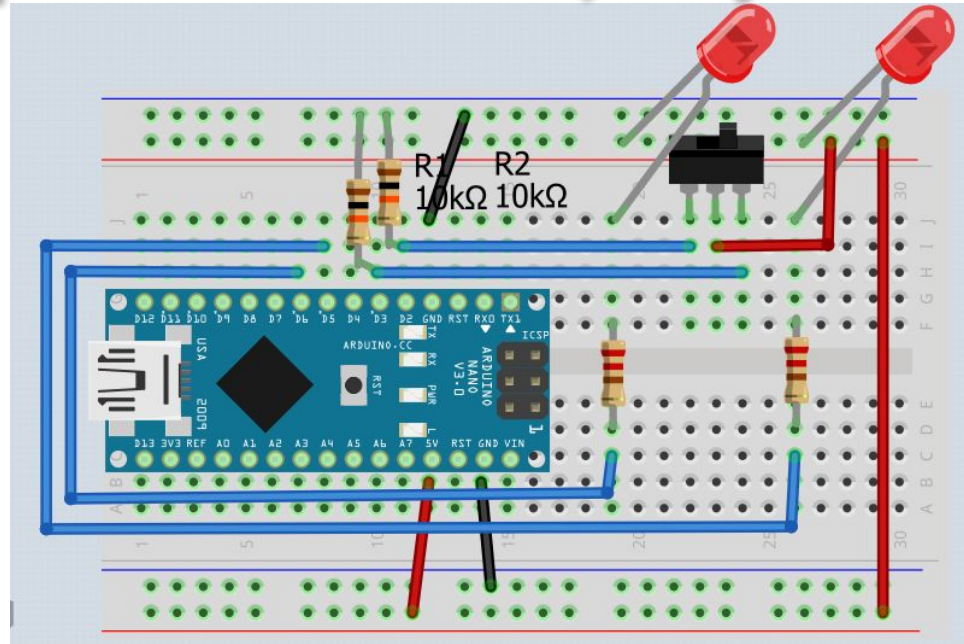
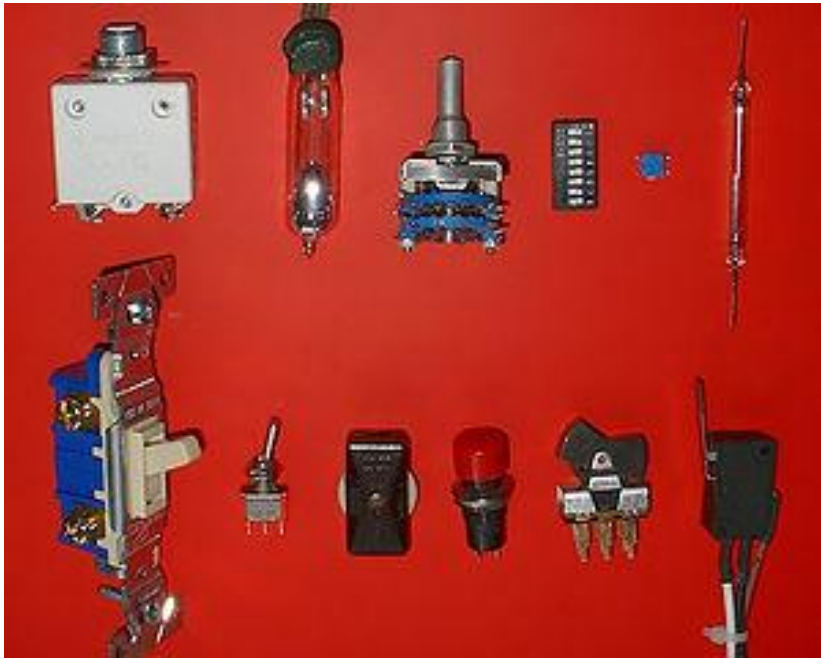


