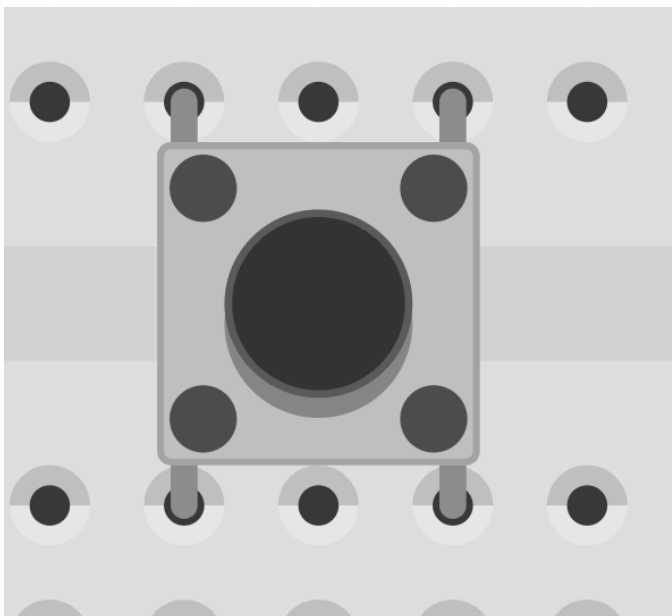


Przycisk monostabilny

Czyli jaki?

Przycisk monostabilny (inaczej zwany przyciskiem chwilowym) to rodzaj przycisku, który **pozostaje w jednym stanie** (zazwyczaj otwartym, czyli “wyłączonym”) i **zmienia swój stan tylko wtedy, gdy jest wciśnięty**. Gdy puścisz przycisk, wraca on do swojego pierwotnego stanu. Dlatego mówi się, że ma tylko jeden stabilny stan – stan wyłączony (otwarty).



Budowa przycisku monostabilnego

1. Obudowa: Przycisk jest zamknięty w obudowie, która utrzymuje wszystkie jego części wewnętrzne. Obudowa chroni mechanizm i zapewnia, że użytkownik może wygodnie naciskać przycisk.

2. Dwa styki: Wewnątrz przycisku znajdują się dwa metalowe elementy zwane stykami. Jeden z nich jest podłączony do źródła zasilania (np. pin Arduino lub inny obwód), a drugi do elementu, który ma zostać aktywowany (np. masa lub inny punkt obwodu). Gdy te dwa styki zostaną połączone, zamykają obwód, co pozwala prądowi przepływać.

3. Mechanizm sprężynowy: Kluczowym elementem przycisku monostabilnego jest sprężyna. Sprężyna utrzymuje przycisk w jego stabilnym stanie – zazwyczaj otwartym (czyli takim, w którym obwód jest przerwany). Sprężyna sprawia, że po puszczeniu przycisku wraca on do swojego pierwotnego stanu.

Jak działa przycisk monostabilny?

