

FT8 Image Transceiver プログラム開発体験

Shu JA3GQJ

ChatGPT に「WSJTX」の FT8 で JPEG 画像を送受信する Python プログラムを書かせてみたのでその体験のメモを PDF ファイルに残す。

1. ChatGPT に指示（依頼）した画像送受信のプロセス

ChatGPT に指示（依頼）したことを纏めると概略次のようになるが、ChatGPT とのやり取りは数えきれない回数であった。

1-1. ChatGPT に指示（依頼）した送信画像準備のプロセス

- ・送信用画像をローカルフォルダーから選択して取り込んでキャンバスに表示する。
- ・画像取り込み用ボタンを用意する。
- ・キャンバスに表示した画像を ftp サーバーの指定するフォルダーにアップロードする。
- ・画像アップロード用のボタンを用意する。
- ・アップロードが完了したらその時間をキャンバスの下に表示する。

1-2. ChatGPT に指示（依頼）した画像送信のプロセス

- ・ C:\WSJTX\wsjtx\bin\wsjtx.exe の WSJTX を起動する機能とボタンを用意する。
- ・起動した WSJTX の送信許可 (n) ボタンを On にする機能とボタンを用意する。
- ・送信停止 (H) ボタンを操作する機能も必要である (ボタンは不要)。

1-3. ChatGPT に指示（依頼）した画像受信のプロセス

- ・ WSJTX が受信したコードを記録しているファイル ALL.txt をファイル A とする。
- ・ファイル A の内容は相手が送信するごとに変化する。
- ・この変化を監視して送信相手が変わるごとにコールサインを抽出する。
- ・送信局の送信画像がある URL のリストを作って WSJTX と同じフォルダーに保存したテキストファイル FT8IMGUsers.LST をファイル B とする。
- ・ファイル A から抽出したコールサインでファイル B のリストを検索して該当するコールサインを見つけてそのコールサインの URL から該当局の画像をダウンロードしてキャンバスに表示する。
- ・表示したキャンバスの下に受信時間と送信局のコールサインも表示する。
- ・受信画像をディジサイドにアップロードする Python のコードとボタンを用意する。

1-4. 通信ソフトに WSJTX を使用する理由

ファイル A に必要な WSJTX が自身で保存するテキストファイルの名前が ALL.TXT で固定され、内容だけが受信数に応じて増加するだけなので利用し易いからである。

JTDX は記録するフォーマットは WSJTX と同じであるがファイル名が日替わりするので扱い難く Python のコードが複雑になる。

2. FT8 Image Transceiver

完成したソフト名を **FT8 Image Transceiver** とした。図 1 が起動時の画面である。

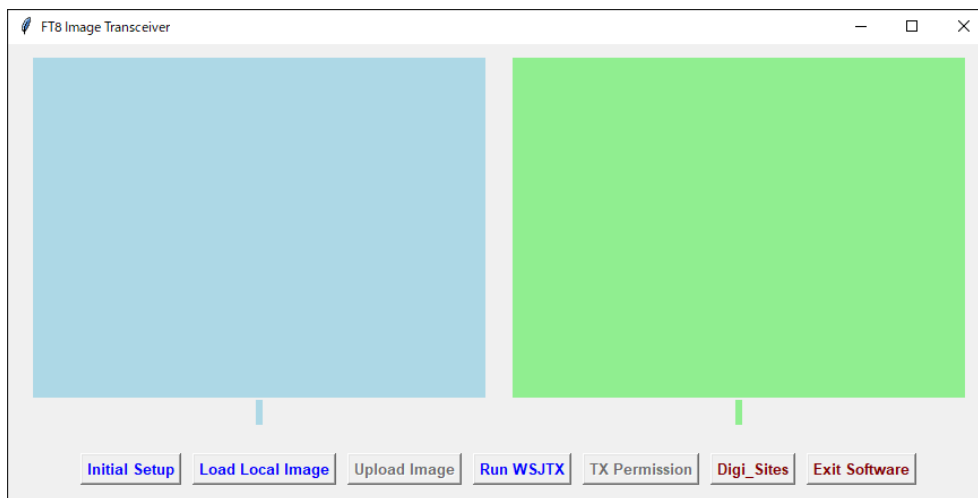


図 1

画面には ChatGPT に指示して作った七つのボタンがある。

• 「Initial Setup」 ボタン

クリックすると図 2 の画面がポップアップして

- ① 送信画像アップロード用 ftp サーバーの情報
- ② ファイル A とファイル B のディレクトリー (File A Path と File B Path)
- ③ デジサイトにアップロードする画像があるディレクトリー (Local file Path)
- ④ 受信画像アップロード用デジサイトの ftp サーバーの情報
を入力して保存する。

