МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

"ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

Кафедра "Обчислювальна техніка та програмування"

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

**до курсового проекту**

на тему Розробка програмного забезпечення

для отримання зображення з ip-камери з використанням

протоколу передачі TCP

Виконав студент 3 курсу, групи КІТ-26в

Гатанюк Н.С.

Керівник Шевердін І.В

Харків 2018

РЕФЕРАТ

Пояснювальна записка до курсового проекту: 52с., 8 рис., 13 джерел інформації.

Ключові слова: GIT, RFC, С++, HTTP, TCP, UDP, Qt Creator, WIRESHARK.

У роботі описуються особливості розробки програмного забезпечення з використанням мови. Доцільність даної розробки полягає в тому, щоб забезпечити зручні та швидкі розрахування та отримання результату після великих розрахунків.

**З М І С Т**

[Перелік позначень і скорочень 4](#_Toc406602864)

[Вступ 5](#_Toc406602865)

[1   Огляд використовуваних методів і технологій 6](#_Toc406602866)

[1.1   Системи керування версіями 6](#_Toc406602867)

[1.1.1   Централізовані системи керування версіями 6](#_Toc406602868)

[1.1.2   Розподілені системи контролю версій 7](#_Toc406602869)

[1.1.3   Основи Git 7](#_Toc406602870)

[2   Розробка програмного комплексу 8](#_Toc406602871)

[2.1   Засоби, що використовувались 9](#_Toc406602872)

[2.1.1   Мова програмування С++ 9](#_Toc406602873)

[2.1.2   Робота з GitHub 10](#_Toc406602874)

[2.1.3   Wireshark 11](#_Toc406602875)

[2.2   Технології, що використовувались 11](#_Toc406602876)

[2.1.1   QPushButton](file:///C:\Users\Nata\Downloads\Telegram%20Desktop\курсач%20ИПЗ%20конец%20(2).docx#_Toc406602873) 12

[2.1.2   QLineEdit](file:///C:\Users\Nata\Downloads\Telegram%20Desktop\курсач%20ИПЗ%20конец%20(2).docx#_Toc406602874) 12

[2.1.3   QTextEdit](file:///C:\Users\Nata\Downloads\Telegram%20Desktop\курсач%20ИПЗ%20конец%20(2).docx#_Toc406602875) 12

2.1.4 QAbstractSocket 13

2.1.5 QTcpSocket 13

2.1.6 QLabel 13

2.1.7 QBoxLayout 14

2.1.8 QCursor 14

[3   Опис програми 15](#_Toc406602878)

[Висновки 1](#_Toc406602879)8

[Перелік джерел інформації 20](#_Toc406602880)

Додаток 1 Діалог програми з камерою …...………….………………………….. 21

Додаток 2 Блок - схема алгоритму ...………………………...……………………22

Додаток 3 Початкова форма програми для комп’ютера ………………………...23

Додаток 4 Результат роботи програми для компютера з ОС Windows ...………24

Додаток 5 Результат роботи для компютера з Mac ОС ………………………... 25

Додаток 6 Початкова форма та вікно програми для телефона ………………….26

Додаток 7 Результат роботи програми для телефона ……...…………...……… 27

# Перелік позначень і скорочень

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

HTML – HyperText Markup Language.

HTTP – HyperText Transfer Protocol.Вступ

Мета роботи – продемонструвати ефективність роботи та можливості мови програмування з отриманням даних, аналізу цих даних, у вигляді роботи з будь-якою камерою та відтворення даних з використанням мови С++ у власній мобільній та комп’ютерній програмі.

Була розроблена програма для «спілкування» з камерою, отримання з неї даних, їх аналіз та запис у необхідному форматі в написаному додатку.

У пояснювальній записці розглянемо різні види протоколів, визначимо плюси та мінуси цих протоколів. Розглянемо різні варіанти програми. Проблеми та варіанти їх вирішення.

Для реалізації нашої програми було використано текстовий редактор. Він допомагає робити не складний інтерфейс, та розширює можливості мови С++.

Дана програма є тільки першим етапом розробки, тому графічний інтерфейс досить малий як для повноцінного програмного забезпечення. Цей етап вже може приймати чітке зображення та виводити його на екран. Наступним кроком повинно стати зменшення розміру зображення, розширення для більшої кількості камер та зменшення або зовсім усунення затримки підчас виводу кадру.

Кожен день програмісти в усьому світі розробляють все більше нових функцій, які в майбутньому зможуть покращити цю програму.

У пояснювальній записці наводяться технічні деталі процесу розробки програмного комплексу – від налаштування середовища до покрокової реалізації функціоналу.

Найголовніше це те, для чого цю програму можна використовувати. Оскільки це тільки проміжний етап, тому вона може допомагати тимчасово дивитись на те, що відбувається за невеликий час, бо необхідно перезапускати програму з певним часом. Розширивши цей функціонал, програму можна використовувати для з’єднання декількох камер водночас і спостерігати. Дана програма найкраще підходить для використання в охоронній сфері діяльності.

# Огляд використовуваних методів і технологій

## Системи керування версіями

Система керування версіями (СКВ) – це система, яка зберігає зміни у файлі або декількох файлах, щоб потім можна було бачити потрібні старі версії. У роботі використовуються початкові коди програм, але насправді можна керувати версіями практично будь-яких типів файлів.

