

Technique SDR 12

Après quelques années loin de l'ADS-B, je remets en service une station de réception autour d'une clé RTL-SDR et d'une antenne 1090 MHz. Objectif : repartir de zéro, sous Windows puis sous Raspberry Pi, pour recevoir, décoder, afficher et partager les données émises par les avions. Une reprise très pratique, entre Dump1090, ADSB.im et les inévitables petits tâtonnements du terrain.

Introduction

En 2021 et 2022, j'ai publié une série d'articles consacrés aux modes de réception SDR. Vous avez pu les retrouver dans la revue Radioamateurs-France ainsi que sur les sites Hamanalyste.org et ON5VL.org des radioamateurs de Liège, en Belgique.

Dans un but avant tout pédagogique, j'y partageais mes recherches, mes essais, mes réussites, mais aussi mes tâtonnements, afin d'aider tous ceux qui, comme moi, souhaitaient découvrir concrètement les techniques de réception SDR en les pratiquant.

Une fois cette série achevée, je me suis assez vite séparé d'une bonne partie de mon matériel pour voguer vers d'autres centres d'intérêt. Mais le passage presque quotidien, au-dessus de ma tête, des Mirage 2000 et des Rafale de la base aérienne 118 de Mont-de-Marsan, comme celui des nombreux avions de ligne qui sillonnent le ciel des Landes à haute altitude, m'a donné envie de reprendre ces expérimentations.

Je me suis donc équipé d'une nouvelle clé de réception ADS-B et d'une antenne dédiée au 1090 MHz. Toujours animé par le même désir de partager mes essais pas à pas, j'ai rouvert mon traitement de texte pour y consigner mes avancées, mais aussi les difficultés rencontrées en chemin.

Ce douzième volet reprend donc le fil de nos expérimentations aéronautiques. Nous repartirons volontairement de zéro, d'abord sous Windows avec l'installation du pilote de la clé, puis avec une version récente de Dump1090 permettant de décoder et d'afficher les trames ADS-B reçues localement. Nous reviendrons ensuite à la Raspberry Pi, avec l'image Pi24 proposée par Flightradar24, avant d'élargir notre station vers d'autres réseaux grâce à l'image ADSB.im et à la découverte d'OpenSky Network.

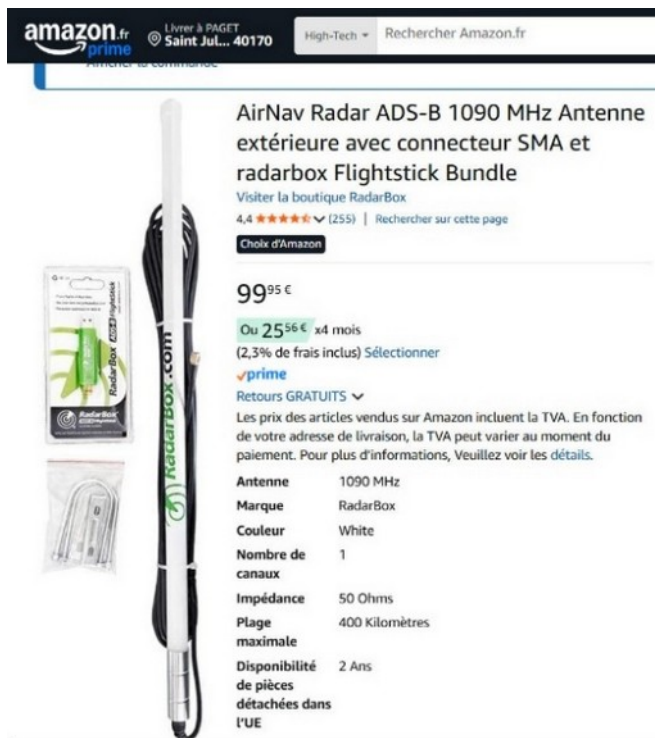
Comme toujours, les spécialistes du domaine relèveront peut-être ici ou là quelques approximations, voire certaines erreurs. J'espère qu'ils ne m'en tiendront pas rigueur et qu'ils accepteront, le cas échéant, de me les signaler afin que nous puissions les corriger ensemble. De même, si certains lecteurs connaissent de meilleures solutions d'installation, de réglage ou d'utilisation, qu'ils n'hésitent pas à me les faire connaître.

Nous pouvons maintenant reprendre le sujet depuis le début, avec le traitement des signaux GPS ADS-B et la remise en service d'une station de réception personnelle

Retour sur le traitement des signaux GPS ADS-B

On repart à zéro

Après quelques hésitations je me décide finalement à acquérir le récepteur **USB ADS-B AirNav RadarBox FlightStick** et son antenne omnidirectionnelle extérieure.

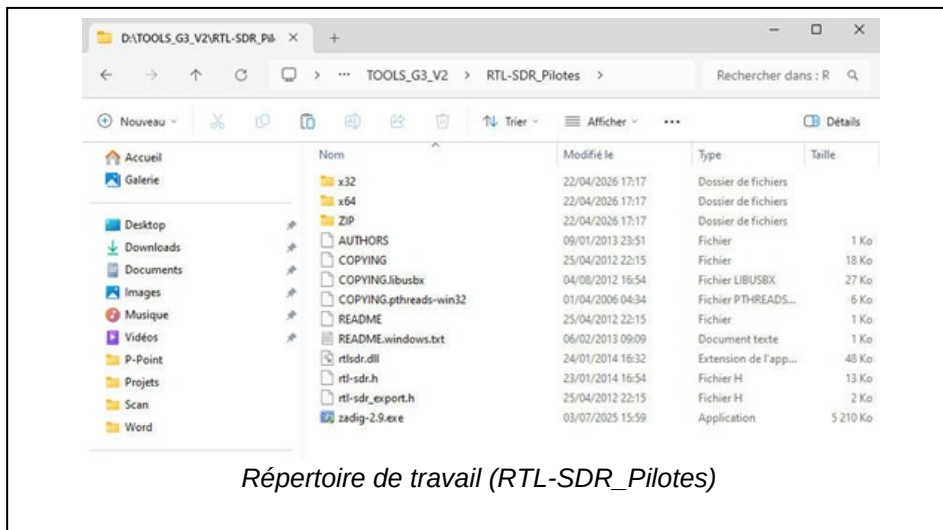


Mon objectif premier n'est pas d'installer, sous Windows, le logiciel fourni (je n'ai d'ailleurs jamais réussi à le faire fonctionner !!), mais de disposer, à un coût raisonnable, d'une antenne et d'un récepteur de qualité, afin de reprendre dans de bonnes conditions certains essais qui ne m'avaient pas pleinement satisfait.

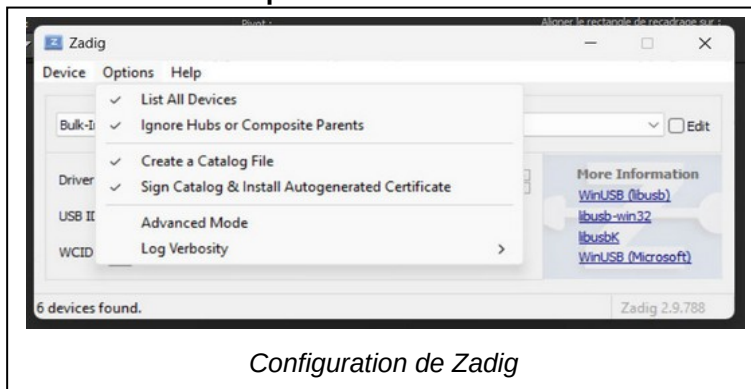
Ce début d'article pourra sans doute paraître redondant au regard de mes précédents textes. J'ai pourtant choisi, comme l'annonce ce titre, de repartir de zéro, afin d'accompagner pas à pas tous ceux qui souhaiteraient découvrir ce domaine.

La première étape consiste en l'installation d'un pilote pour la clé USB à l'aide du logiciel **Zadig**. Je reprends ma précédente méthode :

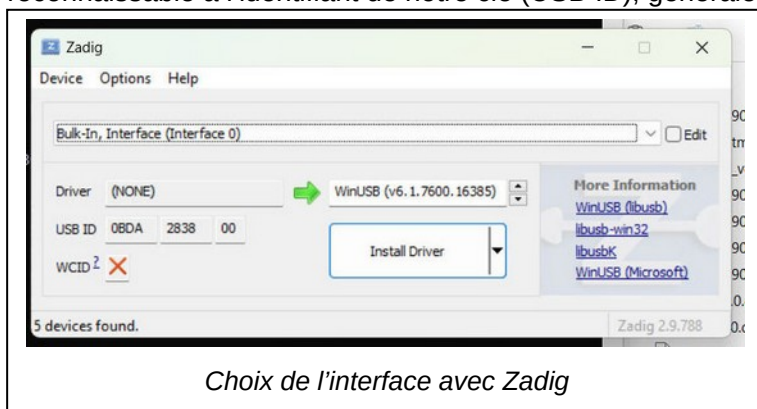
- Téléchargement du logiciel **Zadig** (zadig-2.9.exe) depuis <https://zadig.akeo.ie/downloads/>,
- Téléchargement et décompression des pilotes RTL-SDR depuis https://osmocom.org/attachments/download/2242/R_eIWithDeblInfo.zip,
- Regroupement du logiciel **Zadig** et du pilote rtl-sdr.dll qui convient (x32 ou x64) dans un même répertoire.



- Lancement de **Zadig** en mode administrateur depuis l'explorateur de fichiers. Sélectionnez alors **Options** et cochez le choix **List All Devices**.



- Installation (ou remplacement) du pilote actuel (ici NONE) par le pilote WinUSB. Dans la fenêtre Zadig, le pilote installé est indiqué à gauche, le pilote à installer à droite. Dans la liste déroulante sélectionnez **Bulk-In, Interface (Interface x)**. Cette interface est reconnaissable à l'identifiant de notre clé (USB ID), généralement **0BDA 2838 0x¹**



- Clic sur **Install (ou Replace) Driver**. Dans certains cas un message d'alerte signale que *Windows ne peut pas vérifier l'éditeur de ce pilote*. Il faut alors choisir **Installer ce pilote quand même**. Faites cette manipulation pour les interfaces 0 et 1.

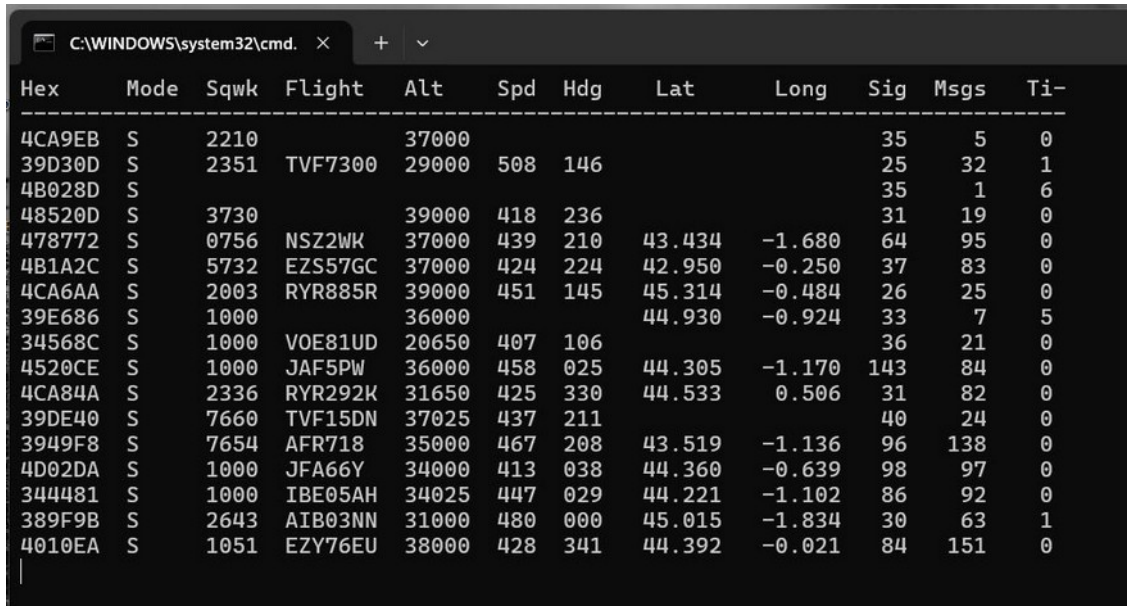
¹ VID, PID, ID pour : Identifiant du vendeur, Identifiant du produit et Rang de l'interface

Windows - Dump1090 – V0.4.8.0

En 2021, afin de visualiser sous Windows le trafic aérien autour de mon domicile, j'avais associé une ancienne version du logiciel **Dump1090**² au programme **Virtual Radar**³.

La mise en place de l'ensemble s'était révélée assez fastidieuse, pour un résultat qui ne m'avait pas entièrement satisfait.

Pour mémoire, car nous y reviendrons plus loin, lorsque j'aborderai le code **Sqwk**, je reproduis ci-dessous l'écran affiché, à l'époque, par dump1090.



Hex	Mode	Sqwk	Flight	Alt	Spd	Hdg	Lat	Long	Sig	Msgs	Ti-
4CA9EB	S	2210		37000					35	5	0
39D30D	S	2351	TVF7300	29000	508	146			25	32	1
4B028D	S								35	1	6
48520D	S	3730		39000	418	236			31	19	0
478772	S	0756	NSZ2WK	37000	439	210	43.434	-1.680	64	95	0
4B1A2C	S	5732	EZS57GC	37000	424	224	42.950	-0.250	37	83	0
4CA6AA	S	2003	RYR885R	39000	451	145	45.314	-0.484	26	25	0
39E686	S	1000		36000			44.930	-0.924	33	7	5
34568C	S	1000	VOE81UD	20650	407	106			36	21	0
4520CE	S	1000	JAF5PW	36000	458	025	44.305	-1.170	143	84	0
4CA84A	S	2336	RYR292K	31650	425	330	44.533	0.506	31	82	0
39DE40	S	7660	TVF15DN	37025	437	211			40	24	0
3949F8	S	7654	AFR718	35000	467	208	43.519	-1.136	96	138	0
4D02DA	S	1000	JFA66Y	34000	413	038	44.360	-0.639	98	97	0
344481	S	1000	IBE05AH	34025	447	029	44.221	-1.102	86	92	0
389F9B	S	2643	AIB03NN	31000	480	000	45.015	-1.834	30	63	1
4010EA	S	1051	EZY76EU	38000	428	341	44.392	-0.021	84	151	0

Les avions détectés au lancement de dump1090.bat

Mes recherches sur Internet m'ont heureusement permis de découvrir une autre version de dump1090, développée par Gisle Vanem⁴ et repérée sur le forum FlightAware⁵.

Voici donc la méthode que j'ai suivie :

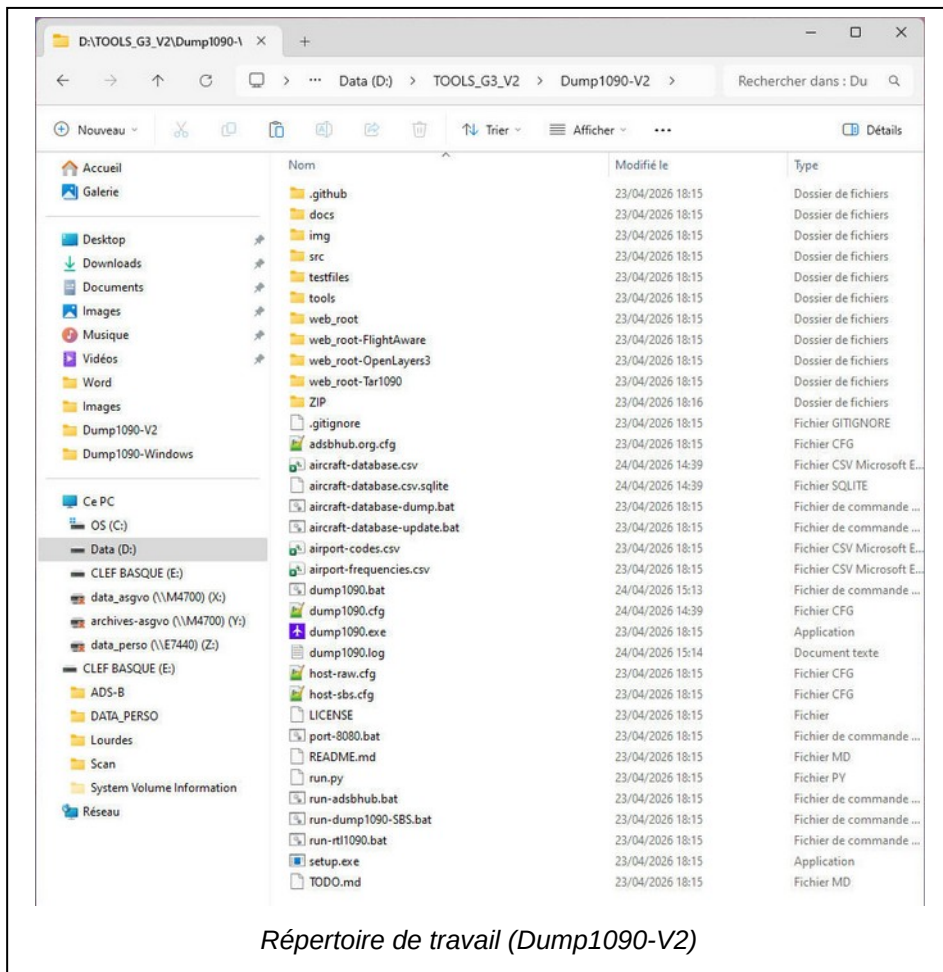
- J'ai téléchargé le nouveau package logiciel depuis l'URL <https://github.com/gvanem/Dump1090/archive/refs/heads/main.zip> puis j'en ai décompressé l'ensemble ;
- Après décompression j'ai retrouvé parmi les nombreux de fichiers du package, notre exécutable dump1090.exe. Sa taille était ici de 2,04 Mo (2 143 744 octets).