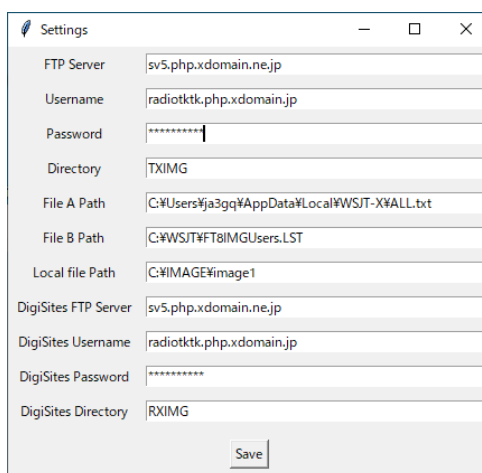


図 2

入力したデータ (情報) は「**FT8 Image Transceiver**」と同じフォルダー (C:\IMAGE) にファイル名「**ftp_settings**」として保存、二回目以降はその情報を自動で読み込んでそれぞれの目的に使用する。「**ftp_settings**」はメモ帳などで開くことができる。

フォルダーのディレクトリーは送信画像用「TXIMG」、デジサイト用「RXIMG」で既存のソフト悪影響を与えないように専用のフォルダーにした。

• 「Load Local Image」 ボタン

ボタンをクリック → フォルダーを開く (図 3) → 送信する画像を選択 → 読込んだ画像をキャンバスに表示 (図 4) → 「Upload Image」 ボタンが有効になる。



図 3

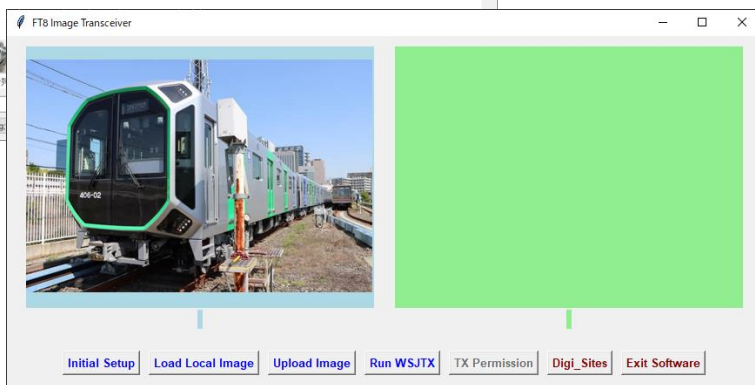


図 4

• 「Upload Image」 ボタン

ボタンをクリック → 画像のファイル名を `image1` にして `ftp` サーバーのフォルダー「TXIMG」にアップロードする。アップロードが終わるとその時間を表示する (図 5)。

