

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Одеська національна академія харчових технологій
Навчально-науковий інститут комп'ютерних систем і технологій
"Індустрія 4.0" ім. П.М. Платонова
Факультет Комп'ютерної інженерії, програмування та кіберзахисту

**XIX Всеукраїнська науково-технічна конференція
молодих вчених, аспірантів та студентів**

**“СТАН, ДОСЯГНЕННЯ І ПЕРСПЕКТИВИ
ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ”**

Матеріали конференції. Частина 2



Одеса
22 квітня 2019 р.

Стан, досягнення і перспективи інформаційних систем і технологій / Матеріали XIX Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 22 квітня 2019 р. - Одеса, Видавництво ОНАХТ, 2019 р. - 68 с.

Збірник включає матеріали доповідей її учасників, які об'єднані по секціях кафедр: комп'ютерної інженерії (КІ), інформаційних технологій та кібербезпеки (ІТтаКБ).

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

Організаційний комітет

Голова – д.т.н., проф., **Єгоров Б.В.**, ректор ОНАХТ.

Співголови:

Поварова Н.М. – к.т.н., доц., проректор з наукової роботи ОНАХТ,
Котлик С.В. – к.т.н., доц., директор ННІКСіТ "Індустрія 4.0" ОНАХТ,
Даріуш Долива, д.математичн.наук, уповноважений декана факультету Інформатики УІтаПЗ, м. Лодзь, Польща,
Ковалюк Т.В. - к.т.н., доц. кафедри АСОІтаУ НТУУ «Київський політехнічний інститут».

Члени оргкомітету:

Плотніков В. М. – д.т.н., проф., завідувач кафедри ІТтаКБ ОНАХТ,
Артеменко С.В. – д.т.н., проф., завідувач кафедри КІ ОНАХТ,
Князєва Н.О. – д.т.н., проф. кафедри КІ ОНАХТ,
Хобін В.А. – д.т.н., проф., завідувач кафедри АТПтаРС ОНАХТ,
Тарасенко В.П. – д.т.н., проф., завідувач кафедри СКС НТУУ «Київський політехнічний інститут»,
Невлюдов І.Ш. – д.т.н., проф., завідувач кафедри КІТАМ ХНУРЕ,
Мельник А.О. – д.т.н., проф., завідувач кафедри ЕОМ НУ “Львівська політехніка”,
Жуков І. А. – д.т.н., проф., завідувач кафедри КСтаМ НАУ.

Матеріали подано українською, російською та англійською мовами.
Редактор збірника Котлик С.В.

прийнятний результат в більшості ситуацій, істотно зменшуючи час рендерингу. Джерело світла типу *Spot* (прожектор) поширює промені, що виходять з точки, в певному напрямку конічним потоком і висвітлює область всередині конуса. Використовується для імітації світла прожекторів, для отримання ефекту каустики води в певних областях.

Ефект проходження світлових променів через водний шар посилюється за рахунок додавання туману. Для імітації туману в *Maya* включено розділ атмосферних ефектів (*Rendering > Environment ... > Atmosphere*). Радіус дії туману по відношенню до камери можна коригувати. Крім цього можна встановлювати щільність води (*Water Density*), її глибину (*WaterDepth*), спад (*Water Light Decay*) і розсіювання світла (*Water Light Scatter*) в воді.

Візуалізація є завершальним етапом підготовки тривимірного проекту і має не менше значення, ніж проведені перед цим моделювання, освітлення і текстурування сцени, оскільки від результату рендерингу залежать ефектність і реалістичність підсумкового зображення. Не останню роль в цьому відіграє правильне формування освітлення.

ПРОГРАМНИЙ МОНІТОР МІКРОКОНТРОЛЕРІВ STM32

старший викладач О.Г. Шевченко, магістрант А.С. Бардаков

Донецький національний технічний університет (м. Покровськ , Україна)

Будь-який процес керування являє собою у першу чергу обмін повідомленнями між керуючим та виконуючим пристроями. У більшості випадків роль керуючого пристрою може виконувати персональний комп'ютер (ПК), а виконуючого – мікроконтролер (МК) [1]. Дана стаття присвячена опису власної розробки - спеціалізованої термінальної програми-монітору апаратно-програмного комплексу ПК-МК STM32, з використанням USB-підключення.

Функціонально програмний монітор реалізує: пошук серед USB-пристроїв необхідного мікроконтролера; встановлення з'єднання; організацію обміну з пристроєм; аналіз інформації; візуалізацію даних; логування дій.

У відповідності до визначених функцій, архітектура додатку складається з таких компонентів: ініціалізація пристрою – модуль, який виконує функції сканування USB-пристроїв та пошуку серед них мікроконтролера; обмін повідомленнями та їх аналіз – модуль, що виконує функції обміну повідомленнями та аналізу вхідних даних з використанням власного протоколу і ведення журналу подій; обробка подій – модуль, який містить обробники подій: взаємодія користувача з елементами інтерфейсу, під'єднання та від'єднання мікроконтролера, надходження повідомлення від мікроконтролера.

Для того, щоб зрозуміти, яку саме інформацію містить повідомлення, було розроблено спеціальний протокол для перетворення вхідного набору байтів у структурований набір даних. Повідомлення має чітко визначені фрагменти: перші три байти (заголовок пакету) задають код операції та розмір

повідомлення. В залежності від значення коду операції, блок аналізу повідомлень визначає які саме дані містять подальші байти пакету. Інформація про розмір пакету слугує для перевірки цілісності вхідних даних. Якщо розмір повідомлення не збігається з числом, вказаним у заголовку пакету, пакет вважається пошкодженим. Ініціатором обміну між ПК та МК виступає ПК, який, в залежності від дій користувача формує та відправляє до МК відповідні запити.

