

www.sylvainmahe.xyz

LE BLOG

de Sylvain Mahé

contact@sylvainmahe.xyz



Article: Sylvain Mahé

contact@sylvainmahe.xyz

[Retour](#)

[Suite](#)

Lire l'état d'un bouton avec GpioRead.h

La classe **GpioRead.h** permet de lire l'**état 0 ou 1** (0V ou +5V) d'un ou plusieurs ports de l'automate programmable avec la possibilité d'utiliser une **résistance de rappel interne au microcontrôleur**, et d'avoir un système **anti-rebonds** bien pratique pour filtrer les rebonds et autres parasites provoquées par les boutons et interrupteurs.

Gpio (pour General Purpose Input/Output ou entrée/sortie pour un usage général) est le nom que l'on donne aux ports d'entrée/sortie génériques (c'est-à-dire sans fonctions particulières) principalement dans le monde des microcontrôleurs.

Exemple d'utilisation de GpioRead.h:

```
#include "../module/1284p/GpioRead.h"

int main()
{
    GpioRead myButton = GpioRead (1, true, 10);

    while (true)
    {
        myButton.read();

        if (myButton.continuous == true)
        {
            //allumer une del
        }
        else
        {
            //éteindre une del
        }

        if (myButton.momentary == true)
        {
            //allumer une del pendant la durée de ce tour de boucle uniquement
        }
    }

    return 0;
}
```

Dans cet exemple, un objet **myButton** de type **GpioRead** est déclaré:

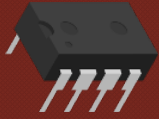
- Le 1er paramètre indique l'utilisation du port numéro **1** de l'automate programmable en entrée.
- Le 2ème paramètre **true** permet d'utiliser une **résistance de rappel interne**.

*Attention, ne connectez pas un bouton sans résistance de rappel externe à un port de l'automate programmable **sans avoir au préalable défini ce paramètre sur vrai** afin d'utiliser une résistance de rappel interne, dans le cas contraire **vous risquez de griller le microcontrôleur** !*

- Le 3ème paramètre **10** est le **temps en millisecondes** pour filtrer les rebonds.

*A noter que **10ms** est une valeur qui fonctionne bien pour filtrer la plupart des boutons. Plus votre bouton ou interrupteur sera d'une qualité médiocre, plus il faudra augmenter cette valeur.*

Plus loin cet objet **myButton** appelle la fonction **read** dans une boucle ce qui permet de lire l'état du port concerné.



www.sylvainmahe.xyz

LE BLOG

de Sylvain Mahé

contact@sylvainmahe.xyz



Retour

Suite

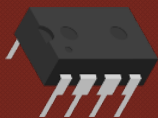
Cet état **0** ou **1** est stocké dans deux variables, **continuous** et **momentary**:

- continuous signifie que **tant que le bouton est appuyé, cette variable renvoie vrai**.
- momentary signifie que si le bouton est appuyé, **la variable renvoie vrai à la première lecture** (1er tour de boucle généralement), mais qu'elle **renvoie faux aux lectures suivantes** (aux tours de boucle suivants), en attendant que le bouton soit relâché puis par la suite appuyé une seconde fois.

Cet état momentané permet d'appeler des blocs de code une seule fois lorsqu'un bouton est pressé (sans être relâché immédiatement), cela peut être utile pour réaliser par exemple des incréments de valeurs, ou bien encore des signaux sonores lorsqu'on presse des boutons...

Bien entendu vous pouvez utiliser cette classe pour d'autres applications (lire autre chose que des boutons ou interrupteurs).

Ports des automates programmables concernés par la GPIO:



www.sylvainmahe.xyz

LE BLOG

de Sylvain Mahé

contact@sylvainmahe.xyz



[Retour](#)

[Suite](#)

Automate programmable MODULABLE M20:

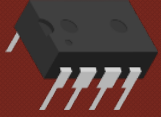
- Port 1 (PD0)
- Port 2 (PD1)
- Port 3 (PD2)
- Port 4 (PD3)
- Port 5 (PD4)
- Port 6 (PD5)
- Port 7 (PD6)
- Port 8 (PD7)
- Port 9 (PB0)
- Port 10 (PB1)
- Port 11 (PB2)
- Port 12 (PB3)
- Port 13 (PB4)
- Port 14 (PB5)
- Port 15 (PC0)
- Port 16 (PC1)
- Port 17 (PC2)
- Port 18 (PC3)
- Port 19 (PC4)
- Port 20 (PC5)

Automate programmable MODULABLE M32:

- Port 1 (PB0)
- Port 2 (PB1)
- Port 3 (PB2)
- Port 4 (PB3)
- Port 5 (PB4)
- Port 6 (PB5)
- Port 7 (PB6)
- Port 8 (PB7)
- Port 9 (PD0)
- Port 10 (PD1)
- Port 11 (PD2)
- Port 12 (PD3)
- Port 13 (PD4)
- Port 14 (PD5)
- Port 15 (PD6)
- Port 16 (PD7)
- Port 17 (PC0)
- Port 18 (PC1)
- Port 19 (PC2)
- Port 20 (PC3)
- Port 21 (PC4)
- Port 22 (PC5)
- Port 23 (PC6)
- Port 24 (PC7)
- Port 25 (PA7)
- Port 26 (PA6)
- Port 27 (PA5)
- Port 28 (PA4)
- Port 29 (PA3)
- Port 30 (PA2)
- Port 31 (PA1)
- Port 32 (PA0)

Récapitulatif des fonctions et variables de cette classe:

```
bool continuous = false;
bool momentary = false;
GpioRead (const unsigned char PIN, const bool PULL_UP_INTERNAL, const unsigned char DEB
void read();
```



www.sylvainmahe.xyz

LE BLOG

de Sylvain Mahé

contact@sylvainmahe.xyz



[Retour](#)

[Suite](#)

design du blog: sylvain mahé