Системи керування версіями зазвичай використовуються при розробці програмного забезпечення для відстеження, документування та контролю над поступовими змінами в електронних документах: у сирцевому коді застосунків, кресленнях, електронних моделях та інших документах, над змінами яких одночасно працюють декілька людей

### Централізовані системи керування версіями

Централізована система контролю версії (клієнт-серверна) – система, дані в якій зберігаються в єдиному «серверному» сховищі. Весь обмін файлами відбувається з використанням центрального сервера. Є можливість створення та роботи з локальними репозиторіями (робочими копіями).

Переваги:

* загальна нумерація версій;
* дані знаходяться на одному сервері;
* можлива реалізація функції блокування файлів;
* можливість керування доступом до файлів;

Недоліки:

* наявність мережевого з'єднанні з основним сервером для оновлення робочої копії та збереження змін.

### Розподілені системи контролю версій

Розподілена система контролю версії (англ. Distributed Version Control System, DVCS) – система, яка використовує замість моделі клієнт-сервер, розподілену модель зберігання файлів. Така система не потребує сервера, адже всі файли знаходяться на кожному з комп'ютерів.

Переваги:

* Кожний з розробників працює зі своїм власним репозитарієм;
* Рішення щодо злиття гілок приймається керівником проекту;
* Немає потреби в мережевому з'єднанні;

Недоліки:

* не можливо контролювати доступ до файлів;
* відсутня загальна нумерація версій файлів;
* потрібно значно більше дискового простору на сервері для зберігання всіх гілок як глобальних, так й локальних;
* відсутня можливість блокування файлів.

До розподілених систем відносять: Git, Mercurial, Bazaar, Monotone, Codeville, BitKeeper.

### Основи Git

У Git зовсім не такі поняття про інформацію і роботу з нею як в інших системах, хоча користувальницький інтерфейс дуже схожий.

Git - це набір консольних утиліт, які відстежують і фіксують зміни в файлах (найчастіше мова йде про вихідний код програм, але ви можете використовувати його для будь-яких файлів на ваш смак). З його допомогою ви можете відкотитися на більш стару версію вашого проекту, порівнювати, аналізувати, зливати зміни і багато іншого. Цей процес називається контролем версій. Існують різні системи для контролю версій.

# Розробка програмного комплексу

Завдання програми було отримати зображення з камери. За допомогою деяких можливостей вибраної мови, було розроблено простий алгоритм.

У мене виникло питання «Що краще обирати, udp чи tcp». По-перше, після невеликих тестів було зрозуміло ,що під час звичайного стріму по tcp, з часом з‘являється затримка, яка з кожним кадром все більша і більша. Плюси обраного протоколу такі, що у будь-якій ситуації отримає пакет з камери, якщо буде втрата пакету, тоді програма знову відправить запит і отримає необхідний кадр. Кадри зберігаються в потрібнй послідовності.

Якщо писати програму за допомогою udp, тоді є декілька проблем ,які ми зможем бачити. По-перше, як вже було сказано, udp не дає гарантії, що усі фрейми дійдуть чи дійдуть в іншій послідовності, що приводить до розмивання зображення. Втрати пікселей чи зовсім втрати зображення. Цей протокол допомагає в тій ситуації, коли якість зображення не важлива, коли не страшно, якщо якийсь кадр пропаде.

Проте, tcp дозволяє виводити зображення без втрат картинок, без «зламаних» пікселей. Також виводить зображення в точності як потрібно, в гарній якості, яка була задана запитом програми, та видана камерою після обробки програмного запиту.

Різні мережеві програми основані на гарній допоміжній програмі wireshark. Завдяки цій програмі був розроблений програмний алгоритм. Для виводу зображення необхідно йти по всьому масиву байт, які приходить з камери. Після запиту починається відправка зображення. На рисунку 1 зображено хід роботи «діалогу» програми з камерою.

На рисунку 2 зображено алгоритм функції зчитування фреймів з мережі, аналіз та запис у віджет. Цієї функції достатньо, щоб зрозуміти чому з’являється затримка. Затримка з’являється, тому що функція аналізує кожен кадр і виводить його на екран, і якщо робити так кожен кадр, то через деякий час кількість не прочитаних фреймів накопиться швидше ніж програма буде виводити зображення.

Ця програма є проміжним результатом, адже є інші протоколи для роботи з камерами, які мають більше плюсів і використовують різні методи ніж ті ,що використовуються при реалізації даного проекту. Qt Creator дозволяє писати графічний інтерфейс і додавати туди віджети, що працюють з зображенням. Ось тому тср – це не такий і поганий вибір. Бо, наприклад, використовуючи rtp/rtsp протокол, тоді виходить консольна програма. А ця програма використовує звичайний графічний інтерфейс, щоб зробити легкий спосіб взаємодії користувача з програмою.

Ця програма має мінуси. Одна програма на один комп’ютер, і це різниця від звичних консольних програм. Коли консольну можна запустити декілька разів у фоні. Але ця програма зразу після запуску стріма починає виводити зображення, що робить її більш практичною ніж консольна.

