Générateur d'un message CW pour balise

Jean-Paul Gendner F5BU f5bu@orange.fr

Le montage décrit génère périodiquement un message CW d'un texte, pré-enregistré dans le microcontrôleur, suivi de la température, typiquement pour une balise radio. Il est toutefois facile de le modifier pour générer différents messages pré-enregistrés à la demande, ou d'autres fonctions.

Trois signaux de sorties sont disponibles : un « drain ouvert » pour court-circuiter le signal PTT à la masse, un signal tout ou rien pour commander la CW par du « shift keying » et un signal analogique 700 Hz, proche d'une sinusoïde, pour envoyer le son de la CW vers une entrée micro.

Le diagramme du fonctionnement de base est donné par la figure 1.

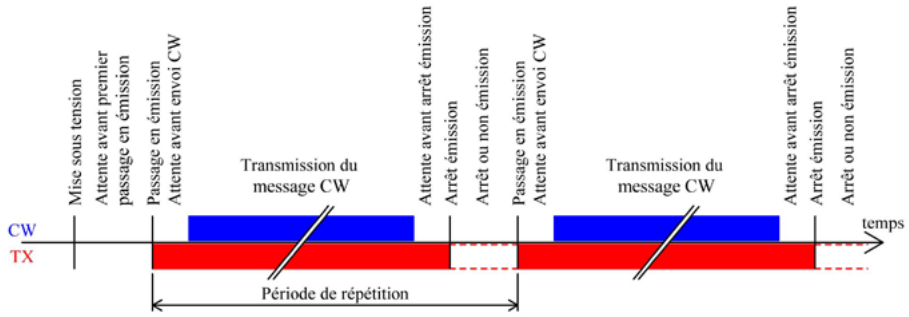

Figure 1. Diagramme montrant le fonctionnement de base

Comment générer un signal proche d'une sinusoïde

Pour obtenir un signal proche d'un signal sinusoïdal, une astuce trouvée sur Internet est utilisée, mais je ne sais plus qui en est l'auteur car sa découverte remonte à plusieurs années.

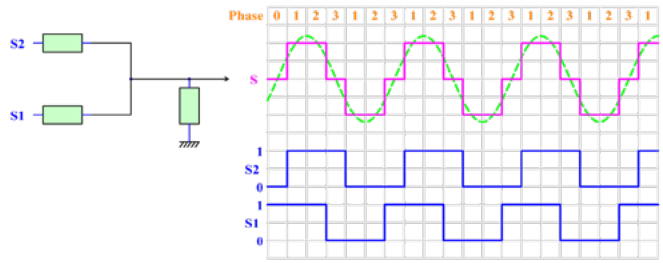

Figure 2. Principe utilisé pour approcher une sinusoïde

Deux signaux rectangulaires décalés de T/6 sont générés par le microcontrôleur sur deux sorties, S1 et S2, pour permettre, par simple sommation, d'obtenir, après un filtre passe bande, un signal proche d'une sinusoïde.

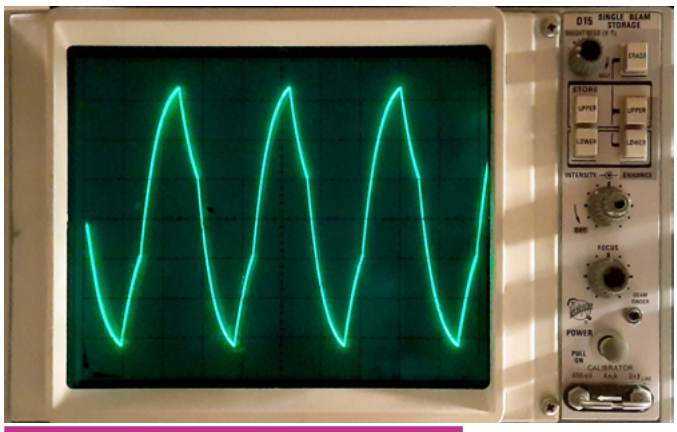

Figure 3. La forme d'onde obtenue

Au niveau programmation, durant la génération de CW, un timer provoque une interruption tous les T/6 et 3 phases, gérées cyclique-ment, se succèdent. Trois phases seulement et non 6, car la généra-tion des deux demi-périodes est identique. Une variable Phase sert d'indicateur de phase :

- Phase 0 : état de repos : attente de génération de CW : S1=1, S2=0 pour que le condensateur du filtrage passe haut se charge à la valeur moyenne de la tension de sortie de S1 et S2.
- Phase 1 : inversion du signal S2 en début de phase et initialisation de la phase 2.
- Phase 2 : initialisation de la phase 3.
- Phase 3 : inversion du signal S1 en début de phase et initialisation de la phase 1.

