

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

ДОПУСТИТИ ДО ЗАХИСТУ  
Завідувач кафедри

\_\_\_\_\_Литвиненко О.Є.

«\_\_\_» \_\_\_\_\_ 2021 р.

**ДИПЛОМНА РОБОТА**  
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ  
"БАКАЛАВР"**

Тема: \_\_\_\_\_ Програмний модуль системи відеоспостереження  
\_\_\_\_\_ приватного будинку

Виконавець: \_\_\_\_\_ Мізинець О.С.

Керівник: \_\_\_\_\_ Нечипорук О.П.

Нормоконтролер: \_\_\_\_\_ Тупота Є.В.

**Київ 2021**

# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет \_\_\_\_\_ кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютеризованих систем управління

Освітнього ступеня \_\_\_\_\_ бакалавр

Напрямок (спеціальність) 123 "Комп'ютерна інженерія"  
(шифр, найменування)

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Литвиненко О. Є.

« \_\_\_\_\_ » \_\_\_\_\_ 2021 р.

## ЗАВДАННЯ на виконання дипломної роботи (проекту)

Мізинця Олександра Сергійовича

(прізвище, ім'я, по батькові)

**1. Тема роботи:** \_\_\_\_\_ "Програмний модуль системи відеоспостереження  
\_\_\_\_\_ приватного будинку"

затверджена наказом ректора від "04" \_\_\_\_\_ лютого 2021 року № 135 /ст.

**2. Термін виконання роботи:** з \_\_\_\_\_ 21.12.2019 до \_\_\_\_\_ 28.02.2020

**3. Вихідні дані до роботи:** 1) вимоги до програми;  
2) основні операції хмарного сховища.

**4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):**

1) аналіз принципів роботи систем контролю і управління доступом;  
2) принципи роботи систем відеоспостереження;  
3) описання програмної реалізації інтеграції системи контролю управління доступом з системою відеоспостереження.

**5. Перелік обов'язкового графічного матеріалу:**

1) діаграма класів механізму складування об'єктів;  
2) діаграма класів механізму команд;  
3) основні модулі пакету «Система зв'язку сервера»;  
4) схема алгоритму занесення даних з камер до файлів сервера.

## 6. Календарний план

№ п/п	Етапи виконання дипломної роботи	Термін виконання етапів	Примітка
1	Провести аналіз літератури за темою дипломної роботи та аналіз існуючих систем	17.05.21-19.05.21	
2	Зробити вибір компонентів системи	19.05.21-23.05.21	
3	Розробити структуру програмних засобів системи цифрового репозитарію	24.05.21-26.05.21	
4	Розробити програмні засоби цифрового репозитарію	27.05.21-29.05.21	
5	Провести налаштування програмних засобів на сервері	30.05.21-02.06.21	
6	Написати пояснювальну записку	03.06.21-10.06.21	
7	Підготувати презентацію	11.06.21-14.06.21	

## 7. Дата видачі завдання « 17 » травня 2021 р.

Керівник дипломного проекту \_\_\_\_\_ Нечипорук О.П.  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_ Мізинець О.С.  
(підпис студента)

## РЕФЕРАТ

Пояснювальна записка до дипломного проекту “Програмний модуль системи відеоспостереження приватного будинку”: 59 с., 16 рис., 19 літературних джерел, 1 додаток.

ІНТЕГРОВАНІ ОХОРОННІ СИСТЕМИ, НЕСАНКЦІОНОВАНИЙ ДОСТУП, СИСТЕМА КОМПЛЕКСНОЇ БЕЗПЕКИ, КЛАСИ ЗАХИСТУ, ОХОРОНА ПРИМІЩЕНЬ, ПРОГРАМНА ІНТЕГРАЦІЯ КОМПОНЕНТІВ ОХОРОННОЇ СИСТЕМИ

Об’єкт дослідження – інтеграція системи контролю доступу з системою відеоспостереження.

Предмет дослідження – програмний модуль системи відеоспостереження приватного будинку.

Метою роботи є інтеграція системи контролю доступу з системою відеоспостереження.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ.....	6
ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРИНЦИПІВ РОБОТИ СИСТЕМ КОНТРОЛЮ І УПРАВЛІННЯ ДОСТУПОМ .....	9
1.1. Основні задачі охорони приватних будинків.....	9
1.2. Огляд існуючих систем контролю доступу на об'єкт.....	13
1.3. Висновки до розділу .....	16
РОЗДІЛ 2 ПРИНЦИПИ РОБОТИ СИСТЕМ ВІДЕОПОСТЕРЕЖЕННЯ.....	18
2.1. Структура і основні елементи систем відеоспостереження .....	18
2.2. Інтегровані системи безпеки.....	28
2.3. Висновки до розділу .....	34
РОЗДІЛ 3	35
ОПИСАННЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ІНТЕГРАЦІЇ СИСТЕМИ КОНТРОЛЮ УПРАВЛІННЯ ДОСТУПОМ З СИСТЕМОЮ ВІДЕОПОСТЕРЕЖЕННЯ .....	35
3.1. Описання існуючої СКУД.....	35
3.2. Архітектура програмного забезпечення інтеграції з СКУД.....	41
3.3. Збереження об'єктів.....	44
3.4. Висновки до розділу .....	56
ВИСНОВКИ.....	57
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
Додаток А 61	
Лістинг основного програмного модуля.....	61

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

АС	– автоматизована система
БД	– база даних
ЕОМ	– електронно-обчислювальна машина
ЗВ	– засоби відображення
ОС	– операційна система
ОЗП	– оперативний запам'ятовуючий пристрій
ППП	– пакет прикладних програм
СКУД	– система керування управлінням доступом

## ВСТУП

Бачити на власні очі - це найкращий спосіб мати точні відомості про все, що стосується вашого життя. Відеокамера дозволяє бачити те, що зазвичай складно побачити складно побачити на власні очі, додає можливостей будинку і на роботі і допомагає уникнути неприємностей.

Ні вдома, ні в офісі ви не застраховані від крадіжки, зловживань співробітників або домашніх працівників. Відеоспостереження особливо важливо, коли мова - про наших найдорожчих людей: для забезпечення безпеки дитини, який залишається під опікою няні. Або старшого члена сім'ї, або хворого родича, коли він знаходиться під наглядом доглядальниці. За допомогою камери ви завжди будете бачити, що відбувається з вашими рідними. Вам не доведеться даремно хвилюватися. І в той же час у разі потреби ви маєте можливість контролювати ситуацію, вчасно втручатися в хід подій.

Що відбувається в будинку, коли ви залишаєте його? Про це знято навіть кумедний повнометражний мульт, що виробляють домашні вихованці, коли вони надані самим собі. Тепер ви це дізнаєтеся достовірно, адже відео може транслюватися онлайн. І до речі, зможете перевірити скарги сусідів, які стверджують, що ваш улюблений пес голосно виє, коли вас немає, тому що в камері є мікрофон. А якщо ваш вихованець дійсно засумував, ви може його підбадьорити - в поворотній камері є і динамік! Рідний голос відразу заспокоїть одного. Або можна озвучити якусь команду, щоб він не втрачав форму.

За допомогою динаміка вдається навіть попередити підготовлюване злочин. Досить оголосити: "Я все бачу!", - і кримінальний запал напевно згасне. Але ж завжди краще попередити злочин, ніж потім ловити і карати злочинця.

Спостерігати - не завжди означає не довіряти. Бездротові Wi-Fi-камери, які можуть бути легко перенесені з місця на місце незамінний інструмент в маркетингу. З їх допомогою легко оцінювати поведінку покупців, наприклад. І з поправкою на це розміщувати товари до Ваших потреб. Крім того, за допомогою камери дуже зручно контролювати наявність товару на полиці. І в разі потреби,

без скарг і нагадувань з боку покупців - поповнювати запас. Це - прямий шлях до збільшення прибутковості бізнесу, підвищення лояльності покупців і зміцненню іміджу продавця.

Взагалі в будь-якій ситуації, коли довірені особи (співробітники, помічники) допущені до управління товарно-матеріальними цінностями - відеокамера може зіграти визначальну роль. Все не без гріха, і у деяких ця межа дозволеності часом не збігається з виробничими нормами. Про таку ситуацію навіть є приказка: «По воді ходити - так не напиться!» І вже якщо через руки людини проходять десятки і сотні товарів - що-небудь може до цих рук і пристане. Щоб уникнути втрат і не завжди успішних розглядів, відеокамера - кращий аргумент. Уже сам факт установки камери на робочому місці дисциплінує і позбавляє працівника спокуси. Ну, а якщо зловживання сталося, відео - кращий, найвагоміший і чарівний аргумент в суперечці.

Як бачимо, камери відеоспостереження - необхідний компонент безпеки і поліпшення управління в професійній сфері. А в побуті - простий і доступний спосіб істотно підвищити якість життя. Їх оцінять як в офісі, в виробничих приміщеннях різного призначення, магазинах - так і в житловому будинку.

Об'єкт дослідження – інтеграція системи контролю доступу з системою відеоспостереження.

Предмет дослідження – програмний модуль системи відеоспостереження приватного будинку.

Метою роботи є інтеграція системи контролю доступу з системою відеоспостереження.



РОЗДІЛ 1  
АНАЛІЗ ПРИНЦИПІВ РОБОТИ СИСТЕМ КОНТРОЛЮ І УПРАВЛІННЯ  
ДОСТУПОМ

1.1. Основні задачі охорони приватних будинків

Рано чи пізно це трапляється з кожною компанією: вам знадобиться нова система відеоспостереження. Це може бути система, яку ви встановлюєте вперше або хмарний апгрейд вже існуючих камер - в будь-якому випадку вам потрібно зробити щось для посилення контролю. І поки вас будуть займати функції і переваги різних технологій, фінансову службу зацікавить питання повернення інвестицій. Легко уявити, що сучасне відеоспостереження - це просто, зручно і надійно. Але чи вигідно? Скільки тисяч потрібно віддати за «просто», «зручно», «надійно»? Як відбити вкладення? Сьогодні ми розглянемо, як вибрати систему відеоспостереження, щоб співвіднести витрати з вигодою, і покажемо кілька калькуляторів, які допоможуть вам швидко зорієнтуватися у вимогах до камер і архіву відео.

