

Open9x, un logiciel

Vous avez peut être déjà entendu parler de firmwares (logiciels d'exploitation) open-sources, c'est-à-dire gratuits, développés pour la radio chinoise Turnigy 9x (disponible en France chez Jamara sous le nom de Air Jump). Ces firmwares sont également exploitables dans les radios « persos » à base de micro-contrôleurs économiques Atmel. Open9x, le dernier en date - certainement le plus abouti - offre maintenant des possibilités fantastiques.

Texte et Photos : Franck Aguerre

Les premiers firmwares gratuits pour la 9x sont apparus en 2010 avec TH9x de Thomas Husterer. Il y a ensuite eu ER9x par Erez Raviv, Gruvin9x par Bryan Rentoul, Ersky9x par Mike Blandford, puis, pour finir, Open9x (<http://code.google.com/p/open9x/>) par Bertrand Songis. En parallèle, des cartes mères d'upgrade ont été développées (Gruvin9x et Sky9x). Il est même possible de réaliser sa propre carte mère (les plans et schémas des cartes pré-citées étant eux aussi open-sources).

Des logiciels pour tous ?

De prime abord, ces programmes peuvent sembler réservés à un public averti. En fait, ils sont faciles d'emploi et surtout ils sont évolutifs grâce à l'interaction possible des utilisateurs. En effet, la structuration imposée par l'hébergeur Google Code donne trois grands principes :

Un partage public de l'intégralité du code source et des outils de développement.

Une traçabilité par tickets des remontées de bug et des propositions (argumentées...) d'amélioration par les

utilisateurs. Pour des raisons de transparence et de vérification croisée, chaque ticket est soumis à discussion publique avant validation puis implémenté.

tation par l'équipe de développement.

Une traçabilité par version (courante en phase de développement ou « release » officielle pour la diffusion auprès des utilisateurs) de chaque modification du code.

Ce fonctionnement est à la fois souple et réactif tout en garantissant une parfaite maîtrise du code ainsi que sa pérennité, bien plus d'ailleurs qu'avec n'importe quel constructeur.

De plus, pour réussir à implémenter sur le modeste processeur ATmega64 de la radio des fonctions aussi puissantes que des phases de vol généralisées, le code doit être d'un niveau de pureté et d'acuité très élevé. Chaque bit mémoire ou chaque micro-seconde de temps de calcul compte, poussant les développeurs à maîtriser leur ouvrage dans ses moindres détails autant que dans sa globalité.

Le téléchargement (à la carte, suivant les options souhaitées) et la mise à jour des firmwares sont confiés au logiciel Companion9x, lui aussi open-source, multilingue (idem Open9x : 7 langues, dont le français) et multiplateforme (Windows, Linux, Mac). Il assure aussi la sauvegarde des modèles et offre des fonctions inédites : il convertit les données modèles lors d'un changement de version du firmware et, nec plus

ultra, il permet de programmer ou modifier des modèles ainsi que les simuler sur une radio virtuelle.

Bref, même si leurs contributeurs de ces programmes sont bénévoles, on touche au très haut niveau, tant technique que dans l'engagement sur la durée.

Une programmation exceptionnelle

Dans le cas de la radio 9x, la connexion avec l'ordinateur nécessite de câbler 6 fils pour brancher un programmeur USBASP (3€ sur eBay). Rien de bien sorcier, les soudures se font sur des pads bien accessibles. Les utilisateurs de télémesure devront aussi re-router deux entrées du micro-contrôleur, ce qui demande un peu plus de doigté. Tout est expliqué dans la notice téléchargeable



Voici la modeste 9x (importée par Jamara sous la désignation « Air Jump ») grâce à qui « tout » a commencé. A noter qu'elle est construite très sérieusement. Elle est d'une qualité étonnante au regard de son tarif.

gratuit étonnant

sur le site d'Open9x.

Un avant-goût des possibilités d'Open9x, via son prédécesseur ER9x, a été présenté dans mon article sur la programmation des radios (Modèle Magazine n°728). Open9x apporte de nombreux perfectionnements, notamment les phases de vol, les variables globales, un formatage de servo puissant et... bien d'autres. Le code a aussi été largement optimisé, avec par exemple des calculs plus rapides.

Une notice complète d'Open9x étant disponible sur son site internet (http://code.google.com/p/open9x/wiki/Open9x_Manual_FR), je n'évoquerai ici que les points saillants.

