

m5Stack Basic : API et programmes arduino

Boutons

Parametres

- read
- lastChange
- isPressed
- wasPressed
- pressedFor
- isReleased
- releasedFor
- wasReleased
- wasReleasefor

Lire (Read)

Syntaxe:

```
uint8_t read();
```

Description:

- Lire l'état du bouton

Paramètres:

- nul

Retour:

- uint8_t:
 - 1 : Appuyez sur le bouton
 - 0 : Relâchement du bouton

[Exemple0001.ino](#)

```
#include <M5Stack.h>

void setup() {
  M5.begin();
  M5.Lcd.println("Please pressed Button A.");
}

void loop() {
```

```
M5.Lcd.setCursor(0, 0);  
M5.Lcd.printf("Button A Status: %d ",M5.BtnA.read());  
}
```

dernier changement (lastChange)

Syntaxe:

```
uint32_t lastChange();
```

Description:

- Obtenir l'heure de la dernière mise à jour du statut

Paramètres:

- nul

Retour:

- uint32_t : Heure de la dernière mise à jour du statut ms

[Exemple0002.ino](#)

```
#include <M5Stack.h>  
  
void setup() {  
  M5.begin();  
  M5.Lcd.println("Please pressed Button A.");  
}  
  
void loop() {  
  M5.update();  
  M5.Lcd.setCursor(0, 0);  
  M5.Lcd.printf("The last change at %d ms /n",M5.BtnA.lastChange());  
}
```

est pressé (isPressed)

Syntaxe:

```
uint8_t isPressed();
```

Description:

- Déterminer si un bouton est enfoncé

Paramètres:

- nul

Retour:

- uint8_t:
 - 1 : Le bouton est enfoncé
 - 0 : Le bouton n'est pas enfoncé

[Exemple0003.ino](#)

```
#include <M5Stack.h>

void setup() {
  M5.begin();
  M5.Lcd.println("Please pressed Button A.");
}

void loop() {
  M5.update();
  M5.Lcd.setCursor(0, 0);
  if (M5.BtnA.isPressed()) {
    M5.Lcd.println("Button is Pressed.");
  }else{
    M5.Lcd.println("Button is Released.");
  }
  delay(20);
}
```

était pressé (wasPressed)

Syntaxe:

```
uint8_t wasPressed();
```

Description:

- Déterminez si un bouton est passé de relâché à enfoncé.

Paramètres:

- nul

Retour:

- uint8_t:
 - 1 : Bouton de déclenchement pour passer de l'état relâché à l'état enfoncé

- 0 : Le bouton non déclenché passe de l'état relâché à l'état enfoncé

Exemple04.ino

```
#include <M5Stack.h>

void setup() {
  M5.begin();
  M5.Lcd.println("Please pressed Button A.");
}

void loop() {
  M5.update();
  if (M5.BtnA.wasPressed()) {
    M5.Lcd.println("Button is pressed.");
  }
  delay(20);
}
```

presséPour (pressedFor)

Syntaxe:

```
uint8_t pressedFor(uint32_t ms);
```

Description:

- Déterminez si un bouton a été enfoncé pendant plus longtemps que le délai d'expiration spécifié.

Paramètres:

- uint32_t ms :
 - Délai d'attente d'appui sur le bouton ms

Retour:

- uint8_t:
 - 1 : Relâchement du bouton de déclenchement au-delà du délai d'attente
 - 0 : non déclenché

Exemple005.ino

```
#include <M5Stack.h>

void setup() {
  M5.begin();
```

```
M5.Lcd.println("Please pressed Button A.");
}

void loop() {
  M5.update();
  if (M5.BtnA.pressedFor(2000)) {
    M5.Lcd.println("Button A was pressed for more than 2 seconds.");
    delay(1000);
  }
}
```

est publié (isPressed)

Syntaxe:

```
uint8_t isPressed();
```

Description:

- Déterminer si un bouton est dans l'état relâché

Paramètres:

- nul

Retour:

- uint8_t:
 - 1 : Le bouton est relâché.
 - 0 : Les boutons ne sont pas relâchés.

[Exemple0006.ino](#)

```
#include <M5Stack.h>

void setup() {
  M5.begin();
}

void loop() {
  M5.update();
  if (M5.BtnA.isReleased()) {
    M5.Lcd.println("Button is released.");
  }else{
    M5.Lcd.println("Button is Pressed .");
  }
  delay(20);
}
```

publiéPour (pressedFor)

Syntaxe:

```
uint8_t pressedFor(uint32_t ms);
```

Description:

- Déterminer le délai d'expiration du relâchement du bouton

Paramètres:

- uint32_t ms :
 - Délai de relâchement du bouton ms

Retour:

- uint8_t:
 - 1 : Relâchement du bouton de déclenchement au-delà du délai d'attente
 - 0 : non déclenché

[Exemple007.ino](#)

```
#include <M5Stack.h>

void setup() {
  M5.begin();
}

void loop() {
  M5.update();
  M5.Lcd.setCursor(0, 0);
  if (M5.BtnA.releasedFor(2000)) {
    M5.Lcd.println("Button A was released for more than 2 seconds.");
    delay(1000);
  }else{
    M5.Lcd.println("Button A is pressed");
  }
}
```

a été libéré (wasReleased)

Syntaxe:

```
uint8_t wasReleased();
```

Description:

- Déterminer si un bouton est enfoncé et relâché.

Paramètres:

- nul

Retour:

- uint8_t:
 - 1 : Le bouton de déclenchement passe de l'état enfoncé à l'état relâché
 - 0 : Le bouton non déclenché passe de l'état enfoncé à l'état relâché

Exemple008.ino

```
#include <M5Stack.h>

void setup() {
  M5.begin();
  M5.Lcd.println("Please pressed Button A.");
}

void loop() {
  M5.update();
  if(M5.BtnA.wasReleased()) {
    M5.Lcd.println("Button is Released.");
  }
  delay(20);
}
```

étaitLibérépour (wasReleasefor)

Syntaxe:

```
uint8_t wasReleasefor(uint32_t ms);
```

Description:

- Déterminez si un bouton a été enfoncé pendant plus que le délai spécifié et déclenchez-le lorsque le bouton est relâché.

Paramètres:

- uint32_t ms :
 - Délai de relâchement du bouton ms

Retour:

- uint8_t:

- 1 : Relâchement du bouton de déclenchement au-delà du délai d'attente
- 0 : non déclenché

Exemple009.ino

```
#include <M5Stack.h>

void setup() {
  M5.begin();
  M5.Lcd.println("Please pressed Button A.");
}

void loop() {
  M5.update();
  if (M5.BtnA.wasReleasedfor(3000)) {
    M5.Lcd.println("OK");
  }
}
```

From: <http://chanterie37.fr/fablab37110/> - Castel'Lab le Fablab MJC de Château-Renault

Permanent link: <http://chanterie37.fr/fablab37110/doku.php?id=start:arduino:esp32:m5stack:programmes>

Last update: **2024/11/18 18:27**

