

Тема 1. Знайомство з Arduino

Урок 1.1. Макетний стенд Arduino і редактор програм

1.1.1. Основні елементи Arduino Uno



Мікроконтролер аналог мікропроцесора в звичайному ПК;

Кнопка скидання здійснює скидання мікроконтролера і повторити запуск програми;

Порт USB забезпечує зв'язок з ПК і живлення пристрою;

Світлодіод №13 світлодіод, з'єднаний з цифровим виходом №13;

Живлення +9 додаткове живлення від зовнішнього джерела (батарея, блок живлення);

Підключення Arduino Uno до ПК

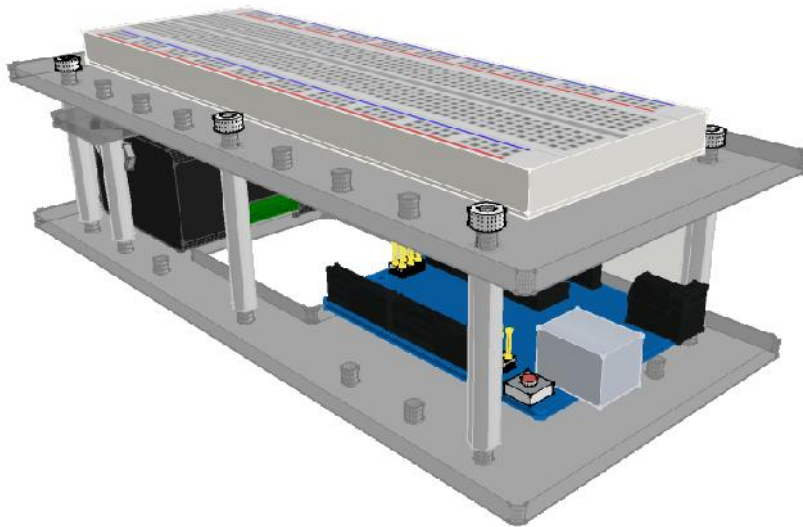
Для роботи з Arduino Uno буде потрібно будь-який персональний комп'ютер з вільним USB портом. У разі підключення Arduino до ПК, зовнішнє живлення пристрою не потрібно.

Запуск Arduino Uno з зовнішнім джерелом

Для включення мікроконтролера, досить подати на нього живлення або від ПК через USB кабель, або безпосередньо через спеціальний роз'єм зовнішнього живлення. При цьому, напруга зовнішнього живлення може варіюватися від + 7В до +12В.

1.1.2. Макетний стенд і макетна плата

Макетний стенд, який використовується в даному курсі, складається з двох поверхів. При цьому, є можливість нарощувати додаткові поверхи як зверху, так і знизу стенду.

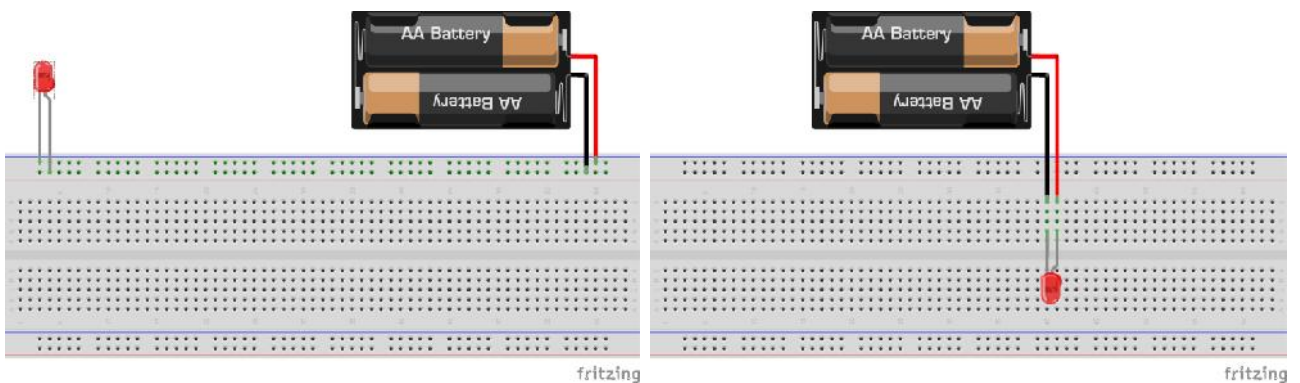


На нижньому поверсі стенду розміщується сам контролер Arduino Uno і, опціонально, акумуляторна батарея для автономної роботи. На верхньому поверсі змонтована макетна плата, яка дозволяє здійснювати збірку електричних ланцюгів без пайки. Нижче наведена схема плати, на якій вказані всі можливі сполуки її провідників.



По краях плати йдуть довгі провідники, які частіше називають шиною живлення. Провідники ж розміщені в середині плати, з'єднуються в напрямку, перпендикулярно напрямку шини живлення.