Démarrage et écran principal

Par sécurité, les positions des interrupteurs et du manche de gaz sont contrôlées au démarrage avec un blocage de l'émission. L'écran principal est ensuite affiché, avec trois zones :

- **pourtour** : positions des trims de manche.
- **partie haute** : nom du modèle, tension d'accu, phase de vol en cours, chrono principal.
- **partie basse (variable)** : positions des servos (graphique ou numérique) ou état des organes (de commande et de commutation) ou second chrono.

Depuis cet écran, quatre menus principaux sont accessibles avec un appui long sur les flèches de navigation :

• **télémesure** : FrSky, Jeti, etc., suivant le module d'émission utilisé.

• **statistiques** : graphique d'utilisation du manche ou de la voie de gaz, durée d'allumage de la radio, chronos 1 et 2, durée moteur (en minutes / secondes ou en % relativement à la durée de vol), état microcontrôleur (mémoire libre, latence de calcul, etc.).

• **réglages radio** : mode de pilotage, gestion du rétro-éclairage, tension d'alerte de batterie vide, gestion des sons, échantillonnage des manches (1024 ou 2048 pas), gestion de l'écolage, infos système et firmware, calibration des manches.

• **réglages modèles** : liste des modèles de modèles, paramétrage de la télémesure, configuration du modèle (nom, chronos, organe de « gaz », alarmes à l'allumage, finesse des trims, etc.), phases de vol, mixeurs et formatages de manches et de servos. Pour les débutants, l'option « template » propose aussi des pré-programmation de mixages.

Le synoptique n°1 montre l'enchaînement des menus (seuls les premiers écrans sont montrés). La structuration est limpide, avec un titre clair par écran et un enchaînement logique et intuitif entre les écrans (par fréquence d'utilisation et suivant le cheminement logique des informations).

Deux écrans (formatages de manches et mixeur) sont à géométrie variable, avec une ligne par formatage ou mixage jusqu'à, respectivement, 14 et 32 lignes. Chacune peut être éditée, mais aussi copiée, déplacée ou supprimée, permettant à l'utilisateur de créer sa program-

mation à la carte. Plusieurs lignes peuvent coexister, les actives apparaissant en caractères gras, ce qui permet en seul coup d'œil d'identifier le cheminement des ordres.

La logique de programmation

Elle est basée sur le découpage fonctionnel suivant (cf. synoptique 2), « centré gouvernes » comme chez Multiplex :

• **Banque d'éléments de commande et de commutation** : c'est une liste à géométrie variable suivant le contexte. Elle est utilisée dans tout le logiciel, incluant les manches, potentiomètres et interrupteurs physiques, mais aussi les trims, les voies de l'entrée d'écolage, les voies émises, les interrupteurs virtuels et momentanés (actifs un bref instant après la commutation de l'interrupteur associé) et une activation permanente (ON). Les deux états (actif / inactif) des interrupteurs sont distingués, tandis qu'une fonction « autoswitch » (reconnaissance automatique à la manipulation) simplifie leur sélection.

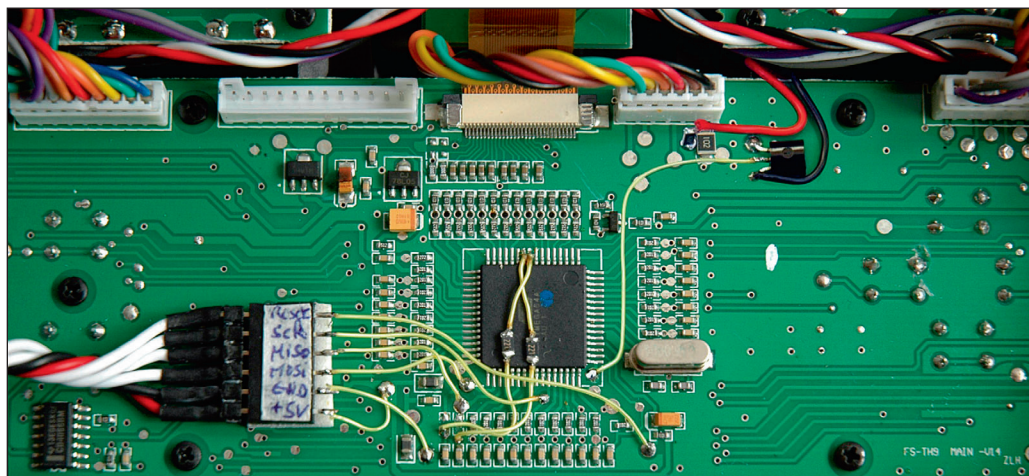
• **Phases de vol** : au nombre de 5, avec pour chacune un nom, un interrupteur et deux fondus de transition (activation et désactivation). Elles différencient les trims, les formatages de manche, les variables globales et les mixages, avec une possibilité, unique sur le marché, de communalisation. Chaque trim ou

mixage peut en effet être communs à plusieurs phases via, pour chacun, un tableau à cocher (une case par phase).