² Version V1.10.2910.14 proposée par Abdul Azzam Ajhari (<https://github.com/abdazzamajhari>), et téléchargeable depuis l'URL : <http://www.satsignal.eu/software/dump1090-win.1.10.3010.14.zip>

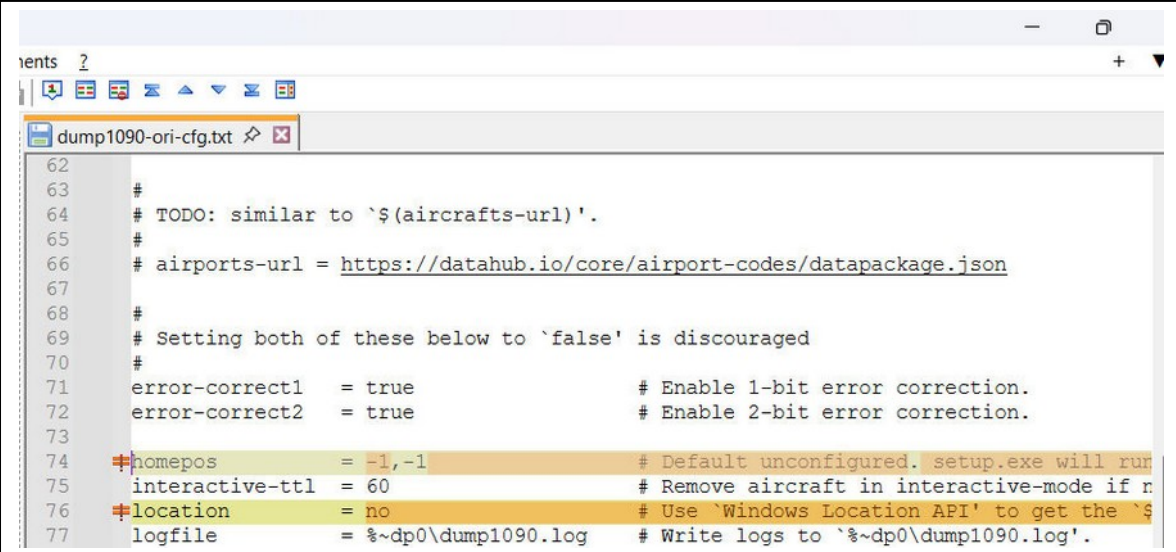
³ Disponible depuis la page <http://www.virtualradarserver.co.uk/Download.aspx>

⁴ <https://github.com/gvanem>

⁵ <https://discussions.flightaware.com/t/dump1090-for-windows-with-tar1090/97084>



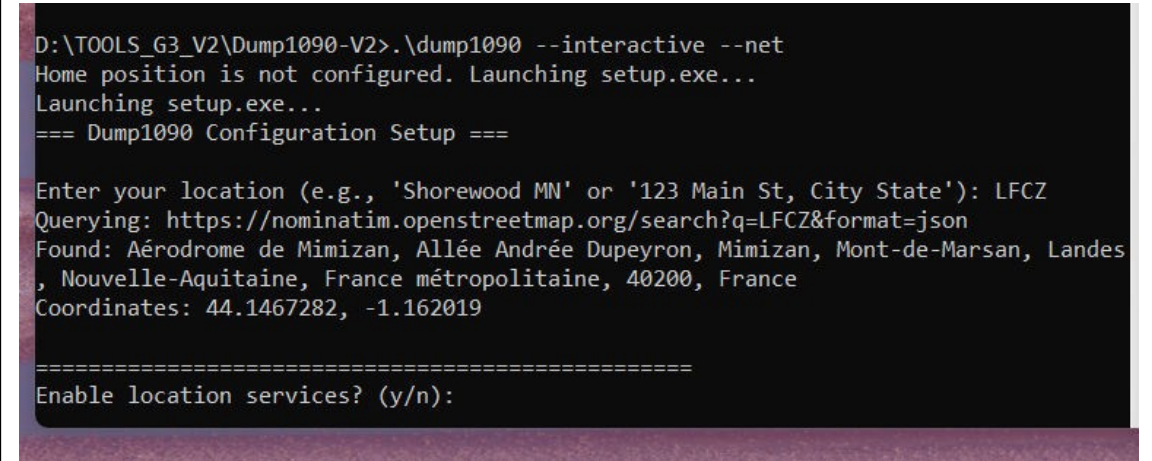
- Par habitude, j'ai créé un fichier dump1090.bat contenant les lignes suivantes :
 - o dump1090.exe --interactive --net
 - o pause
- Un coup d'œil au fichier dump1090.cfg m'a montré qu'une procédure particulière se déclencherait en absence de coordonnées géographiques correctes ;



```
62
63 #
64 # TODO: similar to `${aircrafts-url}'.
65 #
66 # airports-url = https://datahub.io/core/airport-codes/datapackage.json
67
68 #
69 # Setting both of these below to `false' is discouraged
70 #
71 error-correct1 = true           # Enable 1-bit error correction.
72 error-correct2 = true           # Enable 2-bit error correction.
73
74 homepos = -1,-1                # Default unconfigured. setup.exe will run
75 interactive-ttl = 60            # Remove aircraft in interactive-mode if n
76 location = no                  # Use `Windows Location API' to get the `§
77 logfile = %~dp0\dump1090.log   # Write logs to `%~dp0\dump1090.log'.
```

Fichier dump1090.cfg avant la procédure de setup.

- Lors du premier lancement, ma position géographique m'a effectivement été demandée.



```
D:\TOOLS_G3_V2\Dump1090-V2>.\dump1090 --interactive --net
Home position is not configured. Launching setup.exe...
Launching setup.exe...
=== Dump1090 Configuration Setup ===

Enter your location (e.g., 'Shorewood MN' or '123 Main St, City State'): LFCZ
Querying: https://nominatim.openstreetmap.org/search?q=LFCZ&format=json
Found: Aérodrome de Mimizan, Allée Andrée Dupeyron, Mimizan, Mont-de-Marsan, Landes
, Nouvelle-Aquitaine, France métropolitaine, 40200, France
Coordinates: 44.1467282, -1.162019

=====
Enable location services? (y/n):
```

Premier lancement de dump1090.bat

- Étant proche de l'aérodrome de Mimizan, j'en ai indiqué le code OACI⁶ Celui-ci a été reconnu, et ses coordonnées ont été automatiquement enregistrées dans le fichier de configuration. Je n'ai donc pas eu besoin d'ouvrir le fichier dump1090.cfg pour modifier la ligne homepos.⁷ La mise à jour s'est faite d'elle-même ;

⁶ Organisation de l'aviation civile internationale (ICAO - International Civil Aviation Organization).

⁷ Ce que j'ai réalisé plus tard aidé du site : <https://www.coordonnees-gps.fr/conversion-coordonnees-gps> afin de préciser mes coordonnées exactes.

```

62
63 #
64 # TODO: similar to `$(aircrafts-url)'.
65 #
66 # airports-url = https://datahub.io/core/airport-codes/datapackage.json
67 #
68 #
69 # Setting both of these below to `false' is discouraged
70 #
71 error-correct1 = true # Enable 1-bit error correction.
72 error-correct2 = true # Enable 2-bit error correction.
73
74 #homepos = 44.14617282,-1.162019
75 interactive-ttl = 60 # Remove aircraft in interactive-mode if r
76 #location = yes
77 logfile = %~dp0\dump1090.log # Write logs to `%~dp0\dump1090.log'.
    
```

Fichier dump1090.cfg après la procédure de setup

- Alors que le processus d'initialisation s'achevait, j'ai autorisé les modifications à apporter à mon pare-feu ;

ICAO	Callsign	Reg-num	Cntry	DEP	DEST	Altitude	Speed	Lat	Long	Hdg
02A1A7	-	TS-INH	TN	-	-	-	809	-	-	318
39644A	-	F-GZCK	FR	-	-	11271	834	+44.956	+0.235	213
39D310	-	-	FR	-	-	11575	-	-	-	-
3C6548	-	D-AI7H	DE	-	-	11583	821	-	-	56
4CAE65	-	-	IE	-	-	11271	822	+45.084	+0.738	183
AS4A3F	-	N44MG	US	-	-	12177	-	+44.945	-2.525	-
3985A9	AFR561V	F-HBNJ	FR	CDG	MAD	10654	820	+44.230	-0.661	207
440053	EJU19FR	OE-ICK	AT	OPO	ORY	10966	810	+45.008	-0.7	-
407E54	EXS12DR	-	GB	ALC	MAN	11583	828	+44.074	+0.3	-
400FE2	EZY360H	G-EZBR	GB	LGW	ALC	11278	792	+44.059	+0.7	-
391EFB	FAZON	-	FR	?	?	350	-	-	-	-
398645	FHBSF	F-HBSF	FR	?	?	2224	-	-	-	-
3CEDB8	LUA918A	-	DE	?	?	12185	804	+43.610	-1.6	-
AS4A3F	LXJ484	N484FX	US	?	?	13099	837	+44.156	-0.9	-
020112	RAM1631	CN-RGQ	MA	BRU	TTU	10662	816	+44.667	-0.3	-
4CAE55	RVR2154	-	IE	BHX	RWJ	10052	855	+44.837	+1.1	-
48194C	SMR200	HB-JPC	CH	OPO	ZRH	10959	823	+42.853	-0.6	-
347256	VOE50CB	EC-NTL	ES	PMI	BOD	4965	601	+44.514	+0.1	-
115.5	92	0	s	-	-	-	-	-	-	-

Premier lancement de dump1090.bat

- Suivant les conseils du tutorial j'ai ensuite quitté puis relancé mon batch dump1090.bat.

ICAO	Callsign	Reg-num	Cntry	DEP	DEST	Altitude	Speed	Lat	Long	Hdg	Dist	Msg	Seen
0A0026	-	7T-VJS	DZ	-	-	-	850	-	-	340	-	8	7 s
347288	-	EC-NTP	ES	-	-	10966	826	+42.745	+0.159	55	188.9	94	5 s
43EB9F	-	2-NYAW	GG	-	-	10662	810	-	-	155	-	3	15 s
398562	AFR88EA	F-HBLC	FR	CDG	BIO	7006	728	+43.736	-1.301	210	47.1	6177	0 s
3807FA	AIB01AE	-	FR	?	?	6351	808	+43.754	-0.233	120	86.2	4720	11 s
3451D8	ANE1300	EC-MJO	ES	BRU	MAD	10052	795	+43.860	-1.147	214	31.9	6990	0 s
343694	ANE314G	EC-LKF	ES	AMS	MAD	10662	798	+43.204	-1.378	207	106.3	5134	1 s
3C54A2	EWG27V	-	DE	DUS	AGP	11278	794	+42.914	-0.958	216	138.1	3246	1 s
407E61	EZY25CU	-	GB	PMI	LGW	11575	810	+44.777	+0.136	347	124.6	4429	0 s
408184	EZY74VW	-	GB	TLS	BRS	10357	769	+45.423	-0.250	329	159.1	1749	19 s
399980	FHGMA	F-HGMA	FR	?	?	609	-	-	-	-	-	539	2 s
346185	IBE05ZT	EC-NEA	ES	ORY	MAD	10045	805	+44.711	-0.091	211	105.7	484	0 s
4CC2A3	ICE597	TF-FIN	IS	BCN	KEF	11035	826	+45.013	-0.155	336	125.0	4121	25 s
4ACB23	NSZ4HP	-	SE	AGP	HAU	11575	856	+43.074	-2.872	6	182.1	138	0 s
0201C3	RAM770T	-	MA	CMN	ORY	10966	878	+43.930	-1.452	26	33.4	3614	0 s
4D225D	RYR299H	9H-QDN	MT	PMI	BRS	12185	820	+44.733	-0.069	338	108.5	6347	0 s
4D223F	RYR588G	9H-QAS	MT	ALC	NWI	10966	849	+43.980	+0.313	345	119.3	658	1 s
4CA564	RYR66BN	EI-DPY	IE	VLC	BHX	11575	831	+44.065	+0.186	349	108.0	1283	0 s
4B1807	SWR8YB	HB-JCP	CH	LIS	GVA	10966	827	+43.121	-0.223	58	136.8	897	0 s
4951CC	TAP921	CS-TNL	PT	ZRH	OPO	11278	793	+43.534	-0.228	234	101.2	1051	0 s
4853D3	TRA15N	PH-HXJ	NL	BRU	ALC	11880	816	+44.615	+1.136	195	189.9	852	0 s
39DE4F	TVF43MN	-	FR	ORY	RAK	11271	789	+44.121	-0.724	208	35.1	6684	0 s
3964EE	TVF51DF	F-GZHO	FR	ORY	BIQ	8552	813	+44.693	-0.512	207	79.7	556	0 s
4D204A	WZZ66HG	9H-GMT	MT	WAW	OPO	11263	793	+43.003	-1.403	244	128.6	1637	17 s

Second lancement de dump1090.bat

Nous avons alors retrouvé, décodées et mises en forme, les informations contenues dans les trame ADS-B.

Un coup d'œil au fichier dump1090 . Log montre que le processus d'installation a également permis le téléchargement et/ou la mise à jour de plusieurs fichiers bases d'enrichissement local qui permettront à **Dump1090** d'enrichir l'affichage en retrouvant l'immatriculation et le type des aéronefs :

```

-----
14:39:32.892: Starting: D:\TOOLS_G3_V2\Dump1090-V2\dump1090.exe --interactive --net
Build info: Microsoft cl 19.50.35727, built on: 4 Apr 2026

14:39:33.360: Force updating 'D:\TOOLS_G3_V2\Dump1090-V2\aircraft-database.csv' since it does not exist.
14:39:33.360: File 'C:\Users\Francois\AppData\Local\Temp\dump1090\aircraft-database-temp.zip' doesn't exist (or is truncated). Forcing a
download.
14:39:33.360: Force updating 'C:\Users\Francois\AppData\Local\Temp\dump1090\aircraft-database-temp.zip' from 'https://s3.opensky-
network.org/data-samples/metadata/aircraftDatabase.zip'
14:39:36.346: Copied 'C:\Users\Francois\AppData\Local\Temp\dump1090\aircraft-database-temp.csv' -> 'D:\TOOLS_G3_V2\Dump1090-V2\aircraft-
database.csv'
14:39:36.346: Deleting 'D:\TOOLS_G3_V2\Dump1090-V2\aircraft-database.csv.sqlite' to force a rebuild in 'aircraft_CSV_load()'
14:39:36.346: using Sqlite file: "D:\TOOLS_G3_V2\Dump1090-V2\aircraft-database.csv.sqlite".
14:39:36.346: Loading 'D:\TOOLS_G3_V2\Dump1090-V2\aircraft-database.csv' could take some time.
14:39:38.833: Creating SQL-database 'D:\TOOLS_G3_V2\Dump1090-V2\aircraft-database.csv.sqlite'... 14:39:40.629: Created 519998 records
14:39:40.801: pair_content():
pair[1] -> fg: 7, bg: 1
pair[2] -> fg: 2, bg: 1
pair[3] -> fg: 4, bg: 1
pair[4] -> fg: 6, bg: 2
pair[5] -> fg: 0, bg: 7
pair[6] -> fg: 15, bg: 254

14:39:40.817: Found 1 device(s):
14:39:40.817: 0: AIRNAV ADSB_1090 SN: 00000010 (currently selected)
14:39:41.223: Supported gains: 0.0, 3.4, 6.8, 10.2, 13.7, 17.1, 20.7, 24.0, 27.8, 31.2, 34.6, 38.2, 41.6, 45.3, 48.8, 52.7.
14:39:41.254: Tuner gain set to 31 dB.
14:39:41.332: Tuned to 1090.000 MHz.
14:39:41.332: Gain reported by device: 31.20 dB.
14:39:41.348: Modes.rtlsdr.index: 0 (name: '<none>')
Modes.sdrplay.index: -1 (name: '<none>')
Modes.sample_rate: 2.4 MS/s
Modes.rtltcp.remote: <none>
Modes.selected_dev: ADSB_1090 (AIRNAV)
Modes.bytes_per_sample: 2
Modes.trailing_samples: 326
Modes.input_format: 1 / INPUT_UC8
Modes.DC_filter: 1
Modes.measure_noise: 1
Modes.phase_enhance: 0
Modes.demod_func: demod_2400()
Modes.FIFO_bufs: 15
Modes.FIFO_acquire_ms: 100
Modes.FIFO_dequeue_ms: 100
Using converter: convert_uc8_generic(), 'UC8, float path'

14:39:41.442: Added 0 IPv4 and 0 IPv6 addresses to deny.
IPv4 DNS: udp://192.168.1.1:53
IPv6 DNS: udp://[2001:4860:4860::8888]:53
14:39:41.457: Modes.home_pos: 1.1620190 W, 44.1467282 N

Ln 1, Col 1 225194 caractères Texte brut 90% Windows (CRLF) UTF-8 avec marque d

```

Trame ADS-B (source : <https://www.radartutorial.eu/13.ssr/sr25.fr.html>)

- `aircraft-database.csv` : base OpenSky⁸ des métadonnées aéronaves : adresse ICAO/Mode-S, immatriculation, constructeur, type, opérateur, etc. Elle permet à Dump1090 d'associer un code ICAO reçu à un avion identifiable.
- `aircraft-database.csv.sqlite` : version SQLite locale, reconstruite à partir du CSV précédent, afin d'accélérer les recherches dans cette grosse base sans relire tout le fichier texte.

