

## Instruction Arduino « map »

### map(value, fromLow, fromHigh, toLow, toHigh)

#### Description

Ré-étalonne un nombre d'une fourchette de valeur vers une autre fourchette. Ainsi, une valeur basse source sera étalonnée en une valeur basse de destination, une valeur haute source sera étalonnée en une valeur haute de destination, une valeur entre les deux valeurs source sera étalonnée en une valeur entre les deux valeurs destinations, en respectant la proportionnalité. Cette fonction est très utile pour effectuer des changements d'échelle automatiques.

Cette fonction ne contraint pas les valeurs à rester dans les limites indiquées, car les valeurs en dehors de la fourchette sont parfois attendues et utiles. L'instruction constrain() doit être utilisée également avant ou après cette fonction, si les limites de la fourchette utilisée doivent être respectées.

Noter que la limite basse de chaque fourchette peut être supérieure ou inférieure à la limite haute, dès lors l'instruction map() peut être utilisée pour inverser l'ordre des valeurs, par exemple :

```
y = map(x, 1, 50, 50, 1); // y évolue en sens inverse de x (càd si x = 1, y=50 et inversement)
```

Cette instruction supporte également des valeurs négatives, tel que dans cet exemple :

```
y = map(x, 1, 50, 50, -100);
```

Cette utilisation est aussi valide et fonctionne normalement.

L'instruction map() utilise des valeurs entières qui ne peuvent fournir les décimales, alors que les calculs le devraient. La partie décimale est tronquée, et les valeurs ne sont pas arrondies ou moyennées.

#### Syntaxe

```
map (valeur, limite_basse_source, limite_haute_source, limite_basse_destination,  
limite_haute_destination)
```

#### Paramètres

valeur : le nombre à ré-étalonner

limite\_basse\_source: la valeur de la limite inférieure de la fourchette de départ

limite\_haute\_source: la valeur de la limite supérieure de la fourchette de départ

limite\_basse\_destination: la valeur de la limite inférieure de la fourchette de destination

limite\_haute\_destination: la valeur de la limite supérieure de la fourchette de destination

#### Valeur renvoyée

La valeur ré-étalonnée

#### Exemple

```
/* Etalonne une valeur analogique 10 bits (0-1023) sur 8 bits (0-255) */
```

```
void setup() {}
```

```
void loop()
```

```
{
```

```
  int val = analogRead(0);
```

```
  val = map(val, 0, 1023, 0, 255); // Ré-étalonne la valeur entre 0 et 1023 sur une fourchette entre 0 et 255
```

```
  analogWrite(9, val);
```

```
}
```