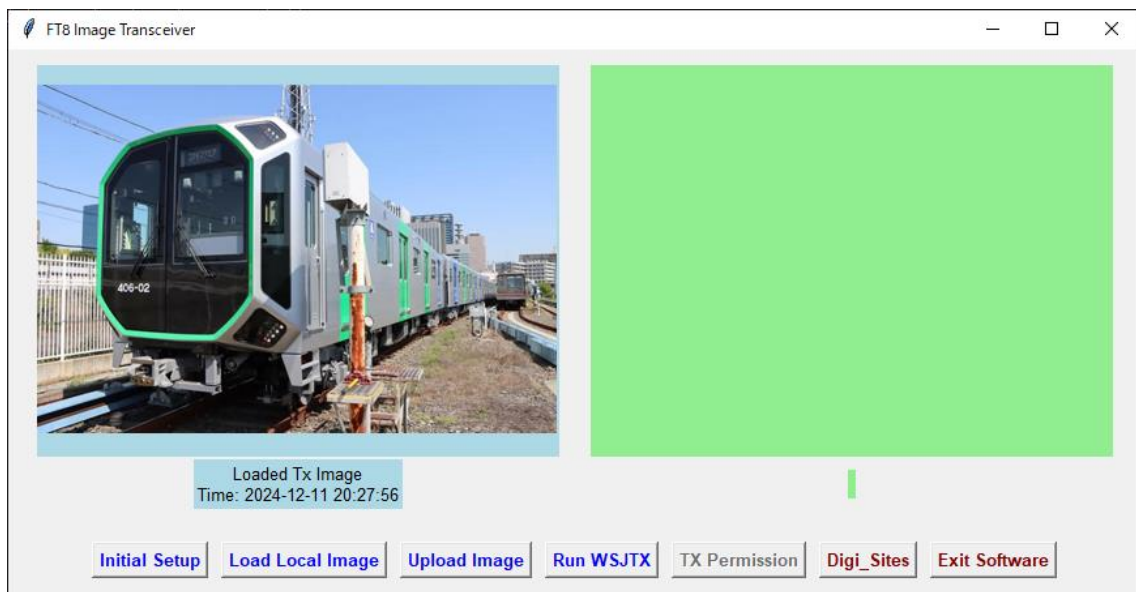


図 5

- 「Run WSJTX」 ボタン

ボタンをクリック → WSJTX が起動する (図 6)。

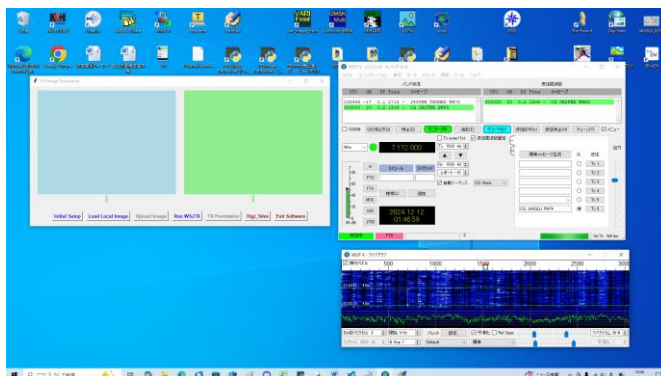


図 6

- 「TX Permission」 ボタン

送信画像読み込み終了 → ボタンをクリック → WSJTX の「送信許可(n)」ボタン On (赤色) になり → 定時に送信が始まる。

- 「Digi Sites」 ボタン

ボタンをクリック → 受信画像が Digi-Sites のフォルダーにアップロードされる。

問題点

既存の Digi-Sites を利用してファイル名 image1 でアップロード出来たが、アップロードするたびに上書きされてしまう、送信局のコールサインが表示されないなどの問題があるので、相手局へのリアルタイムの受信レポートの代わりに Digi-Sites を利用する方法を別途考えて既存の Digi-Sites に悪影響を与えない配慮が必要である。

- 「Exit Software」 ボタン

ボタンクリック → 「FT8 Image Transceiver」 を閉じる。

画像の受信

受信用の操作ボタンは「Digi Sites」ボタンのみである。FT8 の信号の受信が終わると自動的に送信相手のコールサインと受信時間がキャンバスに表示される。(図 7)

