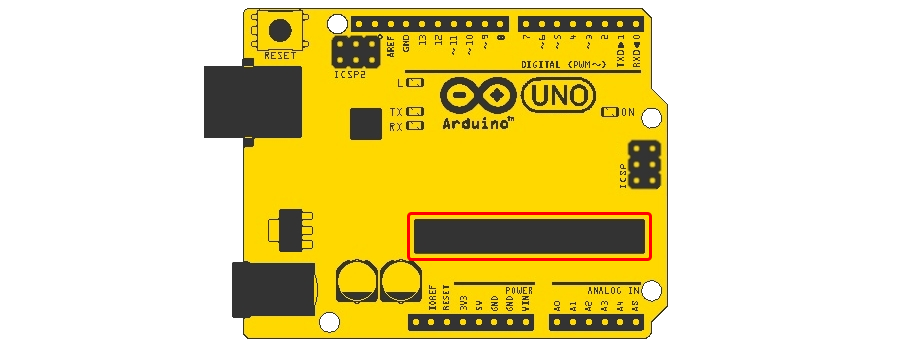
**Знайомство з Arduino**

Arduino — платформа, на якій можливо реалізувати величезну кількість ідей, як тих, що були створені вже до вас так і тих, які ви придумаєте самостійно. Ця невелика плата вміє генерувати як прості сигнали у вигляді п’яти або нуля вольт, логічних одиниці та нуля, так і трішки складніші. Якщо вміло користуватися цими двома положеннями, стане можливою реалізація різних інтерфейсів передачі даних. Ось наприклад інтерфейс SPI уже реалізували для вас. Окремі контакти можуть видавати сигнал ШІМ. Ще деякі уміють навпаки зчитувати показники, які на них подають. На платформі дуже легко створювати мобільні проекти, адже живити їх можна не лише від USB, а й від акумуляторів, батерейок чи генераторів. Контролює цими всіма можливостями ваша програма, яка записується у контроллер за допомогою програматора. Тож, я думаю, ви вловили тему «деб’ютного» уроку. Сьогодні ми поговоримо про те, з яких компонентів складається плата, для чого вони потрібні, що з ними варто робити, а що може зашкодити всьому проекту.

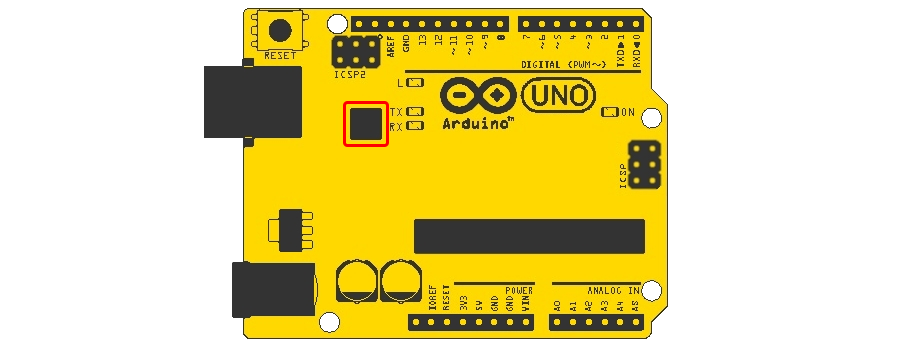
**Мікроконтроллер**

Почнемо з головного елементу, який ви програмуєте і який виконує всю роботу — мікроконтролера.



Мікроконтроллер – це та чорна мікросхема з багатьма ніжками, яка знаходиться на кожній платі Arduino. Саме у цей мікрочіп приходить і із нього виходить інформація, яку обробляє програма написана вами, яка також знаходиться в середині нього. Всі ці дані там зберігаються у вигляді нулів та одиниць, на практиці це просто закриті або відкриті транзистори. Всі решту елементи на платі називають обв’язкою. Обв’язка служить для того, щоб допомогти контроллеру взаємодіяти із зовнішніми елементами. Слід враховувати те, що на різних моделях Arduino, різні мікроконтроллери ATMega (числовий індекс). Виготовляє їх компанія ATMEL, а за числовим індексом можна визначити скільки на борту флеш пам’яті та на якій максимальній частоті (за нормального живлення) може працювати чіп.

