

# Générateur d'indicatif pour relais, transpondeurs et balises

Jean-Paul - F1LVT / ADRASEC 38  
F1LVT@yahoo.fr

Quand on émet sur les bandes radioamateurs, on a régulièrement besoin d'un générateur d'indicatif. La réglementation impose qu'une station radioamateur transmette son indicatif régulièrement. Une solution simple consiste à envoyer cet indicatif en Morse. C'est ce que fait notre montage, grâce à un PIC et quelques composants périphériques.

Le montage présenté est la reprise et l'adaptation de travaux et d'articles publiés en particulier dans la revue QST. Il existe de nombreux autres montages réalisant cette fonction de générateur d'indicatif. Dans les années 2000, beaucoup d'OM se sont entraînés à la programmation des PIC en réalisant ce type de dispositif.

Le montage réalisé présente l'avantage d'être très simple, de fonctionner sans quartz extérieur et d'utiliser un PIC disponible et pas cher (entre 1,50 € et 2 €). Ce montage ne prétend pas avoir une grande originalité par rapport aux montages existants et publiés, c'est avant tout un montage utilitaire pour répondre à la réglementation.

## 1- Rappel de la réglementation radioamateur

Dans le dernier décret paru (décision 2012-1241 du 2 octobre 2012, publié au JO du 7 mars 2013) [1], l'article 4 précise que :

*« Au cours de leurs émissions, les stations d'amateur doivent transmettre leur indicatif d'appel à de courts intervalles, et au moins :*

- au début et la fin de toute émission ;*
- toutes les quinze minutes au cours de toute émission d'une durée supérieure à 15 minutes sur une même fréquence ;*
- en cas de changement de fréquence d'émission, au début de toute période d'émission sur la nouvelle fréquence. »*

En conséquence tout transpondeur ou relais ADRASEC, ou toute balise qui émet sur une fréquence ADRASEC doit transmettre son indicatif, comme la balise « La Plume » qui émet sur 431,990 MHz. Après avoir montré le fonctionnement du dispositif et la programmation du PIC, nous verrons comment coupler ce module à notre balise.

## 2- Les transpondeurs ADRASEC

Beaucoup d'ADRASEC utilisent des transpondeurs VHF – UHF pour leur transmission. Ces transpondeurs peuvent facilement être construits avec des émetteurs récepteurs bi-bandes comme le Kenwood TM-V7. En fonctionnement, le problème de la transmission régulière de l'indicatif se pose.

Kenwood a essayé de trouver une solution sur la dernière génération de ses transpondeurs. À l'intérieur de la logique de l'émetteur récepteur TM-V71, un générateur d'indicatif en Morse a été intégré. Toutes les cinq minutes, l'indicatif en Morse est transmis alternativement sur la bande VHF et sur la bande UHF. Même si le transpondeur n'est pas utilisé, il transmet son indicatif si il est sous tension. Si vous faites fonctionner le TM-V71 en transpondeur pendant une longue période, vous vous apercevez vite que vous devez choisir entre couper l'écoute du transpondeur ou couper le transpondeur lui-même ...

Bien avant que Kenwood propose cette solution, nous avons travaillé sur ce sujet en particulier pour le transpondeur du Moucherotte, qui est en fonctionnement continu en point haut depuis 1999, à 1900 m au dessus de Grenoble. Nous nous étions aperçus que l'entrée microphone des transpondeurs Kenwood restait active lorsque le transpondeur était en fonctionnement. C'est-à-dire que si le TM-V7 est mis en position transpondeur avec le micro branché, il retransmet à la fois le son reçu en VHF et le son reçu par le microphone. En conséquence, pour que le transpondeur retransmette son indicatif, il suffit de lui envoyer cet indicatif par l'entrée microphone. En plus la prise microphone a une alimentation 8V qui permet d'alimenter directement le module générateur d'indicatif et le rend complètement autonome.

