l'échantillonnage sub-Nyquist en pratique

capture fiable des hautes fréquences à l'aide du sous-échantillonnage

Sebastian Westerhold (AI5GW) (Allemagne)

Quiconque s'occupe de systèmes d'échantillonnage connaît rapidement les limites imposées par le théorème d'échantillonnage de Nyquist-Shannon. Selon cette règle, les signaux dont la fréquence est supérieure à la moitié de la fréquence d'échantillonnage ne peuvent plus être détectés de manière fiable. Mais ce n'est que la moitié de la vérité. Le sous-échantillonnage, également appelé échantillonnage sub-Nyquist, est une méthode fiable pour capturer les signaux de fréquence plus élevée. Cet article présente cette méthode de manière pratique, en utilisant un simple Arduino Uno.

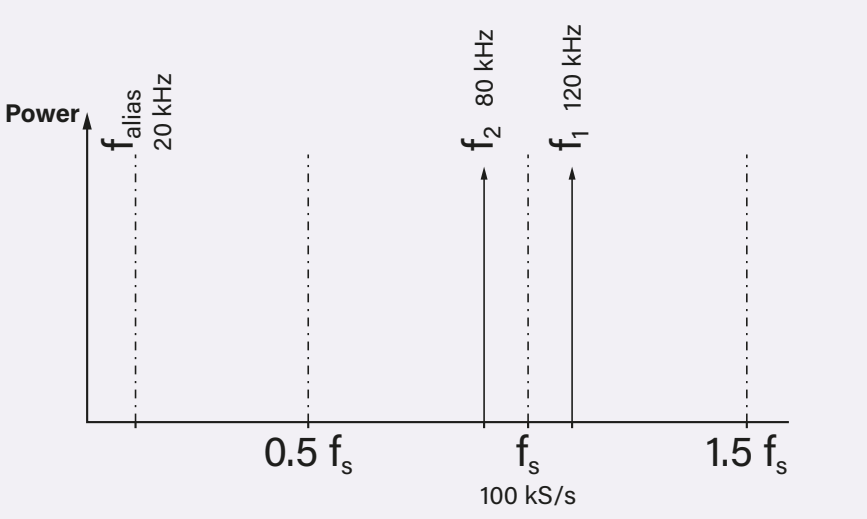

Figure 1. À une fréquence d'échantillonnage de 100 kS/s, un signal de 120 kHz (f_1) et un signal de 80 kHz (f_2) produisent un effet de repliement (f_{ALIAS}) de 20 kHz.

Le problème

Le théorème d'échantillonnage de Nyquist-Shannon est très important lorsqu'il s'agit de systèmes d'échantillonnage. Sa version très simplifiée stipule que la fréquence d'échantillonnage doit être au moins le double de la fréquence la plus élevée apparaissant dans le signal à échantillonner. Vous avez certainement déjà entendu ou lu quelque chose de ce genre. Que se passe-t-il si vous échantillonnez un signal sinusoïdal pur de 120 kHz avec une fréquence d'échantillonnage de seulement 100 kS/s (kHz) ? Il en résultera ce que l'on appelle un repliement à 20 kHz. Les échantillons acquis ressembleraient donc à un signal sinusoïdal d'une fréquence de 20 kHz (120 kHz - 100 kS/s) qui aurait été échantillonné. Et les erreurs de repliement doivent être évitées par tous les moyens. Vraiment ? La question de savoir si les erreurs de repliement constituent un facteur d'interférence ou peuvent être utilisées comme une technique utile dépend de l'application spécifique. Dans l'exemple de la fréquence d'échantillonnage de 100 kS/s, il serait impossible de distinguer un signal de 20 kHz d'un signal de 120 kHz ; les deux signaux ressembleraient à un signal de 20 kHz. Incidemment, un signal de 80 kHz produirait également un repliement de 20 kHz (100 kS/s - 80 kHz) (**figure 1**). Toutefois, cette circonstance n'est critique que si une distinction entre les fréquences du signal mentionnées est nécessaire. Si toutes les composantes de fréquence effectivement présentes dans le signal se situent dans une plage clairement identifiable, l'échantillonnage peut être significatif si les effets de repliement sont intentionnellement exploités. L'unicité est assurée si toutes les composantes de fréquence apparaissant dans le signal à échantillonner se situent dans la même zone de Nyquist.
La plage de fréquences allant de 0 Hz (CC) à la moitié de la fréquence d'échantillonnage

