

УДК 004.75

**Микола Серпученко,  
Ірина Клименко****ОГЛЯД СУЧАСНИХ ТЕХНОЛОГІЙ СТВОРЕННЯ  
ЦИФРОВИХ ПРИСТРОЇВ НА БАЗІ МІКРОПРОЦЕСОРНИХ СИСТЕМ****THE EXPLORATION OF MODERN TECHNOLOGIES  
OF THE CREATION OF DIGITAL DEVICES BASED  
ON MICROPROCESSOR SYSTEMS**

У цій статті розглядається сучасний ринок мікроелектроніки та досліджуються архітектури сучасних мікроконтролерів та мікропроцесорних систем. В статті визначається поняття сучасної мікропроцесорної системи та розглядаються їх технологічні можливості і області застосування.

**Ключові слова:** мікропроцесорна система, система на кристалі, *Arduino*.

This article analyzes the modern market of microelectronics and explores the architecture of modern microcontrollers and microprocessor systems. The article defines the concept of a modern microprocessor system and examines its technological capabilities and application areas.

**Key words:** microprocessor system, System on a Chip, *Arduino*.

**Актуальність теми дослідження.** Тенденції розвитку сучасних технологій виготовлення процесорів і їх застосування з кожним роком набирають все більших обертів. Застосовуються нові технології, збільшується кількість ядер на одному кристалі, зростає розрядність процесорів, збільшується кеш пам'ять усіх рівнів, застосовуються нові набори інструкцій і багато іншого. Метою даної статті є огляд сучасного ринку мікроелектроніки та дослідження архітектур сучасних мікроконтролерів та мікропроцесорних систем. В статті визначається поняття сучасної мікропроцесорної системи та розглядаються їх технологічні можливості і області застосування. Також, в статті наведений приклад програмування обчислювальної платформи *Arduino*.

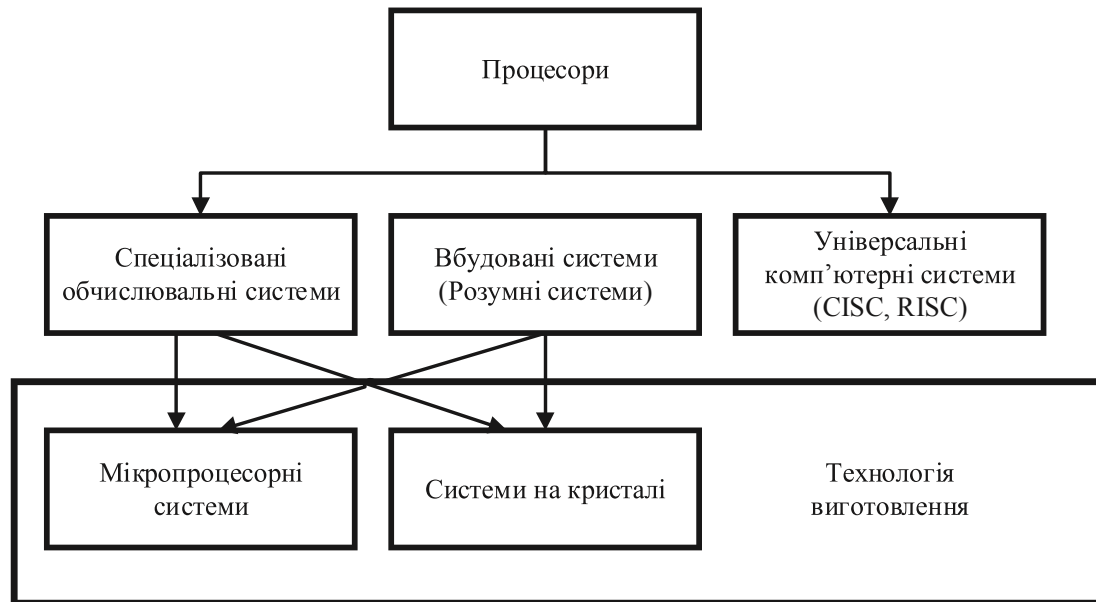
**Постановка проблеми.** На сьогоднішній день існує багато різних мікропроцесорних систем і їх досить важко класифікувати. Сучасний ринок мікроелектроніки швидко змінюється, що ускладнює завдання його дослідження.

