

Utiliser l'IA pour piloter facilement son poste radio

Jean-Luc Levant – F4GSC

octobre2026

Face à la complexité et au manque de convivialité de certains postes, il est souvent nécessaire de ressortir leur documentation pour modifier le moindre réglage. Leur utilisation devient alors fastidieuse, voire rebutante.

Le but de cet article est de montrer qu'il est possible d'améliorer l'ergonomie d'un poste en développant de petites interfaces dédiées à nos utilisations spécifiques.

Pour cela, nous allons utiliser deux outils simples d'emploi pour les OM non spécialistes de la programmation et de l'informatique (Claude Opus 5.5⁽¹⁾ et Thonny 5.0.0⁽²⁾). Le langage Python est aussi utilisé, mais il n'a pas besoin d'être maîtrisé : cette compétence est confiée à Claude, et reste transparente pour l'OM non rompu à ce langage.

Utiliser une IA pour améliorer le pilotage de nos postes radio

À l'occasion d'un déménagement, j'ai ressorti de mes cartons un Yaesu FT-897, qui couvre les bandes HF, VHF et UHF. C'est un très bon poste, mais d'une technologie déjà ancienne.



Figure 1. J'utilise le Yaesu FT-897.

Pour le configurer, nous disposons d'un petit écran vert de 5 cm par 3 cm et d'un jeu de touches. J'avoue que ce n'est guère ergonomique, sauf à l'utiliser tous les jours. Au bout de quelques heures, j'ai décidé de développer des applications logicielles qui correspondent à mon utilisation.

J'ai confié à Claude le développement des programmes, écrits en Python. Leur exécution est assurée par Thonny, un environnement d'exécution de scripts Python.

En quelques minutes, j'ai pu mettre en œuvre un programme de base qui m'a permis de configurer mon poste depuis la console de Thonny : fréquence, décalage de fréquence, CTCSS et mode de transmission (LSB, USB, FM et CW).

Par la suite, j'ai ajouté les fonctions suivantes :

- la lecture et l'affichage du S-mètre en fonction du temps ;
- la lecture de la puissance émise ;

- un analyseur de spectre ;
- un scanner qui affiche le spectre d'une plage de fréquences, avec la possibilité de s'arrêter automatiquement dès qu'il rencontre une station ;
- ...

Les applications sont infinies : elles permettent d'automatiser des fonctions et, finalement, de tirer le meilleur parti de son poste. Le pilotage depuis la face avant reste bien sûr possible.

De quoi avons-nous besoin ?

En réalité, il faut peu de choses :

- un PC ;
- un transceiver muni d'une interface CAT ;
- une interface CAT vers USB (câble) ou vers Bluetooth (j'utilise le Bluetooth) ;
- une IA (Claude) ;
- un environnement d'exécution pour le script Python généré par l'IA (Thonny).

Je précise encore une fois qu'il n'est pas nécessaire de maîtriser le langage Python (je ne le maîtrise pas moi-même), car je sous-traite cette compétence à Claude. De même, il n'est pas nécessaire de connaître Thonny sur le bout des doigts. Il reçoit le programme généré par l'IA par un simple copier (Claude) / coller (Thonny), et un bouton permet d'en lancer l'exécution. Il faut juste savoir installer dans Thonny les modules logiciels qui assurent les fonctions de communication, de calcul, d'affichage... J'explique ce point plus loin.

Interface PC vers le transceiver

Le FT-897 possède un connecteur mini-DIN 8 broches (prise CAT/LINEAR) dédié à l'interface CAT. Plusieurs solutions existent. Pour limiter le nombre de câbles, j'ai choisi une interface Bluetooth qui se branche directement sur ce connecteur. Bien sûr, un câble, la solution historique, peut aussi être utilisé.

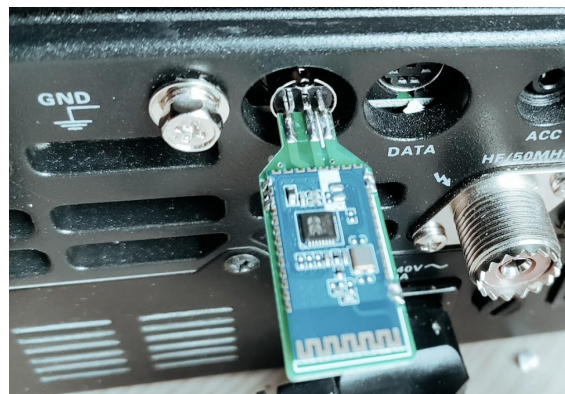


Figure 2. Interface Bluetooth permettant la communication entre le FT-897 et le PC.