# Лекція 9 Перемикач, цикл while

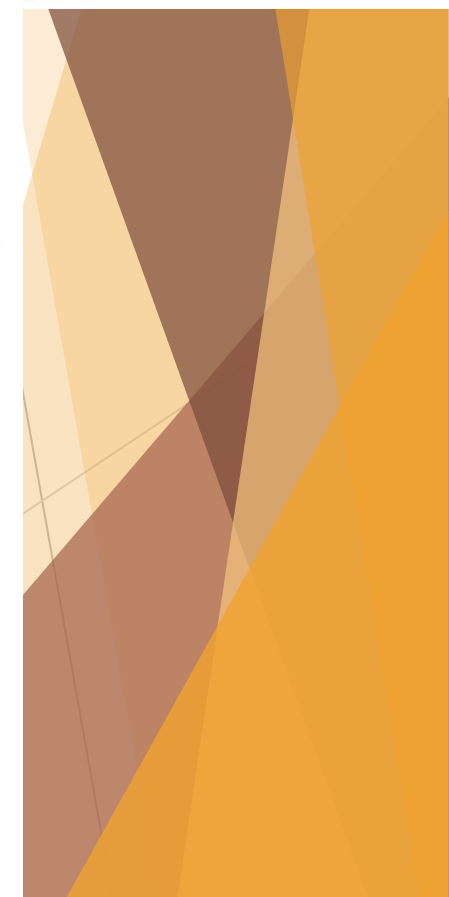
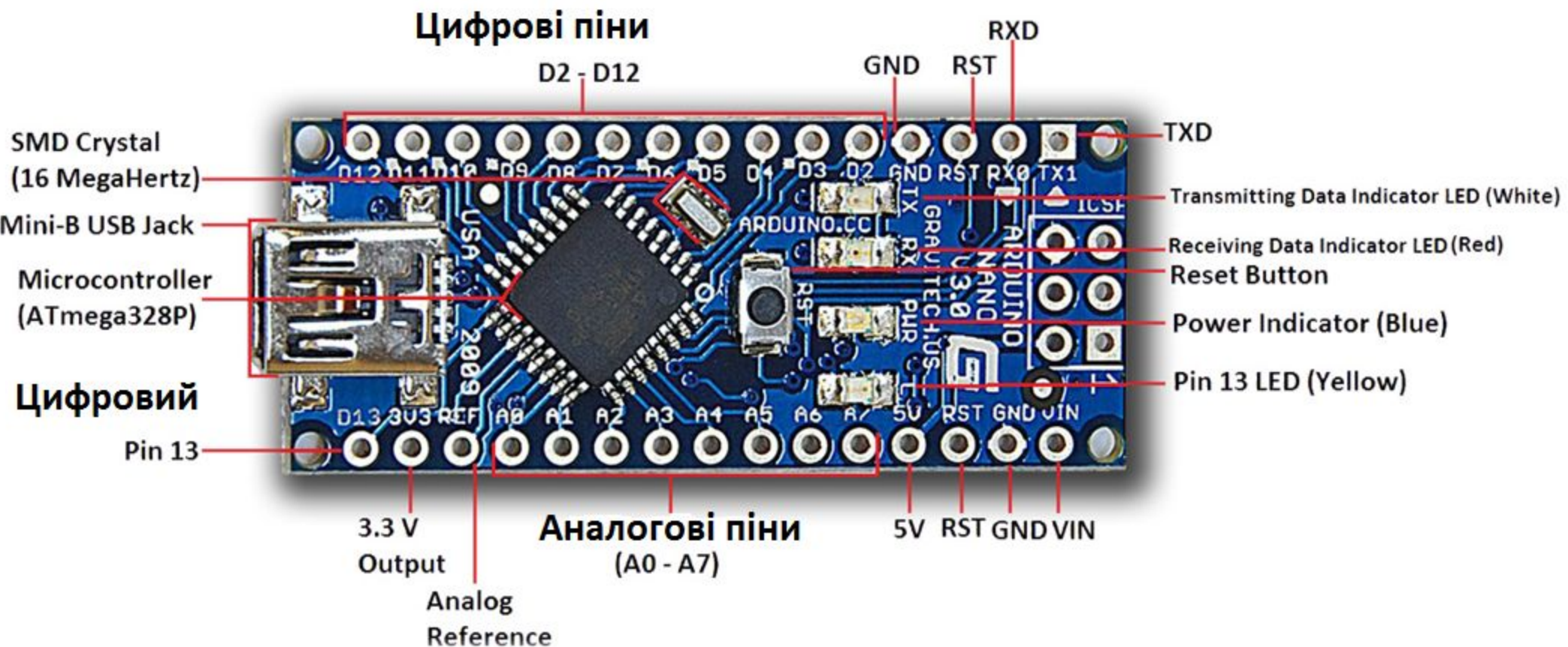
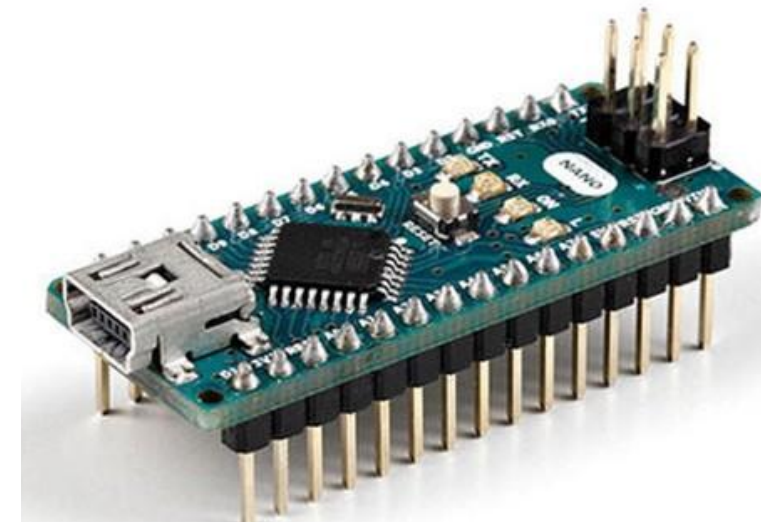
## while



Згадуємо Arduino  
Arduino -це  
Мікроконтролерний пристрій,  
який можна програмувати за  
допомогою комп'ютера.



