

# ezButton

*Knihovna ezButton (easy button) je navržena tak, aby usnadnila používání nejen začátečníkům, ale i zkušeným uživatelům. Vytvořila ji společnost ArduinoGetStarted.com.*

*Knihovnu lze použít pro tlačítka, momentové spínače, přepínače, spínače s magnetickým kontaktem (dveřní senzor)...*

Začátečníci se při používání tlačítek obvykle potýkají s následujícími problémy:

- neošetřený vstup Arduina
- zákmity kontaktu
- detekce při sepnutí nebo rozpojení spínače
- obsluha více než jednoho spínače

## ***Základní vlastnosti knihovny***

- připojuje k určenému pin vnitřní pull-up rezistor, který zabraňuje nedefinované logické úrovni na pinu
- ošetřuje zákmity mechanických kontaktů spínačů, které mohou způsobit špatně dohledatelné chyby
- umožňuje vyvolat událost sepnutím nebo rozpojením kontaktu
- sčítá počet sepnutí kontaktu a ukládá ho do proměnné
- všechny funkce knihovny jsou neblokující

## Třída ezButton

### Konstruktor

#### ezButton()

---

Vytvoří novou instanci třídy ezButton, přiřazenou konkrétnímu pinu.

#### Syntaxe:

```
ezButton(pin);
```

#### Parametr:

`pin` (*const int*) – pin, na který bude spínač připojen.

#### Návratová hodnota:

Jméno nové instance třídy ezButton.

#### Příklad:

```
// jméno nové instance třídy ezButton je button  
ezButton button(7);
```

### Členské funkce (metody)

#### setDebounceTime()

---

Nastavuje dobu prodlevy pro odeznění zákmitů kontaktů spínače. Čas odezvy se projeví při použití funkcí `getState()`, `isPressed()`, `isReleased()` a `getCount()`.

#### Syntaxe:

```
button.setDebounceTime(time);
```

#### Parametr:

`time` (*unsigned long*) – doba pro odeznění zákmitů kontaktů spínače v milisekundách (ms).

## Návratová hodnota:

Funkce nic nevrací.

## Příklad:

```
#include <ezButton.h>
ezButton button(7);
// vytvoří novou instanci třídy ezButton a připojí ji k pinu 7

void setup()
{
  button.setDebounceTime(100);
  // čas odezvy je nastaven na 100 milisekund
}

void loop()
{
}
```

## getState()

---

Vrací stav tlačítka. Pokud je nastavena prodleva, vrátí se jeho stav až po prodlevě. Před použitím této funkce MUSÍTE nejprve volat funkci `loop()`.

## Syntaxe:

```
button.getState();
```

## Parametr:

Funkce nemá žádné parametry

## Návratová hodnota:

Stav spínače.

Protože knihovna ezButton používá vnitřní pull-up rezistor, bude jeho stav HIGH (log.1), když není sepnut, a LOW (log.0), když je sepnut. Pokud je nastavena doba prodlevy, je vráceným stavem stav po uplynutí prodlevy.

## Příklad:

```
#include <ezButton.h>
ezButton button(7);
// vytvoří novou instanci třídy ezButton a připojí ji k pinu 7

void setup()
```

```

{
}

void loop()
{
  button.loop();
  // nejprve je nutno volat funkci loop()
  int btnState = button.getState();
}

```

## getStateRaw()

---

Vrací aktuální stav spínače bez ohledu na nastavenou prodlevu. Je ekvivalentní funkci `digitalRead()`.

### Syntaxe:

```
button.getStateRaw();
```

### Parametr:

Funkce nemá žádné parametry

### Návratová hodnota:

Okamžitý stav spínače bez ohledu na nastavenou prodlevu.

### Příklad:

```

#include <ezButton.h>
ezButton button(7);
// vytvoří novou instanci třídy ezButton a připojí ji k pinu 7

void setup()
{
}

void loop()
{
  int btnState = button.getStateRaw();
}

```

## isPressed()

---

Kontroluje, zda je v době jejího volání spínač sepnut. Před použitím této funkce **MUSÍTE** zavolat funkci `loop()`.

