

# MySondy GO

*par*

*Mirko Dalmonte (IZ4PNN)*

*Aldo Moroni (IW2DZX)*

*Jordan Antonio Provesi (IZ2ZUZ)*



Une fois terminé le travail sur le Raspberry pour réaliser le projet MySondy Server (et son application MySondy Finder), il était désormais clair qu'il manquait une étape fondamentale : un système dédié à la recherche sur le terrain.

Avant de commencer le développement de MySondy GO, j'ai fait, comme toujours dans ces cas, une simple enquête pour comprendre exactement ce qui existait au niveau matériel/logiciel concernant la recherche sur le terrain en matière de radiosondage. Il est immédiatement apparu que le choix n'était pas très large ; en fait, certains projets étaient à nouveau basés sur Raspberry et d'autres (au moins deux) sur un petit circuit de développement appelé TTGO.



Pour des raisons de portabilité et de consommation d'énergie je me suis donc orienté vers ce TTGO (433MHz).

Un TTGO n'est rien d'autre qu'un ESP32 (on peut penser à un Arduino beaucoup plus puissant avec bluetooth et wifi) connecté à une puce SX1278 (un émetteur-récepteur intégré 137-525 MHz).

Le TTGO a été créé pour les applications 433 MHz mais peut également fonctionner à d'autres fréquences. En fait, quelques changements dans les bibliothèques suffisent et ici, il est réglé sur 403 MHz.

L'ESP32 était quelque chose de familier pour moi, j'avais fait des projets intéressants dans le passé, je pouvais travailler assez facilement.

Il y avait déjà deux firmwares qui peuvent être installés sur le TTGO : un payant (50 dollars pour la licence) et un logiciel gratuit.

Aucun de ceux-ci ne correspondait, même en partie, à ce que je voulais réaliser; le payant ne représentait certainement pas ce Ham-Spirit que j'ai toujours essayé de promouvoir à ma petite manière.

Ces deux firmwares pour le TTGO sont capables de recevoir les radiosondes, d'afficher quelques informations sur l'écran intégré et d'interagir avec l'utilisateur via un serveur web toujours très problématique sur cet appareil. Les deux logiciels semblaient à première vue être en quelque sorte liés car l'un avait pris «repère» sur l'autre, les mécanismes étaient plus ou moins les mêmes.

MySondy GO devrait être tout de suite différent des deux projets existants: il doit être entièrement confié à une application Android capable de gérer toute la configuration TTGO (donc aussi tous les paramètres des différentes librairies) et en temps réel afficher, dans un environnement simple et agréable, les données entrantes et les positions de la sonde et de notre smartphone, respectivement à l'aide de tous les capteurs de positionnement de notre téléphone portable (GPS, accéléromètre, gyroscope et magnétomètre).

La connexion bluetooth entièrement automatique doit constituer un autre support fondamental pour l'échange de données qui aurait lieu peu de temps après, en utilisant un protocole encore à inventer.

Il était enfin temps de commencer, les idées étaient assez claires et les utilisateurs du groupe Facebook de Radiosonde Italie commençaient à me donner d'autres idées et conseils.

Par exemple, le problème de certains TTGO liés à un décalage de fréquence anormal a été mis en évidence. La solution était alors d'intégrer un "Tuner" capable de régler automatiquement l'offset jusqu'à une erreur de + - 500KHz. Pour mener à bien cette étape délicate, je me suis appuyé sur l'AFC (Automatic Frequency Control) de notre SX1278. Utilisé avec une certaine ruse, il est d'une importance fondamentale pour résoudre le problème.

Tant d'idées et tant de fonctionnalités ne pouvaient que me faire entrer en conflit avec les ressources limitées du TTGO.

Le formatage de la mémoire par défaut offre même la possibilité de mettre à jour notre firmware via OTA (Over The Air) mais au coût élevé de pouvoir développer du code pour une taille maximale de 1 Mo.

Seule la bibliothèque Bluetooth faisait 500 Ko, MySondy GO était déjà plus du double; comment faire? Abandonner les mises à jour via OTA et préparer une table de partition qui me permettrait d'atteindre au moins 3 Mo a été l'étape décisive.

En ce moment MySondy GO n'utilise pas la connexion wifi mais j'ai réussi malgré tout à insérer les librairies pour les futures implémentations (certains utilisateurs me demandent déjà de pouvoir envoyer des données vers des serveurs externes, grâce à cet appareil cela sera sans doute possible).

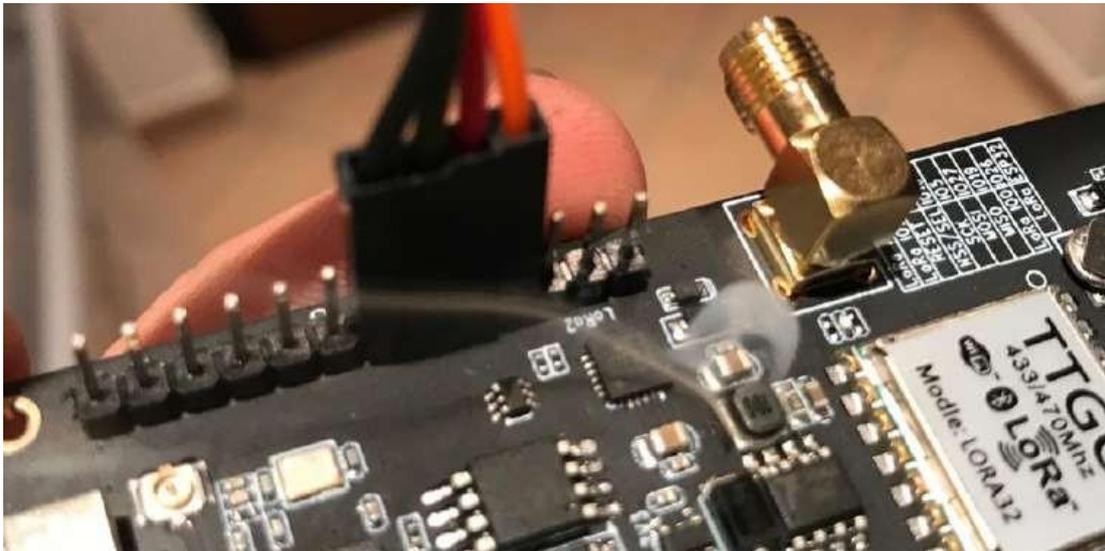
Pour le décodage des différentes sondes, je me suis appuyé sur les bibliothèques que j'avais créées pour MySondy Server (plus précisément, ce sont les bibliothèques usuelles utilisées, par exemple, également sur le RaspBerry de dx1APRS, que j'ai personnalisées et modifiées afin de lire la télémétrie complète du RS41).

Leur intégration a été simple et rapide et pour le moment elles ne sont pas utilisées à pleine capacité car MySondy GO a été conçu pour la recherche sur le terrain, la plupart des données entrantes ne sont pas affichées (du moins toutes les données météorologiques).

Les jours passaient, les soirées s'allongeaient parfois jusqu'au matin et mes amis Aldo Moroni et Jordan Antonio Provesi ont commencé, je le crains, à ne pas me supporter.

Leur contribution a été fondamentale comme d'habitude. Aldo Moroni a été une véritable référence pour moi, il a représenté et représente certainement l'un des plus grands experts en radiosondes que je connaisse et possède l'une des stations les plus performantes de tous les temps.