**Аналіз останніх досліджень і публікацій.** За останні роки з'являються все більше нових різноманітних статей, присвячених мікропроцесорним системам та побудованих на них цифрових пристроях. Однак, сучасні технології використання таких систем досі розкриті не повністю.

**Постановка завдання.** Завдання полягає в аналізі сучасного ринку мікроелектроніки та вивченні архітектури сучасних мікроконтролерів та мікропроцесорних систем. Іншою метою статті є визначення поняття сучасної мікропроцесорної системи та вивчення її технологічних можливостей.

**Викладення основного матеріалу.** Сучасні тенденції розвитку мікропроцесорних систем можна представити через узагальнену класифікацію процесорів за областю їх застосування в комп'ютерних системах (рис. 1).

Представлена класифікація є результатом сучасної тенденції універсалізації комп'ютерних систем. Така універсалізація призвела до широкого поширення кластерних суперкомп'ютерів, або так званих високопродуктивних кластерів. Серед очевидних переваг, обумовлених особливостями архітектури та програмного забезпечення кластерних систем – їх відносна дешевизна, простота програмування, простота адаптації програмного забезпечення, відпрацьованого на персональних комп'ютерах.



*Рис. 1.* Узагальнена класифікація процесорів

Універсальні обчислювачі або просто процесори це мікросхеми, які дозволяють виконувати дії над інформацією, такі, як логічну обробку та арифметичні обчислення. Сучасні універсальні процесори обробляють інформацію, представлену на вхідних мовах високого рівня та використовуються в ПК. Універсальні процесори характеризуються широкою областю застосування. Серед них розрізняються процесори з універсальним набором команд (CISC) та скороченим набором команд (RISC).

У CISC (Complex Instruction Set Computer) процесорах використовуються складні команди змінної довжини з великим різноманіттям режимів адресації операндів. Такі команди нагадують оператори мов високого рівня. Такі процесори базуються на фон-Неймановській архітектурі, яка передбачає принцип спільного зберігання команд і даних в пам'яті комп'ютера.

У RISC (Reduced Instruction Set Computer) процесорах розмір команди постійний і, як правило, обмежений одним словом оперативної пам'яті. У таких процесорах усі операції виконуються тільки над операндами, які знаходяться в регістрах процесора. Для завантаження операндів у регістри з оперативної пам'яті існують спеціальні команди. Такі процесори мають гарвардську архітектуру, в якій пам'ять даних і пам'ять команд розділені. Так само, шина даних та шина команд – це два окремі канали.

В умовах сучасного науково-технічного прогресу, вимоги завдань до комп'ютерних систем безперервно зростають, вимагаючи їх ускладнення і

удосконалення. Це підвищує вартість, складність розробки і експлуатації задаче-орієнтованих комп'ютерних систем і ставить під питання економічну і практичну доцільність їх реалізації. Враховуючи, також, що при вирішенні задач інших класів продуктивність обчислень різко знижується.

Спеціалізовані та убудовані обчислювальні системи (ОС) є об'єктно-орієнтованими, що забезпечують найкращу продуктивність за рахунок орієнтації архітектури на рішення специфічної задачі. В результаті абсолютної відповідності цільового призначення, кінцевий користувач завжди отримує максимальну продуктивність обчислень. Сучасна тенденція розробки спеціалізованих та убудованих обчислювальних систем – це широке втілення програмованих логічних інтегральних схем, що надає нових можливостей для удосконалення таких систем – розміщення більш складних пристроїв на меншій площині кристала, економія внутрішніх ресурсів ПЛІС, адаптація до різних класів задач, універсалізація, мультизадачний режим функціонування.

*Спеціалізовані та убудовані обчислювальні системи* використовують для створення різного роду інтелектуальних зовнішніх пристроїв, здатних самостійно виконувати ряд функцій з обробки інформації. Важливою особливістю таких ОС є те, що на них може реалізуватися система команд, найбільш зручна для вирішення певного класу задач. Так, ОС, яка використовується в інтелектуальному пристрою зв'язку з об'єктом, може мати систему команд, орієнтовану на роботу з модулями вводу-виводу аналогових і дискретних сигналів, а також на первинну обробку інформації і реалізацію законів регулювання. Це дозволяє робити такі пристрої особливо ефективними саме для даного застосування.

