



# Electronique Numérique

## Systemes à microprocesseur

S3 : Fonctions Mbed

Frédéric Giamarchi

IUT GEII Nîmes

Centre Spatial Universitaire (Montpellier Nîmes)

Université de Montpellier

*frederic.giamarchi@umontpellier.fr*

# Mbed Fonctions API

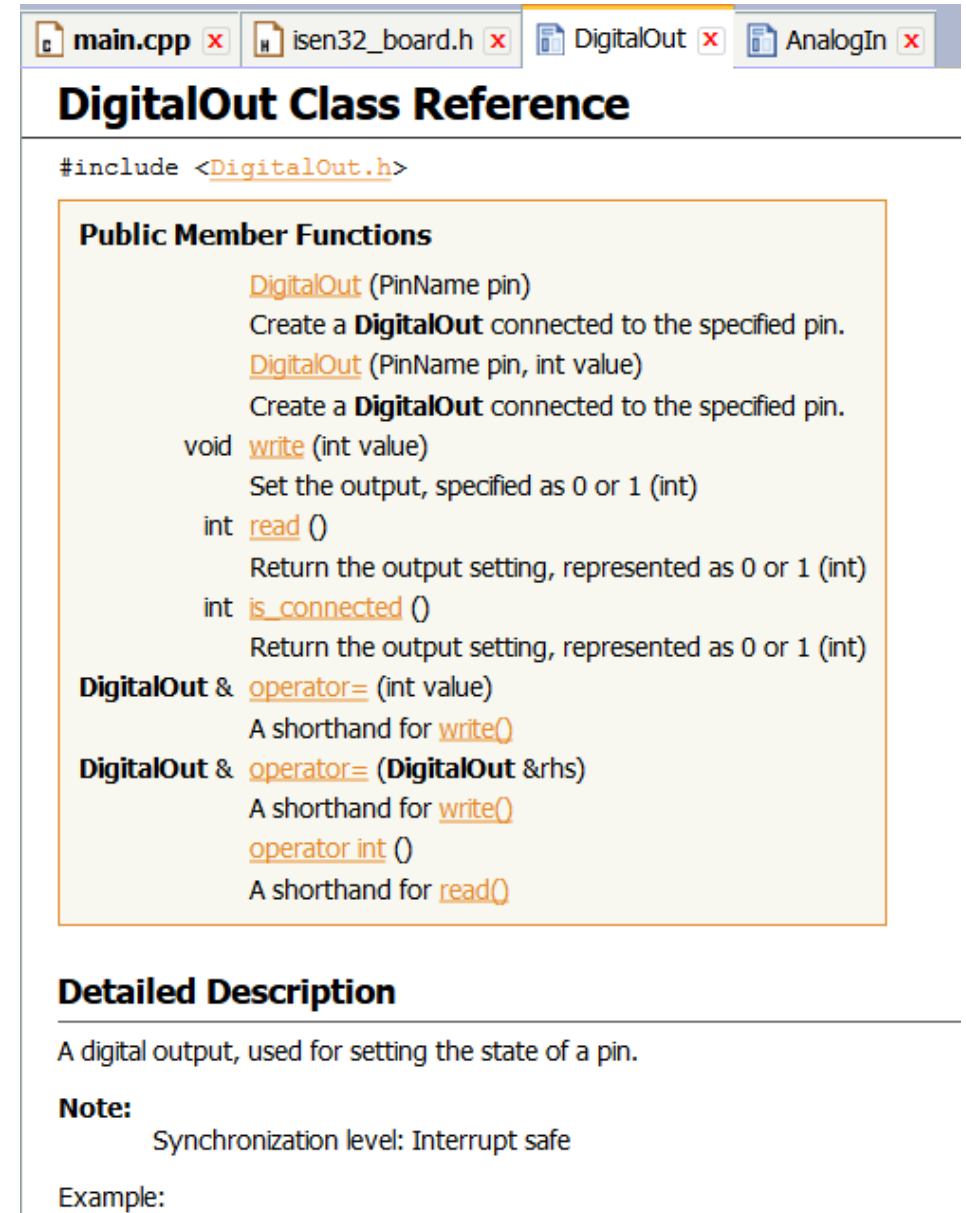
## (Application Programming Interfaces)