Deux autres fichiers sont présents dans le package, ils contiennent des informations associées aux aéroports et à leurs fréquences radio⁹ :

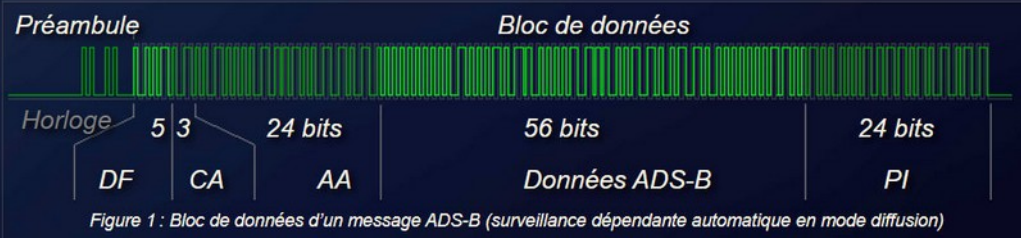
- `airport-codes.csv` : base des aéroports : codes ICAO/IATA ou locaux, nom, type d'aéroport, pays, région, ville desservie, coordonnées, etc. Elle sert surtout à interpréter les routes, origines/destinations ou codes d'aéroports.
- `airport-frequencies.csv` : base des fréquences radio associées aux aéroports : tour, sol, ATIS, approche, UNICOM, etc., avec fréquence en MHz.

Revenons maintenant à la description des données affichées par dump1090 en provenance des trames ADS-B.

⁸ <https://opensky-network.org/data>

⁹ <https://ourairports.com/data/>

Les impulsions du préambule sont utilisées pour la synchronisation avec une horloge de référence, ce qui est nécessaire pour décoder les informations du bloc de données pour la **modulation de l'espacement des impulsions** (MEI). Dans le champ « code du format » envoyé se retrouve le nombre décimal 17, ce qui correspond au binaire **10001_b**.



Les trois bits suivants (appelé capacité ou CA) désignent le sous-type de message ADS-B, cet arrangement permet un total de huit types. Les 24 bits qui suivent donnent l'identificateur de l'appareil tel qu'inscrit à OACI.

Les 56 bits suivants donnent le rapport ADS-B lui-même selon le type choisi dans le champ CA. Ce rapport peut inclure :

- Le type d'aéronef et ses lettres d'identification (« lettre de queue ») ;
- L'altitude de l'appareil ;
- La latitude encodée ;
- La longitude encodée grossièrement ;
- La vitesse de vol.

Les derniers 24 bits donnent un code de parité qui permet de détecter toute erreur de données au décodage.

Trame ADS-B (source : <https://www.radartutorial.eu/13.ssr/sr25.fr.html>)

De gauche à droite :

- **ICAO** : Il s'agit - ici en hexadécimal – de l'identification OACI 24 bits de l'appareil. C'est l'identifiant unique du transpondeur de l'avion, de l'hélicoptère ou du planeur. Les 24 bits, permettent plus de 16 millions de combinaisons (soit 2 à la puissance 24) évitant que deux avions aient le même code transpondeur. Chaque pays, en fonction de son importance, gère une plage de code. Il correspond à la colonne **Hex** de la précédente version.
- **Callsign** : Identifiant de vol transmis par l'avion. Dans la pratique, il se compose le plus souvent de deux parties :

Partie	Exemple	Signification
Préfixe compagnie / opérateur	AFR, RYR, IBE, EZY, TVF	Code OACI à trois lettres de la compagnie ou de l'opérateur
Numéro ou suffixe de vol	88EA, 5886, 052T, 77OT	Numéro de vol opérationnel, parfois complété par une ou deux lettres

Exemples tirés de la capture :

Callsign	Lecture probable
AFR88EA	Vol Air France, indicatif radio AFR 88EA
RYR5886	Vol Ryanair, indicatif RYR 5886

Callsign	Lecture probable
IBE052T	Vol Iberia, indicatif IBE 052T
EZY770T	Vol EasyJet UK, indicatif EZY 770T
TRA15N	Vol Transavia, indicatif TRA 15N

Le **Callsign** ADS-B n'est pas toujours identique au numéro de vol public affiché aux passagers. Les compagnies utilisent souvent des suffixes alphanumériques pour éviter les confusions radio entre deux vols ayant des numéros proches. C'est pourquoi on voit des indicatifs comme AFR88EA, EZY770T ou TVF43MN, et pas seulement des formes très simples comme AFR1234.

Cet identifiant peut changer d'un vol à l'autre à la différence du **Reg-num**.

- **Reg-num** : Numéro d'immatriculation officiel de l'appareil. Exemple : F-HBLC, EC-MJO, 9H-QDN. Également connu sous l'intitulé "tail number," c'est, en principe, le numéro que l'on retrouve peint sur l'avion, souvent à l'arrière du fuselage ou près de l'empennage.

Partie	Exemple	Signification
Préfixe national	F, EC, G, EI, 9H, HB	Pays ou registre d'immatriculation
Suffixe individuel	HBLC, MJO, DNQ, DCL, QDN, JFD	Identifiant propre à l'appareil dans ce registre

Exemples tirés de la capture :

Reg-num	Lecture
F-HBLC	Appareil immatriculé en France
EC-MJO	Appareil immatriculé en Espagne
G-DNQ	Appareil immatriculé au Royaume-Uni
EI-DCL	Appareil immatriculé en Irlande
9H-QDN	Appareil immatriculé à Malte
HB-JFD	Appareil immatriculé en Suisse

- **Cntry** : Pays d'immatriculation ou d'enregistrement de l'appareil. Il est ici affiché sous forme de code pays : FR, ES, GB, IE, etc.
- **DEP** : Aéroport de départ: CDG, ORY, BRU, PMI, etc. Il s'agit d'une information enrichie par base de données et non une donnée ADS-B brute transmise par l'avion¹⁰.

¹⁰ Cette est par exemple disponible depuis le site : <https://www.data.gouv.fr/datasets/airports-locations-data>

- **DEST** : Aéroport de destination au format IATA à trois lettres également issu d'une base de données de routes/indicatifs.
- **Altitude** : Altitude de l'appareil. Alors que l'altitude est classiquement donnée en pieds, elle est ici affichée en mètres. Exemple : 10966 correspond à environ 36 000 pieds.
- **Speed** : Vitesse sol de l'appareil. Normalement exprimée en nœuds, elle est ici convertie en km/h.
- **Lat** : Latitude de l'appareil en degrés décimaux. Le signe + indique l'hémisphère Nord.
- **Long** : Longitude de l'appareil en degrés décimaux. Le signe - indique l'ouest du méridien de Greenwich ; le signe + indique l'est.
- **Hdg** : Cap suivi par l'appareil, en degrés de 0 à 359.
0 = North, 90 = East, 180 = South, 270 = West.
- **Dist** : Distance entre notre station de réception et l'appareil. Elle est exprimée en kilomètres.
- **Msg** : Nombre de messages Mode S / ADS-B reçus pour cet appareil depuis son apparition dans la liste.
- **Seen** : Temps écoulé, exprimé en secondes, depuis la dernière réception d'un message pour cet appareil.
- **I** : Indicateur graphique d'activité ou de rafraîchissement de l'affichage console. Il ne correspond pas à une donnée aéronautique.

À la différence de l'affichage interactif classique de **dump1090**, cette nouvelle présentation ne montre pas le code **Sqwk**¹¹. Ce code, également appelé squawk ou *Mode 3/A*, correspond au code transpondeur généralement attribué par le contrôle aérien, puis émis par l'appareil. En exploitation normale, il sert à identifier un vol sur les écrans radar, afin d'associer sans ambiguïté un écho radar à l'appareil suivi. Ce code est unique dans la zone où l'avion est contrôlé, mais il n'est pas nécessairement conservé pendant tout le trajet. Il peut être modifié lorsqu'un appareil change de secteur, de centre de contrôle ou de pays.

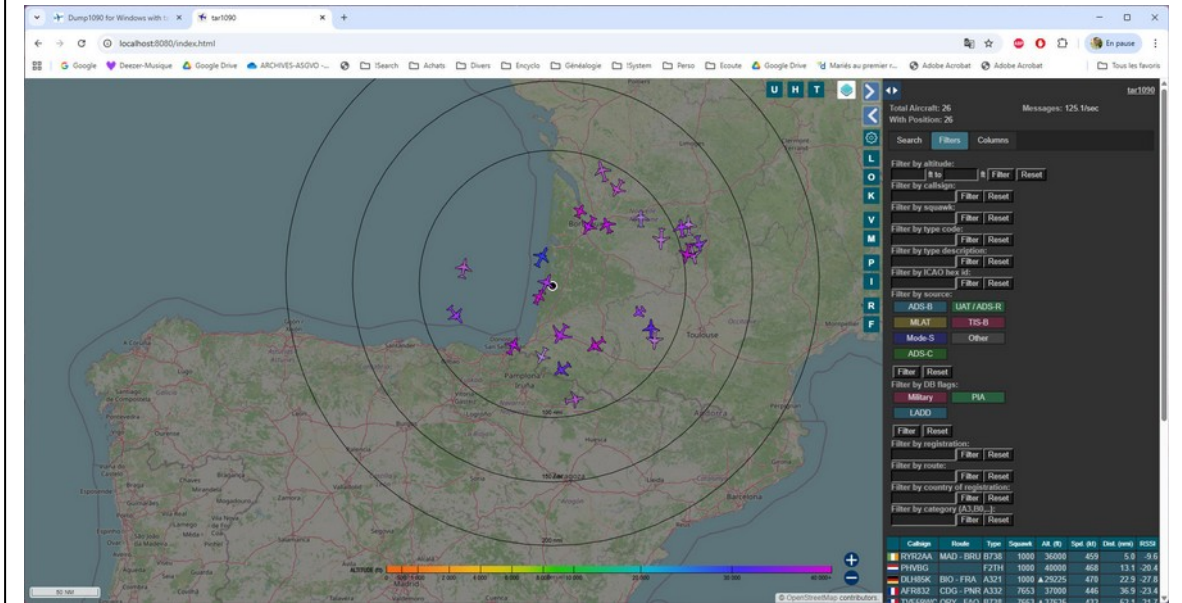
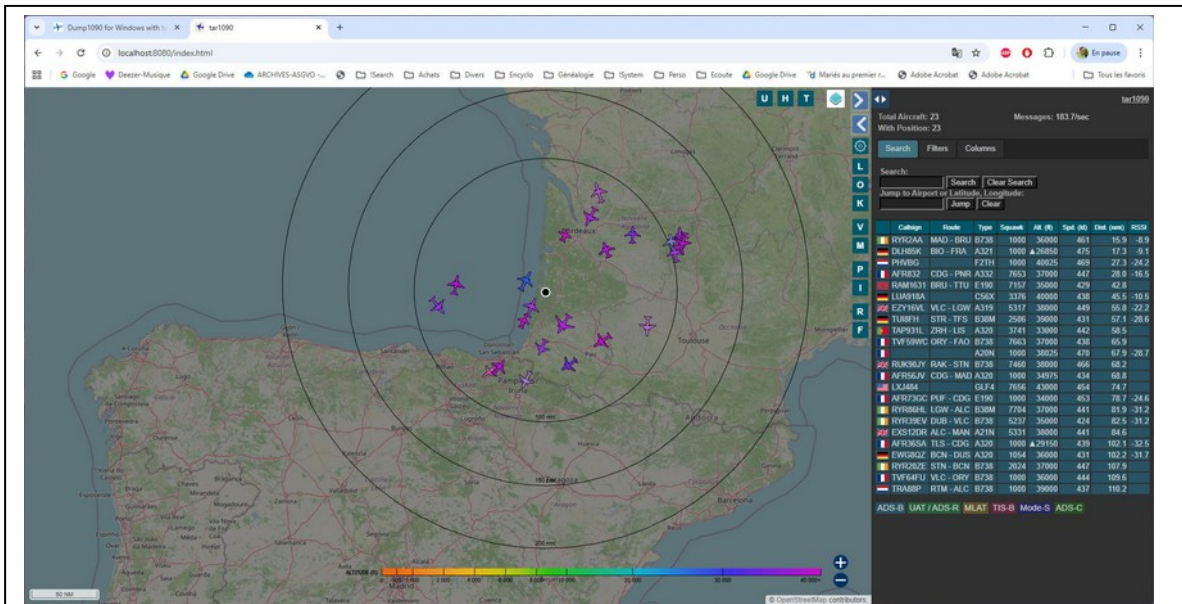
Certains codes sont toutefois réservés aux situations d'urgence : 7500 pour une intervention illicite, 7600 pour une panne radio et 7700 pour une urgence générale. Dans ces cas, c'est le pilote qui sélectionne volontairement le code approprié sur son transpondeur afin d'alerter immédiatement les services de contrôle.

Il est temps maintenant de visualiser sur une véritable carte mes propres données en appelant l'URL `localhost:8080`.

¹¹ https://en.wikipedia.org/wiki/List_of_transponder_codes

Retour sur le traitement des signaux GPS ADS-B - Version I – avril 2026

François PAGET, F-80543 - f80543@gmail.com



Appel depuis Google Chrome de localhost : 8080

Raspberry Pi – Flightradar24

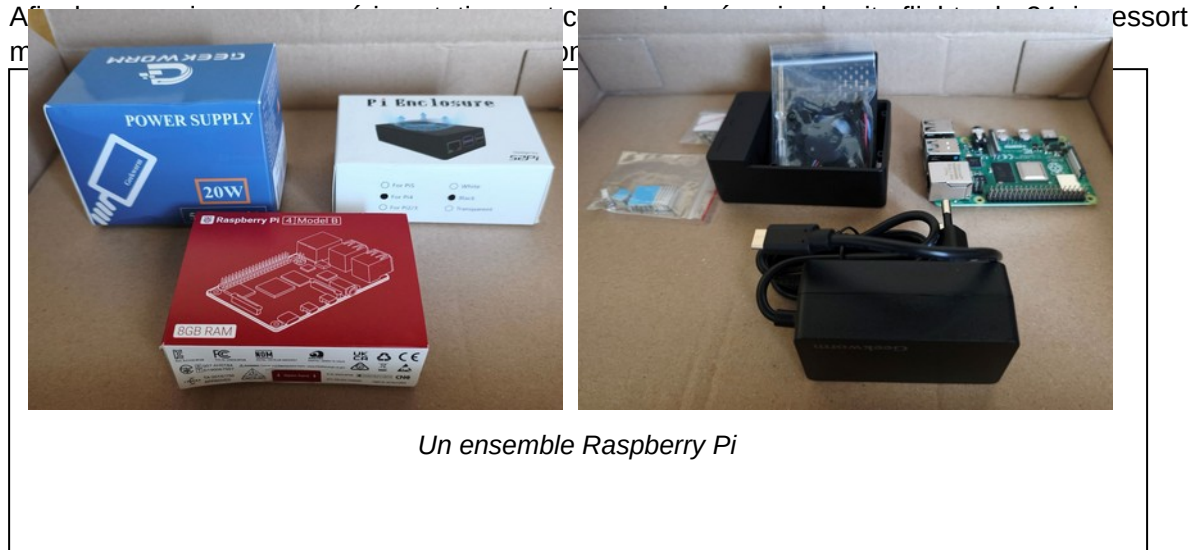
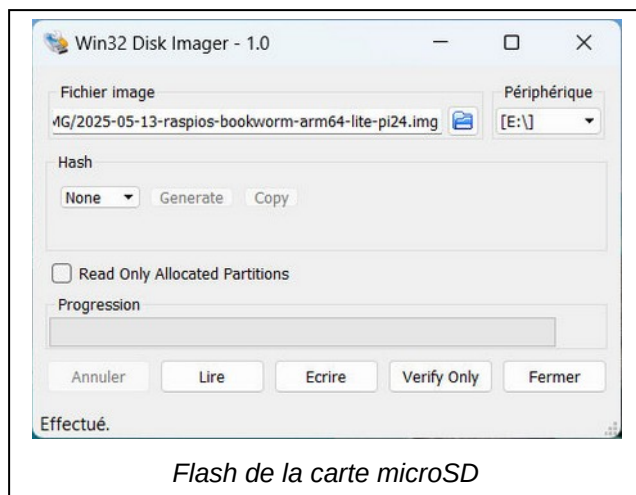


Image Pi24

La première étape consiste à flasher sur une carte microSD de 8Go l'image Pi24 proposée par le site. Celle-ci est disponible depuis l'adresse <https://www.flightradar24.com/build-your-own>.

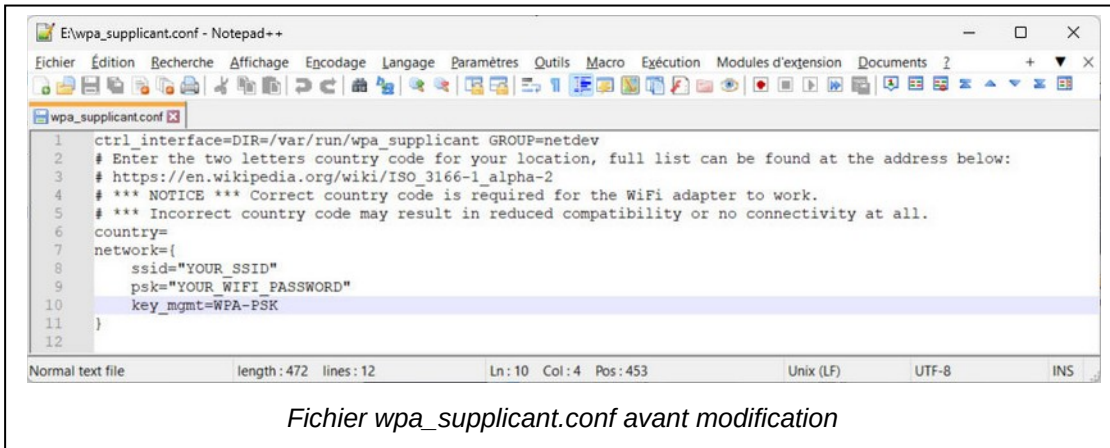
Plusieurs tutoriels conseillent pour cela le logiciel **Etcher** (<https://etcher.balena.io/>). Après plusieurs essais infructueux (écran bleu en milieu du traitement et/ou durant la phase de vérification), je me retourne vers le bien connu **Win32 Disk Imager** disponible à l'adresse <https://win32diskimager.org/>.



