

Pour remplacer ma vieille boîte perroquet du siècle dernier, j'avais découvert la platine BX184 sur le site de funkamateuer, et mon ami José m'a ramené un kit de Weinheim. Pour ceux qui ne veulent pas cannibaliser leur beau micro, le site vend une coquille vide pour ce kit.

Le câblage est prévu pour plusieurs marques de TRX (ICOM ; YEASU et d'autres TRX en modifiant le câblage sur le connecteur prévu pour sur le BX184).

Mon MH31 avait déjà un bip bip pour essais hyper made F6CXO, il fallait ragouter la nouvelle platine. Après démontage de la vieille platine micro, on s'aperçoit qu'on aura plus accès au switch arrière utilisé pour le bip bip, et qu'il faudra rajouter un inter.

Le micro a dans le bas 2 petites plaques de ferraille destinées simplement à faire du poids pour faire «mieux» HI. Un trou, quelques coups de lime et l'inter est en place, vérifier en posant la platine qu'il ne touche pas.



La platine après avoir terminé son câblage est prête pour être mise en place.

Platine et micro montés



Le BIP BIP



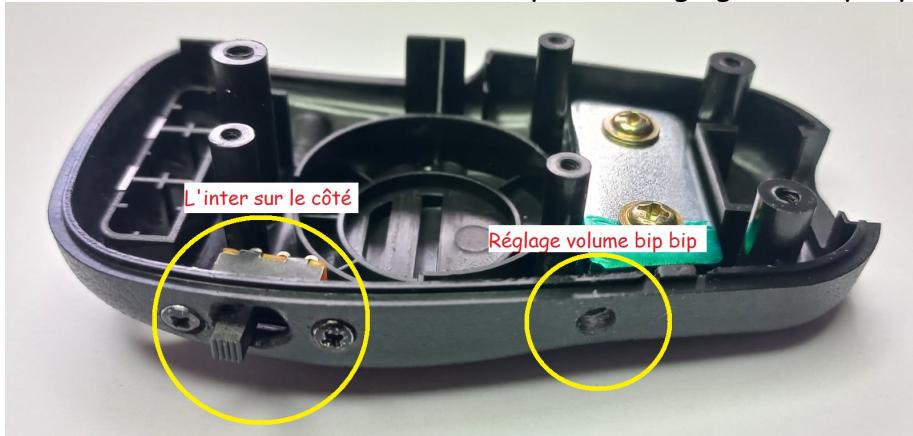
Tout est là, le bip bip sur le côté



Le Bip bip est fixé à la colle chaude la résistance de réglage niveau sur le côté.

Christophe F1JKY a aussi intégré d'une belle manière un bip bip et un BX184 dans son MH31.

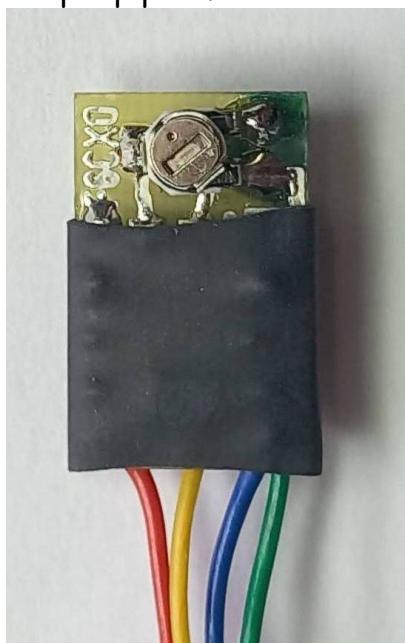
L'inter est sur le coté avec un trou pour le réglage du bip bip.



Tout est en place



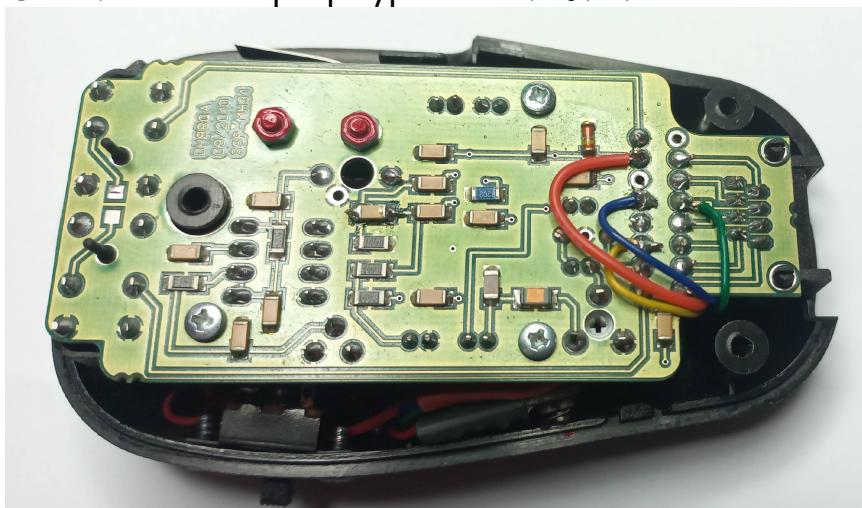
Le bip bip prêt.



Vue arrière sur le trou de réglage niveau du BX184



Le BX184 et le bip bip hyper chez F1JKY.



Le câblage pour FT817/818.



<https://www.dh8bqa.de/bx-184/>

https://www.box73.de/product_info.php?products_id=2360

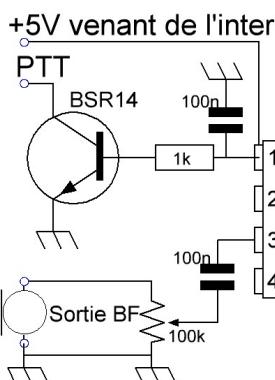
Suite à la première version de 2011, et après avoir acheté un FT817 ; il était nécessaire de l'adapter dans le nouveau micro.

Le montage est le même, le PIC aussi.

https://f6cxo.fr/wafx_res/Files/Bip_bip_a_PIC_pour_essais_hypers_sur_FT817.pdf. Le .HEX est disponible sur le site à la ligne au dessus.

L'avantage, c'est que du 5V est dispo, et un inter à glissière réutilisable pour cette application.

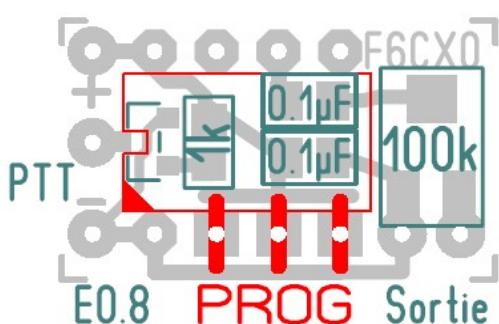
Ne disposant que d'un RT, une commande PTT par transistor a été rajoutée.



