

FT8 Image Transceiver プログラム開発のバージョンアップ体験

Shu JA3GQJ

2024年12月「WSJTX」の FT8 で JPEG 画像を送受信する Python プログラムを ChatGPT に書いてもらった体験のメモを PDF ファイルに残した。

その後

- ① プログラムのタイトルとデジタル時計の表示
- ② 送信画像読み込み時間の表示
- ③ 画像受信時間と送信相手のコールサインの表示
- ④ FT8 の「コールサイン/IMAGE」, 「CQ コールサイン グリッドロケータ」、
「コールサイン/HYBRID」等コードの識別
- ⑤ 画像受信を知らせる wav サウンドの選択

など、気付いた点の追加を ChatGPT とやり取りして、次のような関数（モジュールとコンポーネント）でプログラムをバージョンアップした体験メモを PDF ファイルに残す。

設定管理:

save_settings:

初期設定で入力した FTP サーバーやパスのデータを保存する。

load_ftp_settings:

FTP サーバーやパスの設定を保存したファイルから読み込む。

ファイル操作:

extract_url_from_files:

WSJT ログファイル ALL.txt からコールサインを抽出し、関連する URL を取得する。

download_image_with_limit:

URL から受信画像をダウンロードし、C:\¥IMAGE¥downloaded_images に最大画像数を 30 枚に制限して保存する。

画像のアップロードと表示:

upload_image:

読んだ送信予定の画像を初期設定で入力（指定）した FTP サーバーにアップロードする。

display_image:

読んだ送信予定の画像を Tkinter の Canvas 上に表示する。

open_image_window:

受信画像をクリックしたら新しいウィンドウに拡大表示する。

画像受信を知らせるサウンド処理:

save_file_a & load_file_a:

画像受信を知らせるサウンドファイル名をテキストファイルに保存・読み込み。

on_select_sound_click:

サウンドファイルを選択し保存する GUI (4 ページの sound ファイル選択プルダウンメニュー) コンポーネント。

サウンドファイル再生 (wave & pyaudio) : on_modified でファイル変更イベント (画像受信) 時に再生。

リアルタイム監視:

FileAHandler:

WSJTX の log ファイル ALL.txt の変化の監視を行い、画像ダウンロードやサウンド再生をトリガーする。

watchdog モジュールで、WSJTX の log ファイル ALL.txt の変化を検知。

デジタル時計:

DigitalClock:

現在時刻を 1 秒ごとに更新し、ソフトのタイトルと一所に GUI (3 ページの図 1 に示すコントロールパネル) に表示。

WSJTX との連携:

launch_wsjtx:

WSJTX ソフトウェアを起動する。

send_keys & stop_tx:

WSJTX の送信許可キー、送信停止キーを送出する。

GUI (3 ページの図 1 のコントロールパネル) 操作:

ImageUploaderApp:

メインの GUI クラス (Python のプログラム用語)。画像アップロード、画像受信を知らせる wav サウンドの選択、

WSJTX 操作、ファイル監視などの操作を行う。

アプリケーション終了:

exit_software:

GUI (ソフト) 終了時に監視プロセスを停止し、コントロールパネル (3 ページの図 1) を閉じる。

動作の流れ

1. ソフト起動時、FTP 設定データをロードし、ファイル監視 (watchdog) を開始する。
2. GUI (デスクトップに表示される図 1 のコントロールパネル) で次の操作を行う。
 - ・画像のローカル表示 (load_local_image)。
 - ・画像の FTP アップロード (upload_image 関数で送信用画像, upload_digisites_image 関数で受信画像をそれぞれ FTP の Directory にアップロード)。
 - ・画像受信を知らせるサウンドファイルの選択と保存。
 - ・WSJTX の起動およびキー操作。
 - ・ファイル監視イベント (on_modified) により、FT8 が新規コードを受信した時に相手局の URL から画像をダウンロードして Tkinter の Canvas に表示して wav サウンドを再生して画像の受信を知らせる。
 - ・アプリ終了時に全リソースを解放 (exit_software)。

1. FT8 Image Transceiver- ver.05-01-05

Python のファイル名は「ft8imgtxrx.py」、ソフト名は FT8 Image Transceiver でバージョンは ver.05-01-05 とする。図 1-1 がソフトのコントロールパネルである。