Pour permettre un futur accès Wi-Fi sur le Raspberry Pi, il est nécessaire de modifier le fichier `wpa_supplicant.conf` présent sur la carte en saisissant, entre guillemets :

- le code à deux lettres de notre pays (`country=«FR»`),
- le nom de notre routeur dans `ssid`,
- son mot de passe dans `psk`.

Cette modification s'effectue depuis le PC avec un éditeur de texte. Pour ma part, j'utilise Notepad++.



```
1 ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
2 # Enter the two letters country code for your location, full list can be found at the address below:
3 # https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2
4 # *** NOTICE *** Correct country code is required for the WiFi adapter to work.
5 # *** Incorrect country code may result in reduced compatibility or no connectivity at all.
6 country=
7 network={
8     ssid="YOUR_SSID"
9     psk="YOUR_WIFI_PASSWORD"
10    key_mgmt=WPA-PSK
11 }
12
```

Fichier *wpa_supplicant.conf* avant modification

Le logiciel PuTTY va nous permettre d'accéder au Raspberry Pi. Il est disponible à l'adresse : <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>



Download PuTTY: latest release (0.83)

[Home](#) | [FAQ](#) | [Feedback](#) | [Licence](#) | [Updates](#) | [Mirrors](#) | [Keys](#) | [Links](#) | [Team](#)
Download: [Stable](#) · [Snapshot](#) | [Docs](#) | [Privacy](#) | [Changes](#) | [Wishlist](#)

This page contains download links for the latest released version of PuTTY. Currently this is 0.83, released on 2025-02-08.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a [perman](#)

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the worth trying out the [development snapshots](#), to see if the problem has already been fixed in those versions.

Package files

You probably want one of these. They include versions of all the PuTTY utilities (except the new and slightly experimental Windows pterm). (You probably want the 64-bit x86 version. The 32-bit version is only for backward compatibility with very old PCs / versions of Windows.)

We also publish the latest PuTTY installers for all Windows architectures as a free-of-charge download at the [Microsoft Store](#); they usually t

MSI ("Windows Installer")

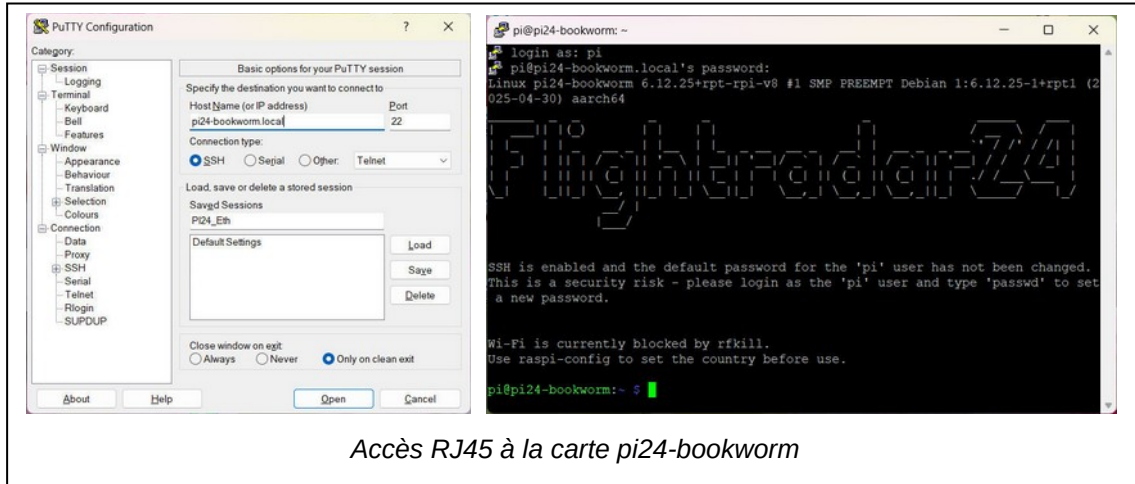
64-bit x86:	putty-64bit-0.83-installer.msi	(signature)
64-bit Arm:	putty-arm64-0.83-installer.msi	(signature)
32-bit x86:	putty-0.83-installer.msi	(signature)

Unix source archive

.tar.gz:	putty-0.83.tar.gz	(signature)
----------	-----------------------------------	-----------------------------

Site de téléchargement pour PuTTY

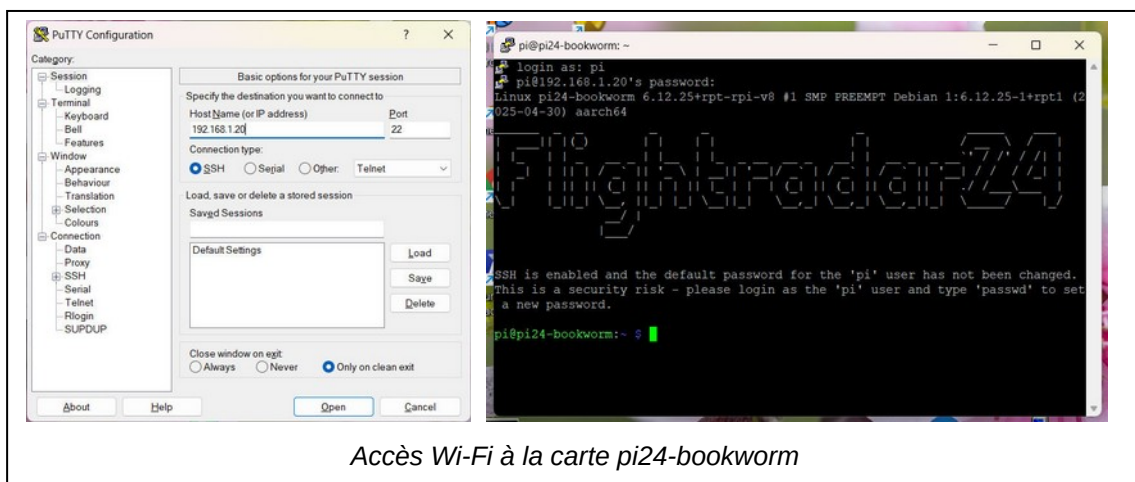
NOTA : PuTTY peut également vous permettre d'accéder à votre Raspberry Pi à l'aide d'un câble RJ45, sans passer par la modification immédiate du fichier `wpa_supplicant.conf`. Utilisez dans ce cas le nom d'hôte de la carte : `pi24-bookworm.local` puis les crédeniels par défaut : `pi / raspberry`.



Accès RJ45 à la carte pi24-bookworm

Une information précisant que l'accès Wi-Fi n'est pas disponible apparait en bas de la fenêtre. Il faudra alors définir une adresse IP statique (IP fixe). Notez que depuis Bookworm le système de gestion réseau a changé, c'est désormais **NetworkManager** qui est utilisé. Cette nouvelle manière d'attribuer une adresse IP fixe au Raspberry Pi est disponible avec ce tutoriel : <https://www.electro-info.ovh/index.php?id=290>

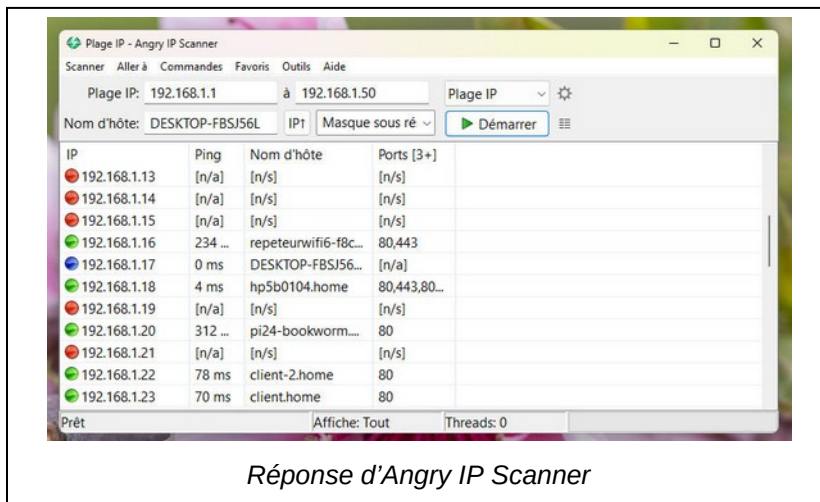
Revenons à l'accès Wi-Fi. Ma carte est maintenant en place avec son fichier de configuration `wpa_supplicant.conf` modifié lui permettant de rejoindre mon réseau local. Il me faut pour cela trouver son adresse IP. L'utilitaire **Angry IP Scanner**, un logiciel libre de balayage de port utilisé pour rechercher la présence de périphériques informatiques connectés à un réseau TCP/IP va réaliser cette recherche. Il est disponible à l'adresse <https://angryip.org/>.



Accès Wi-Fi à la carte pi24-bookworm

Ma Box Internet ayant comme IP 192.168.1.1¹², je fais une recherche sur la plage 192.168.1.1 à 192.168.1.50

¹² Information disponible via la commande `ipconfig` lancée depuis le Terminal Windows (via le raccourci « Touche Windows + X », puis sélectionner Terminal Windows).

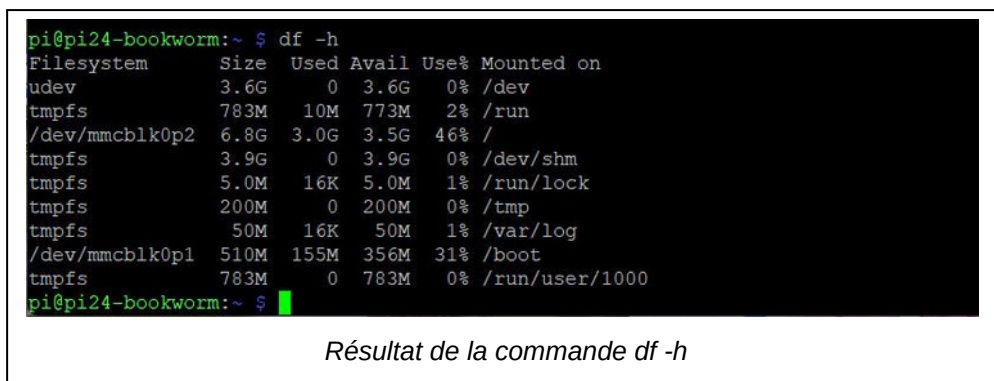


Dans mon cas, la carte a comme adresse IP 192.168.1.20. Il ne me reste plus qu'à utiliser PuTTY pour y accéder et réaliser quelques manipulations.

Vérification de l'espace attribué au système

La taille de l'espace attribué au système sur la carte *MicroSD* est obtenu via la commande :

- `df -h`



Mises à jour des dépôts et des versions

Exécution des deux commandes suivantes :

- `sudo apt-get update`
- `sudo apt-get upgrade`

Version de Raspbian

Pour trouver la version de Raspbian, il suffit de saisir la commande suivante :

- `cat /etc/os-release`

Le retour de la requête confirme la version 12 (bookworm)

```
pi@pi24-bookworm:~ $ cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 12 (bookworm)"
NAME="Debian GNU/Linux"
VERSION_ID="12"
VERSION="12 (bookworm)"
VERSION_CODENAME=bookworm
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
pi@pi24-bookworm:~ $
```

Voir sa version système sous Raspbian OS

Versions de Python

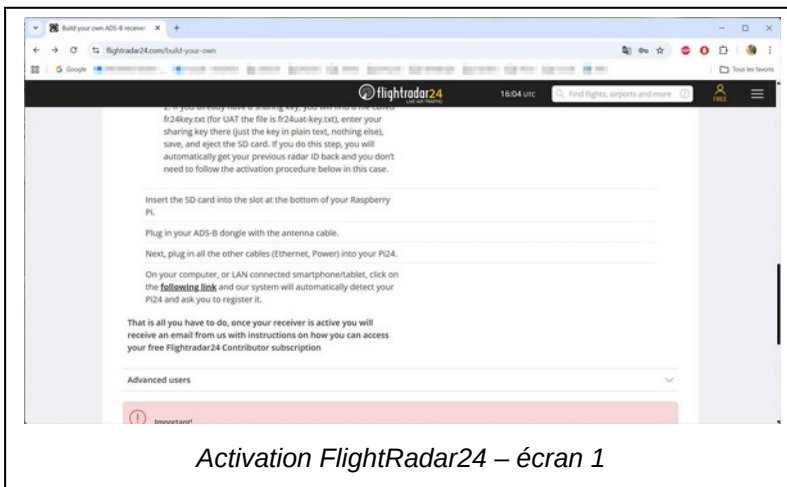
Les commandes suivantes permettent de connaître les versions installées du logiciel *Python* :

- `python --version`
- `python3 --version`

Pour ma part, le retour de ces requêtes ont indiqué la version 3.11.2

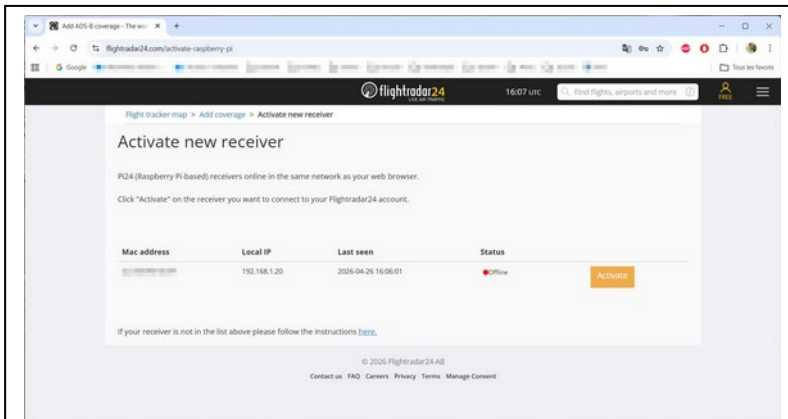
Activation des échanges

Mon flightstick maintenant en place, il reste encore à activer les échanges avec le site FlightRadar24. Depuis le bas de la page <https://www.flightradar24.com/build-your-own> dirigeons nous vers le lien « **following link** ».



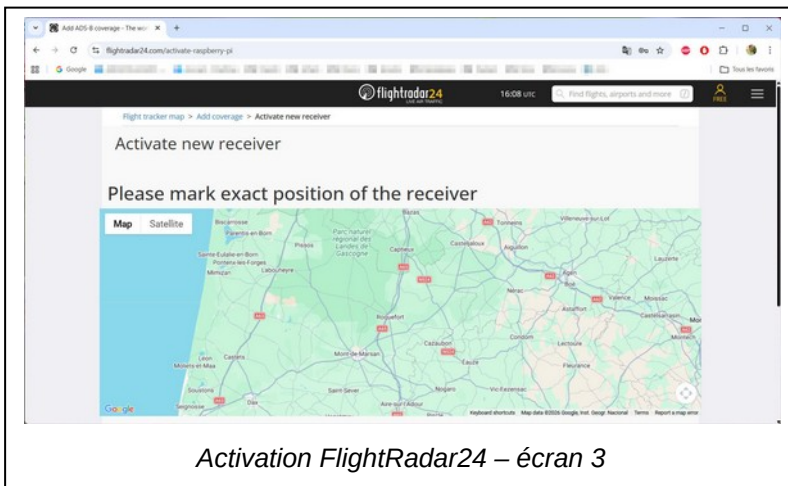
Activation FlightRadar24 – écran 1

Tout semble correct. Ma (future) station de réception est reconnue. Il ne me reste plus qu'à l'activer.



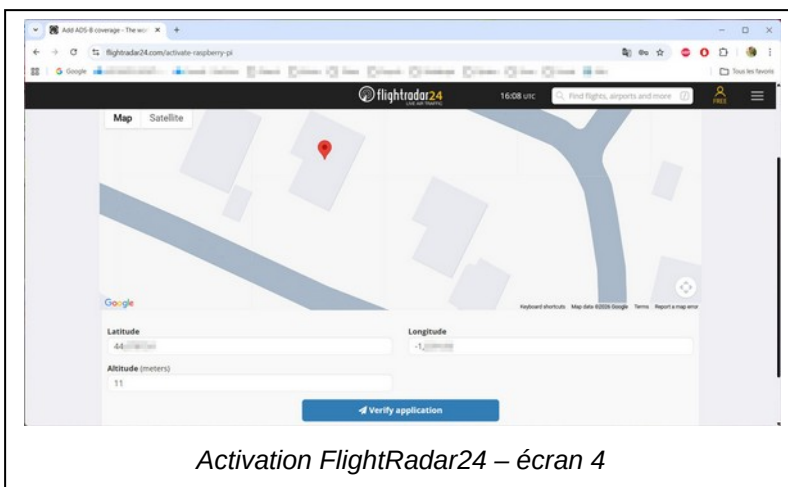
Activation FlightRadar24 – écran 2

Je dois pour cela définir ma position exacte.



Activation FlightRadar24 – écran 3

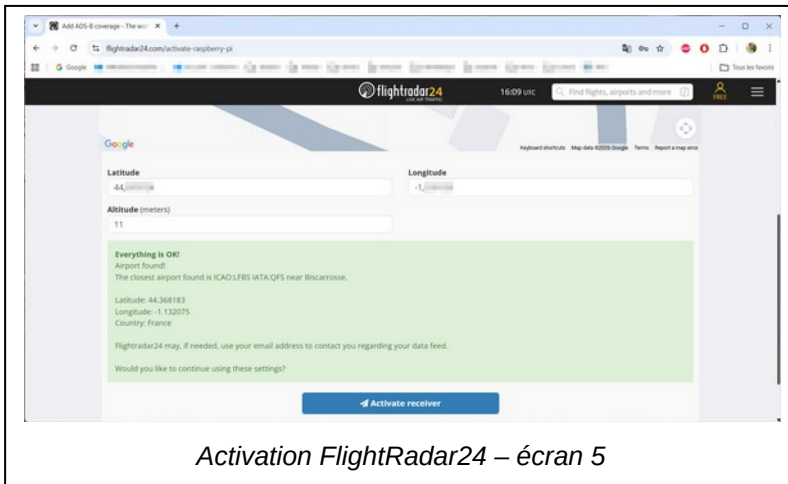
Une fois celle-ci repérée, mes coordonnées GPS (latitude, longitude) ainsi que mon altitude me sont retournées.