Для того, щоб розробити програму:

1. Було використано wireshark, щоб аналізувати мережу пакетів.
2. Був розроблений клієнт з протоколом TCP, для «розмови» з камерою та для отриманні фреймів із зображенням.
3. Після отримання фреймів, вони були проаналізовані за допомогою написаної функції.
   1. Звичайними функціями С/С++ було витягнуто з сокету фрейми у байтовому форматі.
   2. Завдяки міткам у фреймах було відкинуто інформацію про кадри.
   3. І масив байт ,який був отриманий, був завантажений у спеціальний контейнер, який вивів на екран зображення.

## Засоби, що використовувались

### Мова програмування С++

С++ – мова програмування високого рівня з підтримкою кількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Розроблена Б'ярном Страуструпом в AT&T Bell Laboratories 1979 року та початково отримала назву «Сі з класами». Згодом Страуструп перейменував мову на C++ у 1983 р. Базується на мові С. Вперше описана стандартом ISO/IEC 14882:1998, найбільш актуальним же є стандарт ISO/IEC 14882:2014. У 1990-х роках С++ стала однією з найуживаніших мов програмування загального призначення. Мову використовують для системного програмування, розробки програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм, наприклад, відеоігор. С++ суттєво вплинула на інші популярні сьогодні мови програмування: С# та Java.

При створенні С++ прагнули зберегти сумісність з мовою С. Більшість програм на С вправно працюватимуть і з компілятором С++. С++ має синтаксис, заснований на синтаксисі С.

Нововведеннями С++ порівняно з С є:

* підтримка об'єктно-орієнтованого програмування через класи;
* підтримка узагальненого програмування через шаблони;
* доповнення до стандартної бібліотеки;
* додаткові типи даних;
* обробка винятків;
* простори імен;
* вбудовані функції;
* перевантаження операторів;
* перевантаження імен функцій;
* посилання і оператори управління вільно розподіленою пам'яттю.

### Робота з GitHub

Створивши репозіторій я зробила невеликий крок до своєї мети. Це дозволяє контролювати його протягом всього часу та завжди тримати під рукою. Також за допомогою цього, мій проект може подивитись будь хто, отримати пояснення, документацію та код самої програми.

**GitHub** – один з найбільших [веб-сервісів](https://uk.wikipedia.org/wiki/Веб-служба) для спільної розробки [програмного забезпечення](https://uk.wikipedia.org/wiki/Програмне_забезпечення). Існують безкоштовні та платні тарифні плани користування сайтом. Базується на системі керування версіями Git і розроблений на Ruby on Rails і Erland компанією GitHub, Inc (раніше Logical Awesome). Назва репозиторію – зазначене при створенні репозитория ім'я користувача (username) в англійській транслітерації.

При реєстрації на GitHub, був створений великий репозиторій з ім’ям: naaata16. В ньому був створений репозиторій, в якому зберігається проект та README файл з поясненням.

### Wireshark

Wireshark – програма для аналізу мережевих пакетів Enternet і інших мереж з вільним вихідним кодом. Має графічний інтерфейс користувача і значно більше можливостей із сортування і фільтрації інформації. Програма дозволяє користувачеві переглядати весь трафік, що проходить по мережі, в режимі реального часу, переводячи мережну карту в promiscuous mode. Wireshark розпізнає структуру найрізноманітніших мережевих протоколів, і тому дозволяє розібрати мережевий пакет, відображаючи значення кожного поля протоколу будь-якого рівня. Оскільки для захоплення пакетів використовується pcap, існує можливість захоплення даних тільки з тих мереж, які підтримуються цією бібліотекою. Проте, Wireshark вміє працювати з безліччю форматів початкових даних, відповідно, можна відкривати файли даних, захоплених іншими програмами, що розширює можливості захоплення.

Програма розповсюджується під вільною ліцензією GNU GPL і використовує для формування графічного інтерфейсу кросплатформову бібліотеку GTK+.

**2.2 Технології, що використовувались**

У ході роботи було використано велика кількість різних засобів. Для редагування коду використано програму Qt Creator. Так як раніше працювала з цією програмою, вирішила розробити новий проект в цьому редакторі. Цей редактор має багато плюсів перед іншими. Він є безкоштовним та спеціалізується на розробці програм на С++, тому в ньому з самого початку стоять необхідні бібліотеки та утиліти, ніж в інших редакторах.

Мій проект зберігається в репозиторії. Це дозволяє контролювати його протягом всього часу та завжди тримати під рукою. Також за допомогою цього, мій проект може подивитись будь хто, отримати пояснення, документацію та код самої програми.

Ще одна програма допомагала бачити, як проходила «співбесіда» моєї програми з камерою, тому що wireshark спеціалізується на отриманні та виведення на екран пакетів даних, що передаються по мережі.

**2.2.1 QPushButton**

Використовуючи QPushButton розробники можуть створювати і управляти кнопками. Цей клас є простим у використанні і налаштуванні, тому це один з найкорисніших класів в Qt. В основному кнопка відображає текст, але також може відображатися і іконка.