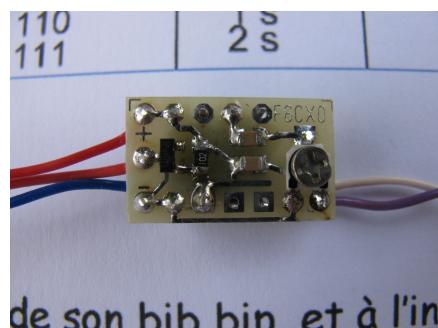
Configuration pin (0;1;2)	Durée ton	Durée entre 2 tons
000	rien	000
001	100 ms	100 ms
010	200 ms	200 ms
011	200 ms	500 ms
100	500 ms	200 ms
101	500 ms	200 ms
110	1 s	1 s
111	2 s	1 s

Maintenant, il ne reste qu'à câbler, choisir le rythme de son bip bip, et à l'intégrer dans le micro. On regarde dans son micro la position de l'inter à glissière TONE, on dessoude les fils sur l'inter et selon notre position 1 ou 2 on court-circuite ou non le condensateur de $0.33\mu F$.

Le circuit imprimé 18*11 mm

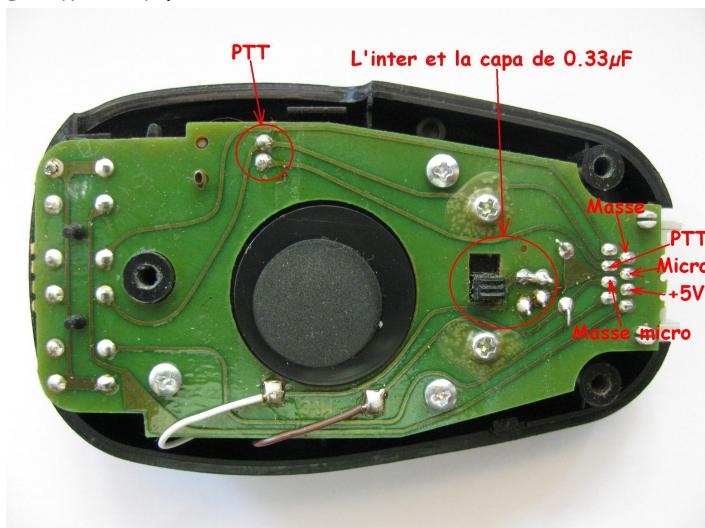


Câblé



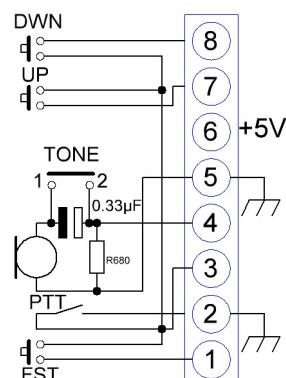
Le choix du rythme se fait par mise à la masse ou au + des pins 5,6 et 7.

Le micro MH-31



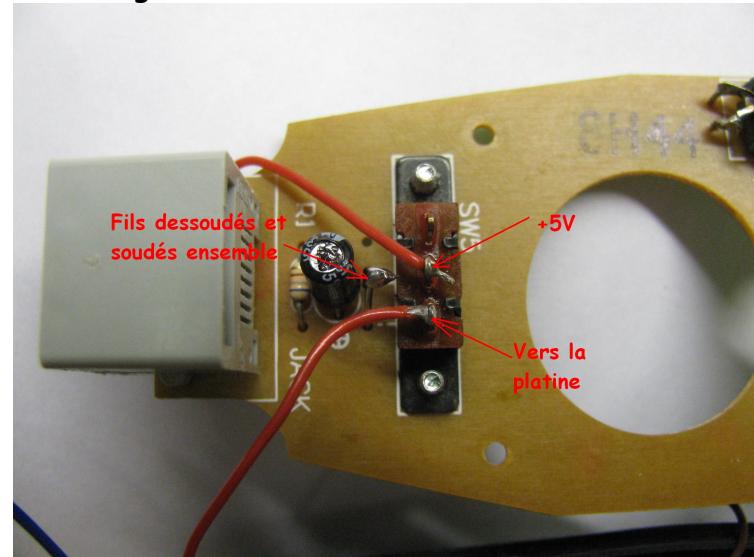
et son schéma.

MH-31

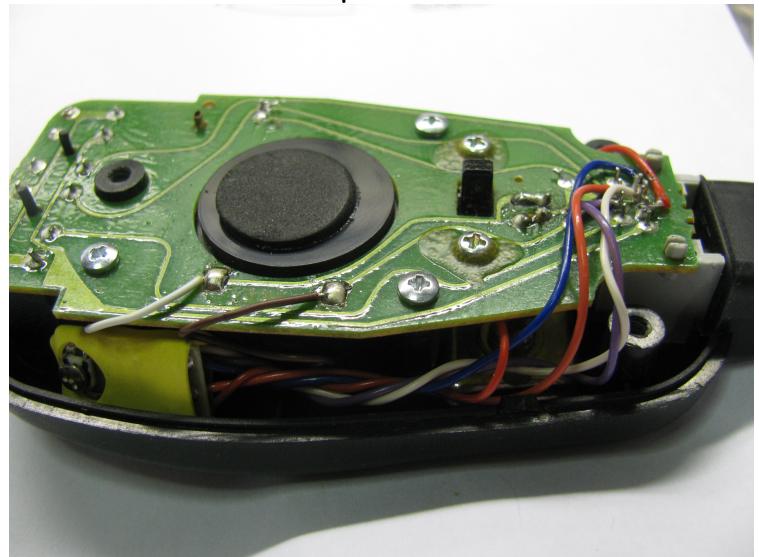


Le commutateur TONE qui court-circuite ou non le condensateur de $0.33\mu F$. Le +5V est soudé à la cosse centrale et l'alim 5V du Circuit sur la cosse du bas (voir photo). Le reste du câblage consiste au -5V, au PTT et au + & - micro.