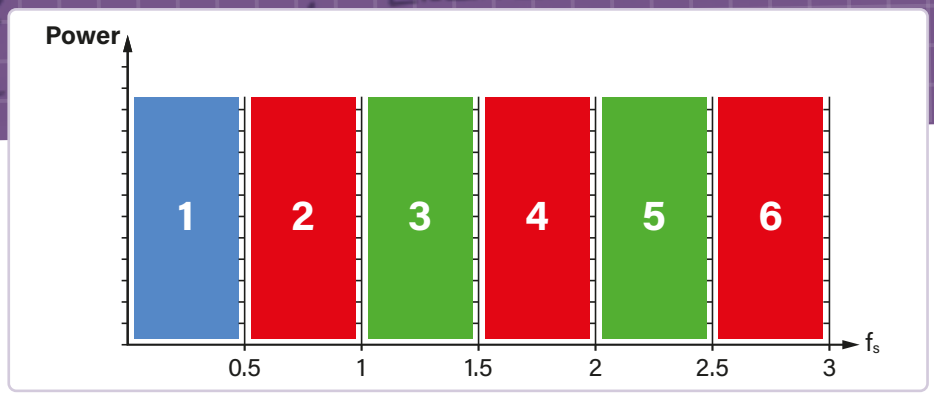

Figure 2. Zones de Nyquist 1 to 6. zone 1 = plage normale, zones 2, 4, 6 = plages d'inversion, zones 3 and 5 = zones non inversées.

(f_s) est appelée zone 1 de Nyquist (**figure 2**). Cette zone est la plage de fonctionnement « normale » d'un système d'échantillonnage. La zone 2 de Nyquist s'étend de 0,5 f_s à f_s. Ces zones se poursuivent ensuite théoriquement à l'infini à intervalles de 0,5 f_s. Les zones de Nyquist avec des nombres pairs ont une caractéristique particulière : à l'intérieur de ces zones, une inversion du spectre a lieu. Par conséquent, si la largeur de bande d'un signal à capturer est limitée à la plage de l'une de ces zones de Nyquist, le sous-échantillonnage peut être utilisé pour capturer des signaux dont les composantes de fréquence dépassent de manière significative la fréquence d'échantillonnage.

Un exemple pratique

Dans le cadre d'un projet, la fréquence d'un signal dans la plage d'environ 160 kHz à 165 kHz doit être acquise de manière fiable avec un Arduino UNO [2]. Pour des raisons techniques, le taux d'échantillonnage a été fixé à environ 154 kS/s en utilisant les registres appropriés (*prescaler* 1:8) (à proprement parler, il s'agit de 153,846 S/s ; par souci de clarté, le chiffre arrondi est utilisé dans cet article). Ainsi, la fréquence du signal est supérieure de quelques kHz à la fréquence d'échantillonnage elle-même, mais inférieure à 1,5 f_s (correspondant à la zone 3 de Nyquist). En conséquence, le repliement à attendre du fait du sous-échantillonnage se situe entre 6 kHz (160 kHz - f_s) et 11 kHz (165 kHz - f_s).
Le code Arduino est structuré de telle sorte que le convertisseur A/N de l'Arduino utilise une routine de service d'interruption (ISR) pour échantillonner le signal à l'entrée analogique 0 jusqu'à ce qu'un tampon de 512 échantillons soit rempli. La fréquence du signal (de repliement) est alors calculée à partir de ces valeurs avant que les 512 échantillons suivants soient échantillonnés et que le jeu recommence.
Pour déterminer la fréquence, l'algorithme de Goertzel a été utilisé, une forme spéciale de la transformée de Fourier discrète qui permet d'économiser des ressources [1]. Le résultat est ensuite envoyé à des fins de démonstration sur le port série, visuellement formaté pour le *Serial Monitor* d'Arduino (**figure 4**).
En utilisant le même principe, on pourrait, par exemple, capturer une FI de 455 kHz à partir d'un récepteur radio dans la zone 6 de Nyquist (3 f_s, équivalent à environ 462 kHz) en utilisant le sous-échantillonnage et démoduler les signaux FSK (NAVTEXT, RTTY, etc.) même avec un Arduino UNO à vitesse relativement faible.