Pour cette fonction, nous avons réalisé un générateur d'indicatif qui produit en Morse l'indicatif du transpondeur. A l'époque, nous avons utilisé un module disponible chez Elektor [2]. Quand le transpondeur est en fonctionnement, l'indicatif est envoyé toutes les minutes en Morse, en surimpression avec la modulation transmise. Avec une porteuse, on entend clairement l'indicatif ; en phonie, le passage de l'indicatif ne gêne pas la transmission.

Nous sommes même allés encore plus loin en utilisant le délai entre deux passages d'indicatif pour transmettre l'information du fonctionnement sur batteries. En fonctionnement normal l'indicatif est envoyé toutes les minutes ; en fonctionnement sur batteries l'indicatif est envoyé toutes les six secondes, ce qui prévient les utilisateurs et qui les incite à être très court sur le transpondeur.

Comme Elektor ne fournit que des PIC programmés, il faut recommander un PIC (si il existe encore) en cas de panne ou de nouvelle installation. C'est pour cela que nous nous sommes intéressés à ces générateurs d'indicatif pour pouvoir les construire nous-mêmes et les monter sur les transpondeurs. Le montage présenté peut être utilisé avec un TM-V7 ou un TM-V71 par exemple, en étant auto-alimenté par le TX et transmettant l'indicatif en surimpression sur l'émission.

### 3- Les balises d'exercice en bande radioamateur

Un autre besoin est apparu récemment, c'est pour des balises d'exercice transmettant sur la bande radioamateur. Par exemple « La Plume » émet sur ce 431,990 MHz en reproduisant exactement la modulation des balises Cospas Sarsat. Pour respecter la réglementation, elle doit transmettre régulièrement un indicatif radioamateur.

Avec le montage que nous allons décrire, « La Plume » transmet un indicatif radioamateur en Morse. Pour cette balise, c'est une version simplifiée du générateur d'indicatifs qui a été utilisée ; autour du PIC et il ne reste plus que trois composants. Le passage de l'indicatif ne gêne absolument pas le modulateur de phase, l'émission de la trame Cospas Sarsat bloque temporairement (pendant 500 ms) la modulation de l'indicatif.

### 4- Construction d'un générateur d'indicatif

C'est relativement simple à faire à partir des travaux déjà publiés. En 1998, dans la revue américaine QST, deux articles sont parus sur ce type de réalisation. Le premier de AA5OY intitulé « A PIC of an IDer » [3], le second quelques mois plus tard de W2FS « Using PIC Microcontrollers in Amateur Radio Projects » [4].

Initialement, le programme de W2FS a été développé pour un 16F84. Il ensuite été transposé par DF1NF sur un 16F627 [5]. Le 16F84 est devenu difficile à trouver de nos jours. Il y a été remplacé par de nouvelles générations de PIC plus performantes. Par exemple, on peut utiliser l'horloge interne du PIC, ce qui supprime le quartz extérieur. Pour notre part, nous avons adapté ce logiciel au PIC 16F628 car c'est un PIC plus facile à trouver que le 16F627 et il a deux fois plus de mémoire.

Nous allons décrire la réalisation en deux parties, d'abord la partie logicielle, puis la partie carte électronique.

#### 4.1- La partie logicielle

Pour le soft à entrer dans le PIC, il faut suivre les étapes suivantes.

##### **Phase 1**

Le programme en annexe est en assembleur. Même sans avoir de bases en assembleur, il est assez facile de suivre sa logique et de l'adapter. Les commentaires du programme sont en anglais (y compris les parties modifiées), mais ils sont faciles à comprendre. Pour le récupérer, il faut faire un copier-coller de la partie limitée par les lignes de tirets, et l'enregistrer avec une extension de type « .asm ». Par exemple on peut l'appeler « morse.asm »

Il faut ensuite modifier ce programme. Pour cela, il faut :