Приклади використання макетної плати



1.1.2. Огляд елементів навчального набору

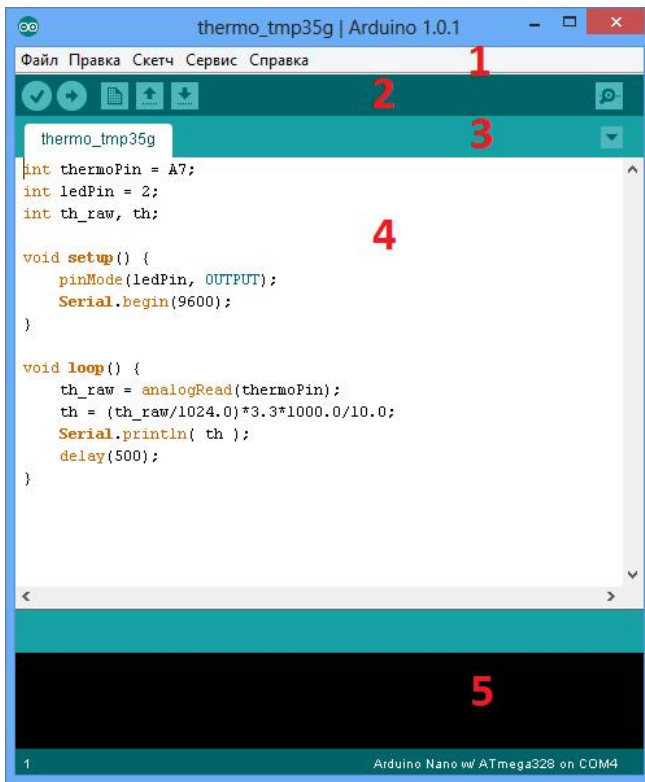
 <p>Светодиод красный - 2шт Светодиод зеленый - 2шт Светодиод желтый - 2шт</p>	 <p>Резистор 220 Ом - 6шт Резистор 10 кОм - 2шт Резистор 1кОм - 2шт</p>	 <p>Кнопка 2 шт</p>
 <p>Потенциометр регулировочный 47 кОм - 1шт</p>	 <p>Потенциометр подстроечный 10 кОм - 1шт</p>	 <p>Индикатор светодиодный сегментный - 1шт</p>
 <p>Дисплей жидкокристаллический - 1шт</p>	 <p>Драйвер двигателей постоянного тока - 1шт</p>	 <p>Двигатель постоянного тока с редуктором - 1шт</p>
 <p>Серво-двигатель - 1шт</p>	 <p>Датчик температуры - 1шт</p>	 <p>Датчик освещенности - 1шт</p>
 <p>Датчик Холла - 1шт</p>	 <p>Колесо с магнитом - 1шт</p>	 <p>Ультразвуковой дальномер - 1шт</p>

1.1.4. Редактор программ Arduino IDE

1. Головне меню

Меню редактора включає в себе наступні основні елементи: файл, правка, скетч, сервіс і довідка. Розглянемо докладніше кожен з них.

У пункті **Файл** можна знайти команди, що відповідають за створення нової програми, читання старої, збереження її змін, а також команди для завантаження програми на мікроконтролер.



- Створити - створити нову програму (скетч);
- Відкрити - відкрити існуючу програму;
- Папка зі скетчами - відкрити програму із заданої папки;
- Приклади - відкрити приклад програми;
- Закрити - закрити поточне вікно.
- Зберегти - зберегти зміни в раніше збережену програму;
- Зберегти як - зберегти нову програму, із зазначенням імені;
- Завантажити - завантажити програму в Arduino;
- Завантажити за допомогою програматора - завантажити програму за допомогою програматора;
- Налаштування друку – налаштування принтера;
- Друк - вивід на друк коду програми;
- Налаштування – налаштування редактора;
- Вихід - вихід з Arduino IDE.

Пункт меню **Правка** містить команди, пов'язані з редагування текст програми, включаючи копіювання, вставку, настройку відступів і пошук за ключовим словом.

У розділі **Скетч** розміщуються команди для контролю за процесом компіляції програми.







- Перевірити / Компілювати - компілювати програму;
- Показати папку скетчів - відкрити системну папку з програмами;
- Додати файл - додати до проекту файл з даними або програмою;
- Імпортувати бібліотеку - підключити до програми бібліотеку зі списку встановлених.

Пункт меню **Сервіс** включає в себе допоміжні функції для роботи з самим мікроконтролером.

- Автоформатування - автоматична розстановка відступів, переносів рядків і т.п .;
- Архівувати скетч - архівація папки з програмою, і збереження архіву в вказане місце;
- виправити - кодування і перезавантажити
- Монітор порту - відкрити вікно для обміну даними з мікроконтролером;
- Плата - вибір поточної плати (в даному випадку Arduino Uno);
- Послідовний порт - вибір порту, до якого підключений;

Нарешті, меню **Довідка** містить докладний опис всіх функцій самого редактора Arduino IDE, а також всілякі команди і прийоми роботи з платформою Arduino.