Фізично, спеціалізовані та убудовані системи можуть бути малими як, наприклад, портативні пристрої (цифрові годинники або MP3-плеєри), або великими, як світлофори, пристрої керування виробництвом та системи керування ядерними електростанціями. Відповідно, складність може бути як низькою – з одним чипом керування, так і дуже високою – з багатьма пристроями, периферійними приладами, комп'ютерними мережами тощо.

*Системи на кристалі* або *System on a Chip (SoC)* будуються на програмованих логічних інтегральних схемах (ПЛІС). На відміну від звичайних цифрових мікросхем, логіка роботи ПЛІС не визначається цілком при виготовленні, а створюється шляхом програмування.

*Розумні системи*, які також називаються «*Embedded Systems*» – це спеціалізована комп'ютерна система, повністю убудована всередині керованого пристрою, призначена для виконання обмеженої кількості функцій [1].

### **Технології виготовлення спеціалізованих та убудованих обчислювальних систем**

Існує два широко застосовуваних поняття в досліджуваній області – це мікроконтролери або мікропроцесори і мікропроцесорні системи.

*Мікропроцесор (Мр)* або мікроконтролер являє собою пристрій, який реалізує основні елементи комп'ютерної системи на єдиній інтегральній схемі.

*Інтегральна схема* являє собою напівпровідниковий кристал, який містить набір електронних компонентів, з'єднаних між собою для виконання певної цифрової функції.

*Мікропроцесорна система (MPU)* створюється на одній або на декількох інтегральних схемах діючих як єдине ціле.

Сучасні мікропроцесорні системи включають в себе функціональні можливості годинника, центрального процесору (*CPU*), арифметико-логічного блоку (*ALU*), блоку з плаваючою комою (*FPU*), блоку керування (*CU*), блоку керування пам'яттю (*MMU*), переривань, інтерфейсів вводу-виводу, і кеш-пам'яті. Спеціалізовані мікропроцесори можуть також використовуватися в якості графічних процесорів (*GPUs*), блоків обробки сигналів (*DSPs*), блоку нейронної обробки (*NPU*) і мікроконтролерів.

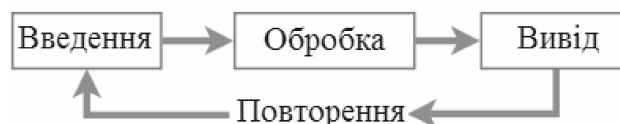
*Система на кристалі (SoC)* – обчислювальна система, яка містить всі компоненти комп'ютерної системи, включаючи додаткові функції, які зазвичай надаються допоміжними мікросхемами. Вони можуть включати в себе такі речі, як *Wi-Fi*, *Ethernet*, *SD*-карта, *ADC*, *DAC*, *LCD* драйверів і ПЛІС. *SoCs* здатні працювати з повноцінними сучасними операційними системами і з усіма їхніми функціями.

### **Функціональність мікропроцесорних систем**

У найширшому розумінні базова функціональність мікропроцесора полягає в:

- безперервному зчитуванні цифрових даних, що складаються з інструкцій і, можливо, значень;
- виконанні їх шляхом інтерпретації інструкцій; виконанні певної операції над значеннями;
- виведення результату.

Узагальнена базова функціональність мікропроцесорної системи зображена на рис. 2.



**Рис. 2.** Узагальнена базова функціональність мікропроцесорної системи

У той час як базові функції властиві усім мікропроцесорам, вони сильно розрізняються за типом і розміром даних, які вони обробляють, за типами операцій, які вони підтримують, за те, як вони виконують ці операції, за їх призначенням і за їх характеристикам продуктивності.

