

# Manuel de Universal converter

25 octobre 2008

Sous ce nom un peu pompeux de «Universal converter» se cache en fait un simple programme pour TI84 créé dans le cadre d'un concours de programmation sur [le site TI-Bank](#). Vous trouverez dans ce document une partie destinée à un hypothétique utilisateur et une partie décrivant de fonctionnement interne.

## Table des matières

<b>1</b>	<b>Manuel de l'utilisateur</b>	<b>1</b>
1.1	Conversions d'unités	1
1.2	Conversions de dates	1
1.3	Conversions de bases	2
1.4	Constantes	2
<b>2</b>	<b>Manuel du programmeur</b>	<b>2</b>
2.1	Conversions d'unités	2
2.2	Conversions de bases	3
2.3	Conversions de dates	3
2.4	Constantes	3

## 1 Manuel de l'utilisateur

Uconvert est un programme pour TI84+ (non testé sur TI83. Devrait fonctionner mais (encore) plus lentement), permettant de convertir une pléthore d'unités et de bases. La liste détaillée se trouve dans le document *Unites.ods* annexe.

*Remarque :* La conversion d'unités informatique peut être déroutante : en effet, on y trouve le trop peu connus kibioctet, abrégé kio. Il s'agit en fait de la [version normalisée](#) du kilooctet. Depuis 1998,  $1 \text{ kilooctet} = 1000 \text{ octets}$ , tandis que  $1 \text{ kibioctet} = 1024 \text{ octets}$ . Il en va de même pour le Mébioctet, le Gibioctet et le Tébioctet. C'est ainsi.

### 1.1 Conversions d'unités

Le premier menu demande d'abord le type d'unités (Longueurs, surfaces, volumes...), puis la valeur que vous souhaitez convertir. Ensuite, il faut simplement sélectionner quelle est l'unité d'entrée et l'unité de sortie.

*Exemple :* Je doute qu'un exemple soit utile ici, mais j'ai de la place et ça ne tue pas d'arbres. D'ailleurs je pourrai raconter ma vie, mais bon. On souhaite convertir  $150\text{cm}^3$  en litres. On ouvre donc Super Units Converter  $v1.0^{\text{©tm}}^{\text{®}}$  et on sélectionne «Unités» dans le menu principal. Puis on sélectionne «Volumes», on entre 150, on sélectionne  $\text{cm}^3$  et litres. Et voilà le travail.

### 1.2 Conversions de dates

Une date «brute» est, par exemple, 3124 secondes. On peut convertir cette chose en secondes, minutes, heures, jours et années ou convertir un certain nombre d'années, jours, heures, minutes et secondes en une valeur brute.

*Exemple* : Convertissons 3124 secondes en version «propre» : on sélectionne, dans le menu, *brute*→*propre* puis *Entrée en secondes*, c'est-à-dire 1. On écrit ensuite 3124 et on obtient 52 minutes et 4 secondes. Si on veut vérifier ce résultat (qui est juste, bien entendu), on sélectionnera *propre*→*brute*, suivi de *sortie en secondes* et 0 années, 0 jours, 0 heures, 52 minutes et 4 secondes. **3124!**

*Remarque* : Il n'y a pas de mois, car ce n'est pas une durée fixe (28, 29, 30 ou 31 jours).

Autre remarque : une année, à cause des années bisextiles, vaut 365,25 jours (un jour en plus tous les 4 ans) et non 365.

### 1.3 Conversions de bases

Petit rappel sur les bases : nous, humains, utilisons maintenant la base 10. C'est-à-dire que nous écrivons les nombres de cette façon :  $592 = 5 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$ . La machine que vous utilisez pour lire ces lignes les écrit en base 2, donc avec seulement deux chiffres :  $592_{10} = 10\ 0101\ 0000_2 = 1 \times 2^9 + 0 \times 2^8 + \dots + 1 \times 2^6 + 1 \times 2^4$ . Si vous ne comprenez pas tout, documentez vous sur [wikipédia](https://fr.wikipedia.org/wiki/Num%C3%A9rique), par exemple. Généralement, on utilise les bases 10, 2, 16 et plus rarement 3 et 8. Si Uconvert vous en propose 34, il y en a 29 qui ne serviront sans doute jamais. C'était juste plus simple à mettre en place.