Activation FlightRadar24 – écran 4

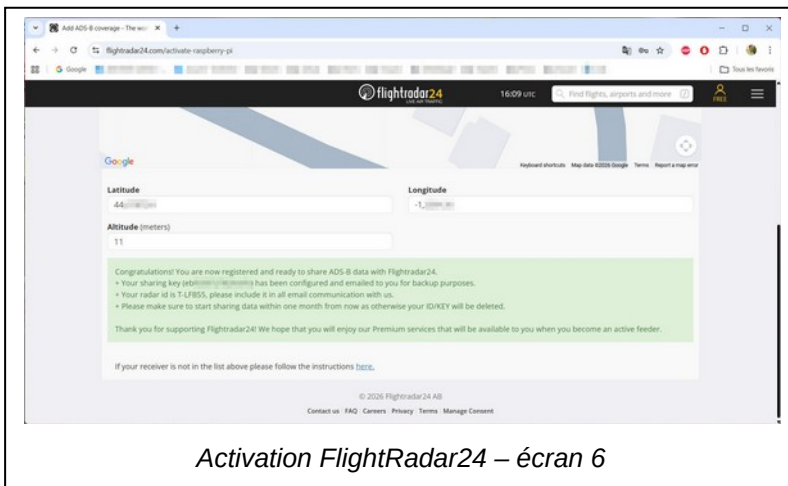
Je clique alors sur « **Verify application** ».

Tout est OK, le site me retourne, pour information, l'aéroport le plus proche de mon habitation.



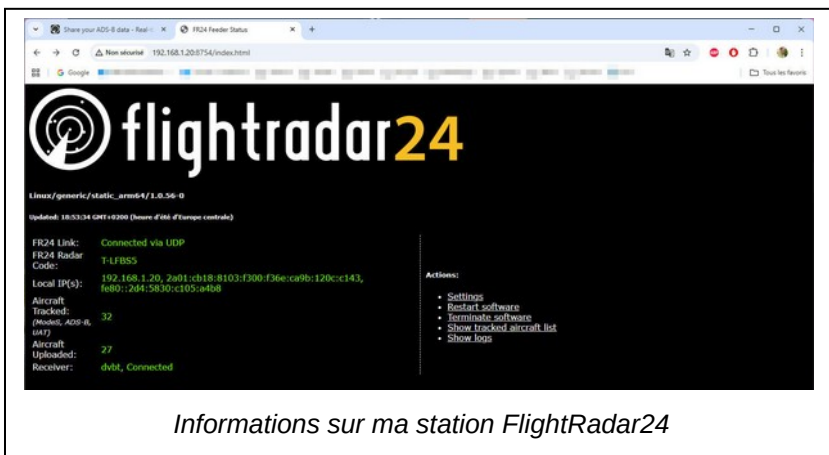
Activation FlightRadar24 – écran 5

Je choisis alors « **Activate receiver** ».
Je reçois en retour le nom de radar qui m'a été attribué.



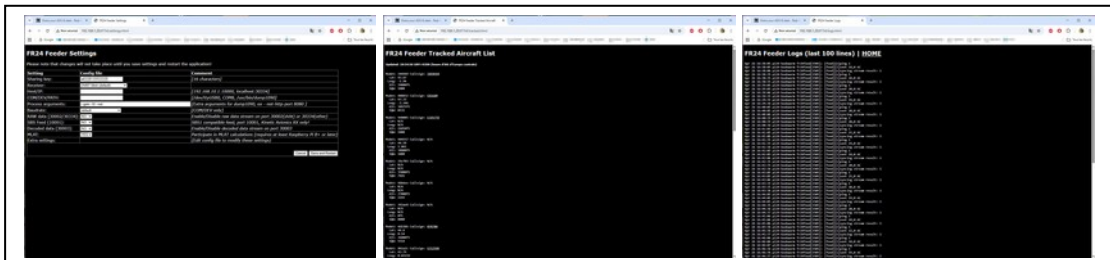
Activation FlightRadar24 – écran 6

En me déplaçant à l'adresse http://ip@my_pi:8754/index.html (dans mon cas : <http://192.168.1.20:8754/index.html>), il est possible de vérifier l'état d'activité de sa station.



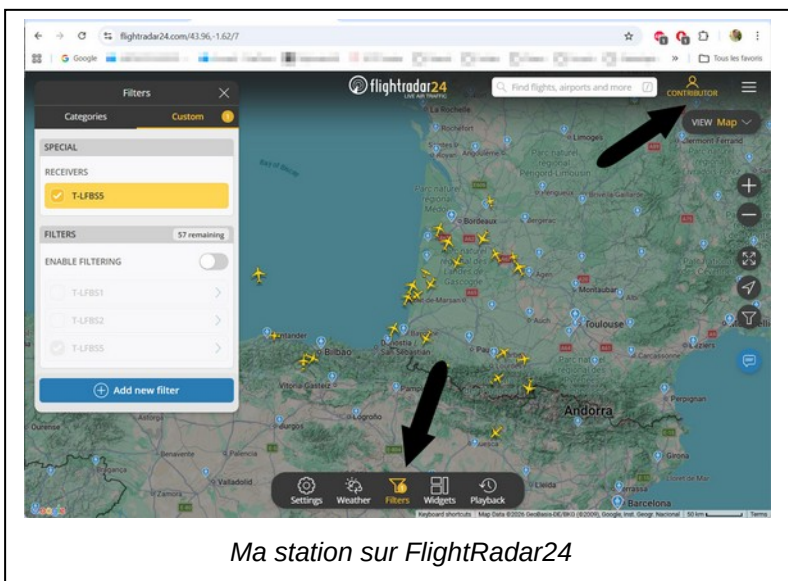
Informations sur ma station FlightRadar24

Plus de détails sont disponibles en utilisant les choix proposés à droite de l'écran.



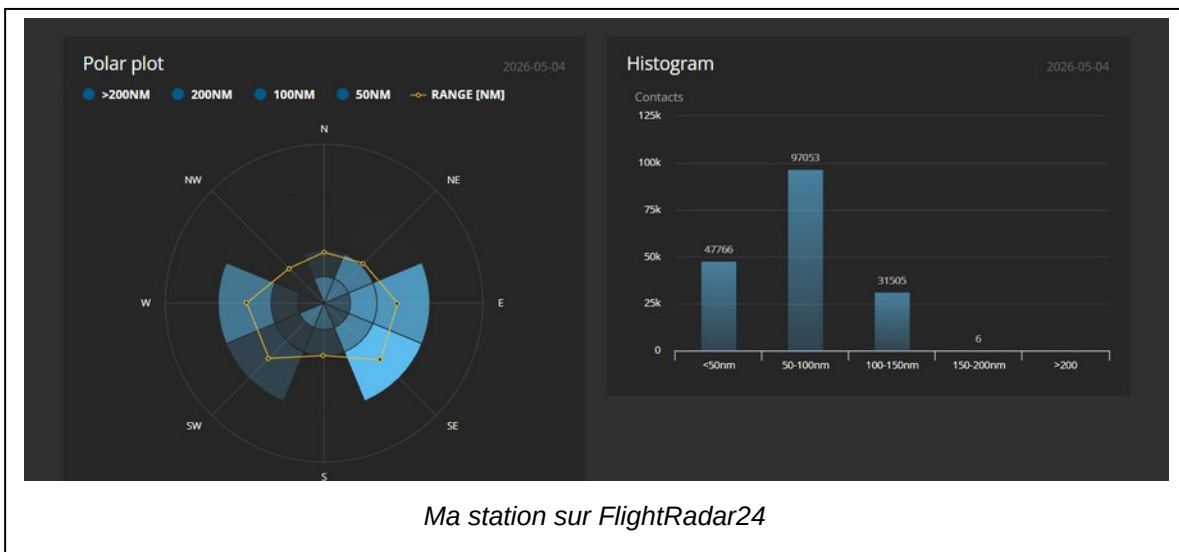
Informations sur ma station FlightRadar24

Sur le site, la mise en place d'un filtre permet de visualiser l'étendue de sa zone de réception. Cette manipulation est possible sans abonnement dans la mesure où le site vous a reconnu comme contributeur.

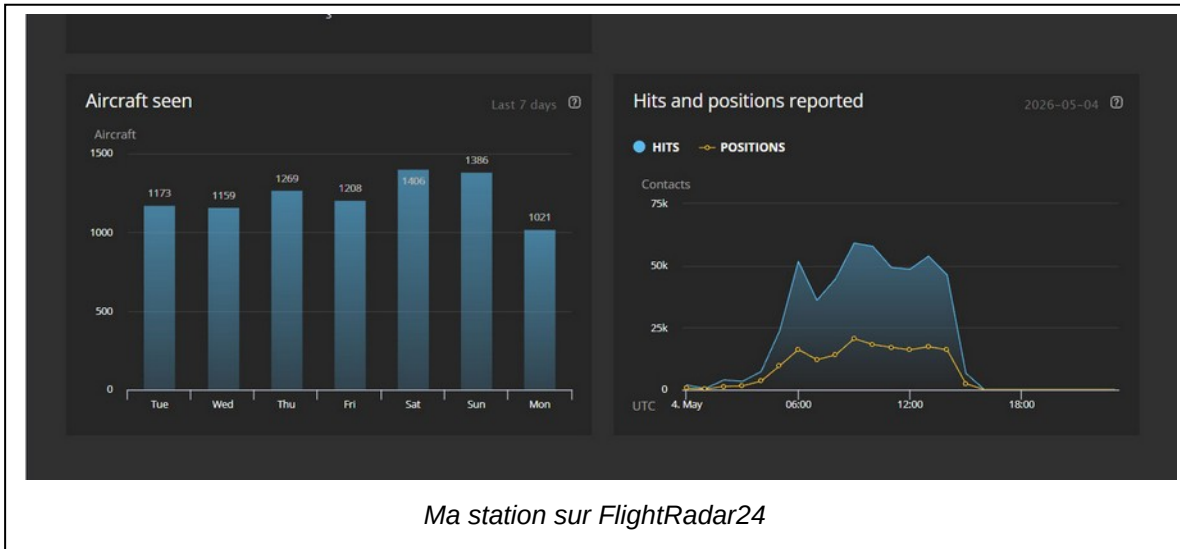


Ma station sur FlightRadar24

Depuis l'accès **Contributor**, le choix *My Data Sharing*, puis *Show Statistics* offer divers outils précisant la qualité de votre réception.



Ma station sur FlightRadar24



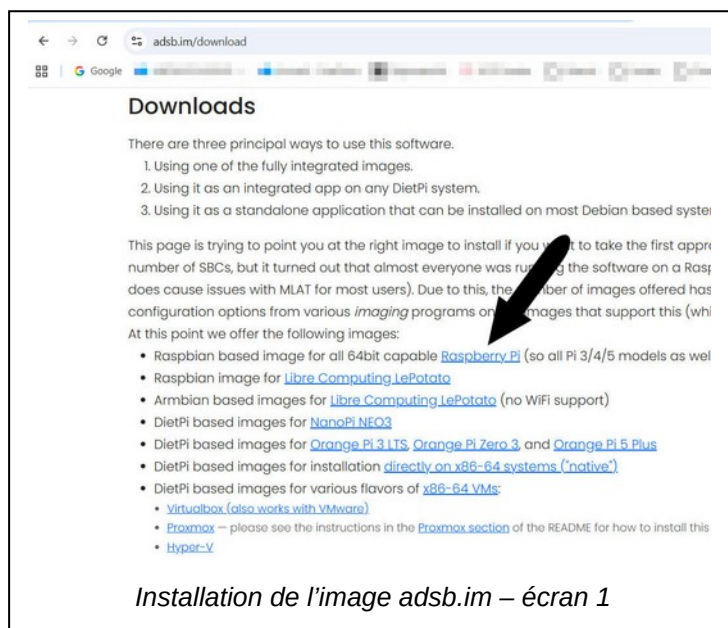
Ma station sur FlightRadar24

Raspberry Pi – OpenSky Network - Image ADSB.im

Après l'installation de l'image Pi24 proposée par Flightradar24, nous disposons déjà d'une station ADS-B fonctionnelle, capable de recevoir les avions visibles depuis notre antenne et de partager automatiquement ces données avec son réseau mondial. Mais cette première approche n'est pas la seule possible. À côté des grands services commerciaux de suivi aérien, il existe aussi des réseaux plus ouverts, à vocation technique, collaborative ou scientifique. C'est le cas d'OpenSky Network, qui se présente comme une plateforme de données ouvertes sur le trafic aérien, orientée notamment vers la recherche, et collectant des données ADS-B, Mode-S, ADS-C, FLARM ou encore VHF. Pour y contribuer depuis un Raspberry Pi, OpenSky propose une image prête à l'emploi permettant de transformer notre Raspberry Pi en station de réception multi-réseaux. Son principal avantage est de ne plus limiter notre station à une seule plateforme : une même installation peut alimenter OpenSky et, si on le souhaite, d'autres agrégateurs ADS-B, le tout au moyen d'un assistant de configuration accessible depuis un navigateur, sans passage obligatoire par la ligne de commande.

Mise en place de l'image adsb-im

Tout comme FlightRadar24, Opensky¹³ propose, pour la Raspberry Pi, l'utilisation d'une image disponible depuis le site <https://adsb.im/home> à flasher sur une carte microSD. Celle-ci nous arrive sous la forme d'un fichier compressé nommé : `adsb-im-raspberrypi64-pi-2-3-4-5-v3.0.11.img.xz`

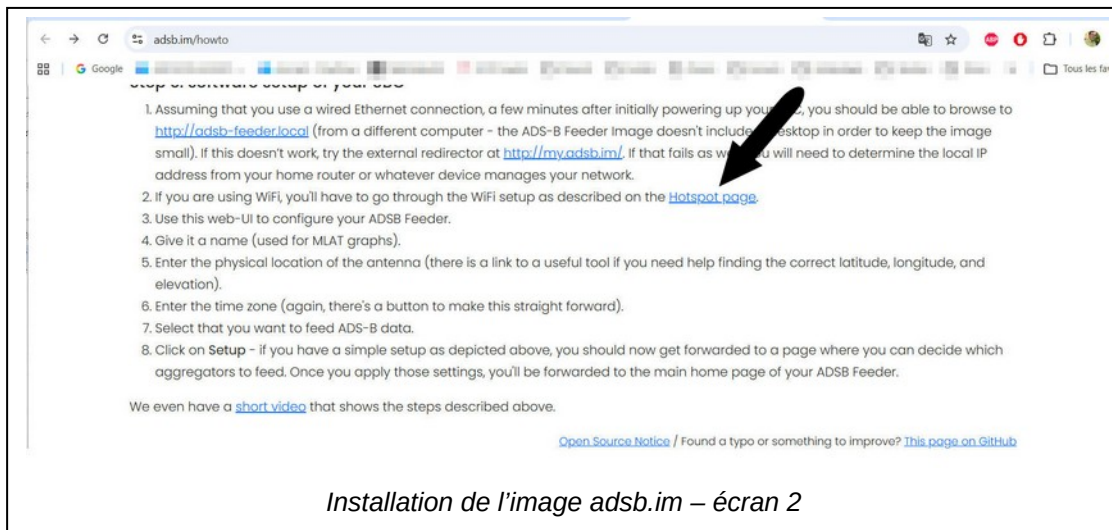


La version de WinZip en version d'essai de 21 jours gratuit, disponible à l'adresse <https://www.winzip.com/fr/download/winzip/> permet sa décompression. Il ne nous reste plus ensuite qu'à utiliser **Win32 Disk Imager** pour la graver sur la micro carte.

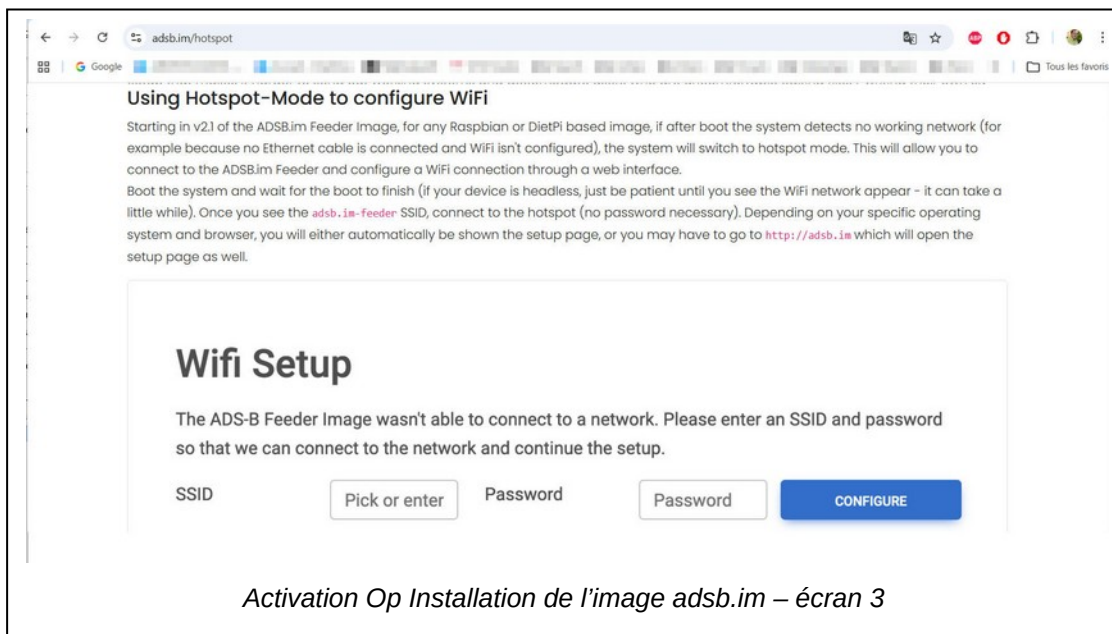
Activation des échanges

Après s'être créé un compte sur le site. La marche à suivre est disponible depuis la page <https://adsb.im/howto>.