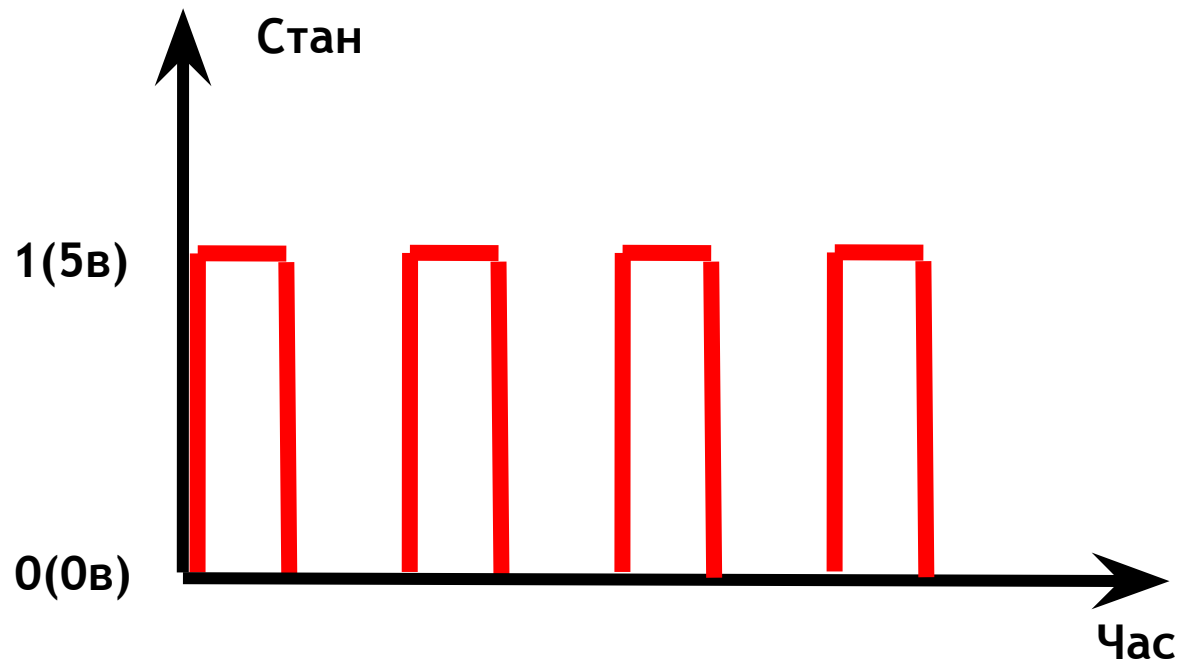
# Arduino NANO

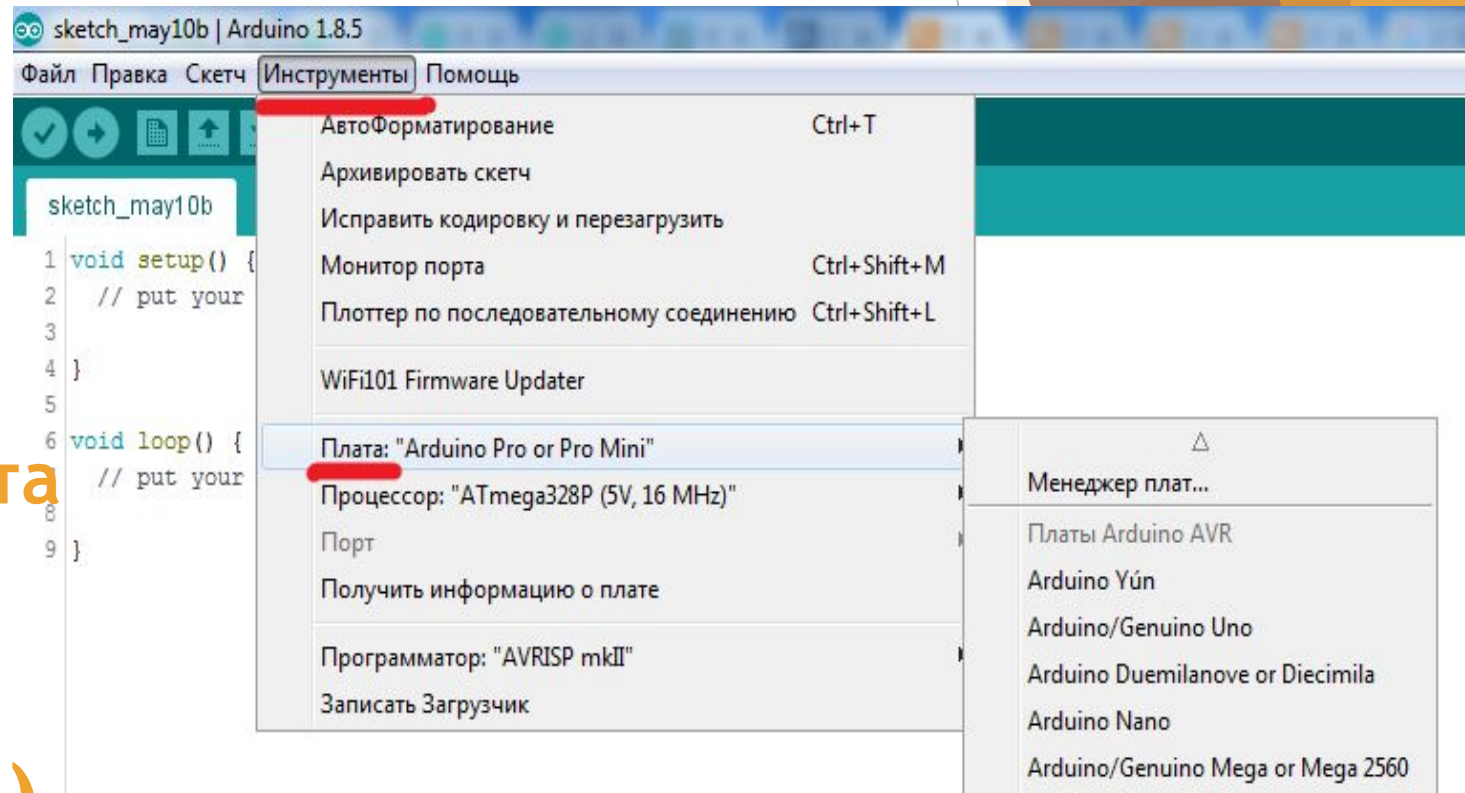
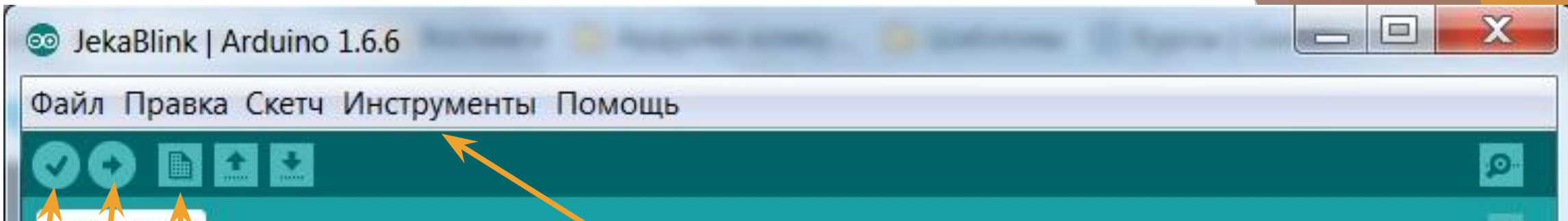




# Згадуємо дискретні(цифрові) входи та виходи

Цифровий сигнал - це сигнал який може набувати двох значень високого HIGH та низького LOW або ж 1/0.



