

Le Port GPIO

GPIO (General Purpose Inputs/Outputs)

Configurer les horloges de périphériques, les GPIOs et la pin en lecture/écriture sur le GPIO.

Rappels de syntaxes :

Préfixe = la base

Les écritures en décimal :42 en octale :052 en hexadécimal : 0x2A en binaire :0b101010.

Suffixe = le codage machine

non signé suffixe u : 42u ->32bits

signé pas de suffixe :42 -> complément à deux 32bits

long suffixe l :42.0u -> complément à deux 64 bits

float suffixe f :42.0f -> IEE 754 sir 32 bits

double suffixe d:42.0d -> indisponible sur stm32f3

Opérations bits à bits (opération binaire des entiers)

Décalage Logique droite Décalage Logique gauche

01011110 ->

<- 01011110

>>1 = 001011110

<<1 = 010111100

Décalage shift décale tous les chiffres et le vide est remplis avec des zéros.

Dans ce cas un décalage vers la gauche de n rangs équivaut à une multiplication par 2^n pour les non-signés ou entier positif.

Dans ce cas un décalage vers la droite de n rangs équivaut à une division par 2^n pour les entiers positifs. Ce n'est pas le cas pour les non-signés, on obtient un résultat qui n'a pas grand sens mathématiquement.

Ce type d'opération s'avère plus rapide que des instructions de multiplication ou division.

Détails : Lorsqu'on effectue un décalage à droite -, certains bits de notre nombre vont sortir du résultat et être perdus. Cela a une conséquence : le résultat est tronqué ou arrondi. Plus précisément, le résultat d'un décalage d'un décalage à droite de n rangs sera égal à la partie entière du résultat de la division par 2^n .

Décalage Arithmétique

Similaire au décalage logique à un détail près : Celui de droite, le bit de signe n'est pas modifié et on remplit les vides par le décalage avec le bit de signe.

Décalage Logique droite Décalage Logique gauche

11011110

100111100

>>1 = 111011110

<<1 = 10111100

Ces instructions sont équivalentes à une multiplication/division par 2^n que le nombre soit signé ou non. A un détail près l'arrondi n'est pas fait de la même façon pour les nombres positifs et négatifs ($9/2=4$ et $(-9)/2=-5$)

Rotation

Similaire mais les bits sortant sont réinjectés à la place libérée par le décalage. Utile en cryptographie algorithme de chiffrement ou si on doit manipuler des données bit par bit comme un calcul du nombre de bit à 1 dans un nombre.

Mais dans le code ?

En java il n'y a pas de distinction de décalage à gauche (logique ou arithmétique) ont le même effet donc (<<) Droite (>>> logique) (>> arithmétique)

En C/C++ le type est déterminé par le type d'entier à décaler. Les entiers signés décalage arithmétique tandis que la logique est utilisée sur les entiers non signés.

Décalages

0b100101

0b101101

<< 3 = 0b100101000

>> 2 = 0b001101

Opérations bits-à-bits

Conjonctions & ET logique

```
0b1010  [0&0=0] [0&1=0] [1&1=1] (Test)
&0b1100
= 0b1000
```

Disjonction | OU

```
0b1010  [0|0=0] [0|1=1] [1|1=1] (Mettre)
|0b1100
= 0b1110
```

Xor ^

```
0b1010  [0^0=0] [0|1=1] [1|1=0]
^0b1100
= 0b0110
```

Négation ~

```
0b1010 = 0b0101
```

Les masques

