

# Програмно-апаратна реалізація відеокамери, сумісної зі стандартом GigE Vision

Марченко<sup>f</sup> В. І., ORCID [0000-0002-0689-8689](https://orcid.org/0000-0002-0689-8689)

e-mail [valera.marchenko96@gmail.com](mailto:valera.marchenko96@gmail.com)

Ходнєв Т. А., [0000-0001-9168-0504](https://orcid.org/0000-0001-9168-0504)

e-mail [t.khodniev@kpi.ua](mailto:t.khodniev@kpi.ua)

Варфоломєєв<sup>s</sup> А. Ю., к.т.н, ORCID [0000-0002-6990-7140](https://orcid.org/0000-0002-6990-7140)

e-mail [a.varfolomeiev@kpi.ua](mailto:a.varfolomeiev@kpi.ua)

Кафедра конструювання електронно-обчислювальної апаратури, Факультет електроніки  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Київ, Україна

*Реферат*—У роботі розглянуто інтерфейси передачі потокового відео, описані їх переваги і недоліки та обґрунтовано доцільність проведення досліджень в сфері використання стандарту GigE Vision. На основі відкритих бібліотек Aravis та Video4Linux створено програмний емулятор GigE Vision сумісної камери. Показано, що з огляду на створювані затримки, безпосереднє використання прикладного програмного інтерфейсу Video4Linux для первинного захоплення відео не є доцільним. Для уникнення вказаних затримок запропоновано власну програмно-апаратну реалізацію підсистеми захоплення відео, яка побудована на базі ARM+FPGA системи-на-кристалі. Дана програмно-апаратна підсистема є спеціалізованим контролером доступу до пам'яті, що розміщується в FPGA частині та враховує особливості передачі відеоданих через мережу, за рахунок чого і дозволяє мінімізувати відповідні затримки.

Бібл. 10, рис. 3.

*Ключові слова* — потокова передача відео; відеокамера; Ethernet; GigE Vision; FPGA; ARM; SoC

## I. ВСТУП

Для реалізації автоматизованих систем спостереження, візуального контролю та керування промисловими лініями і науковим устаткуванням на сьогодні широко застосовують набули системи комп'ютерного зору. У таких системах центр оброблення інформації та об'єкт, за яким треба спостерігати або контролювати, часто розподілені територіально, що потребує організації між ними каналу передавання відео. При цьому зазначений канал має задовольняти особливим вимогам щодо якості, швидкодії та надійності роботи, оскільки у випадку втрати кадрів, їх пошкодженні, а також у разі збільшення затримок передавання відеоінформації, адекватна реакція системи комп'ютерного зору не може гарантуватись. Очевидно, що перелічені вимоги не здатні забезпечити всі існуючі технології передавання відеопотоку, тому проведемо їх короткий аналіз.

На сьогодні можна виділити 3 основні типи каналів передавання відео, що використовуються для промислових задач:

- аналогові рішення;
- цифрові, із застосуванням спеціалізованих інтерфейсів: USB, CoaXPress [1], SDI [2], Camera Link [3];

- цифрові, на основі мережних протоколів: RTSP, GigE Vision.

Аналогові рішення тривалий час були одними з найбільш поширених. Цьому сприяли відносно невисока вартість, мінімальна затримка поширення відеосигналу в каналі передавання, проста й відпрацьована роками технологія використання. Водночас, не зважаючи на наявність на ринку більш сучасних аналогових стандартів, таких як AHD, HD-TVI, HD-CVI, вони все ж не здатні забезпечити якість і гнучкість притаманну цифровому відео: вищу захищеність від завад, ширший динамічний діапазон, змінну роздільну здатність тощо. Слід відзначити, що для подавання аналогового відео на обчислювальний засіб, де це відео має оброблятися, необхідний спеціальний пристрій захоплення.