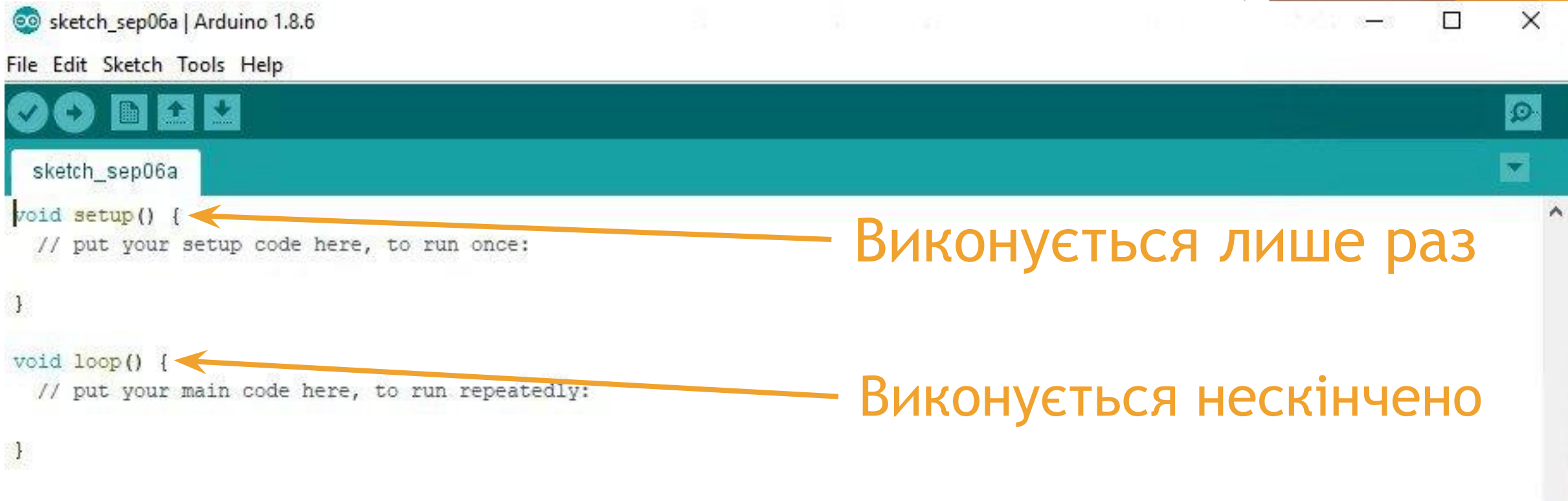


Новий скетч

Компіляція(перевірка) та  
Завантаження

Компіляція(перевірка)

# Що таке `void setup()` та `void loop()`.



The image shows a screenshot of the Arduino IDE interface. The window title is "sketch\_sep06a | Arduino 1.8.6". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for a checkmark, a right arrow, a document, an upload arrow, and a download arrow. The file name "sketch\_sep06a" is displayed in the editor's title bar. The code in the editor is as follows:

```
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

Two orange arrows point from the code to explanatory text on the right:

- An arrow points from the `void setup() {` line to the text "Виконується лише раз" (Executed only once).
- An arrow points from the `void loop() {` line to the text "Виконується нескінчено" (Executed infinitely).

# Які типи змінних вивчали?

Змінна типу **integer (int)**.

Змінна типу **boolean (bool)**.

Змінна з плаваючою комою **float**.

Змінна типу **char**.

Текстова змінна типу **String**

# Змінна типу `intenger` (`int`)

це така комірка пам'яті яка дозволяє записувати значення цілих чисел від -32 768 до 32 767

```
int val=0;
```



# Змінна типу **boolean (bool)**

це така комірка пам'яті яка дозволяє записувати логічні значення тобто правда, або неправда **true/false**, також можна записувати так: **1- це true, 0- це false**

```
bool val1=1;//true  
bool val2=0;//false  
bool val3=true;//1  
bool val4=false;//0
```

# Змінна з плаваючою комою **float**

це така комірка пам'яті яка дозволяє записувати значення чисел з плаваючою комою від  $-3.4028235E+38$  до  $3.4028235E+38$

```
float val=1.00;
```

# Змінна типу char

це така комірка пам'яті яка дозволяє записувати значення символів

```
char val='a';
```

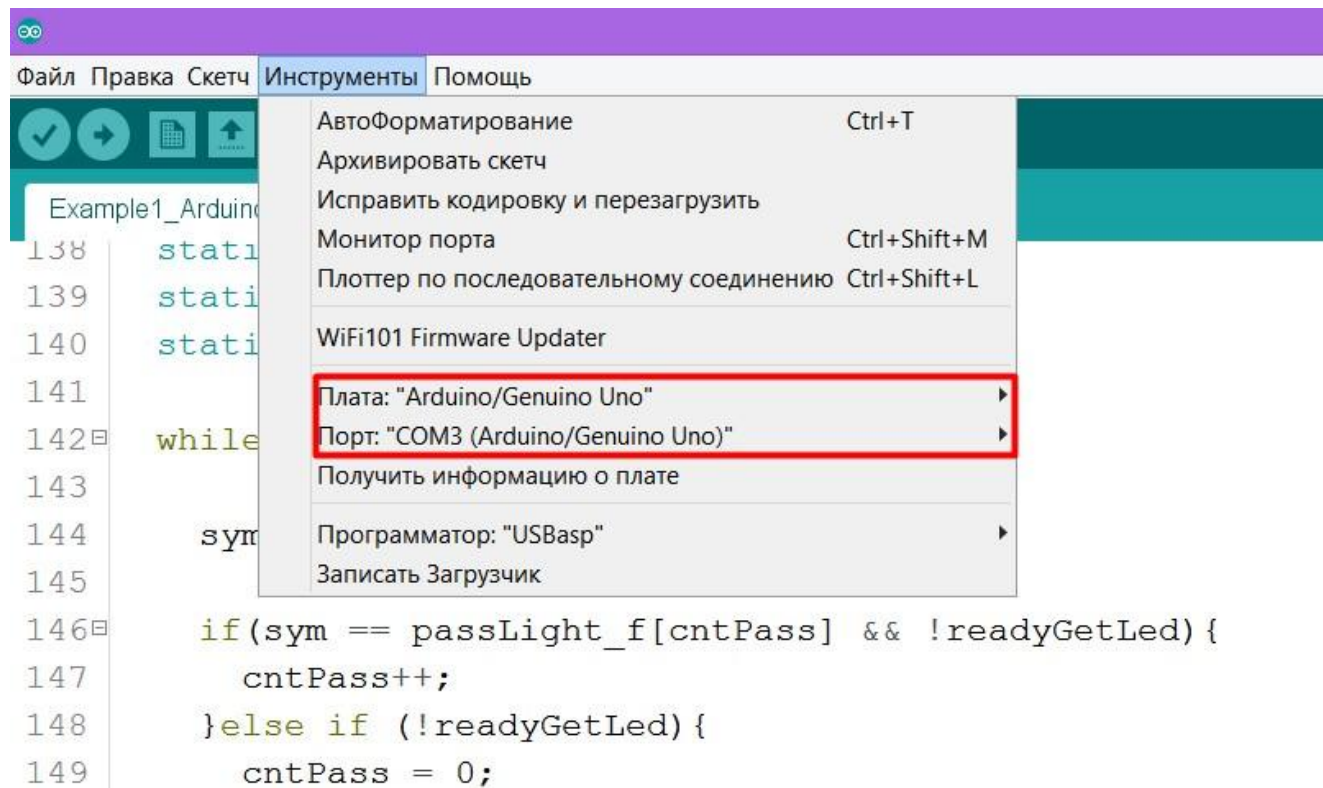
# Текстова змінна типу **String**

це така комірка пам'яті яка дозволяє записувати текстове значення