### **Технічні характеристики мікропроцесорних систем**

Технічні характеристики мікропроцесорів засновані на мікроархітектурі убудованого процесора, задіяної напівпровідникової технології і властивості всієї системи. Нижче наведені деякі загальні специфікації:

- технологія – напівпровідникова технологія, яка використовується для створення MPU (наприклад, CMOS, BiCMOSiTTL)
- процес – розмірі правила дизайну. Розмір функції сам по собі (наприклад, 10 мкм).
- мікроархітектура – функціональний опис базової схеми мікропроцесора.
- розмір слова – розмір слова мікропроцесора зазвичай відноситься конкретно до використовуваному розміру слова даних – тобто максимальній

довжині операнда, використовуюваного для керування стандартними цілочисельними значеннями.

- кількість ядер – кількість фізичних ядер, убудованих в чіп.
- порядок байтів – порядок байтів, який використовується мікропроцесором при роботі з багатобайтовими значеннями.
- базова частота – внутрішня робоча частота ядра ЦП. Це один з багатьох параметрів, які використовуються для оцінки продуктивності мікропроцесора.
- пакет – це фізичний корпус мікропроцесора. Найчастіше поєднується з гніздом, що представляє собою з'єднання, які знаходяться на самій платі, де вставлений пакет.

### **Архітектура мікропроцесорних систем**

Мікропроцесори виготовляються з кремнієвого матеріалу, який містить крихітні електричні компоненти, убудовані на поверхню мікрочіпа. У стандартний мікропроцесор зазвичай входять наступні компоненти:

*Арифметичний і логічний блок (ALU)* виконує математичні обчислення, такі як віднімання, складання, ділення і булеві функції. Булеві функції – це тип логіки, використовуваної для функціональних елементів. *ALU* також виконує порівняння і логічне тестування. Процесор передає сигнали в *ALU*, який інтерпретує інструкції і виконує обчислення.

*Регістри* – тимчасові місця зберігання даних. Ці області пам'яті підтримують дані, такі як інструкції комп'ютера, адреси сховища, символи та інші дані. У деяких командах комп'ютера може знадобитися використання певних регістрів як частина команди. Кожен регістр має певну функцію, таку як регістр команд, лічильник, регістр акумулятора і регістр адреси пам'яті.

*Блоки керування* приймають сигнали від ЦП, які інструктують блок керування переміщати дані з одного мікропроцесора на інший. Блок керування також направляє арифметичний і логічний блок. Блоки керування складаються з декількох компонентів, таких як декодер, схеми синхронізації і логіки керування. Працюючи разом, ці пристрої передають сигнали в певні місця на процесорі. Наприклад, декодер отримує команди від додатка. Декодер інтерпретує інструкції і вживає заходи. Він відправляє сигнали в *ALU* або направляє регістри для виконання певних завдань. Блок логіки керування передає сигнали в різні секції мікропроцесора і регістрів, які інформують ці компоненти про виконання дій. Годинники посилають сигнали, які синхронізують і забезпечують своєчасне виконання команд і процесів.

*Шини.* Мікропроцесори мають систему шин, які переміщують дані. Шини відносяться до класифікацій провідників, які мають конкретні завдання і функції. Шина даних передає дані між центральним процесором і оперативною пам'яттю (ОЗУ) – основний пам'яттю комп'ютера. Шина керування відправляє інформацію, необхідну для координації і керування декількома завданнями. Адресна шина передає адресу між ЦП і ОЗП для оброблюваних даних.

*Кеш пам'ять.* Деякі вдосконалені мікропроцесори мають кеші пам'яті, які зберігають останні дані, які використовуються ЦП. Накопичувачі пам'яті прискорюють обчислювальний процес, тому що ЦПУ не повинен звертатися до більш повільного ОЗУ для вилучення даних. Багато комп'ютерів мають кеші 1

або 2 рівня, деякі системи мають кеші рівня 3. Рівень кеша вказує порядок, в якому ЦП перевіряє дані, починаючи з рівня 1. Виробники часто інтегрують кеші рівня 2 і рівня 3 в мікропроцесор, що збільшує швидкість обробки.