Особливість використання цифрових інтерфейсів Camera Link, CoaXPress, SDI так само, як і у випадку з аналоговою передачею, полягає у необхідності використання спеціалізованих плат відеозахоплення. При цьому інтерфейси CoaXPress [1] та SDI [2] дозволяють виконувати передавання даних на відстань до 200 м по коаксіальному кабелю, причому зі збільшенням відстані, допустима швидкість відеопотоку в них знижується. Інтерфейс Camera Link розрахований на меншу дальність передавання (до 10 м [3]) і використовує багатопарний кабель з 28 витими парами.



Окремо варто виділити інтерфейс USB. Його головною особливістю є універсальність і доступність майже на всіх сучасних обчислювальних засобах. Швидкість обміну через USB може досягати  $5 \div 20$  Гбіт/с при використанні інтерфейсу третього покоління (USB 3), чого достатньо для передавання відео високої та надвисокої роздільної здатності. При цьому також забезпечується мала затримка отримання даних. Недоліком цієї технології є коротка відстань каналу – при швидкісному з'єднанні ( $\geq 5$  Гбіт/с) довжина стандартного кабелю не має перевищувати 5 м. Існують версії USB, що використовують оптоволоконний кабель, проте і вони дозволяють здійснювати передавання на декілька десятків метрів.

Мережний канал, так само як і USB може використовувати стандартний фізичний інтерфейс з'єднання, зокрема, Gigabit Ethernet (стандарту IEEE 802.3ab), який доступний на більшості сучасних обчислювальних пристроїв. При цьому на відміну від шини USB довжина мережного каналу може сягати 100 м (при використанні стандартного кабелю категорії 5e). З'єднання на основі технології Gigabit Ethernet має меншу швидкість в 1 Гбіт/с (для найбільш розповсюдженої на сьогодні версії 1000 Base-T), якої, тим не менш, достатньо для передавання відеопотоку високої чіткості без стиснення в реальному часі із частотою кадрів до 60 Гц. На основі мережного каналу функціонує ряд протоколів передавання відеопотоку. Серед них найбільш поширеними є технологія RTSP (Real-Time Streaming Protocol), що використовується більшістю сучасних IP-камер та GigE Vision – спеціалізований протокол для високопродуктивних промислових камер машинного зору.

Основним недоліком RTSP-технології є те, що вона оптимізована, а тому і найчастіше використовується для передавання стисненого відео. З одного боку це зменшує навантаження на канал, але з іншого – збільшує накладні витрати на кодування та декодування кадрів, підвищуючи затримку їх отримання кінцевим споживачем (системою комп'ютерного зору), а також знижує надійність передавання, оскільки втрата малої частини стиснутого потоку може призводити до псування цілого або навіть декількох кадрів (залежно від алгоритму стиснення). Варто відмітити, що передавання не стиснутого відео за допомогою технології RTSP створює вище навантаження на канал, ніж аналогічний відеопотік через протокол GigE Vision [4].

Камери стандарту GigE Vision зазвичай не виконують стискання відео, передаючи його «як є» безпосередньо із сенсора. Хоча вони і потребують швидкісного каналу передавання, проте позбавлені додаткових накладних витрат, пов'язаних з кодуванням і декодуванням потоку, а також є більш захищеними від сильного пошкодження кадрів. Зазначені переваги, дальність каналу, універсальність та доступність устаткування Ethernet, робить технологію GigE Vision однією з найбільш придатних для реалізації територіально розподілених систем комп'ютерного зору.

З огляду на сказане вище, в даній публікації вирішено розглядати саме технологію GigE Vision. При

цьому мета роботи полягає у розробці прототипу власної програмно-апаратної відеокамери, що відповідає даному стандарту.

## II. СТВОРЕННЯ ПРОТОТИПУ ВЛАСНОЇ КАМЕРИ

При проектуванні пристрою передавання відео було використано відкриту бібліотеку з відкритим кодом Aravis [5]. Загалом дана бібліотека призначена для універсального захоплення відеопотоків з різноманітних GigE Vision пристроїв. Проте, цікавою її особливістю є наявність спеціального модуля, що програмно емулює роботу реальної камери. Цей модуль використовується здебільшого для внутрішніх цілей налагодження самої бібліотеки і може генерувати лише просте зображення з градієнтами, що з часом змінюються.

