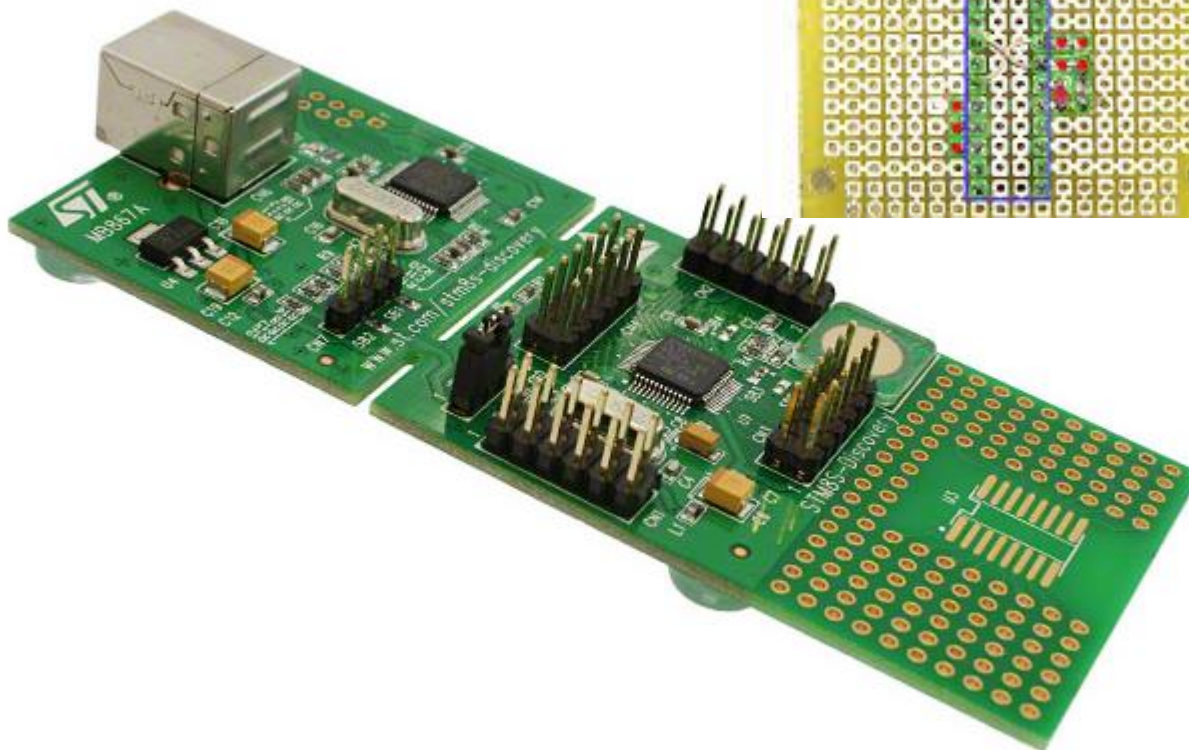


5. Відлагоджувальні плати на базі МК сімейства AVR

Макетна плата – пристрій для складання та моделювання прототипів електронних пристроїв. Макетні плати поділяються на два типи: для монтажу за допомогою пайки і без такої.

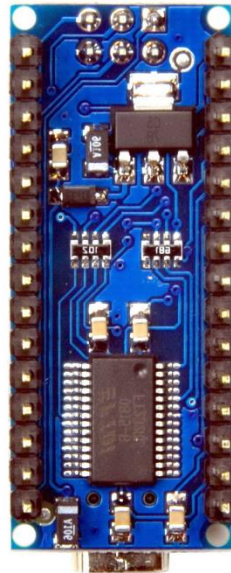
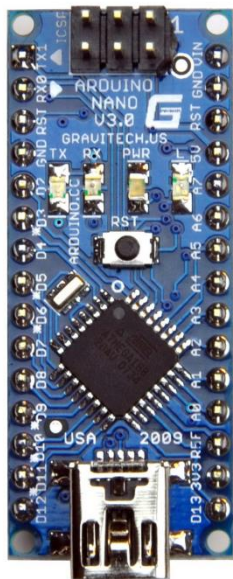


Відлагоджувальні плати – це друковані плати з встановленим на ній мікроконтролером з усією необхідною йому стандартної обв'язкою. На плати також встановлюють: схеми зв'язку з комп'ютером, розведення для підключення плат розширення, макетну область для монтажу прикладних схем користувача. Служить для тестування програм написаних для проекту, який розробляється.



Платформа Nano, побудована на мікроконтролері ATmega328/168, має невеликі розміри. Nano розроблена і продається компанією Gravitech. Arduino Nano може отримувати живлення через підключення до роз'єму Mini-B USB, або від нерегульованого 6-20 В (вивід 30), або регульованого 5В (вивід 27) зовнішнього джерела живлення.

Мікроконтролер ATmega168 має 16 кБ флеш-пам'яті для зберігання коду програми, а мікроконтролер ATmega328, в свою чергу, має 32 кБ. ATmega168 має 1 кБ ОЗУ і 512 байт EEPROM, а ATmega328 - 2 кБ ОЗУ і 1 Кб EEPROM. Рідний Arduino Nano – клон Gravitech.