Remarques

Pour que le sous-échantillonnage fonctionne comme souhaité, le CAN utilisé doit avoir la bande passante analogique requise. Celle-ci est généralement beaucoup plus élevée que la fréquence d'échantillonnage. Dans la fiche technique, vous trouverez généralement ce chiffre sous la forme d'une « bande passante à pleine puissance ». Avec l'ATmega328P, le sous-échantillonnage semble fonctionner très proprement jusqu'à la gamme des MHz, d'après mes propres expériences. Malheureusement, mes expériences avec le Raspberry Pi Pico n'ont pas donné de résultats prometteurs.

VF : Maxime Valens — 220629-04

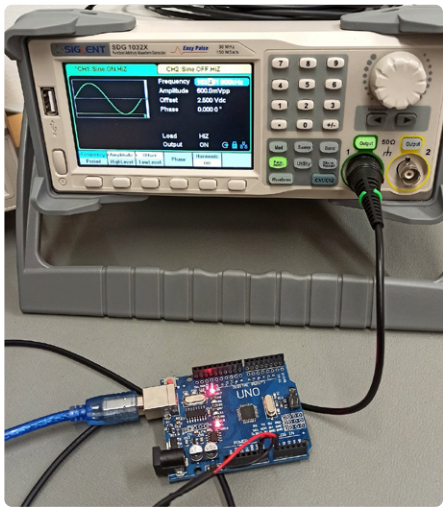

Figure 3. Système de test avec Arduino et un générateur de signaux.

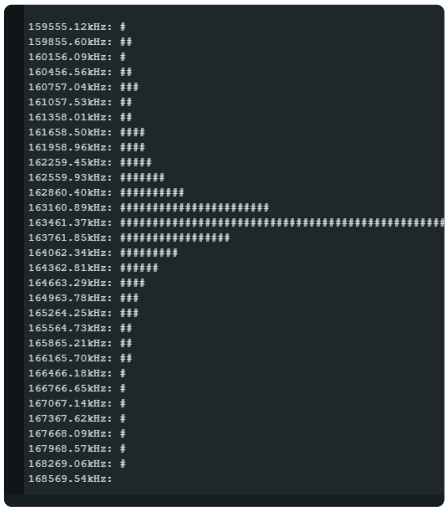

Figure 4. Les résultats sont affichés dans le Serial Monitor.

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (sebastian@baltic-lab.com), ou contactez Elektor (redaction@elektor.fr).

Produits

- **Siglent SDG1032X générateur de signaux à 2 voies (30 MHz)**
  https://www.elektor.fr/20276
- **OWON XSA810 analyseur de spectre (1 GHz)**
  https://www.elektor.fr/19714
- **Radio Logicielle HackRF One Great Scott Gadgets (1 MHz à 6 GHz)**
  https://www.elektor.fr/18306
- **MonoDAQ-U-X Système USB polyvalent d'acquisition de données (50 kéch/s)**
  https://www.elektor.fr/18766

LIENS

[1] Sebastian Westerhold (2022) : Bibliothèque Goertzel pour Arduino : https://github.com/AI5GW/Goertzel

[2] Croquis Arduino de cet article : https://www.elektormagazine.fr/220629-04