¹³ <https://opensky-network.org/feed>

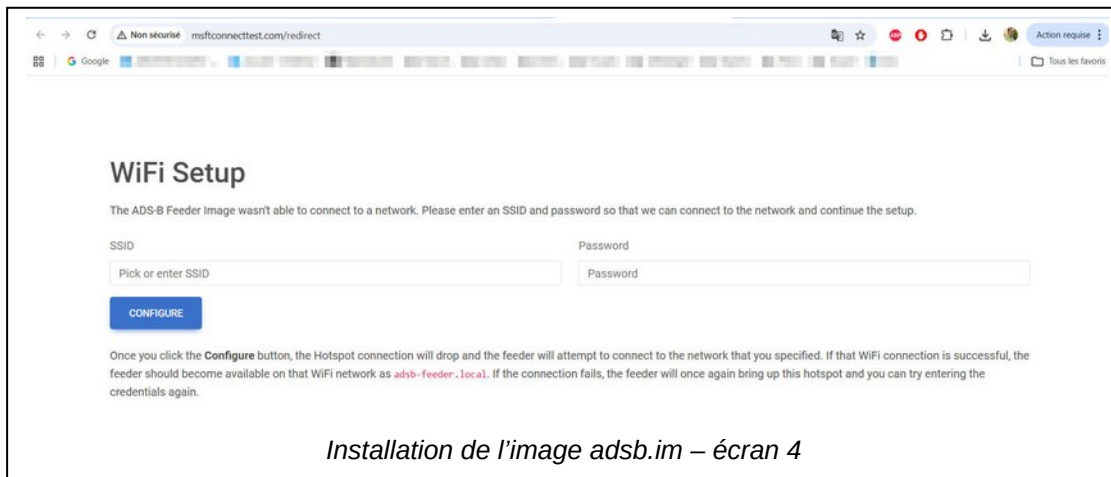


Privilégiant l'accès Wi-Fi, je suis le lien indiqué vers la page <https://adsb.im/hotspot>.



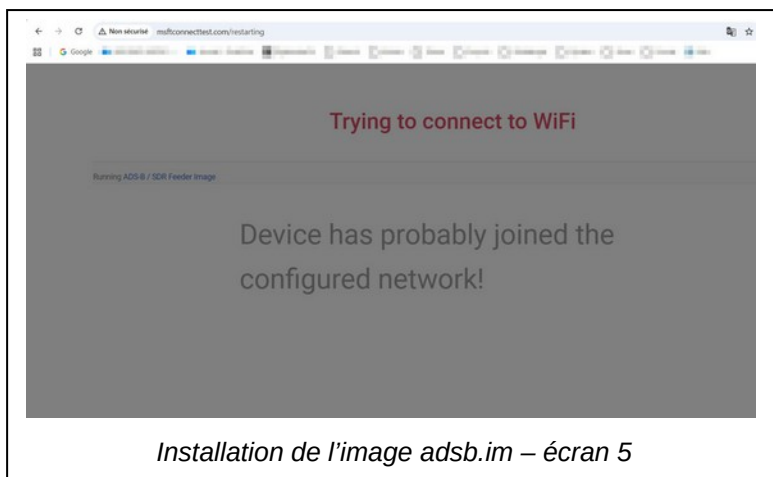
Cette page explique qu'il faut maintenant démarrer le système Raspberry Pi et attendre quelques minutes afin que s'achève la séquence de démarrage. Le système ne détectant aucun réseau fonctionnel, par exemple parce qu'aucun câble Ethernet n'est connecté, et constatant que le Wi-Fi n'est pas encore configuré, il va basculer en mode point d'accès.

Dans mon cas, le Raspberry Pi fonctionne sans écran ni clavier, plusieurs minutes ont été nécessaire avant de voir apparaître dans mon environnement un nouveau Wi-Fi avec le SSID `adsb.im-feeder`.

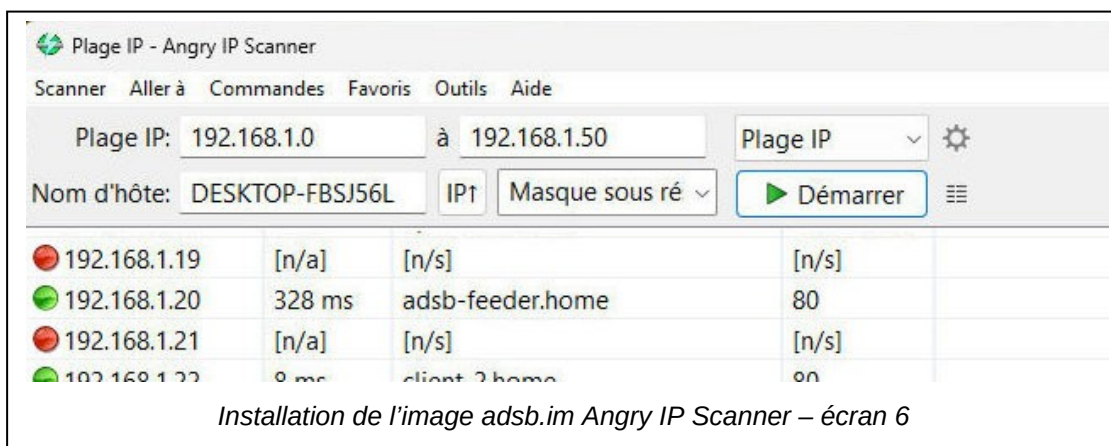


La connexion se fait sans aucun mot de passe. Dans mon cas, la page de configuration s'est ouverte automatiquement dans mon navigateur. Si ce n'est pas le cas, rendez-vous à l'adresse <http://adsb.im> qui affiche également la page de configuration.

Une fois les informations d'identification inscrite le choix Configure, ouvre une fenêtre d'attente qui s'est, dans mon cas, achevée sur un succès.



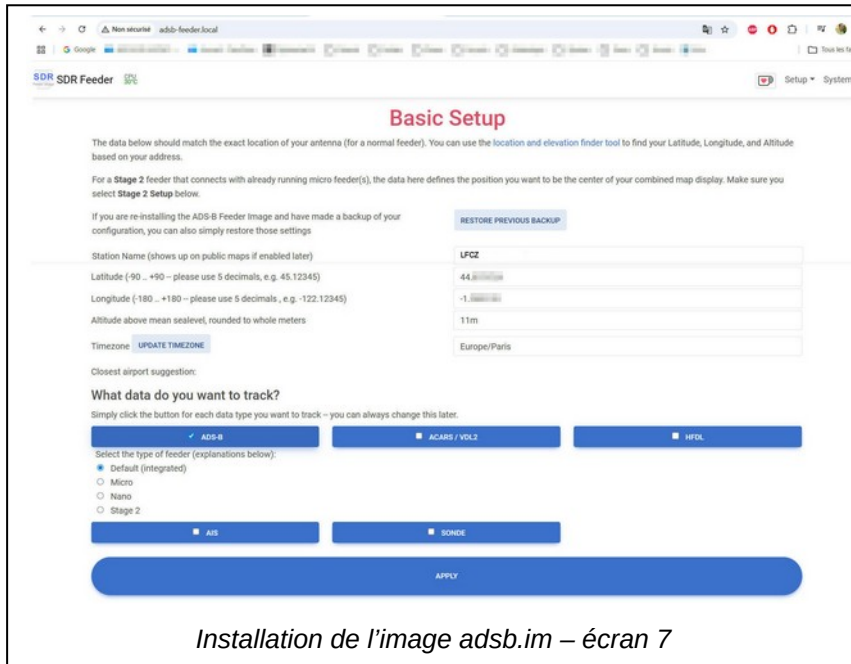
Le logiciel Angry IP Scanner n'en donne une première confirmation.



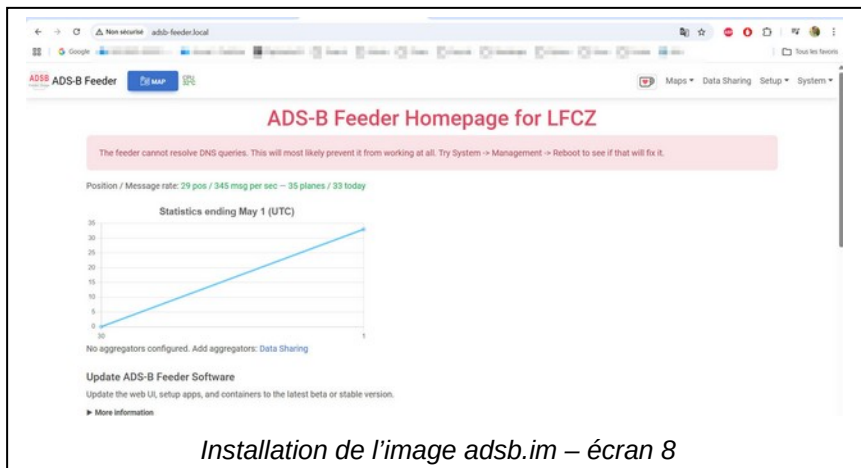
Accédant à la page locale `adsb-feeder.local`, il s'agit maintenant d'entrer les données exactes de géolocalisation de notre antenne. Ayant déjà ces données, je n'ai pas à utiliser le lien proposé (*location and elevation finder tools*).

Je prend comme nom de station le code AITA de l'aérodrome le plus proche de mon domicile.

Le fuseau horaire ayant été détecté automatiquement, je le maintient à Europe/Paris.

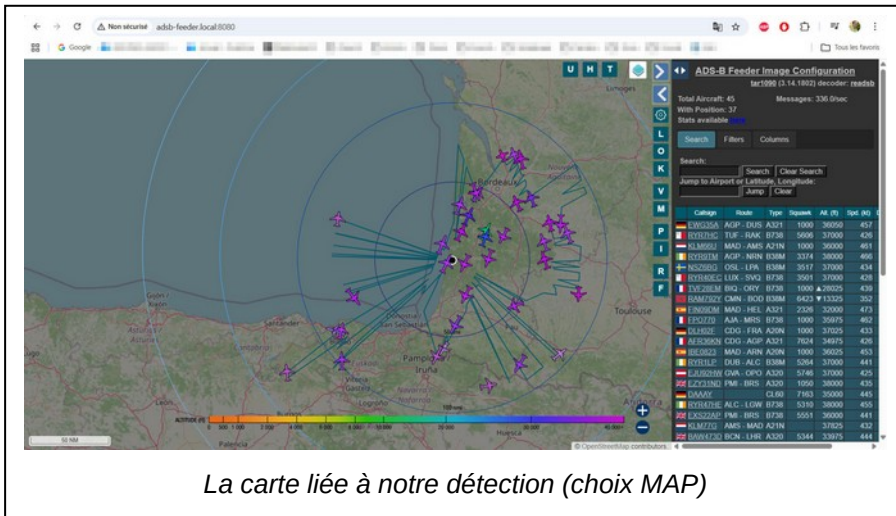


Finalement, et avant de choisir APPLY, j'indique ADS-B comme données à suivre.

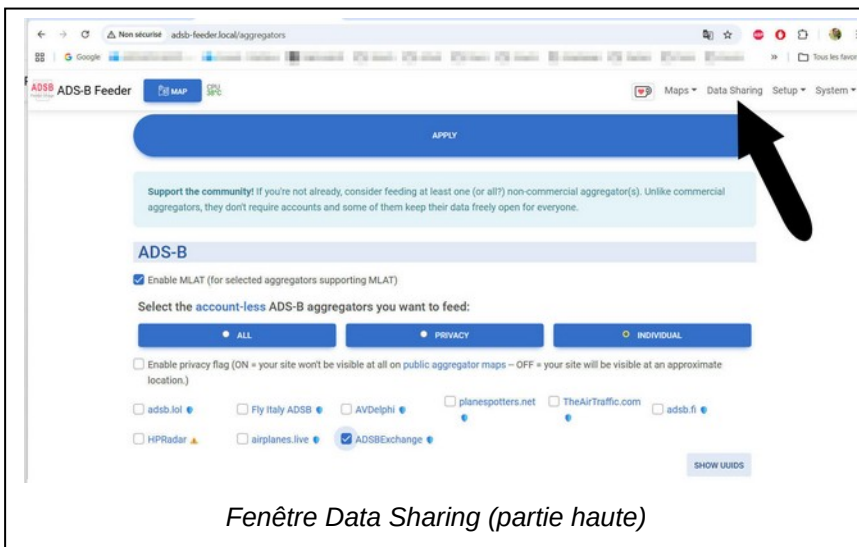


Quelques instants plus tard, la page d'accueil de notre station apparaît. Comme signalé en encart, le message d'erreur concernant la résolution des requêtes DNS disparaîtra au prochain redémarrage.

Le choix MAP (bouton bleu en haut à gauche de l'interface) affiche la carte liée à notre réception.



L'un des avantages de cette distribution **ADSB.im** est qu'elle ne limite pas notre station à un seul réseau de suivi aérien mais transforme notre Raspberry Pi en station ADS-B multi-réseaux, tout en laissant à l'utilisateur le choix des plateformes qu'il souhaite alimenter. À partir du menu *Data Sharing*, disponible en haut à droite de l'interface, il est possible de choisir les plateformes vers lesquelles seront envoyées nos données ADS-B.



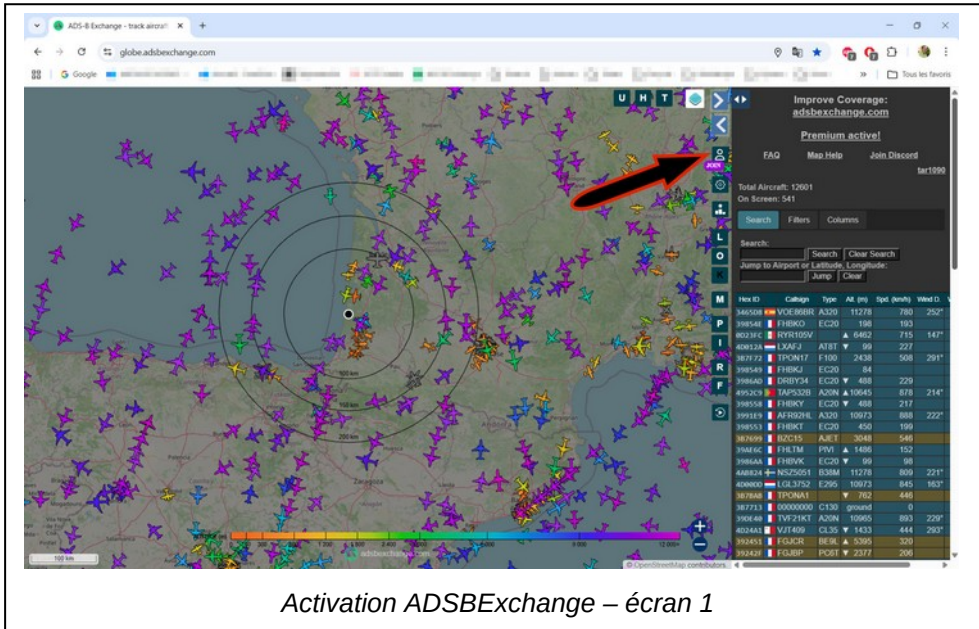
Sur la partie haute de l'écran, il est possible de choisir, individuellement ou en bloc, des agrégateurs ne nécessitant pas de compte utilisateur. Il y a là des plateformes commerciales et non commerciales. Après avoir autorisé l'envoi de mes données MLAT vers les agrégateurs qui les prennent en charge, j'accepte d'envoyer mes données vers **ADSBExchange** avant de me diriger vers leur site.

Lien avec ADSBExchange

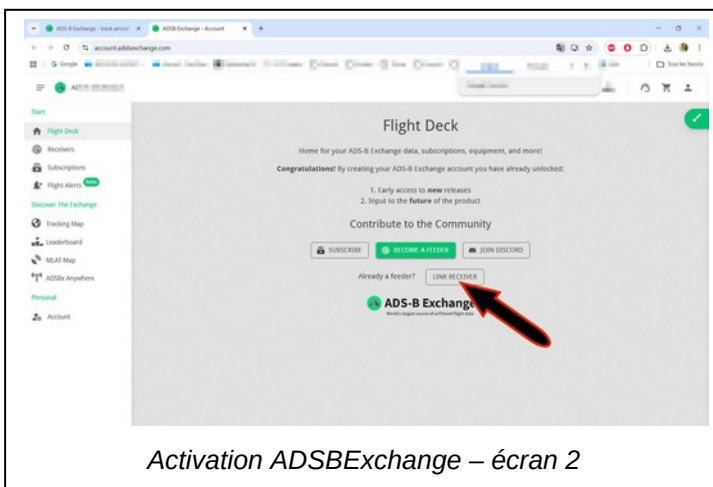
ADSB Exchange est une plateforme collaborative à base communautaire, aujourd'hui exploitée dans un cadre commercial. Il est alimenté par un vaste réseau de récepteurs ADS-B, Mode S et MLAT installés par des passionnés dans le monde entier. Sa particularité est de revendiquer une diffusion des données « non filtrée », y compris pour les appareils militaires, privés ou commerciaux lorsque leurs signaux sont reçus. Depuis janvier 2023, ADS-B Exchange appartient

à JETNET, société spécialisée dans les données et l'intelligence de marché aéronautiques, et propose aussi des produits commerciaux comme des API ou des données historiques.

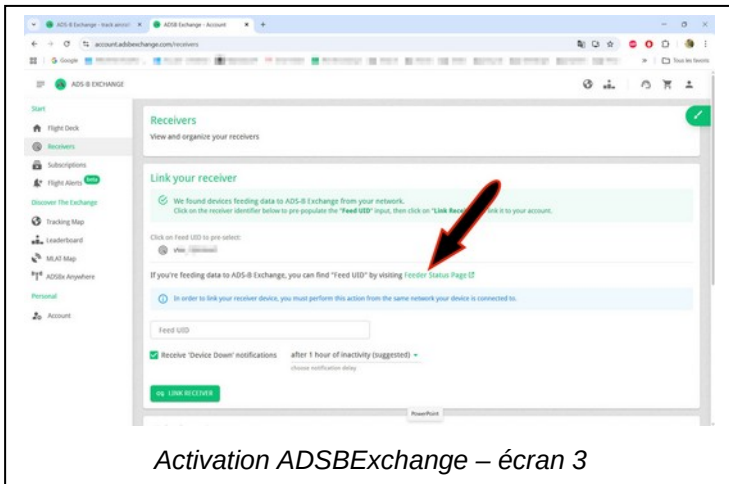
Depuis la page d'accueil la procédure débute en choisissant **Join**.



