« GénéCall »
Générateur d'indicatif en Morse pour balises et relais
Entrée directe de l'indicatif au moment de la programmation du PIC

Jean-Paul YONNET
F1LVT / ADRASEC 38
F1LVT@yahoo.fr
www.F1LVT.com

Différentes montages permettant d'émettre un indicatif en Morse existent déjà. Le plus souvent l'indicatif est programmé dans le code source. Par exemple pour le montage intitulé « Générateur d'indicatif pour relais, transpondeurs et balises » [1], il faut modifier le programme en assembleur « .asm » pour le PIC 16F628, puis le compiler avec MPLAB pour obtenir le fichier « .hex » à entrer dans le PIC [2, 3, 4]. Et enfin, dernière étape, il faut programmer le PIC.

L'originalité du montage « GénéCall », c'est **d'entrer l'indicatif dans la mémoire du PIC au moment de sa programmation finale**. Il n'y a plus besoin de modifier un programme en langage C ou en assembleur et de le recompiler. Le programme **« genecall.hex »** est toujours le même. Il est téléchargeable sur le site www.F1LVT.com. C'est au moment où on programme physiquement le PIC qu'il faut ajouter l'indicatif dans la mémoire EEPROM. Il suffit d'utiliser un système de programmation comme un programmateur JDM ou PICkit3 par exemple, et d'ajouter les informations de l'indicatif dans la partie 'EEPROM Data' ou 'Data Mem'. Pour que l'indicatif puisse être entré dans la mémoire, il faut qu'il soit codé en hexadécimal. Cette procédure de codage est expliqué dans la partie « Codage de l'indicatif ».

Le problème de l'entrée de l'indicatif sans l'inclure dans le code a été difficile à résoudre. Les premiers essais ont été réalisés en utilisant les interrupteurs pour entrer les bits, en utilisant l'entrée GP3 (broche 4) pour passer en mode programmation et les 3 autres pour mettre la valeur à entrer (0 ou 1), valider l'entrée, et commuter entre entrée ou effacement. Mais la procédure d'entrée de l'indicatif s'est révélée être longue, fastidieuse, avec un risque élevé d'erreur. C'est pour cela que nous avons finalement choisi un système différent, en entrant l'indicatif au moment de la programmation du PIC. C'est relativement facile à faire et cela réduit les risques d'erreur en affichant clairement les données entrées en mémoire.

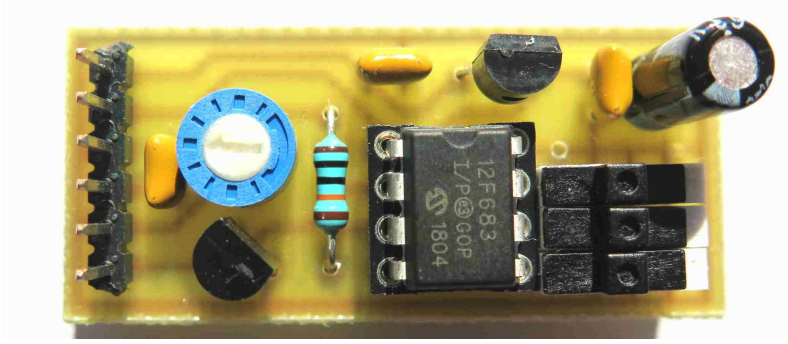

Photo 1 : « GénéCall » en version complète avec l'alimentation 12V et les 3 interrupteurs pour programmer la temporisation

Ce montage est conçu autour d'un PIC de type 12F683 (Photo 1). C'est un composant de petite taille, qui n'a que 8 broches, mais qui est relativement puissant : horloge interne à 8 MHz, résistances de 'pull-up' internes, 256 octets de mémoire, etc. Au démarrage, le programme du PIC lit la mémoire et décode l'indicatif. Ensuite en fonctionnement, le PIC génère l'indicatif en Morse à intervalle régulier. Cet intervalle est paramétré par 3 interrupteurs (ou 3 pontages), permettant d'obtenir une temporisation allant de 4 secondes à 3 minutes.