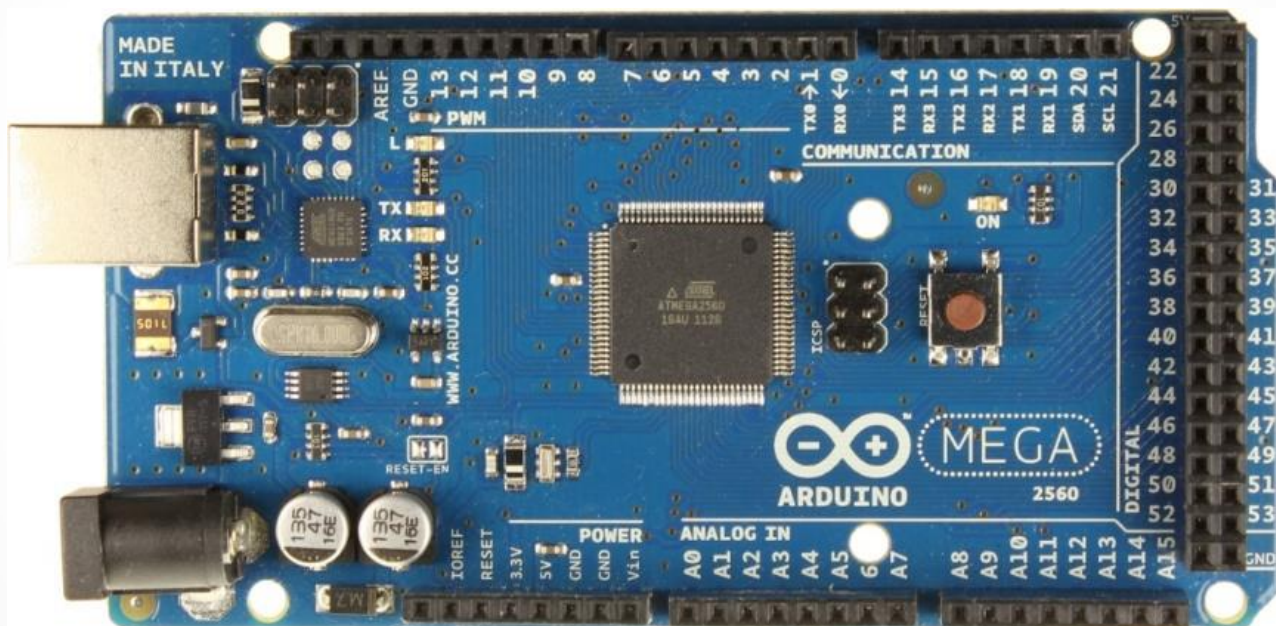


Arduino Uno R3 – популярна лінійки контролерів Arduino виконана на мікроконтролері ATmega328. Контролер має 14 цифрових вхід / виходів, 6 з яких підтримують режим широтно-імпульсна модуляція (ШИМ), 6 аналогових входів.

Плата також має вбудований USB конектор, вхід для живлення, роз'єм для внутрисхемного програмування ICSP (In-Circuit Serial Programming) і кнопку перезавантаження. Живлення може здійснюватися як через USB, так і з зовнішнього джерела живлення. Чіп FTDI – відповідає за USB з'єднання (UART-TTL).

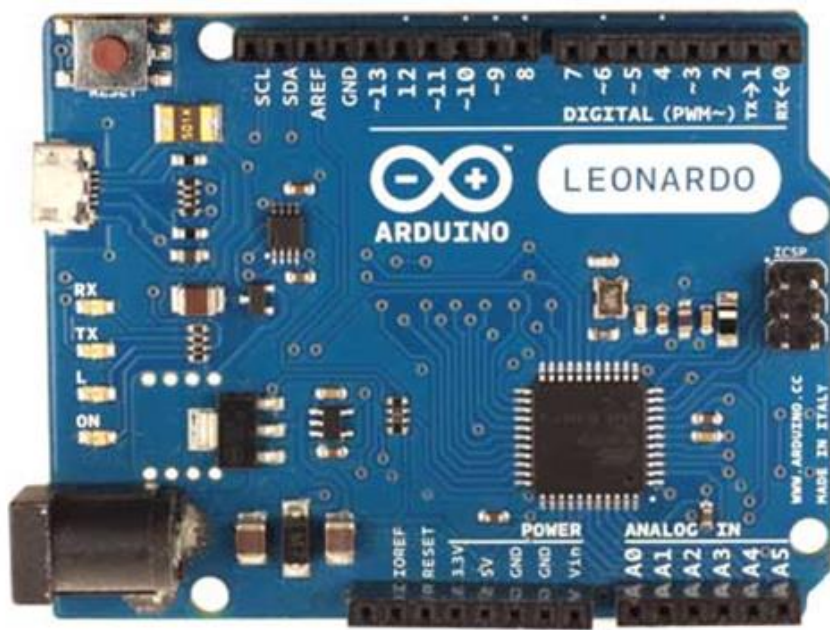


Arduino Mega 2560 Rev3 – виконана на мікроконтролері ATmega2560. Схожа на плату Arduino Uno R3, але надає розробникам більшу кількість портів вхід / виходу, а також підтримує 4 порти послідовної (Serial) передачі даних UART-to-TTL.