- Analog I/O
  - AnalogIn Lecture d'une tension sur une ligne en entrée
  - Analogout Fixe un tension sur une ligne en sortie
- Digital I/O
  - DigitalIn Configure une ligne en entrée logique
  - DigitalOut Configure une ligne en sortie logique
  - PwmOut Configure une ligne en sortie PWM
  - InterruptIn Configure une ligne en entrée pour déclencher un évènement lorsque la ligne change d'état
- Timers
  - Timer Créer, démarre, arrête et lit un compteur (timer)
  - wait, wait\_ms pause pendant une durée choisie
  - time lecture et réglage de l'horloge temps réel
- Digital Interfaces
  - Serial Fonctions du port série (UART) asynchrone
  - I2C Fonctions du bus I2C (Inter Integrated Circuit) synchrone
  - SPI Fonctions du bus SPI (Serial Peripheral Interface) synchrone
  - CAN Fonctions du bus CAN (Controller Area Network) asynchrone
- Autres fonctions USB, Mbed OS, File System, Networking



# Fonctions Mbed (OS)

- La librairie des fonctions mbed est appelée en début de chaque programme.
  - `#include "mbed.h"`
- Pour consulter les diverses fonctions et des exemples de leur utilisation, sélectionner chaque fonction dans mbed/Classes



The screenshot shows a code editor with several tabs: `main.cpp`, `isen32_board.h`, `DigitalOut`, and `AnalogIn`. The active tab is `DigitalOut`, which displays the **DigitalOut Class Reference**. The reference includes the following information:

```
#include <DigitalOut.h>
```

**Public Member Functions**

- `DigitalOut (PinName pin)`  
Create a **DigitalOut** connected to the specified pin.
- `DigitalOut (PinName pin, int value)`  
Create a **DigitalOut** connected to the specified pin.
- `void write (int value)`  
Set the output, specified as 0 or 1 (int)
- `int read ()`  
Return the output setting, represented as 0 or 1 (int)
- `int is_connected ()`  
Return the output setting, represented as 0 or 1 (int)
- `DigitalOut & operator= (int value)`  
A shorthand for `write()`
- `DigitalOut & operator= (DigitalOut &rhs)`  
A shorthand for `write()`
- `operator int ()`  
A shorthand for `read()`

**Detailed Description**

A digital output, used for setting the state of a pin.

**Note:**  
Synchronization level: Interrupt safe

Example:

# DigitalOut

Configure et contrôle une ligne logique en sortie

- DigitalOut **Name(Pin);** Déclaration d'une ligne en sortie
  - **Name** est le nom que vous souhaitez donner à cette broche
  - **Pin** est nom d'une des broches du  $\mu\text{C}$
- Broches
  - PORT A à H
- Fonctions associées
  - **Name = 1;** mettre à 1
  - **Name = 0;** mettre à 0

```
#include "mbed.h"

DigitalOut myled(LED1);

int main() {

    while(1) {
        myled = 1;      // LED is ON
        wait(0.2);     // tempo de 200 ms
        myled = 0;     // LED is OFF
        wait(1.0);     // tempo de 1 sec
    }
}
```

# DigitalIn

Configure et contrôle une ligne logique en entrée

- DigitalIn **Name(Pin);** Déclaration d'une ligne en entrée
  - **Name** est le nom que vous souhaitez donner à cette broche
  - **Pin** est nom d'une des broches du  $\mu\text{C}$
- Broches
  - Port A à H
- Fonctions associées
  - variable = **Name;** renvoie 0 ou 1
  - **Name.mode(PullUp);** PullUp, PullDown, PullNone

```
#include "mbed.h"

DigitalIn BP(PC_0);
DigitalOut Led(LED1);

int main() {
    while(1) {
        if(BP == 0) {           // Test appui sur bouton
            Led = !Led;        // Inverse l'état de la ligne
        }
        wait(0.25);           // Tempo de 250 ms
    }
}
```