Обрана вами система повинна, як мінімум, не бути збитковим рішенням. Розглянемо абсолютно звичайний випадок в малому бізнесі. Система відеоспостереження (разом з сервісом розпізнавання осіб) Допомогла охорони затримувати на 15% більше магазинних злодіїв. Середня вартість збитку від кожного злодія становить 5 000 рублів, а кількість щорічних інцидентів - 80. Зростання ефективності охорони на 15% додатково економить бізнесу 60 000 рублів - ось сума, яку вже можна враховувати при розрахунку інвестицій.

Кафедра КСУ				НАУ 21 09 63 000 ПЗ			
<i>Виконав</i>	<i>Мізинець О.С.</i>			Аналіз принципів роботи систем контролю і управління доступом	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Нечипорук О.П.</i>				Д	9	59
<i>Консульт.</i>					123 СП 436Б		
<i>Норм. контр.</i>	<i>Тупота Є.В.</i>						
<i>Зав. Каф.</i>	<i>Литвиненко О.Є.</i>						

Відеоспостереження - неразова історія, обмежена одним набором функцій. Вигода зростає щорічно в міру розширення можливостей. Ви можете розпізнати обличчя людей з «чорного списку» або лояльних клієнтів, враховувати робочий час, контролювати касові операції, аналізувати потік відвідувачів - і все на одних камерах. При цьому постійними витратами (на хмарних рішеннях) будуть: електрику, інтернет, тарифний план. Ось, власне, і все - в хмарі для клієнтів немає витрат на реєстратори, записуючі пристрої та амортизацію.

Які завдання потрібно вирішити

Це повинен бути ваш перший питання при виборі системи. Причини установки відеоспостереження визначають, які функції знадобляться і скільки камер потрібно встановити.

З досвіду Ivideon завдання відеоспостереження часто ідентичні у різних компаній. Загальні практики застосування камер включають в себе:

- Контроль крадіжки - видимі камери самі по собі є стримуючим фактором і можуть запобігти крадіжкам.
- Відеодокази ПП - відеоспостереження дає реальну інформацію про різні інциденти.
- Сторонній моніторинг - можливість спостерігати за об'єктами з будь-якої точки світу.
- Контроль співробітників - безконтактний пропуск в приміщення по обличчю, облік робочого часу, розмежування прав доступу на об'єкті, контроль викладки товарів.
- Контроль касових операцій - запобігання шахрайським діям касира з відеопідтвердження.
- Аналіз потоку клієнтів - підрахунок відвідувачів; розпізнавання емоцій, демографії та статі; детекція черг; робота з лояльними клієнтами через інтеграцію з CRM.

Які системні вимоги

У вас можуть бути різні вимоги до камер для різних областей. Наприклад, відстань нічного бачення може бути ключовим фактором для зовнішнього

спостереження.

Також варто відразу вирішити, як буде здійснюватися робота з аналітикою. Уже кілька років на ринку присутні камери з вбудованими аналітичними модулями. Рішення подібного класу дозволяють економити на трафіку, що важливо при розгортанні систем на об'єктах з обмеженою пропускною спроможністю каналів зв'язку. З мінусів можна відзначити високу ціну, складне і дороге технічне обслуговування. Оновлення та масштабування системи, побудованої на базі локальної аналітики, також утруднено.

Для підключення системи відеоспостереження можна використовувати цифровий відеореєстратор (DVR), мережевий відеореєстратор (NVR), гібридні пристрої (як Ivideon Bridge), ПК або використовувати вбудовану прошивку самих камер. До DVR підключаються аналогові камери - записи обробляються безпосередньо на пристрої. NVR працюють з IP-камерами, як по дроту, так і бездротовим способом. Гібридні пристрої працюють з будь-якими камерами. Камери, які мають вбудовану підтримку хмари або відеоаналітики, потребують тільки в стабільному інтернет-каналі.

Правда, при виборі обладнання в традиційних системах відеоспостереження потрібно враховувати, що не всі пристрої сумісні один з одним. Наприклад, деякі NVR сумісні з IP-камерами тільки певних виробників.

Одна з ключових характеристик системи відеоспостереження - ємність сховища. У випадку з хмарним зберіганням записів питання відпадає сам собою - ніяких обмежень на трафік немає, а термін зберігання запису залежить тільки від кількості доступних днів на тарифному плані. При виборі локальної записи на РЕГИСТОР, ПК або на SD-карті камери потрібно враховувати заздалегідь, скільки пам'яті знадобиться. Відповідь залежить від декількох факторів - докладніше розглянемо в розділі з калькуляторами.

Ціна рішення буде залежати від усіх цих параметрів, а також від будь-яких інших функцій, які ви можете вибрати. З огляду на цей факт, не варто дивуватися, що вартість систем відеоспостереження для малого бізнесу може відрізнятися на порядок.

Калькулятори розрахунку обсягу сховища і трафіку

Система з 4-х IP-камер, яка працює 24 години на добу, з роздільною здатністю 2 мегапікселя і частотою 5 кадрів / сек, генерує 2,79 терабайта MJPEG відео на NVR. Навіть один постійний відеопотік з частотою 20 кадрів / сек і мінімальним дозволом 704 × 480 може заповнити диск ємністю 1 ТБ за 42 дні.

У найпростішому випадку визначити обсяг пам'яті можна за дозволом, кількістю камер і часу, протягом якого ви хочете зберегти відзнятий матеріал. Можете перевірити тут, тут або тут.

Біда в тому, що ці калькулятори (і багато інших) дають різні результати при введенні одних і тих же параметрів! На щастя, є формула, яка дозволяє власноруч отримати близькі до правди дані:

Місце для зберігання (ГБ) = бітрейт (Кбіт / сек) \* 1000/8 \* 3600 \* 24 \* Камери \* Дні / 1000 000 000

де:

- 1000/8 = конвертувати в байти;
- 3600 = конвертувати з секунд на годину;
- 24 = конвертувати з години на день;
- Камери = загальна кількість камер;
- Дні = загальна кількість днів запису.

Якщо у вас є 2 камери, максимальна швидкість яких становить 1024 Кбіт / сек, і ви хочете записувати 7 днів без перерви, то:

Необхідну дисковий простір = 1024 \* 1000/8 \* 3600 \* 24 \* 2 \* 7/1000 000 000 = 154,8288 ГБ

Ви також можете використовувати формулу розрахунку пропускної здатності мережі для IP-камери, яка допоможе розрахувати вимоги до швидкості інтернет-каналу:

Пропускна здатність (Мбіт / сек) = бітрейт (основний) \* N + бітрейт (суб) \* M

де:

- N & M - кількість IP-камер для основного потоку і підпотоків.

Що стосується бітрейта, ви можете дізнатися його в технічних характеристиках IP-камер.

Приклад: якщо у вас є одна IP-камера з основним потоком 6144 кбіт / с, а інша з підпотоків тисячі двадцять чотири кбіт / с, необхідна пропускна здатність

буде дорівнює:

Необхідна пропускна здатність =  $1 * 6144 \text{ кбіт / сек} + 1 * 1024 \text{ кбіт / сек} = 7,168 \text{ Мбіт / сек}$ .

Інші калькулятори для систем відеоспостереження

## 1.2. Огляд існуючих систем контролю доступу на об'єкт

У кожному окремо взятому випадку перед системою ставляться певні завдання. Це або охоронна функція, або забезпечення контролю, нагляд. Що Ви, як замовник, хочете отримати на виході від системи? В цьому ми сьогодні і спробуємо розібратися.

Ви - замовник. Перед вами стоїть прикладна задача, наприклад, покрити територію продовольчого магазину, з торговою площею, підсобними приміщеннями, паркуванням. При цьому, бюджет у вас строго лімітований, але, як зазвичай водиться в РФ «Бажано б подешевше». На цьому етапі починається пошук максимально бюджетних рішень. Ринок рясніє пропозиціями «монтаж системи з 5 вуличних камер всього за 20000 грн.» і подібними. Можна сміливо відкидати подібні пропозиції, банально керуючись здоровим глуздом. Навіть якщо систему за ці гроші встановлять, толку від неї не буде ніякого.

Якщо мова йде про продовольчому магазині, найчастіше ставиться завдання контролю персоналу і крадіжок товару (як покупцями так і персоналом безпосередньо). Стало бути від системи потрібно, щоб картинка була чітка, повинні бути добре помітні особи, самі товари. Для камери над касою вимоги ще вище. Бажано, щоб були помітні номінали банкнот. Системи за 20-30 тисяч грн. на 4 камери (з установкою і налаштуванням) ну ніяк не зможуть дати такої якості картинки. Отже на виході ми отримаємо зернисту картинку, силует людини, що пограбував вас, і ніякої юридичної підтримки в особі органів правопорядку. І добре, якщо це буде баночка коли, а не плазма за 100к +.

Разом, ми отримуємо:

- Витрачений час і гроші
- ніякої користі

- Даремні залізяки, що займають місце

На цьому годі було робити висновок, що відеореєстратор з бюджетної лінійки не має свого примінення. Подібний варіант добре підійде для нагляду за паркуванням (якщо камери дозволяють хоча б розрізнити номерні знаки), нагляду за під'їздом. Грубо кажучи в областях, не пов'язаних з фінансовими ризиками. Якщо ж мова йде про комерційну площі, будь то магазин, салон, ресторан і т.д. - потрібен дещо інший варіант.

Якщо ви все ще не відмовилися повністю від системи з відеореєстратором, то давайте розглянемо трохи інший варіант. Припустимо, побіжний пошук в інтернеті привів вас на сайт магазину, який продає реєстратори і камери стандарту HD-SDI. Ви, приблизно прикинувши геометрію приміщення, вирішили, що вам цілком вистачить 8 камер, щоб покрити і парковку, і торговий зал з касами, і підсобне приміщення. Це цілком реальний сценарій для невеликого магазину площею 150-200 кв.м. Наприклад, ви замовили собі реєстратор на 8 каналів з 8 камерами, припустимо 2 вуличних і 6 для приміщень. Бюджет, зрозуміло виріс в рази. Один тільки реєстратор стандарту HD-SDI коштує близько 15.000 р. На камерах ви вирішили заощадити. На момент пошуку, самими бюджетними рішеннями виявилися:

- Мініатюрна камера 1.3mpx HD за 3400
- Вулична камера 1.3 mpx HD за 4300
- Сам реєстратор за 14.600

Разом ми отримали бюджет тільки апаратної частини 43.600 грн.. Ціна без монтажу, без настройки. За це все зазвичай беруть ще 10-15 тисяч зверху. Гарзд, розщедрилися, віддали 60.000 плюс мінус. Отримали хорошу картинку, детекцію руху, ІК підсвічування. Все добре, все красиво. Але треба це якось моніторити. Монітора в комплекті немає. Додаємо сміливо ще 5000 до бюджету. Запускаємо систему, все видно чітко. Але є 1 нюанс. При виборі камер, ви не подивилися на кути огляду. А у найдешевших моделей, які я вибрав для прикладу, вони такі 25-60 градусів для вуличної камери і 52 градуса для купольної. Хочеш не хочеш, а замало буде. І видно не так багато. Зіткнувшись з такою проблемою після монтажу, вже досить проблематично що або поміняти.