**2.2.2. QLineEdit**

QLineEdit - віджет для введення одно строкового тексту. Застосовуйте його для введення коротких фрагментів тексту, таких як ім'я користувача. У віджета можна обмежити довжину тексту, що вводиться за допомогою маски введення, що пропонує заповнити шаблон, або для додаткового контролю можна застосувати функцію перевірки допустимості, наприклад, щоб переконатися в тому, що користувач вводить коректні дату, номер телефону або подібні величини. У віджета є функції редагування, що дозволяють вибирати частини тексту, вирізати і вставляти, скасовувати і повторювати зміни, як командами користувача, так і засобами API

**2.2.3. QTextEdit**

Клас QTextEdit забезпечує віджет, який використовується для редагування та відображення як звичайного, так і складного тексту. Може відображати зображення, списки та таблиці. Якщо текст є замалим для перегляду у вікні перегляду тексту редагування, з'являться смужки прокрутки. Редагування тексту може завантажувати як звичайний текст

**2.2.4.QAbstractSocket**

Це базовий клас для QTcpSocket та QUdpSocket і містить усі загальні функціональні можливості цих двох класів. QAbstractSocket API «уніфікує більшість відмінностей між двома протоколами QTcpSocket або QUdpSocket. СonnectToHost () встановлює віртуальне з'єднання для розеток, що дозволяє використовувати QAbstractSocket більш-менш однаково, незалежно від основного протоколу. Внутрішньо QAbstractSocket запам'ятовує адресу та порт, переданий connectToHost (), і функції, такі як read () і write (), використовують ці значення.

**2.2.5. QTcpSocket**

Для реалізації клієнта потрібно створити об'єкт класу QTcpSocket, а потім викликати метод connectToHost (), передавши в нього першим параметром ім'я комп'ютера (або його IP-адреса), а другим - номер порту сервера. Об'єкт класу QTcpSocket сам спробує зробити установку зв'язку з сервером і, в разі успіху, вишле сигнал connected (). В іншому випадку буде висланий сигнал error (int) з кодом помилки, визначеним у перерахуванні QAbstractSocket :: SocketError. Це може статися, наприклад, в тому випадку, якщо на зазначеному комп'ютері не встановлено сервер або не відповідає номер порту. Після встановлення з'єднання об'єкт класу QTcpSocket може висилати або зчитувати дані сервера.

Взаємодія клієнта з сервером зображено на рисунку. Висилка на сервер інформації, введеної в однорядковому текстовому полі вікна клієнта, проводиться після натискання на кнопку Send (Вислати).

**2.2.6. QLabel**

Віджет забезпечує відображення тексту або зображення.

QLabel використовується для відображення тексту або зображення. Немає функцій взаємодії з користувачем. Візуальний вигляд етикетки може бути налаштований різними способами, і він може бути використаний для вказівки ключового мономонічного ключа для іншого віджета.

**2.2.7. QBoxLayout**

Клас QBoxLayout вирівнює дочірні віджети горизонтально або вертикально. QBoxLayout займає відведений йому простір (батьківським компоновщиком або parentWidget ()), розділяє його на ряд осередків і змушує кожен керований віджет заповнити одну клітинку.

**2.2.8. QCursor**

Клас QCursor забезпечує курсор миші з довільною формою. Цей клас використовується головним чином для створення курсорів миші, пов'язаних з окремими віджетами, а також для отримання та встановлення позиції курсора миші.

Qt має ряд стандартних форматів курсору, але ви також можете створювати власні формати курсора на основі QBitmap , маски та точки доступу.

Щоб пов'язати курсор з віджетом, використовуйте QWidget :: setCursor (). Щоб пов'язати курсор з усіма віджетами (зазвичай на короткий проміжок часу), використовуйте QGuiApplication :: setOverrideCursor ().Якщо ви хочете створити курсор із власним растровим зображенням, використовуйте конструктор QCursor, який приймає растрове зображення та маску, або конструктор, який використовує піксель як аргументи.

# 

# Опис програми

У коді використано сокетні з'єднання, а саме потокові сокети. Клас QTcpSocket містить набір методів для роботи в TCP - мережна протоколом низького рівня. Було створено об'єкт сервера, для його запуску створений клас MyClient. У конструктор ми передали номер порту. При підключенні камери, ми дізнаємося її IP-адреса, який в подальшому вводимо при вході в програму і отримуємо зображення.

У нашому класі використано різні атрибути, якими управляємо нашим проектом, зберігаємо довжину наступного блоку отриманого від сокета. Використано текстове поле для інформування про події з'єднаннях.

Клас QImage використовували для завантаження растрового зображення на фон нашого проекту. Так само за допомогою цього класу відбувається відео зображення. Сервер отримує велику кількість фотографій, в результаті правильності коду зображення не губляться і відео бачимо в правильній послідовності.

Клас QCursor використовували для створення власного курсора миші. Покажчик являє собою растрову картинку. Використовували віджети, які є будівельними матеріалами для створення графічного інтерфейсу. За допомогою нього створили вікно для введення даних і відстеження процесу.

Якщо буде виявлено помилку, то буде поданий сигнал ERROR з кодом помилки. Слот slotError () викликається при виникненні помилок. У ньому ми перетворимо код помилки в текст, а потім відображаємо його в віджеті нашого текстового поля. Пересилання на сервер інформації, введеної в однорядковому текстовому полі вікна клієнта, здійснюється при натисканні на кнопку GO.