Le câblage du commutateur TONE



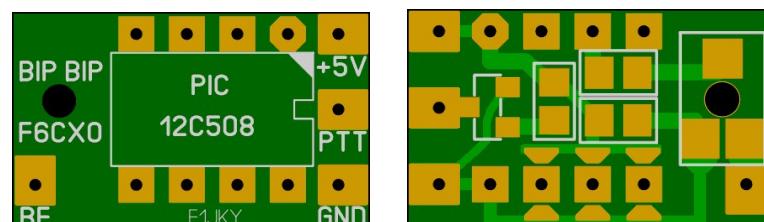
Le CI BIP BIP mis en place.



Il ne reste plus qu'à régler le niveau de BF grâce à la R ajustable.

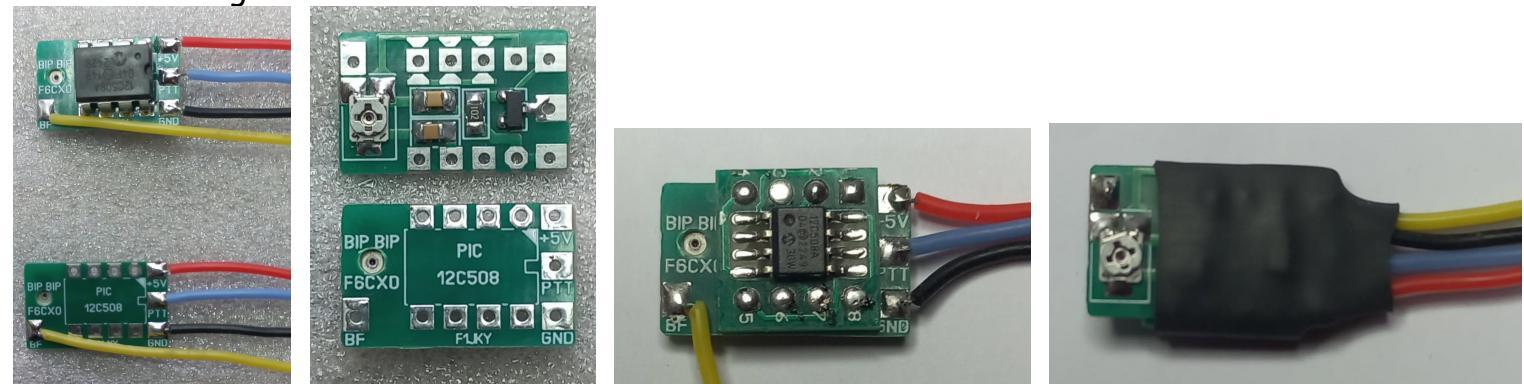
Ce petit montage peut être incorporé dans tout micro avec une alimentation +5V, sinon un petit régulateur sera nécessaire.

Bonne bidouille.



Des nouveaux CIs sont dispo chez CXO.

Vu sur le câblage des nouveaux CI.



Versions montées et câblées par Christophe F1JKY.

Version à 12C508 CMS sur adaptateur.

f6cxo@arobaseorange.fr

f1jky@arobasewanadoo.fr

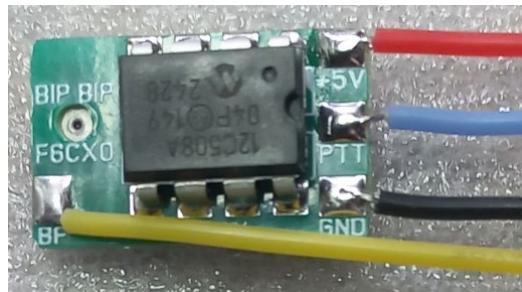
Mon site : <https://f6cxo.fr/>

le site de Christophe <https://f1jky.fr/>

Add-On BIP BIP F6CXO

Programmation des Fusibles du PIC + Infos en vrac

By **F1JKY**



BIP BIP à Pic by F6CXO Gérard

Introduction :

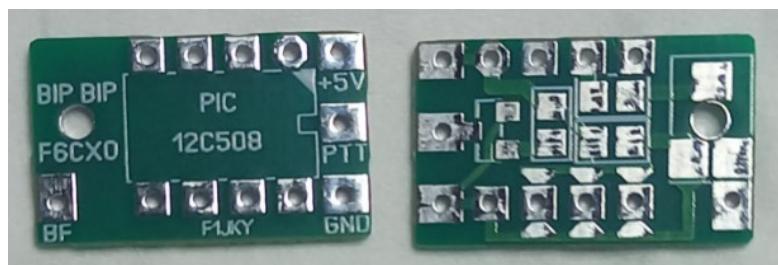
Pour donner suite à la parution de l'article de Gérard F6CXO sur son BIP BIP à Pic à intégrer dans un Micro à des fins de pointage d'antennes (plus particulièrement de Parabole mais pas que ...) dans un cadre de trafic en Hyperfréquence Ham, je voulais rajouter quelques informations supplémentaires pour que les Oms puissent se débrouiller seuls à l'assemblage de leur petit module.

Vous pouvez retrouver l'article de F6CXO dans la revue Radio-REF ou HyperFR ou directement la version originale sur son site internet.

On y va ... :

Le but est de vous donner quelques informations sur notamment le positionnement des fusibles du PIC lors de la programmation de celui-ci, sinon, cela ne fonctionnera pas et comme le choix du PIC s'est porté sur un 12C508A, c'est du « one shot » ... comprenez par-là que lorsque vous l'aurez programmé, bien ou mal, il vous sera impossible de l'effacer pour le reprogrammer, c'est sans filet ... donc autant mettre toutes les chances de son côté !

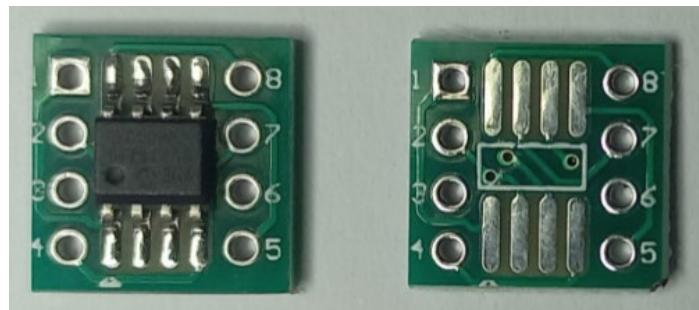
En aparté, j'ai repris le design de Gérard avec son accord et j'ai fait tirer qqes PCB que je lui ai transmis. Donc Gérard F6CXO a quelques PCB d'avance. Pas des tonnes non plus ... mais s'il y a vraiment de la demande, on peut toujours en refaire fabriquer moyennant un peu de délais ! Je suis sûr que si vous demandez gentiment, Gérard se fera un plaisir de vous en rétrocéder quelques-uns, hi ! ;o))



New PCB by F1JKY

Avant de passer dans le vif du sujet, je voulais aussi vous donner une petite astuce pour gagner encore un petit peu d'épaisseur en utilisant un PIC 12C508A CMS en boîtier SOIC8 + un petit PCB d'adaptation SOIC8 vers DIL8 trouvable sur Aliexpress par exemple. Pour les PIC 12C508A en SOIC8 ou en DIL8, je vous conseille de les commander chez RS Particuliers pour éviter les problèmes de fake (SOIC8 RS-4670827 / DIL8 RS-3281707 / pot 100K RS-4617258).