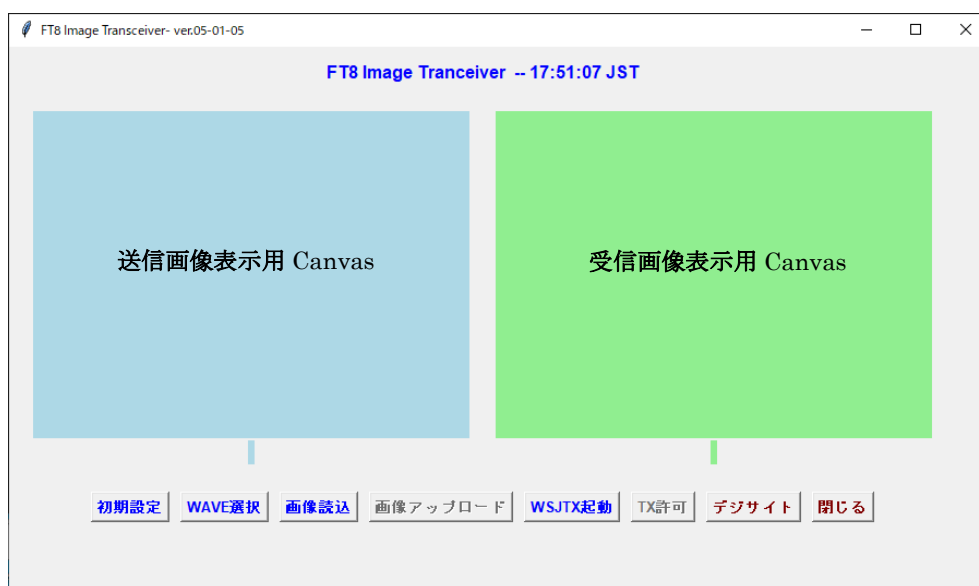


図 1-1

Tkinter の画面左に送信画像表示用 Canvas、右に受信画像表示用 Canvas、各 Canvas の下に八つのボタン（初期設定、WAVE 選択、画像読込、画像アップロード、WSJTX 起動、TX 許可、デジサイト、閉じる）を配置した。

・「初期設定」ボタン

ボタンをクリックすると図1-2の画面がポップアップ →

- ① FTP Server, Username, Password, Directory (送信画像アップロード用 ftp サーバーの情報)
- ② File A Path と File B Path (ファイル A (FT8 受信コードの log(ALL.txt)) とファイル B (送信局の送信画像がある URL のリスト) のディレクトリーの情報)
- ③ Local file Path (デジサイトにアップロードする受信画像があるフォルダーのディレクトリーの情報)
- ④ DigiSites FTP Server, DigiSites Username, DigiSites Password, DigiSites Directory (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)

のデータを入力 → 「Save」ボタンをクリックして保存。

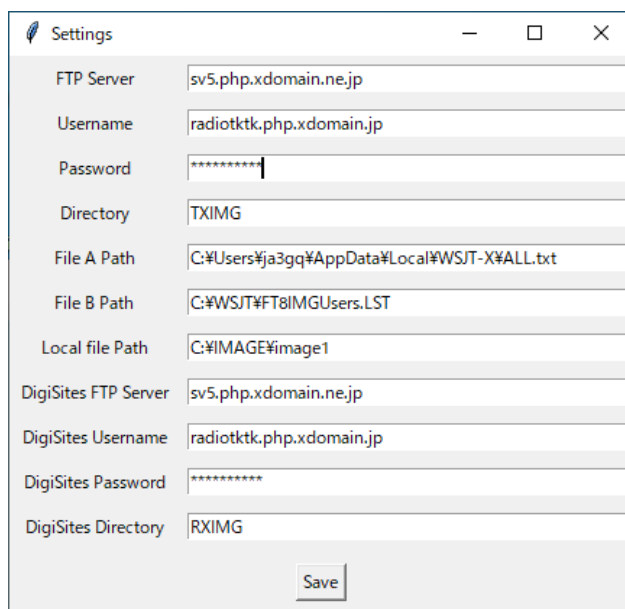


図1-2

入力データ (情報) は「FT8 Image Transceiver- ver.05-01-05」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開発ソフト専用で送信画像用を「TXIMG」、デジサイト用を「RXIMG」にした。

・「WAVE 選択」ボタン

ボタンをクリック → wave ファイル選択画面 (5 ページ図1-3) が開く
→ リストを表示 → 選択して「保存」ボタンクリックで閉じる。

リストは別途 C:\¥IMAGE に自分で選んだ曲に wave1.wav, wave 2.wav, wave 3.wav のファイル名をつけて保存しておく。ここで選択した wave ファイル(例えば sound1) は、別途 C:\¥wavfile¥file_a.txt に選択した名前(例えば sound1) だけが保存され、次回から自動で読み込まれて画像を受信したことを知らせる。フォルダー wavfile はプログラムが自動生成する。

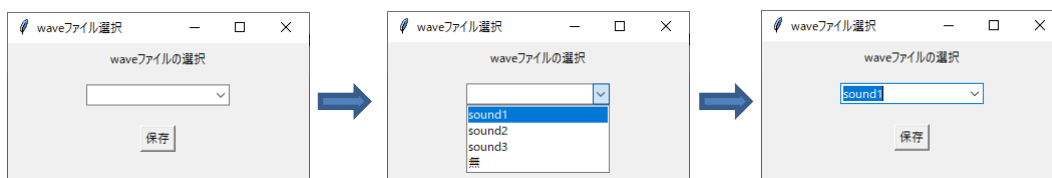


図 1-3

・「画像読込」ボタン

ボタンをクリック → 送信画像保存フォルダーを開く(図 1-4) → 送信する画像を選択してダブルクリック → 画像が Canvas に表示(図 1-5)される → 「画像アップロード」ボタンが有効になる。