В той же час, зазначений модуль емуляції камери нескладно змінити так, щоб він почав передавати не тестове градієнтне зображення, а реальне відео. Це власне і було зроблено в рамках роботи [6].

Оскільки бібліотека Aravis орієнтована на роботу з UNIX-подібними операційними системами [5], подальша робота виконувалась в середовищі Linux. Зокрема, для первинного захоплення відеопотоку з камери використовувалась стандартна бібліотека Vide4Linux [7]. З її допомогою отримувались чорнобілі кадри з вбудованої камери портативного комп'ютера, які зберігались в окремому буфері оперативної пам'яті і надалі передавались бібліотеці Aravis, яка в свою чергу надсилала даний буфер у мережу Gigabit Ethernet, емулюючи роботу справжньої GigE Vision камери. Структура створеної програмної частини показана на Рис. 1 [6].

Для тестування створеного емулятора камери, до комп'ютера, на якому він працював, через мережу Gigabit Ethernet за принципом точка-точка підключався інший комп'ютер. На цьому комп'ютері запускалася програма захоплення GigE Vision відеопотоку, створена так само на основі бібліотеки Aravis (argv-viewer). При цьому даній програмі вдалося успішно отримувати і відображати відеопотік, що генерувався емулятором камери. Приклад її роботи показано на рис. 2.

Не зважаючи на те, що успішну передачу відеопотоку вдалося досягти, під час тестування також було виявлено наявність незначних, але візуально помітних затримок отримання кадрів. Для з'ясування причини їх виникнення нижче оцінено необхідну пропускну здатність каналу та затримку при первинному захопленні відео через підсистему Vide4Linux.

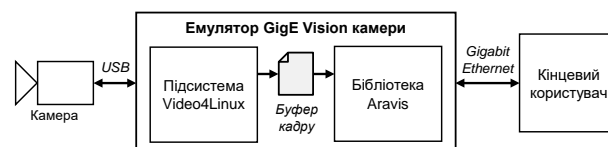


Рис. 1 Структурна схема програмної частини модифікованого емулятора камери

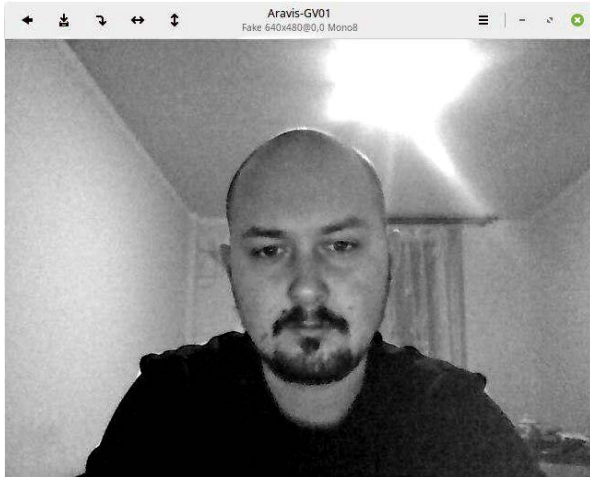


Рис. 2 Приклад результату захоплення відеопотоку, отриманого зі створеного емулятора камери

При тестуванні емулятора камери роздільна здатність кадрів, що передавалися становила  $640 \times 480$  пікселів, а частота їх зміни – 30 Гц. Тоді, пропускна здатність каналу, необхідна для передавання вказаного відеопотоку має складати приблизно:

$$C = m \cdot n \cdot f \cdot b \cdot \frac{MTU}{MTU - k} \cdot 10^{-6} =$$

$$= 480 \cdot 640 \cdot 30 \cdot 8 \cdot \frac{1500}{1500 - 28} \cdot 10^{-6} \approx 75,1 \quad (\text{Мб/с}) \quad (1)$$

де  $m$  – висота кадру;  $n$  – ширина кадру;  $f$  – частота кадрів;  $b$  – глибина кольору в бітах;  $MTU$  – максимальний розмір даних в одному пакеті в байтах;  $k$  – кількість службової інформації в байтах.