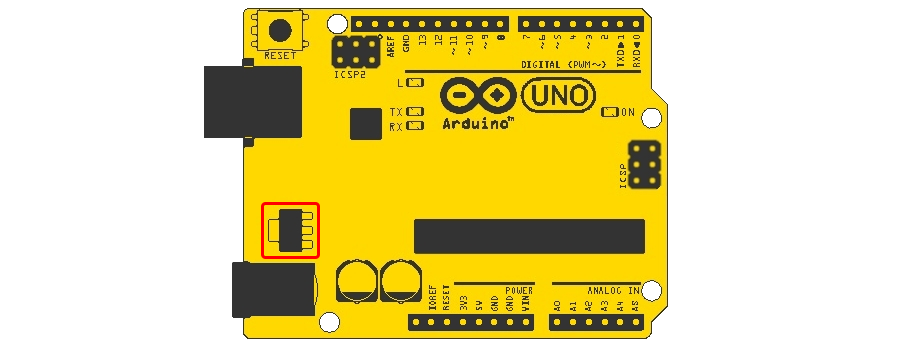
**Програматор**

Наступним важливим елементом на платі є ще один мікроконтроллер..  


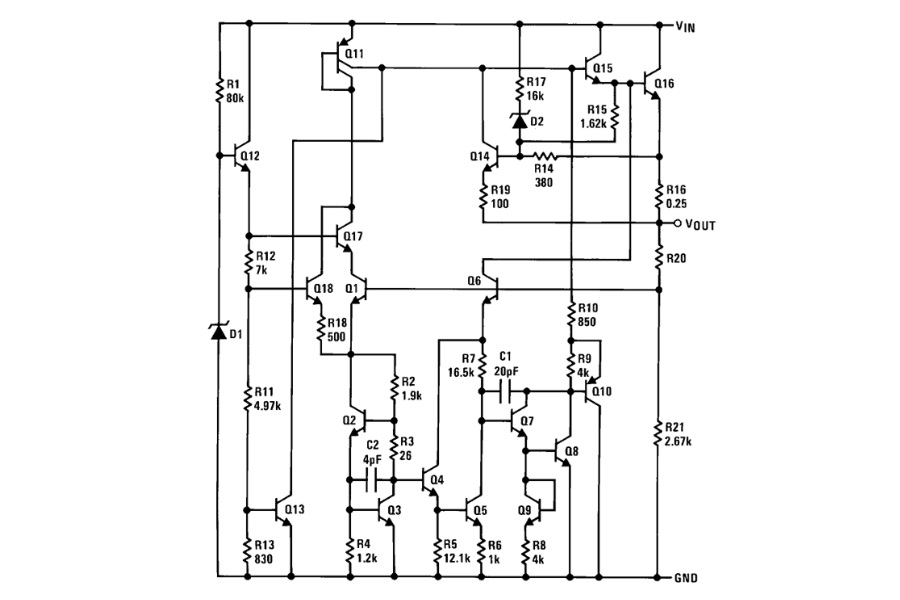
Навідміну від попереднього, цей служить як програматор, також на ньому реалізований інтерфейс передачі даних UART. Він також має свою обв’язку із резисторів та конденсаторів. Він сильно простіший ніж основний мікроконтроллер, оскільки не призначений для виконання важких завдань, у ньому також є записана програма, основною задачею якої є перенесення вашого коду у головний мікроконтроллер. За програматор та його прошивку вам турбуватися не потрібно, це вже зробили за вас, попередньо написавши програму та завантаживши її за допомогою іншого програматора ще на заводі.

**Стабілізатор**

Добре, давайте надалі розглядати будову на найбільш популярній платі — Uno. Ще одною мікросхемою, яку ви могли бачити, є стабілізатор напруги, їхні маркування можуть відрізнятися, проте суть у них одна — знизити вхідну напругу до 5-ти вольт, які потрібні для нормальної роботи мікроконтроллера.



У такого стабілізатора є три ніжки, три контакти. На один з них подається напруга яка може змінюватись з часом, прикладом може послужити звичайна батарейка типу крона, напруга на клемах у якої буде по мірі використання зменшуватись. Наступний контакт з’єднується з мінусом схеми. А з третього ми можемо, або в даному випадку мікроконтроллер може брати стабільних 5 вольт. У наступних уроках ми будемо говорити про те? Чому саме 5 вольт та про те, чому важливо, щоб вони були стабільними. А зараз можна подивитись на приблизну будову цієї маленької мікросхемки.