Pour convertir, vous entrez la base du nombre entré, celle du nombre à sortir, et hop.

*Exemple* : Pour convertir 592 de la base 10 vers la base 2, on entre *Base entrée : 10*, *Base sortie : 2* et *Nombre : 592*.

### 1.4 Constantes

Bah c'est des constantes. Rien de fantastique ni de très utile.

## 2 Manuel du programmeur

Ce programme, écrit en Ti-Basic, est, aux premiers abords, très compliqué à comprendre du fait qu'il est illisible. Je vais donc en expliquer les grandes lignes.

Tout d'abord, on a le menu principal, qui conduit à 5 labels :

- conversions d'unités,
- conversions de bases,
- conversions de temps,
- constantes,
- à propos.

Je ne détaillerais pas la partie «à propos», non qu'elle soit inintéressant, mais ce n'est pas le sujet.

### 2.1 Conversions d'unités

Le menu de sélection de types d'unités prends en compte un **getkey**→**K**. Si  $K=105$ , le type d'unité en cours ( $M$ ) est sélectionné. Si  $K=24$ , on décrémente le type d'unité et si  $K=26$ , on l'incrémente. La lenteur du menu n'est pas due, bien entendu, à mon ignorance totale en matière d'optimisation, mais à la lenteur de la calculatrice elle-même. Selon le type d'unités sélectionné, on entrera les labels du menu suivant dans Str1-9, le Str0 étant réservé pour plus tard.

*Remarque* : On est alors confronté à un problème : il n'y a que 9 chaînes de caractères, et on peut avoir besoin de plus. Il faudra alors ajouter les labels à partir de 10 dans le menu lui-même.

On a maintenant une boucle *for* à deux itération : elle permet de sélectionner l'unité d'entrée et l'unité de sortie, stockées dans  $L_1$ . Ensuite, selon la valeur  $M$  (type de conversions) et la valeur  $N$  (unité entrante/sortante), on placera les valeurs adéquates dans  $L_2$ . Ces valeurs sont en fait calculées par rapport à la valeur de référence, qui est la plupart du temps, la valeur du système international.

*Exemple* : la valeur du kilomètre sera  $10^3$  car la valeur de référence est le mètre.

La conversion se fera donc d'abord de la valeur d'entrée vers la valeur de référence, puis de la valeur de référence vers la valeur de sortie. Dans les second et troisième menus, la sélection se fait comme dans le premier. La plupart des conversions se font par de simples multiplications. Cependant, les températures sont des fonctions un peu plus compliquées, puisque ce sont des additions et multiplications. On mettra donc la valeur de  $N$  à 1, puis on appliquera la fonction et stockera directement le résultat dans  $\Theta$ . Si  $(L_1(1)! = 1$  et  $L_2(1) = 1)$  ou  $(L_1(2)! = 1$  et  $L_2(2) = 1)$ , on affichera directement  $\Theta$  au lieu de  $\frac{L_1(1) \times \Theta}{L_1(2)}$

## 2.2 Conversions de bases

C'est un algorithme de conversion d'une base vers décimal, puis de décimal vers la base souhaitée. Le premier utilise la chaîne de caractère entrée et ajoute une valeur en fonction de la position de ce caractère, sa valeur et la base d'entrée. La seconde procède simplement avec une suite de divisions euclidiennes.

## 2.3 Conversions de dates

Là encore, c'est plutôt simple : le mode *brute*→*propre* utilise la fonction intégrée **timeCnv()** et le mode *propre*→*brute* fait quelques multiplication et divisions.

## 2.4 Constantes

Le menu est le même que pour la conversion d'unités.