Оскільки відеопотік GigE Vision передається в UDP дейтаграмах [8], то значення  $k$  має дорівнювати 28 байт: 8 байт на UDP-заголовок і 20 байт на IP-заголовок [9]. Типове значення  $MTU = 1500$  байт.

Як видно з формули (1), необхідна для передавання потоку пропускна здатність становить 80,1 Мбіт/с, що приблизно на 25% менше номінальної пропускної здатності 100 мегабітного каналу,

який використовувався під час тестування. Це свідчить про відсутність перевантажень каналу мережі. Водночас, при безпосередньому виведенні на екран відео, отриманого бібліотекою Vide4Linux, наявність затримок також виявилась присутньою. Це дозволяє зробити висновок, що основним джерелом затримок у створеному прототипі камери є саме підсистема первинного захоплення відео, що працює на основі бібліотеки Video4Linux. Однією з імовірних причин, чому це відбувається є те, що зазначена бібліотека очікує отримання повного кадру із відеосенсора чи камери, і лише після цього виконує його відображення чи надсилання мережею. З огляду на це, процедуру захоплення, основу на бібліотеці Video4Linux пропонується замінити на програмно-апаратну частину, яка реалізуватиметься на ПЛІС. Цьому питанню присвячено наступний розділ.

### III. ПРОГРАМНО-АПАРATНА РЕАЛІЗАЦІЯ GIGE VISION СУМІСНОЇ КАМЕРИ НА ПЛІС

Отже, з метою підвищення швидкодії камери та зниження затримки при передаванні відео, її частину, що відповідає за захоплення відео (прикладний програмний інтерфейс Video4Linux), вирішено замінити на програмно-апаратну реалізацію на базі ПЛІС (FPGA). Для зручності, при цьому, вирішено використовувати систему-на-кристалі (SoC) ARM+FPGA:

- бібліотека Aravis працюватиме на ARM-процесорі;
- підсистема первинного захоплення відео представлятиме з себе ряд модулів реалізованих на FPGA, які виконуватимуть роль контролера прямого доступу до пам'яті (DMA), що виконуватиме передавання кадрів із відеосенсора до ОЗП ARM-процесора.

Структура системи, що пропонується, показана на рис. 3. В ній, зокрема, відеосенсор сполучається з FPGA частиною, яка формує відеопотік, що зберігається у ОЗП ARM-процесора і далі, завдяки бібліотеки Aravis, порціями по мірі готовності передає кадр через мережний канал.

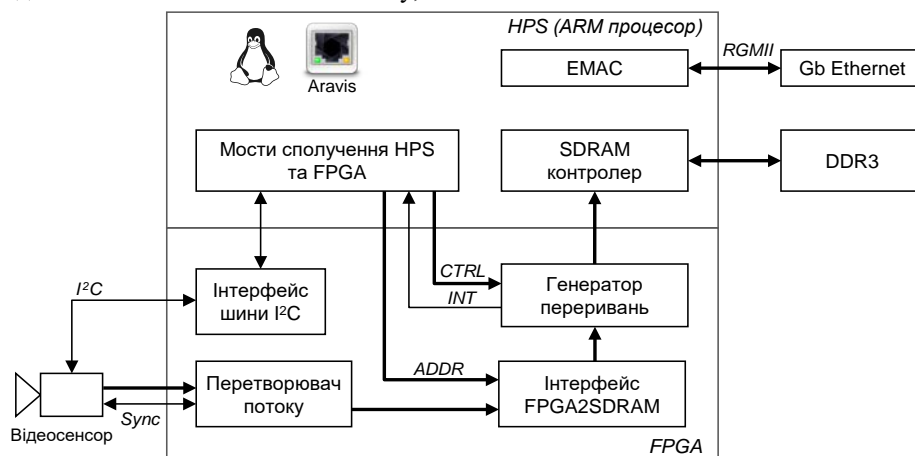


Рис. 3 Структурна схема GigE Vision сумісної камери, реалізованої на FPGA-SoC системі