Fonctionnement du PIC

Le PIC 12F683 est alimenté par la broche 1 (entre 2V et 5V) et la broche 8 est à la masse. Les broches 5, 6, 7 sont utilisées pour paramétrer la durée de l'intervalle entre 2 indicatifs (Tableau I). Le signal audio de l'indicatif en Morse sort par la broche 3 sous forme de créneaux. Pour piloter le passage en émission, la commutation est effectuée par la broche 2. Cette sortie passe au niveau haut 100 ms avant l'envoi de l'indicatif.

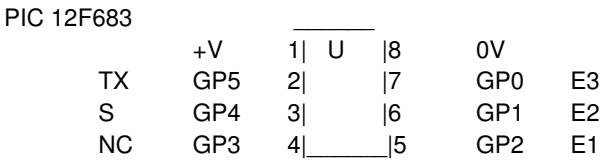

Tableau I : Entrées – Sorties du PIC 12F683

Intervalle entre 2 indicatifs

GP2	GP1	GP0	Intervalle
E1	E2	E3	(en secondes)
1	1	1	180 s
1	1	0	120 s
1	0	1	60 s
1	0	0	40 s
0	1	1	20 s
0	1	0	12 s
0	0	1	6 s
0	0	0	4 s

Tableau II : Paramétrage de l'intervalle de temps entre 2 indicatifs
0 = à la masse
1 = en l'air, tiré à +V par la résistance interne de pull-up

L'intervalle entre 2 indicatifs dépend de l'état des entrées E1, E2 et E3. Si les 3 entrées sont en l'air, l'intervalle est de 3 minutes (Tableau II). Si les 3 entrées sont à la masse, il est réduit à 4 secondes. Avec E3 en l'air, on obtient avec E1 et E2 les valeurs 180 s (3 minutes), 60 s (1 minute), 20 s et 6 s.

Codage de l'indicatif

L'indicatif est codé en hexadécimal sur 8 octets. C'est la juxtaposition des différents éléments de l'indicatif, codés 2 bits par 2 bits.

Les éléments de l'indicatif sont codés sous la forme :
-- les points : « 01 »
-- les traits : « 10 »
-- l'intervalle entre 2 caractères : « 00 ».
-- le code « 11 » est utilisé pour la fin de transmission de l'indicatif.

Premier exemple, pour « **F1LVT** » le code Morse se traduit par :

« . . - . . - - - - . - . . . . - - »

La traduction du Morse en binaire donne :
01 01 10 01 00 01 10 10 10 10 00 01 10 01 01 00 01 01 01 10 00 10

En regroupant les chiffres par groupe de 4 et en ajoutant des 11 à la fin de transmission jusqu'à 8 octets (64 bits), on obtient :
0101 1001 0001 1010 1010 0001 1001 0100 0101 0110 0010 1111 1111 1111 1111 1111
| | | | | | |

En hexadécimal, ce nombre se met sous la forme (8 octets avec des F à la fin) :
59 1A A1 94 56 2F FF FF

C'est cette valeur « 59 1A A1 94 56 2F FF » qu'il faut entrer dans l'EEPROM lors de la programmation du PIC 12F683.

Second exemple avec un indicatif plus long : « **HB9XYZ** »

-- Code Morse : « . . . . - . . . - - - - . - . . - - . - - - - - . . »
-- Conversion en binaire :
0101 0101 0010 0101 0100 1010 1010 0100 1001 0110 0010 0110 1000 1010 0101 1111
| | | | | | |

-- Lecture en hexadécimal des 8 octets avec des F à la fin :
« 55 25 4A A4 96 26 8A 5F »

Dans le cas où aucune valeur ne serait entrée dans les octets de la mémoire, le programme le détecte et il met à la place un pseudo caractère en Morse qui constitue le message par défaut (G&K).

La taille maximale de l'indicatif en Morse est de 8 octets dans le programme, soit 64 bits ou 32 éléments de 2 bits. Une lettre et la temporisation qui la suit sont écrites au maximum sur 10 bits. Un chiffre et la temporisation qui la suit sont écrits sur 12 bits. La dernière lettre fait au maximum 8 bits. Une lettre, un chiffre et 3 lettres font au maximum 50 bits, ce qui rentre sans problème dans la mémoire utilisée de 64 bits du PIC. C'est largement suffisant pour tous les indicatifs français. Les indicatifs plus longs, ceux à 6 caractères comme HB9XYZ, rentrent toujours car ils font au maximum 60 bits. La mémoire totale du 12F683 fait 256 octets alors que seulement 8 octets sont utilisés. En cas de besoin, il serait possible d'augmenter la taille de la mémoire utilisée, ce qui permettrait de transmettre des messages beaucoup plus longs, par exemple du type « CQ DX CQ DX DE F9XYZ ».