Як не крути, а крадіжки продовжуються. Що ми можемо зробити в подібній ситуації? Практично нічого, тому що система не піддається розширенню принципово. Додатково каналів на реєстратор не додати.

Самим бюджетним рішенням даної проблеми буде перенесення вуличних камер на інший реєстратор наприклад, і додавання ще двох камер з великим кутом огляду в приміщення. А це ще витрати і на монтаж і на нове обладнання. Таким чином система поступово підбирається до 100.000 грн. за 8 камер з картинкою HD.

Раптом, температура падає до -30, і ваші вуличні камери спочатку починають «артефачити» а потім просто відключаються ... Хо-хо-хо скупий замовник. А не подивився ти на температурний режим вуличних камер ... А заявлена температура то до -20 ... Знову заміна, знову монтаж, знову гроші.

Припустимо, що Ви зробили всі ці моторошні і лякають дії, вдруге провели генеральне прибирання магазину після монтажу. Сидіть собі спокійно в кабінеті і дивіться 2 екрану. П'єте чай, дивіться на годинник, час 9-ї вечора. Час йти додому. Ви прийшли додому, повечеряли, лягли спати. А о 3 годині ночі вам дзвонить, наприклад, майор поліції Фролов, і дуже офіційно говорить, що Ваш магазин вночі піддався пограбування.

Потрібно було копіювати дані кудись ще, скажете ви. Абсолютно вірно. І практично всі реєстратори з коробки підтримують технологію FTP. Тільки скажіть чесно, ви хоч раз бачили, щоб хтось на відеореєстраторові (вони з пультами зазвичай, а не з клавіатурами) налаштував бекап на якийсь окремо піднятий FTP сервер з вільним терабайтом місця, який повинен стояти десь в дата центрі в ідеалі, ну або хоча б в квартирі господаря цього самого магазину? Я, за свою практику, не бачив жодного разу. Більш того, я жодного разу не бачив, щоб підприємства, які займаються продажем і установкою систем, пропонували таку опцію. У теорії - можливо. На практиці - юзабіліті кульгає.

У вас з'явилися вимоги до системи:

1. Система повинна бути продуманим
2. Картинка повинна бути чіткою
3. Камери в приміщенні повинні бути з широким кутом огляду (70+)

4. Камера на касі повинна бути з невеликим кутом огляду (25-40)
5. Вулична камера повинна бути з підсвічуванням і працювати при температурі нижче 30 градусів (Росія все таки)
6. Архів повинен дублюватися на зовнішнє сховище
7. Потрібна можливість постійного моніторингу системи (дада, ви тепер параноїк. Ще б після такого ...)
8. Система не повинна коштувати як новий лексус
9. Система повинна масштабуватися

З огляду на всі вищевикладені вимоги, ви починаєте порівнювати ціни на ринку відеореєстраторів та IP відеоспостереження. Досвід минулих помилок вас навчив того, що резервне копіювання дуже важливо. Відштовхуючись від цього, ви ризикуєте, і ставите собі IP систему. Підбираєте камери, встановлюєте, і потрапляєте в рай. Камери цілком собі пристойної якості коштують близько 5 тисяч. Сервер в межах 15 (з монітором, мишею, клавіатурою). Комутатор та вітуха, ну нехай ще 3 тисячі грн.. Монтаж всього цього добра - близько 10 тисяч (простіше все таки, без метушні з харчуванням, зазвичай таки PoE). Навіть якщо купити ще один комп і поставити його додому як дублюючий - цінник вище 50.000 грн. не підніметься. Але зате:

Якщо промахнувся, і залишилося сліпа пляма, докупив камеру і встроїв в комутатор - Масштабіруемость. Архів можна дублювати на хмару (Ivideon наприклад вміє) або інший сервер - Безопасность. Возможность замінити будь-яку з камер на дорожчу і якісну без втрат часу і грошей - Гнучкість .Просмотр з будь-якого пристрою підключеного до мережі - Доступность. Вся система цілком коштує значно дешевше аналогових друзів - Економічна доцільність.

### 1.3. Висновки до розділу

На основі аналізу інформації з розділу можна правильно зорієнтуватися і акцентувати увагу на неочевидних проблемних місцях в системах відеоспостереження і уникнути помилок, які можуть призвести до серйозних наслідків. IP відеоспостереження – не є вирішенням всіх проблем, десь



доцільніше ставити аналоговий реєстратор, деє підключення до інтернет для юр. осіб буде дорожче ніж сам цей реєстратор варто.

Університет Флориди (США) проводить щорічне Національне дослідження безпеки роздрібної торгівлі (NRSS), щоб анонімно збирати показники втрат в роздрібних продажах. Це дослідження стало основою калькулятора повернення інвестицій для систем відеоспостереження. До речі, за даними NRSS 25% втрат в роздрібній торгівлі можна запобігти за допомогою звичайної системи відеоспостереження (навіть без розпізнавання осіб).

зручний калькулятор об'єктивів з фіксованою фокусною відстанню допоможе визначити, який саме об'єктив на камері вам підходить.

наступний калькулятор при виборі об'єктивів враховує і сценарії використання.

використовуйте цей інструмент для порівняння налаштувань частоти кадрів в DVR і ви побачите, як зниження частоти кадрів впливає на плавність відео.

Нарешті, варто згадати про професійний софт, який знаходить застосування в монтажних організаціях. Компанії за допомогою даних рішень автоматизують більшість рутинних завдань інженерів і наочно демонструють замовнику, що він отримає в результаті інсталяції камер. При наявності вільного часу і інтересу, ви можете самостійно ознайомитися з настройками IPVM Camera Calculator, CCTV Design Lens Calculator або JVSG.

## РОЗДІЛ 2

### ПРИНЦИПИ РОБОТИ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ

#### 2.1. Структура і основні елементи систем відеоспостереження

Витрати на зберігання даних найчастіше стають основним пунктом витрат при створенні системи відеоспостереження. Втім, вони були б незрівнянно більше, якби в світі не існувало алгоритмів, здатних стискати відеосигнал. Проте, наскільки ефективні сучасні кодеки, і які принципи лежать в основі їх роботи, ми і поговоримо в сьогоднішньому матеріалі.

Для більшої наочності почнемо з цифр. Нехай відеозапис буде вестися безперервно, в дозволі Full HD (зараз це вже необхідний мінімум, у всякому разі, якщо ви хочете повноцінно використовувати функції відеоаналітики) і в режимі реального часу (тобто, з фреймрейтом 25 кадрів в секунду). Припустимо також, що обраний нами обладнання підтримує апаратне кодування H.265. В цьому випадку при різних настройках якості зображення (високому, середньому і низькому) ми отримаємо приблизно такі результати.

Але якби стиснення відео не існувало в принципі, ми б побачили зовсім інші цифри. Спробуємо розібратися, чому. Відеопотік являє собою не що інше, як послідовність статичних картинок (кадрів) в певному вирішенні. Технічно кожен кадр є двовимірним масивом, що містить інформацію про елементарні одиницях (пікселях), які формують зображення. В системі TrueColor для кодування кожного пікселя потрібно 3 байта. Таким чином, у наведеному прикладі ми б отримали бітрейт:

$$1920 \times 1080 \times 25 \times 3 / 1048576 = \sim 148 \text{ Мб / с}$$

Кафедра КСУ				НАУ 21 09 63 000 ПЗ			
Виконав	Мізинець О.С.			Принципи роботи систем відеоспостереження	Літера	Аркуш	Аркушів
Керівник	Нечипорук О.П.				Д	18	59
Консульт.					123 СП 436Б		
Норм. контр.	Тупота С.В.						
Зав. Каф.	Литвиненко О.Є.						

кодк	Інтенсивність руху в кадрі	Використання дискового простору за добу, ГБ
Н.265 (Висока якість)	висока	138
Н.265 (Висока якість)	Середня	67
Н.265 (Висока якість)	низька	41
Н.265 (Середня якість)	висока	86
Н.265 (Середня якість)	Середня	42
Н.265 (Середня якість)	низька	26
Н.265 (Низька якість)	висока	81
Н.265 (Низька якість)	Середня	39
Н.265 (Низька якість)	низька	24

З огляду на, що в добі 86400 секунд, цифри виходять воістину астрономічні:

$$148 \times 86400/1024 = 12487 \text{ ГБ}$$

Отже, якби ми записували відео без стиснення в максимальній якості при заданих умовах, то для зберігання даних, отриманих з однієї єдиної відеокамери протягом доби нам би треба було 12 терабайт дискового простору. Але навіть система безпеки квартири або малого офісу передбачає наявність, як мінімум, двох пристроїв відеофіксації, тоді як сам архів необхідно зберігати протягом декількох тижнів або навіть місяців, якщо того вимагає законодавство. Тобто, для обслуговування будь-якого об'єкта, навіть вельми скромних розмірів, потрібен був би цілий дата-центр!

На щастя, сучасні алгоритми стиснення відео допомагають істотно економити дисковий простір: так, використання кодека H.265 дозволяє скоротити обсяг відео в 90 (!) Разів. Домогтися таких вражаючих результатів вдалося завдяки цілому стеку різноманітних технологій, які давно і успішно застосовуються не тільки в сфері відеоспостереження, але і в «цивільному» секторі: в системах аналогового і цифрового телебачення, в аматорській і професійній зйомці, і багатьох інших ситуаціях.