Ce PIC en SOIC8 + son PCB d'adaptation au PCB BIP BIP F6CXO, vous permet de gagner environ 1 à 2mm d'épaisseur, ce qui peut être intéressant pour le loger encore plus facilement dans un micro à main d'origine ou la place est parfois comptée. Le mieux aurait été de refaire complètement un PCB intégrant le SOIC8, mais un manque de temps et le fait qu'il est plus facile pour tout à chacun de programmer / souder un PIC en DIL8 m'a fait mettre en stand-by ce projet, d'où cette solution alternative et de facilité qui se révèle tout aussi polyvalente.



PCB d'adaptation SOIC8 vers DIL8

avec un exemple d'un PIC 12C508A CMS soudé dessus



PIC 12C508A SOIC8 soudé sur la platine BIP BIP F6CXO

Et maintenant ... :

Passons maintenant au positionnement des fusibles pour la programmation du PIC 12C508A ...

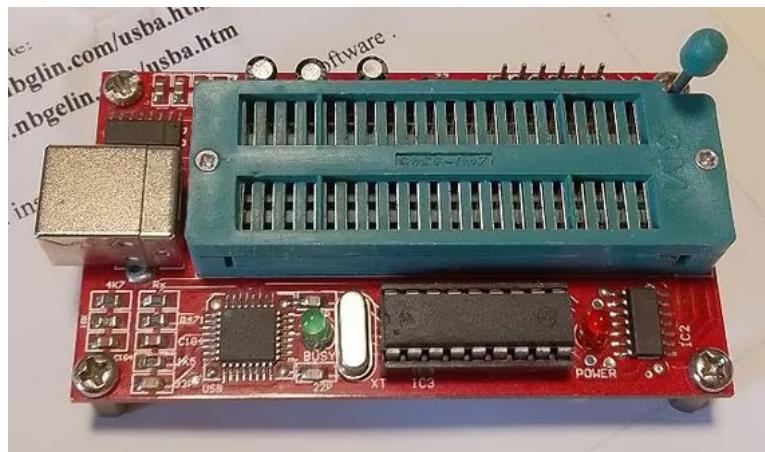
Lors de la programmation d'un PIC avec une interface et un PC, il faut s'assurer que « les fusibles » soient positionnés correctement dans le programme qui gère l'interface de programmation. Ceux sont en fait un positionnement de Bits à une adresse bien particulière permettant de « dire » au PIC comment il doit fonctionner pour exécuter son programme (les informaticiens vont crier en lisant cela, mais faisons simple afin de saisir le pourquoi pour le néophyte).

Quand les choses sont bien faites par le créateur du programme, ces fusibles sont en principe prépositionnés dans son programme mais ce n'est pas toujours le cas et les interfaces de programmation ne prennent pas toujours en compte cette information lorsqu'elle est présente dans le programme, d'où l'importance d'y faire attention.

Sur le site de Gérard F6CXO, vous pourrez récupérer le fichier « .HEX » qui contient le programme du BIP BIP et qu'il faudra utiliser pour programmer le PIC via votre interface et son logiciel associé.

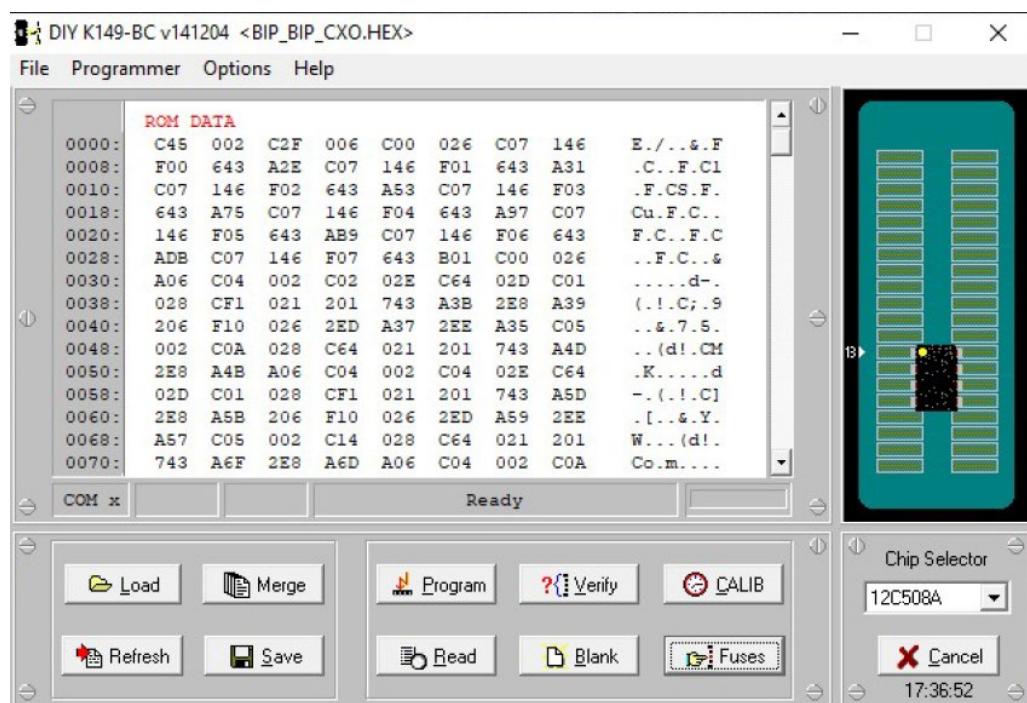
Gérard F6CXO utilise une interface différente des miennes et ayant constaté des différences pour la programmation du PIC et surtout de ses fusibles suivant la logique utilisée dans les logiciels de programmation, je vais vous donner ci-dessous ce qu'il faut cocher sur ces différentes solutions pour programmer correctement le PIC.

- 1- Gérard F6CXO utilise ce programmateur (K149) qui est maintenant assez ancien, dédié pour la programmation des PIC et difficilement trouvable sur les sites de VPC Chinois (voir le K150 qui semble très similaire et encore vendu). L'avantage, c'est qu'il a une interface USB et que son programme associé MicroPro tourne encore sur Win10 Pro.



