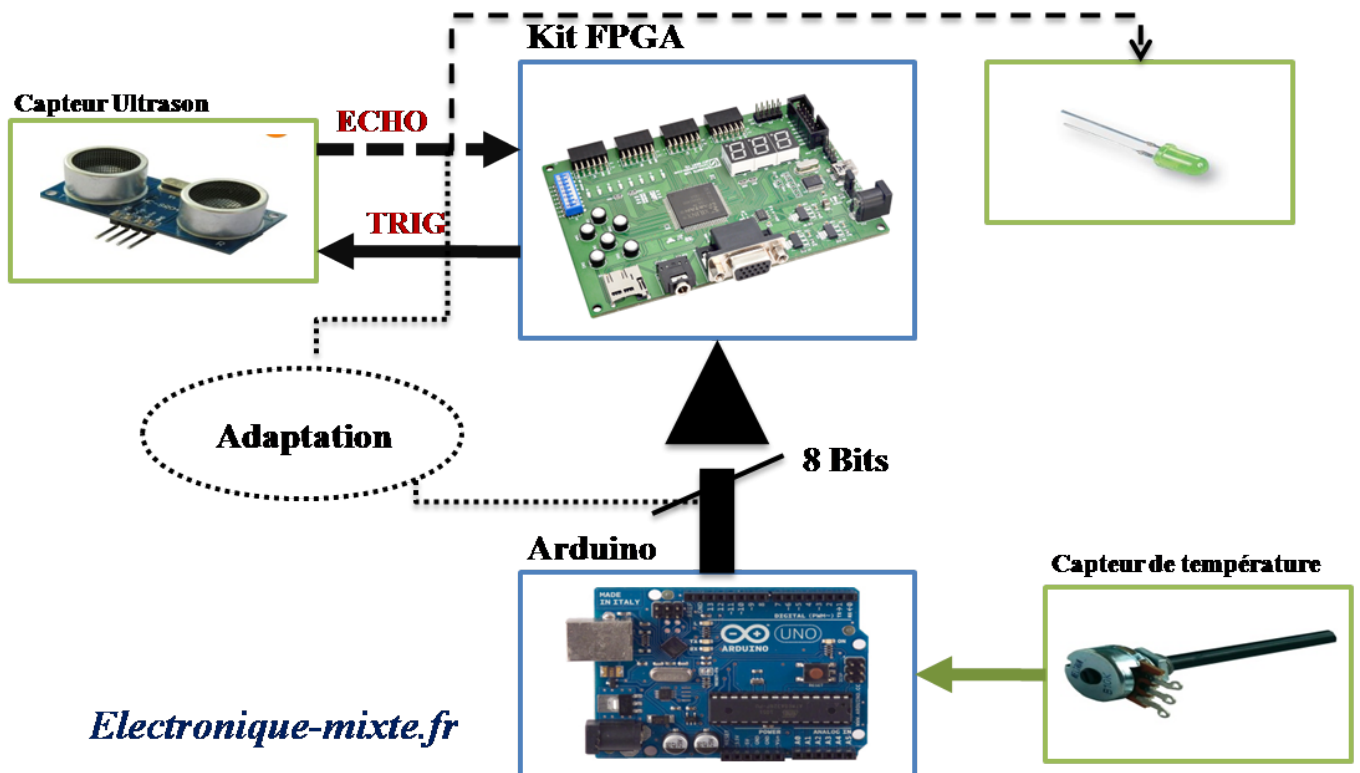


Projet électronique FPGA 4 #3/ 3 Capteur de distance ultrasonique à base du FPGA et Arduino - capteur ultrasonique

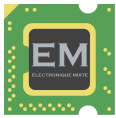


Les objectifs du projet

1. Se familiariser avec le codage en virgule fixe
2. Choix de la position de la virgule (précision)
3. Choix de nombre des bits
4. Opérations sur les **données** en virgule fixe
5. Implémentation d'une équation en virgule fixe
6. Savoir la relation entre la distance, température pour un capteur ultrasonique