# wait

Génère une temporisation

- `wait(float value);` valeur en sec
- `wait_ms(int value);` valeur en ms
- `wait_us(int value);` valeur en  $\mu$ s

- **Remarque :**

- Fonction déconseillée
- Cette fonction entre dans une boucle d'attente.
- Le  $\mu$ C est bloqué dans cette fonction
- Il existe une autre alternative.

```
#include "mbed.h"

DigitalOut myled(LED1);

int main() {
    while(1) {
        myled = 1;      // LED is ON
        wait(0.2);      // 200 ms
        myled = 0;      // LED is OFF
        wait(1.0);      // 1 sec
    }
}
```

# AnalogIn

Lecture d'une tension sur une ligne en entrée analogique

- AnalogIn **Name(Pin)**; Déclaration d'une ligne en entrée analogique
  - **Name** est le nom que vous souhaitez donner à cette broche
  - **Pin** est nom d'une des broches du  $\mu$ C, en mode Analogique
- Broches
  - A0 à A7
- Fonctions associées
  - **Name.read()**; renvoie 0.0 à 1.0
  - **Name.read\_u16()**; renvoie un 16 bits

```
#include "mbed.h"

AnalogIn Mes(A0);
DigitalOut Led(LED1);

int main() {
    while(1) {
        if(Mes.read() > 0.5)
            Led = 1;    // LED is On
        else
            Led = 0;    // LED is ON
    }
}
```

# Serial

## Communication série asynchrone

- Serial `Name(Serial_TX, SERIAL_RX);` Déclaration d'une liaison série
  - `Name` est le nom que vous souhaitez donner à cette fonction
- Broches
  - USB par défaut
- Fonctions associées
  - `Name.printf(char* chaine_a_envoyer);`
  - `Name.baud(int value);` change vitesse
- Paramètres
  - `%d %x %f \t \r \n`

```
#include "mbed.h"

Serial pc(SERIAL_TX, SERIAL_RX); // 9600 Baud
int8_t var = 0;
int main() {
    pc.printf("Bonjour Frederic\r\n");
    while(1) {
        var++;
        pc.printf("var = %d\n");
        wait(1);
    }
}
```



# Serial

## Contrôle d'une liaison série asynchrone

- Lecture d'un message envoyé sur le port série
- Cette information est asynchrone
  - `Name.getc()` est une fonction qui bloque le  $\mu$ C en attendant un message.
- Il faut détecter la présence d'un message entrant, avant de lire un 1<sup>er</sup> caractère.
  - `Name.readable()`
- Si le message est long, il faut mémoriser tout le message avant de le traiter.

```
#include "mbed.h"
Serial pc(SERIAL_TX, SERIAL_RX); // 9600 Baud

int main() {

    while(1) {
        if(pc.readable())
            c = pc.getc();
        // action
    }
}
```