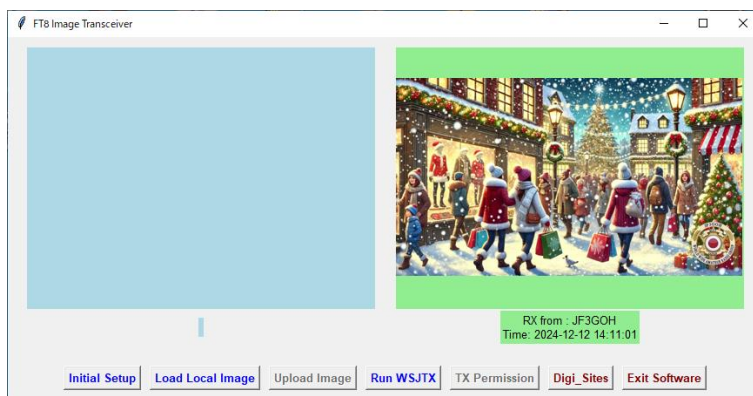
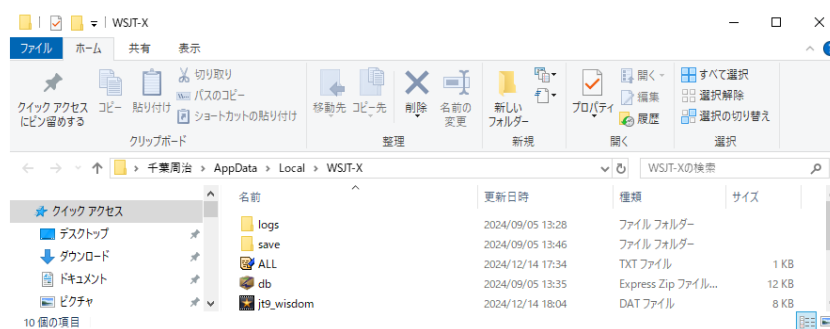


図 7

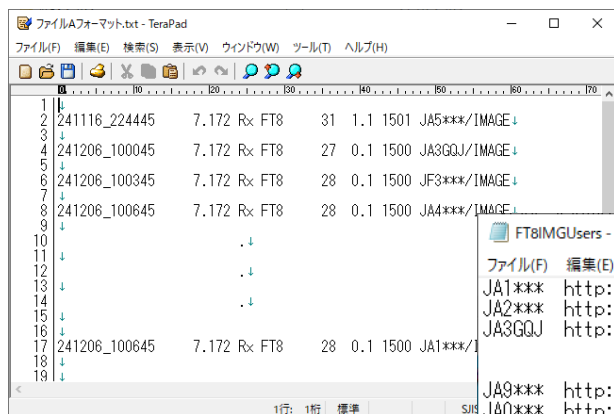
備忘録

- WSJTX のインストール先はデフォルトの `C:\¥WSJT¥wsjtx¥bin¥wsjtx.exe` でなければならない
- WSJTX の設定は TX マクロに JA3GQJ/IMAGE を追加する以外はウェブページなどの説明にある通常の設定でよい。
- WSJTX が記録する受信コードのテキストファイル ALL.txt は WSJTX のファイル→「ログディレクトリーを開く」を選択したらポップアップする次のフォルダーにある。



- WSJTX の起動、送信許可 (n)、送信停止 (H) の操作をソフトから行う為の関数の開発には苦勞した。
- WSJTX の起動にはひと工夫必要で WSJTX の送信許可 (n) ボタンをソフトから On するには「Alt + n」キーを送ればよいが、開発ソフトと、WSJTX は同時に走らせる必要がある。開発ソフト側のボタンをクリックして WSJTX を起動することは簡単である。しかし、この時カーソルが開発ソフト側に移っているので WSJTX 起動ボタンを押すためにはカーソルが WSJTX 側ではなくなるので「Alt + n」キーを送っても送信許可 (n) ボタンは手動の時のように On (有効) にならない。そのため WSJTX の起動するときにはカーソルの位置が開発ソフト側と WSJTX 側の両方で機能する配慮必要であった。
- 更に、送信許可 (n) ボタンが On (有効) になると送信は毎時 15 秒毎に繰り返し送信されてしまう現象を手動で On/Off する動作と同じことを自動化する必要があった。
- 自動化に当たっては毎時 15 秒毎に繰り返し送信される操作を一回の送信で終わらせる必要もあった。
- プロセスを自動化には苦勞した。手を変え、品を変えて ChatGPT に指示 (依頼) をしたが旨くいかず、最終的に時計を作って「Tx Permission」ボタンをクリックして毎時 14 秒毎の時間を時計から検出して WSJTX が送信を開始する 1 秒前に「Alt + n」キーを送って「送信許可 (n)」ボタンが On (有効) なるようにした。その結果、WSJTX の毎時 15 秒毎の送信が自動化出来た。そして、「Alt + n」キーを送った 13 秒後に WSJTX の「送信停止 (H)」ボタンを On (有効) にする「Alt + H」キーを送って送信の繰り返しをなくした。