Depuis la page **Flight Deck**, cliquez ensuite sur **Link Receiver**.

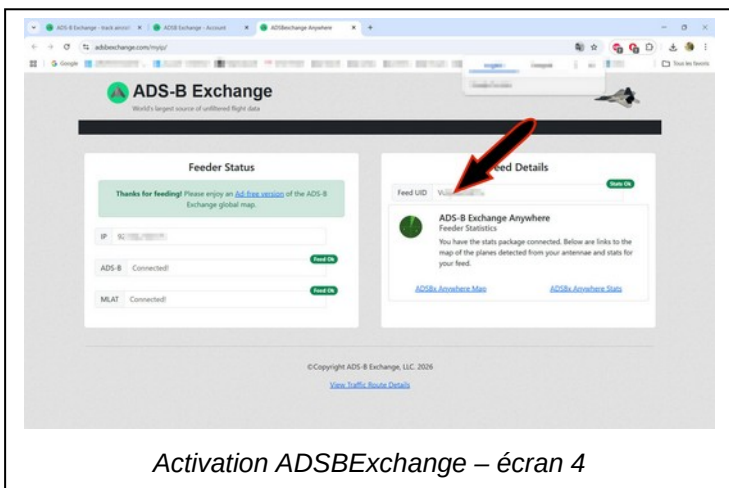


Une nouvelle page s'ouvre. Elle va vous permettre de saisir le numéro d'identification unique de notre station, appelé UID pour *Unique Identifier*. Avant de le renseigner, cliquez sur **Feeder Status Page** afin de retrouver ce numéro.



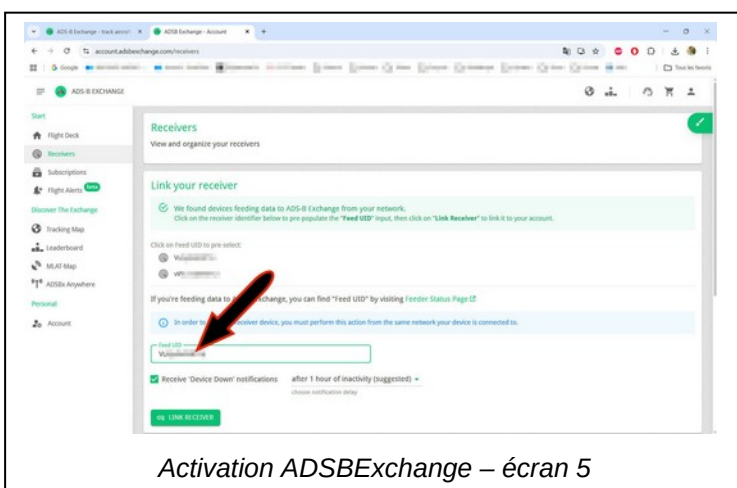
Activation ADSBExchange – écran 3

Il apparait dans la fenêtre **Feed Details**.



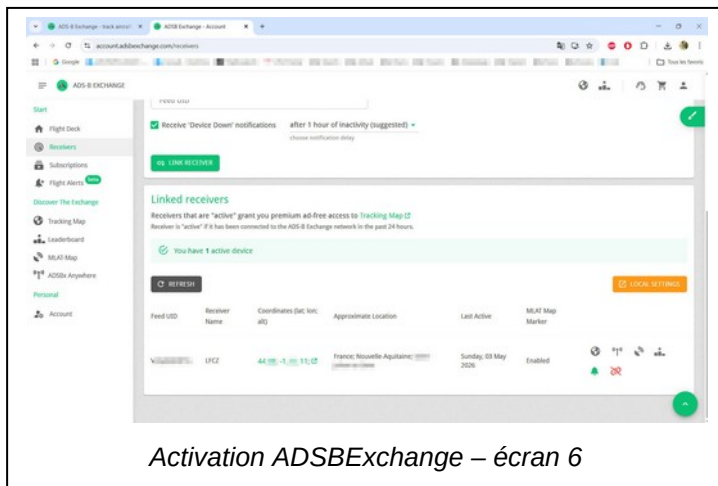
Activation ADSBExchange – écran 4

Il ne reste plus ensuite qu'à le reporter dans la zone de saisie **Feed UID** de la page précédente.



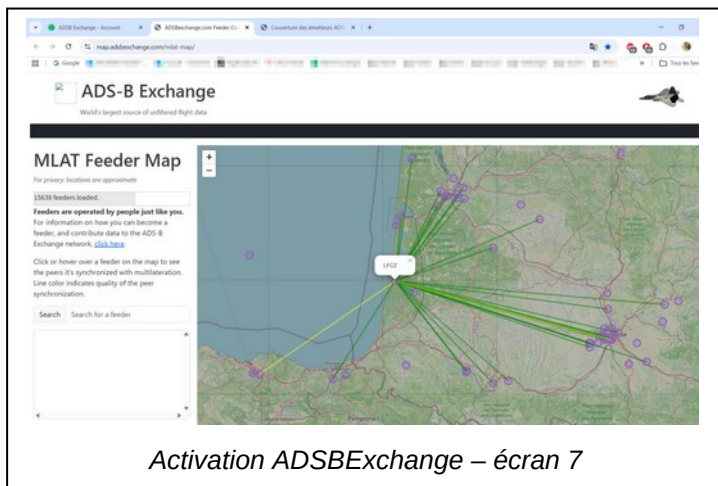
Activation ADSBExchange – écran 5

Celui-ci apparaît alors au bas de cette même page dans la fenêtre d'informations **Linked receivers**, confirmant que votre récepteur est bien pris en compte.



Alimentation MLAT

La page qui s'affiche en suivant le lien : <https://map.adsbexchange.com/mlat-map/> est particulièrement intéressante. Elle montre la **couverture MLAT** de notre station. Dans mon cas, le récepteur est bien reconnu sous l'identifiant **LFCZ**, ce qui confirme son intégration au réseau.



Le sigle **MLAT** vient de l'anglais *Multilateration*, ou **multilatération** en français. Cette technique permet d'estimer la position d'un avion qui émet des signaux **Mode S**, mais qui ne transmet pas directement sa position GPS. C'est le cas de certains appareils militaires, d'avions anciens, ou encore d'aéronefs dont le transpondeur fournit des informations d'identification et d'altitude, mais pas de latitude ni de longitude.

Avec l'ADS-B classique, le principe est simple : l'avion connaît sa position grâce à son GPS embarqué, puis il la diffuse directement dans ses messages radio émis sur 1090 MHz. Un seul récepteur au sol peut alors l'afficher sur une carte, puisqu'il reçoit déjà la position complète de l'appareil.

Avec le Mode S seul, la situation est différente. Le signal radio est bien reçu, mais il ne contient pas toujours la position de l'avion. Pour la retrouver, ADS-B Exchange utilise alors les réceptions simultanées de plusieurs stations au sol. Chaque station capte le même message, mais pas exactement au même instant. Ces très petites différences de temps d'arrivée du signal sont comparées par les serveurs du réseau.

Il ne s'agit donc pas d'une triangulation au sens classique du terme. En triangulation, on mesure des angles. En multilatération, on exploite des différences de temps, donc des différences de distance. Comme les ondes radio se propagent à la vitesse de la lumière, un écart minime dans le temps d'arrivée correspond déjà à une différence mesurable de distance entre l'avion et les récepteurs.

Pour que le calcul soit possible, plusieurs conditions doivent être réunies. Les récepteurs au sol doivent être assez nombreux autour de la zone suivie. Leurs positions géographiques doivent être connues avec précision. Leur horloge doit également être correctement synchronisée, car le calcul repose sur des écarts de temps extrêmement faibles. Enfin, le serveur central doit pouvoir comparer les messages reçus par les différentes stations et en déduire la position la plus probable de l'appareil.

La page **MLAT Feeder Map** visualise justement cette coopération entre récepteurs. Les cercles violets représentent des stations contributrices du réseau ADS-B Exchange. Pour des raisons de confidentialité, leurs emplacements sont volontairement approximatifs, comme le précise le message affiché à gauche de la carte.

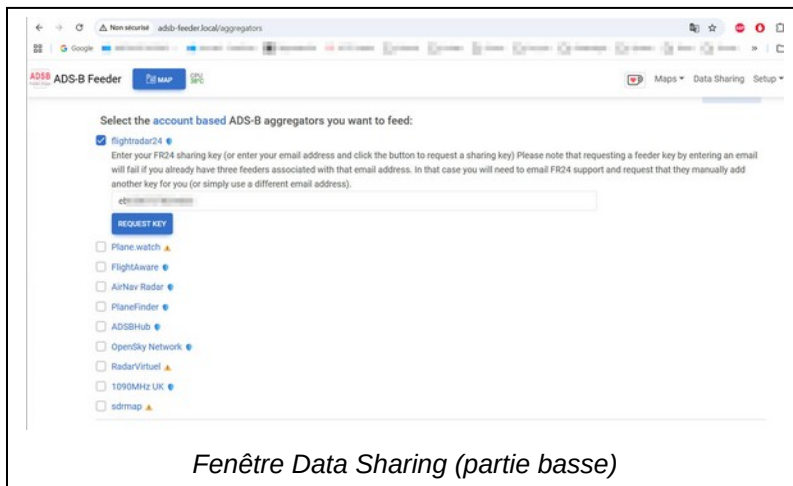
En sélectionnant ma station, identifiée ici par **LFCZ**, la carte affiche les autres récepteurs avec lesquels elle est synchronisée pour participer aux calculs MLAT. Les traits reliant les stations montrent ces liens de synchronisation. Leur couleur renseigne sur la qualité de cette synchronisation : plus la liaison est bonne, plus elle pourra être utile pour les calculs de position. Sur cette capture, on constate que ma station est reliée à de nombreux récepteurs situés dans le Sud-Ouest, mais aussi plus loin vers Toulouse, Bordeaux, les Pyrénées et le nord de l'Espagne.

Lien avec Flightradar24

Flightradar24 est l'un des sites de suivi aérien les plus connus du grand public. Né en Suède en 2006 comme projet de passionnés, il s'appuie aujourd'hui sur un vaste réseau mondial de récepteurs ADS-B, complété par d'autres sources de données, afin d'afficher en temps réel la position de nombreux avions civils. Aujourd'hui, le site revendique plus de 50 000 récepteurs ADS-B répartis dans le monde.

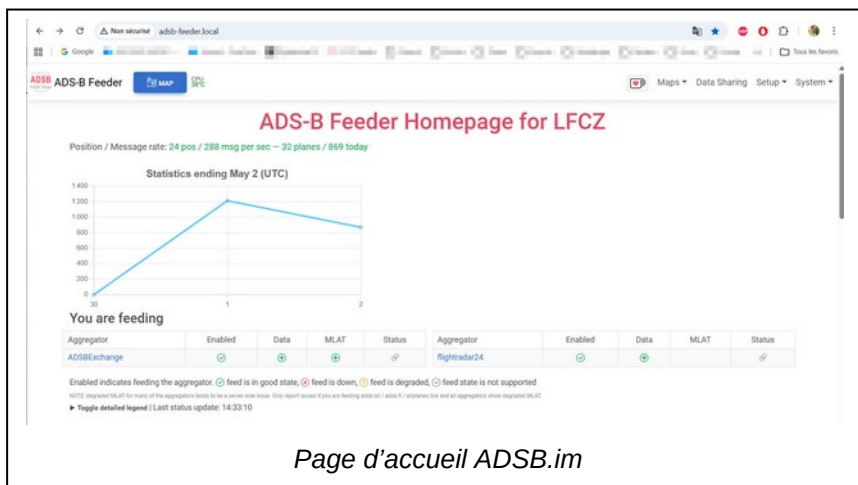
Même s'il est alimenté en grande partie par une communauté de stations contributrices, à la différence d'**ADSBExchange**, **Flightradar24** est clairement un service commercial. L'accès de base reste largement gratuit pour l'utilisateur, mais le site propose aussi des abonnements payants, des fonctions avancées, ainsi qu'une offre d'API donnant accès à des données aéronautiques en temps réel ou historiques.

Revenons donc maintenant sur la page d'accueil de notre station **ADSB.im** sur laquelle nous avons autorisé l'envoi de nos données vers **ADSBExchange**. Sur la partie basse, une seconde liste regroupe les agrégateurs associés à un compte. Choisissons donc **Flightradar24** en cochant le service. L'interface déploie alors une zone destinée à saisir l'identifiant que le site a préalablement créé. Je reprend donc ici celui utilisé dans nos manipulations Windows.



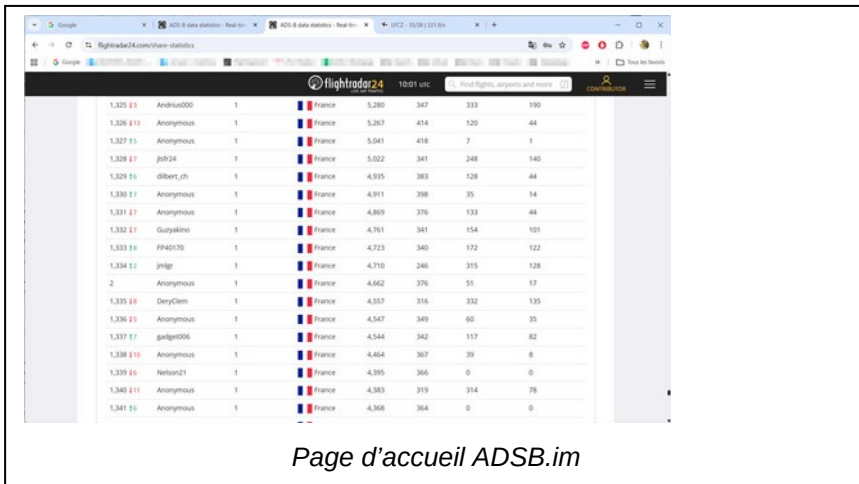
Fenêtre Data Sharing (partie basse)

De retour sur la page d'accueil, l'interface s'est enrichie de nouvelles informations. Elle affiche désormais les statistiques de réception de notre station ainsi que l'état des flux envoyés vers les agrégateurs sélectionnés. Lors des premières minutes, certains indicateurs peuvent apparaître en jaune ou en rouge, le temps que les connexions se stabilisent et que les agrégateurs valident les données reçues. Dans mon cas, tous sont passés au vert dans l'heure qui a suivi.



Page d'accueil ADSB.im

FlightRadar24 m'a attribué comme nom de station T-LFBS5. A ce titre je me suis donné comme nom d'utilisateur FP40170 (en lieu et place de Anonymous). Sur leur page de statistiques : <https://www.flightradar24.com/share-statistics>, et après avoir restreint le pays à la France, je retrouve ma station et son rang.



The screenshot shows the ADSB.im website interface. At the top, there's a search bar and navigation tabs. Below, a table lists aircraft data. The table has columns for flight number, aircraft name, country, and several numerical columns representing data points.

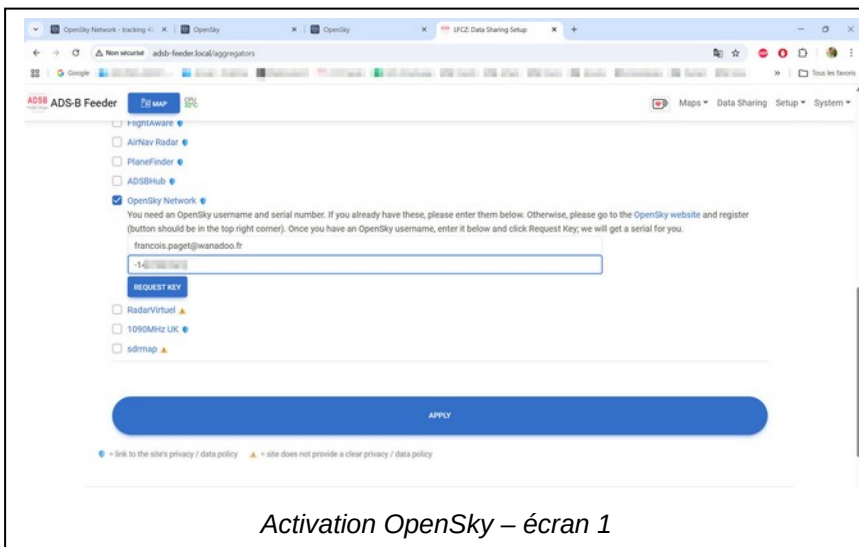
Flight	Aircraft	Country	Altitude	Speed	Heading	Vertical Rate	Horizontal Rate
1,325 13	Andru200	France	5,280	347	333	190	
1,326 113	Anonymous	France	5,267	414	120	44	
1,327 15	Anonymous	France	5,041	418	7	1	
1,328 17	js924	France	5,022	341	248	140	
1,329 16	dilbert_ch	France	4,935	383	128	44	
1,330 17	Anonymous	France	4,911	398	35	14	
1,331 17	Anonymous	France	4,869	376	133	44	
1,332 17	Guzykino	France	4,761	341	154	101	
1,333 16	FF40170	France	4,723	340	172	122	
1,334 12	jmgjr	France	4,710	246	315	128	
2	Anonymous	France	4,662	376	51	17	
1,335 18	DeryClem	France	4,557	316	332	135	
1,336 15	Anonymous	France	4,547	340	40	35	
1,337 17	galgen06	France	4,544	342	117	82	
1,338 110	Anonymous	France	4,464	367	39	8	
1,339 16	Nelson21	France	4,395	366	0	0	
1,340 11	Anonymous	France	4,383	319	314	78	
1,341 16	Anonymous	France	4,368	364	0	0	

Page d'accueil ADSB.im

Lien avec OpenSky network

Le lien avec le réseau **OpenSky Network** se configure de manière comparable à celui des autres agrégateurs nécessitant un compte utilisateur. Depuis la page **Data Sharing**, nous sélectionnons **OpenSky Network**, puis nous indiquons le nom d'utilisateur créé lors de notre inscription préalable sur le site du réseau.