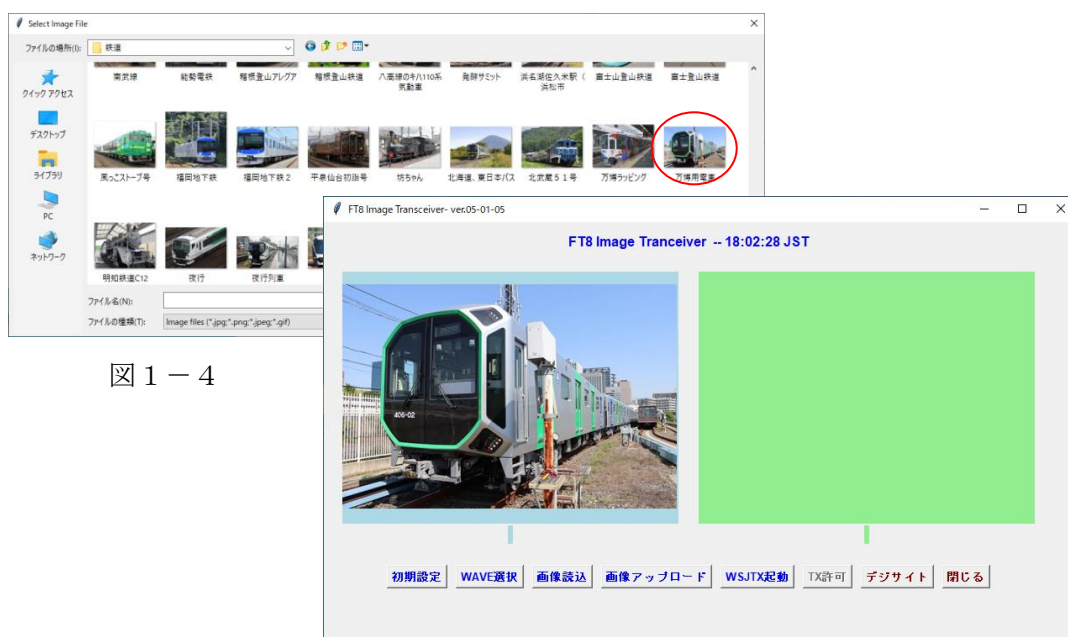


図 1-4

図 1-5

・「画像アップロード」ボタン

「画像読込」ボタンで読込んだ送信画像が Canvas に表示されると、このボタンが有効になる → ボタンをクリック → 画像のファイル名が image1 になって ftp サーバーのフォルダー「TXIMG」にアップロードされる。アップロードが終わるとその時の時間がキャンバスに表示される(6 ページ・図 1-6)。



図 1-6

・「WSJTX 起動」 ボタン

ボタンをクリック → **WSJTX が起動する** (図 1-7)。

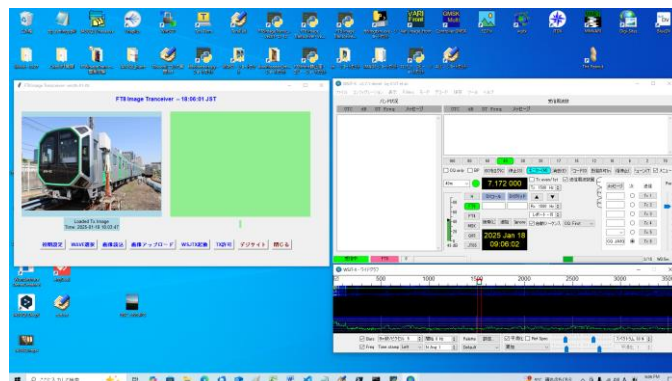


図 1-7

・「TX 許可」 ボタン

送信画像読み込み終了 → **WSJT-X の起動を確認** → このボタンが有効になる
 → ボタンをクリック → **WSJT-X のキーボード Alt+N に相当する信号が送られる**
 → 「送信許可(n)」ボタンが **On (赤色)** になり → 定時に送信が始まる。
WSJT-X が起動中、キーボードのファンクションキー **F3** をクリックしたら **WSJT-X の**
キーボード-ショートカットリスト (巻末に記録しておく) が表示される。

・「デジサイト」 ボタン

ボタンをクリック → 受信画像がデジサイトのフォルダーにアップロードされる。
 既存のデジサイトにファイル名 **image1** でアップロード出来た。しかし、アップロード
 するたびに上書きされ、送信局のコールサインが表示されない。

- ・「閉じる」ボタン

ボタンをクリック → 「FT8 Image Transceiver- ver.05-01-05」 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 1-8)



図 1-8

2. 開発した Python プログラムを EXE ファイル形式に変換

- ・ Python がインストールされていない環境でも開発したプログラムが使用できるように Windows10 用の実行ファイルである EXE ファイル形式に変換してみた。仲間に配布したら使ってもらえるかも・・・。
- ・ 実行ファイルに変換するツールがいろいろある中で「PyInstaller」をインストールして使おうとしたが何故か変換ができない問題にぶつかった。
- ・ そこで、ChatGPT に相談したら Python のバージョンに関係があるのではないかと、いうので最新バージョンの 3.13.1 から 3.10.0 にバージョンダウンをしたら次の手順で簡単に変換できた。