За допомогою connectToHost встановлюємо зв'язок між сервером і клієнтом. Першим параметром у цей метод передається ім'я комп'ютера, а другим - номер порту. Сокет відправляє сигнал connected (), як тільки буде створено з'єднання, а також сигнал readyRead () - при готовності надати дані для читання. В результаті ми поєднуємо ці сигнали з певними слотами.

На рисунку 1 зображено приклад діалогу програми. В ньому червоним кольором зображено запит на камеру, а синім відповідь, яку програма зчитує у фреймах, які приходять на камеру.

GET /axis-cgi/mjpg/video.cgi?resolution=320x240 HTTP/1.1

Host: 10.168.0.186

Як це розщифрувати:

GET — це метод запиту.

/axis-cgi/mjpg/video.cgi?resolution=320x240 — це форма запиту камери AXIS.

Вона запрошує motionjpg, розміром одного кадру 320х240.

Host 10.168.0.186 — це адреса самої камери в мережі.

HTTP/1.0 200 OK

Cache-Control: no-cache

Pragma: no-cache

Expires: Thu, 01 Dec 1994 16:00:00 GMT

Connection: close

Content-Type: multipart/x-mixed-replace; boundary=myboundary

--myboundary

Content-Type: image/jpeg

Content-Length: 10572

Content-Type: multipart/x-mixed-replace; boundary=myboundary

Розповідає який стрім чекати і яка буде мітка. boundary=myboundary Це мітка між двома різними кадрами.

--myboundary Перша мітка, показує що кадр почався.

Content-Type: image/jpeg — Тип кадру і його формат.

Content-Length: 10572 — Розмір одного кадру в байтах.

10572 байт = 10КБайт одне фото.

1Мбайт = 102 кадра.

Якщо казати звичайною мовою, то коли запускається стрім, то він не скінчено буде приходити на необхідний порт. Тобто ,якщо відправляється запит, ми будем отримувати у відповідь нескінчений потік даних, але з кожним кадром будемо затримувати ці дані, виводячи кожний кадр.

# Висновки

Після не довгої, але все ж таки тяжкої праці була розроблена програма для спілкування з камерою, отримання з неї даних, їх аналіз та запис у необхідний віджет.

У пояснювальній записці наводяться технічні деталі процесу розробки програми – від налаштування середовища до покрокової реалізації функціоналу.

В ході виконання роботи була розгорнута та налаштована система, що демонструє ефективність роботи та можливості мови програмування з отриманням бітових даних та аналізу у вигляді роботи з Axis Video Camera з використанням Qt Creator для управління проектом. Результатом роботи став реальний стрім зображення з камери. Програма реалізована на мові С++. Обмін повідомленнями йде за звичайною схемою «запит-відповідь» на основі HTTP і TCP протоколів. Програма створена для роботи на ОС Windows та є версія програми під Android. Було створено емблему даної програми для Android та фон програми для ОС Windows. Було використано потокові сокети. За допомогою редактора було застосовано різні класи та методи , що допомогли завантажити растрове зображення на фон проекту, створити власний курсор. На початку запуску програми зображення виводяться без затримки, та через деякий час зображення починає відставати від реального часу. З кожною секундою все більше накопичується зображень і в результаті ми отримуємо затримку між зображенням на нашому комп’ютері чи телефоні та реальністю. В результаті сервер отримує велику кількість фотографій, та завдяки правильності коду ,зображення не губляться і відео бачимо в правильній послідовності. Щоб програма працювала ,наш комп’ютер чи телефон повинен бути підключений до одного і того ж самого інтернету ,що і камера. Для початку потрібно отримати ip-адресу нашої камери за допомогою спеціальної програми, а далі просто вести цю адресу в формі та натиснути на копку «GO». Далі результат зображення нашої камери ми бачимо на екрані. Хоча програма є тільки першим етапом, та зображення маємо досить чітке. Мінус цієї програми на даному етапі –це затримка зображень та досить простий інтерфейс.

Для більш подальшої розробки хотілось би, щоб цей стрім можна було би програвати на якійсь HTML сторінці, що дуже полегшить роботу для користувача. Чи краще зробити програму, яка вже містила різні підключені камери в одній програмі з можливістю програвати їх зображення.

З кожним днем у світі з`являється різні нові контейнери, нові засоби кодування, нові мови програмування і технічний прогрес не стоїть на місці. Вдосконалюються бібліотеки, люди розвиваються, машини оновлюються і це приводить до того, що для вирішення різних задач необхідно буде набирати лише одну команду.

# Перелік джерел інформації

1.Postel, J., "Internet Protocol", STD 5, [RFC 791](https://tools.ietf.org/html/rfc791), September 1981.