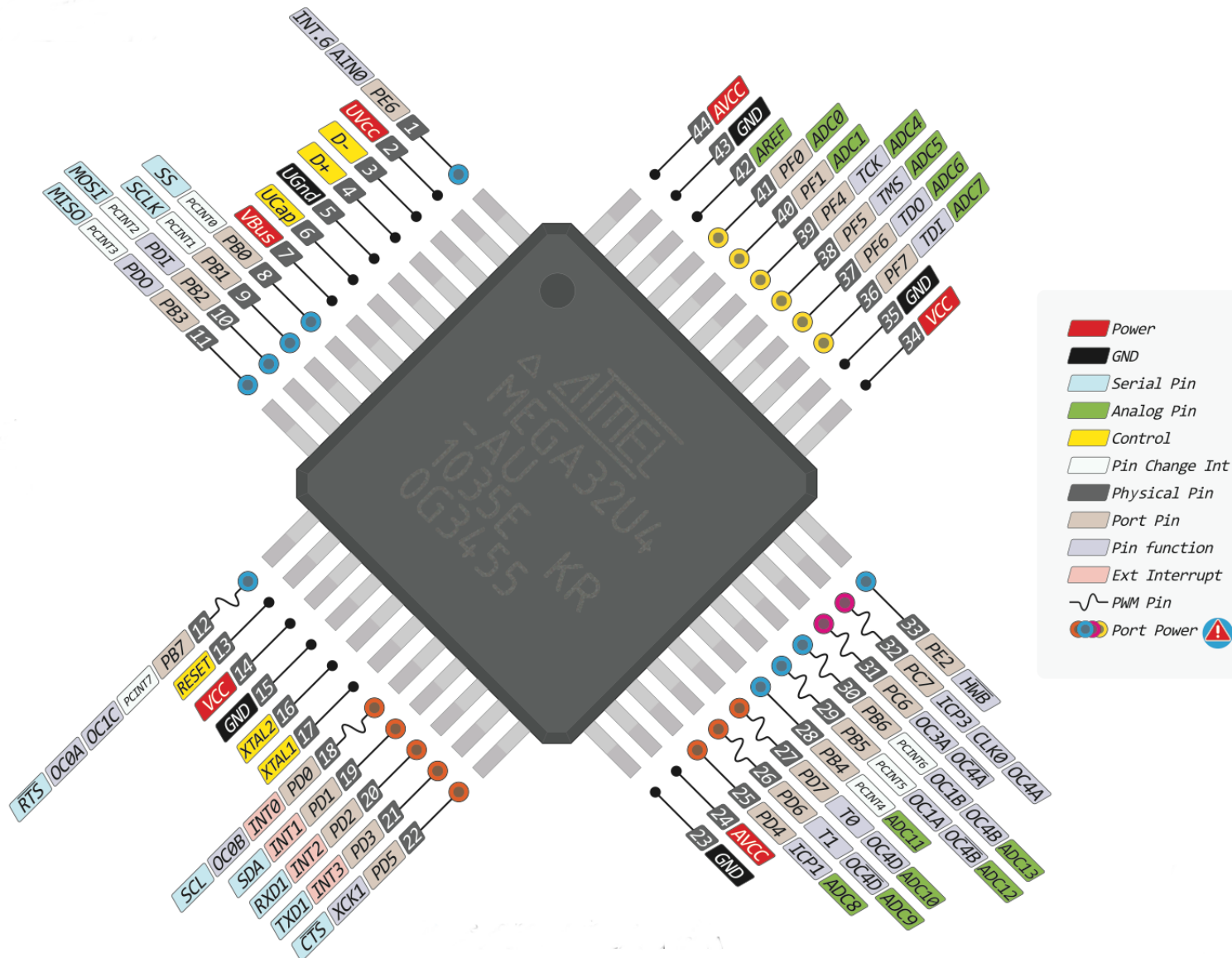


Arduino Leonardo – це пристрій на базі мікроконтролера ATmega32U4. Візуально схожа на плату Arduino Uno R3. У його склад входить все необхідне для роботи з даними мікро контролером: 20 цифрових входів / виходів (7 з яких можуть працювати в якості ШІМ-виходів, 12 - в якості аналогових входів), кварцовий резонатор на 16 МГц, роз'єм мікро-USB, роз'єм живлення, роз'єм для внутрисхемного програмування ICSP і кнопка перезавантаження.

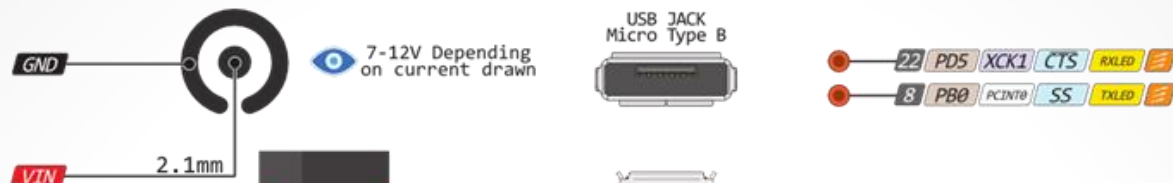
Відмінність Leonardo від всіх попередніх плат полягає в тому, що його USB-контролер вбудований безпосередньо в мікроконтролер ATmega32U4. Завдяки цьому при приєднанні до комп'ютера Leonardo може визначатися не тільки як віртуальний COM-порт, але і як звичайна миша, клавіатура тощо.