Windows10 デスクトップ左下の検索欄 (図 2-1) に「command」と入力 →

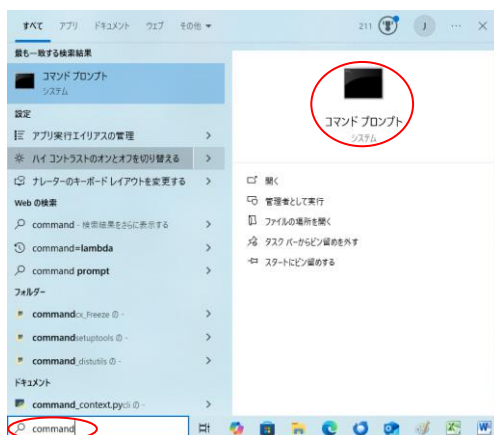


図 2-1

コマンド プロンプト (図 2-2) を開くコマンド →

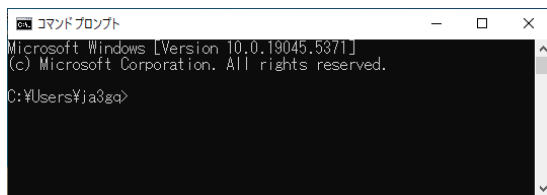


図 2-2

→ ディレクトリーを ft8imgtxrx.py がある C:\¥IMAGE に変更する (図 2-3)

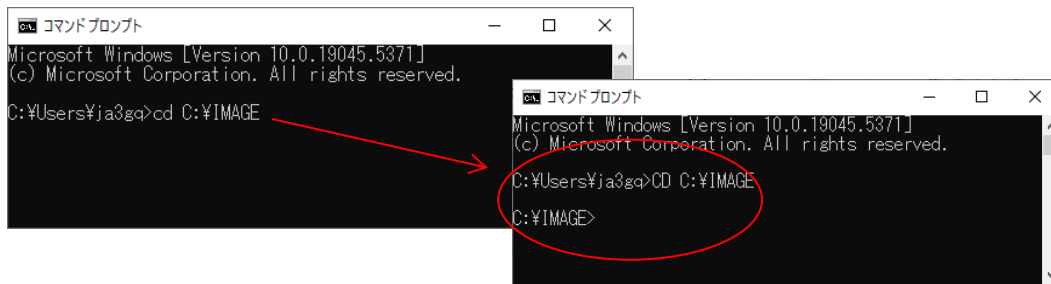


図 2-3

→ 「pyinstaller --onefile --noconsole ft8imgtxrx.py」と入力する (図 2-4) →

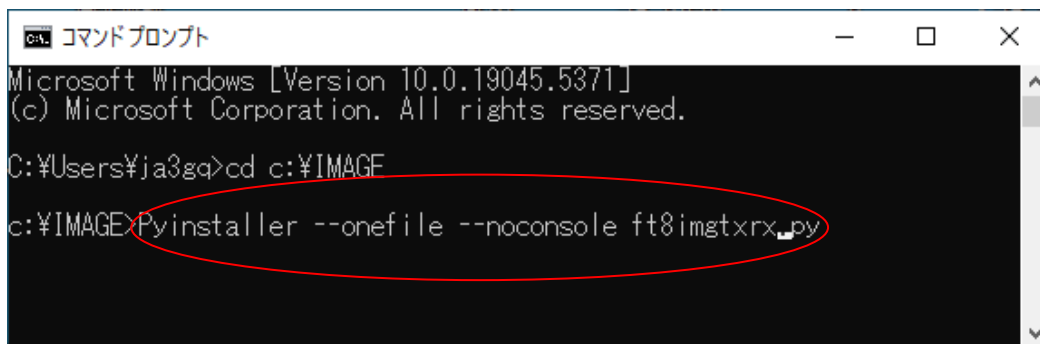


図 2-4

コマンド プロンプトの画面に変換過程が表示される → 最後に「65960 INFO: Building EXE from EXE-00.toc completed successfully.」の文字が出て変換完了 (図 2-5) →