- entrer l'indicatif à émettre. Le programme utilise F1LVT comme exemple, mais il est facile de modifier le code (l'indicatif se trouve dans la seconde partie du programme) pour transmettre n'importe quel indicatif, voire même des commentaires du type « balise adrasedec ». La taille de la mémoire du PIC est très largement suffisante.
- à la fin du programme, il faut définir l'intervalle entre deux transmissions du message. La valeur 'd10' correspond à une transmission toutes les 40s. Par exemple cette valeur peut être multipliée par 5, c'est-à-dire 'd50' pour obtenir un intervalle de 3 minutes (200 secondes).
- en cas de besoin, la vitesse de transmission peut être modifiée par les paramètres en début du programme, 'd99' et 'd33'. Augmenter ces 2 valeurs (en respectant le facteur 3) permet de réduire la vitesse de transmission du Morse, et inversement.

##### **Phase 2**

Il faut ensuite transformer ce programme d'assembleur en exécutable, c'est-à-dire de « morse.asm » en « morse.hex ». Pour cela il faut utiliser le logiciel MPLAB de Microchip. C'est un logiciel gratuit.

Dans MPLAB, il faut créer un projet et le compiler. Si il n'y a pas d'erreur dans le programme, MPLAB génère le fichier « morse.hex ».

##### **Phase 3**

Il faut enfin entrer ce programme « morse.hex » dans le 16F628. Pour cela il faut utiliser un dispositif de programmation (un JDM par exemple) et un logiciel de pilotage de ce dispositif comme « PICPgm » (ou un équivalent comme WinPIC800). Quand le programme est entré dans le PIC, il reste plus qu'à vérifier le fonctionnement sur la carte électronique.

Toute cette procédure peut paraître compliquée pour quelqu'un qui n'a jamais programmé de PIC et qui n'est pas équipé, mais elle est relativement simple et naturelle pour tous ceux qui ont déjà travaillé avec des microcontrôleurs PIC de Microchip.

#### 4.2- La partie carte électronique

Le montage est relativement simple. Sur la Photo 1, le connecteur d'entrée-sortie est à gauche et le PIC 16F628 à droite. Sur le connecteur, on a successivement de haut en bas :

- l'alimentation 12 V. L'entrée accepte toute tension entre 8V et 30V.
- l'alimentation directe 5V. Si le système est alimenté en 12V, ce point peut servir de sortie 5V.
- la masse de l'alimentation.

- la sortie BF directe du PIC (SA), après une résistance série. Cette sortie est très utile pour la modulation AM. C'est celle que nous allons utiliser pour « La Plume ».
- la sortie BF après atténuation par résistance et potentiomètre (SB). C'est la sortie la mieux adaptée pour l'entrée microphone de l'émetteur.
- la masse BF.
- la masse de la commutation TX. Les 3 masses sur le connecteur sont reliées.
- la sortie TX, qui permet le passage en émission de l'émetteur par mise à la masse de cette broche au travers d'un transistor.



Photo 1 : La carte complète

Le schéma est présenté sur le Figure 1 et le circuit imprimé sur la Figure 2. Seulement 5 pattes (sur 18) du PIC 16F628 sont utilisées. Les pattes 4, 5 et 14 sont respectivement le Reset, la masse et le +5V du PIC. La sortie active du PIC est en 7 ; le signal est émis sous la forme de créneaux de 5V d'amplitude. Pour utiliser ce signal sur une entrée microphone, il faut l'atténuer, filtrer les hautes fréquences et enlever la composante continue. C'est effectué par R2, C4, et P1. La patte 8 du PIC permet de commuter l'émission réception. En réception, cette patte est au niveau bas. Lors de l'émission de l'indicatif, cette patte passe au niveau haut (5V) et met l'entrée TX à la masse.