```
String val="Hello";
```

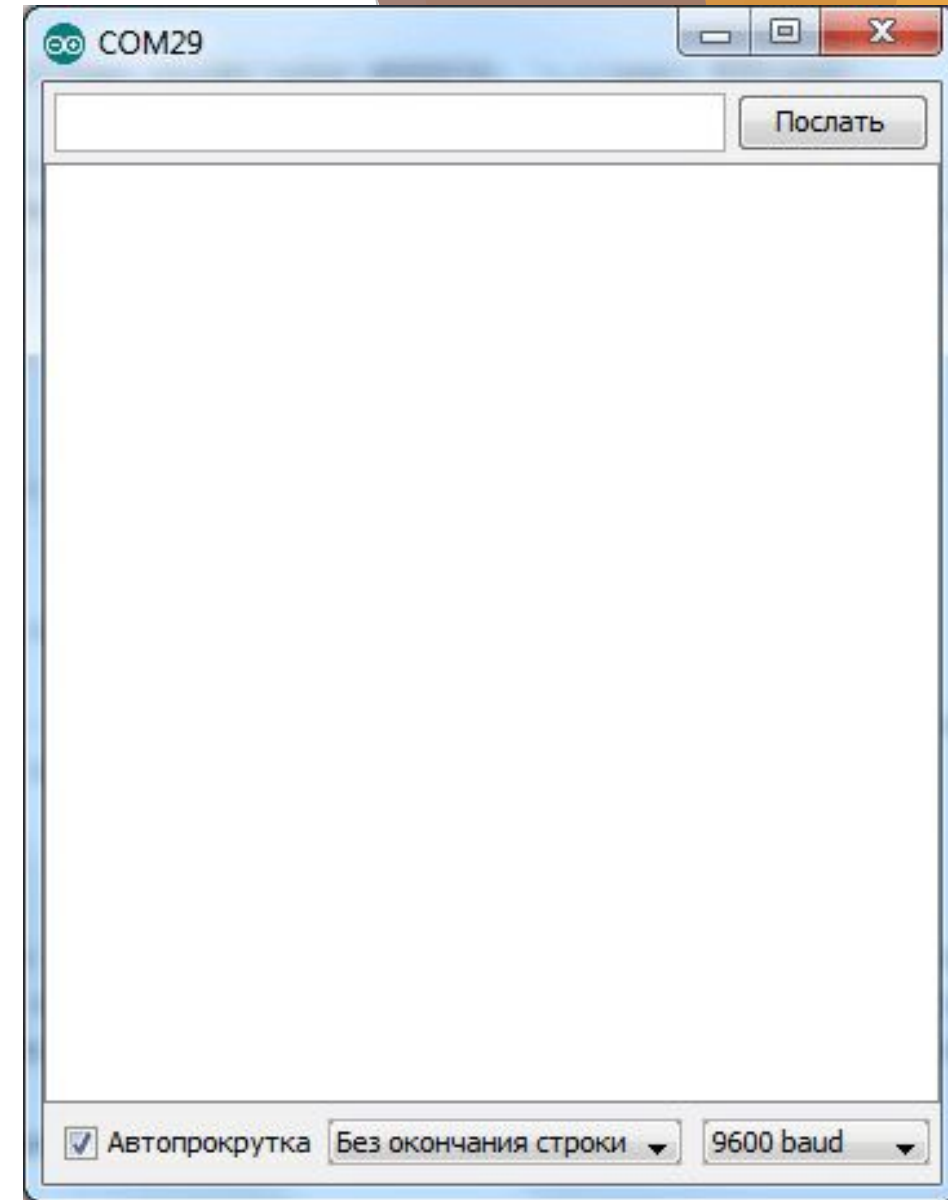
# Згадуємо монітор порту

Монітор порту (com port,uart) - це порт для зв'язку з платою Arduino.



The screenshot shows the Arduino IDE interface. The 'Tools' menu is open, and the option 'Порт: "COM3 (Arduino/Genuino Uno)"' is highlighted with a red rectangle. The menu also shows 'Плата: "Arduino/Genuino Uno"', 'Программатор: "USBasp"', and 'Записать Загрузчик'. The background shows a sketch with C++ code.