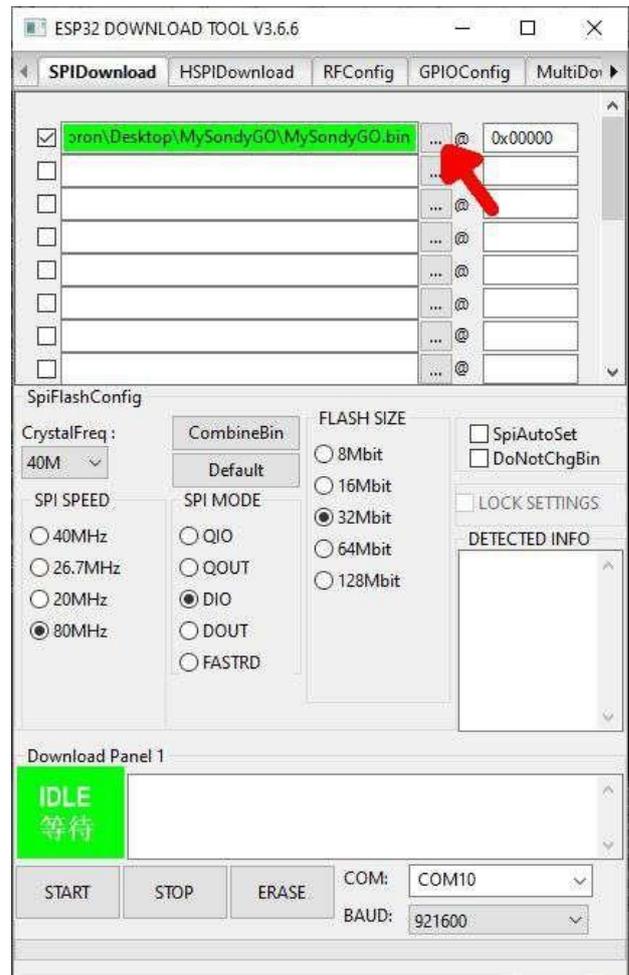
Avec la modestie et la disponibilité que seules les personnes de son calibre peuvent avoir, il a joué un rôle déterminant depuis le projet MySondy Server dans l'étude du décodage, dans le test de logiciels encore immatures, dans les conseils et dans les décisions communes. Jordan Antonio Provesi m'a permis d'intégrer la prise en charge de plusieurs types d'écrans et a soigneusement testé les différentes versions de firmware jusqu'à ce qu'il fasse littéralement fumer son TTGO.



Actuellement il a créé un groupe d'achat (sur le groupe Facebook de Radiosonde Italia) pour ceux qui ont l'intention de se procurer un TTGO à un prix raisonnable et pour ceux qui veulent en recevoir un déjà programmé avec tout ce dont ils ont besoin.

# Installation du firmware

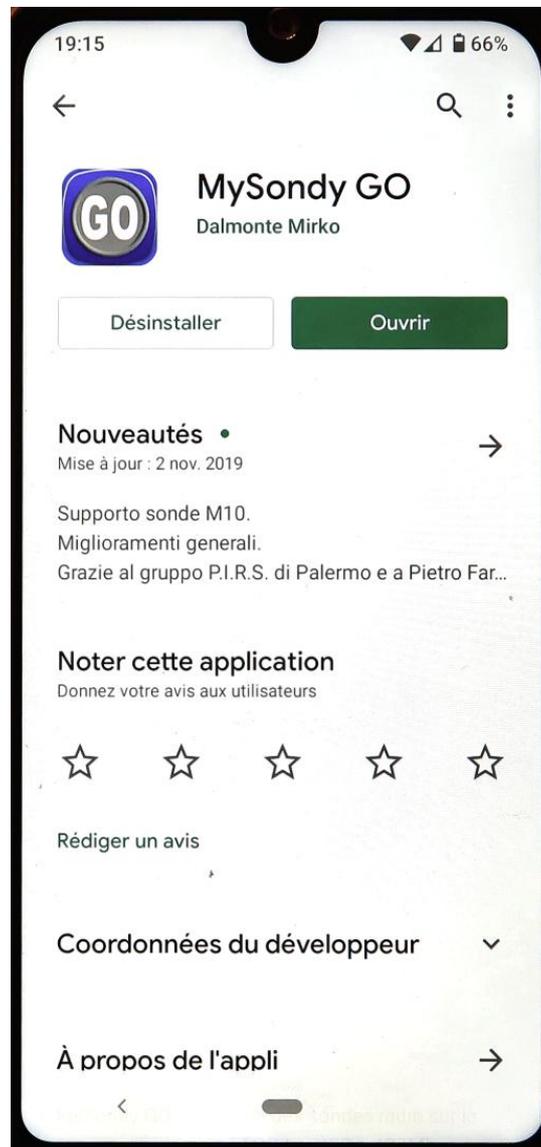
Une fois le firmware terminé (dès ses toutes premières versions d'Alfa), il ne restait plus qu'à créer une image bin à partir de mon habitude de travailler uniquement depuis la ligne de commande et avec des éditeurs de texte, ce qui ne m'a certainement pas aidé à créer un guide simple et pratique pour cette étape délicate .  
Une fois de plus Aldo Moroni est intervenu, a créé un guide assez éloquent et simple pour l'utilisateur.



En quelques secondes, l'image du firmware est copiée dans la mémoire du TTGO. Notre appareil est opérationnel (la version au moment de la rédaction est la 1.20).

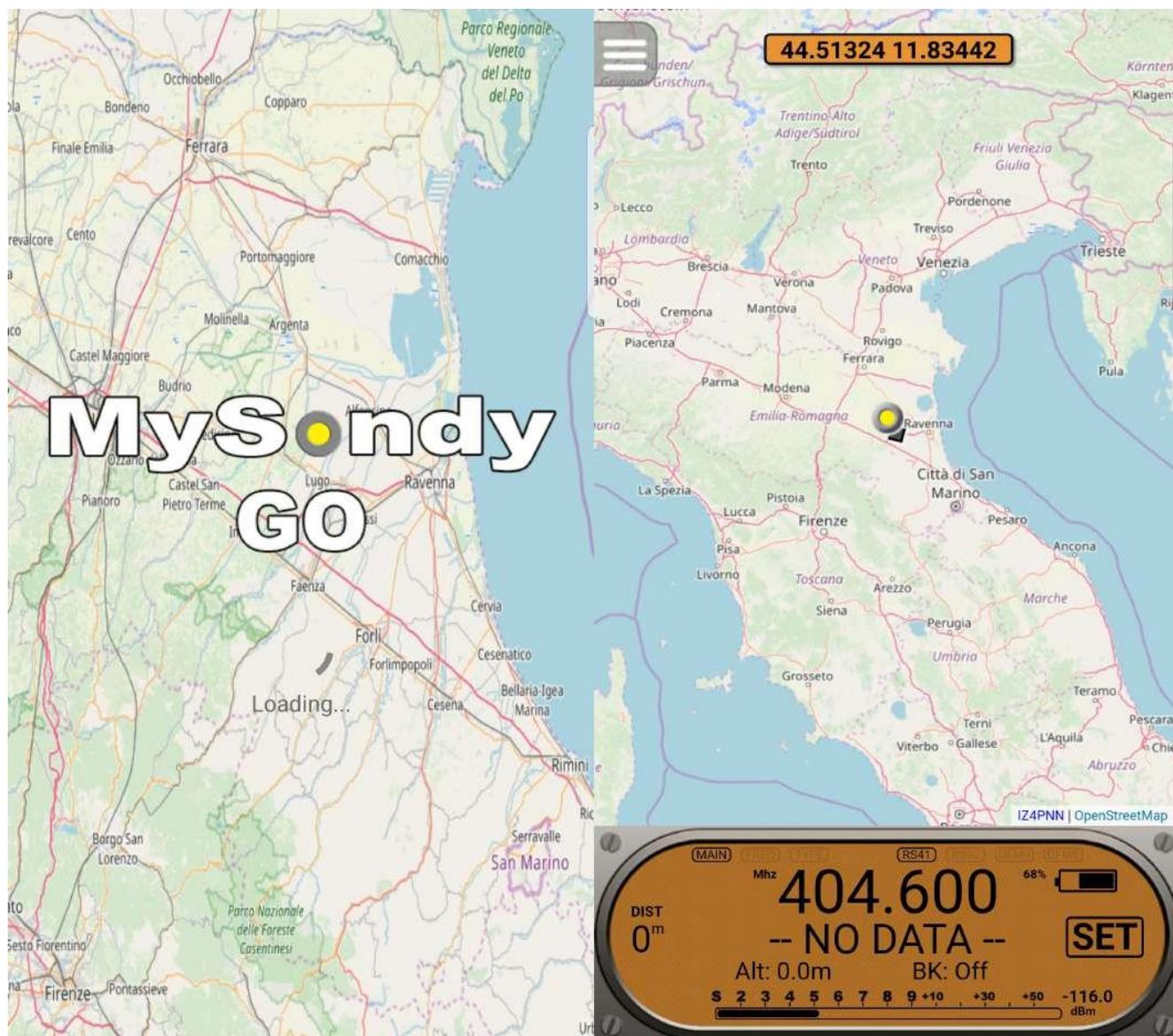
# Installation de l'app Android

L'application MySondy GO est disponible dans le Google Play Store, son installation est simple et intuitive.