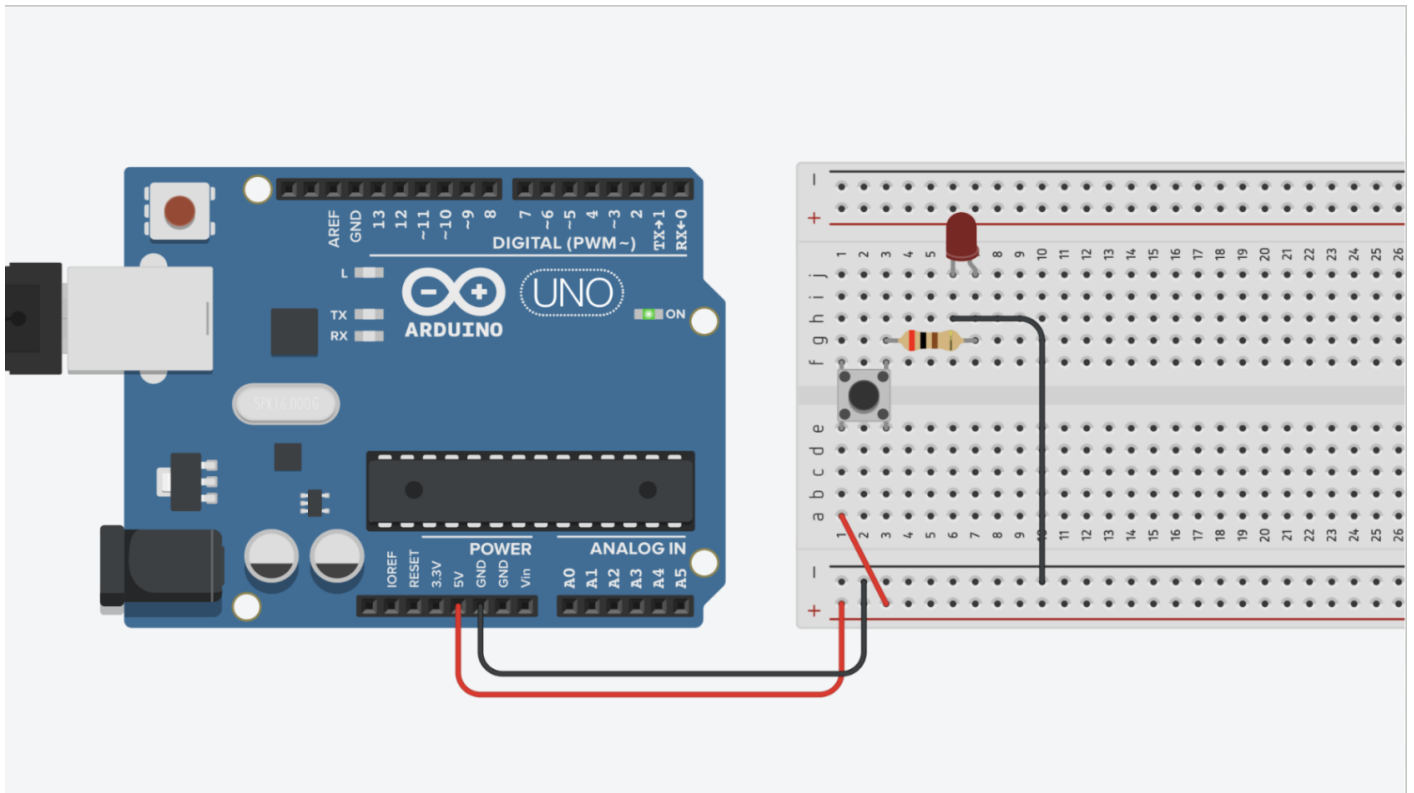
- - Stan spoczynku (brak nacisku): Kiedy przycisk nie jest wciśnięty, obwód jest przerwany, co oznacza, że prąd nie płynie. Mówimy, że obwód jest otwarty.
 - Stan aktywacji (nacisk): Kiedy przycisk jest wciśnięty, zamyka on obwód i pozwala na przepływ prądu. W tym momencie obwód jest zamknięty.

Po puszczeniu przycisku, automatycznie wraca on do stanu otwartego, przerywając obwód. Stąd nazwa "monostabilny", ponieważ tylko jeden stan (otwarty) jest stabilny bez zewnętrznej ingerencji.

Schemat dla Arduino

Schemat dla włącznika działającego z LEDem bez użycia kodu.

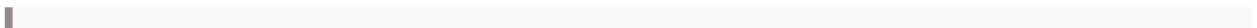
1. 1x przycisk monostabilny
2. 1x LED
3. rezystor 180 Ohm
4. Arduino UNO
5. 4x Jumpery



Jak działa powyższy schemat:

1. Stan początkowy (przycisk nie wciśnięty):
 - W stanie spoczynku (kiedy przycisk nie jest wciśnięty), styki wewnątrz przycisku są rozłączone.
 - Oznacza to, że obwód jest otwarty, więc prąd nie przepływa.
 - W takim stanie prąd nie może przejść przez przycisk, ponieważ nie ma fizycznego połączenia między stykami.
2. Wciśnięcie przycisku:
 - Kiedy naciskasz przycisk, mechanizm sprężynowy pozwala, aby część przycisku przemieszczała się w dół.
 - Podczas tego ruchu, styki wewnątrz przycisku łączą się, zamykając obwód.
 - Gdy styki zostaną połączone, prąd może przepływać przez przycisk, co oznacza, że urządzenie podłączone do obwodu (np. dioda LED lub silnik) zostaje aktywowane.
3. Po zwolnieniu przycisku:
 - Gdy puścisz przycisk, sprężyna wewnątrz automatycznie powraca go do pierwotnej pozycji.
 - Styki wewnątrz przycisku rozłączają się, co powoduje przerwanie obwodu.
 - Prąd przestaje płynąć, a urządzenie podłączone do obwodu przestaje działać (np. dioda LED gaśnie).