Призначення виводів МК ATmega32U4 та плати Arduino Leonardo



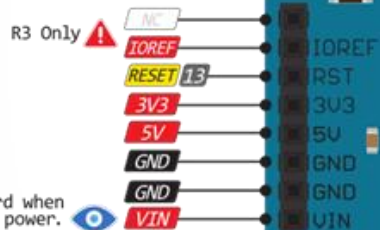
LEONARDO PINOUT



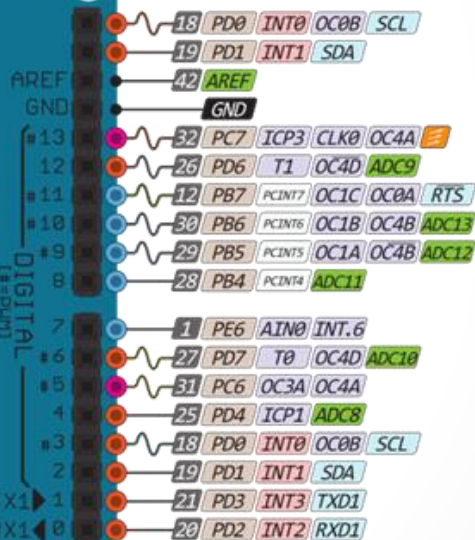
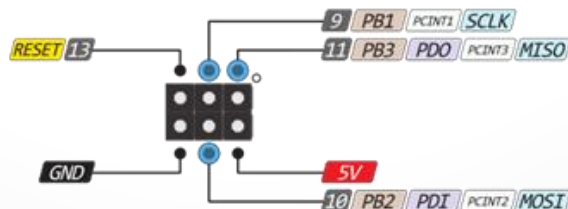
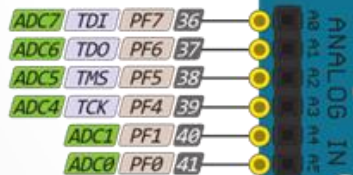
⚠ Absolute MAX per pin 20mA recommended 10mA

⚠ Absolute MAX 200mA for entire package

IOREF provides a logic reference voltage for shields that use it. It is connected to the 5V bus.



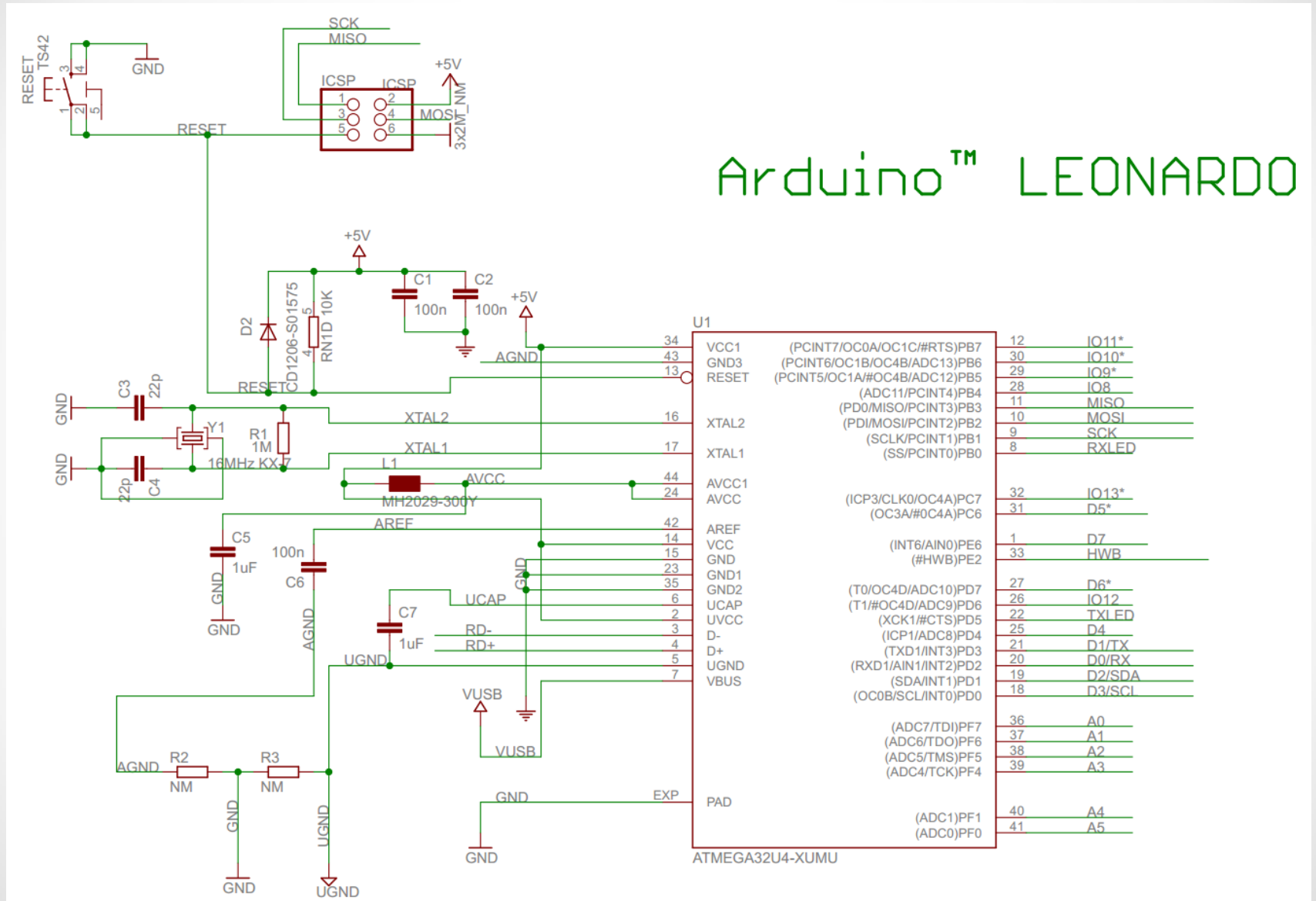
The input voltage to the board when it is running from external power. Not USB bus power.



