

Si on veut qu'un code C++ fonctionne sur la carte STM32F3Discovery il doit contenir ces caractéristiques :

- ✓ Les bibliothèques de STM32 et ceux des fonctions utilisées
- ✓ La définition des Pin utilisées
- ✓ L'initialisation des Ports et des Pin GPIO :
  - Mode(entrée/sortie)
  - Configuration
- ✓ La fonction principale
- ✓ Une boucle infinie contenant les instructions à exécuter

### ☒ LED externe

La formule de calcul de résistance d'une LED électroluminescente :

$$R = (V_s - V_f) / I_f$$

LED rouge, orange, jaune (ambre) : 1.8V à 2V (1mA à 20mA)

LED verte standard (vert clair) : 1.8V à 2.2V (1mA à 20mA)

LED bleu : 3.2V à 3.9V (1mA à 20mA)

Les niveaux des tensions en sortie : 0V / 3.1V

### ☒ Signal carré de fréquence 100Hz

Pour Obtenir un signal carré de fréquence 100Hz on met une boucle avec un compteur allant de 0 à 30000 et on met à 1 la sortie puis la même durée avec sortie mise 0.

```

1 #include <stm32f30x.h>
2 void delay (unsigned int j);
3 int main (void)
4 {
5     RCC->AHBENR |= RCC_AHBENR_GPIOEEN;
6
7     GPIOE->MODER = (GPIOE->MODER & 0xFFFF3FFFF) | 0x00040000;
8     GPIOE->OTYPER = GPIOE->OTYPER & 0xFFFF3FFFF;
9     GPIOE->OSPEEDR = (GPIOE->OSPEEDR & 0xFFFF3FFF);
10    GPIOE->PUPDR = (GPIOE->PUPDR & 0xFFFF3FFFF);
11    while(1)
12    {
13        delay(4000);
14        GPIOE ->ODR= (GPIOE->MODER & 0xFFFFDFFF) | 0x00000200;
15        delay(500);
16        GPIOE ->ODR= (GPIOE->MODER & 0xFFFFDFFF);
17    }
18 }
19 void delay(unsigned int j)
20 {
21     unsigned int k,l;
22     for (k=0;k<j;k++)
23         for (l=0;l<1427;l++);
24 }
25
26
27

```

<=> 0x00200000  
 configuration de l'horloge périphérique AHB pour le port E

01 dans les bits 18 et 19 pour configurer le PE9 en mode sortie

00 dans les bits du PE9(etat de reinitialisation)

mis a 1 du PE9

mis a 0 du PE9

fonction qui utilise un compteur pour bloquer le programme principale pendant un certain temps j(pas optimale)

## Le Timer

Le STM32F303VCT6 dispose d'une variété de Timers qui ont la même architecture et servent à des fins différentes. Ils peuvent utiliser l'horloge interne ou des horloges externes (différentes fréquences : HSI, LSI, HSE, LSE) et sont connectés à différents Bus (APB1, APB2, AHB).

Le SysTick est un timer qui se trouve dans le Bus principale AHB (profite des 72MHz) et qui peut utiliser l'horloge interne ou externe. Pour le configurer il faut ajouter l'instruction suivante dans le code :

```
SysTick_Config(Reload_Value);
```

avec  $Reload\_Value = SysTick\ Counter\ Clock\ (Hz) \times Desired\ Time\ base\ (s)$

Le Reload\_Value représente le temps que le timer attend avant l'exécution du programme d'interruption qui se trouve dans la fonction SysTick\_Handler () dans le fichier stm32f30x\_it.c .

- Configuration du systick pour avoir une fréquence de 100 Hz :  
 $Reload\_Value = SystemCoreClock / 1000$   
 On ajoute dans le programme principale : `SysTick_Config(SystemCoreClock / 1000);`