Schemat dla przycisku podającym sygnał na pin 8 wywołującym



1. 1x przycisk monostabilny
2. rezystor 10 kOhm
3. Arduino UNO
4. 4x Jumpery

Dlaczego jest nam potrzebny rezystor o wartości 10 kOhm? - Pulldown

Rezystor typu **“pull-down”** w układzie z przyciskiem monostabilnym jest potrzebny, aby zapewnić stabilny stan logiczny (niski - LOW) na wejściu cyfrowym Arduino, gdy przycisk nie jest wciśnięty. Pomaga to uniknąć niepożądanych zakłóceń, które mogą prowadzić do przypadkowych odczytów sygnału.

Problem bez rezystora pull-down:

Jeśli podłączysz przycisk bez rezystora typu “pull-down”, pin wejściowy Arduino może znajdować się w tzw. stanie pływającym (ang. “floating”) – oznacza to, że pin nie jest ani podłączony do napięcia, ani do masy. W takim stanie pin może losowo odbierać zakłócenia z otoczenia, co może prowadzić do nieprawidłowych odczytów sygnału.

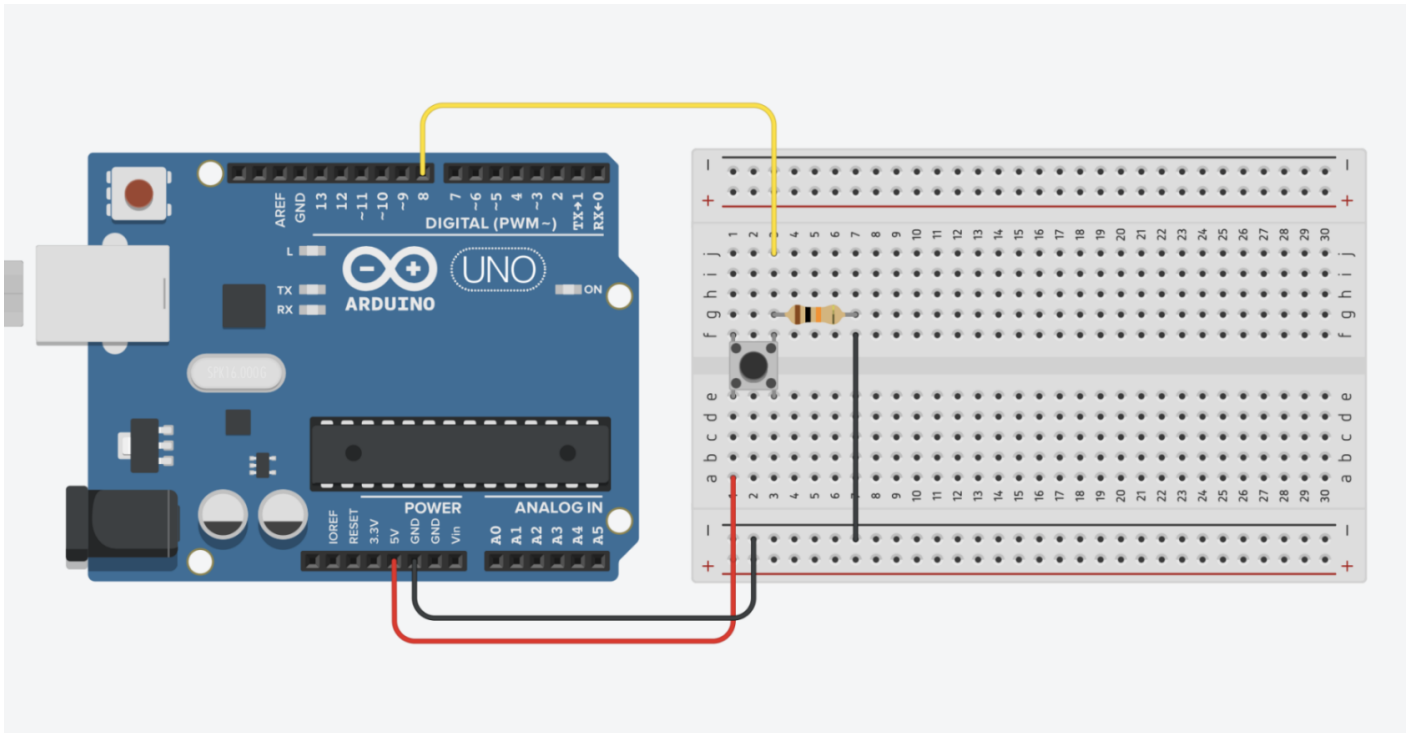
Gdy przycisk nie jest wciśnięty, Arduino może nie wiedzieć, czy pin jest w stanie niskim (LOW), czy wysokim (HIGH), co prowadzi do przypadkowych odczytów. W rezultacie program może błędnie reagować, np. myśląc, że przycisk został wciśnięty, gdy w rzeczywistości tak się nie stało.