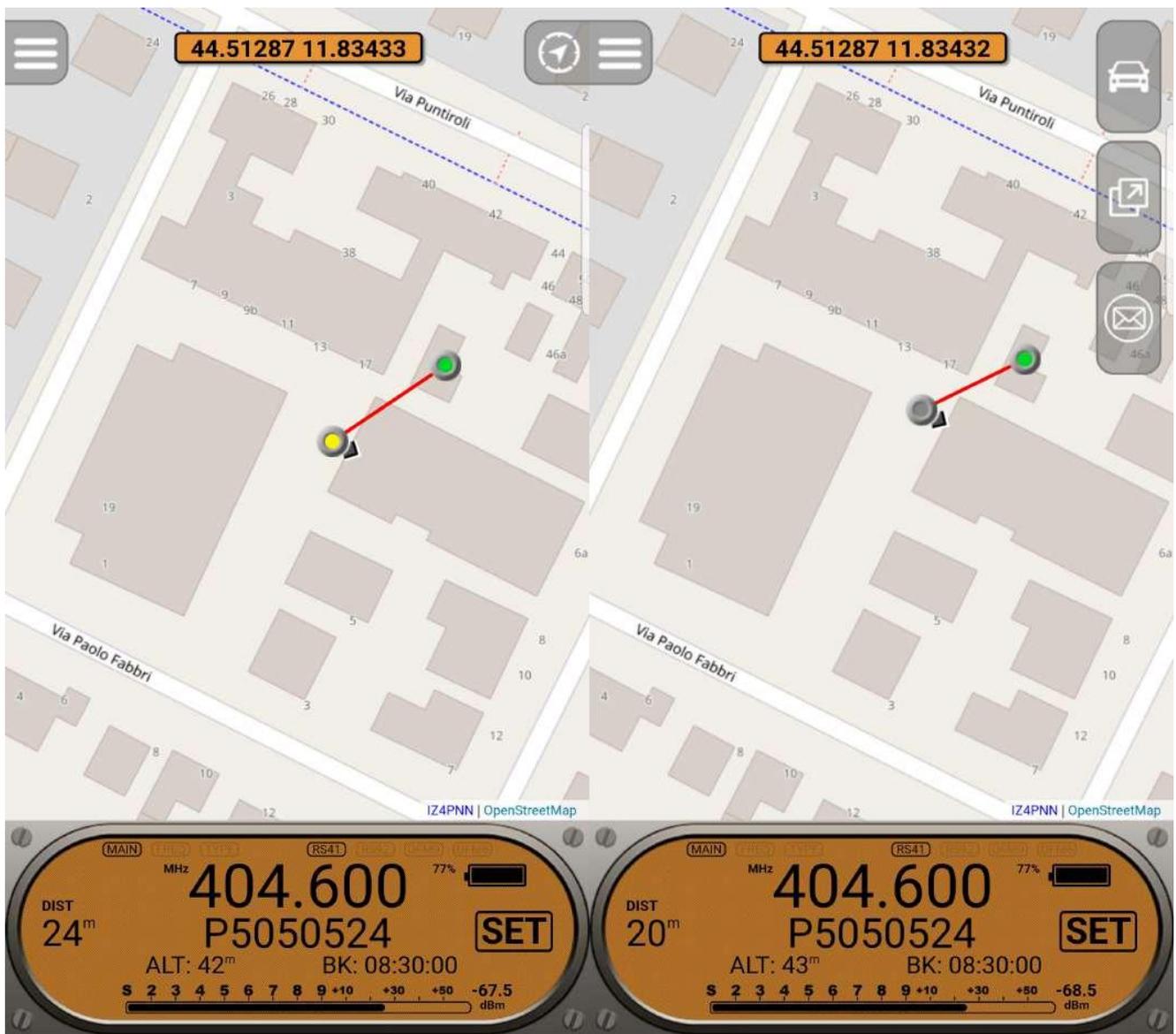


Au démarrage de MySondy GO après nous avoir demandé les autorisations habituelles, il prend en charge tous les aspects de la connexion Bluetooth; après l'autorisation d'activation, la recherche de TTGO actifs à proximité commence. Si la recherche aboutit, l'application effectue le couplage et établit immédiatement la connexion. A partir de ce moment, notre TTGO est constamment synchronisé avec notre smartphone qui est capable de le contrôler et de recevoir toutes les données nécessaires.

# Utilisation de l'application



MySondy GO est livré avec une carte, un affichage principal, un affichage secondaire (dédié aux coordonnées) et un petit bouton qui nous permet d'accéder au menu. L'affichage principal montre toutes les données reçues du TTGO, c'est-à-dire la fréquence actuelle, le type de sonde, le nom et la distance de la sonde éventuellement reçue, le temporisateur de burst killer dans le cas actif (RS41), l'état de la batterie et le puissance du signal reçu (Recommandation technique de la région 1 de l'IARU, S9 -93 dBm).



A la réception d'une sonde, une icône apparaît qui représente sa position: elle restera verte clignotante tant que son signal sera reçu, puis elle deviendra rouge.

L'icône jaune clignotante de notre position indique que le TTGO est connecté et communique correctement avec le smartphone. A l'inverse, un voyant rouge clignotant indique que le TTGO n'est pas connecté ou que la connexion n'est pas stable. La même icône est équipée d'une petite flèche qui indique notre direction par rapport à la carte et nous guidera vers le constat en utilisant également une trace rouge (qui peut être désactivée depuis le menu) qui relie notre position actuelle à la sonde.

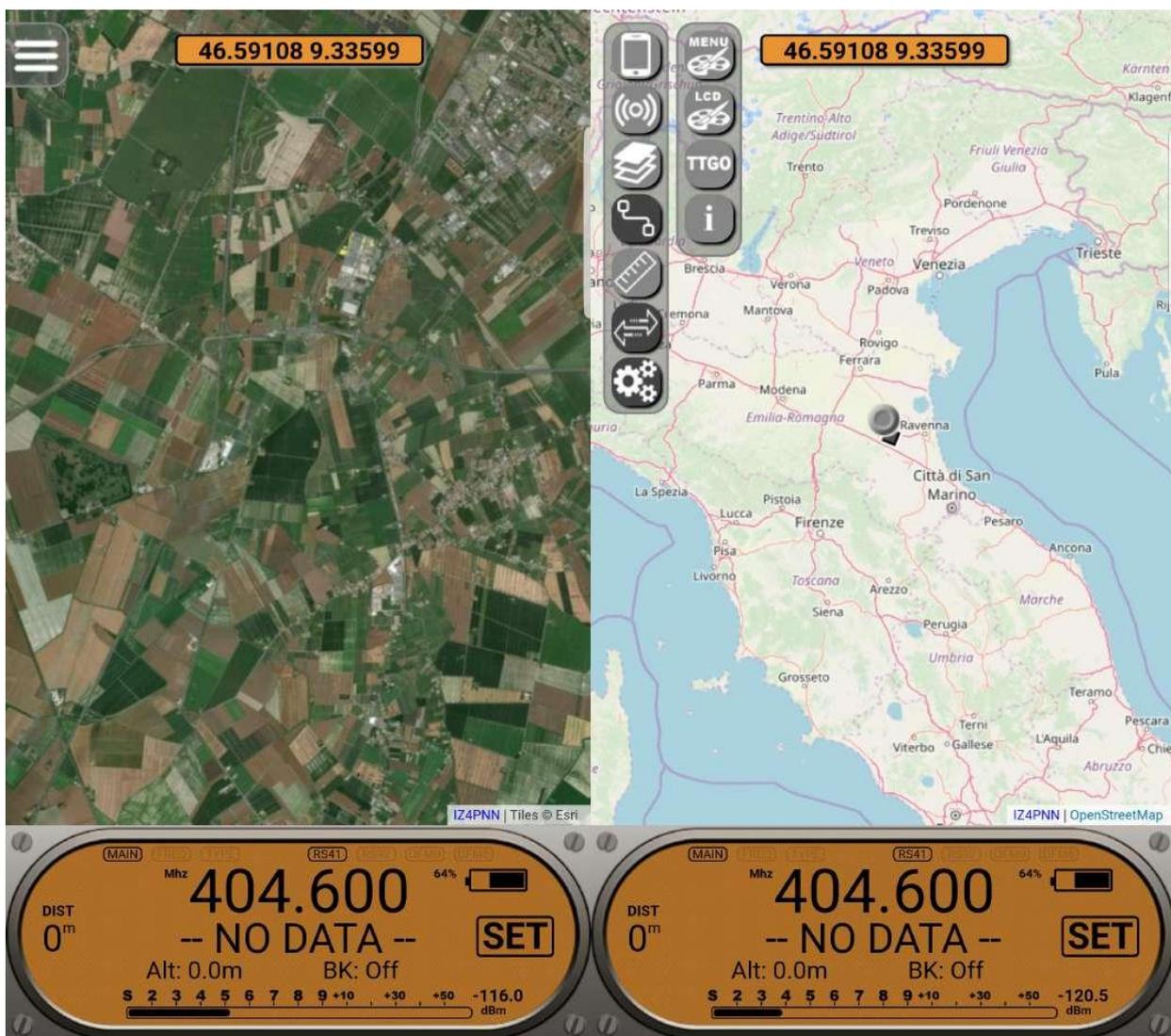
Lorsqu'une sonde est effectivement reçue, un bouton supplémentaire apparaît en haut à droite; c'est le menu de partage qui nous permet d'envoyer les coordonnées de la sonde au logiciel de navigation GPS par défaut, de partager la position de la sonde avec les différents logiciels cartographiques ou d'envoyer les données de la sonde (nom, type et coordonnées) par email ou tout logiciel de messagerie (WhatsApp, etc.)