Interface de Programmation pour PIC utilisé par F6CXO

Le logiciel de programmation associé au programmateur de Gérard se présente ainsi :



Le fichier « BIP_BIP_CXO.HEX » est chargé dans le logiciel et le bon type de PIC est sélectionné

Passons à la vérification du positionnement des fusibles dans le logiciel :



Les fusibles doivent être positionnés comme ci-dessus, rien d'extraordinaire mais c'est important, surtout concernant le paramètre de l'oscillateur qui va indiquer au PIC comment il doit générer son horloge.

- 2- Voici maintenant une très vieille interface de programmation pour PIC fonctionnant avec un port COM et qui va rappeler bien des souvenirs à certains d'entre vous ... le [PIC-01](#) et son logiciel associé [IcProg v1.06c](#).

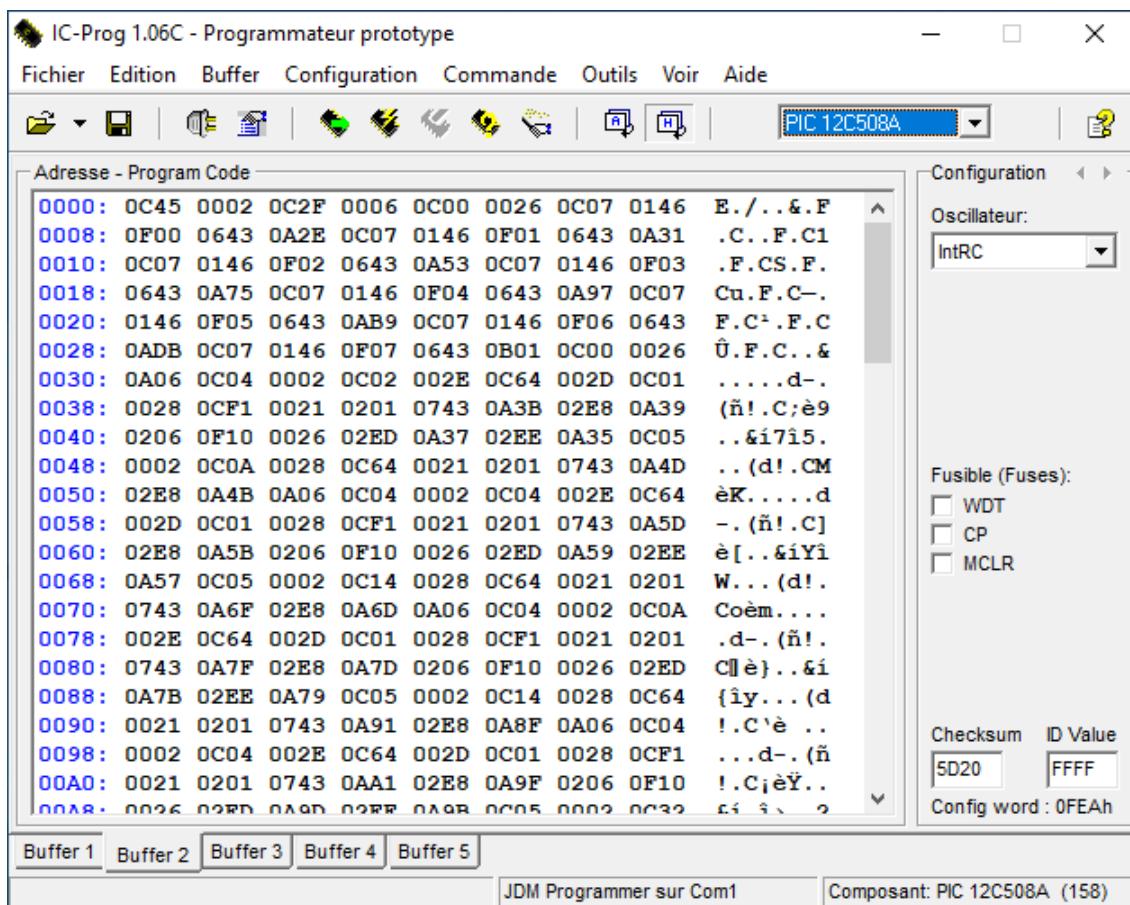
Oui, je sais, il y a bien mieux de nos jours, mais cela fait encore le taf et il en faut pour tout le monde ... le pire, c'est que cela fonctionne encore très bien sous Win10 Pro 64bits, un PC de bureau avec une carte mère assez récente dont il a fallu tirer un câble dB9 Mâle sur la façade arrière pour l'avoir à disposition. Eh oui, malgré des légendes urbaines comme quoi il n'y a plus de port COM sur nos PC depuis au moins 10 ans et que Windows 10 ne sait plus faire Figurez-vous que bon nombre de carte mère intègre encore un port COM mais comme il s'agit d'une vieille techno dont seulement certains dinosaures l'utilisent encore (dont je fais partie), ce n'est pas mis en avant sur les documentations et cela se comprend. Il faut bien étudier la datasheet de sa carte mère et câbler un câble sur les Pins prévues à cet effet en récupérant un câble sur une de vos anciennes bécane avant qu'elle ne parte à la déchèterie.

Bref, je referme cet aparté bien que je me serve encore régulièrement de cette interface pour des anciens PIC car j'avoue avoir eu quelques incompréhensions sur des interfaces plus récentes dont celle que je vous présenterai en dernier et dont il a fallu comprendre la logique utilisée pour retrouver ses petits sur le positionnement des fusibles, même si ce n'est pas si évident que cela car pas grand-chose n'est clairement expliqué (ça m'a couté plusieurs PIC pour la cause ... grrr ...).