- C:¥のディレクトリーにフォルダー「IMAGE」を作って開発したソフトを置く。
- ファイル A とファイル B のテキストファイル（フォーマット）

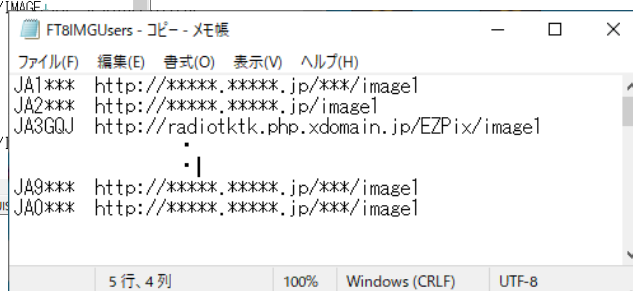


```

1 |
2 | 241116_224445 7.172 Rx FT8 31 1.1 1501 JA5***/IMAGE↓
3 | ↓
4 | 241206_100045 7.172 Rx FT8 27 0.1 1500 JA3GGJ/IMAGE↓
5 | ↓
6 | 241206_100345 7.172 Rx FT8 28 0.1 1500 JF3***/IMAGE↓
7 | ↓
8 | 241206_100645 7.172 Rx FT8 28 0.1 1500 JA4***/IMAGE↓
9 | ↓
10 | ↓
11 | ↓
12 | ↓
13 | ↓
14 | ↓
15 | ↓
16 | ↓
17 | 241206_100645 7.172 Rx FT8 28 0.1 1500 JA1***/
18 | ↓
19 | ↓

```

ファイル A



```

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
JA1*** http://****.****.jp/**/image1
JA2*** http://****.****.jp/image1
JA3GGJ http://radiotktk.php.xdomain.jp/EZPix/image1
.
.
.
JA9*** http://****.****.jp/**/image1
JA0*** http://****.****.jp/**/image1

```

ファイル B

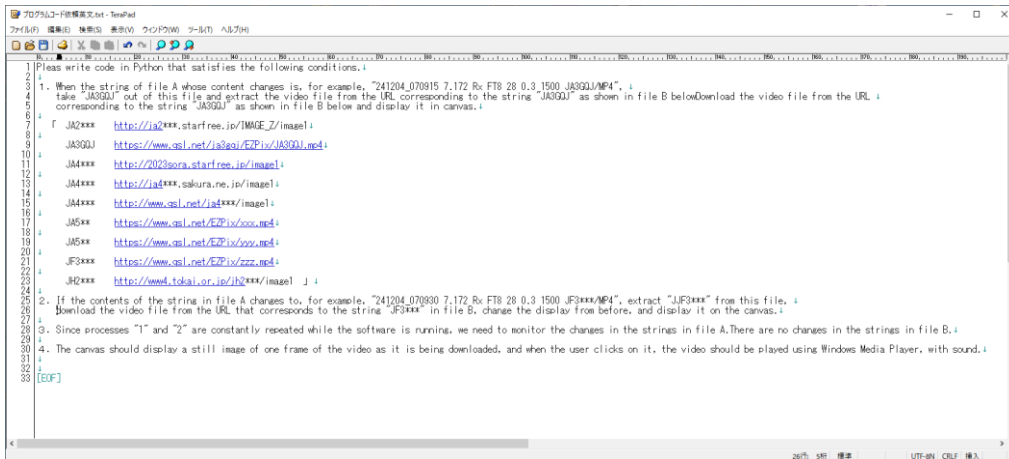
- 巻末 APPENDICES に ChatGPT に指示（依頼）をした内容の一部を残す。
- FT8 Image Transceiver のソースコードを別冊テキストファイルに残す。