Більшість оптичних відеосенсорів мають інтерфейс, який включає: 3 сигнали синхронізації (тактову частоту, сигнали формування рядка та кадру), паралельну чи швидкісну послідовну шини даних, а також 2 сигнали для ініціалізації, що зазвичай реалізовані шиною I<sup>2</sup>C. Зазначена конфігурація інтерфейсу визначає структуру блоків апаратної частини у FPGA. Зокрема, інтерфейс шини I<sup>2</sup>C передбачений для безпосереднього керування параметрами сенсора. Перетворювач потоку – це блок, що виконує початкову синхронізацію із сенсором та отримує від нього пікселі зображення по шині даних. Зазвичай у якості цієї шини використовується паралельний інтерфейс розрядністю від 8 до 16 біт або спеціальний послідовний інтерфейс MIPI-CSI [10]. При цьому перетворювач потоку також може формувати із вхідних даних, отримуваних від сенсора, сигнали більшої розрядності для більш ефективного їх передавання через інтерфейс FPGA2SDRAM – блок, що реалізує зв'язок між ПЛІС та ОЗП ARM-процесора. При цьому на інтерфейс FPGA2SDRAM по шині ADDR подається адреса спільного буфера ОЗП, в який має завантажуватись кадр із сенсора. Генератор переривань аналізує кількість переданих до ОЗП даних і генерує запити на переривання, щоб сповістити бібліотеку Aravis про готовність чергової порції відеоданих, які можна надіслати через мережу.

Блок генерування переривань є ключовим в забезпеченні максимальної швидкодії та мінімальних затримок при передаванні відео мережею. В запропонованій програмно-апаратній реалізації, він дозволяє визначати моменти готовності тільки певної частини кадру, передавання якої можна вже розпочинати, не очікуючи отримання всього кадру. Контроль блоком формування переривань виконується через окрему шину CTRL, що визначає розмір порції даних, при готовності яких має формуватись запит на переривання INT. При цьому обсяг порції даних може оптиміально обиратись на основі використаного розміру MTU, що зокрема і дозволяє зменшити затримки передавання.

#### ВИСНОВКИ

В роботі запропоновано власний емулятор GigE Vision сумісної камери, побудованої на основі відкритої бібліотеки Aravis. Показано, що використання в такому емуляторі прикладного програмного інтерфейсу Video4Linux для первинного захоплення відео не є доцільним, з огляду на створювані ним затримки. Натомість, для уникнення зазначених затримок запропоновано програмно-апаратну реалізацію камери на основі ARM+FPGA системи-на-кристалі, в якій відеосенсор сполучається з ОЗП ARM-процесора за допомогою спеціалізованого контролера доступу до пам'яті, що реалізується в FPGA частині. Даний контролер має структуру, яка враховує особливості передавання відеоданих через мережу і за рахунок цього дозволяє мінімізувати затримки.

Напрямок подальших досліджень можуть бути експерименти із заміною запропонованого спеціалізованого контролера доступу до пам'яті, реалізованого на ПЛІС, на відповідні інтерфейсні модулі, що

**Надійшла до редакції 19 вересня 2018 р.**

застосовуються у деяких прикладних процесорах, шляхом зміни функціональності драйверів даних інтерфейсних модулів.

#### ВНЕСКИ АВТОРІВ

Варфоломєєвим А. Ю. запропоновано загальну концепцію системи, створено вихідний код для програмного емулятора камери у другому розділі. Програмно-апаратна реалізація камери у третьому розділі розроблена Варфоломєєвим А. Ю. та Марченко В. І. спільно. Експерименти з передачею відео проведені Марченко В. І. та Ходневим Т. А. спільно. Відсоткові внески авторів в роботу: Варфоломєєв А. Ю. – 55%, Марченко В. І. – 30%, Ходнев Т. А. – 15%.