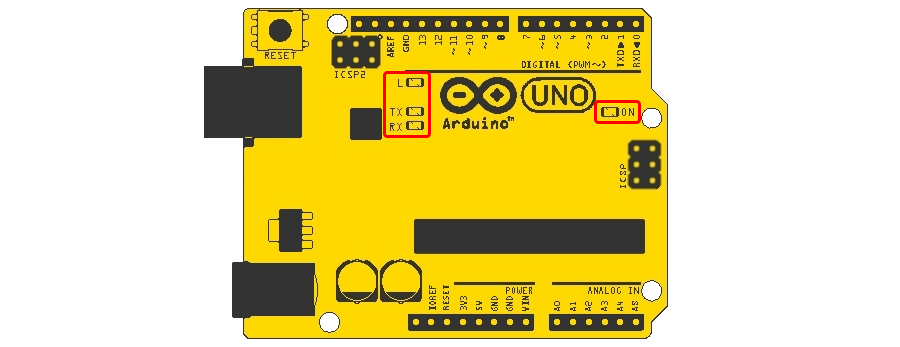
**Кварц**

Я думаю? що для всіх вас знайомий термін «Частота», переважно він асоціюється у вас із мобільною та комп’ютерною електронікою, а саме процесорами, відеокартами чи ОЗП. Так от, мікроконтроллер також працює на певній частоті. Будь яка операція МК (мікроконтроллера), давайте його далі так називати), складається із певних базових дій або ж тактів. Тож з’являється таке поняття як «тактова частота». Тобто така кількість тактів яку мікроконтроллер може виконати за секунду. У Arduino Mega та Arduino Uno ця позначка на рівні 16 МГц. Що ж це значить на практиці? Те, що контроллери встановлені на цих платах можуть виконати 16 000 000 базових операцій в секунду. Здавалося б, це дуже багато, проте не слід плутати базові операції з командами, написаними у середовищі програмування Arduino IDE, так як за однією простою командою, яку ви запишете у своєму скетчі може знаходитись сотня, тисяча, мільйон чи декілька мільйонів базових команд. Тобто для того, щоб виконати вашу команду, мікроконтроллеру потрібно буде у певній послідовності виконати сотні простіших, базових команд. Допомагає підтримувати частоту тактування ось цей невеличкий елемент який носить назву — кварцевий резонатор.



**LED індикатори**

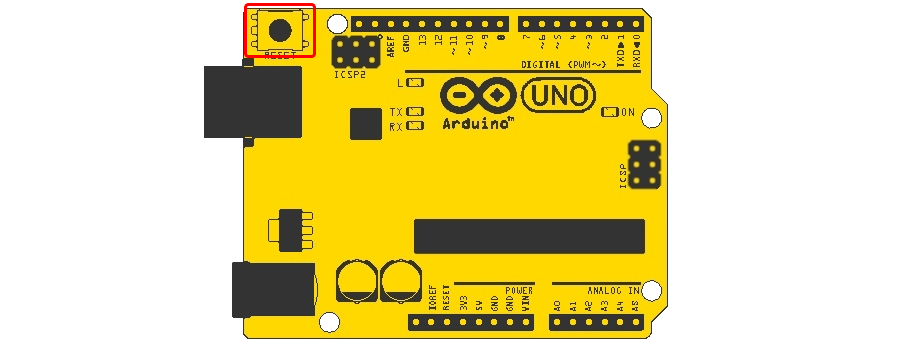
Всі ми любимо, коли на платах миготять якісь світлодіоди, показуючи, що плата працює… Зрештою, на це просто круто дивиться) Але просто так, за для краси, їх би не впаювали у схему, як і всюди, на Arduino вони також для чогось потрібні.



На платах Mega та Uno ми бачимо по чотири світлодіоди: L, TX, RX, ON. Почнемо з «L», він паралельно підключений до 13-го порту, тобто на самій платі ви маєте один світлодіод, який можете блимати просто подаючи «1» або «0» на 13 контакт. Наступні два «TX» та «RX» показують передачу чи прийом даних від хосту, відповідно. Один з прикладів, коли через USB до Arduino будуть приходити дані — буде блимати RX (Receiver Exchanges), а коли плата сама буде надсилати щось — про це буде сигналізувати TX (Transmitter Exchages). І останнє найпростіше, світлодіод з підписом «ON» — сигналізує про те, чи працює плата.