Dans l'écran principal, nous voyons le bouton "SET" qui permet de modifier la fréquence et le type de sonde que nous voulons recevoir.

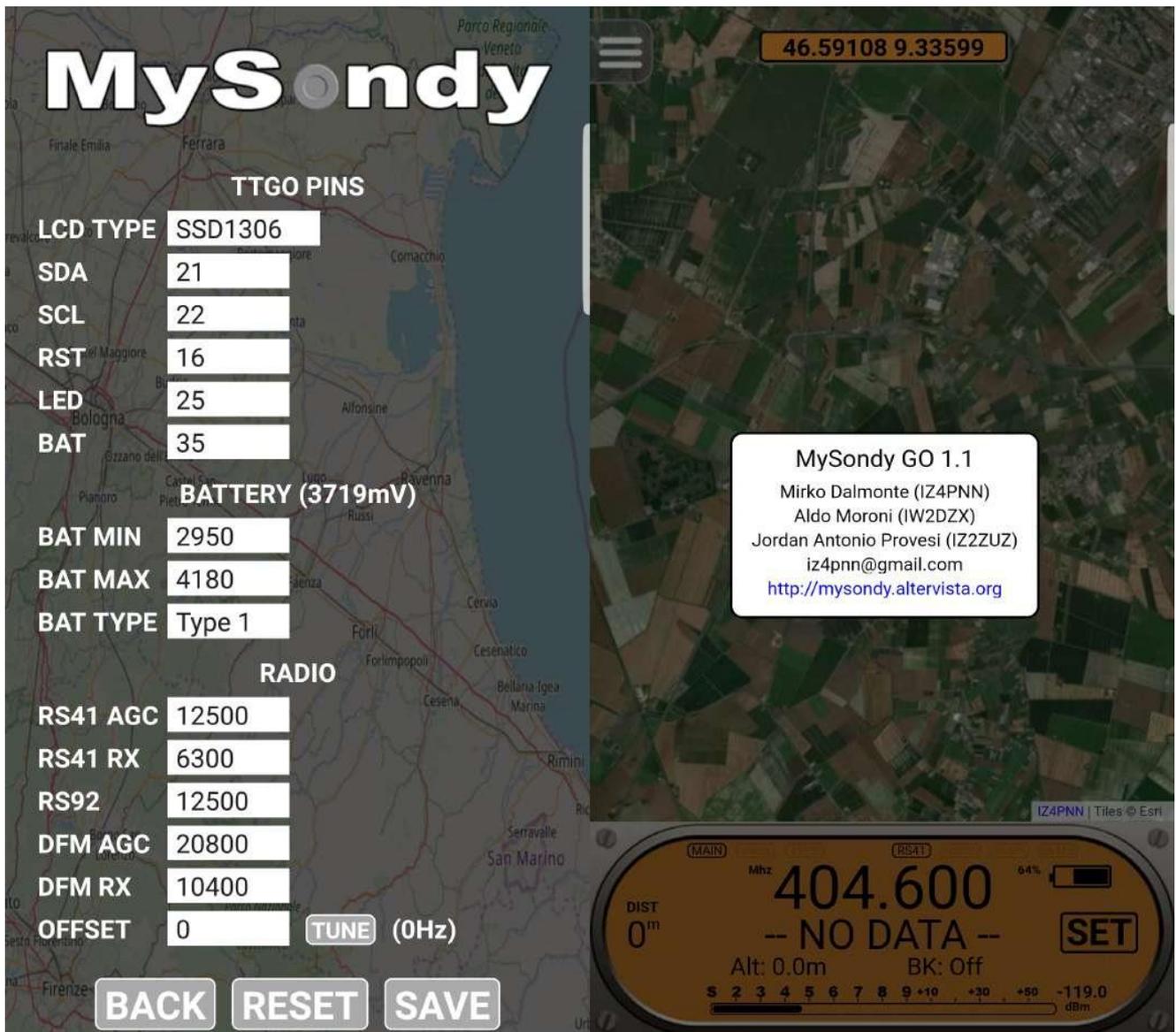


A l'aide du bouton en haut à gauche, comme nous l'avons dit, le menu principal s'ouvre ce qui nous permettra d'agir sur toutes les préférences de l'application et de configurer complètement notre TTGO.

Nous pouvons respectivement centrer la position de notre smartphone sur la carte, centrer la position de la sonde, afficher la carte en mode "satellite", masquer / afficher la trace qui relie la sonde à notre position, activer la mesure sur la carte (cela nous permet de mesurer la distance entre deux points sur la carte au niveau du sol), connecter manuellement le TTGO au cas où il serait déconnecté et ouvrir enfin le sous-menu des paramètres.



Dans le sous-menu des paramètres, on peut modifier les couleurs du menu lui-même, la couleur de fond de l'écran principal et de l'affichage des coordonnées, ouvrir la configuration TTGO et afficher les informations du logiciel.



Dans la configuration TTGO on retrouve trois paragraphes distincts, le premier dédié à la configuration matérielle (TTGO PINS), le second dédié à la gestion de batterie (BATTERY) et le troisième dédié aux paramètres radio (RADIO). En bas, il y a un bouton pour revenir à l'écran principal (BACK), un bouton pour réinitialiser le TTGO à configuration d'origine (RESET) et un bouton pour enregistrer la configuration. Notez que toute la configuration est volontairement enregistrée dans le TTGO et non dans le smartphone; cela nous permet d'obtenir le même résultat quel que soit l'appareil utilisé.

**TTGO PINS:** Cela permet de sélectionner le type d'affichage utilisé et de définir toutes les broches matérielles du TTGO. Dans la plupart des cas, les paramètres par défaut seront optimaux.

**BATTERY:** En définissant un maximum, un minimum et une courbe de décharge, cela nous permet de visualiser l'état de la batterie sur l'écran TTGO et sur l'écran de l'application. Ces paramètres dépendent grandement du type, de la capacité et de l'état de la batterie elle-même. Le conseil est d'essayer différentes configurations jusqu'à ce que vous en trouviez une satisfaisante.

**RADIO:** Cette section contient tous les paramètres radio des sondes que nous allons recevoir (ce sont les paramètres par défaut des bibliothèques de décodage modifiées mentionnées ci-dessus) ainsi que la fonction «Tune».

Le "Tune" peut être effectué manuellement en entrant le décalage souhaité dans le champ approprié ou automatiquement en appuyant sur le bouton idoine.