Le codage en virgule Fixe

En informatique, une représentation d'un nombre en virgule fixe est un type de donnée correspondant à un nombre qui possède (en base deux ou en base dix) un nombre fixe de chiffres après la virgule. Les nombres en virgule fixe sont utiles pour représenter des



quantités fractionnaires dans un format utilisant le complément à deux quand le processeur de l'ordinateur n'a aucune unité de calcul en virgule flottante ou quand une virgule fixe permet d'augmenter la vitesse d'exécution ou d'améliorer l'exactitude des calculs. La plupart des processeurs à faible coût (ex. : [microcontrôleurs](#)) ne disposent pas d'unité de calcul en virgule flottante.

Les bits à gauche de la virgule représentent la partie entière du nombre (au sens premier du terme), c'est-à-dire l'entier se trouvant à gauche de la virgule. Chaque bit à droite de la virgule, ou « décimale binaire », correspond à l'inverse d'une puissance de 2 [wikipedia].

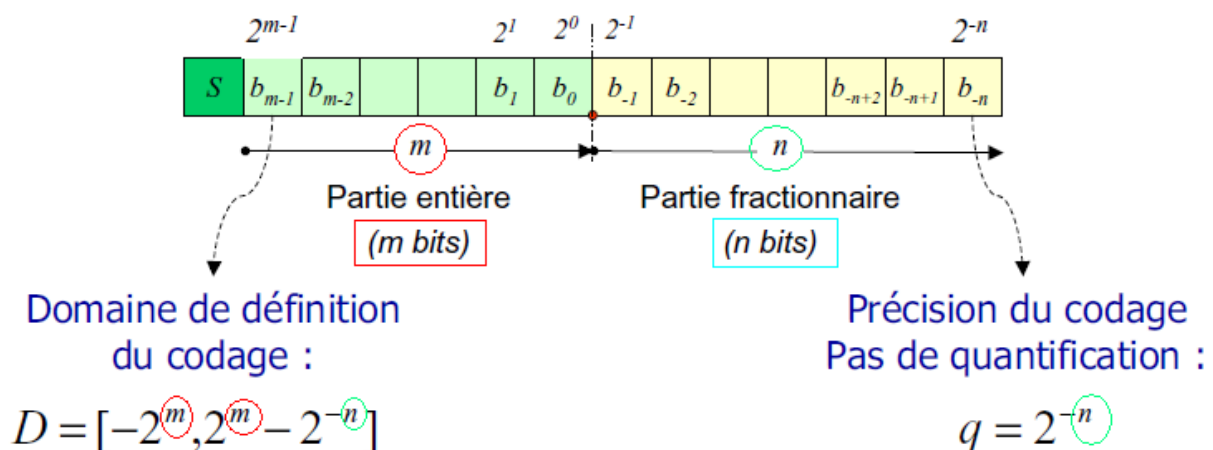
Présentation d'un nombre en virgule fixe en format Qn

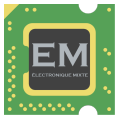
n : le nombre des bits nécessaires pour le codage de la partie fractionnelle

m : le nombre des bits nécessaires pour le codage de la partie entière

Voir les liens des [cours](#) en bas pour plus des renseignements.

$$x = -2^m S + \sum_{i=-n}^{m-1} b_i 2^i$$

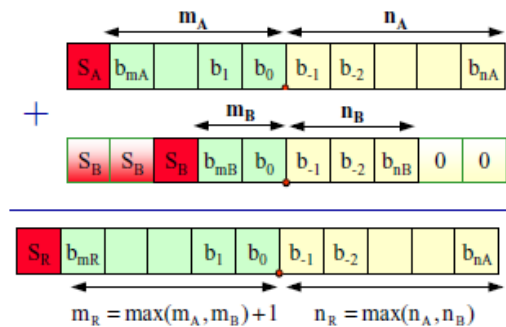




Règles de l'arithmétique en virgule fixe :