2.<https://uk.wikipedia.org/wiki/Qt>

3.[https://uk.wikipedia.org/wiki/Git](https://uk.wikipedia.org/wiki/GitLab)Hub

4.<http://doc.qt.io/archives/qt-4.8/qtextedit.html#details>

5.<http://qt-doc.ru/realizacia-klienta-s-pomoschu-klassa-qtcpsocket.html>

6.https://uk.wikipedia.org/wiki/HTTP

7. <http://doc.qt.io/archives/qt-4.8/qtextcursor.html>

8. <https://wiki.qt.io/How_to_Use_QPushButton/ru>

9. <http://doc.qt.io/qt-5/qcursor.html#details>

10. <http://doc.qt.io/qt-5/qhboxlayout.html>

11. <http://doc.qt.io/qt-5/qabstractsocket.html>

12. <http://doc.qt.io/qt-5/qlabel.html>

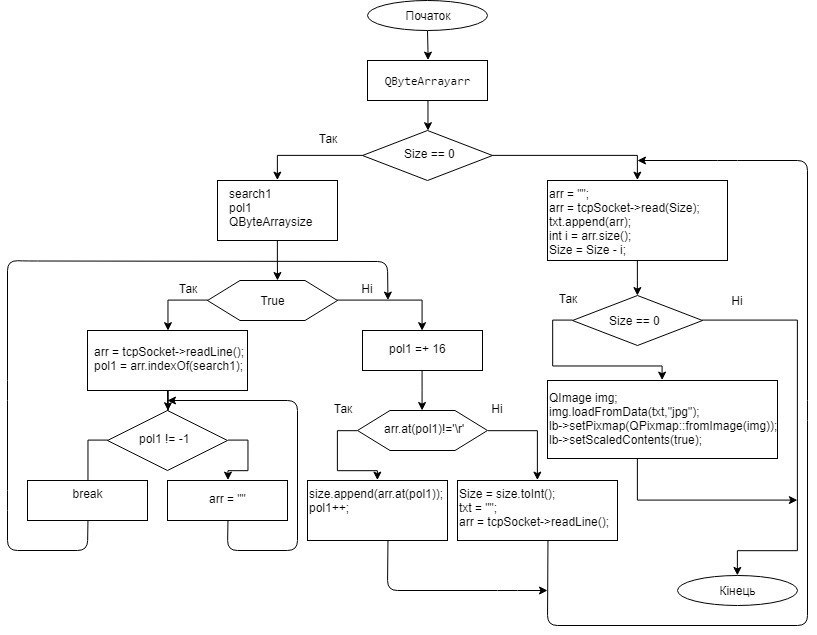
13. <https://proglib.io/p/git-for-half-an-hour/>

Додаток 1 – Діалог програми з камерою



Рисунок 1 - Приклад діалогу програми з камерою

Додаток 2 - Блок-схема алгоритму

Рисунок 2 - Алгоритм функції зчитування фреймів з мережі, аналіз та запис у віджет