Найбільш простий і наочний приклад - колірна субдискретизація. Так називають спосіб кодування відео, при якому навмисно знижується колірне дозвіл кадрів і частота вибірки цветоразностних сигналів стає менше частоти вибірки сигналу яскравості. Такий метод стиснення відеоданих повністю виправданий як з позиції фізіології людини, так і з точки зору практичного застосування в області відеофіксації. Наші очі добре помічають різницю в яскравості, однак набагато менш чутливі до перепадів кольору, саме тому вибіркою цветоразностних сигналів можна пожертвувати, адже більшість людей цього просто не помітить. У той же час, складно уявити, як в розшук оголошують машину кольору «павука, що замислюють злочин»: в орієнтуванні буде написано «темно-сірий», і це правильно, адже інакше прочитав опис авто навіть не зрозуміє,

А ось зі зниженням деталізації все виявляється вже зовсім не так однозначно. Технічно квантування (тобто, розбиття діапазону сигналу на деяке число рівнів з подальшим їх приведенням до заданих значень) працює чудово: використовуючи даний метод, розмір відео можна багаторазово зменшити. Але так ми можемо втратити важливі деталі (наприклад, номер проїжджаючого далеко автомобіля або риси обличчя зловмисника): вони виявляться змащені і такий запис буде для нас марною. Як же бути в цій ситуації? Відповідь проста, як і все геніальне: варто взяти за точку відліку динамічні об'єкти, як все тут же стає на свої місця. Цей принцип успішно використовується з часів появи кодека H.264 і відмінно себе зарекомендував, відкривши ряд додаткових можливостей для стиснення даних.

Це було передбачувано: розбираємося, як H.264 стискає відео

Повернемося до таблиці, з якої ми почали. Як бачите, крім таких параметрів, як дозвіл, фреймрейт і якість картинки вирішальним фактором, що визначає кінцевий розмір відео, виявляється рівень динамічності сцени, що знімається. Це пояснюється особливостями роботи сучасних відеокодеків взагалі, і H.264 зокрема: використовуваний в ньому механізм передбачення кадрів дозволяє додатково стискати відео, при цьому практично не жертвуючи якістю картини. Давайте подивимося, як це працює.

Кодек H.264 використовує кілька типів кадрів:

I-кадри (від англійського Intra-coded frames, їх також прийнято називати опорними або ключовими) - містять інформацію про статичних об'єктах, не змінних на протязі тривалого часу.

P-кадри (Predicted frames, передбачені кадри, також іменовані різницевими) - несуть в собі дані про ділянки сцени, які зазнали змін у порівнянні з попереднім кадром, а також посилання на відповідні I-кадри.

B-кадри (Bi-predicted frames, або двонаправлені передбачені кадри) - на відміну від P-кадрів, можуть посилатися на I-, P- і навіть інші B-кадри, причому як на попередні, так і на наступні.

Що це означає? У кодеку H.264 побудова відеозображення йде наступним чином: камера робить опорний кадр (I-кадр) і вже на його основі (тому він і називається опорним) виробляє віднімання з кадру нерухомих частин зображення. Таким чином створюється P-кадр. Потім з цього другого кадру віднімається третій і також створюється P-кадр зі змінами. Так формується серія різницевих кадрів, які містять тільки зміни між двома послідовними кадрами. В результаті ми отримуємо такий ланцюжок:

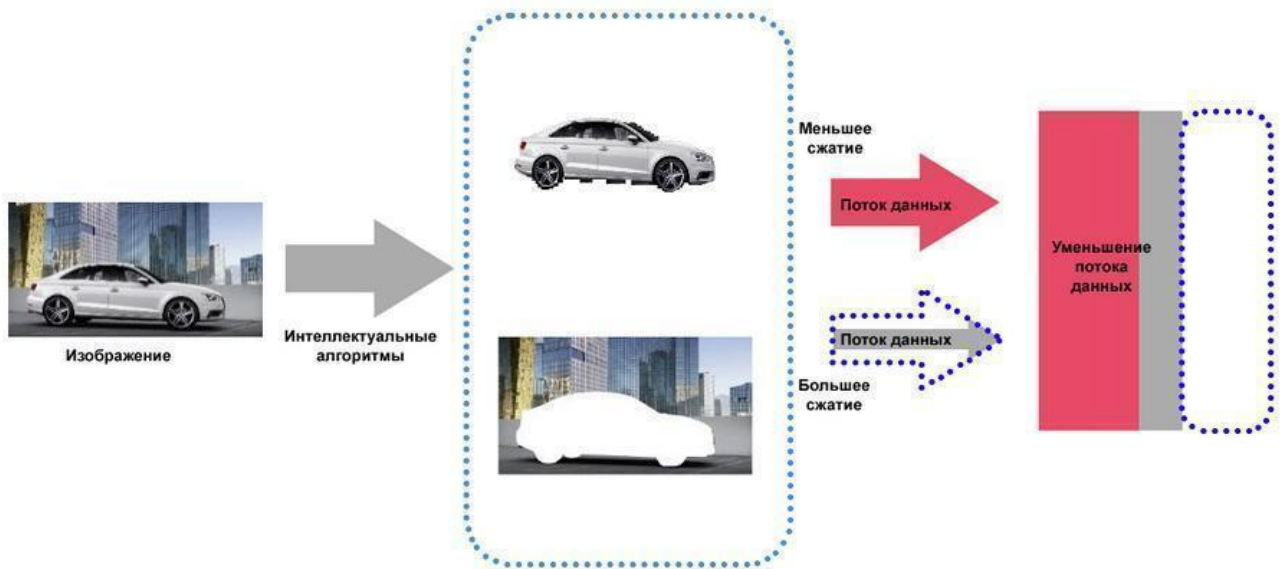
[ПОЧАТОК ЗЙОМКИ] IRRRRRRRRRRRRR- ...

Оскільки в процесі віднімання можливі помилки, що призводять до появи графічних артефактів, то через якусь кількість кадрів схема повторюється: знову формується опорний кадр, а слідом за ним - серія кадрів зі змінами.

... -RRRRRRRIRRRRRR ...

Повне зображення формується шляхом «накладання» P-кадрів на опорний кадр. При цьому з'являється можливість незалежної обробки фону і рухомі

об'єктів, що дозволяє додатково заощадити дисковий простір без ризику втратити важливі деталі (риси осіб, автомобільні номери і т. Д.). У випадку ж з об'єктами, які здійснюють одноманітні рухи (наприклад, що обертаються колесами машин) можна багаторазово використовувати одні і ті ж різниці кадрів.



Незалежна обробка статичних і динамічних об'єктів дозволяє заощадити дисковий простір

Даний механізм носить назву межкадрового стиснення. Передбачені кадри формуються на основі аналізу широкої вибірки зафіксованих станів сцени: алгоритм передбачає, куди буде рухатися той чи інший об'єкт в поле зору камери, що дозволяє істотно знизити обсяг записуваних даних при спостереженні за, наприклад, проїзною частиною.



Кодек формує кадри, проорокуючи, куди буде рухатися об'єкт

У свою чергу, використання двонапрямлених передбачених кадрів дозволяє в кілька разів скоротити час доступу до кожного кадру в потоці,

оскільки для його отримання буде досить розпакувати тільки три кадри: В, що містить посилання, а також І і Р, на які він посилається. В даному випадку ланцюжок кадрів можна зобразити таким чином.

[ПОЧАТОК ЗЙОМКИ] ІВРВРВРВРВРВРВРВР- ...

Такий підхід дозволяє істотно підвищити швидкість швидкого перемотування з показом і спростити роботу з відеоархівом.

У чому різниця між Н.264 і Н.265?

У Н.265 використовуються всі ті ж принципи стиснення, що і в Н.264: фонове зображення зберігається один раз, а потім фіксуються лише зміни, джерелом яких є рухомі об'єкти, що дозволяє значно знизити вимоги не тільки до обсягу сховища, але і до пропускну здатності мережі. Однак в Н.265 багато алгоритми і методи прогнозування руху зазнали значних якісних змін.

Так, оновлена версія кодека стала використовувати макроблоки дерева кодування (Coding Tree Unit, CTU) змінного розміру з дозволом до  $64 \times 64$  пікселів, тоді як раніше максимальний розмір такого блоку становив лише  $16 \times 16$  пікселів. Це дозволило істотно підвищити точність виділення динамічних блоків, а також ефективність обробки кадрів в дозволі 4К і вище.

Крім того, Н.265 обзавівся поліпшеним deblocking filter - фільтром, що відповідає за згладжування кордонів блоків, необхідним для усунення артефактів по лінії їх стикування. Нарешті, покращений алгоритм прогнозування вектора руху (Motion Vector Predictor, MVP) допоміг помітно знизити обсяг відео за рахунок радикального підвищення точності прогнозів при кодуванні рухомих об'єктів, чого вдалося досягти за рахунок збільшення кількості відслідковуються напрямків: якщо раніше враховувалася лише 8 векторів, то тепер - 36.

Крім усього перерахованого вище, в Н.265 була поліпшена підтримка багатопоточних обчислень: квадратні області, на які розбивається кожен кадр при кодуванні, тепер можуть оброблятися незалежно одна від одної. З'явилася і підтримка хвильової паралельної обробки даних (Wavefront Parallel Processing, WPP), що також сприяє підвищенню продуктивності стиснення. При активації режиму WPP обробка CTU здійснюється через підрядник, зліва направо, проте кодування кожної наступної рядка може початися ще до завершення попередньої

в тому випадку, якщо даних, отриманих з раніше оброблених СТУ, для цього достатньо. Кодування різних рядків СТУ з тимчасовою затримкою із зсувом,

Флеш-карти для відеоспостереження: коли значення має не тільки розмір

І знову повернемося до таблички, з якої ми почали сьогоднішня розмова. Давайте підрахуємо, скільки дискового простору нам знадобиться в тому випадку, якщо ми хочемо зберігати відеоархів за останні 30 днів при максимальній якості відеозапису:

$$138 \times 30 / 1024 = 4$$

За нинішніми мірками 4 терабайта для вінчестера індустріального класу - практично ніщо: сучасні жорсткі диски для відеоспостереження мають ємність до 14 терабайт і можуть похвалитися робочим ресурсом до 360 ТБ в рік при MTBF до 1.5 мільйонів годин. Що ж стосується карт пам'яті, то тут все виявляється не так однозначно.