Programmation du PIC 12F683

Pour programmer le PIC 12F683, il faut (en plus du programmateur et des logiciels pour piloter ce programmateur) :
-- 1- Charger le programme « genecall.hex ».
-- 2- Préparer puis entrer dans la mémoire le nombre en hexadécimal qui correspond à l'indicatif à transmettre.
-- 3- Programmer le PIC 12F683.

Cette valeur entrée dans la mémoire lors de la programmation sera lue par le programme au démarrage du PIC. Elle sera ensuite décodée et stockée. Pour pouvoir transmettre l'indicatif en Morse, cette valeur sera convertie en traits et en points.

Certains dispositifs de programmation, comme les programmateurs JDM, sont reliés par une connexion RS232. Il faut utiliser un logiciel de programmation comme WinPIC800 ou PICPgm pour charger le code à programmer.

La Figure 1 montre les écrans qui apparaissent avec PICPgm. Quand le fichier à entrer dans le PIC, « genecall.hex », a été chargé dans PICPgm, on voit apparaitre le code sous l'onglet « Code Mem » (Figure 1-a). Sous l'onglet « Data Mem » qui correspond à l'EEPROM du PIC, la page initiale est remplie de FF. C'est là où il faut entrer l'indicatif en hexadécimal « 59 1A A1 94 56 2F FF » (Figure 1-b).

Quand tout est prêt, il ne reste plus qu'à programmer le PIC 12F683.

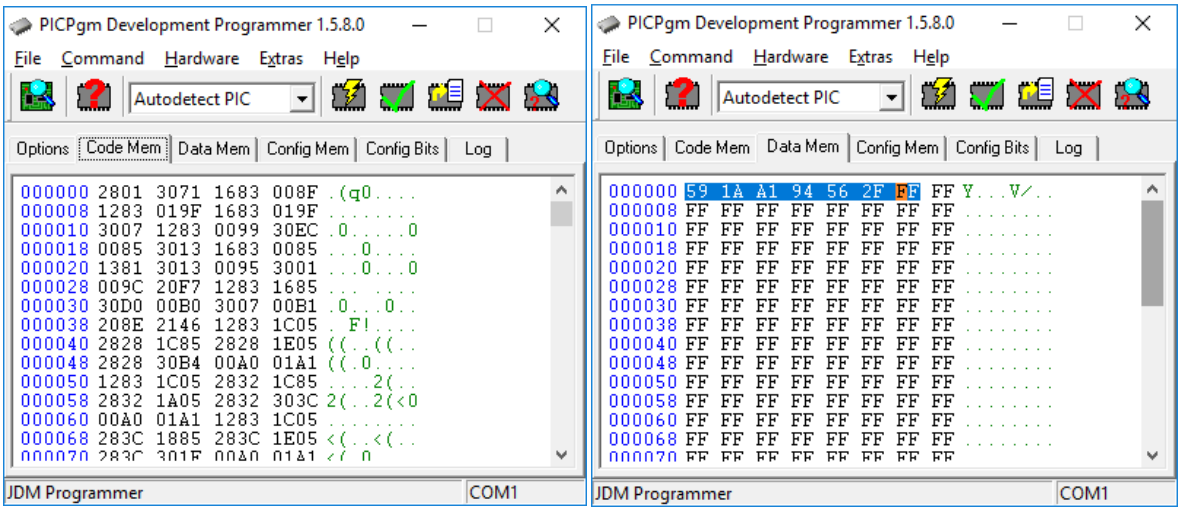

Figure 1 : Ecrans qui apparaissent lors de la programmation avec PICPgm
a – chargement du programme
b – entrée directe du nombre en hexadécimal représentant l'indicatif

Les dispositifs de programmation plus récents se connectent sur une prise USB. C'est le cas par exemple du PICkit3 de Microchip. Associé avec le logiciel PICkit3, on obtient l'écran présenté sur la Figure 2. Dans la fenêtre centrale, on voit apparaitre le code « genecall.hex » qui a été chargé préalablement. Le contenu de la mémoire est dans la fenêtre du bas. C'est là où il faut entrer l'indicatif en hexadécimal « 59 1A A1 94 56 2F FF ». Il ne reste plus qu'à lancer la programmation du PIC 12F683.