### **Сімейства мікропроцесорних систем**

Сімейство мікропроцесорів є групою мікропроцесорів, мікроконтролерів, мікросхем або компонентів, розробленої конкретними виробником, які поділяють ту ж загальну архітектуру і набір функцій. У конкретному сімействі чіпи можуть бути додатково класифіковані відповідно до їх моделлю, яка може мати різні можливості, тактові частоти і додатки. Моделі можуть бути далі розділені на ступені, які є незначними уточненнями і налаштуваннями для виправлення і підвищення стабільності і надійності моделі.

### **Мікропроцесорні системи сімейства *Arduino***

*Arduino* – це обчислювальна платформа, яка представляє собою комплекс апаратних і програмних компонентів. Ця платформа дає змогу розробляти інтерактивні цифрові системи, які приймають дані від різних комутаторів, датчиків та індикаторів для керування двигунами та іншими фізичними об'єктами [1].

Основними компонентами платформи *Arduino* є мікроконтролер та плата з компонентами периферії, такої як елементи вводу/виводу. Програмна частина складається з середовища розробки *Processing/Wiring* на мові програмування, що є підмножиною C/C++. *Arduino* може використовуватися як для створення автономних інтерактивних цифрових систем, в тому числі таких, що працюють в режимі реального часу, так і підключатися до програмного забезпечення, яке виконується на комп'ютері (наприклад: *Processing*, *Adobe Flash*, *Max/MSP*, *Pure Data*, *SuperCollider*). Інформація про плату (рисунок друкованої плати, специфікації елементів, програмне забезпечення) знаходяться у відкритому доступі і можуть бути використані тими, хто бажає створювати плати власноруч.

### **Апаратна частина плат *Arduino***

Плата *Arduino* складається з мікроконтролера *AtmelAVR*, а також елементів обв'язки для програмування та інтеграції з іншими пристроями. На багатьох платах наявний лінійний стабілізатор напруги +5В або +3,3В. Тактування здійснюється на частоті 16 або 8 МГц кварцовим резонатором. У мікроконтролер записаний завантажувач (*bootloader*), тому зовнішній програматор не потрібен.

На концептуальному рівні усі плати програмуються через RS-232 (послідовне з'єднання), але реалізація даного способу різниться від версії до версії. Новіші плати програмуються через USB, що можливо завдяки мікросхемі конвертера *USB-to-SerialFTDI FT232R*. У версії платформи *Arduino Uno* як конвертер використовується контролер *Atmega8* у SMD-корпусі. Дане рішення дозволяє програмувати конвертер таким чином, щоб платформа відразу розпізнавалася як миша, джойстик чи інший пристрій за вибором розробника зі всіма необхідними додатковими сигналами керування. У деяких варіантах, таких як *ArduinoMini* або неофіційній *Boarduino*, для програмування потрібно підключити до контролера окрему плату USB-to-Serial або кабель.

Плати *Arduino* дозволяють застосовувати значну кількість виводів мікроконтролера як вхідні/вихідні контакти у зовнішніх схемах. Наприклад, у платі *Decimila* доступно 14 цифрових входів/виходів, 6 із яких можуть

генерувати ШІМ-сигнал, і 6 аналогових входів. Ці сигнали доступні на платі через контактні площадки або штирові роз'єми. Також існує багато різних зовнішніх плат розширення, які називаються «*shields*» («щити»), які приєднуються до плати *Arduino* через штирові роз'єми.

Незважаючи на те, що платформу *Arduino* не вийде використовувати в якості комп'ютера, її низькі вимоги потужності дозволяють їй ефективно контролювати інші пристрої.

### ***Raspberry Pi***

*Raspberry Pi* – це недорогий комп'ютер розміром з пластикову карту, який можна підключати до монітора комп'ютера або телевізора. Керувати ним можна за допомогою стандартної клавіатури і миші. З його допомогою можна робити все, що і на звичайному комп'ютері – переглядати веб-сторінки, програвати відео високої роздільної здатності, створювати електронні таблиці, редагувати текстові документи і грати в ігри [2].

### ***Beaglebone***

Комп'ютер *Beaglebone* аналогічний *Raspberry Pi* в розмірах, вимогах потужності і застосуванні. *Beaglebone* перевершує *Raspberry Pi* в обчислювальній потужності, тому є оптимальним вибором для додатків з більш високими вимогами [3].