Une image valant souvent mieux qu'un long discours, la figure 2 montre cela. La figure 3 montre la forme du signal obtenu à la sortie Son.

Le schéma

Le schéma du circuit est celui de la figure 4.
Le cœur du montage est un module Arduino Nano mettant en œuvre un microcontrôleur ATmega328P cadencé à 16 MHz et programmable avec l'IDE (Integrated Development Environment) Arduino (voir lien 1). De très nombreuses documentations de tous niveaux sont accessibles sur Internet et notamment sur le lien 2.

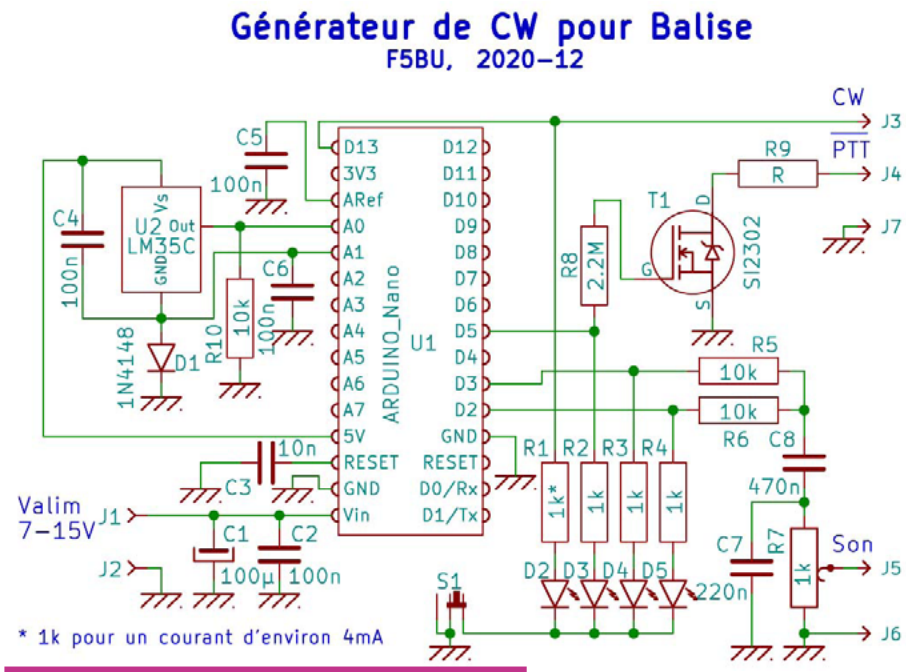

Figure 4. Schéma du générateur de CW

Du coup, le montage n'est vraiment pas compliqué. Quatre diodes électroluminescentes permettent de visualiser l'état des différents signaux : PTT, CW, S1 et S2, et un cavalier permet de couper les quatre DEL en même temps si souhaité. Les valeurs des résistances R1 à R4 sont à choisir en fonction des DEL utilisées.

Les résistances R5, R6 et R7 constituent à la fois l'additionneur des signaux S1 et S2 et un diviseur de tension. C8 en fait un filtre passe haut qui élimine la composante continue, et C7 en fait un filtre passe bas pour approcher un signal sinusoïdal. Si besoin, et si le niveau de sortie est suffisant, un deuxième filtre passe bas peut être mis en cascade pour s'approcher encore davantage d'un signal sinusoïdal.

Les résistances R8, R9 et le transistor MosFet T1 constituent « l'interrupteur » pour la mise à la masse du PTT. N'importe quel MOSFET canal N fait l'affaire, et à condition de modifier la valeur de R8, un transistor bipolaire NPN peut aussi être utilisé.