Додаток 3-Початкова форма програми для комп’ютера

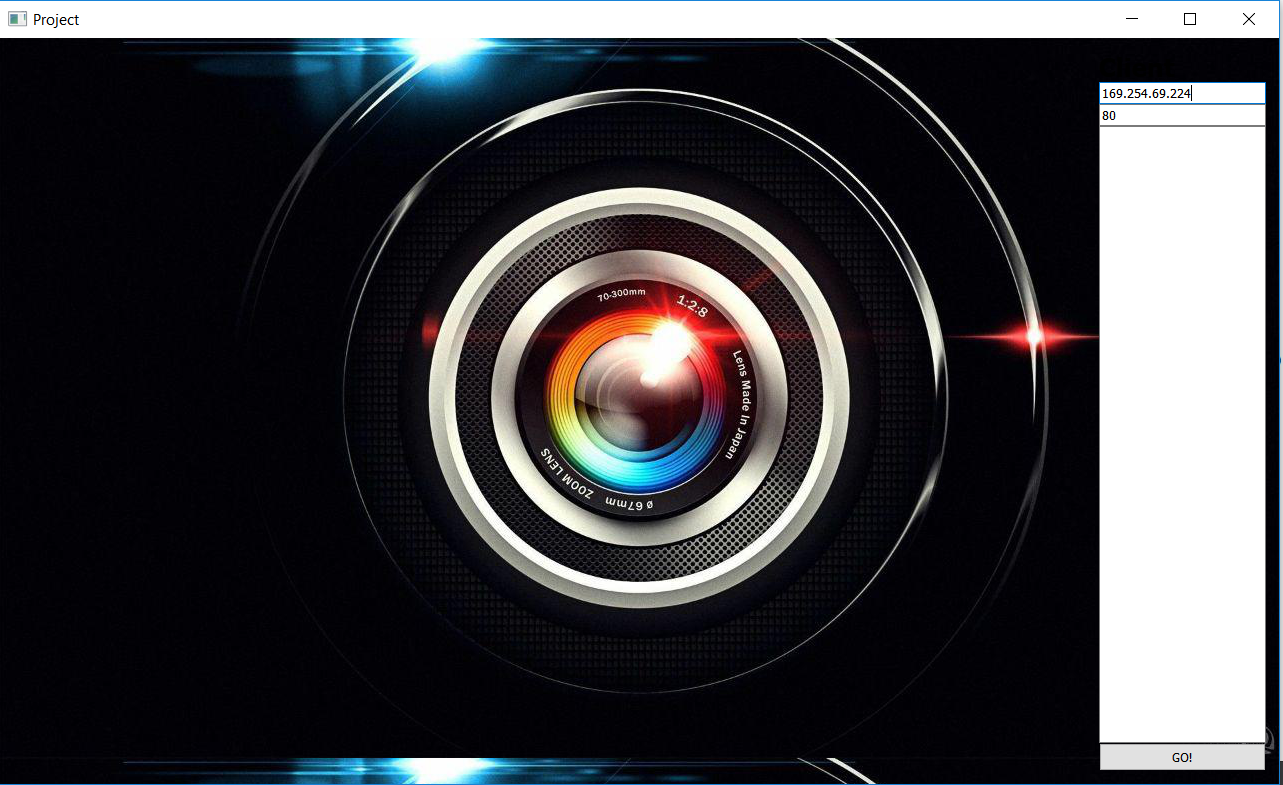


Рисунок 3 – Початковий екран програми перед запуском

Додаток 4 Результат роботи програми для компютера з ОС Windows



Рисунок 4 – Результат роботи програми

Додаток 5 Результат роботи для компютера з Mac ОС



Рисунок 5 –Результат роботи програми