• **Formatage des manches** : il inclut les classiques dual-rate et exponentiel, gérés par demi-course mais aussi des courbes ou des fonctions mathématiques ($x > 0$, $x < 0$, $|x|$, $f > 0$, $f < 0$, $|f|$), avec activation par interrupteur et/ou phase(s) de vol. Par défaut (pas de ligne créée pour le manche), le gain est de 100% et aucune différenciation n'est appliquée.

• **Assignment des fonctions de pilotage aux servos** : elle est directement réalisée au sein des mixages (menu Mixeur), uniquement du type libre (aucun mixage pré-programmé). A chacune des 16 voies peut être affectée plusieurs entrées (ou aucune, si inutilisée) sous la forme d'une ligne de mixage par entrée, avec les attributs suivants : une source (manche, potentiomètre, interrupteur physique ou virtuel, valeur statique, autre voie, entrée PPM), des conditions d'activation (interrupteur, phase(s) de vol), des paramètres (taux de mixage, offset de l'entrée, prise en compte ou non du formatage du manche de l'entrée et/ou de son trim, courbe ou fonction mathématique ou différentiel, alarme liée, retards ou ralentissements aux changements d'état) et des modalités d'interaction avec les autres mixages -de la même voie- qui le précèdent (addition, remplacement, multiplication).

• **Formatage des servos** : il inclut, pour chaque voie, les classiques



La modification de l'électronique de la 9x, pour le flashage, n'est pas très compliquée. Elle nécessite juste un peu de doigté et surtout un fer à souder à pointe fine.

DEBRIEFING



BIEN VU

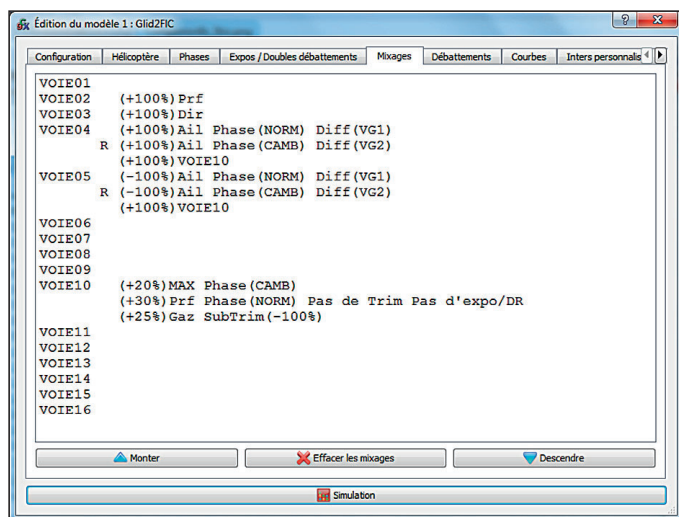
- Puissance de programmation sans concurrence
- Facilité de programmation et d'utilisation
- Ergonomie parfaite
- Stabilité et pérennité



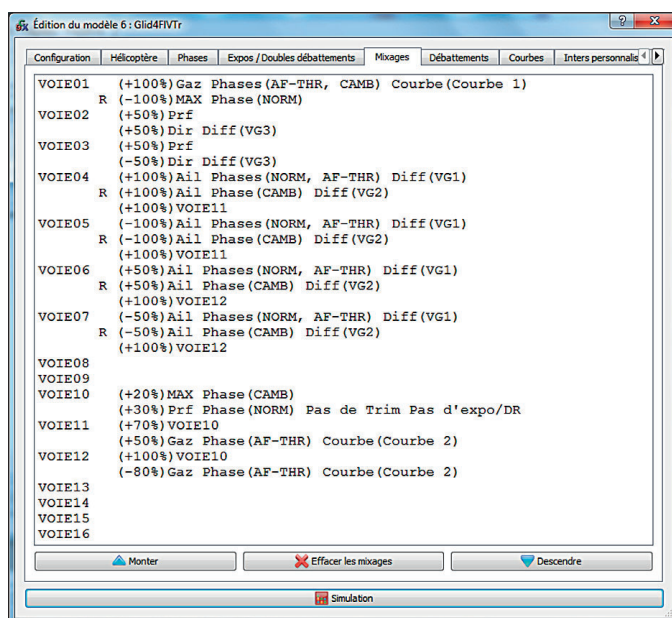
A REVOIR

- Pas assez connu !