- Power
- GND
- Serial Pin
- Analog Pin
- Control
- INT
- Physical Pin
- Port Pin
- Pin function
- Interrupt Pin
- PWM Pin
- Port Power

⚠ The power sum for each pin's group should not exceed 100mA

Частина принципової схеми Arduino Leonardo:



Виводи, що розташовані на платі

1. VIN. Напруга, що надходить в Arduino безпосередньо від зовнішнього джерела живлення. Через цей вивід можна подавати до зовнішнього джерела живлення, так і споживати струм, коли пристрій живиться від зовнішнього адаптера.
2. 5V. На вивід надходить напруга 5В від стабілізатора напруги плати, поза незалежності від типу живлення пристрою. Живити пристрій через виводи 5V або 3V3 не рекомендується, оскільки в цьому випадку не використовується стабілізатор напруги, що може привести до виходу плати з ладу.
3. 3V3 (3.3В), що надходять від стабілізатора напруги на платі.
4. GND. Загальний (Земля).
5. AREF. Цей вивід підключення зовнішнього джерела опорної напруги для АЦП.
6. Reset. Формування низького рівня (LOW) на цьому виводі призведе до перезавантаження мікроконтролера. Зазвичай цей вивід служить для функціонування кнопки скидання на платах розширення.

7. В Arduino Leonardo є 6 (12 у МК) аналогових входів (A0-A5), кожен з яких може представити аналогову напругу у вигляді 10-бітного числа (1024 різних значення). За замовчуванням, вимір напруги здійснюється відносно діапазону від 0 до 5 В. Проте, верхню межу цього діапазону можна змінити, використовуючи вивід Analog Reference (AREF) і функцію `analogReference()`.

8. В Arduino Leonardo є 14 (20 у МК) цифрових входів / виходів (D0-D13). Деякі цифрові висновки Arduino можуть виконувати додаткові функції:

1) Послідовний інтерфейс: висновки D0 (RX) і D1 (TX). Використовуються для отримання (RX) і передачі (TX) даних по послідовному інтерфейсу.

2) Зовнішні переривання: виводи D0 - D3. можуть служити джерелом переривань, що виникають при фронті, спаді або при низькому рівні сигналу на цих виводах.