Додаток 6- Початкова форма та вікно програми для телефона

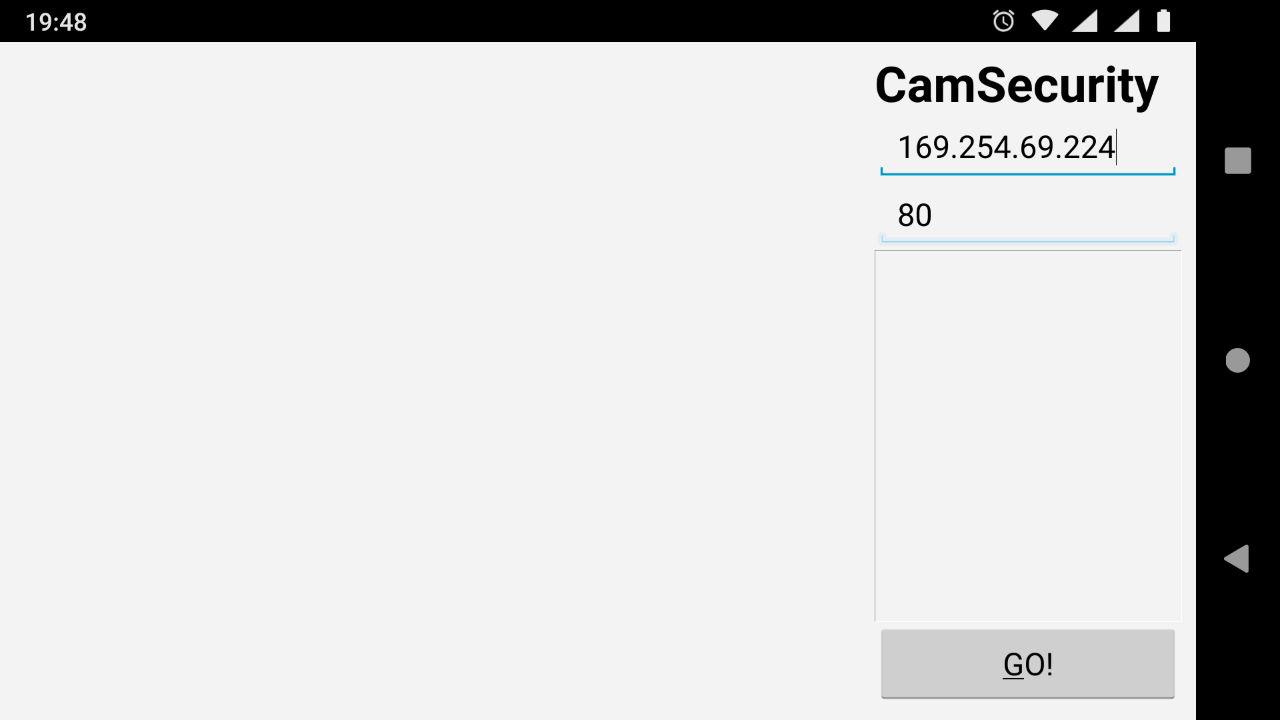


Рисунок 6 - Початковий екран програми перед запуском

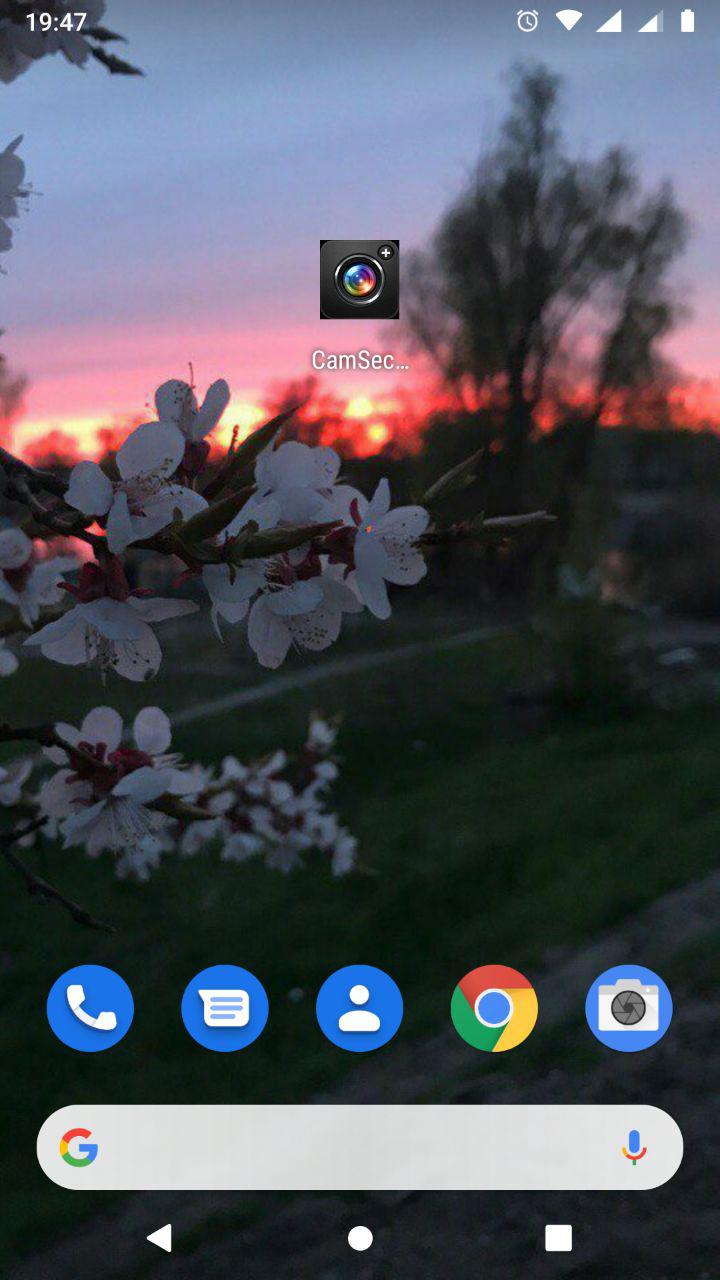


Рисунок 7 – вікно програми

Додаток 7 –Результат роботи програми для телефона

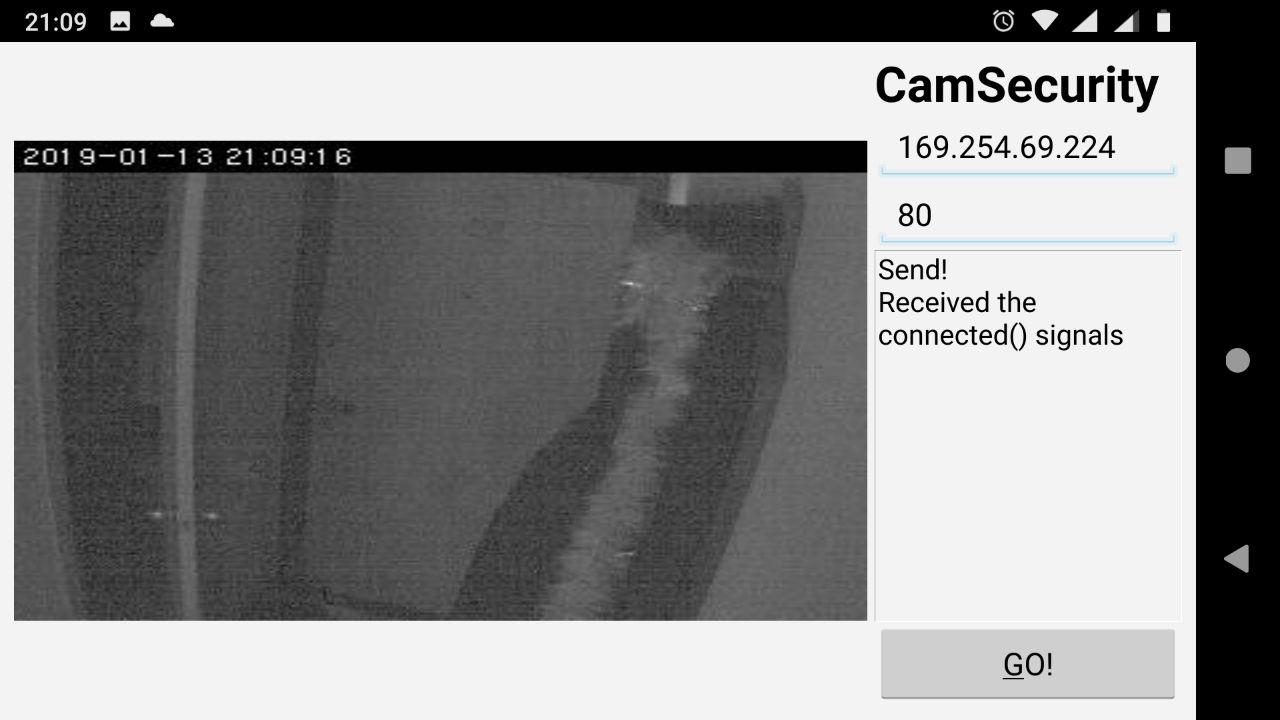


Рисунок 8 – Результат роботи програми