2 . Меню іконок

-  Перевірити / Компілювати програму
-  Завантажити програму в Arduino
-  Створити нову програму
-  Відкрити існуючу програму
-  Зберегти програму
-  Монітор послідовного порту

3. Вкладки

Кожна програма для Arduino може складатися з декількох файлів. Для перемикання між цими файлами служить система вкладок в редакторі. Там же, можна створити нову вкладку, і асоціювати з нею файл в папці з проектом.

4. Вікно програми

Безпосередньо, текст програми створюється і редагується в головному вікні редактора. По суті, вікно редактора являє собою типовий текстовий редактор, з підсвічуванням конструкцій коду.

5. Вікно повідомлень

У самому низу редактора Arduino IDE є невелике вікно, що служить для виведення повідомлень про проблеми, які виникають в процесі компіляції програми, або під час завантаження програми в мікроконтролер.

1.1.5. Основи програмування Arduino

1.5.1. синтаксис

Програмування Arduino здійснюється на мові C++. Нижче приведені основні конструкції і умовності мови, необхідні для успішного проходження даного курсу.

1. Кожен вираз закінчується символом; крапка з комою. наприклад:

```
a = b + c;
```

2. Тіло функцій і складових операторів (if, else, for, while) відокремлюється фігурними дужками (Аналогічно BeginEnd в мові Pascal). наприклад:

```
if ( a>0 )  
{  
    b = a+1;  
}
```

3. Рядки відокремлюються звичайними подвійними лапками ". Приклад:

```
println ( "some text");
```

4. Символи відокремлюються одинарними лапками:

```
symbol = 'a';
```

5. Підключення бібліотек здійснюється за допомогою конструкції:

```
#include <math.h>
```

6. Коментарії в програмі починаються з символів // два прямих слеша. Приклад:

```
// це моя програма
```

1.5.2. типи даних

Оголошення змінної в мові c++ здійснюється за допомогою конструкції виду:

```
тип_змінної ім'я_змінної;
```

приклад:

```
int x, y; // оголошені дві змінні x і y, які мають цілий тип
```

1. Цілі числа

byte від 0 до 255

int від 32 768 до 32 767

word від 0 до 65535

long від 2 147 483 648 до 2 147 483 647

2. Дробові числа

float від 3.4028235E + 38 до 3.4028235E + 38

double еквівалентно float в поточній версії Arduino

3. Масиви

Масиви в C ++ задаються конструкцією типу:

тип_елемента імя_масива [розмір];

Приклад:

int numbers [10]; // задає масив з десяти цілих чисел

4. Рядки і символи

char символ;

Рядки в C ++ є масиви з елементами типу char. приклад:

char my_str [10]; // рядок з десяти символів

5. Інші типи

void порожній тип;

boolean false лібл true (брехня або істина).

1.5.3. Оператори

1. Оператори порівняння

== рівність
!= Неравність
< менше
<= Менше, або одно
> більше
> = Більше, або дорівнює

2. Умови

```
if (a > 0) {  
    ...  
    команди, що виконуються в разі істинності умови  
}  
else {  
    ...  
    команди, що виконуються в іншому випадку  
}  
3. Цикли  
for (k = 0; k < 3; k = k + 1) {  
    ...  
    команди, що виконуються на кожному кроці циклу  
}
```

У дужках послідовно вказується:

- початкове значення ітератора $k = 0$;
- умова продовження циклу $k < 3$ (поки ітератор менше трьох);
- дію над ітератором під час кожного кроку $k = k + 1$ (збільшуємо на одиницю на кожному кроці).

1.5.4. функції

```
тип_функції імя_функції (аргументи) {  
    команди, що виконуються в рамках функції  
    return результат_функції;  
}
```

тип_функції - тип значення. Наприклад, стандартна функція `sin` має тип значення, що повертається `float`.

ім'я_функції - будь-який рядок, що починається з букви, і містить тільки літери і символи підкреслення.

аргументи - перелік аргументів, які функція використовує для своїх дій.

результат_функції - прямилися або число, що визначає значення, що повертається функції.

Нижче наведено приклад функції, що підсумовує два цілих числа.

```
int sum (int a, int b) {  
int result;  
result = a + b;  
return result;  
}
```

1.5.5. мінімальна програма

Мінімальна програма, яка може запустити на Arduino складається всього з двох функцій:

```
void setup () {  
}  
void loop () {  
}
```

Перша функція setup викликається тільки один раз, після перезапуску Arduino. Друга loop, викликається нескінченне число разів під час роботи Arduino.

Завдання

Завдання 1. Підключення сенду до ПК.

Завдання 2. Підключення світлодіода.

Завдання 3с. Підключення світлодіода через шину живлення.