L'appairage Bluetooth avec le PC se fait de façon classique et crée un port COM virtuel (chez moi COM8). C'est par ce port que transiteront les informations échangées avec les différentes applications Python que nous allons créer. La vitesse par défaut du CAT est de 9600 bauds. Il faudra veiller à ce que le FT-897 soit bien réglé sur cette vitesse (menu 019, accessible par un appui long sur la touche F du FT-897).

L'interface CAT/Bluetooth se trouve sur les sites marchands habituels pour moins de 20 €.

CAT TO Bluetooth



Figure 3. Modèle de l'interface CAT utilisée.

L'environnement d'exécution Thonny

C'est lui qui recevra les différents programmes Python générés par l'IA et qui en assurera l'exécution. Mais avant de lancer un programme, il faudra installer dans Thonny tous les modules nécessaires à son fonctionnement. Un module regroupe des fonctions logicielles écrites et validées par d'autres développeurs (sys, time, pyserial...).

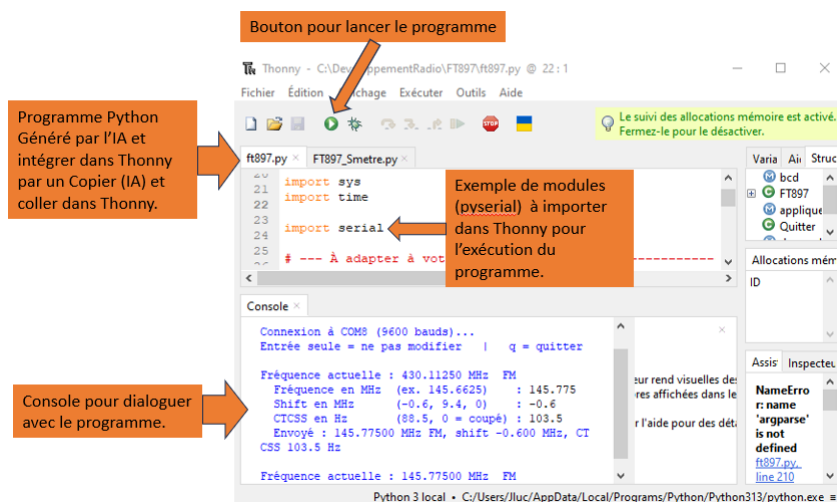


Figure 4. Environnement d'exécution de Thonny.

L'installation de ces modules est simple à partir de la console de Thonny. Il suffit d'y entrer la commande "pip install pyserial", et de la répéter pour chaque module nécessaire. Cela n'est à faire qu'une seule fois, car ces modules restent installés dans le dossier de Python. Il n'est pas nécessaire d'installer les modules "argparse", "sys" et "time", car ils sont déjà intégrés à Python.

```
C:\DeveloppementRadio\FT897>pip install pyserial
Collecting pyserial
  Using cached pyserial-3.5-py2.py3-none-any.whl.metadata (1.6 kB)
  Using cached pyserial-3.5-py2.py3-none-any.whl (90 kB)
Installing collected packages: pyserial
Successfully installed pyserial-3.5
C:\DeveloppementRadio\FT897>
```

Figure 5. Exemple de commande "pip install" pour installer le module "pyserial" qui permet la communication par le port virtuel COMx.

En cas d'erreur lors de l'exécution, un message apparaît dans la console. Il suffit alors de copier ce message et de le coller dans le prompt de Claude, qui indiquera l'origine du problème et la façon de le résoudre.

Choix de l'IA

Toutes les IA ne se valent pas et n'offrent pas les mêmes performances en matière de production de code. Personnellement, j'utilise Claude Opus, mais d'autres IA peuvent être testées.

Fonctionnalités offertes par le CAT

Nous allons demander à Claude de nous décrire les fonctionnalités prises en charge par l'interface CAT. Entrons dans la fenêtre de saisie de Claude le prompt suivant :

quelles sont les possibilités du CAT

Après quelques minutes de réflexion, Claude fournit la réponse détaillée ci-dessous sur les possibilités du CAT.