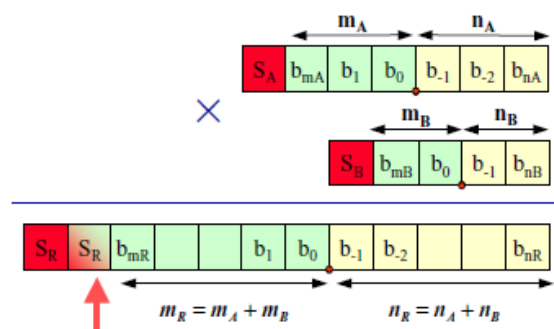
- **Addition $a+b$**

- *Choix d'un format identique*
- *Alignement de la virgule*
- *Extension de bits*



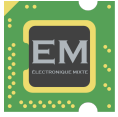
- **Multiplication $a \times b$**

- *Représentation identique*
 - ✓ *Doublement du bit de signe*



Calcul de la résolution binaire en fonction de la résolution décimale

On considère D la résolution décimale d'un nombre décimal (Ex : 10^{-3} pour $D=3$) et n la résolution binaire d'un nombre binaire (Ex : 2^{-3} pour $n=3$)



$$10^D = 2^n$$

$$\Rightarrow \text{Log} (10^D) = \text{Log} (2^n)$$

$$\Rightarrow D = n * \text{Log}(2)$$

$$n = \frac{D}{\text{Log}(2)} = D * 3.322$$

$$n = D * 3.322$$

Exemple

$$x = 10.12$$

$$x = 10 + 0.12 = x_m + x_n$$

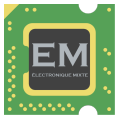
Codage de 10 (x_m) : $10 = (1100)_2$, $m=4$ bits

Codage de 0.12 (x_n) : D'après la relation ci-dessus $0.12 * 10^2 = 12$ (valeur entière, $D=2$)

On prend alors $n \geq 3.32 * 2 = 6.64 = 7$ bits, $n=7$ bits

$$X_n = \text{Round} (0.12 * 2^7) = \text{Round} (15.36) = 15 = (0001101)_2$$

La présentation binaire du nombre $x = 10.12$ sous le format Q7 sur 10 bits est la suivante :

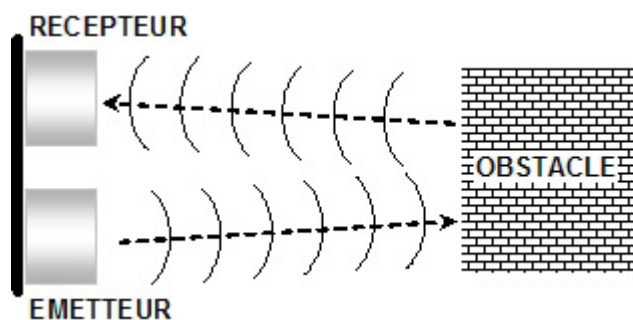


X = 1100,0001101

Alors la largeur non signée de la donnée doit être codée sur 4+7 = 11 bits ou 12 bits pour un nombre signé, la position de la virgule se trouve entre le bit 7 et le bit 8 (virgule virtuelle)

Cours : IRISA & IUT REIMS & ENS LYON

La distance en fonction de la température



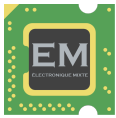
la vitesse du son dans l'air augmente quand la température augmente. $C_{\text{air}} = (331,5 + 0,6 \cdot T_{\text{emp}})$; T_{emp} température en Celcius

$$C_{\text{air}} = (331,5 + 0,6 \cdot T_{\text{emp}})$$

- Cours sur la propagation des ondes : [Lien 1](#) & [Lien 2](#)