## Syntaxe:

```
button.isPressed();
```

## Parametr:

Funkce nemá žádné parametry

## Návratová hodnota:

**TRUE** – pokud je spínač sepnut

**FALSE** – v opačném případě

## Příklad:

```
#include <ezButton.h>  
ezButton button(7);  
// vytvoří novou instanci třídy ezButton a připojí ji k pinu 7  
  
void setup()  
{  
  Serial.begin(115200);  
}  
  
void loop()  
{ button.loop();  
// nejprve je nutno volat funkci loop()  
  if(button.isPressed()) Serial.println("Spínač je sepnut");  
}
```

## isReleased()

---

*Kontroluje, zda je zda je v době jejího volání spínač rozpojen. Před použitím této funkce MUSÍTE zavolat funkci `loop()`.*

## Syntaxe:

```
button.isReleased();
```

## Parametr:

Funkce nemá žádné parametry

## Návratová hodnota:

**TRUE** – pokud je spínač sepnut

**FALSE** – v opačném případě

## Příklad:

```
#include <ezButton.h>
ezButton button(7);
// vytvoří novou instanci třídy ezButton a připojí ji k pinu 7

void setup()
{
  Serial.begin(115200);
}

void loop()
{ button.loop();
  // nejprve je nutno volat funkci loop()
  if(button.isReleased()) Serial.println("Spínač je sepnut");
}
```

## setCountMode()

---

*Nastavuje režim čítače.*

### Syntaxe:

```
button.setCountMode(mode);
```

### Parametr:

**mode** (*int*) – režim čítání.

Povolené hodnoty parametru jsou:

**COUNT\_FALLING** (přičítá při sepnutí spínače)

**COUNT\_RISING** (přičítá při rozpojení spínače)

**COUNT\_BOTH** (přičítá při sepnutí i rozpojení spínače)

### Návratová hodnota:

Funkce nic nevrací

## Příklad:

```
#include <ezButton.h>
ezButton button(7);
// vytvoří novou instanci třídy ezButton a připojí ji k pinu 7

void setup()
{
  Serial.begin(115200);
  button.setDebounceTime(100);
}
```

```

    // čas odezvy je nastaven na 100 milisekund
    button.setCountMode(COUNT_FALLING);
}

void loop()
{
    button.loop();
    // nejprve je nutno volat funkci loop()
    unsigned long count = button.getCount();
    Serial.println(count);
}

```

## getCount()

---

Vrací hodnotu proměnné *count*. Před použitím této funkce **MUSÍTE** zavolat funkci *loop()*.

### Syntaxe:

```
button.getCount();
```

### Parametr:

Funkce nemá žádné parametry

### Návratová hodnota:

Počet sepnutí.

Pokud je nastaven čas prodlevy, je vracená hodnota hodnotou po uplynutí prodlevy.

### Příklad:

```

#include <ezButton.h>
ezButton button(7);
// vytvoří novou instanci třídy ezButton a připojí ji k pinu 7

void setup()
{
    Serial.begin(115200);
    button.setDebounceTime(100);
    // čas odezvy je nastaven na 100 milisekund
    button.setCountMode(COUNT_FALLING);
}

void loop()
{
    button.loop();
}

```

```
    // nejprve je nutno volat funkci loop()
    unsigned long count = button.getCount();
    Serial.println(count);
}
```

## resetCount()

---

Nastaví čítač počtu sepnutí spínače na 0.

### Syntaxe:

```
button.resetCount();
```

### Parametr:

Funkce nemá žádné parametry

### Návratová hodnota:

Funkce nic nevrací

### Příklad:

```
#include <ezButton.h>
ezButton button(7);
// vytvoří novou instanci třídy ezButton a připojí ji k pinu 7

void setup()
{
    Serial.begin(115200);
    button.setCountMode(COUNT_FALLING);
}

void loop()
{
    button.loop();
    // nejprve je nutno volat funkci loop()
    unsigned long count = button.getCount();
    if(count >= 100) button.resetCount();
}
```

## loop()

---

Aktualizuje stav spínače. MUSÍ být volána před použitím funkcí `getState()`, `isPressed()`, `isReleased()` a `getCount()`.



## Syntaxe:

```
button.loop();
```

## Parametr:

Funkce nemá žádné parametry

## Návratová hodnota:

Funkce nic nevrací

## Příklad:

```
#include <ezButton.h>  
ezButton button(7);  
// vytvoří novou instanci třídy ezButton a připojí ji k pinu 7  
  
void setup()  
{  
  Serial.begin(115200);  
  button.setDebounceTime(100);  
  // čas odezvy je nastaven na 100 milisekund  
}  
  
void loop()  
{  
  button.loop();  
  // nejprve je nutno volat funkci loop()  
  int btnState = button.getState();  
  Serial.println(btnState);  
  if(button.isPressed()) Serial.println("Spínač je sepnut");  
  if(button.isReleased()) Serial.println("Spínač je rozpojen");  
}
```