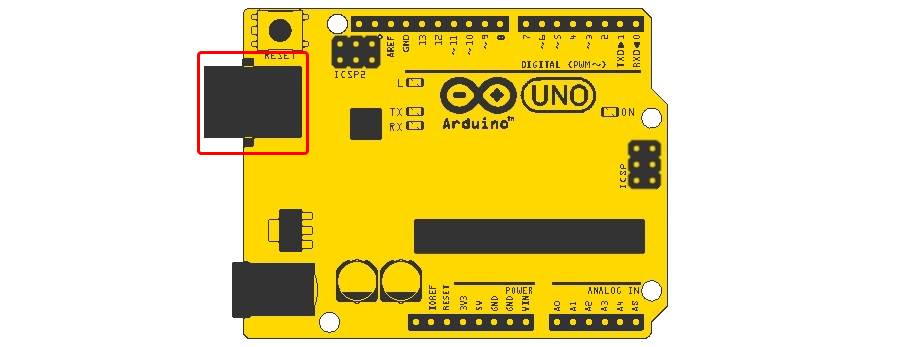
**Інша дрібнота**

Решта дрібних компонентів – це, як уже було раніше сказано, обв’язка – елементи, які допомагають працювати платі. О, точно, ще є кнопка, яка перезапускає плату.



**Порти**

І так, є у нас Arduino, яка багато що може. Тож що та куди підключати, щоб вже швидше приступити до навчання??? Порти. Тут їх багато і на різний смак. Для завантаження скетчів, а також передачі даних до/від комп’ютера через USB інтерфейс, тут знаходиться USB-B, такий ви могли бачити у своєму принтері.

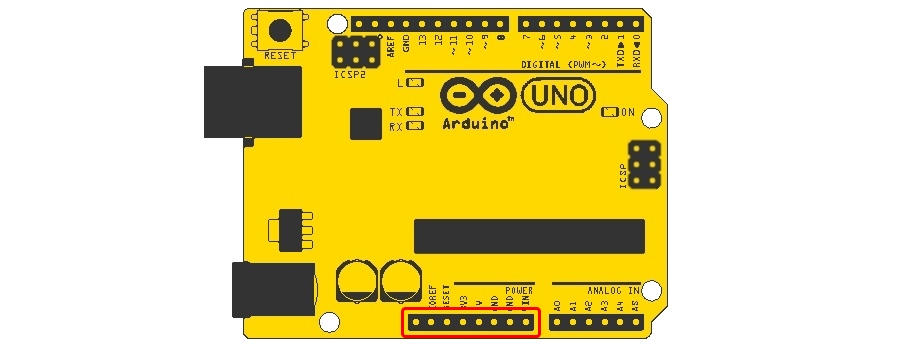


**Роз’єм живлення**



Якщо ж ви хочете, щоб схема живилась не від ПК, а скажімо, наприклад, від акумулятора, використовуйте цей роз’єм. Рекомендований вольтаж 7-12В, а максимально допустимі рамки 6-20В.

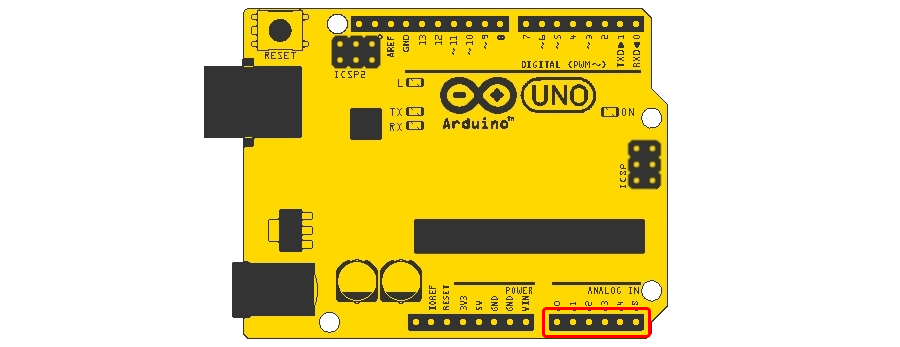
**Піни power**



Контакти «IOREF» та «AOREF» призначені для плат розширення і на даному етапі навчання не знадобляться. Контакт «RESET» виконує ту ж функцію, що і кнопка, для того, щоб перезапустити плату слід на нього подати короткочасний імпульс. Контакт «3,3V» дає для користування 3,3 вольта, які, наприклад, знадобляться для підключення Bluetooth модуля. Контакт «5V» я думаю здогадались))) За ним слідують два «Gnd», це — мінуси, земля. І останній «Vin». На ньому знаходиться така ж напруга, яку ви подаєте на роз’єм живлення.