### **Приклад програмування мікропроцесорної системи *Arduino***

Програми для систем *Arduino* розробляються на C++ подібній мові. Будь-яка така програма складається, що найменше, з двох обов'язкових функцій: *setup()* та *loop()*.

Функція *setup()* запускається один раз, після кожного включення живлення або скидання плати *Arduino*. У тілі даної функції пишеться код для створення змінних, установки режиму роботи цифрових портів, та інших ініціалізацій. Цей механізм буде розглянуто у подальших прикладах.

Функція *loop()* в нескінченному циклі послідовно раз по раз виконує команди, які описані в її тілі. Тобто після завершення функції знову відбудеться її виклик.

Розглянемо, для прикладу, просту програму, що змушує світлодіод проблискувати:

```
void setup()
{
  pinMode(13, OUTPUT);
}
void loop()
{
  digitalWrite(13, HIGH);
  delay(100);
  digitalWrite(13, LOW);
  delay(900);
}
```

Кожен вираз – це наказ процесору зробити щось. Вирази в рамках одного блоку виконуються один за одним, суворо по порядку без пауз і перемикань. Тобто кожен конкретний блок коду читати зверху вниз, щоб зрозуміти що відбувається. Самі блоки також мають свій порядок виконання:

1. Після ввімкнення *Arduino*, або натискання кнопки *RESET*, викликається функція *setup*. Тобто вирази в ній виконуються послідовно підряд.

2. Як тільки робота *setup* завершується, відразу ж викликається функція *loop*.

3. Після того, як робота *loop* завершується, відразу функція *loop* викликається ще раз і так до нескінченності.

Розглянемо докладніше кожен вираз даного коду і з'ясуємо чому виконання цієї програми призводить до мерехтіння світлодіода. Як відомо, піни *Arduino* можуть працювати і як виходи і як входи. Коли є необхідність чимось керувати, тобто видавати сигнал, потрібно перевести керуючий пін в стан роботи на вихід. У нашому прикладі керування світлодіодом відбувається на тринадцятому піні, тому тринадцятий пін перед використанням потрібно зробити виходом. Це робиться виразом в функції *setup*:

```
pinMode(13, OUTPUT);
```

Вирази бувають різними: арифметичними, деклараціями, визначеннями та умовними. В даному випадку в виразі здійснюється виклик функції. В *setup* викликається функцію з ім'ям *pinMode*. Вона встановлює заданий за номером пін в заданий режим: вхід або вихід. Номер піна та режим вказується в круглих дужках, через кому, відразу після імені функції. У нашому випадку необхідно, щоб тринадцятий пін працював як вихід. *OUTPUT* означає вихід, *INPUT* – вхід. Уточнюючі значення, такі як 13 і *OUTPUT* називаються аргументами функції.

Розглянемо функцію *loop*:

```
void loop()
{
  digitalWrite(13, HIGH);
  delay(100);
  digitalWrite(13, LOW);
  delay(900);
}
```

Функція *loop*, як зазначалося, викликається відразу після *setup*. І викликається знову і знову після свого завершення. Функція *loop* називається основним циклом програми і ідеологічно призначена для виконання корисної роботи. У нашому випадку корисна робота – це миготіння світлодіодом.

Дослідимо вирази по порядку. Отже, перший вираз – це виклик убудованої функції *digitalWrite*. Вона призначена для подачі на заданий пін логічного нуля (*LOW*, 0 вольт) або логічної одиниці (*HIGH*, 5 вольт) У функцію *digitalWrite* передається 2 аргументи: номер піна і логічне значення. У підсумку, в першу чергу запалюється світлодіод на тринадцятому піні, після подання на нього 5 вольт. Як тільки це зроблено процесор моментально приступає до наступного виразу. В даному випадку – це виклик функції *delay*. Функція *delay* – це убудована функція, яка змушує процесор зробити затримку на певний час. Ця функція приймає лише один аргумент: час в мілісекундах, на який слід зробити затримку. У цьому випадку це 100 мс. Під час затримки стан системи не змінюється, тобто світлодіод продовжує горіти. Як тільки 100 мс закінчуються, процесор одразу переходить до наступного виразу. У цьому прикладі це знову виклик убудованої функції *digitalWrite*. Однак, цього другим аргументом передається значення *LOW*. Тобто на тринадцятому піні встановлюється логічний нуль, подається 0 вольт, та

світлодіод згасає. Після того, як світлодіод згас виконується наступний виразу. Це знову виклик функції *delay*. Цього разу робиться затримка на 900 мс. Як тільки затримка закінчується, функція *loop* завершується. Після завершення вона викликається ще раз і все відбувається знову.