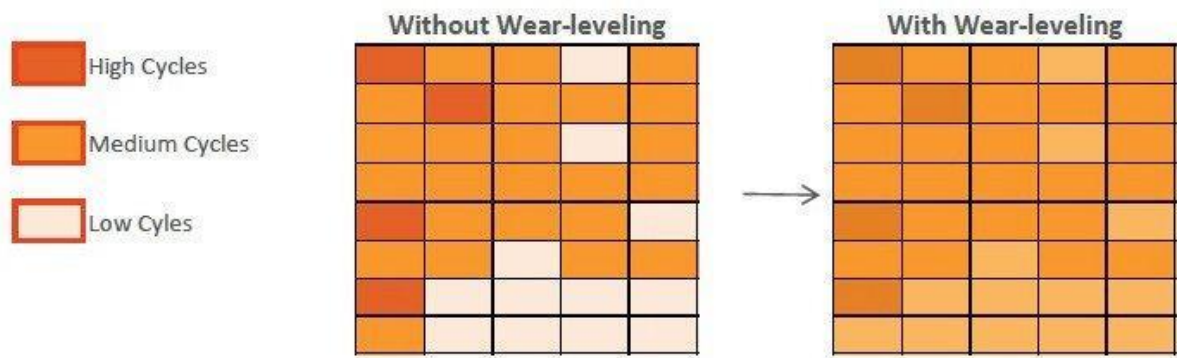
В IP-камерах флеш-карти грають роль резервних сховищ: дані на них постійно перезаписувати, щоб в разі втрати зв'язку з відео сервісом відсутній фрагмент відеозапису можна було відновити з локальної копії. Такий підхід дозволяє істотно підвищити відмовостійкість всієї системи безпеки, однак при цьому самі карти пам'яті відчувають колосальні навантаження.

Як видно з нашої таблиці, навіть при низькій якості зображення і за умови мінімальної активності в кадрі, всього за добу буде записано близько 24 ГБ відео. А це значить, що 128-гігабайтна картка буде повністю переписана менш ніж за тиждень. Якщо ж нам потрібно отримувати максимально якісну картинку, то всі дані на такому носії будуть повністю оновлюватися раз на добу! І це лише за дозволі Full HD. А якщо нам знадобиться картинка в 4K? В цьому випадку навантаження зросте практично в два рази (в заданих умовах відео в максимальній якості зажадає вже 250 ГБ).

При побутовому використанні подібне просто неможливо, тому навіть сама бюджетна карта пам'яті здатна прослужити вам декілька років поспіль без єдиного збою. А все завдяки алгоритмам вирівнювання зносу (wear leveling). Схематично їх роботу можна описати наступним чином. Нехай в нашому розпорядженні є нова флеш-карта, тільки що з магазину. Ми записали на неї



кілька відеороликів, використавши 7 з 16 гігабайт. Через деякий час ми видалили частину непотрібних відео, звільнивши 3 гігабайти, і записали нові, обсяг яких склав 2 ГБ. Здавалося б, можна задіяти тільки що звільнилося місце, однак механізм вирівнювання зносу виділить під нові дані ту частину пам'яті, яка раніше ніколи не використовувалася. Хоча сучасні контролери «тасують» біти і байти куди більш витончено, загальний принцип залишається незмінним.



Нагадаємо, що кодування бітів інформації відбувається шляхом зміни заряду в осередках пам'яті за рахунок квантового тунелювання електронів крізь шар діелектрика, що викликає поступове зношення діелектричних шарів з подальшим вибитком заряду. І чим частіше міняється заряд в конкретній комірці, тим раніше вона вийде з ладу. Вирівнювання зносу якраз спрямована на те, щоб кожна з доступних осередків перезаписувати приблизно однакову кількість разів і, таким чином, сприяє збільшенню терміну придатності карти пам'яті.

Неважко здогадатися, що wear leveling перестає грати хоч скільки-небудь значущу роль в тому випадку, якщо флеш-карта постійно перезаписується цілком: тут на перший план вже виходить витривалість самих чіпів. Найбільш об'єктивним критерієм оцінки останньої є максимальна кількість циклів програмування / стирання (program / erase cycle), або, скорочено, циклів P / E, яке здатне витримати флеш-пам'ять. Також досить точним і в даному випадку наочним (так як ми можемо заздалегідь розрахувати обсяги перезапису) показником є коефіцієнт TBW (Terabytes Written). Якщо в технічних характеристиках вказаний лише один з перерахованих показників, то обчислити інший не складе особливих труднощів. Досить скористатися наступною формулою:

$$TBW = (\text{Ємність} \times \text{Кількість циклів P / E}) / 1000$$

Так, наприклад, TBW флеш-карти ємністю 128 гігабайт, ресурс якої становить 200 P / E, буде дорівнює:  $(128 \times 200) / 1000 = 25,6$  TBW.

Давайте рахувати далі. Витривалість карт пам'яті споживчого рівня становить 100-300 P / E, і 300 - це в найкращому разі. Спираючись на ці цифри, ми можемо з досить високою точністю оцінити термін їх служби. Скористаємося формулою і заповнимо нову таблицю для карти пам'яті ємністю 128 ГБ. Візьмемо за орієнтир максимальну якість картинки в Full HD, тобто в добу камера буде записувати 138 ГБ відео, як ми з'ясували раніше.

Хочете використовувати карти на 64 ГБ або писати відео в 4K? Сміливо ділите розраховані терміни на два: в середньому споживчі карти пам'яті доведеться міняти раз на півроку, причому в кожній камері. Тобто кожні 6 місяців вам доведеться закупувати нову партію флеш-карт, нести додаткові витрати на сервісне обслуговування і, звичайно ж, наражати на небезпеку об'єкт, що охороняється, так як камери доведеться виводити з експлуатації на час заміни.

Нарешті, ще один момент, на який слід звернути пильну увагу при виборі карти пам'яті, - її швидкісні характеристики. В описі практично всіх сучасних флеш-карт можна зустріти запис виду: «Продуктивність: до 100 МБ / с при читанні, до 90 МБ / с при записі; запис відео: C10, U1, V10 ». Тут C10 і U1 означають не що інше, як клас швидкості запису відео, причому якщо заглянути в довідкові матеріали, то класам C10, U1 і V10 відповідає 10 МБ / с. Звідки різниця в 9 разів і чому маркування потрібна? Насправді все досить просто.

У розглянутому прикладі 100 і 90 МБ / с - це номінальна швидкість, тобто максимально досяжна продуктивність карти в операціях послідовного читання і запису за умови використання з сумісним обладнанням, яке саме по собі володіє достатньою продуктивністю. А показники C10, U1 і V1 (10 МБ / с) - це мінімальна стійка швидкість передачі даних в найгірших умовах тестування. Даний параметр необхідно враховувати при виборі карт для камер відеоспостереження по тій простій причині, що якщо він виявиться нижче бітрейта відеопотоку, то це загрожує появою на записи графічних артефактів і

навіть випаданням цілих кадрів. Очевидно, що у випадку з охоронними системами подібне неприпустимо: будь-які дефекти картинки загрожують втратою критично важливих даних - наприклад, доказів, які могли б допомогти при затриманні зловмисника.

Що ж стосується наявності відразу трьох маркувань, то причини цього суто історичні. С10 відноситься до найпершої з створених SD Card Association класифікацій, яка була складена ще в 2006 році, отримавши просту і нехитру назву Speed Class. Поява класифікації UHS Speed Class, на яку вказує маркування U1, пов'язане зі створенням інтерфейсу Ultra High Speed, який сьогодні використовується в переважній більшості флеш-карт. Нарешті, остання класифікація, Video Speed Class (V1), була розроблена SD Card Association в 2016-м в зв'язку з поширенням пристроїв, що підтримують запис відео надвисоких дозволів (4K, 8K і 3D).

Оскільки перелічені класифікації частково перетинаються, ми підготували для вас порівняльну таблицю, в якій швидкісні характеристики флеш-карт зіставлені між собою і співвіднесені з відео різного дозволу.

Слід враховувати, що наведені в таблиці відповідності актуальні для аматорських, напівпрофесійних і професійних відеокамер. В галузі відеоспостереження, де запис в реальному часі ведеться з максимальною частотою 25 кадрів в секунду, а для стиснення відеопотоку застосовуються високоефективні кодеки H.264 і H.265, що задіють кодування з пророкуванням, в переважній більшості випадків буде достатньо карт пам'яті, що відповідають класу U1 / V10, так як бітрейт в таких умовах практично ніколи не перевищує поріг в 10 МБ / с.

З урахуванням всіх перерахованих особливостей компанія Western Digital розробила спеціалізовану серію карт пам'яті WD Purple microSD, яка на даний момент включає в себе дві продуктові лінійки: WD Purple QD102 і WD Purple SC QD312 Extreme Endurance. До першої увійшли чотири накопичувачі об'ємом від 32 до 512 ГБ, в другу - три моделі, на 64, 128 і 256 ГБ. У порівнянні зі споживчими рішеннями, WD Purple були спеціально адаптовані під сучасні

цифрові системи відеоспостереження за рахунок впровадження ряду важливих удосконалень.

Головною перевагою пурпурової серії є істотно більший в порівнянні з побутовими пристроями робочий ресурс: карти лінійки QD102 здатні витримати 1000 циклів програмування / стирання, тоді як QD312 - вже 3000 циклів Р / Е, що дозволяє багаторазово продовжити термін їх служби навіть в режимі цілодобової записи, і робить дані картки ідеально відповідними для експлуатації на особливо охоронюваних об'єктах, де запис ведеться в режимі 24/7. У свою чергу, відповідність класам швидкості UHS Speed Class 1 і Video Speed Class 10 дозволяє використовувати карти WD Purple в камерах високого дозволу, в тому числі для запису в режимі реального часу.

Крім цього, карти пам'яті WD Purple мають і ряд інших важливих особливостей, про які необхідно згадати:

вологостійкість (виріб здатний витримати занурення на глибину до 1 метра в прісну або солону воду) і розширений діапазон робочих температур (від  $-25^{\circ}\text{C}$  до  $+85^{\circ}\text{C}$ ) дозволяють однаково ефективно використовувати карти WD Purple для оснащення як внутрішньобудинкових, так і вуличних пристроїв відеофіксації незалежно від погодних та кліматичних умов;

захист від впливу статичних магнітних полів з індукцією до 5000 Гс і стійкість до сильної вібрації і ударів аж до 500 g повністю виключають ймовірність втрати критично важливих даних навіть у разі пошкодження відеокамери;

функція віддаленого моніторингу допомагає оперативно відслідковувати стан кожної карти і ефективніше планувати проведення сервісних робіт, а значить, додатково підвищити надійність охоронної інфраструктури.

