

Calculs pour DDS AD9951

1 Contourner la perte de précision due à l'arrondi de FTW sur 32 bits

1.1 Principe

$$\begin{aligned} f_{\text{out}} &= \text{FTW} \times \text{SYSCLK} / 2^{32} \\ \text{FTW}_{\text{reel}} &= f_{\text{out}} \times 2^{32} / \text{SYSCLK} \end{aligned} \tag{1}$$

à l'issue de ce calcul, FTW_{reel} est un nombre réel comprenant une partie entière et une partie décimale.

$$\text{FTW}_{\text{reel}} = \text{FTW}_{\text{entier}} + \text{FTW}_{\text{decim}}$$

Par exemple pour une SYSCLK = 240MHz et une fréquence de sortie désirée de 12MHz FTW_{reel} vaudra :

$$\begin{aligned} \text{FTW}_{\text{reel}} &= 12 \times 10^6 \times 2^{32} / 240 \times 10^6 \\ &= 214748364,8 \end{aligned}$$

FTW_{reel} sera tronqué alors en un mot (FTW) de 32 bits, un entier (uint32_t) envoyé au DDS (qui ne peut accepter autre chose), ce qui signifie que la partie décimale (0,8) sera perdue et donc que la fréquence générée ne sera pas exacte.

$$\begin{aligned} \text{FTW} &= \text{FTW}_{\text{entier}} \\ &= \text{FTW}_{\text{reel}} - \text{FTW}_{\text{decim}} \end{aligned}$$

1.2 Calcul de l'erreur relative

Commençons par calculer le rapport entre la fréquence désirée et celle synthétisée :

$$\begin{aligned} R_{\text{ds}} &= \frac{\text{FTW}_{\text{reel}} \times \text{SYSCLK} / 2^{32}}{\text{FTW} \times \text{SYSCLK} / 2^{32}} \\ &= \frac{\text{FTW}_{\text{reel}}}{\text{FTW}} \\ &= \frac{\text{FTW}_{\text{reel}}}{\text{FTW}_{\text{entier}}} \\ &= \frac{\text{FTW}_{\text{entier}} + \text{FTW}_{\text{decim}}}{\text{FTW}_{\text{entier}}} \\ &= \frac{\text{FTW}_{\text{entier}}}{\text{FTW}_{\text{entier}}} + \frac{\text{FTW}_{\text{decim}}}{\text{FTW}_{\text{entier}}} \\ &= 1 + \frac{\text{FTW}_{\text{decim}}}{\text{FTW}_{\text{entier}}} \\ &= 1 + \text{err} \end{aligned} \tag{2}$$

$$\text{avec err} = \frac{\text{FTW}_{\text{decim}}}{\text{FTW}_{\text{entier}}}$$

Remarque 1. err est un nombre sans dimension.

Pour F_{out} = 12MHz vu plus haut, cela donne :

$$\begin{aligned} R_{\text{ds}} &= 1 + \frac{0.8}{214748364} \\ &= 1 + 0.00000000372529031234 \\ &= 1.00000000372529031234 \end{aligned}$$

Cette valeur sera facilement calculée par le programme en C pilotant le DDS.

Cette valeur est très proche de 1, l'erreur est très faible (toutefois il faut noter qu'elle est d'autant plus grande que f_{out} est petite, et qu'elle peut devenir très importante pour les faibles fréquences, de quelque Hz ou dizaines de Hz, dans ce cas le choix d'un synthétiseur DDS piloté par une fréquence d'horloge importante par rapport à la fréquence générée n'est pas du tout judicieux).

Dans le cas qui nous occupe, la réalisation d'un générateur de fréquence piloté par GPS, c'est à dire visant une précision de 10^{-11} , cette erreur très faible (de l'ordre de 10^{-9}) n'est pas acceptable.

$$\begin{aligned}\text{FTW}_{\text{reel}} &= \text{FTW} \times R_{\text{ds}} \\ &= 1.00000000372529031234 \times \text{FTW}\end{aligned}$$

et donc :

$$\begin{aligned}f_{\text{outdesiree}} &= \text{FTW}_{\text{reel}} \times \text{SYSCLK} / 2^{32} \\ &= \text{FTW} \times R_{\text{ds}} \times \text{SYSCLK} / 2^{32} \\ &= f_{\text{syth}} \times R_{\text{ds}}\end{aligned}\tag{3}$$

Nous avons vu que la valeur R_{ds} peut être calculée par le microcontrôleur ATmega pilotant le DDS. Il reste à trouver le moyen de modifier la fréquence synthétisée par le DDS, connaissant R_{ds} . Ce qui peut être fait dans le cas du circuit AD9951 en « raccourcissant » certaines périodes régulièrement espacées dans le temps du signal synthétisé au moyen du sommateur de phase intégré dans le DDS, qui est pilotable par l'envoi d'un mot de 14 bits, soit 16384 pas.