→ 右上の X をクリック → コマンド プロンプトを閉じる →

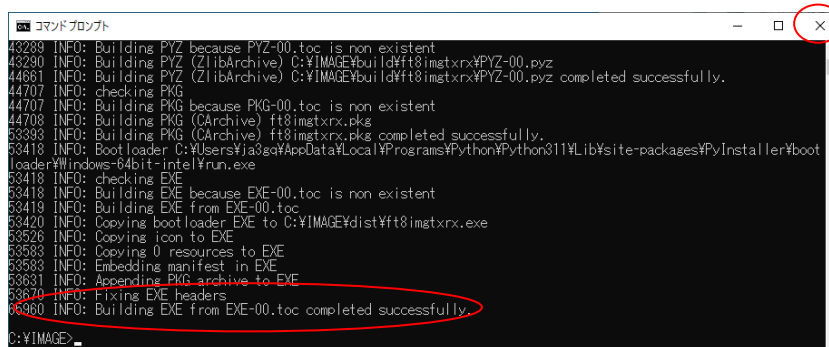


図 2-5

→ フォルダーC:\¥IMAGE にフォルダーbuild および dist とテキストファイルft8imgtxrx.spec が生成された (図 2-6)。 →

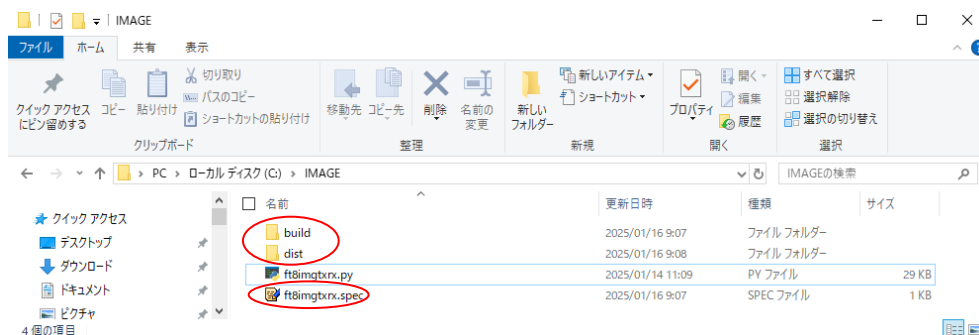


図 2-6

→ フォルダーdist を開く → 実行形式のft8imgtxrx.exe がある (図 2-7) →

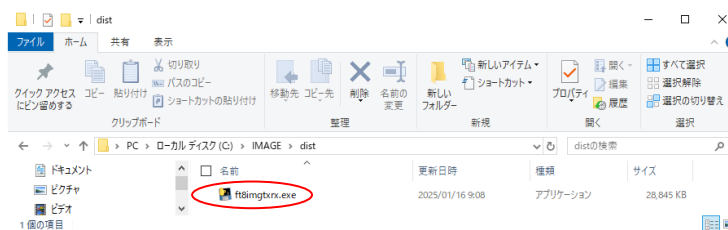


図 2-7

以上変換の手順を記録したが希望する仲間にはft8imgtxrx.exeだけを配布する → 仲間は自分のパソコンのCドライブにフォルダーIMAGE(図 2-8)を作る →

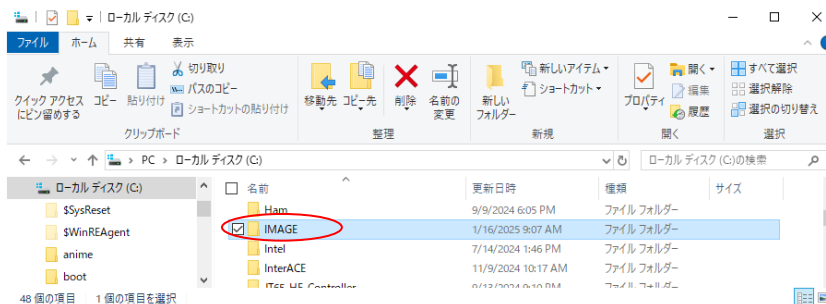


図 2-8

→ そのフォルダーにft8imgtxrx.exeを保存(図 2-9)する。 →

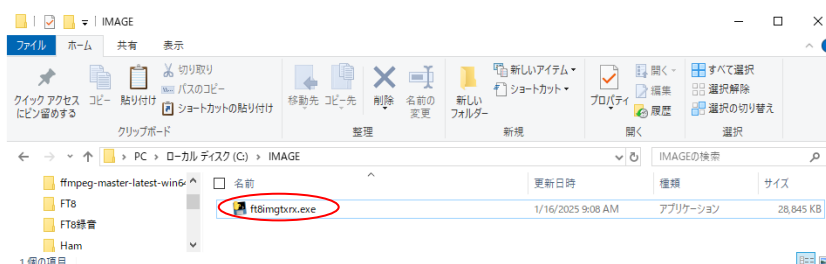


図 2-9

→ ft8imgtxrx.exe をクリックしたらコントロール画面 (図 2-10) が表示される →

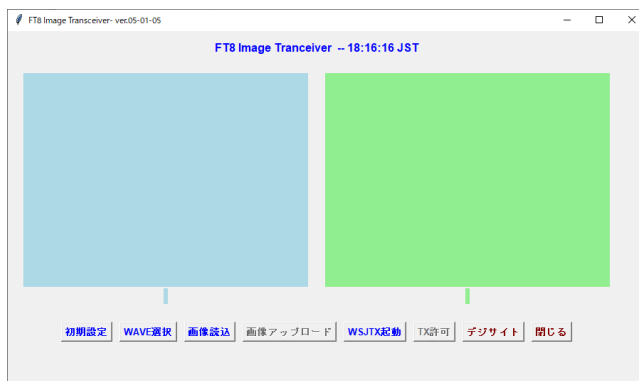


図 2-10

→ 仲間は 3 ページから 7 ページの項目を参考にして使用を開始するであろう。

3. 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 FT8 受信コードの log (ALL.txt) は WSJTX のファイルタグの「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フォルダー (図 3-1) がある。この log (ALL.txt) が開発ソフトのファイル A である。