# PwmOut

## Contrôle d'une ligne logique en mode PWM

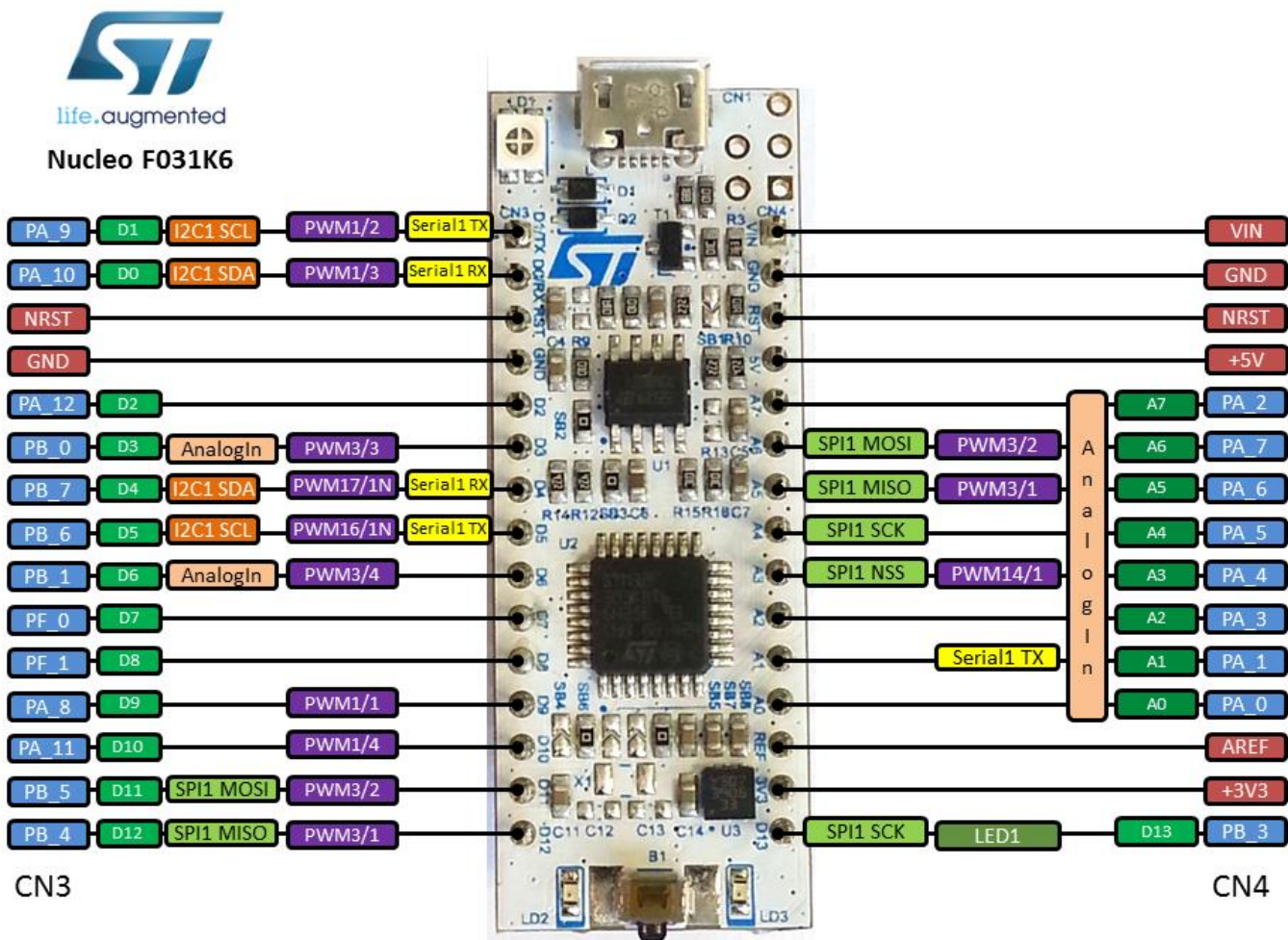
- PwmOut **Name(Pin);** Déclaration d'une ligne en sortie PWM
  - **Name** est le nom que vous souhaitez donner à cette broche
  - **Pin** est nom d'une des broches du  $\mu$ C, en mode PWM
- Broches
  - PB4, PB5 et PA11
- Fonctions associées
  - **Name.write(float value);** rapport cyclique 0 à 1
  - **Name.period(float value);** en s
  - **Name.period\_ms(int value);** en ms
  - **Name.pulsewidth(float value);** temps haut (en s)

```
#include "mbed.h"

PwmOut Moteur(PB_4);

int main() {
    while(1) {
        for(float p = 0.0f; p<1.0f; p+= 0.1f)
            {
                Moteur.write(p);
                wait(0.1);
            }
    }
}
```

# Broches de la carte Nucléo F031K6



# Résumé des périphériques disponibles

- GPIO (25 lignes E/S) avec interruptions externes
- USART (1)
- I2C (1)
- SPI (1)
- ADC (10 canaux) en 12-bit
- RTC

# Liens

- Mbed API
- Mbed home page
- Mbed library
- Mbed compiler
- Extra tutorials concerning Mbed and Nucleo boards
- Putty