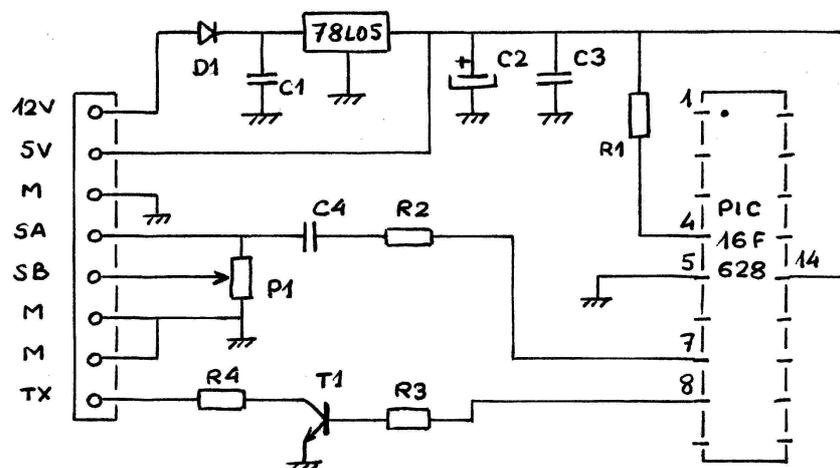


Figure 1 : Schéma du générateur d'indicatif

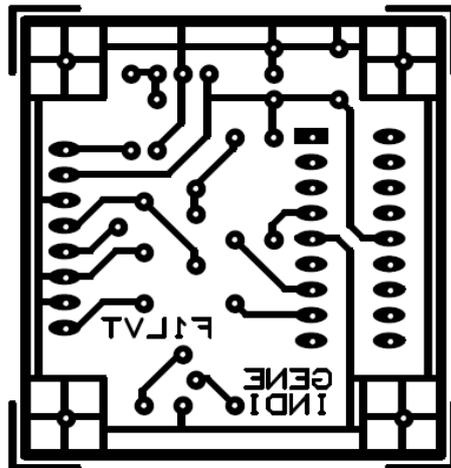


Figure 2 : Le circuit imprimé

#### 4.3- Liste des composants

Composants actifs		
IC1	16F628	Microcontrôleur PIC Microchip
IC2	5V	78L05
D1	1N4148	Diode
T1	BC550	Transistor NPN
Condensateurs		
C1, C3, C4	470 nF	
C2	47 µF	électrochimique
Résistances		
R1, R2, R3	10 kohms	
R4	100 ohms	
P1	1kohms	Potentiomètre

#### 4.4- Mise en route et utilisation

Il faut d'abord commencer par programmer correctement le PIC avec l'indicatif et la temporisation souhaitée.

Dès la mise sous tension, le PIC sur la carte fonctionne automatiquement : il transmet régulièrement l'indicatif. Le seul réglage est le potentiomètre P1, qui permet d'ajuster le niveau BF.

#### 5- Comment envoyer l'indicatif avec « La Plume »

En pratique, c'est relativement simple car « La Plume » est piloté par un module UHF prévu pour une modulation ASK ou OOK (pour On-Off Keying). La puissance suit la valeur du signal modulant (Mod = 0 → P = 0, Mod = 3,6 V → P = 10 dBm ). En pilotant la modulation directement par la sortie SA du PIC, l'indicatif est transmis en CW par le module UHF.

Il aurait aussi été possible d'envoyer le signal SB sur une diode varicap pour moduler la fréquence du quartz du module UHF. On aurait fait un émetteur en modulation de fréquence. C'est plus compliqué et plus difficile à régler que la modulation OOK.

Le générateur d'indicatif est alimenté en 5V par la carte de « La Plume ». Il n'y a que 3 fils : le +5V, la masse et le signal de sortie généré par le PIC (Figure 3). Autour du PIC, il ne reste que la résistance de 10k pour le reset et le condensateur de découplage. Une diode 1N 4148 placée entre la sortie du PIC (pin 7) et le connecteur permet d'abaisser la tension au niveau requis. La pin 8 (contrôle TX) n'est pas utilisée.