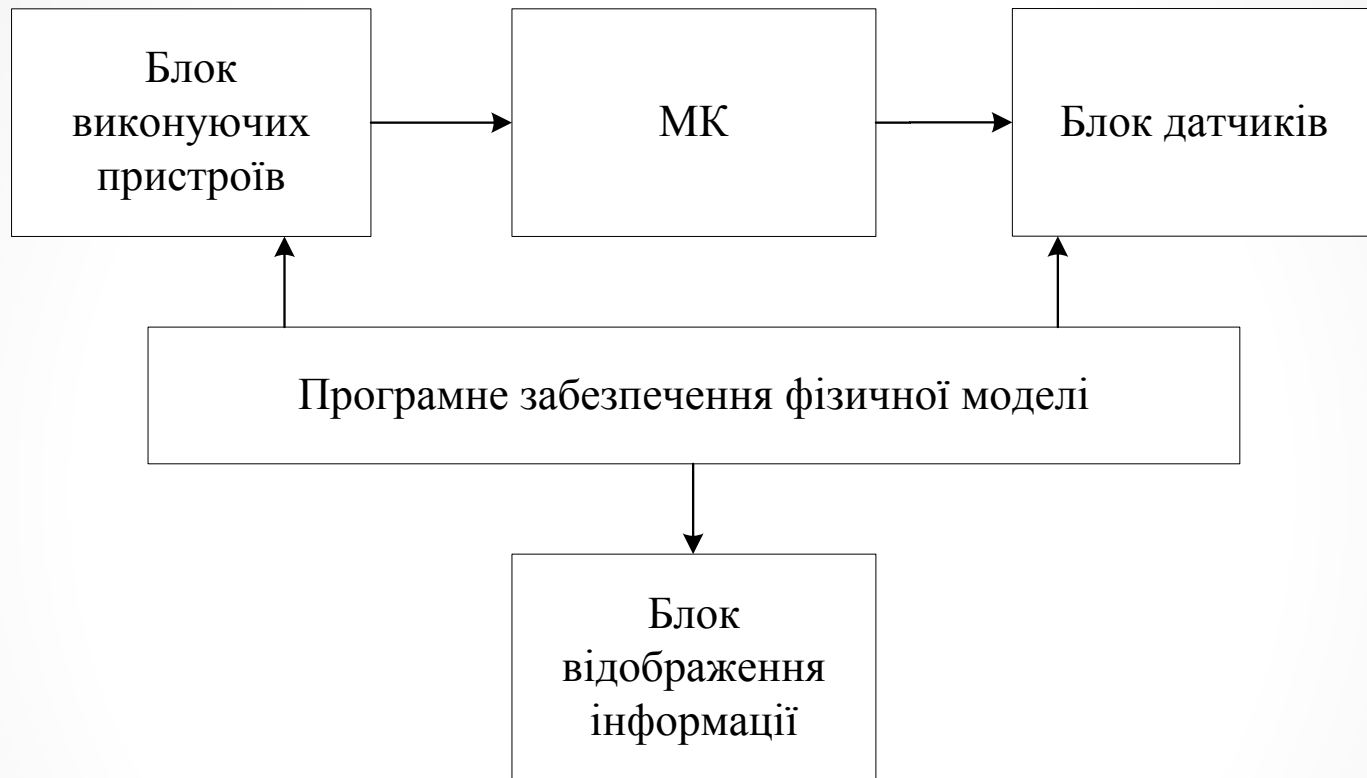
3) ШІМ-виводами є виводи з номерами D3, D5, D6, D9 - D13. За допомогою функції `analogWrite()` можуть виводити 8-бітові аналогові значення у вигляді ШІМ-сигналу.

4) Інтерфейс SPI: виводи ICSP4 (MOSI), ICSP1 (MISO), ICSP3 (SCK). Дані виводи можуть здійснювати зв'язок по інтерфейсу SPI.

5) Вбудований світлодіод, приєднаний до виводу D13. При відправленні значення HIGH – світлодіод включається, при відправці LOW – вимикається.

6) тощо

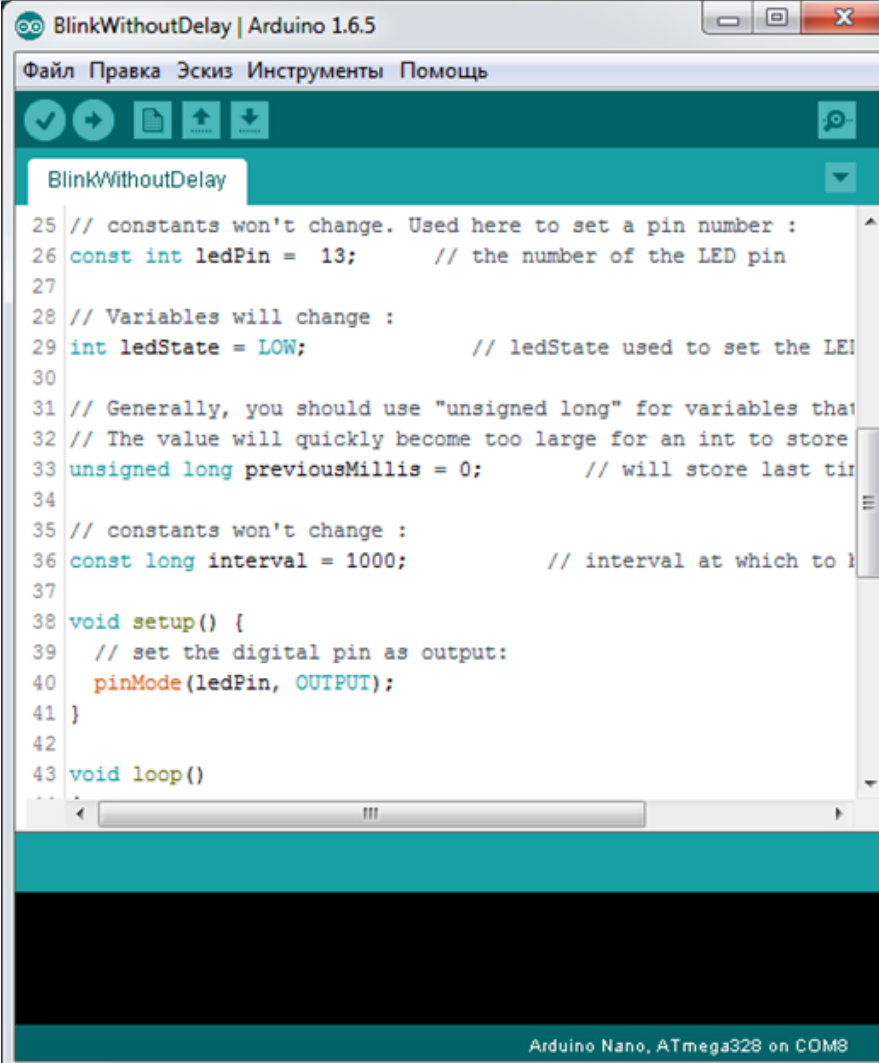
Загальна структурна схема МПС на основі мікроконтролера:





Блок датчиків дає можливість контролювати параметри оточуючого середовища. Блок виконуючих пристроїв – впливати на оточуюче середовище. Блок опрацювання інформації забезпечує обробку даних від датчиків та видає керуючі сигнали блоку виконуючих пристроїв згідно з програмою користувача. Блок відображення інформації дає змогу отримати дані про результати опрацювання даних, поточний стан роботи МПС.

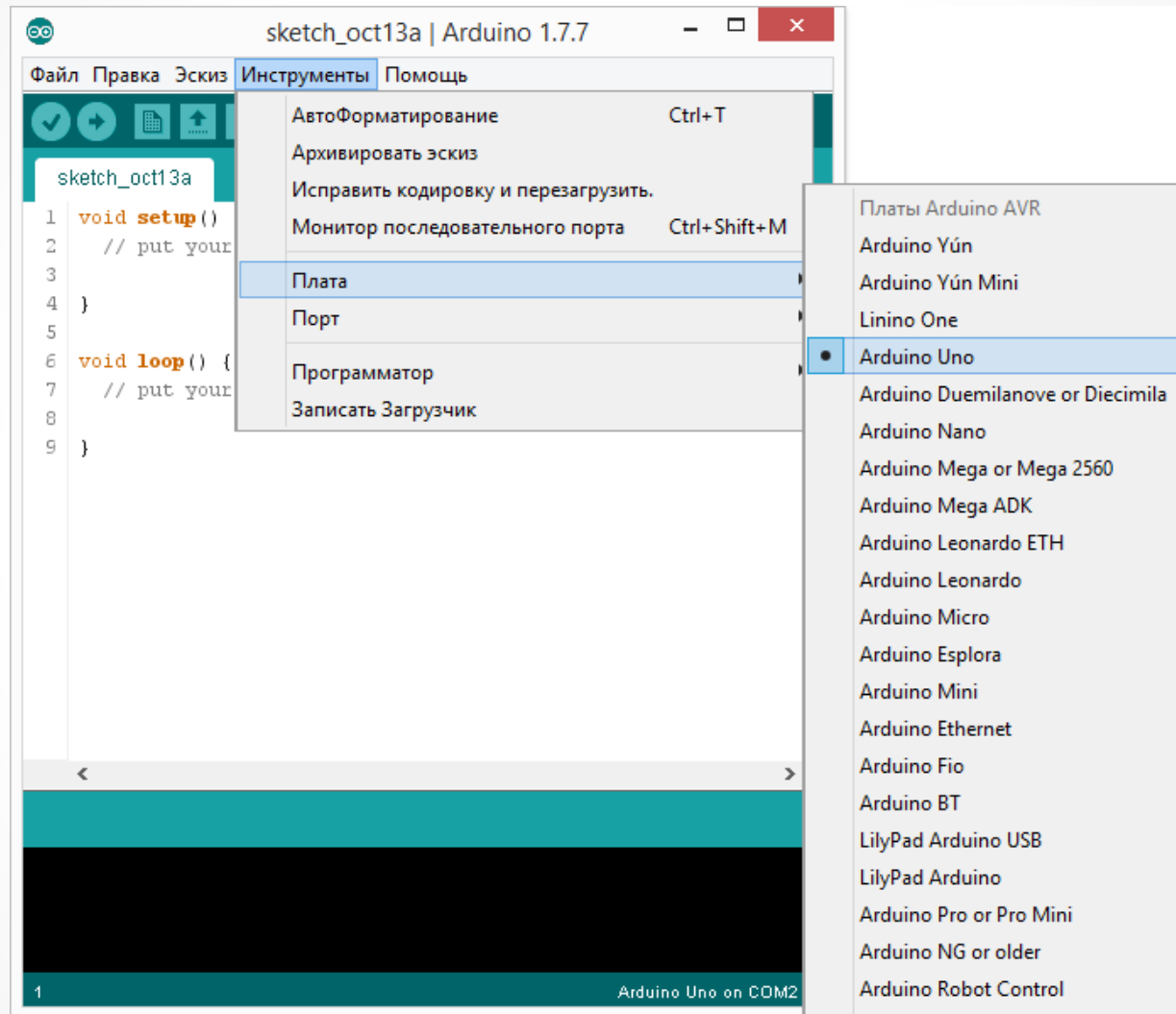
Інтегроване середовище розробки Arduino – це кроссплатформний додаток на Java, що включає в себе редактор коду, компілятор і модуль передачі прошивки в плату.