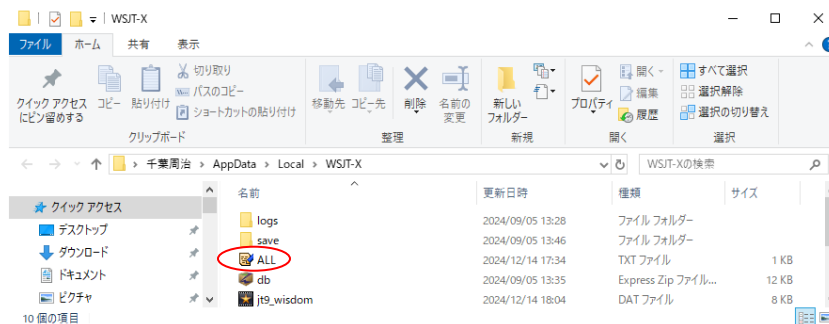


図 3-1

- 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時間がかかった。
- WSJTX を単独で使う場合はキーボードの「Alt + N」キーを押せば WSJTX の送信許可 (n) が On なるが、開発ソフト側から単純に、

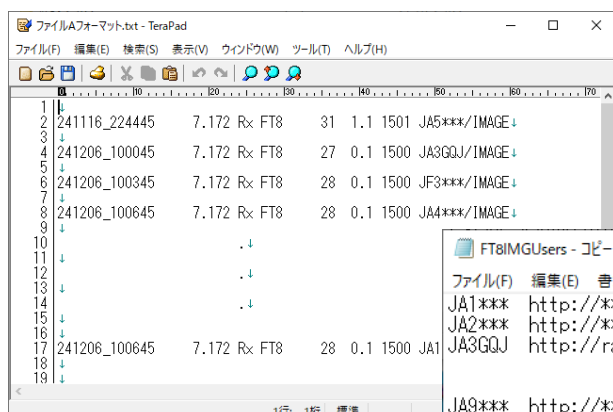
```
WSJTX_PATH = r" C:\WSJT\wsjtx\bin\wsjtx.exe "  
subprocess.Popen([WSJTX_PATH]) WSJTX
```

といったコマンドで WSJTX を起動して、コントロールパネルの「TX Permission」ボタンで「Alt + N」キーを送っても、カーソルがコントロールパネル側に移って

しまつて送信許可 (n) ボタンは WSJT-X を単独で使う時のように On (有効) にならない。従つて、コントロールパネルの「WSJTX 起動」ボタンで WSJT-X を起動するとき、カーソルをコントロールパネルの「TX 許可」ボタンの上に移動しても WSJT-X 側の送信許可 (n) が On なるようにする必要があつた。

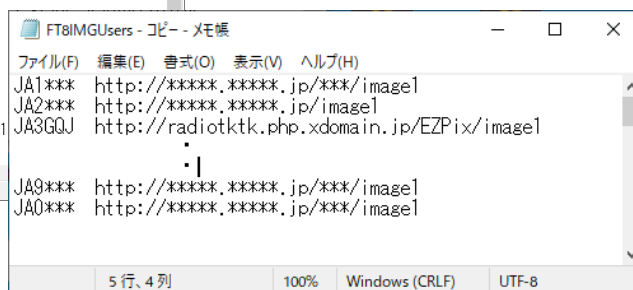
ファンクションキーF8で開く WSJT-X のキーボードショートカットのリストを巻末の APPENDIX に記録しておく。

- 又、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る。このプロセスを一回の送信で終わらせる必要もあつた。
- これらの事柄を実現するため、手を変え、品を変えて ChatGPT に指示 (依頼) をして、最終的に時計を作つて「Tx 許可」ボタンをクリックしたら毎時 14 秒毎の時間を時計から検出して WSJT-X が送信を開始する 1 秒前に「Alt + N」キーを送つて「送信許可 (n)」ボタンが On (有効) になるようにした。
- そして、「Alt + N」キーを送つた 13 秒後に「Alt + H」キーを送つて WSJT-X の「送信停止 (h)」ボタンを On (有効) して送信を停止させ、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」を作つて開発したソフトを置く。
- 下はファイル A (FT8 受信コードの log (ALL.txt)) とファイル B のテキストファイルのフォーマットである。



```
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