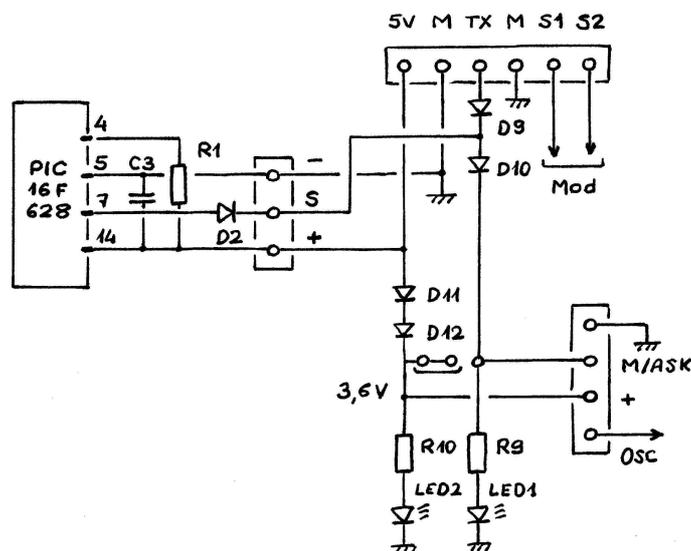


Figure 3 : Connexions à la Plume

Pour la réalisation, nous avons utilisé un morceau du circuit imprimé précédent (Photo 2). Le connecteur à 3 broches permet de faire la liaison avec « La Plume ». Le signal sort au milieu du connecteur. La diode série D2 (1N4148, voir Figure 3) est située sous la carte.



Photo 2 : La carte additionnelle pour transmettre l'indicatif avec « La Plume »

La modulation du module UHF est pilotée simultanément par le PIC du générateur de trames (16F88) et par le PIC du générateur d'indicatif (16F628). Quand on envoie une trame COSPAS SARSAT, la commande TX à 5V maintient le module en émission permanente pendant 500 ms. Même si le générateur d'indicatif envoie son signal en même temps, l'émission reste permanente et la trame PSK est émise correctement. Le système rend la trame COSPAS SARSAT prioritaire sur l'indicatif.

## Synthèse

Globalement c'est un montage simple, très utile et très facile à construire. Le logiciel est en annexe de cet article, et son adaptation est assez aisée.

Seulement une petite partie du PIC est utilisée en terme d'entrées-sorties et de mémoire interne, et il serait certainement possible d'utiliser un PIC plus petit. Cependant, comme le 16F628 est un grand classique, on le trouve assez facilement et son prix est très modéré (les PIC exotiques sont souvent plus onéreux).

## Références

[1] ARCEP, Autorité de Régulation des Communications Electroniques et des Postes, Décision n°2012-1241 du 2 octobre 2012, publié au Journal Officiel de la République Française le 7 mars 2013, texte 72 sur 111

[2] G. Baars, « Générateur CW, auxiliaire pratique du radioamateur », revue Elektor, décembre 1998, p 24 - 28

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

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

[5] [http://www.df1zn.de/ID\\_Keyer/id\\_keyer.html](http://www.df1zn.de/ID_Keyer/id_keyer.html)

## Annexe : Programme en assembleur

Pour obtenir le programme « morse.asm », il faut récupérer le texte entre les 2 lignes de pointillés et l'enregistrer avec l'extension « .asm ».

