

Przed wgraniem szkicu do Arduino upewnij się, czy masz zainstalowane poniższe biblioteki:

**Mysensors v 2.1.1**

**Debouncer**

Jeśli nie – szkic się nie skompiluje. Jak je zainstalować omówiłem w odcinku.

## Sterowanie włącznikiem fizycznym (przyciskiem)

Arduino może reagować na stan niski (GND) lub wysoki. Standardowo zmienia wartość po otrzymaniu masy (GND).

Zmieniamy to w linijce 104:

```
if(value == LOW){    ///stan niski  
lub  
if(value == HIGH){  ///stan wysoki
```

## Sterowanie przekaźnikiem

Za pomocą szkicu możemy sterować przekaźnikami aktywując je stanem niskim (GND) lub wysokim (+5V). Dlatego w zależności od użytych przekaźników należy zmodyfikować linię kodu numer 59 i 60.

Odpowiednio dla stanu niskiego:

```
#define RELAY_ON 0  
#define RELAY_OFF 1
```

I wysokiego:

```
#define RELAY_ON 1  
#define RELAY_OFF 0
```

Standardowo arduino podaje stan wysoki (+5V) w celu aktywacji przekaźnika.

## Problem z pinem 3

Pin 3 w tych arduino jest dosyć problematyczny, dlatego startujemy z pinu numer 4. Jeśli chcielibyśmy użyć również pinu numer 3 – szkic będzie działał, ale przy braku zasilania nie będzie pobierał dla tego pinu stanu z pamięci eeprom (przy każdym restarcie pin 3 będzie podawał stan wysoki, niezależnie od stanu z przed odłączenia zasilania).

## Zmiany w przypadku użycia więcej niż 1 przycisku / przekaźnika

Opiszę w skrócie dodanie kolejnego przekaźnika z przyciskiem. **Jeśli przekaźnik ma być używany bez przycisku (sterowany tylko z domoticza), to w poniższych krokach (w kroku 2) należy użyć numeru pinu, do którego nie jest nic podpięte** (ale nie może być to pin 0 i 1, bo odpowiada za komunikację z raspberry, nie powinien być to też żaden z portów od interfejsów SPI, I2C czy pokładowych i dołączanych diod statusu). Jeśli chcemy użyć kolejnego przekaźnika bez fizycznego przycisku przypisujemy mu w kroku 2 ten sam pin, co w przypadku poprzednika.

**Żeby zachować maks. zgodność z kodem utworzonym powyżej, zaleca się przypinanie pinów po kolei (np. przekaźnik pierwszy – pin nr 4, drugi – pin nr 5, trzeci – pin nr 6) i w taki sposób będzie to przedstawione poniżej.**

**Kolorami oznaczono zmiany w stosunku do 1 sensora, na które trzeba zwrócić koniecznie uwagę, bo pominięcie choć jednej spowoduje błędne bądź irracjonalne działanie programu!**

**Wszystkie poniższe kroki (oprócz 1, w którym zmieniamy istniejącą linię) polegają na duplikacji pewnych obszarów programu.** Jeśli chcemy dodać więcej niż jeden przekaźnik, należy zastosować pętle, by uniknąć nadmiarowego kodu, ale po przestudiowaniu poniższych instrukcji powinno być to dużo łatwiejsze do zrozumienia i wykonania.

### 1. Zwiększenie definicji liczby przekaźników o 1:

Zamiast

```
#define NUMBER_OF_RELAYS 1
```

Wstawić

```
#define NUMBER_OF_RELAYS 2
```

### 2. Powielenie linii

```
#define BUTTON_PIN A1
```

dla przycisku nowego przekaźnika, np w sposób:

```
#define BUTTON2_PIN A2
```

(gdzie **A2** – pin z przyciskiem sterujący nowo dodawanym relayem)

### 3. Powielenie linii

```
#define RELAY_1 4
```

dla nowego przekaźnika, np. w sposób:

```
#define RELAY_2 5
```

(gdzie 5 – pin sterujący nowo dodawanym relayem)

### 4. Dodanie debouncera dla drugiego przycisku poprzez powielenie linii:

```
Bounce debouncer = Bounce();
```

Np.:

```
Bounce debouncer2 = Bounce();
```

#### 5. Powielenie linii ustawiającej tryb pinu:

```
pinMode(BUTTON_PIN, INPUT_PULLUP);
```

dla przycisku nowego przekaźnika, np (bazując na nazwie stałych z poprzednich punktów):

```
pinMode(BUTTON2_PIN, INPUT_PULLUP);
```

#### 6. Powielenie linii dla drugiego przycisku, czyli dodanie linii:

```
debouncer2.attach(BUTTON2_PIN);
```

```
debouncer2.interval(5);
```

w ciele funkcji setup()

7. Jeśli numerowaliśmy piny od przycisków rosnąco (np. przekaźnik pierwszy – pin nr 4, drugi – pin nr 5, trzeci – pin nr 6) to funkcje presentation() i before() pozostawia się bez zmian.

#### 8. Powielenie linii:

```
MyMessage msg(1, V_LIGHT);
```

dla nowego przekaźnika, np.:

```
MyMessage msg2(2, V_LIGHT);
```

#### 9. Powielenie linii

```
if (debouncer.update()) {  
  int value = debouncer.read();  
  if(value == LOW){  
    saveState(1, !loadState(1));  
    digitalWrite(RELAY_1, loadState(1)?RELAY_ON:RELAY_OFF);  
    send(msg.set(loadState(1)));  
  }  
}
```

np. (stosując nazewnictwo zmiennych i stałych przyjęte powyżej)

```
if (debouncer2.update()) {  
  int value2 = debouncer2.read();  
  if(value2 == LOW){  
    saveState(2, !loadState(2));  
    digitalWrite(RELAY_2, loadState(2)?RELAY_ON:RELAY_OFF);  
    send(msg2.set(loadState(2)));  
  }  
}
```



Elektryka to nie żarty, **230V może zabić**. Jeśli nie masz odpowiednich uprawnień lub nie posiadasz wystarczającej wiedzy - nie wykonuj omawianych w odcinku czynności. Nie ponoszę żadnej odpowiedzialności za działania osób powielających czynności przedstawione w filmie.