### 2.3. Інтегровані системи безпеки

Перспективи розвитку галузі відеоспостереження: можливості сучасних систем відеоаналітики

На думку агентства Transparency Market Research, вже до 2022 року загальний оборот ринку відеоспостереження досягне 71 мільярдів доларів США і буде збільшуватися як мінімум на 16,5% щорічно. Хоча в даний час попит підігривається такими факторами, як зросла небезпека терористичних атак і триваюче посилення законодавчих вимог до інфраструктури безпеки і контролю доступу, в найближчому майбутньому акценти змістяться в такі галузі, як ритейл, промисловість, охорона здоров'я та інші. Причиною подібних змін стає стрімкий розвиток інтелектуальних систем відеоаналітики, здатних істотно розширити функціонал звичних IP-камер.

Перша в світі система відеоспостереження була розроблена концерном Siemens в 1942 році спеціально для полігону Пенемюнде, де проводилися випробування ракет «Фау-2». Її спроектував інженер Вальтер Брух - «батько» стандарту аналогового кольорового телебачення PAL, прийнятого в 1966 році в якості державної в Німеччині, Великобританії та інших країнах Західної Європи. При всіх перевагах дане рішення мало один істотний недолік: оператор не міг ні на секунду відлучитися з робочого місця, так як сигнал жодним чином не фіксувався. Так тривало аж до 1951 року, коли з'явилися відеомагнітофони (Video Tape Recorder, VTR), створені компанією Ampex під керівництвом електроінженерії Олександра Понятова, який емігрував до США в кінці 1920-х років.

### Перший відеомагнітофон Ampex VR 1000B

Важливість цього винаходу для індустрії важко переоцінити, адже, згідно з дослідженням IMS Research, навіть фізично здоровий, добре відпочив людина починає пропускати до 45% критично важливих подій, що відбуваються на моніторі, вже через 12 хвилин роботи, а ще через 10 хвилин цей показник прагне до позначці 95%. З урахуванням цього факту, що стався в 2014 році IP-бум не здається таким вже дивним. Незважаючи на появу AHD (Analog High Definition - аналогове відео високої чіткості), мережеві камери виявилися значно більш

затребуваними, ніж поширені в той час аналогові рішення. І справа не тільки в якості картинки. Головне тут - великі можливості, які відкривають цифрові пристрої перед своїми власниками. Адже якщо раніше аналіз отриманої інформації здійснювався лише постфактум, то сьогодні системи безпеки дозволяють вирішувати найширший спектр завдань в реальному часі. Сучасні IP-камери різуче відрізняються від свого прабатька Axis Neteye 200, створеного в далекому 1996 році.

### IP-камера Axis Neteye 200

Багато моделей мережевих камер вже є цілком самодостатніми, а концепція IoT забезпечує можливість створення комбінованої інфраструктури, що об'єднує між собою перш розрізнені пристрої: охоронну сигналізацію, детектори диму, системи контролю доступу та інші модулі, здатні обмінюватися даними навіть без участі сервера. На практиці це дозволяє реалізувати безліч різноманітних сценаріїв. Так, наприклад, мережа з декількох камер зможе прицільно відстежувати підозрілу машину, передаючи естафету від однієї точки спостереження до іншої і формуючи, таким чином, цілісну картину того, що відбувається. Інший варіант розвитку подій: якщо одна з камер виявиться виведеної з ладу зловмисником, сусідні автоматично візьмуть під нагляд сліпу зону, при цьому на пульт оператора надійде тривожний сигнал.

Найчастіше подібні системи допомагають навіть попередити правопорушення. Для прикладу візьмемо просту чотирьохкомпонентну схему, яка функціонує без участі сервера. Для охорони периметра використовується тепловізор. При виникненні позаштатної ситуації (проникнення зловмисника на об'єкт в нічний час) тепловізор відправляє сигнал на PTZ-камеру, яка розгортається в зазначеному напрямку. Одночасно з цим включається зовнішнє підсвічування і активується гучномовець, що транслює попередньо записане повідомлення з вимогою негайно покинути територію, що охороняється. Згідно з

даними Alarm.org, такий підхід дозволяє запобігти аж до 74% незаконних вторгнень.

Альтернативні сценарії використання сучасних систем відеоспостереження

Поява додаткового функціоналу і поліпшення технічних характеристик привели до значного розширення сфери застосування систем відеоспостереження. Одним з найбільш перспективних напрямків, як би дивно це не звучало, виявилася роздрібна торгівля. Аналітики Transparency Market Research вважають ритейл одним з основних драйверів подальшого розвитку індустрії, так як тут системи відеоаналітики дозволяють вирішувати різноманітні маркетингові задачі.

Типовий сценарій - визначення коефіцієнта конверсії на основі відомостей про кількість відвідувачів і кількості клієнтів, які пройшли через касові стійки. Якщо додати сюди систему розпізнавання осіб, інтегрувавши її до чинної програми лояльності, з'являється можливість дослідження поведінки покупців з прив'язкою до соціально-демографічних факторів для подальшого формування персоналізованих пропозицій (індивідуальних знижок, бандлів за вигідною ціною і т. Д.). Простіші варіанти використання виявляються не менш ефективними, допомагаючи заощадити кошти на персоналі і поліпшити враження покупців. Так, наприклад, китайська мегакорпорація Baidu успішно відмовилася не тільки від продажу квитків в належні їй парки розваг, а й від самих білетерів. Відтепер на КПП встановлені термінали розпізнавання осіб,

У Baidu замінили білетерів на стійки з планшетами

Вся процедура займає не більше 0,6 секунди, а точність розпізнавання досягає 99,77%! Користувачі Alipay і зовсім отримали можливість оплачувати онлайн-покупки за допомогою Селфі, а все завдяки інтеграції платіжного сервісу з платформою візуальної ідентифікації Face ++, створеної стартапом Megvii. В даний час її також успішно використовує китайське підрозділ UBER.

Високі технології відкривають і нові горизонти в сфері боротьби з дрібними розкраданнями в супермаркетах. Протистояти цьому явищу вкрай складно, так як професійні злодії діють в складі організованих груп, учасники яких виконують особливі ролі (власне крадіжка, відволікання уваги охоронців, прикриття і т. Д.). Система ж розпізнавання осіб може автоматично встановити факт того, що одні й ті ж люди регулярно відвідують магазин в один і той же час, навіть якщо заходять порізно, і передати інформацію про їхню присутність на пост охорони. Якщо підозри підтвердяться і це дійсно виявляться злодії, їх особи будуть занесені в базу даних ритейлера і при появі зловмисників в будь-якій філії роздрібної мережі буде спрацьовувати тривожний сигнал.

#### Стримуючі фактори розвитку систем відеоспостереження

На папері все виглядає добре, але як йдуть справи з реалізацією? Якщо говорити про систему розпізнавання осіб на зразок тієї, що описана вище, то її впровадження може призвести до істотними витратами. Одна справа - сканування особи анфас на КПП, і зовсім інша - в потоці, під різними кутами і в різних умовах освітлення. Тут достатню ефективність здатне продемонструвати лише тривимірне моделювання осіб в реальному часі з використанням стереокамер і алгоритмів згортальних нейронних мереж, що неминуче вимагатиме модернізувати існуючу інфраструктуру, а також збільшити місткість сховищ даних. Ось тільки багато представників бізнесу не готові йти на додаткові витрати, навіть коли мова заходить про класичних системах безпеки.

За словами старшого експерта відеотехнічних відділу Експертно-криміналістичного центру МВС Російської Федерації Дениса Михайлова, переважна більшість відеозаписів виявляється непридатним для ведення слідчих заходів саме через прагнення підприємців заощадити на зберіганні даних. Типова ситуація - створення двокомпонентної системи реєстрації: на основному сервері зберігаються записи за останній тиждень у високій роздільній здатності з частотою 25 кадрів в секунду, а в архів потрапляє лише перетиснутій картинка з фреймрейтом 5 кадрів в секунду. На жаль, подібне відео ніяк не може допомогти



провести ідентифікацію особи зловмисника, а в ряді випадків не вдається навіть розпізнати автомобільні номери. Іноді доходить до смішного: потерпілі віддають на експертизу відеореєстратори в зборі,

Деякі підприємці йдуть на відверті махінації. У договорі вказується факт установки PTZ-відеокамер високої чіткості, а насправді монтуються дешеві пристрої, які пишуть картинку у форматі 800x600 і не підтримують дистанційне керування. При сприятливому збігу обставин такі системи можуть працювати роками, а власники дізнаються про їхню повну неспроможність лише в разі форс-мажору, коли застаріла інфраструктура не зможе впоратися з поставленими завданнями.

Втім, навіть якщо відволіктися від свідомих зловживань і скупості, суміжні галузі потерпають від цілком об'єктивних проблем. Одним з перспективних напрямків розвитку відеоаналітики є використання машинного зору в промисловому виробництві.

#### Застосування машинного зору в промисловості

Технологія може застосовуватися для виявлення дефектів продукції, виміру габаритних параметрів вантажів, моніторингу допуску співробітників і для інших цілей. У перспективі це дозволило б поставити на потік створення повністю автоматизованих виробничих комплексів, здатних функціонувати під наглядом декількох операторів. Однак, за даними все того ж Transparency Market Research, темпи зростання даного сегмента ринку виявляються значно скромнішими: прогноз на 2022 рік - 28,5 мільярда доларів (проти 71 мільярда в сфері безпеки). Якщо згадати, що в 2014 році ця цифра становила 15,7 мільярда, на виході отримуємо приріст менш ніж в 2 рази за вісім років.

Стримуючих факторів рівно три. Перший - нестача фахівців, що володіють необхідним набором навичок для управління подібними системами. Другий - високі вимоги до якості і захищеності обладнання. Третій - відсутність єдиних

галузевих стандартів. Коли мова йде про безпеку, одним з важливих переваг систем відеоспостереження стає універсальність. Яким би не був об'єкт, що охороняється (офіс, склад, супермаркет, готель), пул вирішуваних завдань виявляється ідентичний, тобто один раз розроблене апаратне забезпечення може застосовуватися будь-де, додатковий же функціонал можна реалізувати на рівні ПО. У той же час кожне промислове підприємство характеризується унікальністю виробничих процесів, і завдання автоматизації вже неможливо вирішити стандартними засобами - для кожної з них потрібно унікальна розробка.