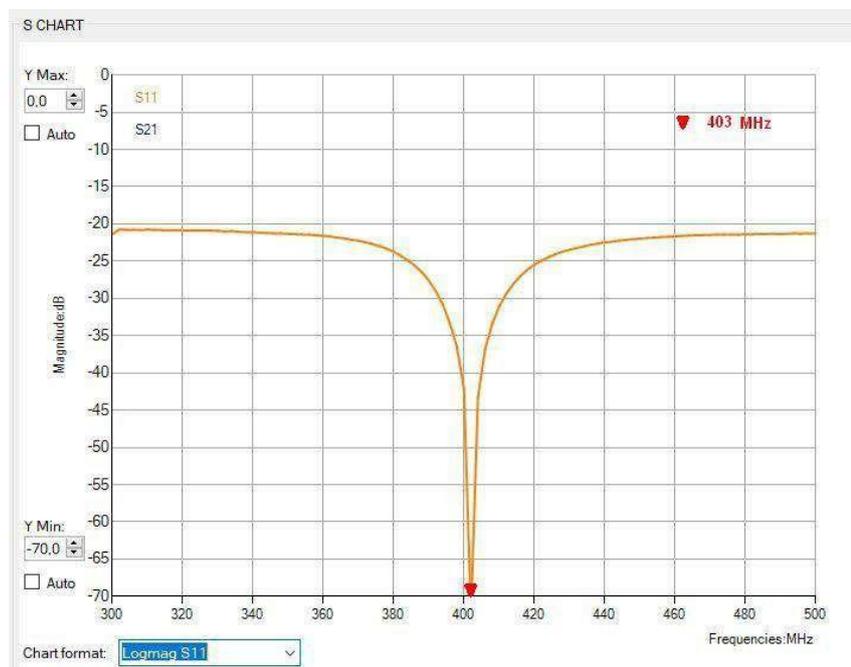
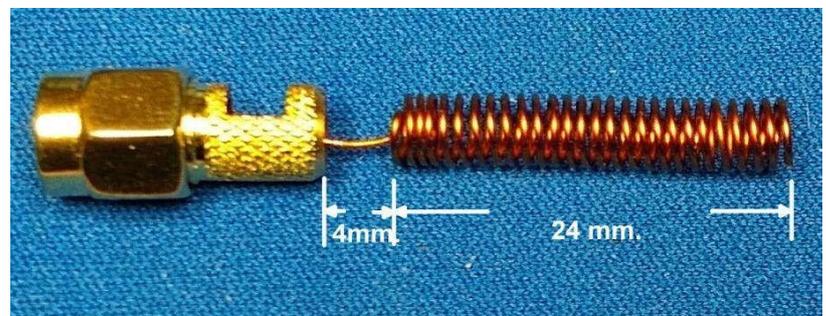
Dans les deux cas, nous verrons clairement l'état actuel de l'AFC (à côté du bouton "Tune"). Cette valeur doit être compensée pour la rapprocher de zéro.

Après nos réglages, nous pouvons sélectionner le bouton "Enregistrer" pour tout enregistrer dans le TTGO.

Si un redémarrage est nécessaire (certains paramètres matériels l'exigent), l'application elle-même nous en informera et nous guidera tout au long de l'opération.

## Modification de l'antenne

Le TTGO est livré avec une petite antenne qui n'est évidemment pas optimisée pour les fréquences des radiosondes. Aldo Moroni a donc décidé de procéder à une modification de manière à avoir le maximum de résultats possibles.



# Conclusions

Vous pouvez trouver tout le matériel nécessaire et toutes les informations pour créer votre MySondy GO personnel sur la page officielle du projet:

<http://mysondy.altervista.org/mysondygo.php>

et tout le support nécessaire au désormais célèbre groupe Facebook créé par Roberto Silvestri, Radiosonde Italia (<https://www.facebook.com/groups/919693011428093>). Nous ne pouvons que remercier le professeur Achille de Santis pour ses efforts de diffusion des radiosondes en Italie et Roberto Silvestri pour nous avoir accueillis dans son groupe pendant un certain temps.

# MySondyGO

(Manuel d'installation sous Windows par Aldo Moroni (IW2DZX) )



## **Matériel requis:**

- **TTGO Esp32 LoRa 433 MHz V2** (version avec ou sans GPS)
- **ANDROID depuis la version 4.4.2**
- TTGO Esp32 LoRa 433 MHz V1 (*changement des broches du LCD: SDA (4) et SCL (15)*)

Le logiciel **MySondyGO** se compose de deux parties:

- Un firmware nommé "MySondyGO140.bin" à charger dans le module TTGO et qui s'occupe de la réception et du décodage du signal.
- Une app Android "MySondyGO " à télécharger sur le PlayStore. Elle gère le TTGO par Bluetooth et affiche le signal reçu.

*Note du traducteur: Tous les fichiers nécessaires se trouvent ici : (il faudra peut-être s'inscrire à Dropbox)*

*[https://dropbox.com/s/3k2u2d1hu54df2s/MySondyGO\\_1\\_40.zip?dl=0](https://dropbox.com/s/3k2u2d1hu54df2s/MySondyGO_1_40.zip?dl=0)*

*et, pour la dernière version du firmware :*

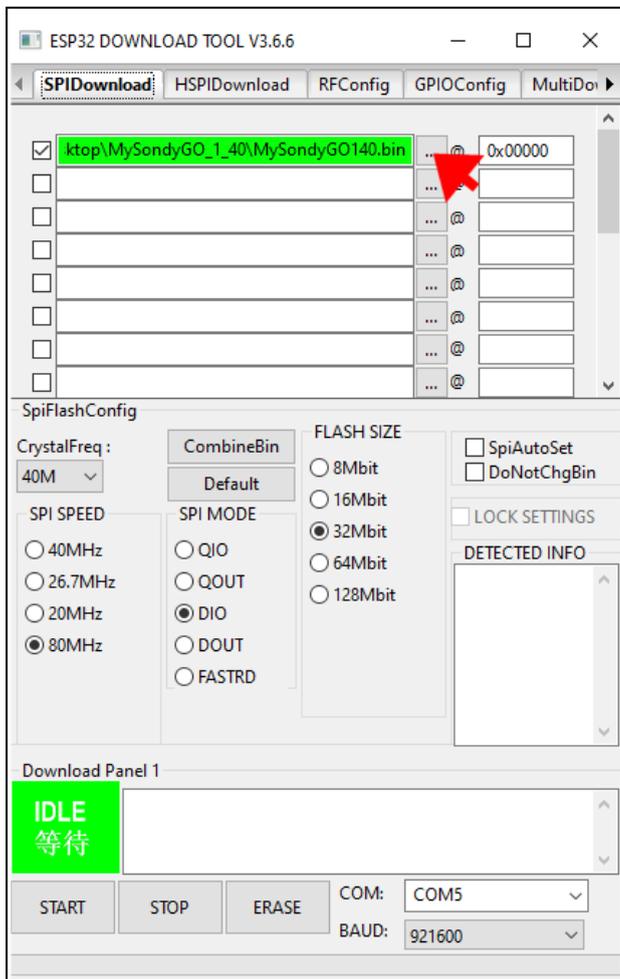
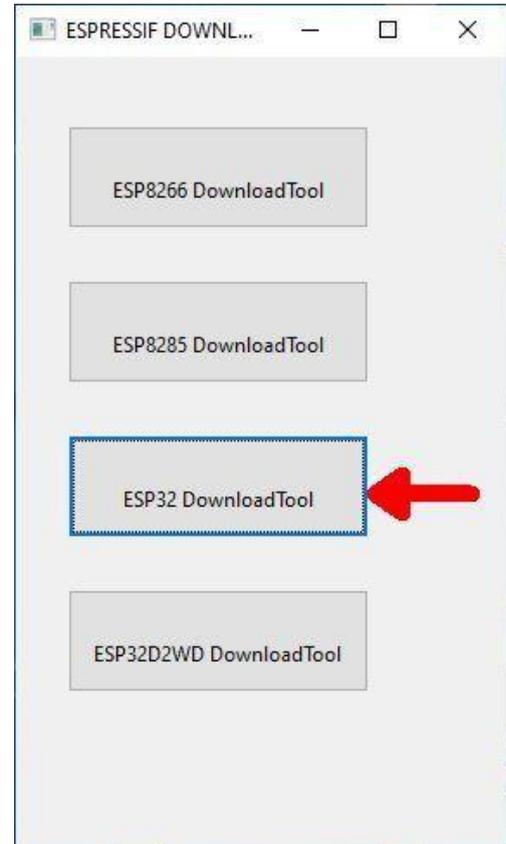
*<https://www.dropbox.com/s/g8c40q2jm3racvn/MySondyGO140.zip?dl=0>*

Pour charger le firmware avec un PC sous Windows 10 il faut d'abord installer un driver qui doit permettre la communication par USB avec le TTGO via un port COM. Ce port COM physique ou virtuel sera ajouté à la liste déjà présente sur le PC.

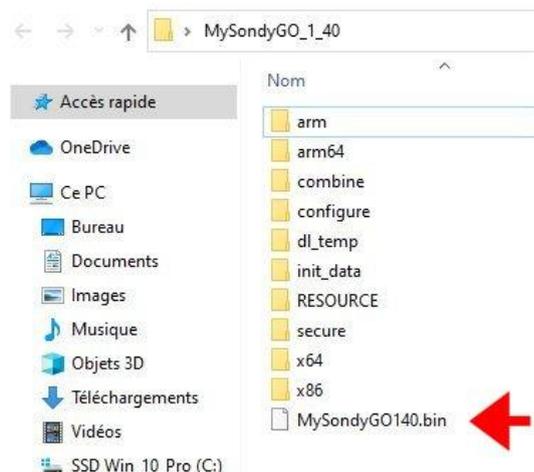
Commencer par décompresser le fichier "MySondyGO\_1\_40.zip", par exemple sur le bureau, pour créer un dossier "MySondyGo" contenant tout le nécessaire pour l'installation.

