein EXIP418</i>	3

Einrichten des Sat->IP Clients Schnipsl

Die Schnipsl-Doku gibts hoffentlich bald mal [hier](#)

Wiederbelebung eines Telestar Digibit R1

Ein Digibit ist okay, solange er läuft. Leider fällt er nunmehr zum wiederholten Mal in eine absolute Totenstarre, wo selbst ein Zurücksetzen auf Werkseinstellungen und ein paar Stunden vom Netz trennen diesmal nicht mehr geholfen haben.

Ein Tipp jedoch hat diesmal noch geholfen: Das Flashen der Firmware von der [Telestar Download Seite](#) (

soft_digibit_r1_linux_001.zip

)

Kodi als Player einrichten

Skin Eminence Startfenster TV- Kanäle Hauptmenu anpassen

Einstellungen/System/Eingabe/Peripheriegeräte: CEC-Adapter:

- Beim Starten von Kodi als aktive Quelle melden
- Beim Starten von Kodi einschalten: TV
- Beim Ausschalten von Kodi ausschalten: TV
- Beim Beenden von Kodi als inaktive Quelle melden
- Geräte bei Bildschirmschoneraktivierung in Standby versetzen
- Gerät aufwecken, wenn der Bildschirmschoner deaktiviert wird
- wenn der TV ausgeschaltet wird: Wiedergabe stoppen
- Geräte zusammen mit Kodi in Standby versetzen

Einstellungen/Dienste/Steuerung: Steuerung über HTTP erlauben

Addons/Aus Repository installieren/LibreELEC Add-ons/PVR-Clients: PVR IPTV Simple Client

Einstellungen/Benutzer-Addons/PVR-Clients: PVR IPTV Simple Client: Entfernter Ort:
<http://192.168.1.113:8080/m3u>

Einstellungen/LibreELEC-Einstellungen: Systemname kodi-wohnzimmer

Kathein EXIP418

Nachdem der Digibit nun wohl endgültig den Löffel abgegeben hat, musste das Sparschwein notgeschlachtet werden und es wurde das letzte noch am Markt verfügbare Gerät, der Kathein EXIP418.

Der erwies sich gleich in zweierlei Hinsicht als hinterhältig:

- Zum einen scheinen seine Eingänge wohl tatsächlich keine variablen Steuersignale auszugeben, um einen Quad LNB nach Bedarf zu steuern, sondern jeder seiner vier Eingänge scheint fest verdrahtet und man **muss** auch alle vier Eingänge beschalten, sonst fehlen Kanalbereiche...
- und er liefert nicht, wie der alte Digibit, eine eigene M3U Kanalliste per HTTP, sondern die muss man sich für seine KODIs selberbauen. Und damit begann das Elend..:

Zuerst einmal ist Kodi auf dem FireTV überaus intrigant. Obwohl die Konfig- Settings des Simple IPTV-Client- AddOns was anderes vorgaukeln, wird eine neue Playliste scheinbar nur dann wirklich wieder geladen, wenn man ganz unten (da, wo man's erst mal nicht sieht..), Aktualisierung auf dauernd bzw. täglich schalten. Ich möchte nicht wissen, wieviele Versuche ich wohl in dem Glauben gemacht habe, er hätte eine neue M3U geladen und es war in Wirklichkeit noch eine Alte.

Dann, nach zwei Wochen googlen und probieren, bin ich auf einen [Hinweis](#) von [Phunkyfish](#) gestoßen, der mir damals schon mal bei den ersten IPTV massiv geholfen hatte.

Scheinbar muss man in der „normalen“ M3P Playlist noch jeden Eintrag mit zusätzlichen KODI-Direktiven (KODIPROP) aufpeppen:

```
#KODIPROP:inputstreamclass=inputstream.ffmpegdirect
#KODIPROP:inputstream.ffmpegdirect.mime_type=video/mp2t
#EXTINF:0,Zwei Music
http://exip418/?src=1&freq=12148&pol=h&ro=0.35&msys=dvbs&mtype=qpsk&plts=off
&sr=27500&fec=34&pids=0,17,18,100,3071,3072
```

Das dann in Verbindung damit, dass Kodi vielleicht auch mal die neueste M3P auch wirklich mal lädt, bringt dann nach Tagen der Tränen endlich mal wieder das Fernsehen auf den Bildschirm..

