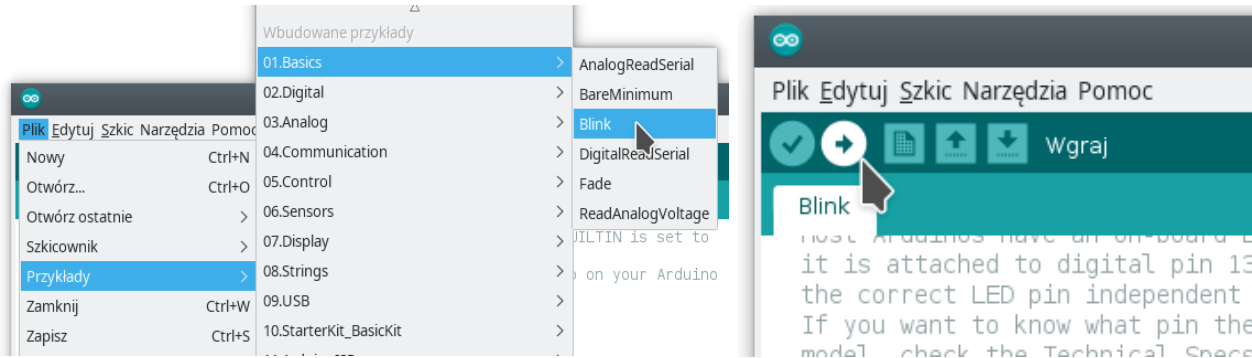


# A1. Porty we/wy

UWAGA! Po zajęciach realizujemy schemat z końca instrukcji.

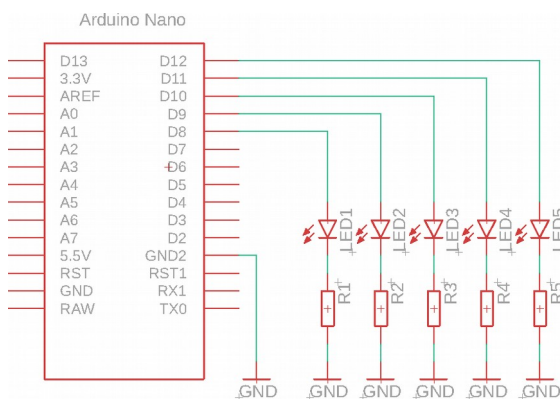
**Zadanie A1.1** Po uruchomieniu Arduino IDE oraz podłączeniu arduino, z menu Plik, wybrać Przykłady → 01.Basics → Blink. Przeanalizować program, zmienić wystąpienia `LED_BUILTIN` w funkcjach `setup()` oraz `loop()` na liczbę 13. Następnie wgrać program na arduino.



## Ustawianie/odczytywanie portów

- |  |  |
|--|--|
| <code>pinMode(port, STAN);</code>      | - ustawia <i>portu</i> jako wejście (INPUT) lub wyjście (OUTPUT) |
| <code>digitalRead(port);</code>        | - odczytuje wartość (0 lub 1) z <i>portu</i>                     |
| <code>digitalWrite(port, STAN);</code> | - ustawia <i>port</i> , na logiczne 0 lub 1 (5V)                 |

**Zadanie A1.2** Podłączyć do pinów D8, D9, D10, D11 oraz D12 diody wg poniższego schematu:



- Co należy zmienić w programie, aby zamiast diody podłączonej do D13 mrgała dioda na D12?
- Zmodyfikować program w taki sposób, by diody:
  - mrgały naraz,
  - mrgały jedna po drugiej,
  - mrgały tam i z powrotem.
- Ustawić **dwie** diody aby mrgały z różną częstotliwością (\*), tj.:  
Np. jedna dioda zapala się na 1s, gaśnie na 1s, druga – zapala się na 0.5s i gaśnie na 0.5s.

## Łączenie warunków

Jeśli wewnątrz funkcji if (lub innej warunkowej) chcemy umieścić więcej niż jeden warunek:

`(digitalRead(3) || digitalRead(4))` – oznacza prawdę, jeśli na wejściu 3 LUB 4 jest logiczne 1 (PRAWDA)

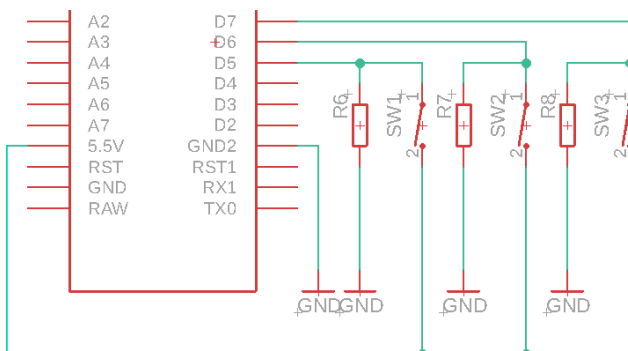
`(digitalRead(3) && digitalRead(4))` – oznacza prawdę, jeśli na wejściu 3 ORAZ 4 jest logiczne 1 (PRAWDA)

## Instrukcja warunkowa `if/else/else if`

Instrukcja warunkowa `if` może być rozbudowana o `else`, którego ciało wykonywane jest gdy poprzednie warunki są niespełnione lub `else if`(warunek), które wykonywane jest gdy wcześniejsze warunki są niespełnione ORAZ spełniony jest *nowy* warunek.

```
if(warunek1){  
    //wykonywane, gdy spełniony warunek1  
}else if(warunek2){  
    //wykonywane, gdy niespełniony warunek 1 i spełniony warunek2  
}else{  
    //wykonywane, gdy niespełnione wcześniejsze warunki  
}
```

**Zadanie A1.3** Rozbudować układ o trzy przyciski (do D5, D6 oraz D7) z rezystorami ściągającymi (10kΩ), wg schematu:



- Ustawić odpowiednie porty jako wejścia.
- Napisać program, który:
  - po wciśnięciu jednego z przycisków zapala odpowiadającą diodę,
  - jeśli zostaną wciśnięte dwa przyciski (konkretne, następnie dowolne) – czwarta dioda, jeśli wszystkie – piąta dioda.