**Аналогові входи**

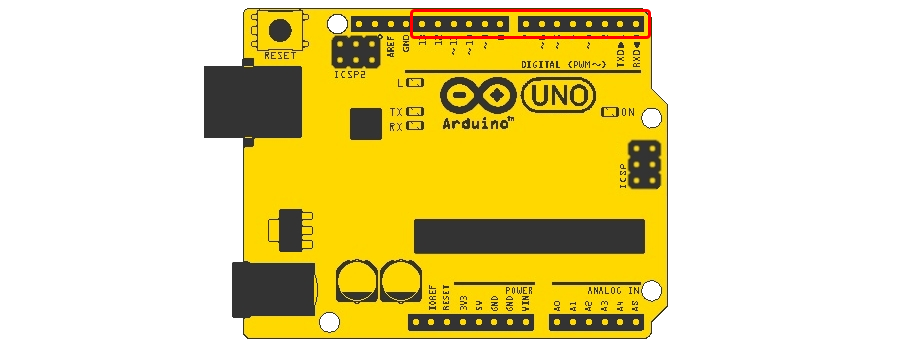
Наступна група контактів маркується «A0» — «A5» і являє собою аналогові піни входу.

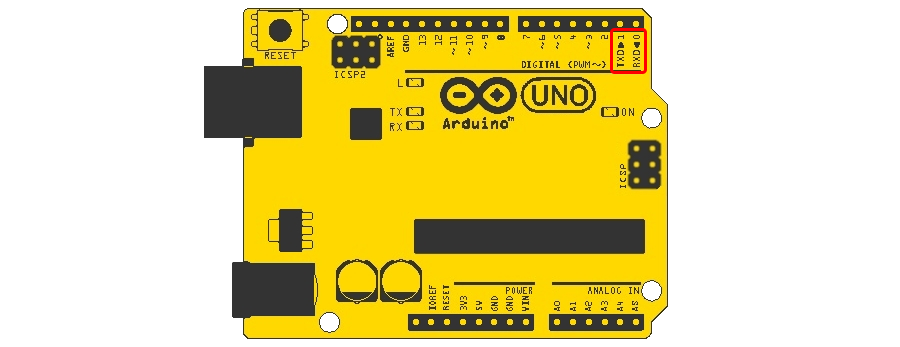


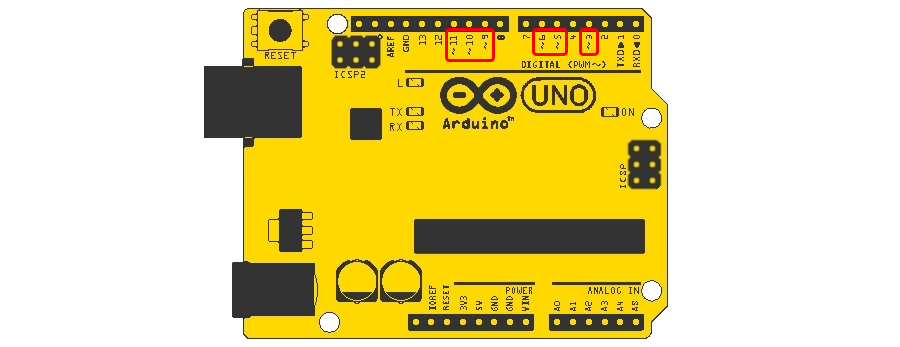
На них можна подавати напругу від нуля до 5-ти вольт, це можуть робити наприклад датчики. Цю напругу мікроконтроллер за допомогою АЦП перетворює у число від 0 до 1023. Тобто, подаємо нуль вольт і маємо цифровий нуль, з яким може працювати програма, подаємо 5 вольт — отримуємо число 1023. Так наприклад, аналогові датчики в залежності від умов дають нам різну напругу, яку ми подаємо на ці піни і отримуємо число, яке множимо, ділимо, віднімаємо, додаємо, в залежності від потреб. В кінцевому результаті за допомогою датчика, цих контактів та певних обрахунків ми отримуємо такі величини як: вага, кут нахилу, рівень освітлення, положення потенціометра та інші. Слід зазначити що ця група може лише приймати струм, відповідно в зворотньому напрямку вона не працює. Тобто у програмі НЕ можна записати число n і чекати, що на вході з’явиться аналоговий сигнал, який можна буде використовувати.