```
138 stati
139 stati
140 stati
141
142 while
143
144 sym
145
146 if(sym == passLight_f[cntPass] && !readyGetLed) {
147     cntPass++;
148 }else if (!readyGetLed) {
149     cntPass = 0;
```





# Команди управління

**Serial.begin(швидкість)**-запуск зв'язку

**Serial.print(дані)**-надсилає дані без переходу на нову строку

**Serial.println(дані)**-надсилає дані з переходом на нову строку

**Serial.read()**-зчитує дані

# Згадуємо аналоговий сигнал

Аналоговий сигнал, може набувати значення в діапазоні сигналу, тобто від 0 до 5в. Чому саме 5в?

# Згадуємо аналоговий сигнал

Arduino NANO 8 аналогових входів, від A0 до A7, позначається буквою A. Для зчитування значення використовується команда :

```
analogRead(пін входу);
```

# Згадуємо Аналогові Входи

При зчитуванні аналогового сигналу Arduino конвертує його в діапазоні від 0 до 1023

Тобто при зчитуванні:

0в-це число 0, а 5в-це число 1023

# Згадуємо Аналогові Виходи

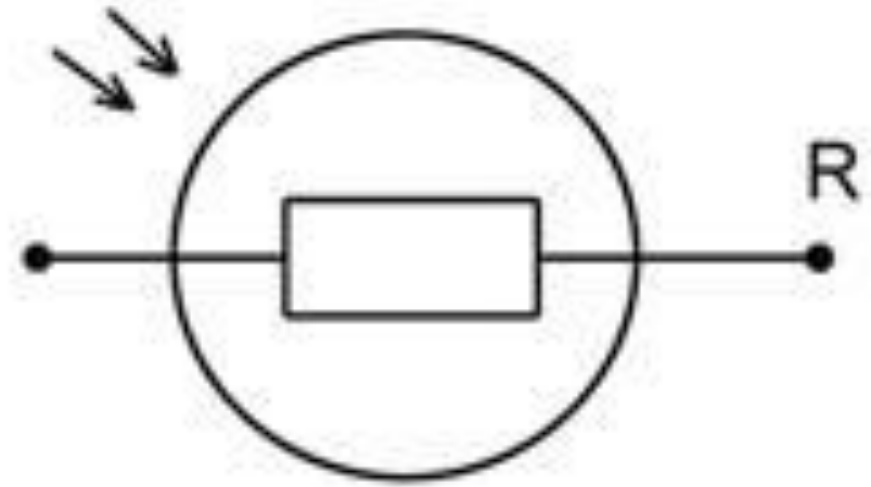
В ARDUINO NANO доступні 6 пінів які можна налаштувати як аналоговий вихід, це 3, 5, 6, 9, 10, 11 піни. Так ви все вірно почули, деякі цифрові піни, можна налаштувати на аналоговий вихід.

Команда управління:

```
analogWrite(пін виходу,0..255);
```



# Згадуємо фоторезистор



# Вивчаємо перемикач

Перемикач або вимикач, іноді тумблер— електромеханічний пристрій для розмикання електричного кола або перемикання електричного струму з одного провідника на інший

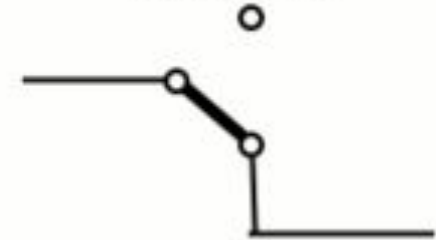


# Вивчаємо перемикач

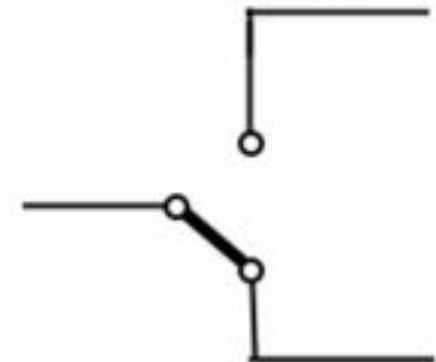
Перемикачі бувають звичайні та прохідні.  
Прохідні перемикають сигнал між середнім,  
та двома або більше контактів



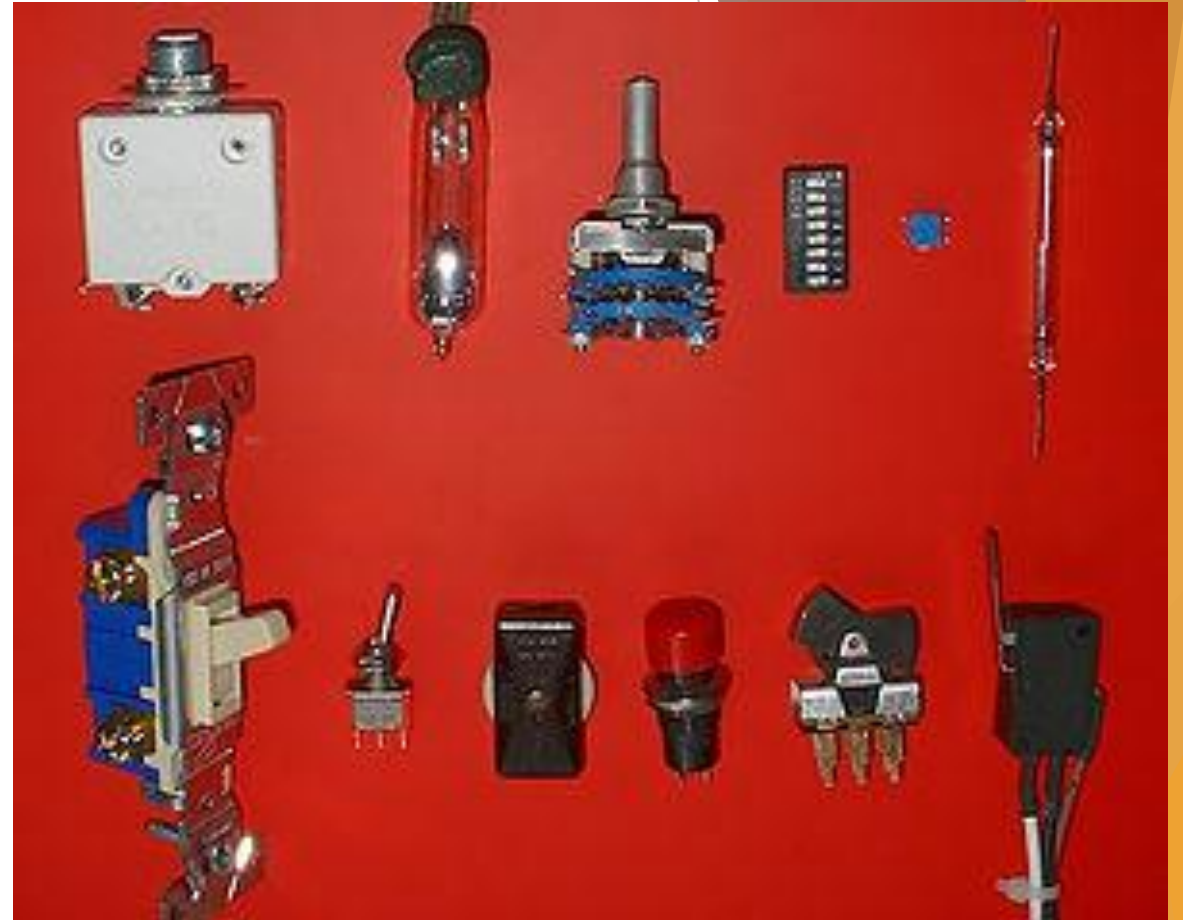
Звичайний  
вимикач



Прохідний  
вимикач



# Які бувають?



# Вивчаємо цикл `while`

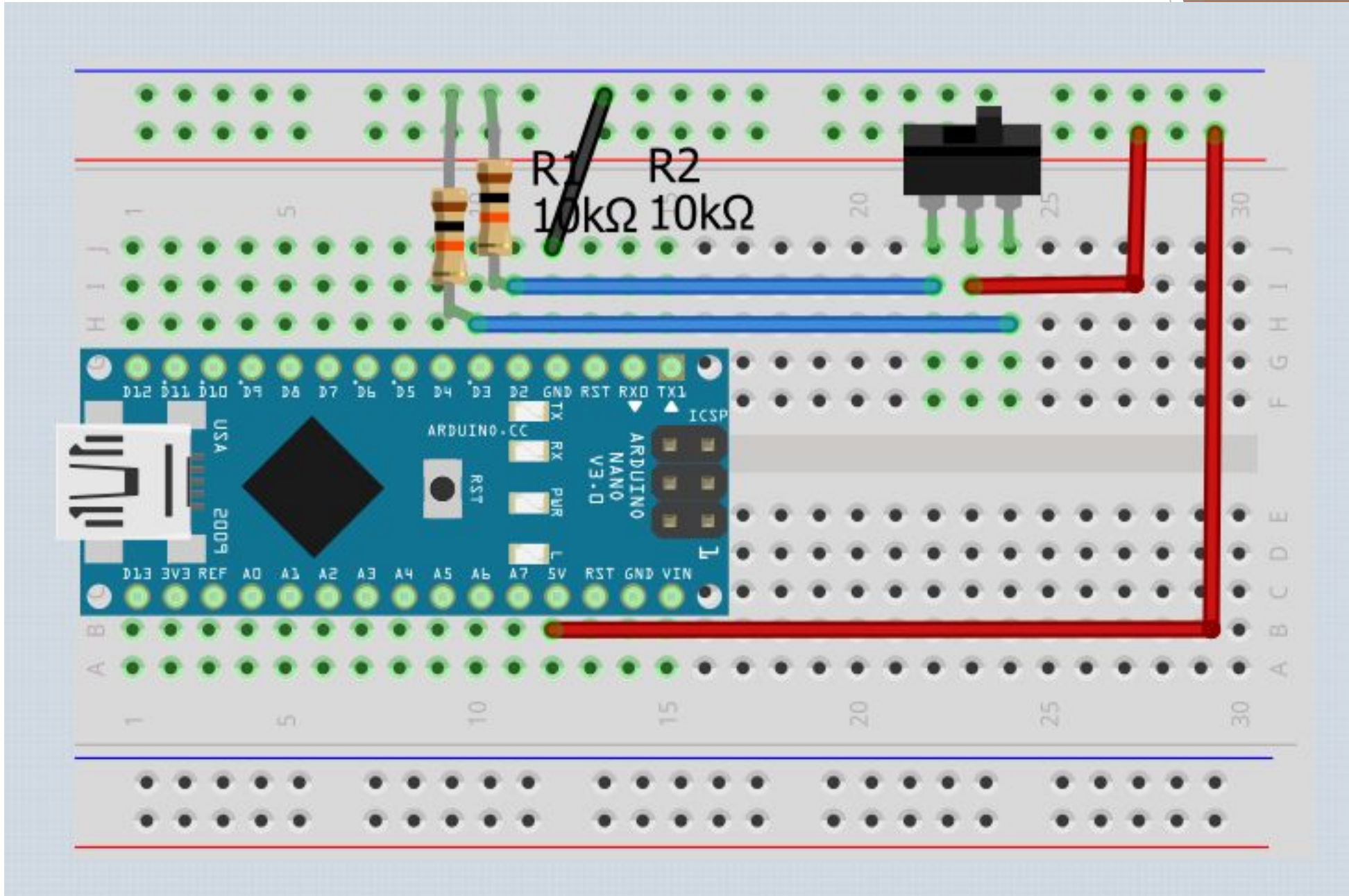
Цикл `while` - це такий цикл який виконується при певній умові, без можливості читання коду якщо умова циклу виконується.



# Синтаксис цикла while