### 2.3. Висновки до розділу

Дійсно, навіть найбільш складні з описаних вище сценаріїв для потреб ритейлу можуть бути реалізовані на базі того ж самого обладнання, яке використовується в системах безпеки, адже його можливості залежать в першу чергу від потенціалу алгоритмів відеоаналітики і їх грамотного використання. До інфраструктурі ж, створюваної на виробництві, пред'являється маса додаткових вимог - починаючи від стійкості її компонентів до впливу агресивних факторів зовнішнього середовища (актуально для важкої промисловості) і закінчуючи створенням унікальних пристроїв контролю якості мікросхем, зварних швів або рухових блоків. Хоча в розробці останніх немає нічого неможливого, фінансувати її буде вже сам клієнт, так як внаслідок специфічності периферії її навряд чи вдасться використовувати де-небудь ще.

## РОЗДІЛ 3

### ОПИСАННЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ІНТЕГРАЦІЇ СИСТЕМИ КОНТРОЛЮ УПРАВЛІННЯ ДОСТУПОМ З СИСТЕМОЮ ВІДЕОСПОСТЕРЕЖЕННЯ

#### 3.1. Описання існуючої СКУД

У стандартному комплекті поставляється безкоштовне програмне забезпечення *PERCo – SL 01* для організації набору функцій системи контролю доступу (рис. 3.1).

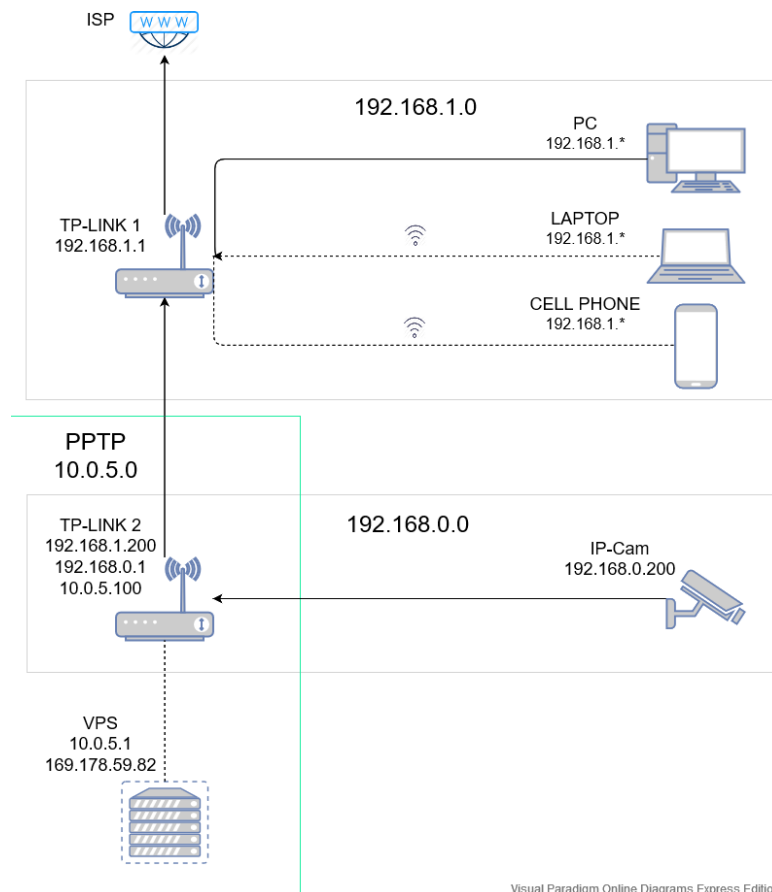


Рис. 3.1. Розташування пристроїв спостереження в системі охорони  
приватного будинку

Кафедра КСУ				НАУ 21 09 63 000 ПЗ			
Виконав	Мізинець О.С.			Описання програмної реалізації інтеграції системи контролю управління доступом з системою відеоспостереження	Літера	Аркуш	Аркушів
Керівник	Нечипорук О.П.				Д	35	59
Консульт.					123 СП 436Б		
Норм. контр.	Тупота Є.В.						
Зав. Каф.	Литвиненко О.Є.						

## Хмарне відеоспостереження з веб-камери і роутера

Якщо ви хочете встановити у себе вдома систему відеоспостереження, але вивчивши ціни на камери і відеореєстратори, така ідея пропадає, то давайте розглянемо бюджетний варіант з використанням веб-камери і роутера.

Нам знадобляться:

роутер TP-Link TL-MR3020

веб-камера (з підтримкою MJPEG)

USB-хаб

USB-флешка

Веб-камера і USB-флешка підключаються до USB-порту роутера через USB-хаб. На роутер встановлюється прошивка WebGlazok. Прошивка створена на базі OpenWrt і використовує програму детектор руху Motion.

Отримуємо просту систему відеоспостереження з функцією виявленням руху та хмарного зберігання, яка не вимагає включеного комп'ютера, відеореєстратора і білого IP-адреси.

Доступні функції:

виявлення руху в заданих зонах із записом 2 кадри в секунду

збереження кадрів по заданому інтервалу часу

зберігання архіву на USB-флешці

запис архіву в хмару (опціонально)

Звичайно, 2 кадри в секунду - це скоріше не відеоспостереження, а фотоспостереження, але все ж для багатьох завдань цього буває достатньо.

Для чого можна використовувати таку систему:

Спостереження в квартирі, будинку, на дачі

Догляд за дітьми

Догляд за тваринами

Контроль над співробітниками в офісі

Візуальний контроль над обладнанням (опалювальними приладами, газовим обладнанням та ін.)

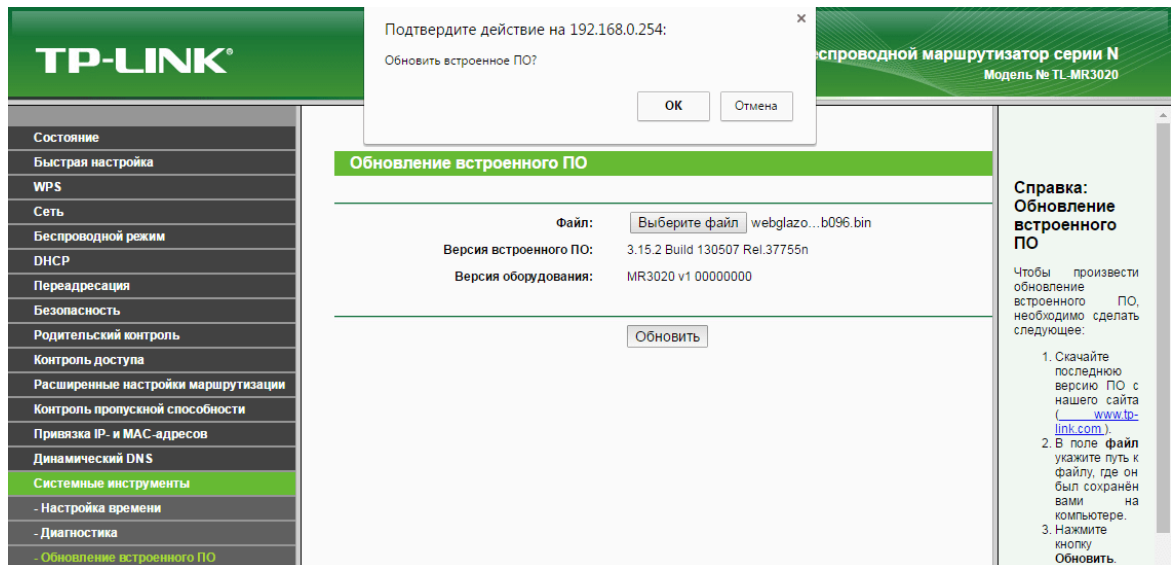
Як зібрати і набудувати систему

Викачуємо останню версію прошивки WebGlazok з офіційного сайту

Заходимо в панель управління роутера за адресою 192.168.0.254

Виконуємо оновлення вбудованого ПО роутера, вказавши завантажену прошивку:

### Оновлення ПО роутера



Пересуваємо перемикач на роутері в положення AP, роутер створить Wi-Fi мережу з ім'ям WebGlazok.com\_XXX

### Перемикач на роутері

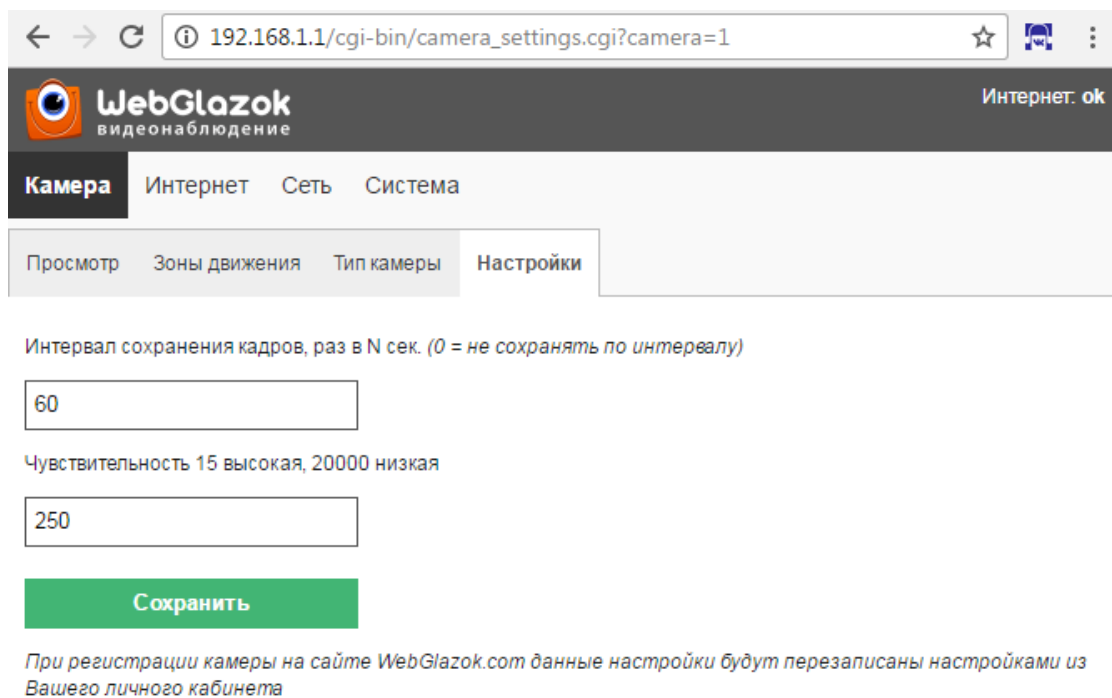


Підключаємося до цієї мережі, пароль - MAC-адресу роутера (надрукований на корпусі)

Заходимо в панель управління роутера за адресою 192.168.1.1, Логін admin, пароль admin.