```
JA1*** http://****.****.jp/**/image1
JA2*** http://****.****.jp/image1
JA3GGJ http://radiotktp.php.xdomain.jp/EZPix/image1
.
.
JA9*** http://****.****.jp/**/image1
JA0*** http://****.****.jp/**/image1
```

ファイル B

- 無料で使用してきた ChatGPT を 2024 年 12 月 US\$20(換算レート / 154 円)の有料に切り替えた。
- 巻末 APPENDIX に ChatGPT に指示 (依頼) した内容の一部の記録を残す。

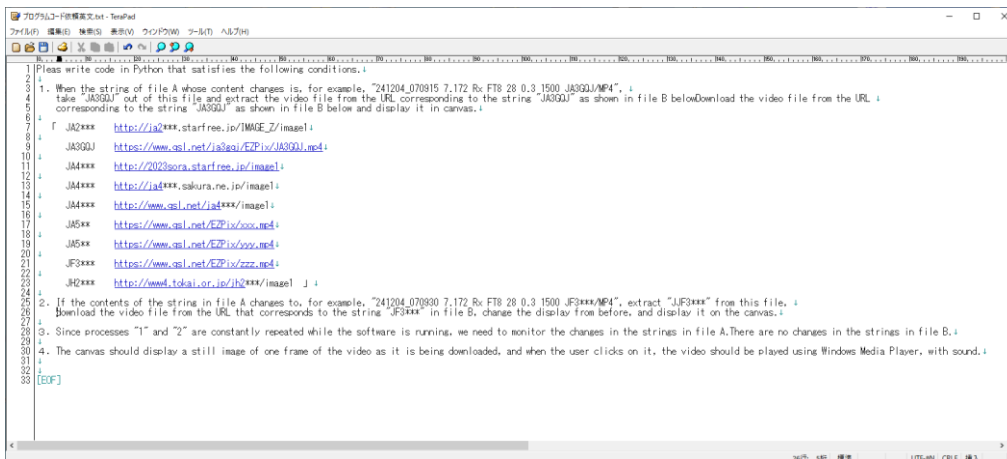
As of 16 January 2025

APPENDIX

1. ChatGPTに指示（依頼）した内容の一部

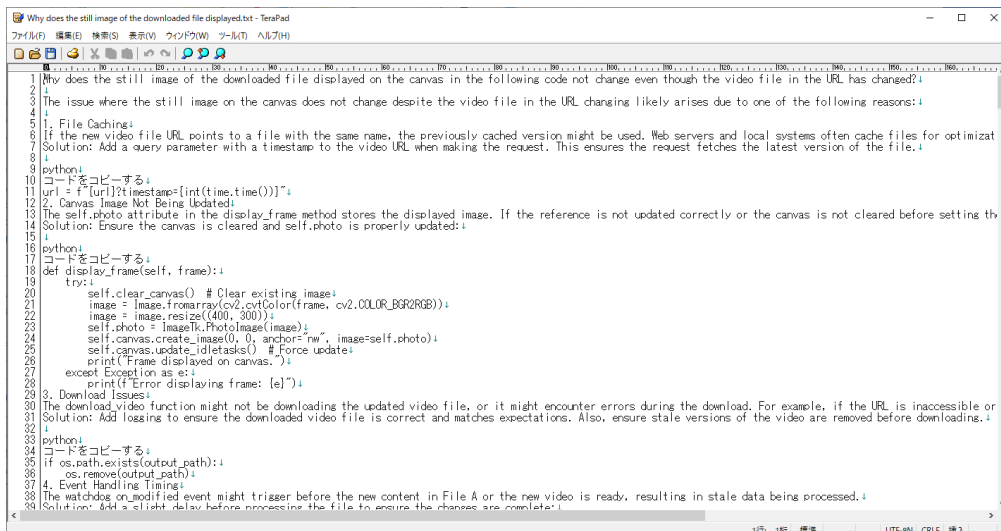
一回の指示（依頼）ではなかなか納得のいく回答は得られず何回も修正する指示（依頼）が必要であった。又、ChatGPTを有料使用にして、指示（依頼）も和文ではなく英文のほうが、希望する内容に近い返事が返って来たような気がする??

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