EXEMPLE DE MIXAGES

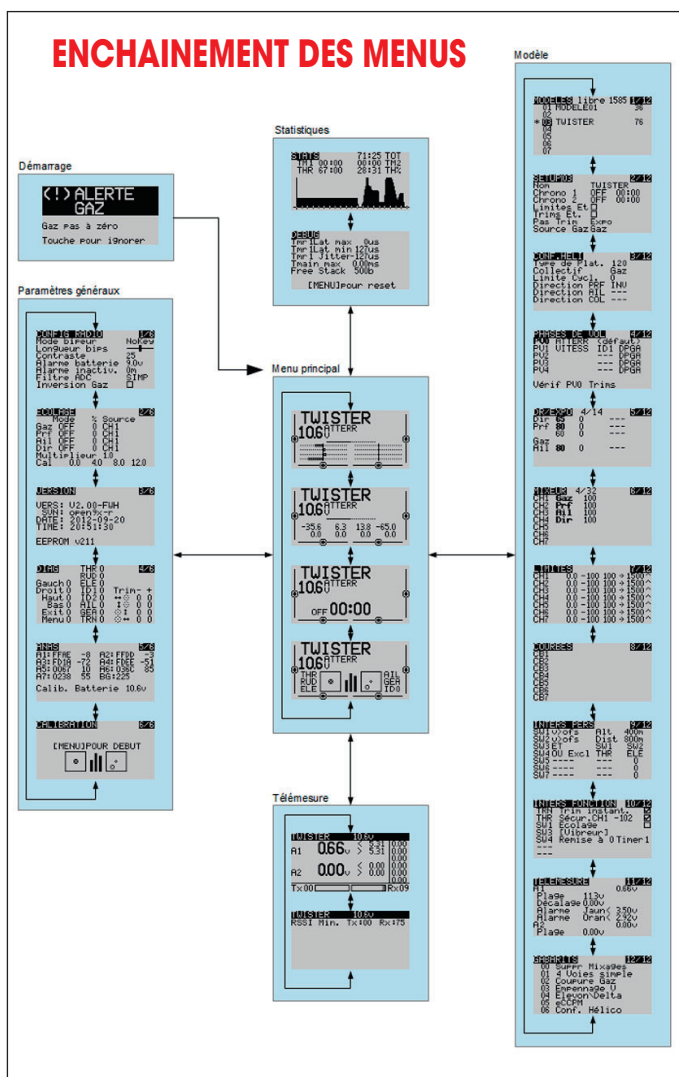


Un exemple de mixages, pour un planeur pur à deux ailerons et deux phases de vol. La fonction courbure (snap-flap en phase « norm » avec couplage indépendant du trim et du dual-rate de profondeur, courbure de griffe en phase « camb » et ailerons relevés en AF via le manche de gaz) est créée dans la voie 10, elle-même utilisée en entrée des voies d'ailerons (4 et 5). Ces dernières ont deux entrées du manche d'aileron, de manière à obtenir un taux de différentiel variable par phase et réglable en vol grâce à deux variables globales (VG). Très simple et puissant.