Voici l'interface PIC-01 de la Sté SEEIT :



Passons au logiciel IcProg v1.06c que j'utilise avec le PIC-01 :



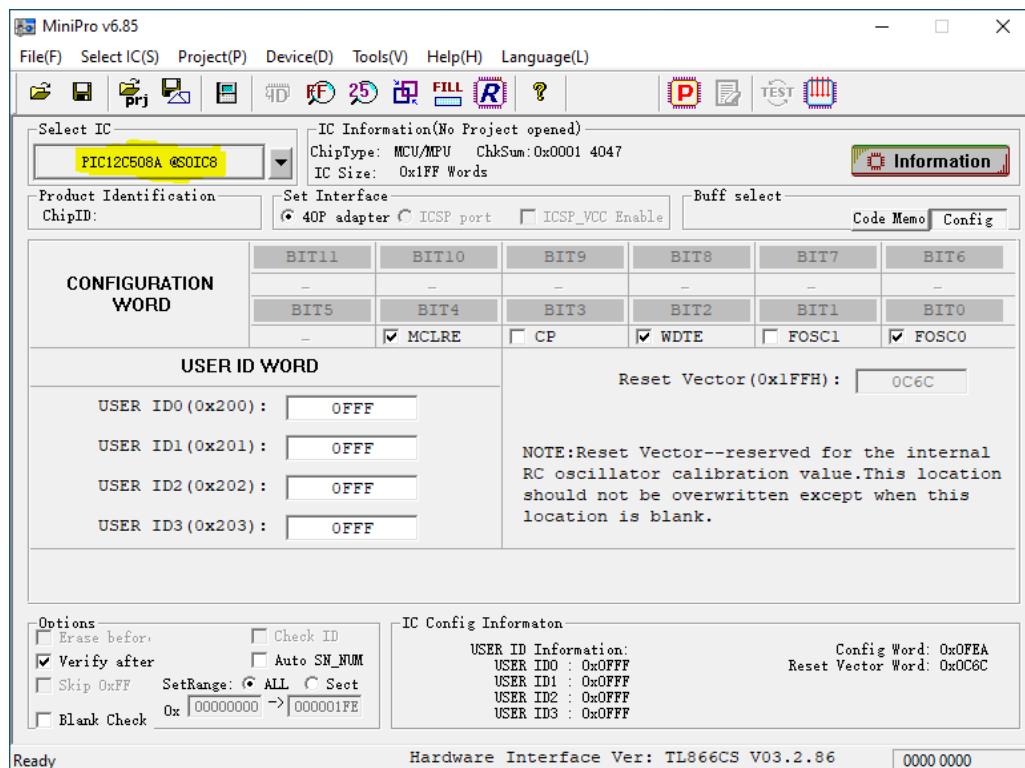
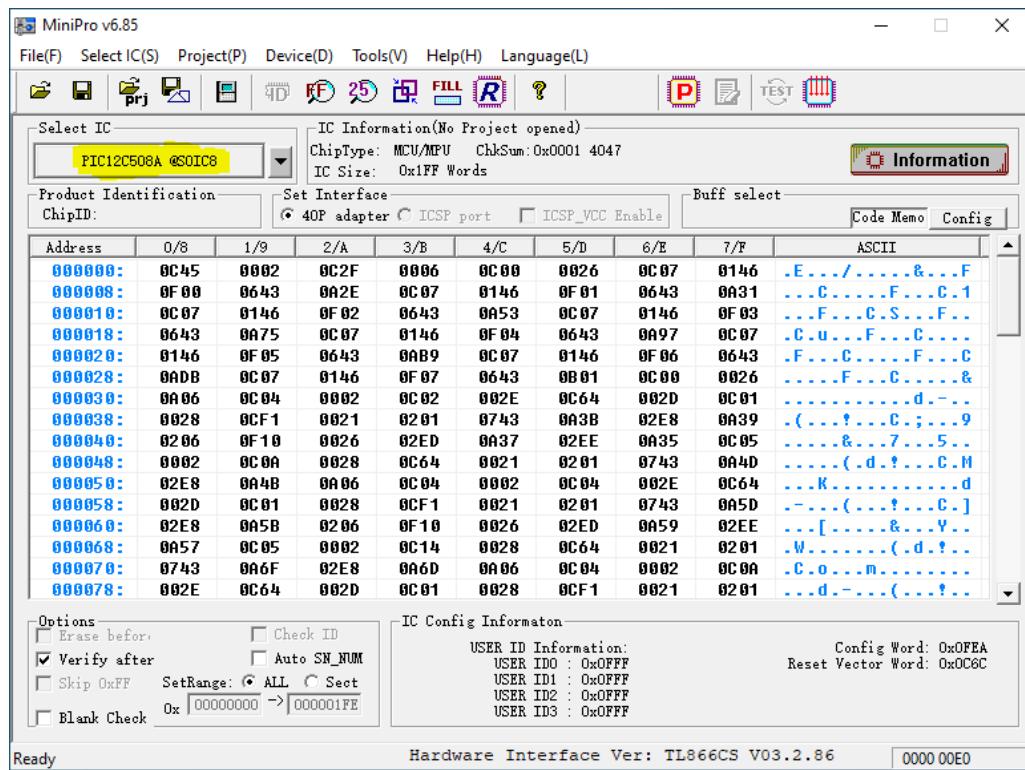
On distingue sur la droite la configuration des fusibles et de l'oscillateur à utiliser pour la programmation du PIC.

- 3- Passons maintenant à une interface plus récente et qui a été assez diffusée à son époque, le **TL866CS** et son logiciel dédié le **MiniPro v6.85**.

Cette interface m'a rendu de bons services mais il y a toujours eu un loup sur la programmation des fusibles des PIC ... les positionnements des fusibles ne semblaient pas correspondre à la logique dont j'avais l'habitude sur les anciennes interfaces qui étaient simplifiées pour l'utilisateur. En gros, sur l'interface de Gérard ou de IcProg, quand on coche une case, on active la fonction ... logique me direz-vous ... sauf que sur le TL866CS ou le tout dernier T76, c'est une logique inverse qu'il faut avoir. Le mieux étant de se reporter à la datasheet Microchip.



Quoi qu'il en soit, j'ai pris le temps de gratter un peu en sacrifiant qques PIC pour la bonne cause et voici la config qui va bien chez moi pour programmer un **PIC 12C508A SOIC8** pour le BIP BIP de F6CXO :



Il faut Cocher : **MCLRE / WDTE / FOSCO**

Pour comprendre cette logique inversée par rapport à nos anciennes interfaces de programmation, le mieux étant de se reporter à la datasheet de programmation du 12C5xx de Microchip :