#### ПЕРЕЛІК ПОСИЛАНЬ

- [1] Japan Industrial Imaging Association Standard, *CoaXPress Standard (ver. 1.1.1)*, Tokyo: JIIA, 2016. URL: [http://jiiia.org/wp-content/themes/jiiia/pdf/standard\\_dl/coaxpress/CXP-001-2015.pdf](http://jiiia.org/wp-content/themes/jiiia/pdf/standard_dl/coaxpress/CXP-001-2015.pdf) [Accessed 19, Aug 2018].
- [2] European Broadcasting Union, "Advice on the use of 3 Gbit/s HD-SDI interfaces," EBU, Geneva, 2011. URL: <https://tech.ebu.ch/docs/techreports/tr002.pdf> [Accessed 19, Aug 2018].
- [3] F. Cuzze, "Appendix D For The Camera Link Specification for The Interface Standard for Digital Cameras and Frame Grabbers," 3M Electronic Solutions Division, Austin, 2004. URL: <http://multimedia.3m.com/mws/media/2974660/3mtm-camera-linktm-app-d-spec-for-dig-camera-frame-grabber.pdf> [Accessed 19, Aug. 2018].
- [4] T. Khodniev, A. Varfolomeiev, O. Lysenko and O. Antonyuk, "Comparison of RTSP and GigE Vision Video Streaming Technologies in Terms of Communication Path Utilization Efficiency: an Experimental Approach," in *International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo)*, Odesa, 2018 [in Publishing].
- [5] E. Pacaud, "Aravis – A vision library for genicam based cameras," 2016. [Online]. Available: <https://aravisproject.github.io/docs/aravis-0.4/> [Accessed 19, Aug 2018].
- [6] V. I. Marchenko and A. Y. Varfolomeiev, "Systema nadiynoyi peredachi potokovogo video cherez Ethernet protokol [The System for Robust Video Streaming Over Ethernet Protocol]," in *XI International Scientific and Technical Conference of Young Scientists "Electronics-2018"*, Kyiv, 2018. pp.153–156. URL: [http://elconf.kpi.ua/wp-content/uploads/2018/06/zbirnik-26\\_06\\_2018.pdf](http://elconf.kpi.ua/wp-content/uploads/2018/06/zbirnik-26_06_2018.pdf) [Accessed 19, Aug. 2018].
- [7] "Linux Media Subsystem Documentation," 2018. [Online]. Available: <https://linuxtv.org/downloads/v4l-dvb-apis/uapi/v4l/capture.c.html> [Accessed 19, Aug 2018].
- [8] Automated Imaging Association, GigE Vision. Camera interface standard for machine vision, Michigan: AIA, 2006.
- [9] V. G. Olifer and N. A. Olifer, *Kompiuternie seti. Printsipi, tehnologii, protokoly: Uchebnik dlya vuzov. 3-e izd.* [Computer networks. principles, technologies, protocols: University textbook. 3rd ed.], St. Petersburg: Piter Publishing House, 2006. ISBN 5-469-00504-6.
- [10] MIPI Alliance, *Specification for Camera Serial Interface 2 (CSI-2)*, New Jersey: MIPI Alliance, Inc., 2009. URL: [http://zink.hexatux.org/svn/filedetails.php?repname=datasheets&path=%2Fmipi%2Fmipi\\_CSI-2\\_specification\\_v01-01-00\\_r0-05.pdf&peg=64](http://zink.hexatux.org/svn/filedetails.php?repname=datasheets&path=%2Fmipi%2Fmipi_CSI-2_specification_v01-01-00_r0-05.pdf&peg=64) [Accessed, 19, Aug. 2018].



УДК 004.773

# Программно-аппаратная реализация видеокамеры, совместимой со стандартом GigE Vision

Марченко<sup>f</sup> В. И., ORCID [0000-0002-0689-8689](https://orcid.org/0000-0002-0689-8689)e-mail [valera.marchenko96@gmail.com](mailto:valera.marchenko96@gmail.com)Ходнев Т. А., [0000-0001-9168-0504](https://orcid.org/0000-0001-9168-0504)e-mail [t.khodniev@kpi.ua](mailto:t.khodniev@kpi.ua)Варфоломеев<sup>s</sup> А. Ю., к.т.н., ORCID [0000-0002-6990-7140](https://orcid.org/0000-0002-6990-7140)e-mail [a.varfolomieiev@kpi.ua](mailto:a.varfolomieiev@kpi.ua)

