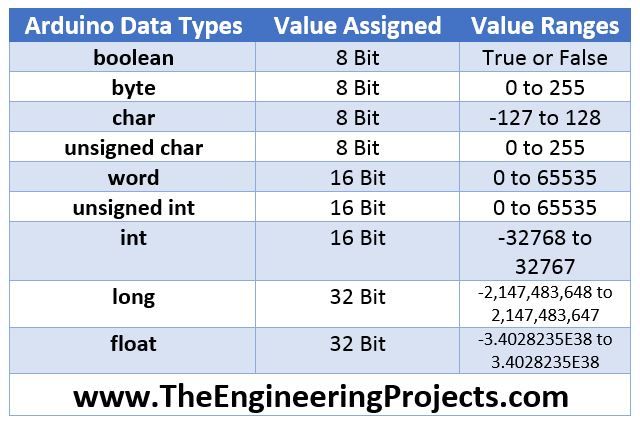
Conversion de types en C Arduino Nano (12/10/22)



Dans le sens de la descente on dit qu’il y a « **promotion** » et « **demotion**» quand on monte.

Noter qu’il peut exister d’autres types tel : long long, double, long double…

On distingue 3 types de conversion en C: **Implicite, assignement, explicite** (typecast)

Conversion Implicite, dite aussi automatique.

Exemple : Int A ; float B ; float C ;

C = A/B ;

Le compilateur effectue les conversions par copie de paires de variables.

Il choisit la variable de type le plus élevé et aligne l’autre variable, si nécessaire, par promotion de son type

Ici, la variable de type le plus élevé est B, float.

Il promeut donc A en type float, effectue la division, et affecte le résultat à C toujours en float.

Conversion par assignement

La valeur à droite du signe = est toujours alignée sur le type de la valeur à gauche.

Exemple : int A ; float B ; long C ;

C = A/B ;

A/B est en float comme ci-dessus mais converti en type long puis affecté à C.

Cette conversion peut être une promotion ou une demotion, selon le type de C.

Ici c’était une demotion.

Conversion explicite (typecast)

Exemple : Int A =4; int B=3 ; float C ;

C = A/B ; donne C=1.00

La division a été exécutée entre deux entiers puis promue en type float, mais la partie décimale était déjà perdue.

On va effectuer un « cast » sur A par (float)A. Noter qu’un cast s’applique à la seule variable le suivant

C = (float)A/B ; donne C = 1.33

A a été promue au type float. B s’est aligné sur ce type et le résultat conservé en float comme C.

**CONCLUSION**

Pour évaluer une expression à droite du signe = le compilateur recherche la variable de plus haute promotion.

Puis il promeut les autres variables en les alignant.

Tout ceci se fait par copie sans affecter le type déclaré des variables.

Pour terminer ce type est converti dans le type à gauche du signe =.

C’est en général une promotion mais ce peut être aussi une demotion.

Piège

Exemple de piège : Int A ; float B ;

L’objectif est de calculer (180- A/100) en flottant.

Naïvement on peut écrire

B = (float)((180-A/100)) ;

Mais la parenthèse étant calculée en entiers, on perd les décimales.

La solution est très simple : utiliser ‘100.0 ‘ Au lieu de 100 ce qui impose un calcul en float, et rend inutile le cast

B = 180-A/100.0;