Після виявлення пристрою за повідомленням від хабу, система USB зчитує його дескриптори для того, щоб визначити, які програмні компоненти необхідно завантажити і сповістити про появу нового пристрою. Будь-який USB-пристрій обов'язково має дескриптори пристрою, конфігурації, інтерфейсів та кінцевих точок. У дескрипторі пристрою є поля, що ідентифікують пристрій в цілому: ідентифікатор виробника (VID – Vendor Id), який призначається USB-IF (організація USB Implementers Forum); ідентифікатор продукту (PID – Product Id) і його версії (DID – Device Id), що визначаються виробником [2].

Для зв'язування нестандартного пристрою зі своїм драйвером використовуються ідентифікатори VID і PID. Так, після визначення пристрою, VID та PID якого співпадають з відповідними ідентифікаторами мікроконтролера, буде отримано ім'я віртуального послідовного порту, що емулюється мікроконтролером.

Після того, як встановлено з'єднання між ПК та МК, відбувається контрольний запит з боку ПК на отримання початкового списку змінних прикладних програм та ресурсів мікроконтролера. У разі успішного отримання відповіді відбувається візуалізація списку змінних та ресурсів у вікні графічного інтерфейсу програми (рис.1).

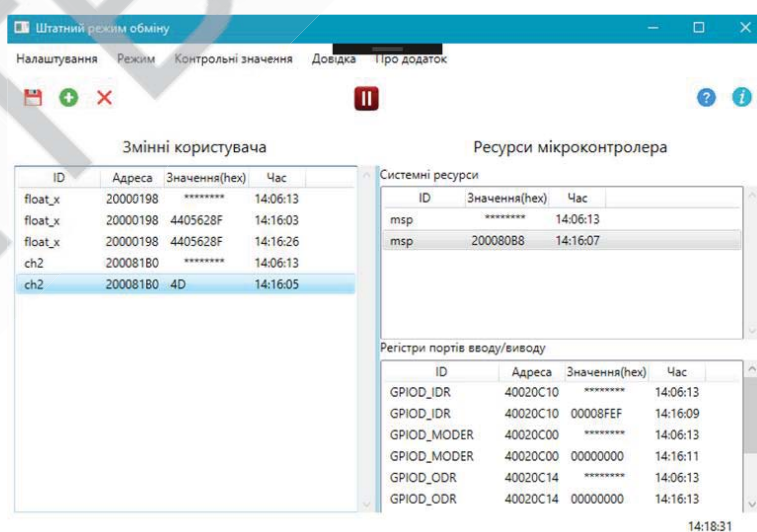


Рис. 1 – Графічний інтерфейс додатку

Для змінних та регістрів портів вводу/виводу надається така інформація: ідентифікатор, адреса у пам'яті МК, значення, час отримання. Для системних ресурсів маємо такий самий набір за виключенням адреси. Слід зазначити, що у

відповідь на контрольний запит, надсилається інформація про змінні/ресурси без поточних значень.

Програмний монітор підтримує два режими роботи - штатний та моніторингу. Під час роботи у штатному режимі відправлення запиту на отримання поточного значення змінної/ресурсу ініціюється користувачем, який може скоротити або збільшити список контролюємих змінних, вказати формат виводу значень змінних, призупинити та продовжувати сеанс обміну. В режимі моніторингу кожне значення оновлюється автоматично з інтервалом, що встановлюється при налаштуванні програми.

Для тестування додатку було використано ПК з операційною системою Windows 10 Pro та мікроконтролер STM32F407[3]. Результати тестування підтвердили працездатність розробленого додатку.

Література

1. Водовозов А.М. Микроконтроллеры для систем автоматики /А.М. Водовозов. – М.: Инфра-инженерия, 2016. - 164с.
2. Интерфейс USB. Часть 4. Дескрипторы и классы. URL: <http://radiohlam.ru/?p=1358>
3. Geoffrey Brown Discovering the STM32 Microcontroller - Indiana University, 2016. – 244p.

«ОГЛЯД ПРОТОКОЛУ DHCPv6»

Білокуров А.О., студент 541 гр., факультет КІШтаКЗ
Одеська Національна Академія Харчових Технологій
Керівник – ст. викл. кафедри КІ Бобрікова І.С.

Мета доповіді – детально розглянути властивості та застосування протоколу *DHCPv6*. Доповідь присвячена огляду протоколу *DHCP* для адресації *IPv6*.

Актуальність теми – необхідність використання більш поширеного адресного простору, тому що майже не лишилось *IPv4* адрес для користування Інтернетом. Тому організація *IETF* розробила протокол *IPv6*, який має великий 128-бітний адресний простір, це приблизно 340 ундецілліонів адрес.

DHCPv6 – нова версія *DHCP* для роботи в мережах на основі протоколу *IPv6* та задання автоматичної конфігурації вузлів *IPv6*. Вузли *IPv6* можуть автоматично генерувати *IP*-адреси, використовуючи авто конфігурацію, або протокол *DHCPv6* може їх призначити, використовуючи дані конфігурації.

Засоби автоматичного призначення *IPv6* адрес:

1. *DHCPv6* – пристрій використовує звичайні процеси пошуків і запитів до *DHCPv6*-серверів для отримання всієї своєї інформації про адресації. Така інформація включає в себе індивідуальну адресу *IPv6*, довжину префікса, адреса шлюзу та адреси *DNS*-серверів. В цьому випадку *DHCPv6*-сервер