

## ITエンジニアのキャリア形成

Author | Rio Fujita  
Twitter | @rioriost

概要 | IT 業界でこの先生きのこるには

### 概要

変化の速い IT 業界で生き残るにはどうしたら良いのか、私見を述べることを許してもらえらなるといふことで、思い付くところをつらつらと書いてみよう。

### キャリアアップ or ステイ

生存バイアスなのかもしれないけれど、ともかくこの業界で 50 歳まで生き残っている一人として、同じく生き残っている人たちを見ると大体ある種の共通点がある。55 歳か 60 歳定年制か、あるいはもっと長く働ける職場か否かに関係なく、ずーっと最新の技術動向に興味を持ち続けてキャッチアップしている人が多い。それは例えば、おもむろに、

```
import configparser
import ipaddress
import os
import sys
import pathlib
```

してみたりするということだ。ちょっとこれだけだと寂しいので、もっと import しちゃったりする。

```
import PySimpleGUI as sg
import rtsp
from PIL import Image, ImageTk
import keyring
```

PySimpleGUI ってことは、なんか GUI を作ろうとしてるし、rtsp や PIL ってことはビデオストリーミングかな？って気分になってくる。keyring があれば、OS 標準のキーストアが使えるから認証が必要な RTSP サーバの雰囲気も漂ってきて、気分が盛り上がるよね。

### メイン

そうそう、main を書かんと。「あー、ばあさんや、main はどこにやったかのう？」「あら、じいさんってば、また main を忘れちゃったんですか？」なんて成りたくないだろ？

```
def main() -> None:
    user_id, user_pw = init_account()
    row_col, hosts = init_hosts()
    open_window(user_id, user_pw, row_col, hosts)
if __name__ == '__main__':
    main()
```

アカウントをセットして RTSP サーバのリストをセットしてウインドウを生成すれば、もう出来上がりだね。はー、Python って便利だなあ。何、init\_account が見つからない、何をバカな、ちゃんと書いたはずだぞ。

### アカウントの初期化

```
def init_account() -> dict:
    user_id = keyring.get_password(APP_NAME, key_id)
    while user_id is None or user_id == "":
        keyring.set_password(APP_NAME, key_id, input("Please
        enter your Tapo ID, part before '@': "))
        user_id = keyring.get_password(APP_NAME, key_id)
    ...
    return user_id, user_pw
```

keyring で読んでくるだけのことでガタガタ抜かすんじゃないよ、全く。macOS だとキーチェーンアクセス.app で確認できるやつだな、/アプリケーション/ユーティリティ/なんて見たことない？ いやいやいや、Hi-Res オーディオ勢なら、Audio MINI 設定.app を使わないなんてこたあないだろ？

### RTSP サーバリストの初期化

インタプリタが文句言う前に、init\_hosts も書いてやるか、大サービスってやつだ、金輪際こんなこたあねえから、気いつけな！

```
def init_hosts() -> None:
    config_path = pathlib.Path(f"--/Library/Preferences/{APP_NAME}
    /config.ini").expanduser()
    if config_path.exists() is False:
        config_path.parent.mkdir(parents=True, exist_ok=True)
        row_num = input_number("Please enter the number of rows
        for streams e.g. 2: ")
        col_num = input_number("Please enter the number of
        columns for streams e.g. 3: ")
    ...
    for key, host in hosts.items():
        tmp_ip = None
        while validate_ip(tmp_ip) is False:
            tmp_ip = input_ip_address(f"IP address for {key}: ")
        hosts[key] = tmp_ip
    ...
    config = configparser.ConfigParser()
    config["LAYOUT"] = {"row": row_num, "col": col_num}
    config["HOSTS"] = hosts
    with open(config_path, "w") as f:
        config.write(f)
```