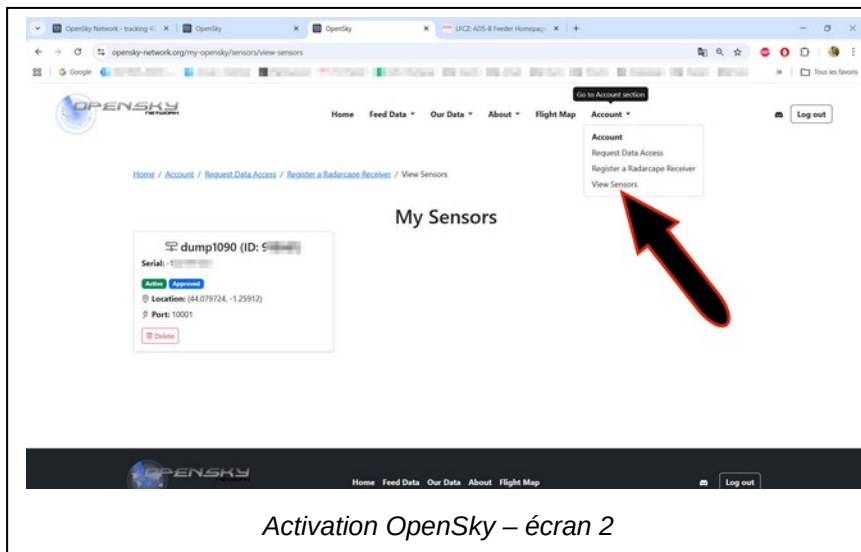
Après avoir cliqué sur **Request Key**, l'interface interroge le service distant. Une fenêtre d'attente apparaît quelques dizaines de secondes, puis disparaît. La clé nécessaire au partage des données est alors automatiquement ajoutée dans la zone de saisie. Il ne reste plus qu'à cliquer sur **Apply** pour enregistrer la configuration.



The screenshot shows the 'ADS-B Feeder' configuration page on the OpenSky Network website. It features a list of data sources with checkboxes. 'OpenSky Network' is checked. Below it, there's a text input field containing 'francois.paget@wanadoo.fr' and a 'REQUEST KEY' button. At the bottom, there's a large blue 'APPLY' button. A footer note indicates that some sites do not provide clear privacy policies.

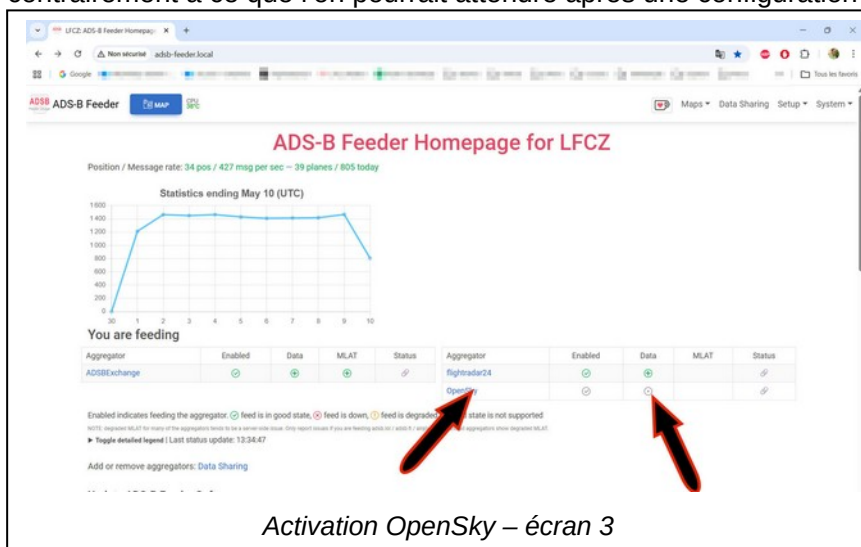
Activation OpenSky – écran 1

De retour sur le site **OpenSky Network**, le menu **Account / View Sensor** confirme que notre station est bien reconnue. L'activation semble donc avoir abouti.



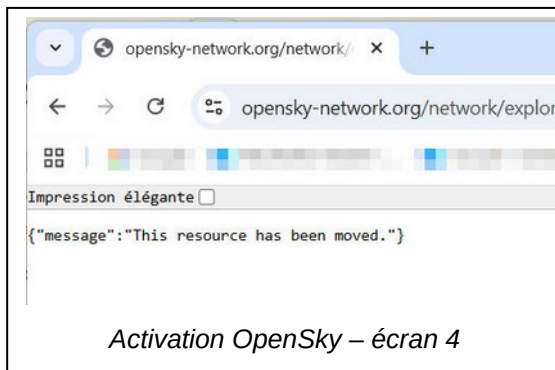
Quelques points restent toutefois à signaler :

- Dans l'interface **ADSB.im**, le voyant **Data** associé à OpenSky ne passe pas au vert, contrairement à ce que l'on pourrait attendre après une configuration réussie.



Le point au milieu du cercle du voyant Data signifie que l'interface locale n'arrive pas à valider le retour d'API d'OpenSky. Comme notre sensor est marqué "Online" sur leur portail, c'est juste un bug d'affichage de notre image : les données partent, mais le retour n'est pas lu.

- Par ailleurs, le lien proposé vers le site OpenSky renvoie un message d'erreur :
« {"message": "This resource has been moved."} »



C'est ici un problème de lien mort dans le code d'ADSB.im avec une absence de mise à jour des URL d'OpenSky. Il n'est donc pas possible d'utiliser ce raccourci pour atteindre notre tableau de bord "My Stations" sur leur site.

Enfin, je n'ai pas trouvé, sur le site **OpenSky**, de lien permettant de visualiser directement les données retransmises par ma station.

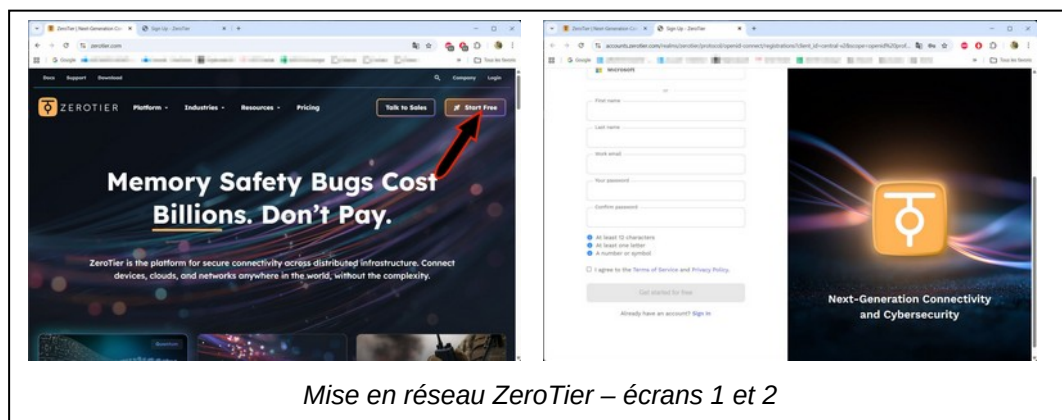
Prise de main à distance

Une information sur la page <https://adsb.im/using> attire mon attention, elle explique qu'il est possible de prendre la main à distance sur sa station à l'aide de l'application **ZeroTier**.

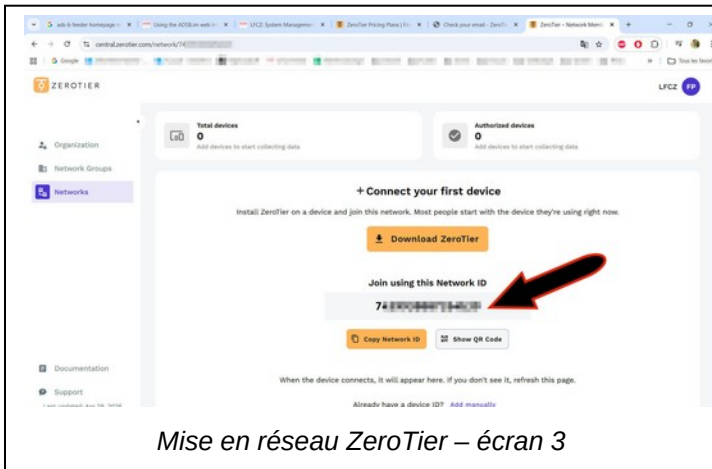


A noter que le lien donné sur cette page (ici en bleu) ne fonctionne pas.

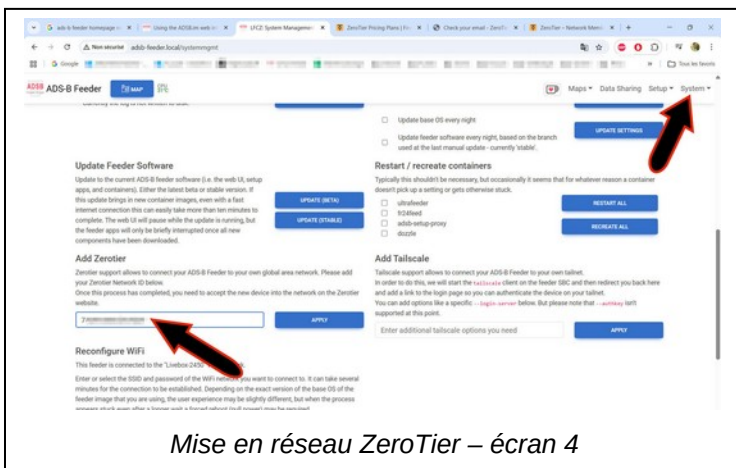
Cette application permet de créer un réseau privé virtuel entre plusieurs appareils, même lorsqu'ils se trouvent sur des réseaux Internet différents. Elle installe une interface réseau virtuelle et permet à chaque machine autorisée de communiquer comme si elle appartenait au même réseau local. Les échanges sont chiffrés de bout en bout et les connexions s'établissent directement entre appareils chaque fois que cela est possible.



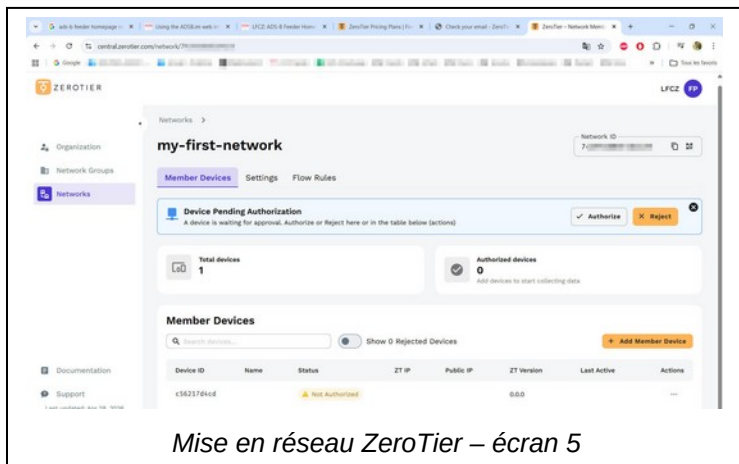
L'image ADSB.im que nous utilisons intégrant cette possibilité, la première étape consiste donc à créer un compte sur le site <https://www.zerotier.com/> puis à choisir l'offre **Personal**, gratuite. En retour, nous obtenons un numéro d'identification réseau.



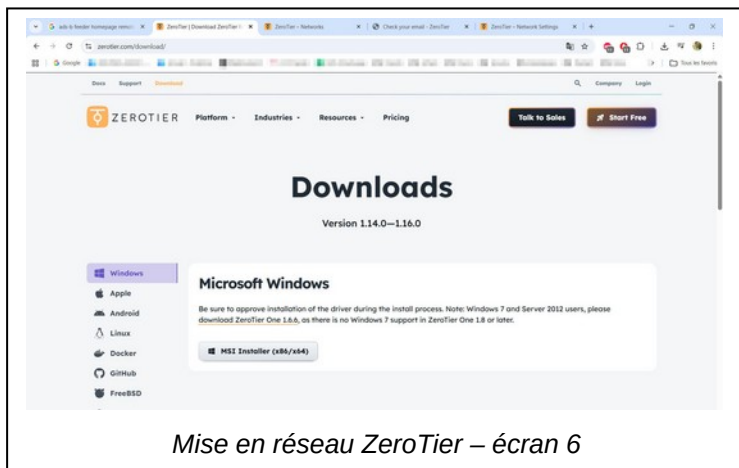
Celui-ci doit être reporté dans la zone de saisie **Add Zerotier** disponible dans l'onglet **Management** lui-même accessible par le choix **System** de la page d'accueil de notre station.



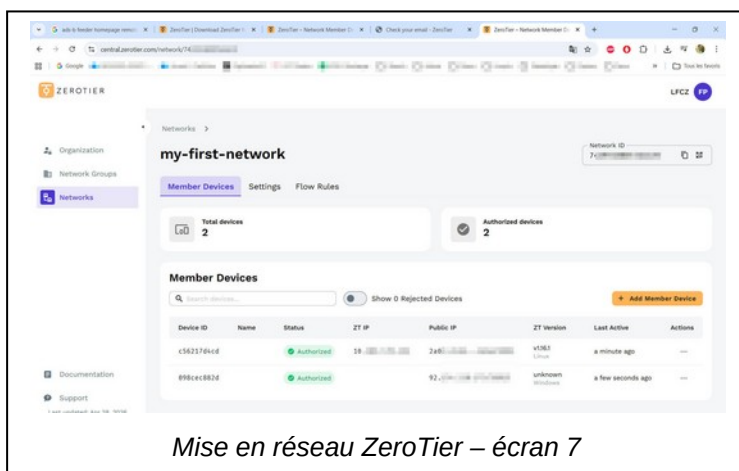
De retour sur le site **ZeroTier**, nous constatons que notre station est désormais reconnue comme un **Member Device** dans **my-first-network**.



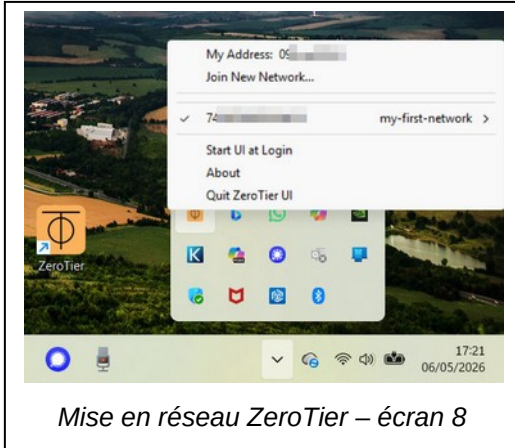
L'étape suivante consiste à installer du client ZeroTier sur notre station de travail Windows. Pour cela, il faut télécharger le logiciel depuis <https://www.zerotier.com/download> puis choisir le fichier MSI Installer (x86/x64).



Après installation, un second **Member Device** apparaît dans **my-first-network**. Une adresse IP est maintenant attribuée à notre Raspberry Pi, reconnue ici, en première ligne, comme un équipement Linux.

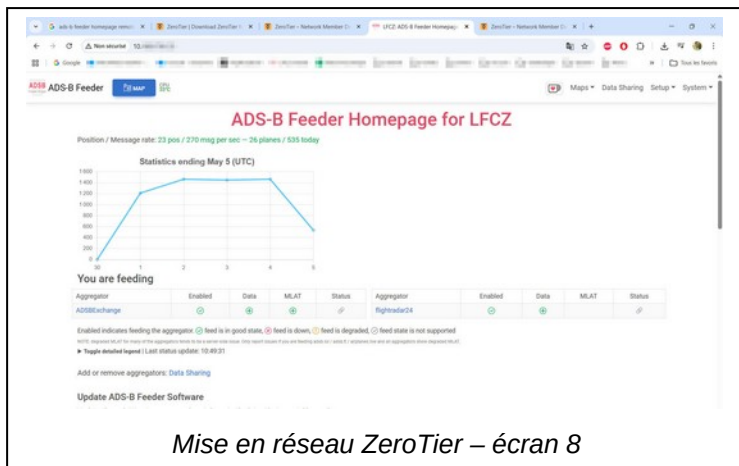


Après avoir retrouvé l'icône de lancement de **ZeroTier** dans le sous-répertoire ProgramData/Microsoft/Windows/Start Menu/Program, j'en place une copie sur le Bureau. Le double-clic semble d'abord inopérant, mais il n'en est rien : l'application se lance et vient simplement se loger parmi les icônes cachées de la barre des tâches.



Mise en réseau ZeroTier – écran 8

Il devient alors possible, depuis le navigateur de notre station de travail, de rejoindre la station **ADSB.im** en indiquant simplement son adresse IP ZeroTier comme adresse à atteindre.



Mise en réseau ZeroTier – écran 8

A suivre..

Avec ce douzième volet, nous avons remis en service une station ADS-B complète : réception des signaux à 1090 MHz, décodage des trames, affichage local des avions et partage des données vers plusieurs réseaux. Le prochain article nous fera franchir une étape un peu différente. Nous nous éloignerons de la technique SDR au sens strict pour nous intéresser à l'exploitation des données déjà reçues et décodées.

Il sera alors question d'API, de fichiers JSON, de petits scripts Python et d'interrogations de services en ligne. Nous verrons comment récupérer les informations produites par notre propre station, les trier, les convertir, les enrichir, puis les comparer avec celles proposées par différentes plateformes. Après avoir appris à recevoir les avions, nous commencerons donc à faire parler les données qu'ils nous transmettent.