Le LM35 permet de mesurer la température. Il s'agit d'un capteur de température facile à utiliser car son signal de sortie est directement de 10 mV/°C sans nécessiter de calibration. Pour pouvoir mesurer des températures négatives, il faut utiliser un LM35C ou LM35CA et l'alimenter de manière à ce que le signal de sortie puisse être négatif par rapport à sa connexion GND. Afin de ne pas avoir à mettre un régulateur de tension, une simple diode D1 est placée en série avec le capteur, comme préconisé dans la documentation (avec deux diodes), et la résistance de tirage R10 permet au signal de sortie de devenir négatif.

La tension aux bornes de cette diode n'étant pas constante, notamment lors de variations de température, la mesure du signal de sortie de température est effectuée en différentiel, c'est à dire en mesurant la tension de sortie du capteur sur A0, puis en soustrayant celle aux bornes de la diode, mesurée sur A1. Considérant que la température la plus basse à mesurer est de -25°, la tension de sortie étant dans ce cas de -250 mV, une seule diode suffit. Ceci permet de prendre comme référence de tension pour le convertisseur analogique numérique, la référence interne de 1,1 V de l'Arduino. Cette dynamique de 1,1 V autorise une mesure de température positive de (1,1-0,6)/0,01=50° avant saturation du CAN du µC, ce qui est considéré comme suffisant. Le CAN étant un convertisseur 10 bits, la résolution est de 1,07 mV soit environ 0,1 °. Le LM35CA est un peu plus exact que le LM35C. Si besoin, le capteur peut être déporté à l'extérieur du montage.

Le programme

Côté programme, c'est le codage de l'alphabet selon Hans Summers G0UPL et Stephen Farthing G0XAR qui est utilisé. Chaque caractère est codé sur 8 bits soit 1 octet : en partant de la gauche, il faut rechercher le premier bit à 0, et à partir de là, un 0 correspond à un point et un 1 à un trait. Par exemple le P est codé 11100110. On ne tient pas compte de 1110 puisqu'on attend le premier bit à 0, et le code est ensuite 0110, soit donc point, trait, trait, point. C'est très simple et efficace, il fallait juste y penser.

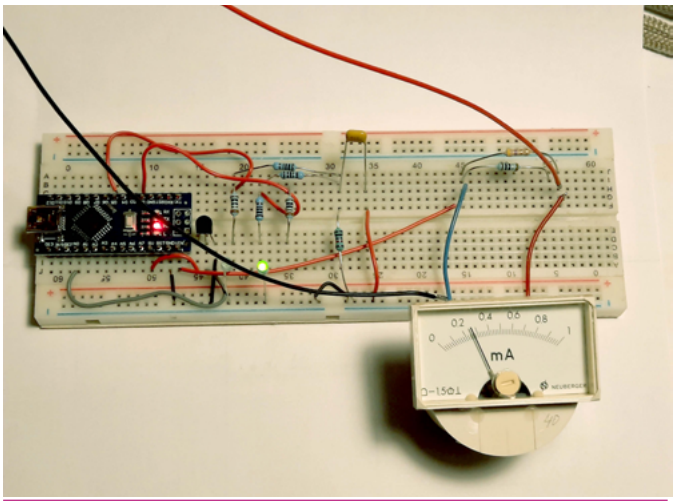

Figure 5. Le prototype (la PE du galvanomètre est de 40 mA)

Pour les séquencements, deux interruptions sont utilisées :

1) Le timer 2 est utilisé pour générer le 700 Hz de la tonalité de la CW. Comme ce son est généré en 6 phases, il faut générer des interruptions à 6 x 700 = 4200 Hz, ce qui correspond à une période de 238,1 µs. L'horloge interne de 16 MHz de l'Arduino est d'abord divisée par un « prescaler » de 32, ce qui donne une fréquence de 500 kHz, ou une période de 2 µs. Il faut donc 119 fois cette période pour avoir des interruptions toutes les 238 µs, car cette valeur doit être entière. Cela fait que la fréquence effective obtenue est de 700,28 Hz.

2) Le Watch Dog timer est programmé pour se déclencher toute les secondes, ce qui permet de régler facilement la durée entre deux débuts de messages juste en décomptant un nombre de secondes.