A screenshot of the Arduino IDE interface. The title bar reads "BlinkWithoutDelay | Arduino 1.6.5". The menu bar includes "Файл", "Правка", "Эскиз", "Инструменты", and "Помощь". Below the menu is a toolbar with icons for opening, saving, and running. A tab labeled "BlinkWithoutDelay" is active. The code editor displays the following C++ code:

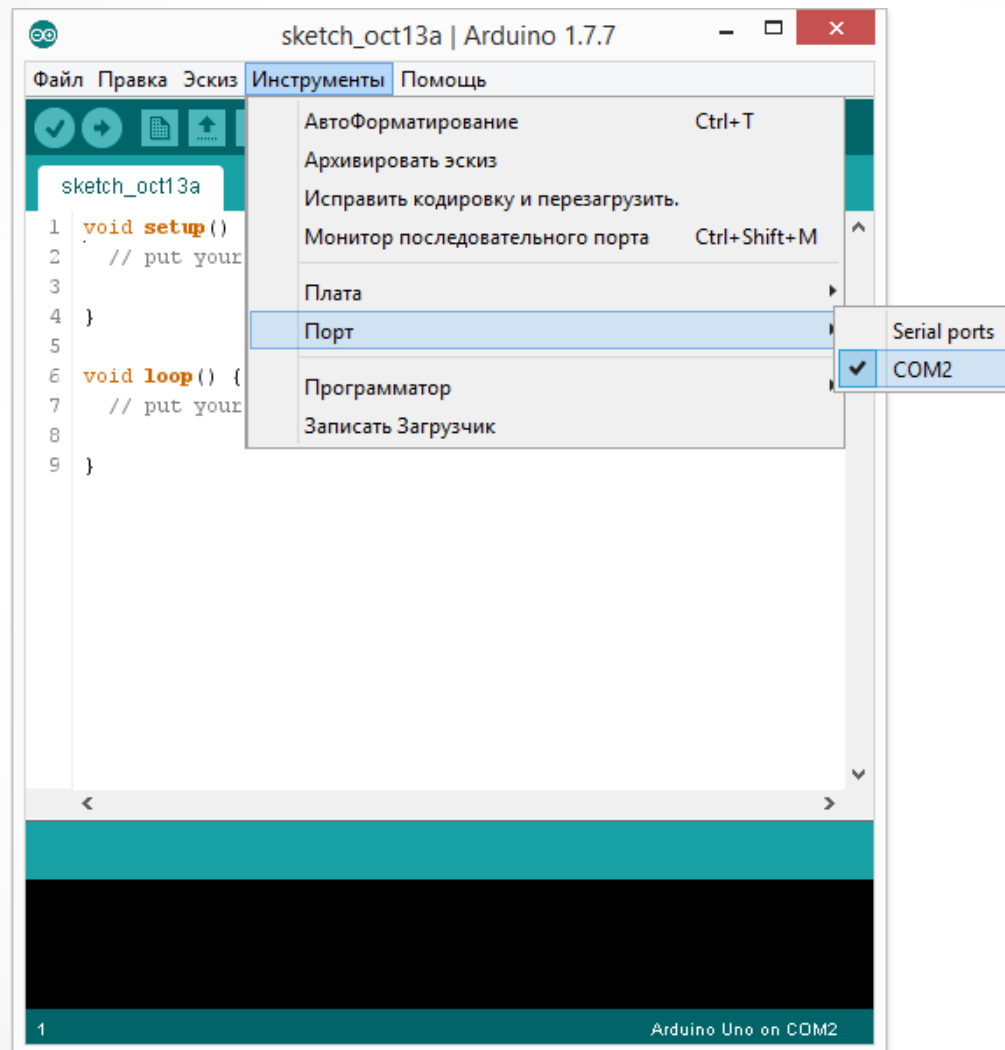
```
25 // constants won't change. Used here to set a pin number :
26 const int ledPin = 13;      // the number of the LED pin
27
28 // Variables will change :
29 int ledState = LOW;          // ledState used to set the LED
30
31 // Generally, you should use "unsigned long" for variables that
32 // The value will quickly become too large for an int to store
33 unsigned long previousMillis = 0;        // will store last time LED was on
34
35 // constants won't change :
36 const long interval = 1000;              // interval at which to blink
37
38 void setup() {
39   // set the digital pin as output:
40   pinMode(ledPin, OUTPUT);
41 }
42
43 void loop()
44 {
45   // ...
46 }
```

The bottom status bar indicates "Arduino Nano, ATmega328 on COM8".

Вибір відлагоджувальної плати Arduino:



Вибір COM-порту в Arduino IDE:



Для створення додатка необхідно виконати наступні кроки:

1. Підключити плату до комп'ютера;
2. Запустити (встановити, якщо відсутнє) середовище розробки для Arduino;
3. Встановити драйвера (для Arduino Leonardo не потрібно);
4. Обрати плату Arduino (в нашому випадку Arduino Leonardo);
5. Обрати послідовний порт;
6. Відкрити або створити програму;
7. Скомпілювати програму (перша кнопка на панелі інструментів середовища);
8. Завантажити прошивку у Arduino (друга кнопка на панелі інструментів середовища).

Приклад програми:

Лістинг 5.1 – Вмикання внутрішнього світлодіода

```
void setup() {  
    // Ініціалізація  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH); // Включити  
}
```