

# Button

- Functions
  - [read](#)
  - [lastChange](#)
  - [isPressed](#)
  - [wasPressed](#)
  - [pressedFor](#)
  - [isReleased](#)
  - [releasedFor](#)
  - [wasReleased](#)
  - [wasReleasefor](#)

## | read

---

機能です:

キー状態を読み取ります:0,離します;1,押します

原型関数です:

```
uint8_t read()
```

使用例です:

```
#include <M5Atom.h>

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

void loop() {
  Serial.println(M5.Btn.read()); //在Arduino打开串口监视器,波特率设置115200
 即可看到读取到的按键状态
  delay(20);
}
```

## | lastChange

---

原型関数です:

```
uint32_t lastChange()
```

機能です:

最後に状態が変化した時間に戻ります

**注意します:**

1. 戻る時間はAtomが初期化された時点からミリ秒単位で計っています。

使用例です:

```
#include <M5Atom.h>

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

void loop() {
  M5.update();
  Serial.printf("The last change at %d ms /n",M5.Btn.lastChange()); //
  串口输出按键状态最后一次发生变化的时间
}
```

## | isPressed

---

機能です:

キーを押す状態に戻ります:キーを押すならば、trueに戻ります;そうでなければfalseに戻ります

原型関数です:

```
uint8_t isPressed()
```

使用例です:

```
#include <M5Atom.h>

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

void loop() {
  M5.update();    //需添加 M5.update()才能读取到按键的状态,细节请见 System
  if (M5.Btn.isPressed()) {    //如果按键按下
    Serial.println("Button is pressed.");
  }
  delay(20);
}
```

## | wasPressed

---

機能です:

キーを押下した状態に戻ります:キーを押下すれば1回だけtrueを返します。

原型関数です:

```
uint8_t wasPressed()
```

使用例です:

```
#include <M5Atom.h>

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

void loop() {
  M5.update();
  if (M5.Btn.wasPressed()) { //如果按键按下
    Serial.println("Button is pressed.");
  }
  delay(20);
}
```

## pressedFor

機能です:

キーを押した状態に戻ります:キーを押して指定の時間を過ぎたら、trueに戻ります。そうでなければfalseに戻ります

原型関数です:

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

パラメータです	タイプです	記述します
ms	uint32_t	ボタンを押す時間(ミリ秒)です

使用例です:

```
#include <M5Atom.h>

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

void loop() {
  M5.update();
  if (M5.Btn.pressedFor(2000)) { //如果按键按下超过2秒
    Serial.println("Button A was pressed for more than 2 seconds.");
    delay(1000);
  }
}
```

## | isReleased

---

機能です:

キー解放状態に戻ります:キーが解放されれば、trueに戻ります;そうでなければfalseに戻ります

原型関数です:

```
uint8_t isPressed()
```

使用例です:

```
#include <M5Atom.h>

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

void loop() {
  M5.update();    //需添加M5.update()才能读取到按键的状态,细节请见System
  if (M5.Btn.isReleased()) {    //如果按键释放
    Serial.println("Button is released.");
  }
  delay(20);
}
```

## releasedFor

機能です:

キー解放状態に戻ります:キー解放が指定の時間を超えたら、trueに戻ります。そうでなければfalseに戻ります

原型関数です:

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

パラメータです	タイプです	記述します
ms	uint32_t	ボタンを押す時間(ミリ秒)です

使用例です:

```
#include <M5Atom.h>

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

void loop() {
  M5.update();
  if (M5.Btn.releasedFor(2000)) { //如果按键释放超过2秒
    Serial.println("Button A was released for more than 2 seconds.");
    delay(1000);
  }
}
```

## | wasReleased

---

機能です:

キー解放状態に戻ります:キーが解放された場合は一回だけtrueを返し、そうでなければfalseを返します

原型関数です:

```
uint8_t wasReleased()
```

使用例です:



```
#include <M5Atom.h>

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

void loop() {
  M5.update();
  if (M5.Btn.wasReleased()) { //如果按键释放
    Serial.println("Button is Released.");
  }
  delay(20);
}
```

## wasReleasefor

原型関数です:

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

機能です:

キーリリース状態に戻ります:キーを押下して指定の時間を過ぎてからリリースすると、1回だけtrueが戻ります。そうでなければfalseが戻ります。

パラメータです	タイプです	記述します
ms	uint32_t	ボタンを押す時間(ミリ秒)です

使用例です:

```
#include <M5Atom.h>

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

void loop() {
  M5.update();
  if (M5.Btn.wasReleasedfor(3000)) { //如果按键按下3s之后释放
    Serial.println("clear screen");
  }
}
```