Das Herunterladen einer Astra- Liste, die Umformatierungsarbeiten und das Bereitstellen zum Download für die Kodi- Clients macht dann dieser Python- Codeschnipsel

[Hier der komplette Python SAT- Proxy Code](#)

[satip_proxy.py](#)

```
#!/usr/bin/env python3

from http.server import HTTPServer, BaseHTTPRequestHandler
import socketserver as SocketServer
import sys
import requests
import logging

PORT = 8000

final_m3u=""

class RequestHandler(BaseHTTPRequestHandler):
    def _set_headers(self):
        self.send_response(200)
```

```
self.send_header("Content-type", "plain/text")
self.end_headers()

def _html(self, message):
    return message.encode("utf8") # NOTE: must return a bytes
object!

def do_GET(self):
    self._set_headers()
    self.wfile.write(self._html(final_m3u))

def do_HEAD(self):
    self._set_headers()

def do_POST(self):
    # Doesn't do anything with posted data
    self._set_headers()
    self.wfile.write(self._html(final_m3u))

import argparse

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "-u",
        "--url",
        help="url to get original m3u list from",
        default="https://raw.githubusercontent.com/dersnyke/satipplaylists/main/satip_astral92e.m3u",
    )
    parser.add_argument(
        "-f", "--favorites", help="the favorite file to read the favorites from"
    )

    args = parser.parse_args()

    r = requests.get(args.url)

    print("Status Code:")
    print(r.status_code)
    if r.status_code != 200:
        sys.exit()
    senders = {}
    favorites = []
    sender = ""
    name = ""
    new_m3u=["'#EXTM3U'"]
    # print (r.text)
    lines = r.text.split("\n")
    for line in lines:
```

```

line = line.strip()
if line[:1] == "#":
    #print(line)
    elements = line.split(",", 1)
    if len(elements) < 2:
        continue
    name = elements[1].strip()
    sender = line
else:
    url=line
    url = url.replace("rtsp:", "http:")
    url = url.replace("sat.ip", "exip418")
    senders[sender] = {"name": name.lower(), "url": url}
if args.favorites:
    with open(args.favorites, encoding="utf8") as fin:
        favorites = fin.read().lower()
for sender, data in senders.items():
    if not favorites or data["name"] in favorites:
        #new_m3u.append(f"{sender}\n{data["url"]}")
        new_m3u += [
            "#KODIPROP:inputstreamclass=inputstream.ffmpegdirect",
            "#KODIPROP:inputstream.ffmpegdirect.mime_type=video/mp2t",
            sender,
            data["url"]
        ]
final_m3u="\n".join(new_m3u)
print(final_m3u)
Handler = RequestHandler
httpd = SocketServer.TCPServer(("", PORT), Handler)

httpd.serve_forever()

```

```

{
  "playlists": {
    "wohnzimmer": {
      "adds": [
        "#KODIPROP:inputstreamclass=inputstream.ffmpegdirect",
        "#KODIPROP:inputstream.ffmpegdirect.mime_type=video/mp2t"
      ],
      "replaces": [
        {
          "from": "rtsp://sat.ip",
          "to": "http://exip418"
        }
      ],
      "stations": [
        "das erste hd",
        "zdf hd",
        "zdf_neo hd",
        "zdfinfo hd",

```

```
    "one hd",
    "3sat hd",
    "ard alpha hd",
    "arte hd",
    "wdr hd aachen",
    "ndr fs hh hd",
    "tele 5",
    "kabel eins",
    "kabel eins doku",
    "nitro",
    "prosieben",
    "pro7 maxx",
    "rtl",
    "rtlzwei",
    "sat.1",
    "sat.1 gold",
    "super rtl",
    "vox",
    "kika hd",
    "n24 doku",
    "dmax",
    "phoenix hd",
    "tagesschau24 hd",
    "deluxe music"
  ]
}
},
"sources": [
"https://raw.githubusercontent.com/dersnyke/satipplaylists/main/satip_astral92e.m3u"
]
}
```

From:

<http://www.koehlers.de/wiki/> - **Steffen Köhlers Online- Bastelbuch**

Permanent link:

<http://www.koehlers.de/wiki/doku.php?id=pc:linvdr2021>

Last update: **2025/09/02 14:58**