Fonction	Ce que le CAT permet
Fréquence	Régler et lire la fréquence du VFO
Mode	Régler et lire le mode (LSB, USB, CW, AM, FM, DIG, PKT...)
Shift relais	Régler le sens (+, -, simplex) et la valeur
CTCSS	Activer ou couper, choisir la tonalité
DCS	Activer ou couper, choisir le code
État en réception	Lire le S-mètre et l'état du squelch
État en émission	Lire le wattmètre, le PTT, l'alerte ROS, le split
Émission (PTT)	Passer en émission et revenir en réception
VFO A/B	Basculer d'un VFO à l'autre
Split	Activer ou couper
Clarifieur (RIT)	Activer ou couper, régler le décalage
Verrouillage	Verrouiller ou libérer la face avant

Mais aussi sur ce qu'il ne sait pas faire.

Fonction	Ce que le CAT ne sait pas faire
Mémoires	Ni écrire, ni rappeler, ni effacer un canal
Réglages du poste	Puissance, volume, squelch, filtres, pas d'accord, menus
Relecture	Le shift, la tonalité CTCSS et le code DCS se règlent mais ne se relisent pas

Pilotage de base du FT-897

L'objectif est de configurer le poste pour trafiquer via un relais en FM. Pour cela, nous allons demander à Claude d'écrire un programme qui règle, depuis la console de Thonny, la fréquence, le décalage en fréquence et le CTCSS. Le prompt suivant est donc fourni à l'IA.

j'utiliser un transceiver yeasu FT-897, avec une interface COM8 (Bluetooth). Je souhaite développer un programme python qui me permet de configurer la fréquence, le shift, le CTCSS par l'interface du protocole CAT

Pour ne pas allonger l'article, je ne reproduis, à titre d'information, que les premières lignes du programme généré par l'IA. Le programme complet peut être téléchargé depuis le site de l'ARML 49 ⁽³⁾.

```
#!/usr/bin/env python3
"""
Piloteage CAT du Yaesu FT-897 : fréquence, shift relais, CTCSS.

Dépendance : pip install pyserial

Mode interactif (saisie dans la console) : lancer le script sans réglage
python ft897_cat.py
python ft897_cat.py --port COM13 --vitesse 9600

Mode direct (une seule commande) :
python ft897_cat.py --lire
python ft897_cat.py --freq 145.6625 --shift -0.6 --ctcss 88.5
python ft897_cat.py --freq 430.250 --shift 9.4 --ctcss 71.9 --tsq

Protocole : chaque commande fait 5 octets -> P1 P2 P3 P4 CMD
Liaison : 8 bits, sans parité, 2 bits de stop, vitesse = menu No-019 (CAT RATE)
"""

import argparse
import sys
import time

import serial

# --- À adapter à votre installation -----
PORT_DEFAULT = "COM8" # port Bluetooth « sortant » du BT04-A
VITESSE_DEFAULT = 9600 # identique au menu No-019 (CAT RATE) et au module

# --- Opcodes CAT (5e octet) -----
CMD_SET_FREQ = 0x01 # P1-P4 : fréquence BCD, pas de 10 Hz
CMD_READ_FREQ = 0x03 # réponse : 4 octets BCD + 1 octet mode
CMD_SET_MODE = 0x07 # P1 : mode
CMD_SHIFT_DIR = 0x09 # P1 : sens du shift
CMD_TONE_MODE = 0x0A # P1 : type de squelch codé
CMD_CTCSS_FREQ = 0x0B # P1-P2 : tonalité TX, P3-P4 : tonalité RX (BCD, 0,1 Hz)
CMD_SHIFT_FREQ = 0xF9 # P1-P4 : décalage relais BCD

MODES = {"LSB": 0x00, "USB": 0x01, "CW": 0x02, "CWR": 0x03, "AM": 0x04,
         "FM": 0x08, "DIG": 0x0A, "PKT": 0x0C}
MODES_INV = {v: k for k, v in MODES.items()} | {0x06: "WFM", 0x88: "FM-N"}

SHIFT_MOINS, SHIFT_PLUS, SHIFT_SIMPLEX = 0x09, 0x49, 0x89

TONE_CTCSS_ENC_DEC = 0x2A # tonalité à l'émission + squelch à la réception (TSQ)
TONE_CTCSS_ENC = 0x4A # tonalité à l'émission seulement (cas usuel des relais)
TONE_OFF = 0x8A

CTCSS_VALIDES = (
    67.0, 69.3, 71.9, 74.4, 77.0, 79.7, 82.5, 85.4, 88.5, 91.5, 94.8, 97.4,
    100.0, 103.5, 107.2, 110.9, 114.8, 118.8, 123.0, 127.3, 131.8, 136.5,
    141.3, 146.2, 151.4, 156.7, 159.8, 162.2, 165.5, 167.9, 171.3, 173.8,
    177.3, 179.9, 183.5, 186.2, 189.9, 192.8, 196.6, 199.5, 203.5, 206.5,
    210.7, 218.1, 225.7, 229.1, 233.6, 241.8, 250.3, 254.1,
)
```

Il n'est pas nécessaire de comprendre ce programme, car l'objectif est d'obtenir des fonctionnalités et d'utiliser au mieux son poste. À titre d'exemple, on peut y voir :

- les codes des commandes CAT, sous la ligne "# --- Opcodes CAT (5e octet) ---" ;
- un tableau recensant tous les codes CTCSS ("CTCSS_VALIDES").