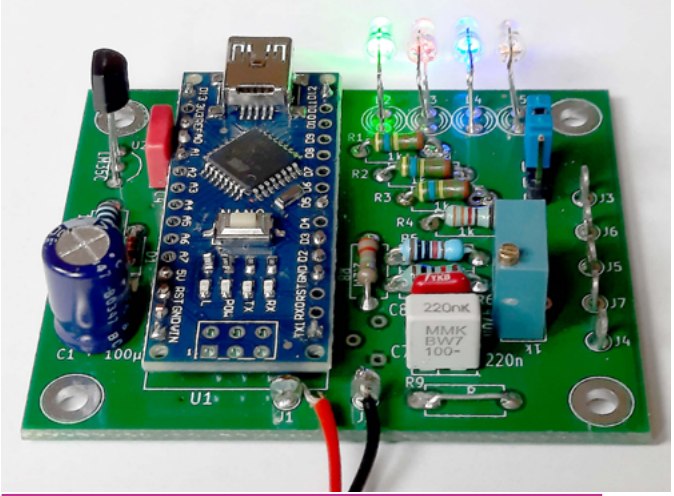

Figure 6. Le montage avec le circuit imprimé définitif

Sont très facilement modifiables dans le programme :

- la durée du point en ms (100, ce qui correspond à environ 11,5 mots/min),
- la période de répétition en s (60, doit être supérieure à la durée d'émission du message),
- le message,
- la fréquence du signal audio en Hz (dans l'équation int(256-500000/(6*700)-.5), remplacer 700 par une autre valeur comprise, en pratique, entre 330 et 1600 sans modifier la valeur du « prescaler »),
- le temps d'attente entre la mise sous tension et le premier passage en émission en ms (9000),
- le temps d'attente entre le passage en émission et l'envoi du premier caractère en ms (2000),
- le temps d'attente entre l'envoi du dernier caractère et le passage en réception en ms (2000),
- l'envoi ou non de la température,
- le maintien du TX en émission permanente.

Figure 7 : Le panneau avant réalisé avec mon programme Galva (voir lien 3)

Le fichier du code source complet, CW Balise.ino de 13 ko, peut être obtenu sur simple demande par courriel à l'auteur.

Le montage

Le montage a d'abord été réalisé sur une platine d'essai (Fig. 5), puis, n'envisageant pas de faire un circuit imprimé, sur un morceau de plaquette à trous avec plan de masse sur le dessus.
Mais, pour finir, j'ai tout de même réalisé un circuit imprimé (Fig. 6) et il me reste 4 circuits si cela peut intéresser. Sur le module Arduino, les résistances pour les DEL POW et L ont été dessoudées.
Le montage peut être alimenté soit par l'entrée 7-15 V soit par la Mini-B USB. Les figures 7 et 8 montrent le panneau avant et le montage en place dans le châssis de la balise.

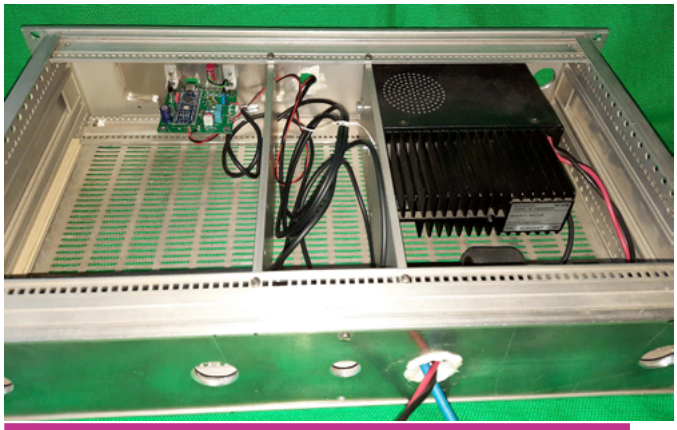

Figure 8. Le montage dans le châssis avec le TX

Consommation

Les consommations mesurées sont :

- DEL en marche avec des résistances de 1 kΩ :
  - entre 18 et 21 mA durant le transmissions de CW ;
  - environ 5 mA entre les transmissions.
- DEL coupées :
  - environ 12,5 mA durant les transmissions de CW ;
  - environ 2 mA entre les transmissions.

Remerciements

Un grand merci à Olivier F4HTB, qui m'a aidé et soutenu pour mettre le pied à l'étrier pour la programmation des Arduino.

Liens

1) http://urls.r-e-f.org/so869ro
2) https://www.arduino-france.com/ (Review, Arduino Nano : Avantages, inconvénients, utilisation et fonctionnement)
3) Décrit et disponible sur mon site : http://f5bu.fr