1. Tout d'abord, installer le driver en fonction du système d'exploitation (si 32 bits avec **CP210xVCPInstaller\_x86.exe**, si 64 bits avec **CP210xVCPInstaller\_x64.exe**).
2. Une fois l'installation du driver terminée, connectez le TTGO via USB sans l'enclencher. Retour au dossier MySondyGo et lancer "flash\_download\_tools\_v3.6.6.exe" pour charger le firmware via la procédure suivante:

Dans le premier écran qui apparaît, sélectionner :  
**ESP32 DownloadTool**

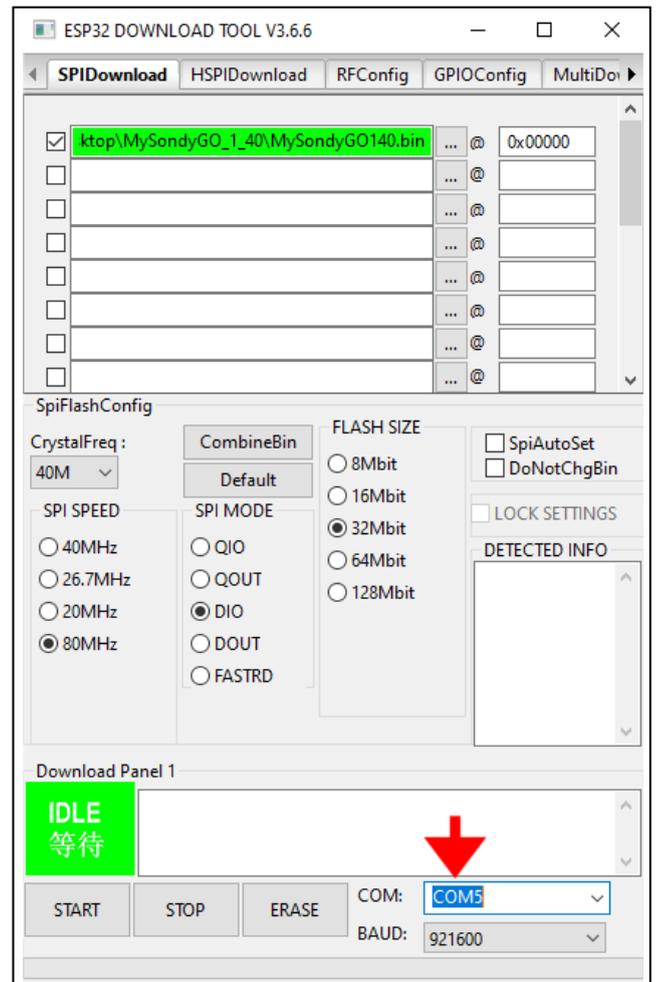


Cliquer sur les 3 points pour indiquer le chemin  
du fichier **MySondyGO140.bin**.



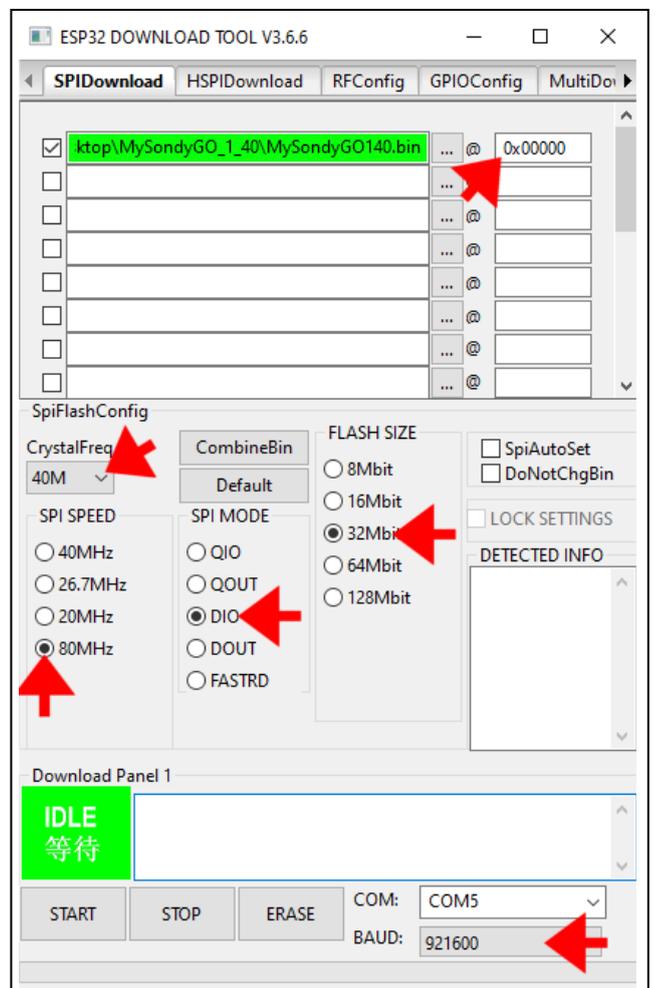
Ensuite enclencher le module TTGO et sélectionner le nouveau port COM qui apparaît dans la fenêtre COM:

(COM5 dans cet exemple)



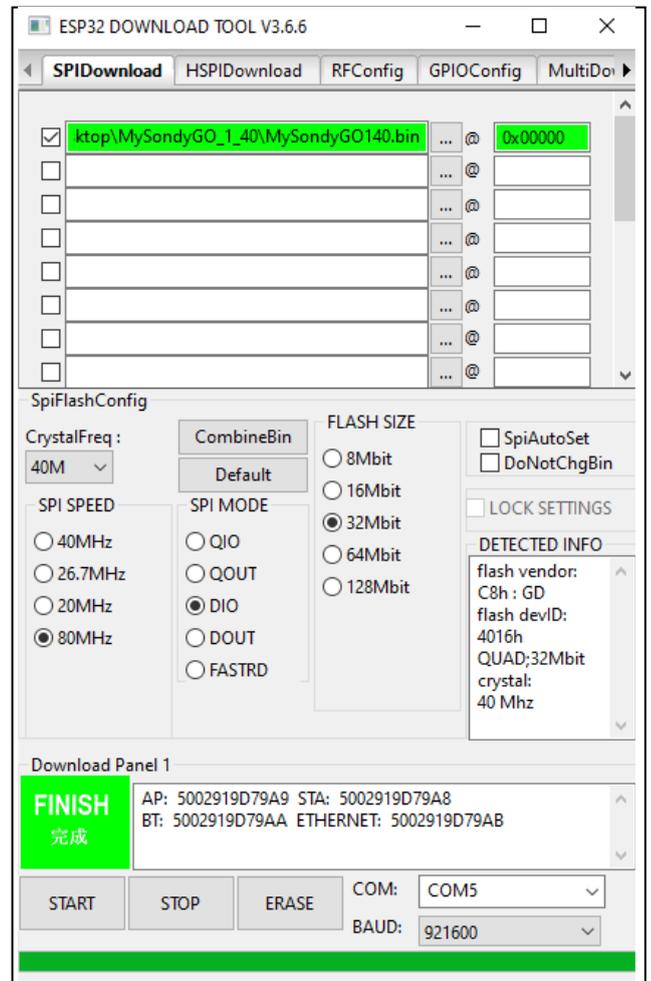
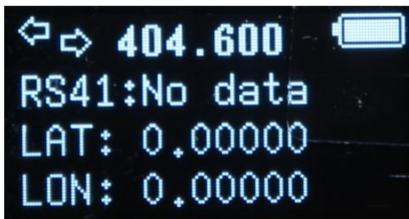
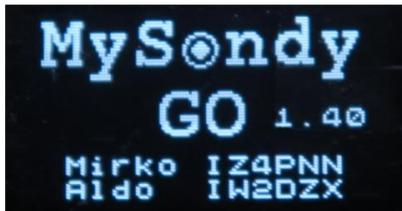
Vérifier que tout correspond à l'image ci-contre, à part le port COM qui peut être différent.