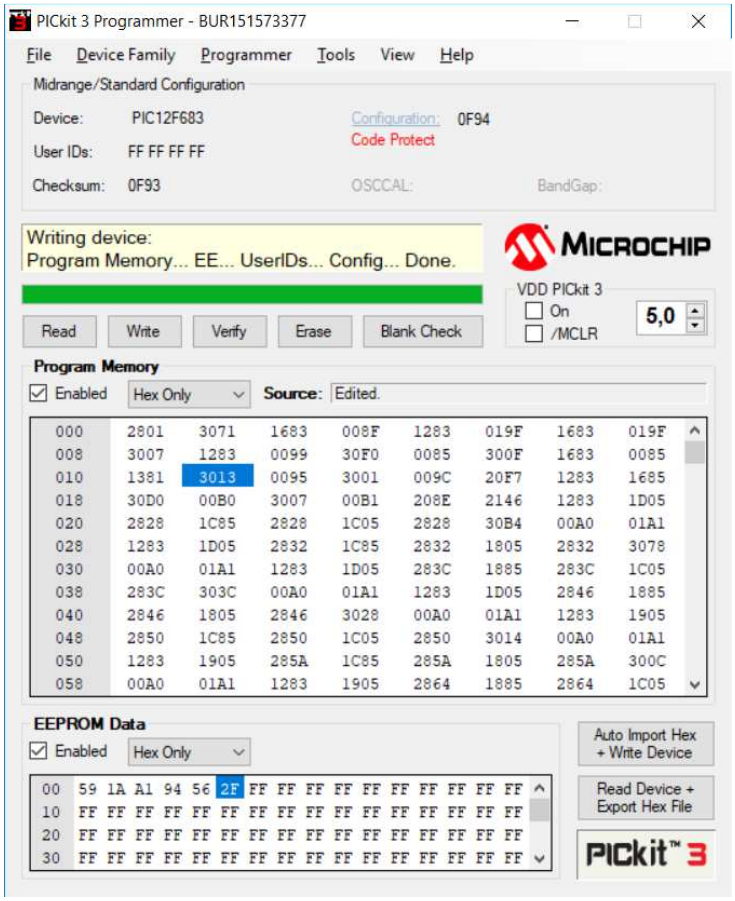

Figure 2 : Ecran qui apparait lors de la programmation avec PICkit3

Les 2 cas présentés ne sont que des exemples. Il existe une très grande variété de systèmes de programmation et nous ne pouvons pas les présenter tous.

Ce qu'il faut surtout retenir, c'est que pour pouvoir programmer le PIC 12F683 il faut récupérer le code « genecall.hex » qui est téléchargeable sur le site www.F1LVT.com. Ce code est général, il ne dépend pas de l'indicatif. Au moment de la programmation du PIC, il faut préparer le nombre en hexadécimal à entrer en mémoire correspondant à l'indicatif. Ce nombre est la succession de « 01 » pour les points, « 10 » pour les traits, et « 00 » pour les intervalles entre 2 caractères Morse. Ce nombre binaire est enregistré en hexadécimal dans la mémoire EEPROM du PIC. Il sera lu au moment du démarrage du PIC.

Construction de « GénéCall »

Le schéma autour du PIC 12F683 est très simple (Figure 3). Les 3 entrées E1 - E2 - E3 permettent de programmer la temporisation entre 2 envois de l'indicatif. En sortie le signal est atténué par R1 et P1 et filtré par C1. La commutation de l'émetteur est effectuée par le MOSFET T1, par mise à la masse. Le PIC peut être alimenté entre 2V et 5V. L'entrée « +3V » est une alimentation directe en basse tension. Avec une tension d'alimentation plus élevée, il faut utiliser l'entrée « +12V » qui passe par un régulateur 78L05.