**Цифрові I/O піни**

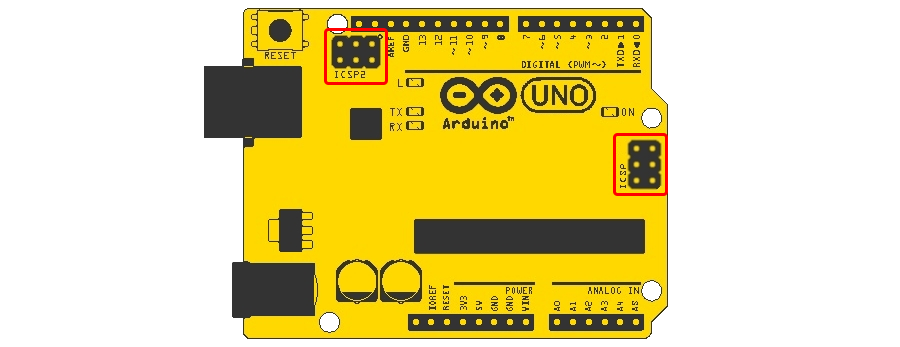
Слідуючими у нас ідуть дві групи цифрових контактів, нумерація яких розпочинається із нуля та закінчується тринадцяткою.



Кожен із них може видавати логічну одиницю або логічний нуль, тобто п’ять або нуль вольт відповідно. Проміжних значень не має. Дуже хорошим є те, що кожен з них може працювати як у режимі виходу (видавати логічні сигнали) так і у режимі входу (зчитувати логічні сигнали), що знову ж таки знадобиться для підключення різноманітних датчиків, які спілкуються за допомогою нулів та одиниць. Придивившись уважніше ви побачите ще якісь незрозумілі символи поблизу номерів цих контактів, а саме «TX», «RX» та «PWM». Про світлодіоди «TX» та «RX» ми вже говорили, конкретніше про те, що вони сигналізують передачу даних через UART TTL. Тож тепер логічно подумаємо, для чого ці позначки ще й біля цифрових контактів?  
  
Саме так, на них, на цих двох контактах, реалізований цей інтерфейс. За допомогою цих двох контактів ви з часом зможете підключити Bluetooth модуль. Уявляєте, наскільки це класний інтерфейс, ви зайняли лише два піни Arduino (не рахуючи живлення) і вже можете передавати на плату та з плати будь які дані з іншого чи на інший пристрій з Bluetooth, яким може бути наприклад ваш смартфон, ноутбук чи наприклад друга плата Arduino! Що ж до «PWM», то про це поговоримо трішки пізніше. Якщо коротко, то PWM (українською Широтно-Імпульсна модуляція, ШІМ) — це такий сигнал, коли логічна одиниця та логічний нуль чергуються, а їхньою тривалістю можна керувати.



Ось ці шість «штирьків» — це реалізація ще одного інтерфейсу передачі даних який носить назву ICSP.

  
Про нього також трохи згодом. Але коротко кажучи за допомогою нього можна програмувати інші мікроконтроллери.

Не менш цікавим інтерфейсом є і I2C, за допомогою якого можливо здійснити підключення вашої старої мишки чи клавіатури до Arduino.

Що ж, стаття вже затягнулась, але все ж скажу ще декілька реально важливих речей на які потрібно зважати. Будьте акуратні із проводами які підключаєте, НЕ залишайте їхні не підключені кінці в повітрі, адже достатньо буде якомусь з напругою торкнутися якогось місця пайки на платі чи ніжки мікросхеми й є шанс попрощатися із платою. Особисто я так спалив Arduino UNO та Bluetooth модуль))) А також будьте обережними із вологою та статикою.

Дякую за привернену увагу)))