設定ファイルが見つからないなら作りゃいいじゃねーか、このすつとこどっこい！ 八つぁんや、そんな乱暴なことを言いなさんな、そこに鰻頭と IP アドレスを用意しといたから、持って帰んな。IP アドレスってのはあれだね、ご隠居、到来物だって。それを言うなら try ものだよ、八つぁんや、IP アドレスは except で一杯引っかけ、お、なんだい、この row とか col ってえのは、碁盤かい、碁石でも並べて、

## ■ ウィンドウ生成&サーバ接続

ではこれより、open\_window を始める！ 接続しちゃえば、まあね、キャリアも繋がるってのもですよ、あ、搬送波は関係ないか？ PySimpleGUI だと、layout にリストを渡すと、良い感じにレイアウトしてくれる。window のインスタンスを生成し、表示するストリーミングの数だけ client を生成しておく。

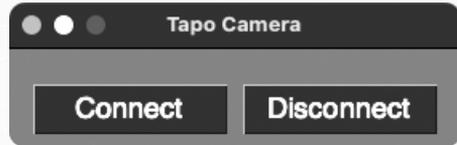
```
def open_window(user_id: str, user_pw: str, row_col: dict, hosts: dict) -> None:
    layout = [
        [[sg.Image(filename='', key=f"image{row}-{col}") for col
         in range(row_col["col"])] for row in range(row_col["row"])],
        [sg.Button('Connect', size=(10, 1), key='-connect-'),
         sg.Button('Disconnect', size=(10, 1), key='-disconnect-')]
    ]
    is_streaming = False
    window = sg.Window('Tapo Camera', layout, location=(32, 32),
                       finalize=True, element_justification='center', font='Helvetica 18')
    clients = {f"{row}-{col}": None for row in range(row_col["row"])
              for col in range(row_col["col"])}
    while True:
        event, values = window.read(timeout=1)
        if event == sg.WIN_CLOSED:
            break
```

Connect ボタンが押されたら、client を RTSP サーバーに接続。ストリーミング中フラグを立てておく。

```
elif event == "-connect-":
    if is_streaming is False:
        .....
        host = hosts[f"stream{row}-{col}"]
        if host != "":
            print(f"Connecting to {host}...")
            rtsp_url = f"rtsp://{user_id}:{user_pw}@{host}:554/stream2"
            clients[f"{row}-{col}"] = rtsp.Client(
                rtsp_server_uri=rtsp_url)
            is_streaming = True
```

繋がったら、client から読み出したフレームを window の image として描画してあげることで、良い感じに RTSP がパラパラ漫画になる。

```
if is_streaming is True:
    .....
    host = hosts[f"stream{row}-{col}"]
    if host != "":
        frame = clients[f"{row}-{col}"].read()
        if frame is not None:
            window[f"image{row}-{col}"].update(data=ImageTk.PhotoImage(frame))
```



▲ 図 1：接続と切断ボタンだけ、大変にシンプル



▲ 図 2：3 x 2 のレイアウトで、5 台のカメラを表示

Disconnect ボタンが押されたら、client の接続を切断して、適当に Image を塗っておく。

```
elif event == "-disconnect-":
    if is_streaming is True:
        is_streaming = False
        .....
        host = hosts[f"stream{row}-{col}"]
        if host != "":
            clients[f"{row}-{col}"].close()
            img = Image.new('RGB', (640, 480), color=0)
            window[f"image{row}-{col}"].update(data=ImageTk.PhotoImage(img))
```

## ■ 実行してみる

コード書き終わったら、実行する。

```
% python3 Tapo_Streamer.py
```

図 1 は実行直後の画面。ちょっと寂しい人は、適当な Image でも配置して良い感じにするがよろし。

接続ボタンをクリックすると、しばらくしてストリーミングが表示される。

## ■ まとめ

コードは、[https://github.com/rioriost/-Tapo\\_Streamer/](https://github.com/rioriost/-Tapo_Streamer/) に放流。

## ■ 著者情報



現在は衛星経由で監視されてるハムスター。

Linux チョットテキル。  
ぺんぎんさんに悪の秘密結社に誘われました。