Une fois les paramètres contrôlés, cliquer sur le bouton **START** pour commencer le chargement du firmware qui dure env. 1 minute.

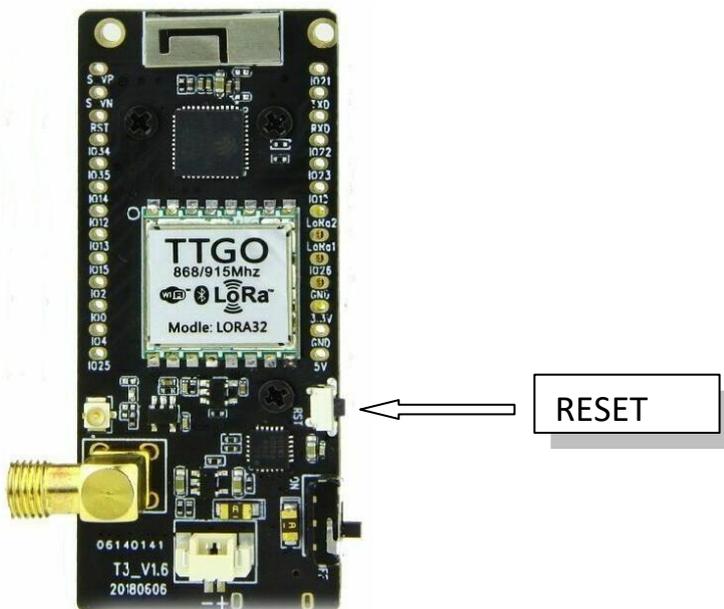
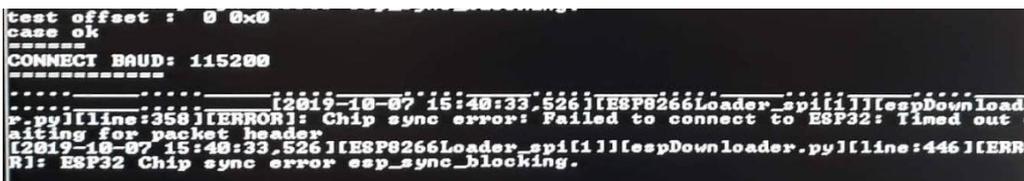


A la fin, le bouton vert "FINISH" confirme que tout s'est bien déroulé. Déclencher le **TTGO**.

Quitter **ESP32 Download Tool** et redémarrer le TTGO qui devrait afficher les écrans suivants:



Si le chargement du firmware échoue et qu'une série de points et lignes apparaissent en alternance dans la fenêtre de commande, presser la touche RESET du TTGO et recommencer.

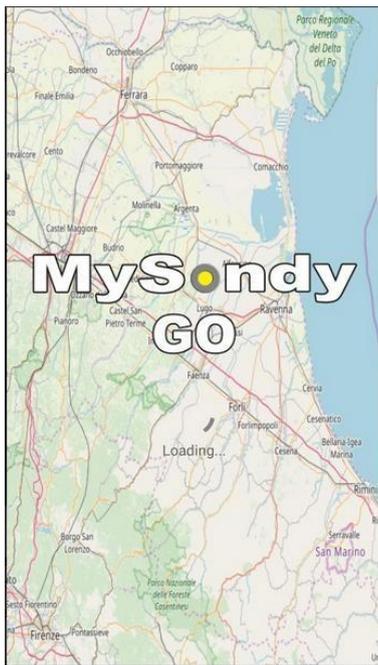


# MySondyGO

## App Android

Sur le smartphone accéder au Play Store 

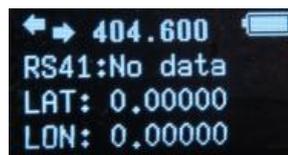
Rechercher l'app **MySondyGO** et l'installer en fournissant toutes les autorisations.



Enclencher le **TTGO** et lancer l'app **MySondyGO**

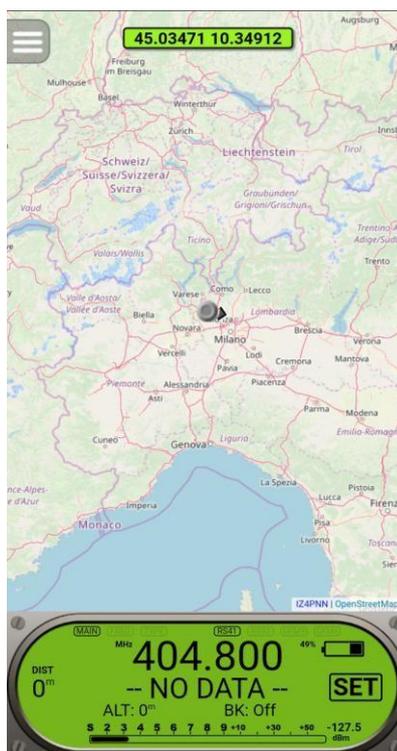
Au démarrage, **MySondyGO** demandera l'autorisation d'associer le bluetooth. L'application activera le bluetooth et l'éteindra lorsque l'application est fermée.

Il effectuera alors une recherche du **TTGO** qui devra afficher les deux flèches "pleines" si la connexion est effective.



Dans le cas contraire (ou par la suite) activer la connexion BT dans le menu ci-dessous.

Menu 



Appuyer sur **SET** pour régler la fréquence de réception, puis sur **NEXT** pour le type de sonde et enfin sur **SEND** pour envoyer les nouveaux paramètres au **TTGO**.

Zoom sur la position GPS

Zoom sur la position de la sonde

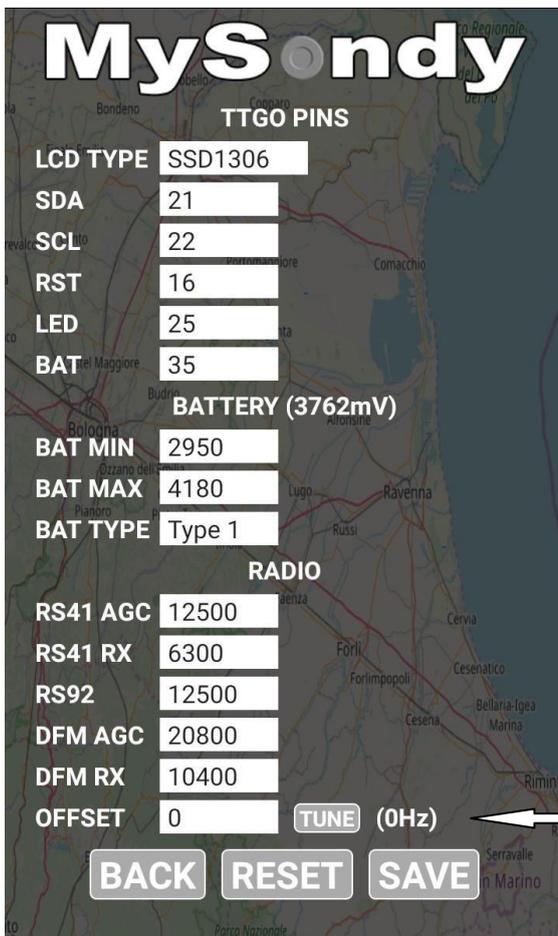
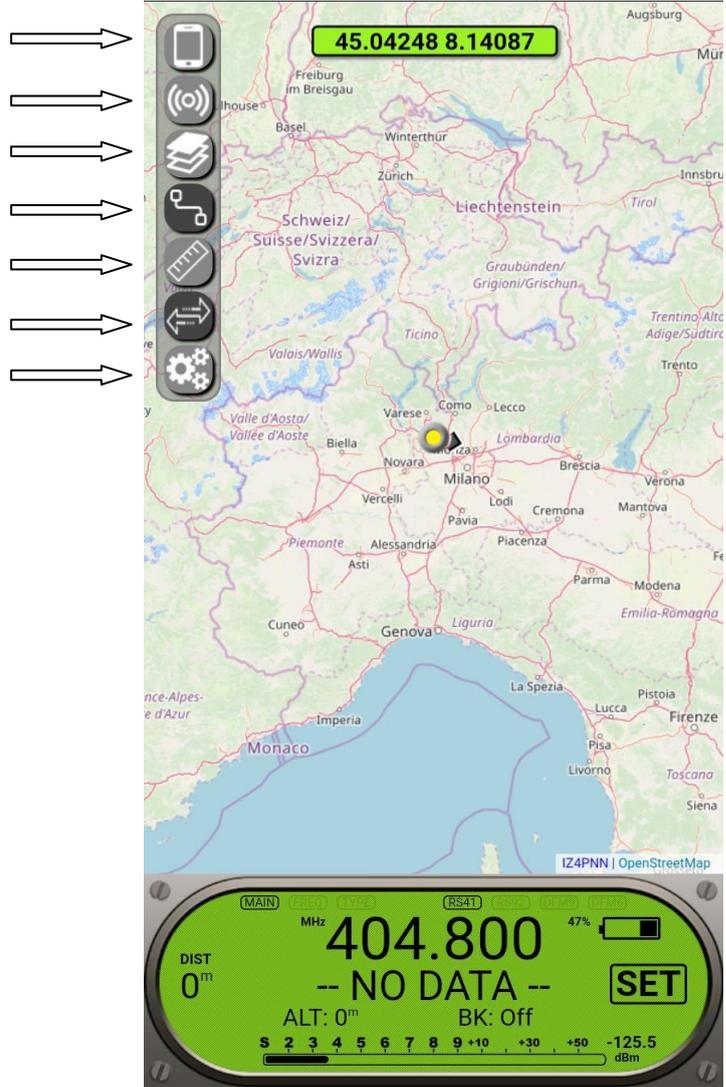
Type de carte