APPENDICES

ChatGPTに指示（依頼）をした内容の一部

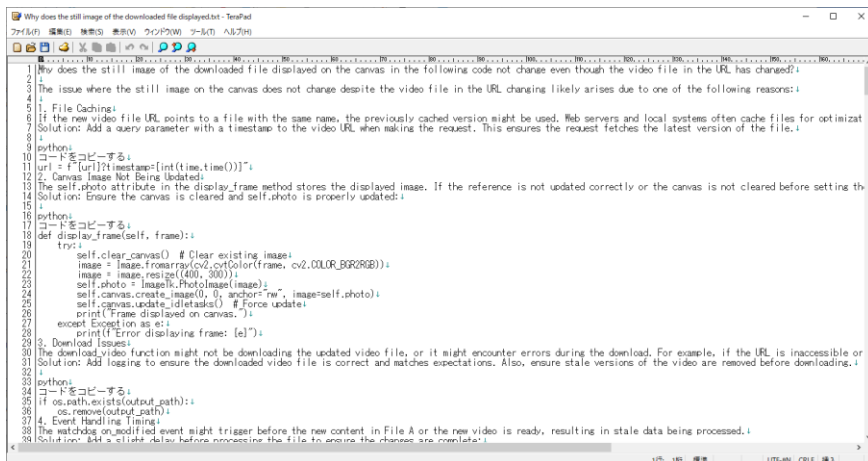
一回の指示（依頼）ではなかなか納得のいく回答は得られず何回も修正する指示（依頼）が必要であった。又、和文で指示（依頼）をするより英文で指示（依頼）したほうが返事の内容が希望する内容に近かったのは気のせいかな？

- Please write code in Python that satisfies the following conditions.



```
1 Please write code in Python that satisfies the following conditions.
2
3
4 1. When the string of file A whose content changes is, for example, "241204_070815 7.172 Rx FT8 28 0.3 1500 JA3GJU/MP4",
5 take "JA3GJU" out of this file and extract the video file from the URL corresponding to the string "JA3GJU" as shown in file B below and display it in canvas.
6
7 [ JA2*** http://ja2***.starfree.jp/IMAGE_Z/image1
8
9 JA3GJU https://www.osl.net/ja3gju/E2iv/JA3GJU.mp4
10
11 JA4*** http://2023sura.starfree.jp/image1
12
13 JA4*** http://ja4***.skura.ne.jp/image1
14
15 JA4*** http://www.osl.net/ja4***/image1
16
17 JA5*** https://www.osl.net/E2iv/xxx.mp4
18
19 JA5*** https://www.osl.net/E2iv/yyy.mp4
20
21 JF3*** https://www.osl.net/E2iv/zzz.mp4
22
23 JH2*** http://www.tokai.or.jp/jh2***/image1 ]
24
25 2. If the contents of the string in file A changes to, for example, "241204_070830 7.172 Rx FT8 28 0.3 1500 JF3***/MP4", extract "JF3***" from this file,
26 download the video file from the URL that corresponds to the string "JF3***" in file B, change the display from before, and display it on the canvas.
27
28 3. Since processes "1" and "2" are constantly repeated while the software is running, we need to monitor the changes in the strings in file A. There are no changes in the strings in file B.
29
30 4. The canvas should display a still image of one frame of the video as it is being downloaded, and when the user clicks on it, the video should be played using Windows Media Player, with sound.
31
32
33 [EOF]
```