---

; Morse Code Generator - Générateur d'indicatif en Morse -

; Origin : QST article Oct 1998 by W2FS, for PIC 16F84

; Modified by DF1NF in 2006 for PIC 16F627

; Slightly modified by F1LVT in 2013 for PIC 16F628

; V-G4 - Message 'F1LVT' - T=40s

```

                list          p=16F628
                radix         hex
                ___config     0x3FF1      ;      Internal RC oscillator

dahlen          equ          d'99'      ;      <== Controls code speed (dah length)
ditlen         equ          d'33'      ;      <== Controls code speed (dit length)
                org          0x000      ;      Increase (though not over 255) for slower code
                MOVLW        00          ;      Decrease for faster code
                MOVWF        0A          ;      Make sure the top number is 3 times the bottom
                GOTO         start      ;
                NOP
                NOP
time           MOVF          30,W
                BTFSC        03,2
                GOTO         endtime
uptop          MOVLW        01
                MOVWF        2D
upagn          CLRf         2C
doagn          DECFSZ       2C,F
                GOTO         doagn
                DECFSZ       2D,F
                GOTO         upagn
                MOVLW        4A
                MOVWF        2C
upone          DECFSZ       2C,F
                GOTO         upone
                DECFSZ       30,F
                GOTO         uptop
endtime        RETLW        00

dah            MOVLW        01          ;      subroutine to do dah
                MOVWF        2E
agn3dah        MOVLW        dahlen
                SUBWF        2E,W
                BTFSC        03,0
                GOTO         enddah
                BSF          06,1      ;      turn on pin B1
                MOVLW        01
                MOVWF        30
                CALL         time      ;      wait 1 millisecond
                BCF          06,1      ;      turn off pin B1
                MOVLW        01
                MOVWF        30
                CALL         time      ;      wait 1 millisecond
                MOVF        2E,W
                INCF        2E,F
                GOTO         agn3dah   ;      loop up to do it again.
enddah         MOVLW        3C          ;      add a small delay
                MOVWF        30
                CALL         time      ;      wait 1 millisecond
                RETLW        00
```

```

dit      MOVLW      01      ;      subroutine to do a dit
        MOVWF      2E
agn3dit  MOVLW      ditlen
        SUBWF      2E,W
        BTFSC      03,0
        GOTO      enddit
        BSF        06,1      ;      turn on Pin B1
        MOVLW      01
        MOVWF      30
        CALL       time      ;      wait 1 millisecond
        BCF        06,1      ;      turn off Pin B1
        MOVLW      01
        MOVWF      30
        CALL       time      ;      wait 1 millisecond
        MOVF       2E,W
        INCF       2E,F
        GOTO      agn3dit    ;      loop up and do it again
        MOVLW      3C      ;      add a small delay
        MOVWF      30
        CALL       time      ;      wait 1 millisecond
        RETLW     00

lspace   MOVLW      0xB4    ;      subroutine to make a
        MOVWF      30      ;      letter space
        CALL       time
        RETLW     00

start    CLRF       04      ; ***** Main program starts here *****
        MOVLW     0xF9    ; Define Port B, Bit 1 and Bit 2 as output
        TRIS      6

top      BSF        06,2    ;      turn on pin B2 (PTT on, wait 3 spaces)

        CALL      lspace
        CALL      lspace
        CALL      lspace

        CALL      dit      ;      transmit F1LVT
        CALL      dit      ;      'F'..-
        CALL      dah
        CALL      dit

        CALL      lspace

        CALL      dit      ;      '1',----
        CALL      dah
        CALL      dah
        CALL      dah
        CALL      dah

        CALL      lspace

        CALL      dit      ;      'L'..-
        CALL      dah
        CALL      dit
        CALL      dit

        CALL      lspace

        CALL      dit      ;      'V'...-
        CALL      dit
        CALL      dit
        CALL      dah

        CALL      lspace

        CALL      dah      ;      'T'-

```

```

CALL      lspace
CALL      lspace
CALL      lspace

BCF       06,2      ;      turn off pin B2 (PTT off)

MOVLW    01
MOVWF    2E

loop      MOVLW     d'10'      ;      this code programs the delay between IDs.
;      <= 4 times this number of seconds between IDs
;      Initial value = 50 for 3 min (200 sec)

SUBWF    2E,W
BTFSC   03,0
GOTO     bottom
MOVLW    10
MOVWF    2F
again    MOVLW     0xFA
MOVWF    30
CALL     time
DECFSZ   2F,F
GOTO     again
MOVF     2E,W
INCF    2E,F
GOTO     loop
bottom   GOTO     top

END

```

---