Exécution du programme de base

Dans Thonny, un appui sur le bouton vert lance le programme. La console indique que la connexion avec le FT-897 s'effectue par le port COM8 à 9600 bauds, puis qu'il faut taper "q" pour quitter l'application.

La fréquence se saisit après le caractère ":" ; je choisis 432.11250 MHz. Le programme affiche aussi la fréquence sur laquelle le poste était réglé (145.7750 MHz) : le CAT sait donc relire la fréquence et le mode du poste.

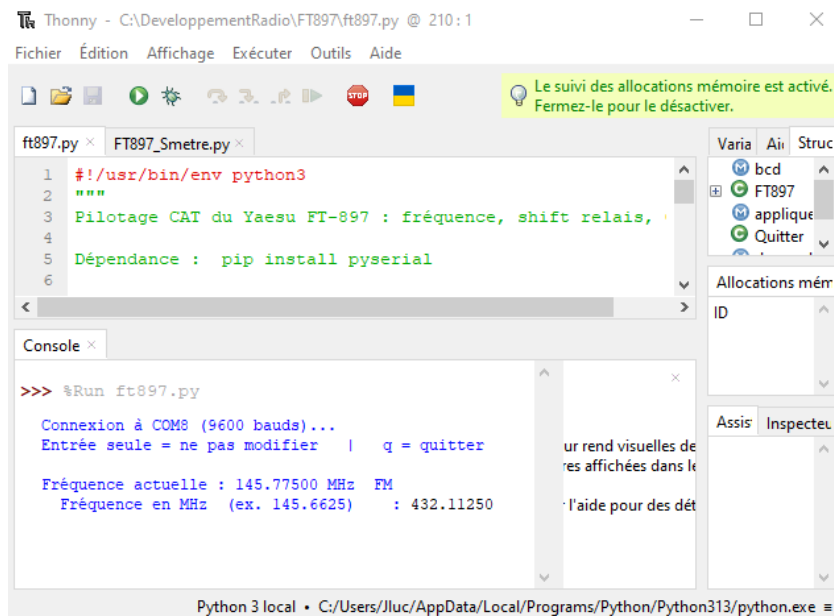


Figure 6. L'exécution du programme permet de préciser la fréquence désirée.

Les autres paramètres sont ensuite entrés de la même manière.

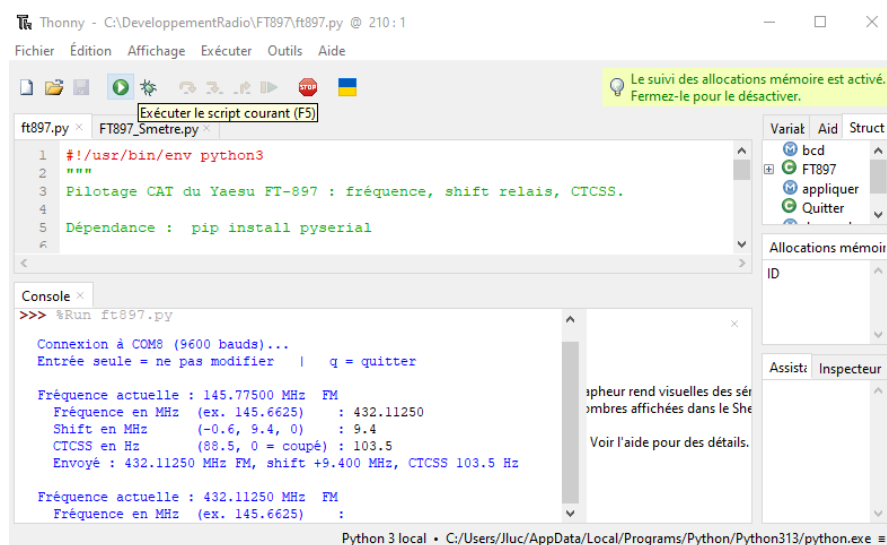


Figure 7. Le décalage et le CTCSS sont ensuite fournis.