### Висновки

На сьогоднішній день мікропроцесорні системи широко використовуються в багатьох пристроях керування та системах з, як правило, малим споживанням енергії. Вони є актуальними у пристроях, що повинні мати відносно невеликі розміри, оскільки мають широкий функціонал можливостей, при своїй помірній ціні. На прикладі програмування мікропроцесорної системи *Arduino* була показана доступність таких систем для користувачів, що не мають поглиблених знань у схемотехніці та мікропроцесорній архітектурі.

### Список літератури

1. В. Н. Кинягин, М. С. Липецкая, И. Б. Андреев «Умные» среды, «Умные» системы, «Умные» производства, 2012. – с. 9-10.
2. Arduino official site, 2018. [Електронний ресурс] – Режим доступу: <https://www.arduino.cc/en/Guide/Introduction>. – Дата звернення: 28.05.2018.
3. Офіційний сайт обчислювальної платформи Raspberry Pi, 2018. [Електронний ресурс] – Режим доступу: <https://www.raspberrypi.org/about/>. Дата звернення: 27.05.2018.
4. Офіційний сайт обчислювальної платформи Beaglebone, 2018. [Електронний ресурс] – Режим доступу: <http://beagleboard.org/about>. Дата звернення: 25.05.2018.

### ДОВІДКА ПРО АВТОРІВ

Серпученко Микола Вадимович – студент, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Serpuchenko Mykola – student, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: [nikolayserpuchenko@gmail.com](mailto:nikolayserpuchenko@gmail.com)

Клименко Ірина Анатоліївна – д.т.н., доцент, професор кафедри обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Klymenko Iryna –Doctor of Technical Sciences, Professor, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: [ikliryna@gmail.com](mailto:ikliryna@gmail.com)

**MykolaSerpuchenko,  
IrynaKlymenko**

## **THE EXPLORATION OF MODERN TECHNOLOGIES OF THE CREATION OF DIGITAL DEVICES BASED ON MICROPROCESSOR SYSTEMS**

**Relevance of the research topic.** The tendency of the development of modern processor manufacturing technologies and their applying evolves every year. There are used new technologies; the number of cores on one crystal increases, the processor runs faster, cache memory of all levels increases, new sets of instructions are used, and more. This article analyzes the modern market of microelectronics and explores the architecture of modern microcontrollers and microprocessor systems. The article defines the concept of a modern microprocessor system and examines its technological capabilities and application areas. Also, there is given an example of the programming of the Arduino computing platform.

**Formulation of the problem.** There are too many different microprocessor systems and it is quite difficult to categorize them all.

**The analysis of recent research and publications.** Over the past years more and more articles are devoted to exploration of the microprocessor systems, in particular, based on them digital devices. However, the modern technologies of usage these systems are not covered enough.

**Setting objectives.** The task is to analyze the modern market of microelectronics and explore the architecture of modern microcontrollers and microprocessor systems. One more aim of the article is to define the concept of a modern microprocessor system and examine its technological capabilities.

**Presentation of the main material.** The article has showed the generalized classification of the modern computing systems. There have been also explained the functionality of microprocessor systems and their generalized structure and architecture. The example of programing microprocessor system Arduino has clarified the way of usage such systems.

**Conclusions.** The modern microprocessor systems are widely used in many control systems and systems with generally low energy consumption. They are relevant in devices that should be relatively small in size, since they have a wide range of functionality at a moderate price. In the programming example of the Arduino microprocessor system, there has been shown the availability of such systems for users without advanced knowledge in circuit design and microprocessor architecture.

**Key words:** microprocessor system, System on a Chip, Arduino.