Toutefois ce qui est facile à représenter en informatique c'est par exemple le nombre :

$$\begin{aligned}\text{err} &= \frac{0.8}{214748364} \\ &\simeq 0.00000000372529\end{aligned}$$

qui ne comprend pas beaucoup de chiffres significatifs, d'autant qu'on peut se contenter de trois ou quatre de ces chiffres, soit

$$\begin{aligned}\text{err} &\simeq 0.00000000372 \\ &\simeq 372 \times 10^{-11} \\ &\simeq 0b101110100 \times 10^{-11}\end{aligned}$$

représentable avec une dizaine de bits.

mais il n'en va pas de même avec le nombre :

$$\begin{aligned}R_{\text{ds}} &= 1 + \text{err} \\ &= 1.00000000372 \\ &= 100000000372 \times 10^{-11} \\ &= 0b1011101001000011101101110100101110100\end{aligned}$$

[calcul fait avec Calc sous Linux : C-style arbitrary precision calculator (version 2.12.4.4)]

Ce nombre demande 37 bits même après n'avoir gardé que trois chiffres (372), parce que c'est en réalité 12 chiffres significatifs (décimaux) qu'il comprend.

Toutefois le langage AVRGCC utilisé pour programmer le microcontrôleur permet d'utiliser facilement des nombres entiers codés sur 64 bits (uint64_t). Nous allons donc pouvoir continuer dans cette voie.

1.3 Ajustement de la fréquence par glissement continu de phase

La phase du signal généré peut être augmentée ou diminuée à tout instant par pas, avec 16384 pas pour 2π radians (360°).

Soit f_s la fréquence synthétisée en l'absence de correction, et f_d la fréquence désirée, c'est à dire synthétisée et modifiée par ajouts de sauts de phase (ces ajouts se faisant d'une manière numérique avant la conversion en sinus analogique, mais peu importe).

Soit P_s et P_d leurs périodes respectives :

$$\begin{aligned} P_s &= 1/f_s \\ P_d &= 1/f_d \end{aligned}$$

Soit R_1 le rapport [nb de périodes modifiées / nb de périodes non modifiées].
($R_1 \ll 1$; par exemple $R_1 = 10^{-6}$, librement choisi)

Remarque 2. R_1 est un nombre sans dimension.

Soit R_2 le rapport [largeur du saut / période du signal]

$$R_2 = \frac{dP}{P}$$

Remarque 3. R_2 est un nombre sans dimension. P et dP doivent être tous les deux exprimés dans la **même** unité, par exemple unité de temps (ns), ou nb de périodes de l'horloge du processeur.

Soit R_m le rapport de modification

$$\begin{aligned} R_m &= R_1 R_2 \\ R_2 &= R_m / R_1 \end{aligned}$$

Remarque 4. R_m est un nombre sans dimension.

Pour la suite des calculs, et dans un souci de représentation des nombres en binaire par le microcontrôleur, nous introduisons le nombre R_3 inverse de R_1 :

$$\begin{aligned} R_3 &= 1/R_1 \\ R_3 &= \frac{\text{nb périodes non modifiées}}{\text{nb périodes modifiées}} \end{aligned} \quad (4)$$

Remarque 5. R_3 est un nombre sans dimension.

ce qui donne :

$$R_2 = R_m \cdot R_3$$

La valeur moyenne de la modification de la période du signal obtenu peut être calculée comme si on effectuait R_m à chaque période. Chaque période subirait alors un raccourcissement (temporel) égal à $R_m \times P_s$.

Donc :

$$\begin{aligned} P_d &= P_s - P_s \times R_m \\ &= P_s (1 - R_m) \end{aligned} \quad (5)$$

Nous avons vu plus haut (3) que :

$$f_{\text{outdesiree}} = f_{\text{syth}} \times R_{\text{ds}}$$

prenons l'inverse :

$$\begin{aligned} \frac{1}{f_{\text{outdesiree}}} &= P_d = \frac{1}{f_{\text{syth}} \times R_{\text{ds}}} \\ P_d &= \frac{1}{f_{\text{syth}}} \times \frac{1}{R_{\text{ds}}} \\ P_d &= P_s / R_{\text{ds}} \end{aligned}$$

Remplaçons P_d par sa valeur calculée (en 5) :

$$P_s (1 - R_m) = P_s / R_{\text{ds}}$$

Nous voulons calculer R_m pour en déduire R_1 et R_2 à appliquer au signal.

$$1 - R_m = \frac{1}{R_{ds}}$$

$$R_m = 1 - \frac{1}{R_{ds}}$$

remplaçons R_{ds} par sa valeur(en 2)

$$\begin{aligned} &= 1 - \frac{1}{1 + \text{err}} \\ &= \frac{1 + \text{err}}{1 + \text{err}} - \frac{1}{1 + \text{err}} \\ &= \frac{1 + \text{err} - 1}{1 + \text{err}} \\ &= \frac{\text{err}}{1 + \text{err}} \end{aligned}$$

Si $\text{err} \ll 1$ alors $(1 + \text{err}) \simeq 1$

Nous pouvons donc écrire, si $\text{err} \ll 1$

$$R_m \simeq \text{err} \tag{6}$$

$$R_2 = R_m R_3$$

$$\simeq \text{err} R_3$$

$$\frac{dP}{P} = \text{err} R_3$$

$$dP = P \cdot \text{err} R_3$$

dP et P peuvent être comptée en nombre de pas, avec $P = 16384$ (dans le cas de l'AD9951).

ce qui donne, en remplaçant err par sa valeur :

$$dP = 16384 \times \frac{\text{FTW}_{\text{decim}}}{\text{FTW}_{\text{entier}}} \times R_3 \tag{7}$$

avec :

dP = nombre de pas d'avance de phase à ajouter en moyenne à chaque période.

et :

$$R_3 = \frac{\text{nb périodes non modifiées}}{\text{nb périodes modifiées}}$$