- Why does the still image of the downloaded file displayed



```
1 Why does the still image of the downloaded file displayed on the canvas in the following code not change even though the video file in the URL has changed?
2
3 The issue where the still image on the canvas does not change despite the video file in the URL changing likely arises due to one of the following reasons:
4
5 1. File Caching
6 If the new video file URL points to a file with the same name, the previously cached version might be used. Web servers and local systems often cache files for optimization.
7 Solution: Add a query parameter with a timestamp to the video URL when making the request. This ensures the request fetches the latest version of the file.
8
9 python:
10 コードをコピーする
11 url = f"{url}?t={int(time.time())}"
12 Canvas Image Not Being Updated:
13 The self.photo attribute in the display_frame method stores the displayed image. If the reference is not updated correctly or the canvas is not cleared before setting the new image, the old image might persist.
14 Solution: Ensure the canvas is cleared and self.photo is properly updated.
15
16 python:
17 コードをコピーする
18 def display_frame(self, frame):
19     try:
20         self.clear_canvas() # Clear existing image
21         image = image.fromarray(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
22         image = image.resize((400, 300))
23         self.photo = imagek.PhotoImage(image)
24         self.canvas.create_image(0, 0, anchor='nw', image=self.photo)
25         self.canvas.update_idletasks() # Force update
26         print(f"Frame displayed on canvas.")
27     except Exception as e:
28         print(f"Error displaying frame: {e}")
29
30 3. Download Issues
31 The download_video function might not be downloading the updated video file, or it might encounter errors during the download. For example, if the URL is inaccessible or the file is too large.
32 Solution: Add logging to ensure the downloaded video file is correct and matches expectations. Also, ensure stale versions of the video are removed before downloading.
33
34 python:
35 コードをコピーする
36 if os.path.exists(output_path):
37     os.remove(output_path)
38 4. Event Handling Timing
39 The watches_on_modified event might trigger before the new content in File A or the new video is ready, resulting in stale data being processed.
40 Solution: Add a slight delay before processing the file to ensure the changes are complete.
```

- The image displayed on canvas2 in the following code with a problem is extracted from the call sign of the transmitting station from the string of a specific code transmitted by an amateur radio operator in WSJTX FT8 mode and try to download from the URL of the corresponding call sign in the local folder list, but it does not work well. Please fix it and write the full code.

```
修正情報1208.txt - TextPad
ファイル 編集 表示 実行 実行/停止 実行/再開 実行/終了
1 |
2 |
3 |
4 | The image displayed on canvas2 in the following code with a problem is extracted from the callisan of the transmitting station from the string of a specific code transmi
5 |
6 | The string of a specific code that an amateur radio operator transmits in FT8 mode of WSJT-X has the following format. The number of lines in the string is always chana
7 |
8 | 241208_100045 7.172 Rx FT8 27 0.1 1500 JA300J/IMAGE:
9 |
10 | 241208_100345 7.172 Rx FT8 28 0.1 1500 JF3xxx/IMAGE:
11 |
12 |
13 |
14 | 241208_100645 7.172 Rx FT8 28 0.1 1500 JA]xxx/IMAGE:
15 |
16 | -----
17 |
18 | *The format of the list in the local folder is as follows and will not change. This is file B.
19 |
20 | JA300J http://radiotk4k.che.xdomain.jp/57Piv/image!
21 |
22 | JA5xxx http://radiotk4k.che.xdomain.jp/!5aa/image!
23 |
24 | JA2xxx http://radiotk4k.che.xdomain.jp/5oo/image!
25 |
26 | JF3xxx http://radiotk4k.che.xdomain.jp/5oo/image!
27 |
28 |
29 | JA7xxx http://j4nrc.sakura.ne.jp/image!
30 |
31 | JA]xxx http://www.gsl.net/zzz/image!
32 |
33 | Defective code to be completed:
34 |
35 | import tkinter as tk
36 | from tkinter import filedialog, messagebox
37 | from ftplib import FTP
38 | import os
39 | import pickle
40 | from PIL import Image, ImageTk, UnidentifiedImageError
41 | import requests
42 | from watchdog.observers import Observer
43 | from watchdog.events import FileSystemEventHandler
44 |
45 | SETTINGS_FILE = "ftp_settings.pkl"
46 |
47 | # Global paths
48 | file_a_path = None
49 | file_b_path = None
50 |
51 | # Load settings or initialize empty settings
52 | def load_ftp_settings():
53 |     try:
54 |         if os.path.exists(SETTINGS_FILE):
55 |             with open(SETTINGS_FILE, 'rb') as f:
56 |                 settings = pickle.load(f)
57 |                 global file_a_path, file_b_path
58 |                 file_a_path = settings.get('file_a_path')
```

- 英文を使用するための翻訳ソフトはフリーの Windows 版「DeepL」を使用した。このソフトは翻訳したい文章（和文、英文どちらでも）を選択して Ctrl キーを押した状態でキー「C」を二回クリックするだけなので ChatGPT と簡単にチャットが出来るので大変便利である。Windows 版「DeepL」のダウンロード先を下記に記録しておく。

<https://www.deepl.com/ja/windows-app>