PIC12C5XX

3.0 CONFIGURATION WORD

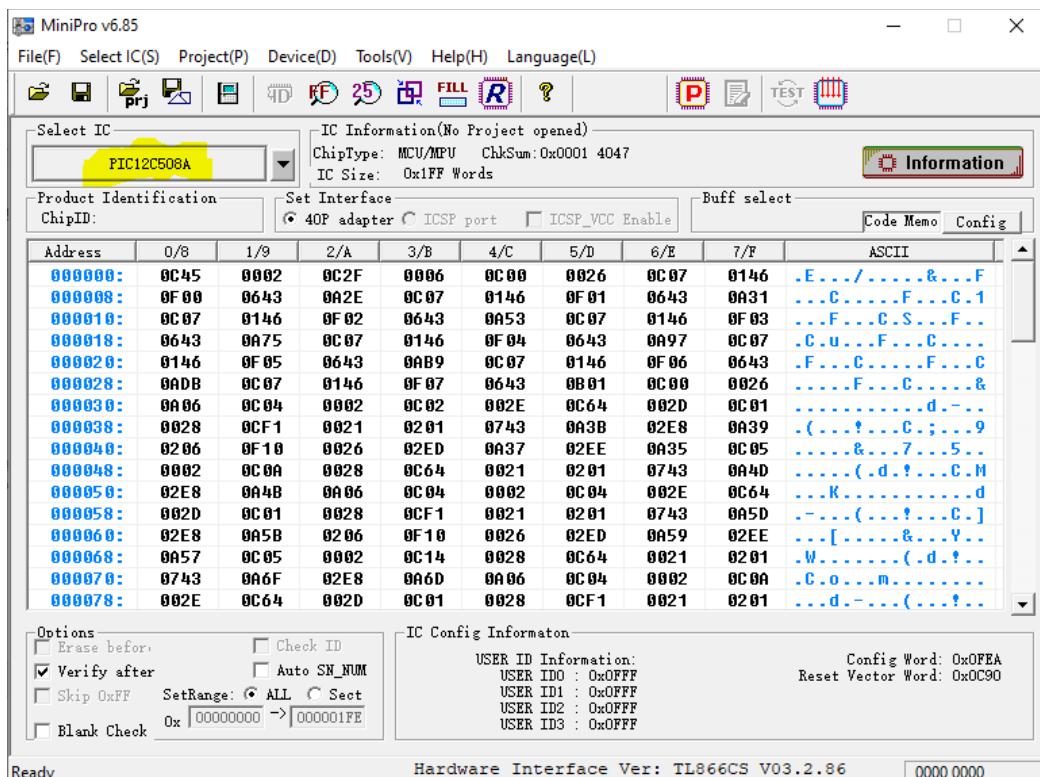
The PIC12C5XX family members have several configuration bits. These bits can be programmed (reads '0'), or left unprogrammed (reads '1'), to select various device configurations. Figure 3-1 provides an overview of configuration bits.

FIGURE 3-1: CONFIGURATION WORD BIT MAP

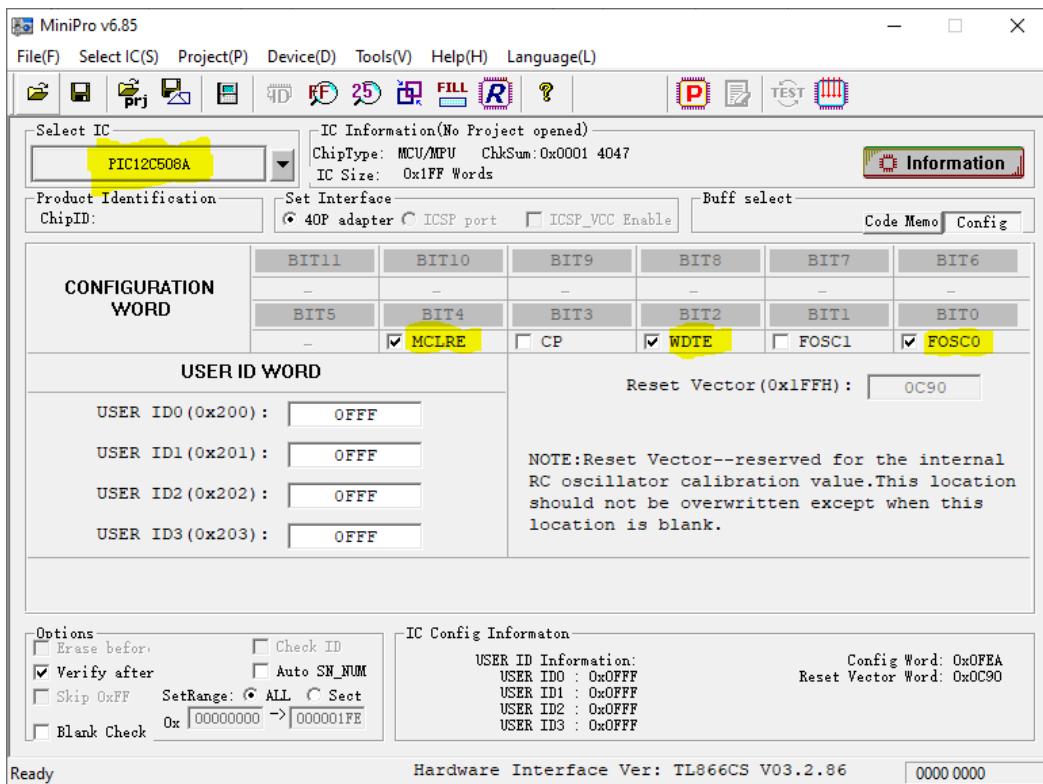
Bit Number:	11	10	9	8	7	6	5	4	3	2	1	0
PIC12C5XX	—	—	—	—	—	—	—	MCLRE	CP	WDTE	FOSC1	FOSC0
bit 11-5: Reserved: Write as '0' for PIC12C5XX												
bit 4: MCLRE: Master Clear Enable bit 1 = MCLR pin enabled 0 = MCLR internally connected to VDD												
bit 3: CP: Code Protect Enable bit 1 = Code memory unprotected 0 = Code memory protected												
bit 2: WDTE, WDT Enable bit 1 = WDT enabled 0 = WDT disabled												
bit 1-0: FOSC<1:0>, Oscillator Selection Bit 11 = External RC oscillator 10 = Internal RC oscillator 01 = XT oscillator 00 = LP oscillator												

Le paragraphe précédent la figure 3-1 est important pour bien comprendre cette logique, logique qui est également appliquée dans les interfaces de programmation des programmeurs TL866CS & T76 de chez XGecu.