Une fois le CTCSS entré et validé par la touche "Entrée", les commandes CAT sont envoyées au FT-897, qui se programme.



Figure 8. Affichage des paramètres de transmission envoyés par l'interface de pilotage.

Le décalage positif est indiqué par le signe “+” en haut à gauche, le CTCSS est bien pris en compte (TEN) et la fréquence est bien celle attendue.

Ajoutons la lecture et l’affichage du S-mètre en fonction du temps

Nous allons lire la valeur du S-mètre et afficher son évolution en fonction du temps. Le graphe sera affiché dans une fenêtre à part. Décrivons ce besoin à Claude par un prompt.

afficher dans une fenêtre à part le niveau du S metre en fonction du temps

Nous ajoutons aussi la mesure et l’affichage de la puissance d’émission.

ajoute le réglage de la puissance?

Je ne présente pas le programme pour les mêmes raisons que précédemment. Il peut être téléchargé depuis le site de l'ARML 49 ⁽³⁾.

Le résultat de l’exécution du programme est montré sur la figure ci-dessous.

```

Console x
fréquence en mhz (ex. 145.6625) : g
Graphe ouvert : fermez la fenêtre pour revenir ici.

Fréquence actuelle : 145.77500 MHz FM S-mètre : S0
Fréquence en MHz (ex. 145.6625) : 145.775
Shift en MHz (-0.6, 9.4, 0) : -0.6
CTCSS en Hz (88.5, 0 = coupé) : 103.5
Envoyé : 145.77500 MHz FM, shift -0.600 MHz, CTCSS 103.5 Hz

Fréquence actuelle : 145.77500 MHz FM S-mètre : S0
Fréquence en MHz (ex. 145.6625) : g
Graphe ouvert : fermez la fenêtre pour revenir ici.
  
```

Figure 9. La fenêtre de Thonny permet de saisir les paramètres de transmission.

Pour faire apparaître le graphe du S-mètre en fonction du temps, il faut saisir la commande “g” dans la console de Thonny. Cette fenêtre regroupe la puissance d’émission en watts (en rouge) et le niveau du signal reçu en points de S-mètre (en bleu).

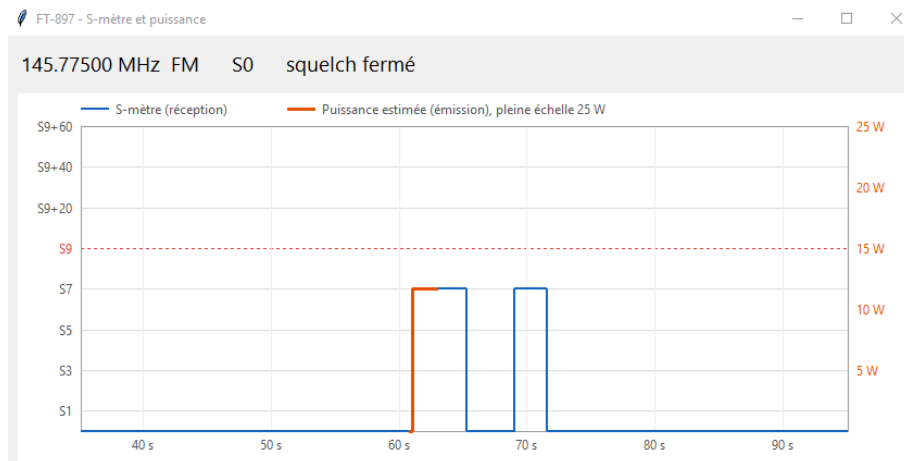


Figure 10. En rouge, la puissance émise pour déclencher l'ouverture du relais. En bleu, le signal reçu, exprimé en points de S-mètre.

Conclusion

L'association de Claude et de Thonny permet de créer facilement ses propres programmes de pilotage CAT du FT-897 (ou d'autres postes).

Dans un deuxième article, je montrerai comment réaliser, avec la même démarche, un analyseur de spectre et un scanner toutes bandes qui s'arrête dès qu'une émission est présente.

Dans un troisième article, nous réaliserons une face avant dotée d'une interface graphique, à partir d'un écran tactile et d'un ESP32.

Merci à Pierre F4HUX et Gilles F6FGH, de l'ARML-49, pour leur relecture attentive et leurs suggestions.

Notes : ⁽¹⁾ <https://claudio.ai>, ⁽²⁾ <https://thonny.org>, ⁽³⁾ <https://arml.r-e-f.org/download/FT897.zip>