Jak działa rezystor pull-down?

Rezystor pull-down zapewnia, że pin jest zawsze podłączony do masy (GND), gdy przycisk nie jest wciśnięty. Dzięki temu na pinie wejściowym panuje stabilny stan niski (LOW). Kiedy naciskasz przycisk, obwód jest zamykany i na pinie pojawia się stan wysoki (HIGH).

Alternatywa - rezystor pull-up:

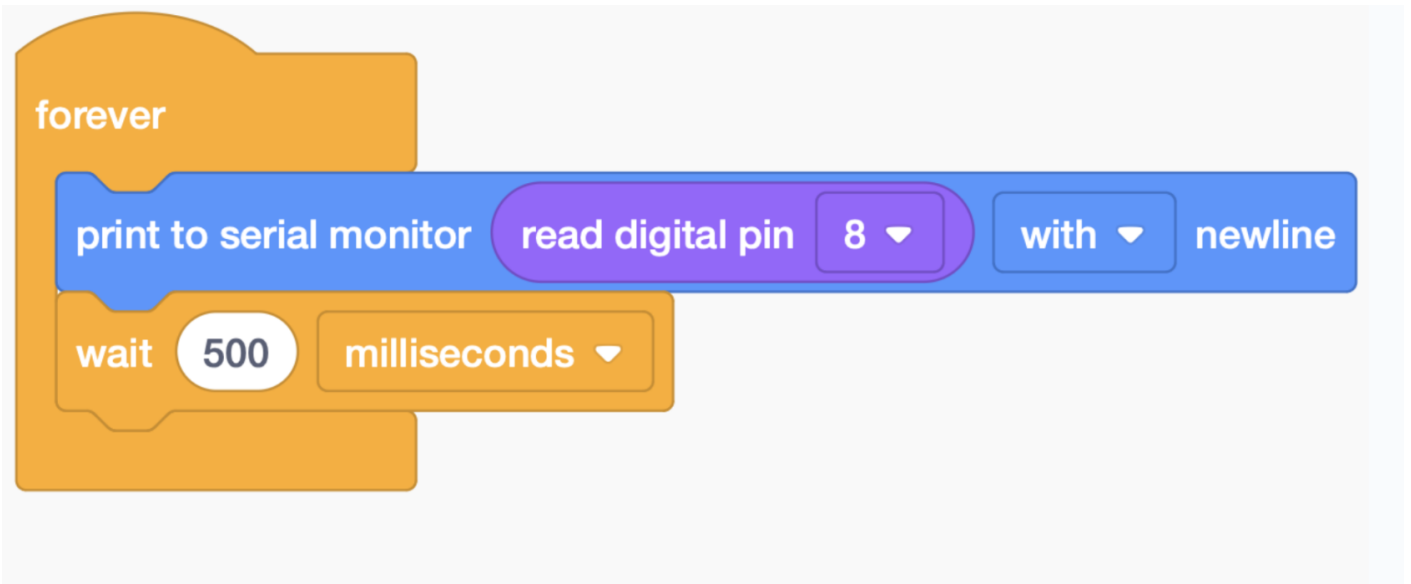
W Arduino często zamiast rezystora typu pull-down używa się wewnętrznego rezystora pull-up, który jest wbudowany w mikrokontroler. W takim przypadku pin domyślnie jest podciągnięty do stanu wysokiego (**HIGH**), a gdy przycisk jest wciśnięty, pin zostaje połączony z masą, co daje stan niski (**LOW**). Używanie **pinMode(pin, INPUT_PULLUP)** aktywuje ten wewnętrzny rezystor



```
// void setup()
{
  pinMode(8, INPUT);
  Serial.begin(9600);
}

void loop()
{
  Serial.println(digitalRead(8));
  delay(500); // Wait for 500 millisecond(s)
}
```

Lub kod z Tinkercada:



Wersja #3

Utworzono 2024-09-27 12:22:59 UTC przez Maciek Naskret

Zaktualizowano 2024-09-28 10:00:16 UTC przez Maciek Naskret