```
1 Please write code in Python that satisfies the following conditions.
2
3 1. When the string of file A whose content changes is, for example, "241204_070830 7.172.Rx FT8 28 0.3 1500 JA3GJ/MP4",
4 take "JA3GJ" out of this file and extract the video file from the URL corresponding to the string "JA3GJ" as shown in file B below and display it in canvas.
5
6 | JA2*** http://ja2***.starfree.jp/image1
7 |
8 | JA3GJ http://www.osl.net/ja3gai/E2Piv/JA3GJ.mp4
9 |
10 | JA4*** http://2023sora.starfree.jp/image1
11 |
12 | JA4*** http://ja4***.skura.ne.jp/image1
13 |
14 | JA4*** http://www.osl.net/ja4***/image1
15 |
16 | JA5** http://www.osl.net/E2Piv/xxx.mp4
17 |
18 | JA5** http://www.osl.net/E2Piv/yyy.mp4
19 |
20 | JF3*** http://www.osl.net/E2Piv/zzz.mp4
21 |
22 | JH2*** http://www.tokai.or.jp/jh2***/image1
23 |
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 [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 does 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}?timestamp={int(time.time())}"
12 2. 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
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 = ImageTk.PhotoImage(image)
24         self.canvas.create_image(0, 0, anchor='nw', image=self.photo)
25         self.canvas.update_idletasks() # Force update
26         print("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
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 watchdog 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 WSJT-X 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.

```

1 The image displayed on canvas, in the following code with a problem is extracted from the call sign of the transmitting station from the string of a specific code frame
2
3 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 chane
4 241208_100845 7.172 R: FT8 27 0.1 1500 JASGGL/DKBE:
5
6 7 241208_100845 7.172 R: FT8 28 0.1 1500 JF3KX4/DKBE:
7
8 241208_100845 7.172 R: FT8 28 0.1 1500 JASKX4/DKBE:
9
10
11
12
13 241208_100845 7.172 R: FT8 28 0.1 1500 JASKX4/DKBE:
14
15
16
17
18
19
20 The format of the list in the local folder is as follows and will not change. This is file B.:
21
22 JASGGL http://radio-hobby.4dnet.ne.jp/FT8/1/msgel!
23
24 JASKX4 http://radio-hobby.4dnet.ne.jp/FT8/1/msgel!
25
26 JF3KX4 http://radio-hobby.4dnet.ne.jp/FT8/1/msgel!
27
28
29 JF3KX4 http://radio-hobby.4dnet.ne.jp/FT8/1/msgel!
30
31
32 JASKX4 http://radio-hobby.4dnet.ne.jp/FT8/1/msgel!
33
34 JASKX4 http://www.on1.net/ftp/msgel!
35
36
37
38
39
40
41
42
43 Defective code to be completed*****
44
45
46 report B:enter as S1:
47 from B:enter report @!dialos, messagebox:
48 from HoldB report FT8:
49 report S0:
50 report S0:
51 from PL import Image, ImageTk, UnidentifedImageError:
52 report S0:
53 from watchdog.observers import Observer:
54 from watchdog.events import FileSystemEventHandler:
55
56 SETTINGS_FILE = "fto_settings.txt":
57
58 # Global method:
59 file_s_path = None:
60 file_b_path = None:
61
62 # Load settings or initialize empty settings:
63 def load_fto_settings():
64     try:
65         if os.path.exists(SETTINGS_FILE):
66             with open(SETTINGS_FILE, "r") as f:
67                 settings = pickle.load(f)
68                 if not file_s_path:
69                     file_s_path = settings["file_s_path"]
70                 if not file_b_path:
71                     file_b_path = settings["file_b_path"]
72

```

- 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）を選択してキーボードの 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>

2. WSJT-X キーボードショートカットキーリスト

WSJT-X - キーボードショートカット	
Esc	Stop Tx, abort QSO, clear next-call queue
F1	Online User's Guide (Alt: transmit Tx6)
Shift+F1	Copyright Notice
Ctrl+F1	About WSJT-X
F2	Open settings window (Alt: transmit Tx2)
F3	Display keyboard shortcuts (Alt: transmit Tx3)
F4	Clear DX Call, DX Grid, Tx messages 1-4 (Alt: transmit Tx4)
Alt+F4	Exit program
F5	Display special mouse commands (Alt: transmit Tx5)
F6	Open next file in directory (Alt: toggle "Call 1st")
Shift+F6	Decode all remaining files in directory
F7	Display Message Averaging window
F11	Move Rx frequency down 1 Hz
Ctrl+F11	Move identical Rx and Tx frequencies down 1 Hz
Shift+F11	Move Tx frequency down 60 Hz (FT8) or 90 Hz (FT4)
Ctrl+Shift+F11	Move dial frequency down 1000 Hz
F12	Move Rx frequency up 1 Hz
Ctrl+F12	Move identical Rx and Tx frequencies up 1 Hz
Shift+F12	Move Tx frequency up 60 Hz (FT8) or 90 Hz (FT4)
Ctrl+Shift+F12	Move dial frequency up 1000 Hz
Alt+1-6	Set now transmission to this number on Tab 1
Ctrl+1-6	Set next transmission to this number on Tab 1
Alt+A	Clear Active Stations for QMAP
Alt+B	Toggle "Best S+P" status
Alt+C	Toggle "Call 1st" checkbox
Alt+D	Decode again at QSO frequency
Shift+D	Full decode (both windows)
Ctrl+E	Turn on TX even/1st
Shift+E	Turn off TX even/1st
Alt+E	Erase
Ctrl+F	Edit the free text message box
Alt+G	Generate standard messages
Alt+H	Halt Tx
Ctrl+I	Add Dx Call to the Ignore List
Ctrl+L	Lookup callsign in database, generate standard messages
Alt+M	Monitor
Alt+N	Toggle "Enable Tx"
Ctrl+O	Open a .wav file
Alt+O	Change operator
Alt+Q	Open "Loc QSO" window
Ctrl+R	Set Tx4 message to RRRR (not in FT4)
Alt+R	Set Tx4 message to RRR3
Ctrl+Shift+R	Refresh Active Stations window
Alt+S	Stop monitoring
Alt+T	Toggle Tune status
Alt+Z	Clear hung decoder status

以上