## Zmienne

Zmienne służą do wygodnego zapamiętywania danych programu pod jakąś nazwą. Ponieważ są różne rodzaje danych, są też różne rodzaje zmiennych. Dwa podstawowe (dla nas) typy to:

- `int` – ang. „integer” – liczba całkowita, przechowuje liczby od -32768 do 32767]
- `bool` – ang. „boolean” – typ logiczny, przechowuje wartości prawda (true, 1) lub fałsz (false, 0). Funkcja, która zwraca wartość tego typu to np. `digitalRead()`.

Interesującym nas typem zmiennej jest zmienna *globalna*, będąca dostępną w każdym miejscu w programie. Zmienne globalne deklaruje się na początku programu (przed `void setup()`) w poniższy sposób:

```
typ_zmiennej nazwa_zmiennej;
```

lub, gdy chcemy od razu zadeklarować jej wartość:

```
typ_zmiennej nazwa_zmiennej = wartość;
```

Przykładowo deklaracja: `int dioda5 = 12;` sprawi, że wszędzie gdzie w programie użyjemy nazwy `dioda5`, komputer będzie wiedział, że mieliśmy na myśli liczbę 12.

**Zadanie A1.4** W programie z zadania A1.3 zastąpić numery portów do których podłączone są diody oraz przyciski bardziej „intuicyjnymi” zmiennymi.

## Obliczenia na zmiennych

Obliczenia na zmiennych zachowują się tak jak uniwersalne znaki obliczeń: + (dodawanie), - (odejmowanie), \* (mnożenie), / (dzielenie), % (reszta z dzielenia). Znak równości (=) oznacza przypisanie „tego co z prawej” do tego co „z lewej” – w przeciwieństwie do porównania (==). Np.:

```
zmienna8 = zmienna5 + 3;
```

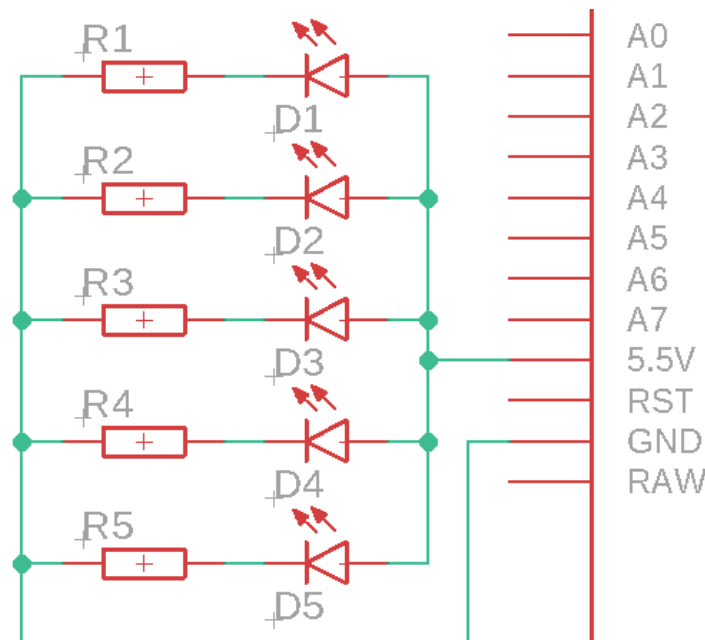
Jeśli zmienna `zmienna5` miała wartość 5, to po wykonaniu tej komendy, `zmienna8` będzie przechowywała wartość 8. W ów czas komenda `digitalWrite(zmienna8, HIGH)`; ustawi port D8 w stan wysoki (i np. zapali diodę do niego podpiętą).

**Zadanie A1.5** Napisać program, który po włączeniu arduino zapali diodę trzecią, następnie, gdy zostanie wciśnięty przycisk 1, gaśnie *obecna* dioda i zapala następna. Po wciśnięciu przycisku 2, gaśnie *obecna* i zapala *następna*. Czyli:

Zdefiniować **nową** zmienną typu `int` o dowolnej nazwie. Zwiększyć wartość *zmiennej* o 1, gdy zostanie wciśnięty przycisk 1 oraz zmniejszyć wartość *zmiennej*, jeśli zostanie wciśnięty przycisk 2. W zależności od tego, jaką wartość przyjmuje *zmienna* zapalić odpowiednią diodę (i zgasić inne).

**Zadanie dodatkowe** Funkcja `millis()` zwraca zmienną `unsigned long` zawierającą czas od uruchomienia arduino w milisekundach. Wykonać zadanie A1.2, punkt z gwiazdką bez użycia funkcji `delay()`.

## Schemat do zrealizowania po zajęciach



Źródła:

Autodesk Eagle, Version 9.3.2, © 2019 Autodesk, Inc. All rights reserved  
[http://akademia.nettigo.pl/zmienne\\_podstawy\\_jezyka\\_arduino/](http://akademia.nettigo.pl/zmienne_podstawy_jezyka_arduino/)

Patryk Król  
v1.9