La distance entre l'émetteur et le récepteur d est égale à $V \cdot t / 2$, avec V la vitesse de propagation de l'onde dans l'air et t le temps du trajet aller/Retour entre l'emetteur et le recepteur. Alors :

$$d = V \cdot t / 2 = C_{\text{air}} \cdot t / 2 = t \cdot (331,5 + 0,6 \cdot T_{\text{emp}}) / 2$$



$$d(m) = t*(165.75 + 0,3*T_{emp})$$

$$d(m) = (N*t_0)*(165.75 + 0,3*T_{emp})$$

$$d(m) = N*(165.75 + 0,3*T_{emp})*83.3333.10^{-9}$$

$$d(m) = N*(13.8125 + 0.025*T_{emp})*10^{-6}$$

$$d(mm) = N*(13.8125*10^{-3} + 0.025*10^{-3}*T_{emp})$$

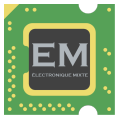
$$d(mm) = N*(K_0 + K_1*T_{emp})$$

- $t_0 = 1/f_0 = 83.33\text{ns}$ pour $f_0 = 12\text{ MHz}$
- N le valeur du compteur sur **24 bits**
- T_{emp} température entre 0° et 100° sur **8 bits**
- $K_0 = 13.8125*10^{-3}$
- $K_1 = 0.025*10^{-3}$

Codage en virgule fixe des constantes K_0 et K_1

- $K_0 = 13.8125*10^{-3} = 0.0138124$

La partie entière est codée sur 0 bits. On respectant la formule $n_{k_0} = D*3.322$ on obtient



Projet électronique FPGA 4 #3/ 3 Capteur de distance ultrasonique à base du FPGA et Arduino - capteur ultrasonique

(pour $D=3$) $n = 3 \times 3.322 = 9.966$. On prend $n_{k_0} = 10$ bits

$$(k_0)_2 = \text{Round}(k_0 \times 2^{10}) = (14)_2 = (.00000011110)_2$$

- $K_1 = 0.025 \times 10^{-3} = 25 \times 10^{-5}$

La partie entière est codée sur 0 bits. On respectant la formule $n_{k_1} = D \times 3.322$ on obtient (pour $D=5$) $n = 5 \times 3.322 = 16.61$. On prend $n_{k_0} = 17$ bits

$$(k_1)_2 = \text{Round}(k_1 \times 2^{17}) = \text{Round}(32.768) = (33)_2 = (.000000000000100001)_2$$

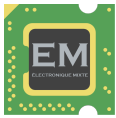
Remarque : Tous les bits des constantes k_0 et K_1 sont consacrés à la partie fractionnelle ($0 < k_0 < 1$ et $0 < k_1 < 1$)

- $K_0(m,n) = (0,10)$, la longueur de la constante K_0 est égale à $m+n = 10$ bits, format en virgule fixe (voir la section ci-dessus)
- $K_1(m,n) = (0,17)$, la longueur de la constante K_1 est égale à $m+n = 17$ bits

Codage de la valeur de la température

La température Temp est codé sur 8 bits, la valeur de la température variée entre 0°C et 100°C . Afin de convertir la valeur binaire en température il suffit d'appliquer la règle de trois ! 0 pour 0°C et 255 (8 bits) pour 100°C .

$$T_{\text{emp}}(^{\circ}\text{C}) = T_{\text{emp}} * 100/255$$



$$T_{\text{emp}}(^{\circ}\text{C}) = T_{\text{emp}} * 0.3922$$

$$T_{\text{emp}}(^{\circ}\text{C}) = T_{\text{emp2}} * K_{\text{temp}}$$

$$\Rightarrow d(\text{mm}) = N * (K_0 + K_1 * T_{\text{emp2}} * K_{\text{temp}}) \quad (1)$$

- $K_{\text{temp}} = 0.3922 = 39.22 * 10^{-2}$

La partie entière est codée sur 0 bits. On respectant la formule $n_{k_{\text{temp}}} = D * 3.322$ on obtient (pour $D=2$) $n = 2 * 3.322 = 6.64$. On prend $n_{\text{temp}} = 7$ bits

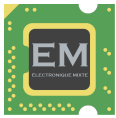
$$(k_{\text{temp}})_2 = \text{Round}(k_{\text{temp}} * 2^7) = \text{Round}(50.2016) = (50)_2 = (.0110010)_2$$

Conclusion

Codage des variables et constantes sur le format $Q(m,n)$ avec n le nombre des bits de la partie entière, et m le nombre des bits de la partie fractionnelle. $m+n$ est le nombre des bits total de la variable.