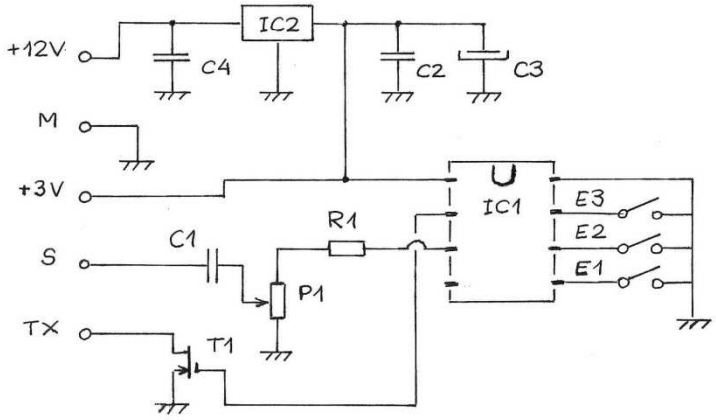

Figure 3 : Schéma du montage

Liste des composants

Résistances

R1	100 kΩ	
P1	100 kΩ	Potentiomètre

Condensateurs

C1, C2, C4	100 nF	
C3	4,7 µF	Electrochimique

Semiconducteurs

IC1	PIC 12F683 programmé
IC2	78L05
T1	2N7000 ou équivalent (BS170 par ex.)

Divers

E1, E2	Interrupteurs de CI

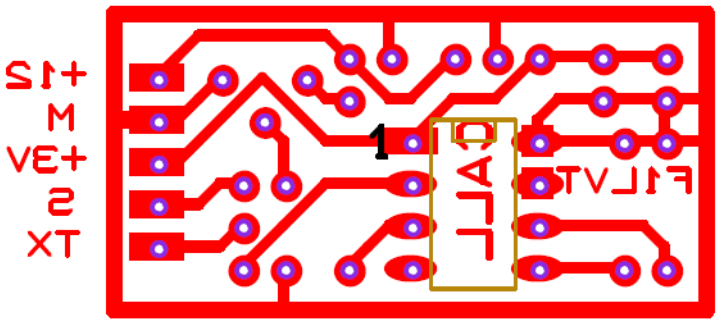

Figure 4 : Circuit imprimé

La figure 4 montre le circuit imprimé. Comme on trouve les interrupteurs de CI par 1, par 2 ou par 4 contacts, c'est un interrupteur double qui a été choisi pour les entrées E1 et E2. Quant à E3, on peut le relier ou non à la broche voisine qui est la masse.

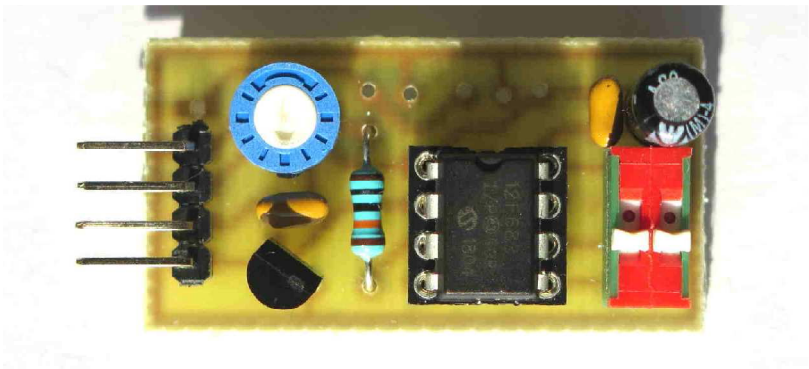

Photo 2 : Module « GénéCall » en version Alimentation 3V
pour un couplage direct avec un TX portable

Dans la réalisation montrée sur la Photo 2, l'entrée « +12 » et le régulateur n'ont pas été câblés car le montage va être alimenté en 3V par la prise micro d'un TX portable (Photo 3), par exemple un TX portable Kenwood ou Baofeng.