Кафедра конструирования электронно-вычислительной аппаратуры, Факультет электроники  
Национальный технический университет Украины  
«Киевский политехнический институт имени Игоря Сикорского»  
Киев, Украина

*Реферат*—В работе рассмотрены интерфейсы передачи потокового видео, описаны их преимущества и недостатки, а также обоснована целесообразность проведения исследований в сфере использования стандарта GigE Vision. На основе открытых библиотек Agavis и Video4Linux создан программный эмулятор GigE Vision совместимой камеры. Показано, что в связи с создаваемыми задержками, непосредственное использование прикладного программного интерфейса Video4Linux для первичного захвата видео не целесообразно. Для минимизации указанных задержек предложена программно-аппаратная реализация подсистемы захвата видео, построенная на базе ARM+FPGA системы-на-кристалле. Данная программно-аппаратная подсистема является специализированным контроллером доступа к памяти, размещаемым в FPGA части и учитывающим особенности передачи видеоданных по сети, за счет чего и позволяет минимизировать соответствующие задержки.

Библ. 10, рис. 3.

*Ключевые слова* — потоковая передача видео; видеокамера; Ethernet; GigE Vision; FPGA; ARM; SoC

UDC 004.773

# Software and Hardware Implementation of Video Camera, Compatible with GigE Vision Standard

V. I. Marchenko<sup>f</sup>, ORCID [0000-0002-0689-8689](https://orcid.org/0000-0002-0689-8689)e-mail [valera.marchenko96@gmail.com](mailto:valera.marchenko96@gmail.com)T. A. Khodniev, [0000-0001-9168-0504](https://orcid.org/0000-0001-9168-0504)e-mail [t.khodniev@kpi.ua](mailto:t.khodniev@kpi.ua)A. Y. Varfolomieiev<sup>s</sup>, PhD, ORCID [0000-0002-6990-7140](https://orcid.org/0000-0002-6990-7140)e-mail [a.varfolomieiev@kpi.ua](mailto:a.varfolomieiev@kpi.ua)

Department of Design of Electronic Computing Equipment, Faculty of Electronics  
National technical university of Ukraine  
“Igor Sikorsky Kyiv polytechnic institute”  
Kyiv, Ukraine



**Abstract**—Different variants of video streaming technologies are considered. From the standpoint of locally distributed computer vision systems implementation, the GigE Vision technology is selected for further investigation, since it is capable of providing relatively long channel distances, high robustness and image quality, low latency of video data transmission, and uses standard connection interfaces available in most of computers. Thereby the implementation of own GigE Vision compatible camera is chosen as the particular objective of the paper. Two ways of achieving of this objective are considered. The first one is based on software implementation solely, and uses Aravis open source library, which is coupled with the Video4Linux API. It is shown that this software solution is capable of video streaming, but has high transmission delays, making it inexpedient for use in real-time computer vision systems. On the basis of experiments and the analysis of channel load, it was found that in the test system, the delays were caused mainly by the Video4Linux library. To overcome this problem, the second variant of camera, which uses the hardware implementation of primary video capturing part, was proposed. It is based on system-on-chip (SoC) solution, which incorporates FPGA and ARM-processor. In this system, the video sensor is connected to the FPGA. In the FPGA, the control of video sensor and the special direct memory access controller are implemented. As in the first camera variant, the Aravis library is responsible for the data transmission over Ethernet (it “resides” in the ARM-processor part of SoC). The direct memory access controller, which is implemented in the FPGA part, fills a dedicated buffer in ARM-processor’s RAM, taking into account the peculiarities of data transmission over the network. Particularly, data are sent using UDP protocol by portions, thus the controller includes the respective module, which generates interrupt requests immediately, when some fraction of pixels are loaded from the video sensor to the buffer. This fraction can be optimally estimated from the properties of network connection and is controlled by the Aravis library. Such a structure of direct memory access controller does not require waiting of a full frame transferring from the video sensor (which appears in Video4Linux API) and consequently allow to perform pipelined processing, which leads to minimization of data transmission delays.

Ref. 10, fig. 3.

**Keywords** — *video streaming; video camera; Ethernet; GigE Vision; FPGA; ARM; SoC*