- T_{emp2} : **(8,0), Taille 8 bits [8+0]**

- K_{temp} : **(0,7), Taille 7 bits**



- **K_0 : (0,10), Taille 10 bits**
- **K_1 : (0,17), Taille 17 bits**
- **N : (24,0), Taille 24 bits**
- **K_t : (0,20) (voir la formule 2), Taille 20 bits**

On peut simplifier la formule (1) de la distance en remplaçant $K_{temp} * K_1$ par $K_t = K_{temp} * K_1$

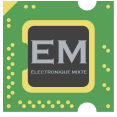
$$d(\text{mm}) = N * (K_0 + T_{emp2} * K_t) \quad (2)$$

- $K_t = K_{temp} * K_1 = (100/255) * 0.025 * 10^{-3} = 9.8 * 10^{-6}$

La partie entière est codée sur 0 bits. En respectant la formule $n_{kt} = D * 3.322$ on obtient (pour $D=6$) $n = 6 * 3.322 = 19.9$. On prend $n_{kt} = 20$ bits

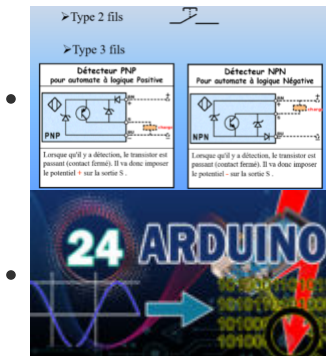
$$(k_t)_2 = \text{Round}(k_t * 2^{20}) = \text{Round}(10.27) = (10)_2 = (.000000000000000001010)_2$$

Pour optimiser la longueur de K_t , on peut intégrer la fonction K_{temp} sur l'Arduino, au lieu de transmettre la valeur brute sur 8 bits, on transmet la valeur de la température [0-100] (après multiplicateur par 100/255 et arrondi de la valeur finale).



Determination de la taille de la variable d (distance) en mm [Formule 2]... A venir ☐

Dernières réalisations



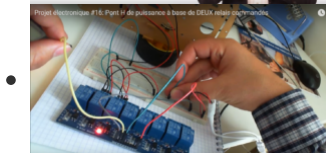
GEMMA #3/3

Arduino #24: Comment convertir un signal analogique en un signal

logique (TOR) - 3 techniques



Quelle est la différence entre la créativité et l'innovation ?

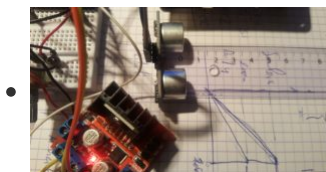


Projet électronique #16: Pont H de puissance à base de DEUX relais

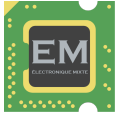
commandés - Commande du sens de rotation d'un moteur à CC






C'est quoi Enigma ?



Projet électronique #17: Sèche-mains ultrason avec Arduino



Projet électronique FPGA 4 #3/ 3 Capteur de distance ultrasonique à base du FPGA et Arduino - capteur ultrasonique

-  Arduino #11: Les tableaux 1D
-  Matlab #6: Les 4 fonctions des Graphiques 2D
-  Projet électronique FPGA #8 : Commande d'un moteur à CC - V2

Articles

- [Projet électronique FPGA #9 : Calcul de la factorielle de n: Implémentation sur carte FPGA](#)
- [Projet électronique FPGA #6 : Commande synchrone multicanaux d'un moteur à CC](#)
- [Projet électronique FPGA #5 : Générateur des signaux #V1](#)
- [Projet électronique FPGA #10 : Commande factorielle d'un moteur à CC avec Arduino et FPGA](#)
- [Projet électronique : Traitement du signal avec Arduino # Lissage & Seuillage d'un signal 3/3](#)
- [Projet électronique FPGA #8 : Commande d'un moteur à CC - V2](#)
- [Projet électronique #16: Pont H de puissance à base de DEUX relais commandés - Commande du sens de rotation d'un moteur à CC](#)