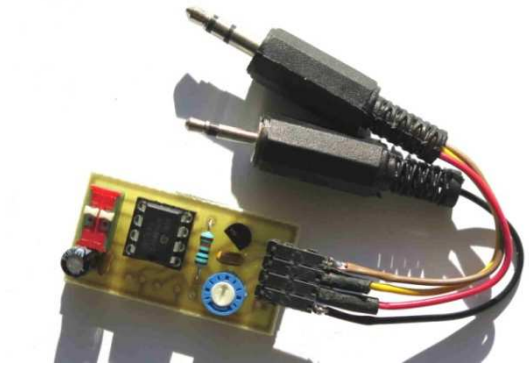

Photo 3 : Connexion de « GénéCall » à un TX portable par 2 fiches Jack

Pour une connexion dans la prise micro d'un TM-V7 utilisé en transpondeur, le « GénéCall » est alimenté en 8V par la prise micro ; il faut passer par l'entrée « +12 » et le régulateur. On envoie tout simplement le signal S sur l'entrée micro avec une temporisation d'une minute, et on entendra passer l'indicatif toutes les minutes quand le transpondeur sera en émission.

Utilisations du Générateur d'indicatif « GénéCall »

L'utilisation est très générale, partout où on a besoin d'identifier une station radioamateur ou d'envoyer un indicatif ou un appel en Morse.

Par exemple pour les transpondeurs VHF – UHF en fonctionnement, il faut transmettre régulièrement l'indicatif. Avec tous ceux qui conservent l'entrée du micro ouverte, comme le Kenwood TM-V7, il suffit d'envoyer la sortie BF (sortie « S » de « GénéCall ») sur l'entrée micro par la fiche RJ 45. Classiquement on envoie l'indicatif toutes les minutes en surimpression de la phonie. On peut même transmettre des informations par l'intervalle de temps entre 2 envois d'indicatif : par exemple 60 secondes en fonctionnement sur secteur et 6 secondes quand le transpondeur se retrouve sur batterie.

Avec le Kenwood TM-V71, l'entrée micro est coupée en fonctionnement en transpondeur. Une solution très simple consiste alors à transmettre l'indicatif par un TX extérieur, comme un Baofeng UV-5R par exemple. Le « GénéCall » peut être auto-alimenté en 3V par le Baofeng, qui fonctionne alors en balise émettant l'indicatif toutes les minutes. Le passage en émission du TX est contrôlé par la sortie « TX » du « GénéCall » et l'indicatif est envoyé par la sortie « S ».

Associé à un TX VHF ou UHF, le module « GénéCall » permet d'obtenir immédiatement une balise, qui transmet l'indicatif à intervalle régulier. Avec tous les TX portables qui ont une alimentation 3V ou 3,5V dans le micro, le montage GénéCall peut être autoalimenté par le TX. Ce sont les TX portables Kenwood, Baofeng, Kirisun, etc et certain Icom. Ce type de fonctionnement est plus largement décrit dans l'article qui présente le module « Génépious », qui un modulateur de signal de balise.

Ce type de balise peut servir de balise chasse au renard, avec par exemple l'émission de l'indicatif toutes les 4 secondes. On peut utiliser une temporisation plus longue pour rendre la recherche plus difficile.

« La Plume » est un autre exemple de balise [5]. Elle transmet en UHF des trames analogues à celle des balises 406. Quand on l'utilise sur la bande radioamateur 430 MHz, il faut faire passer l'indicatif en Morse de temps en temps. « GénéCall » permet alors de faire cette fonction par liaison directe, en envoyant l'indicatif en Morse sans perturber l'émission PSK.

Références

[1] J-P YONNET, « Générateur d'indicatif pour relais, transpondeurs et balises »
http://www.f1lvt.com/files/511-Generateur-dIndicatif.90.pdf

[2] B. Anding, AA5OY, « A PIC of an IDer »,
Revue QST, Jan 1998, p 36 - 38

[3] J. A. Hansen, W2FS, « Using PIC Microcontrollers in Amateur Radio Projects », Revue QST, Oct 1998, p 34 - 40

[4] http://www.df1zn.de/ID_Keyer/id_keyer.html

[5] J-P YONNET, "« La Plume », une petite balise 406 de test »
http://www.f1lvt.com/files/237-Article--La-Plume--V2.74.pdf