La configuration des fusibles est la même pour un **PIC 12C508A** en **DIL8** :



Ci-dessous les Fusibles associés :



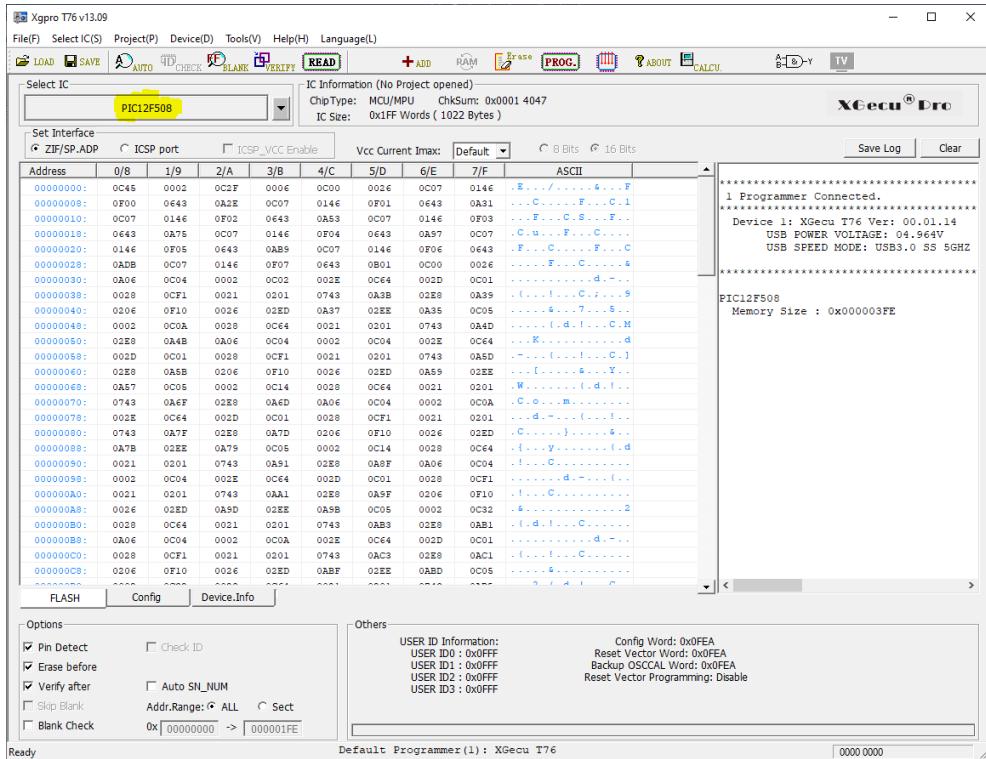
4- Un dernier petit Bonus avec le tout dernier programmeur arrivé au shack, le **T76** de chez **XGecu** :



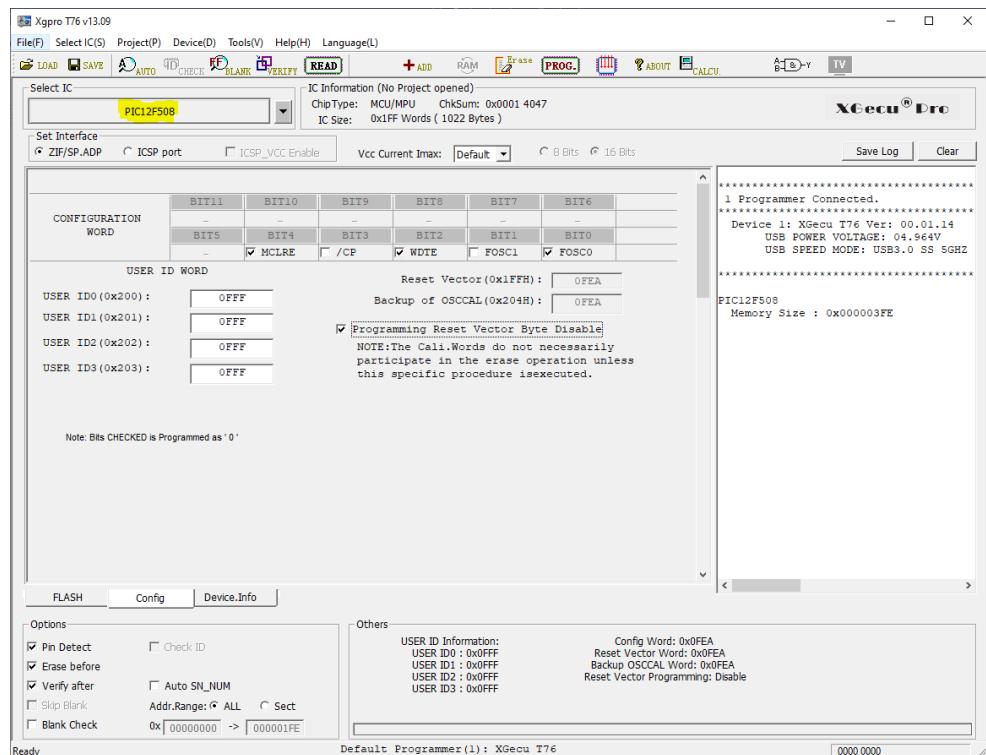
XGecu T76

Ce nouveau programmeur de Chip de chez **XGecu** est sortie depuis environ un an à l'heure ou j'écris ces lignes. Il est capable de supporter une très grosse partie des chip actuels (mémoires, µP ...) soit environ 39659 Chip avec la version 13.09 de son logiciel Xgpro. Ce logiciel est régulièrement mis à jour et permet l'accès à de nouveaux Chip. La curiosité, c'est qu'il est en principe capable de couvrir tous les chip des gammes antérieurs (T48/T56/TL866 série) sauf que le Pic 12C508A a disparu de la liste des composants supportés, seul le Pic 12F508 (réinscriptible) est présent.

Après un test rapide, cela n'empêche pas de lire un **12C508A** en sélectionnant le **12F508** dans la liste des chip disponible dans la **v13.09**, ce qui est assez logique, voyez ci-après le résultat obtenu :



Lecture du Pic 12C508A en sélectionnant un 12F508



On constate que les fusibles lus sont bien positionnés sur : **MCLRE / WDTE / FOSC0**

Pour la cause (hi !), j'ai programmé un **12C508A DIL8** avec la config ci-dessus en sélectionnant comme Chip un **12F508** et cela ne m'a posé aucun problème. Le PIC est parfaitement programmé et fonctionnel (je l'ai essayé en réel sur platine de test). Donc c'est bon à savoir concernant l'utilisation du T76.

Conclusions :

J'espère que ces quelques informations complémentaires vous aideront à mener à bien votre projet d'intégration de ce petit module mis au point par Gérard F6CXO et qu'il vous facilitera le pointage des antennes et donc la réussite de vos QSO.

A vos fers à souder, PC et à bientôt sur l'air !!

© Christophe PIALOT – [F1JKY](#) ©

[Site Internet de F1JKY](#)

- Janvier 2026 -

v3

Quelques Bonus :