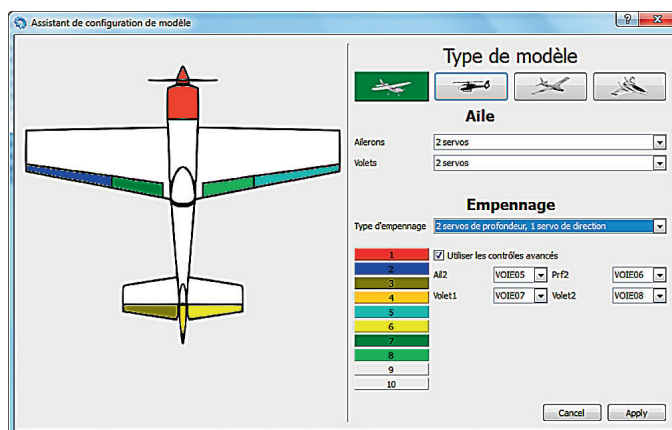
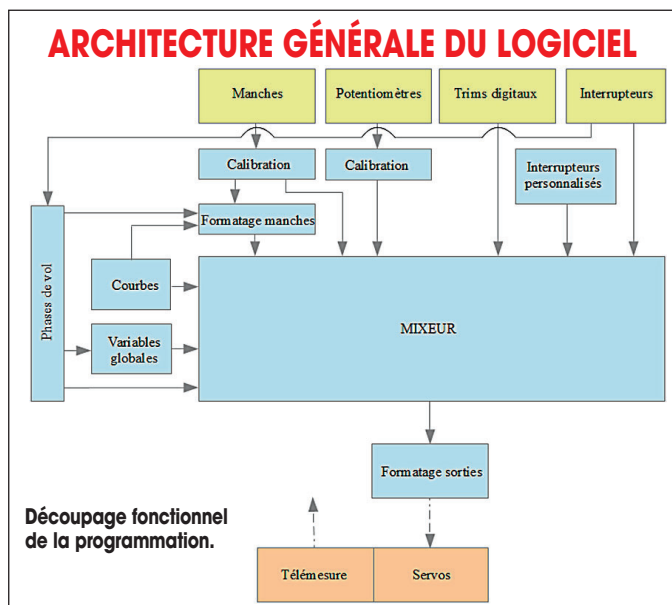
Un autre exemple de mixages, plus complexe : planeur quadro-flap avec empennage en V, moteur et AF croco partageant le manche de gaz (par 1/3 de course) et trois phases de vol. Tous les couplages quadroflap y sont, avec snap-flap, différentiel et courbure différenciés par phase. Bref, c'est très complet et pas si compliqué que cela.

ENCHAÎNEMENT DES MENUS



On peut voir ici l'enchaînement des menus sur les premiers écrans. La structuration est limpide, avec un titre clair par écran.

ARCHITECTURE GÉNÉRALE DU LOGICIEL



Pour les tout premiers pas, l'assistant de configuration de modèle permet de préprogrammer certaines fonctions classiques. Mais ne dispense pas de l'apprentissage de la fantastique logique d'Open9x.

sub-trim, sens et butées de fins de course (servant aussi de gain). On y trouve aussi un très pratique transfert des trims vers les sub-trims et, en option : un sub-trim bis, non affiché dans les positions des servos (idéal pour régler un neutre de manière transparente). Il y a aussi un découplage du gain par rapport au sub-trim (toute la course est décalée au lieu du centre seulement), indispensable à la linéarité des voies à plusieurs entrées ou dans le cas d'un sub-trim recentrant un servo volontairement décalé pour privilégier une demi-course (ex : volet de courbure utilisé en AF croco). Sinon, le gain est adaptatif, les fins de courses physiques étant conservées après le réglage du sub-trim.

- **Valeurs mutualisables** : il s'agit de huit courbes (3, 5, 9 ou 17 points, avec abscisses fixes ou éditables) et de cinq variables globales, destinées aux formatages de manches et aux mixages. Les premières peuvent être utilisées par plusieurs lignes simultanément, en direct ou en abscisse inversée (attribut « ! »). Les secondes peuvent remplacer n'importe quel taux (ex : différentiel d'ailerons) et, pour les réglages en vol, être temporairement liées à un organe de commande (ex : trim de gaz, inutilisé en planeur).

- **Interrupteurs virtuels** : au nombre de 12 et librement paramétrables, ils utilisent des conditions logiques (comparaisons <, <=, >, >=, !=, opérateurs ET, OU, OU-Exclusif) entre deux opérantes (valeur, interrupteur physique ou virtuel (chaînage), position de manche, valeur de sortie de voie et, le top, donnée de télémesure). Par exemple, il est ainsi possible de programmer une coupure du moteur automatique à partir d'une certaine altitude.

- **Fonctions personnalisées** : au nombre de 16 et librement paramétrables, elles permettent de gérer les sécurités de voies (forçage à une valeur définie), l'activation de l'écolage (complet ou voie par voie), le trim instantané (transfert des positions de manches vers les trims), le mode vibreur, les resets compteurs, les lectures de sons ou la modification en vol des variables globales. A chacune peut être associé un interrupteur physique, momentané ou virtuel.

- **Menu hélicoptère (optionnel)** : il facilite la programmation des hélicoptères, néanmoins aisément réalisable sans ce menu (cf. la notice wiki d'Open9x).