```
while (digitalRead(3) == HIGH)
{
    digitalWrite(5, LOW);
    digitalWrite(6, HIGH);
    delay(1000);
    digitalWrite(5, HIGH);
    digitalWrite(6, LOW);
    delay(1000);
}
```

# Зберемо схему

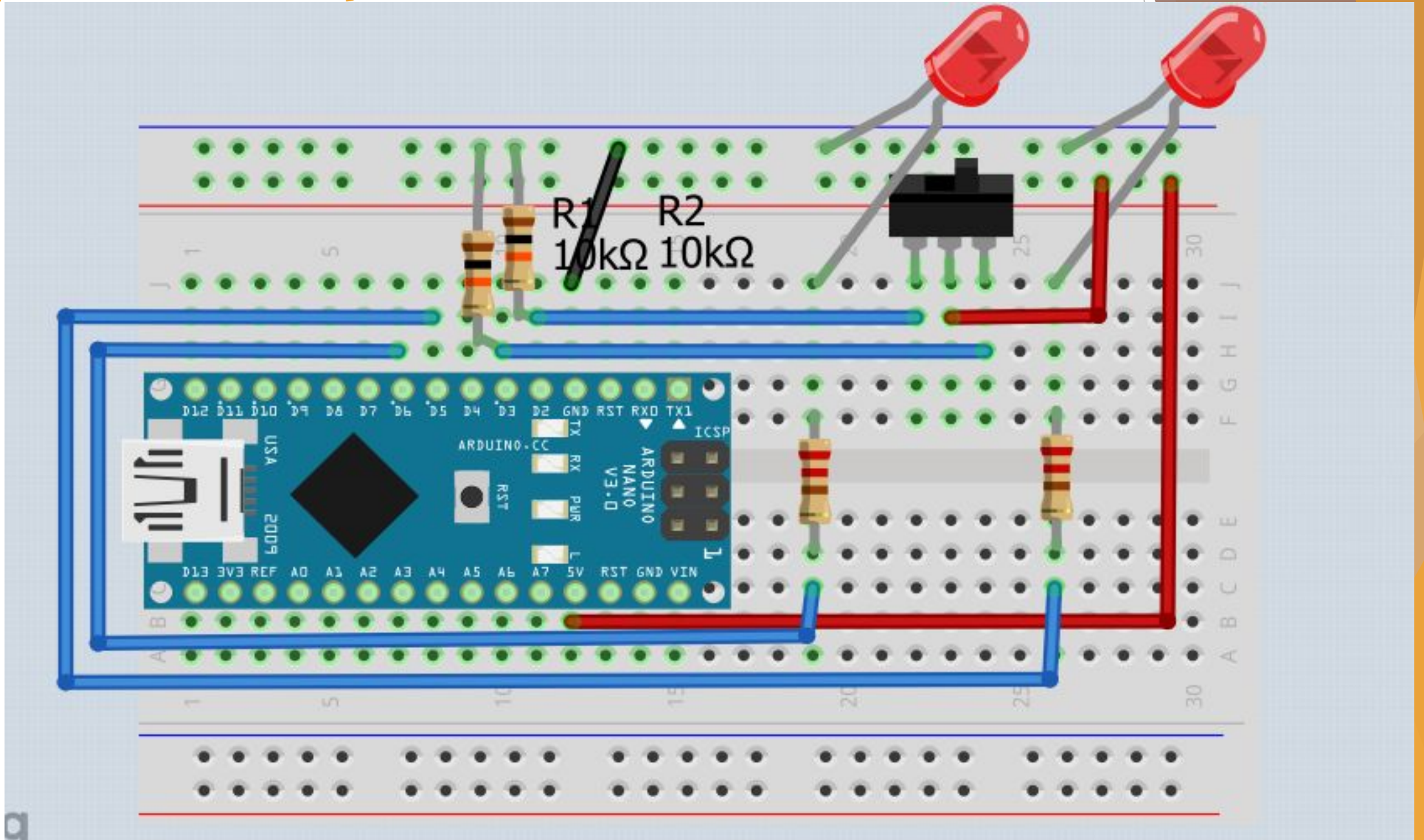


# Пишемо програму

```
void setup() //виконується лише один раз за життя програми
{
    Serial.begin(9600); //вмикаємо монітор порту
    pinMode(2, INPUT); //команда налаштування піна-> на вхід
    pinMode(3, INPUT); //команда налаштування піна-> на вхід
}
void loop() //виконується нескінченно
{
    Serial.print("Pin 2=");
    Serial.print(digitalRead(2));
    Serial.print("Pin 3=");
    Serial.print(digitalRead(3));
    Serial.println();
}
```



# Зберемо схему



# Пишемо програму

```
void setup() //виконується лише один раз за життя програми
{
    Serial.begin(9600); //вмикаємо монітор порту
    pinMode(2, INPUT); //команда налаштування піна-> на вхід
    pinMode(3, INPUT); //команда налаштування піна-> на вхід
    pinMode(5, OUTPUT); //команда налаштування піна-> на вихід
    pinMode(6, OUTPUT); //команда налаштування піна-> на вихід
}
void loop() //виконується нескінченно
{
    Serial.print("Pin 2=");
    Serial.print(digitalRead(2));
    Serial.print("Pin 3=");
    Serial.print(digitalRead(3));
    Serial.println();
    if(digitalRead(2)==HIGH)
    {
        digitalWrite(5, HIGH);
        digitalWrite(6, LOW);
    }
    else if(digitalRead(3)==HIGH)
    {
        digitalWrite(5, LOW);
        digitalWrite(6, HIGH);
    }
}
```



# Пишемо програму

```
void loop() //виконується нескінченно
{
    Serial.print("Pin 2=");
    Serial.print(digitalRead(2));
    Serial.print("Pin 3=");
    Serial.print(digitalRead(3));
    Serial.println();
    if(digitalRead(2)==HIGH)
    {
        digitalWrite(5,HIGH);
        digitalWrite(6,LOW);
        delay(1000);
        digitalWrite(5,LOW);
        digitalWrite(6,LOW);
        delay(1000);
    }
    else if(digitalRead(3)==HIGH)
    {
        digitalWrite(5,LOW);
        digitalWrite(6,HIGH);
        delay(1000);
        digitalWrite(5,LOW);
        digitalWrite(6,LOW);
        delay(1000);
    }
}
```



# Пишемо програму

```
void loop()//виконується нескінченно
{
    Serial.print("Pin 2=");
    Serial.print(digitalRead(2));
    Serial.print("Pin 3=");
    Serial.print(digitalRead(3));
    Serial.println();
    while(digitalRead(2)==HIGH)
    {
        digitalWrite(5,HIGH);
        digitalWrite(6,LOW);
        delay(1000);
        digitalWrite(5,LOW);
        digitalWrite(6,LOW);
        delay(1000);
    }
    Serial.println("mitka 1");
    while(digitalRead(3)==HIGH)
    {
        digitalWrite(5,LOW);
        digitalWrite(6,HIGH);
        delay(1000);
        digitalWrite(5,LOW);
        digitalWrite(6,LOW);
        delay(1000);
    }
    Serial.println("mitka 2");
}
```