Trace on/off

Mesure de la distance

Connexion au **TTGO**

Paramètres divers (voir ci-dessous)



Le seul ajustement à faire éventuellement sera de corriger l'offset du **TTGO**.

L'étalonnage doit être effectué en présence d'un signal d'intensité moyenne. Appuyer sur **TUNE** et **MySondyGO** va scanner une plage de 500KHz autour de la fréquence définie dans l'écran principal.

Quand un signal utile est trouvé, il entre la valeur d'offset et il ne reste plus qu'à appuyer sur **SAVE**.

Lors de la réception d'un signal, la valeur d'offset est indiquée entre parenthèses. De légers écarts sont possibles mais sans influence.

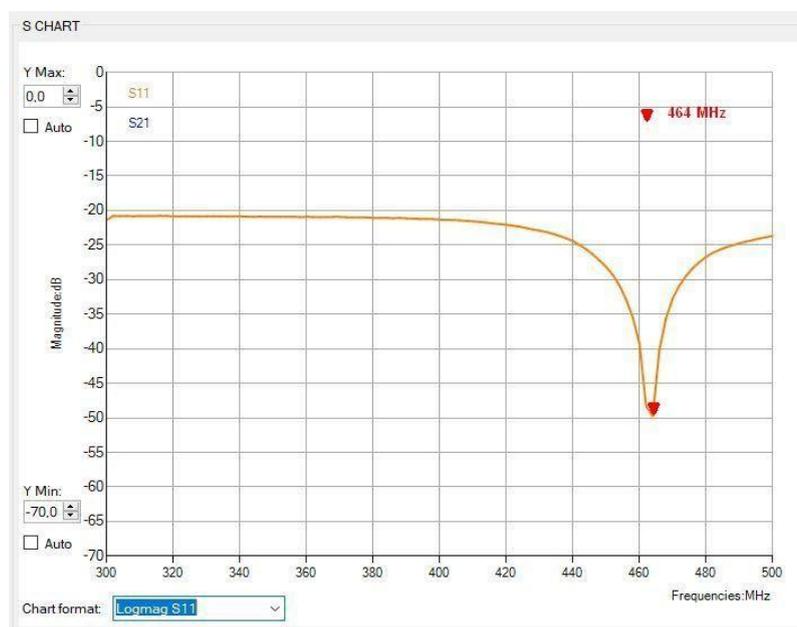
# Modification de l'antenne TTGO pour 400-406 MHz

Aldo Moroni (IW2DZX)

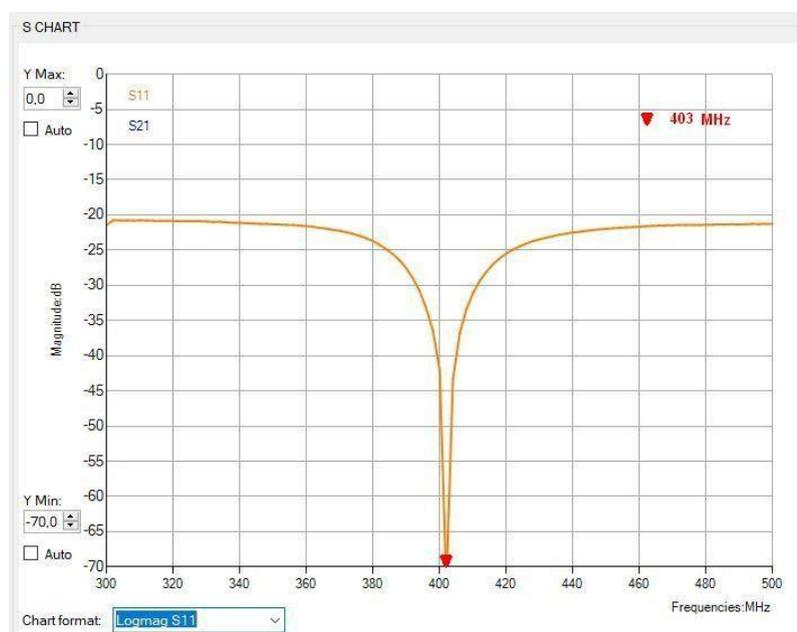
Les modèles TTGO Esp32 LoRa 433MHz sont fournis avec une petite antenne spirale avec revêtement en plastique rigide et connecteur SMA.



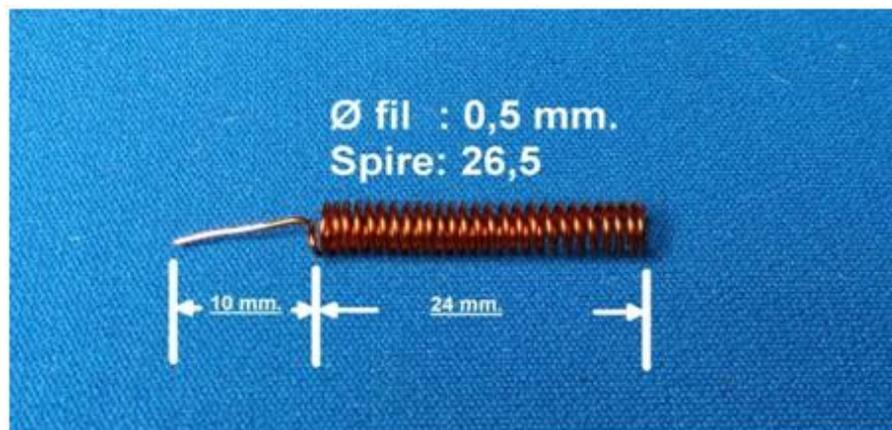
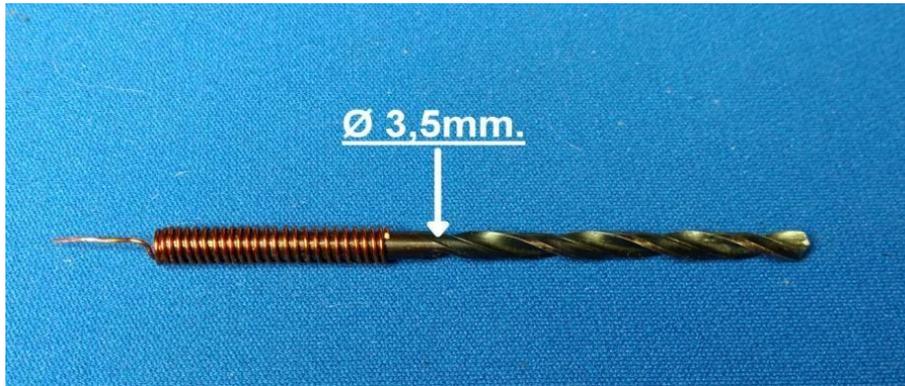
Les 3 exemplaires mesurés résonnaient sur 464 Mhz.



Pour l'adapter à la bande 400-406 MHz, il suffit d'enrouler 26,5 spires de fil de Cu émaillé de 0,5mm sur un mandrin de 3,5mm et de remplacer la bobine originale pour obtenir cette résonance.



Pour ce faire il faut respecter ces dimensions :



Une fois le bobinage réalisé, le revêtement en plastique est retiré en faisant levier avec un tournevis. Remplacez alors l'antenne originale par celle que vous venez de fabriquer et remettez le revêtement en prenant soin de ne pas déformer la bobine.