- **A noter aussi, un point important** : un gain d'environ 75% sur Open9x donne un débattement équivalent à 100% sur la majorité des radios commerciales. La bonne pratique consiste donc à utiliser systématiquement des dual-rates avec une valeur inférieure à 100%. Ce qui,

par la même occasion, laisse de la latitude pour les éventuelles corrections aux trims avant écrêtage des fins de course.

Notre avis

Pour bien situer la logique Open9x :

La majorité des radiocommandes commerciales, excepté celles de Multiplex, proposent un panel de mixages prédéfinis et à la structure figée (delta, etc.), partant des organes de commande avec une éventuelle réaffectation des voies. Ces mixages sont activés au cas par cas et épaulés, en cas d'insuffisance, par des mixages libres. Le tout peut poser des problèmes de cohérence dans le cas de modèles complexes, avec parfois la sensation de lutter contre le soft plutôt que réellement programmer son modèle.

A contrario, Open9x autorise un routage totalement libre, partant de la finalité (les gouvernes, via les voies) et permettant, par chaînage, de créer des fonctions complexes et hiérarchisées. Par exemple, la sortie d'un mixage peut ne pas être affectée physiquement à un servo et servir d'entrée à d'autres mixages (ex : fonction « courbure d'aile »). Ou bien, un mixage peut influencer son voisin en le multipliant ou le remplaçant suivant le contexte. De plus, Open9x permet la différenciation des mixages par groupe de phases de vol, ce qu'aucune radiocommande commerciale n'offre à ce jour.

Le bénéfice de cette approche est triple et fait d'Open9x une formidable boîte à outil :

La structure de mixage est univoque, un seul coup d'oeil dans le menu Mixeur permet de déterminer exhaustivement la provenance des ordres reçus par chaque servo.

La puissance de programmation est, objectivement, sans équivalence.

La souplesse, l'efficacité et la rapidité de programmation sont elles aussi sans équivalence.

La contrepartie à cette latitude quasi totale est que la construction et la structuration des mixages nécessitent un minimum de réflexion de la part de l'utilisateur, qui doit, en tout premier lieu, bien identifier son besoin et définir une stratégie minimale de programmation.

Sur la durée

Open9x, sur une radio 9x et bientôt sur une radio perso (base Arduino) a remplacé toutes mes radios « du commerce » depuis juillet 2012. Depuis cette date, j'ai suivi la majorité des évolutions, ayant contribué à en initier certaines.



Superbe radio conçue et réalisée par Christophe B. avec un logiciel maison et open-source (<http://code.google.com/p/bracame-rc-radio/>). Cela donne des idées ou des envies, non ?



Voilà un bel exemple de radio « perso, sobre et fonctionnelle, conçue et réalisée par Xavier Bonazzi pour les logiciels 9x.

Malgré une utilisation significative (environ 300 à 400 heures par an), aucune défaillance logicielle ou matérielle n'est survenue. Open9x est parfaitement stable et abouti, tandis que la fiabilité des très courants micro-contrôleurs Atmel n'est plus à prouver depuis bien des années.

D'autre part, l'ergonomie d'Open9x me ravit. La navigation est fluide et intuitive. Tout a été réfléchi pour simplifier au maximum l'utilisation. Le contraste avec les capacités de programmation quasiment infinies est flagrant et démontre la maîtrise dont font preuve ses développeurs. De plus, la mémoire limitée du processeur de base de la 9x permet quand même l'utilisation des 16 modèles grâce à l'utilisation dynamique de la mémoire : au lieu de prendre une place définie à l'avance par la totalité des fonctions utilisables (comme la plupart des radios), chaque modèle ne consomme que le

strict nécessaire aux fonctions utilisées, au bénéfice aussi de la rapidité d'exécution des calculs.

Une nouvelle référence absolue

Open9x est de très loin le meilleur logiciel pour radio que j'ai jamais eu l'occasion de tester. Certains brillent par leur puissance comme dans le haut de gamme Futaba, d'autres sont très intuitifs comme chez Graupner ou JR, d'autres encore sont rigoureusement structurés comme chez Multiplex, mais aucun à ce jour ne présente autant de qualités cumulées qu'Open9x. Supplantant dans leur domaine d'excellence tous les meilleurs softs commerciaux, il a complètement bouleversé ma vision des logiciels radios et constitue une référence absolue dans ce domaine...