В панелі управління задаємо чутливість виявлення рухів, зони рухів і інтервал отримання кадрів:

Панель управління роутером з прошивкою WebGlazok



Багато вказати Wi-Fi мережу для підключення до інтернет, воно необхідне для синхронізації часу і відправки знімків в хмару

#### Як працює система

Роутер аналізує зображення з камери і при виявленні рухів зберігає кадри на флешку зі швидкістю 1-2 кадри в секунду. При відсутності рухів кадри зберігаються по заданому інтервалу часу.

На флешці зберігаються JPEG кадри в папках з наступною структурою /cam1/дата/часимінути/файл.jpg

Наприклад: /cam1/20170403/1540/20170403154400-00\_s4\_s.o.jpg

Зображення зберігаються в 2-х копіях - превью і повнорозмірне фото. Переглянути записані кадри можна через комп'ютер, вставивши в нього флешку.

З мого досвіду використання флешки на 16 Гб вистачає приблизно на 2 тижні зберігання архіву при наявності рухів приблизно в 5% випадках від усього часу. Після заповнення пам'яті старі кадри видаляються з флешки.

Система може працювати повністю автономно і без підключення до інтернет.

Однак дана модель роутера не має вбудованого годинника, тому час на кадрах виставляється неправильно. При відключенні харчування час «зупиняється».

Для вирішення цієї проблеми роутер необхідно підключити до мережі інтернет через Wi-Fi. Тоді час синхронізується через інтернет. Трафік при цьому практично не споживається. Підійде навіть слабкий 2G інтернет. За містом інтернет можна роздати через 3G / 4G роутер або звичайний роутер з підключеним 3G / 4G модемом.

#### Автономна робота

Роутер працює від USB роз'єму живлення, тому при відсутності розетки живлення 220 В його можна живити, наприклад, від мобільного акумулятора (павербанка).

Напруга живлення роутера 5 В, виміряна сила струму при роботі камери - 0.6 А, потужність 3 Вт. Таким чином USB павербанка ємністю 10000 mAh (36 Вт / год) вистачить на 12 годин роботи системи.

Також можна підключити автомобільний акумулятор з перетворювачем з 12 В на 5 В.

#### Якщо потрібен онлайн перегляд

Дивитися кадри на флешці не дуже зручно. Щоб знайти цікавить подія потрібно перегорнути на комп'ютері всі файли.

Роутер з прошивкою WebGlazok має можливість зберігати кадри в хмару і там склеювати їх в відеоряд. І ви зможете переглядати камеру через браузер або мобільний додаток. Однак ця послуга платна і коштує 199 руб в місяць (на квітень 2017).

Для зберігання архіву вашої камери в хмарі необхідно підключити роутер до мережі інтернет через Wi-Fi, а також зареєструвати камеру на сайті WebGlazok.com і отримати ключ пристрою.

В особистому кабінеті відображається життєдіяльності камери і картинка з камери. Виявлені руху відзначаються червоним кольором на стрічці.

З особистого кабінету також проводиться настройка камери:

- чутливість датчика руху
- зони виявлення рухів
- інтервал збереження кадрів
- Налаштування повідомлень

На пошту приходять повідомлення про рухи з доданими кадрами, оповіщення про відключення камери від мережі і про втрату сигналу.

Роутер відправляє в хмару кадри з низьким дозволом (превью), які займають по 5-20 Кб. Тому трафіку витрачається мало. Такого дозволу досить, щоб подивитися обстановку, але для детального перегляду потрібен кадр з повним дозволом.

кадр превью

В особистому кабінеті (в браузері або мобільному додатку) під час зупинки на кадрі передпросмотра більше 2-х секунд на роутер відправляється запит для отримання повнорозмірного кадра з високою роздільною здатністю. Роутер шукає цей кадр на USB-флешці і відправляє його в хмару. Процес займає 10-30 секунд в залежності від швидкості інтернет, після чого превью кадр в особистому кабінеті змінюється на повнорозмірний (HD). При повторному запиті, HD кадр вже відкривається швидко, так як він вже збережений в хмарі.

HD кадр

Таблиця співвідношень дозволу кадру і кадру передпросмотра:

Дозвіл камери / Дозвіл передпросмотра 640x480 / 320x240 1280x720 / 320x180 1280x960 / 320x240 1920x1080 / 240x135 1600x1200 / 200x150

Підключення IP-камер

Веб-камера недорога і компактна, легка в підключенні і установці. Однак якщо вам потрібна нічна зйомка і монтаж на вулиці, то без IP-камери не обійтись.

Система WebGlazok підтримує також підключення IP-камер, які підключаються до роутера через LAN порт або по Wi-Fi.

Від IP-камери потрібно тільки картинка в форматі JPEG або потік MJPEG. Сучасні IP-камери, які підтримують протокол ONVIF, вміють віддавати snapshot.

Вся аналітика розпізнавання руху також відбувається на роутері. Можна купити дешеву китайську камеру на Aliexpress за 1300-1800 руб.

Кінцева вартість



Отже, у скільки ж обійдеться комплект такого відеоспостереження? Як веб-камери візьмемо популярну модель Logitech C270 з роздільною здатністю 1280x720.

Ціни на квітень 2021 року:

роутер TP-Link TL-MR3020 - 600 грн.

веб-камера Logitech C270 HD Webcam - 700 грн.

USB-хаб - 150 грн.

USB-флешка 16 Гб - 200 грн.

### 3.2. Архітектура програмного забезпечення інтеграції з СКУД

Сервіс хмарного відеоспостереження, наприклад, iVideon, стали і посправжньому зручними для споживача. З урахуванням невисокої абонентської плати сервіс повинен був давно стати масовим і знайти користувачів в кожному під'їзді і заміському будинку. Що стримує його розвиток?

Можливість простий трансляції відео часто виявляється недостатньою для платного використання сервісу. Восторг від HD-відео в браузері і на телефоні швидко проходить, і користувач задається питанням, навіщо йому потрібна жива трансляція або місячний архів відео. Дивитися нецікаво і колись. 99% архіву не потрібно, однак передачу і зберігання даних при цьому доводиться оплачувати.

Говорячи про головну функцію моніторингу будь-якого житлового об'єкта, користувачеві не вистачає подієвого моніторингу, коли система за допомогою засобів відеоаналітики, аудіоаналітики і людини-оператора чітко обробляє за регламентом всі підозрілі ситуації, наприклад, крик в під'їзді, люди вночі на сходах, дим зі смітєвого контейнера, праздношатаніє на парковці.

У приватному секторі (в заміському будинку), господареві важливо якомога швидше отримати повідомлення про можливий інцидент, наприклад, якщо на території ділянки з'явилися сторонні люди.

У доступних системах відеоспостереження через Інтернет користувачеві не вистачає засобів пошуку за змістом в архіві. Як відразу побачити всі події за

день без стомлюючого перегляду і перемотування відео? Повернувся дитина зі школи додому? Що робив садівник на дачі? Хто подряпав машину на парковці?

Деякі перераховані функції пошуку сьогодні доступні в системах управління відео (VMS), наприклад, компаній ITV і Milestone, але їх застосування обмежено професійними користувачами служби охорони. Їх архітектура, веб-інтерфейс і мобільні додатки не розкривають потенціал відеоаналітики для масового споживача - жителів квартир або заміських будинків.

Від відеоспостереження до відеоаналітика

З цим думками ми почали опрацювання двох рішень для "інтелектуалізації" відеоспостереження в багатоквартирних будинках і дооттеджних селищах.



Віддалений відеомоніторинг будинку на базі хмарного сервісу Kipod розподілена архітектура

Як показано на малюнку, рішення побудовано на базі веб-сервісу віддаленого моніторингу Kipod. Пристрій запису і аналітики (Kipod Server) Розміщується безпосередньо на об'єкті, в багатоквартирному або приватному будинку. До пристрою підключаються мережеві камери і, опціонально, робоча станція консьєржа / охоронця і гучномовець. Аналогові камери домофона і ліфта можна підключити через перетворити (IP-кодер).

Важливою перевагою розподіленого рішення є мінімальне навантаження на канал зв'язку і можливість автономної роботи кожного об'єкта в ситуації, коли канал зв'язку перевантажений або недоступна. Після відновлення каналу, події автоматично синхронізуються з хмарним сервісом. Таким чином, рішення з розподіленою архітектурою буде працювати з десятками камер навіть на бездротовому каналі 3G.

#### централізована архітектура

Альтернативний підхід - повністю централізована архітектура, коли вся обробка і зберігання реалізовані в датацентрі провайдера послуги на базі кластерного серверавідеоаналітики. Переваги централізованого підходу - простота впровадження і зниження вартості обслуговування устаткування. Недоліки - більш високі вимоги до пропускну здатності каналів і, як наслідок, обмеження за кількістю камер і якістю запису відео.

#### Відеопрезентація рішення

Ми підготували кілька відеороликів, які ілюструють роботу системи.

#### 1. Веб-інтерфейс користувача

Дивіться відео HD через браузер

Знаходьте події у фотоальбомі

Надішліть посилання сусідам

#### 2. Проста настройка аудіо і відеодетектора:

охорони периметра

Шуму, крику і звуку розбитого скла

Людини і машини

Вогню і диму

поломки камери

#### 3. Мобільний додаток

Постановка на охорону і зняття

Відео та кадри подій

PUSH-повідомлення про інцидент з відео

#### 4. Віртуальний консьєрж

Завдання віртуального консьєржа - цілодобова обробка тривожних подій, що виникають в результаті роботи відеоаналітики і аудіоаналітики Kipod. Оператори віртуального консьєржа розміщуються в диспетчерському центрі Сінезіс або працюють віддалено. Якщо оператор підтверджує кримінальне дію або порушення порядку, то далі він виробляє ряд заданих дій по регламенту (викликає поліцію, зв'язується з мешканцями, передає голосове повідомлення по гучномовному зв'язку). Послугу віртуального консьєржа ми плануємо запуснути пізніше, за додаткову плату.

Для жителів замських будинків передбачена зручна функція для постановки об'єкта на охорону і зняття з охорони через мобільний додаток Kipod Mobile (Android, iOS). Оператор отримує тривожні події і доступ до камер тільки в разі, якщо об'єкт знаходиться на охороні. PUSH-повідомлення про тривожні ситуаціях приходять жителям на мобільні пристрої.

5. Підключення будь-яким каналом зв'язку

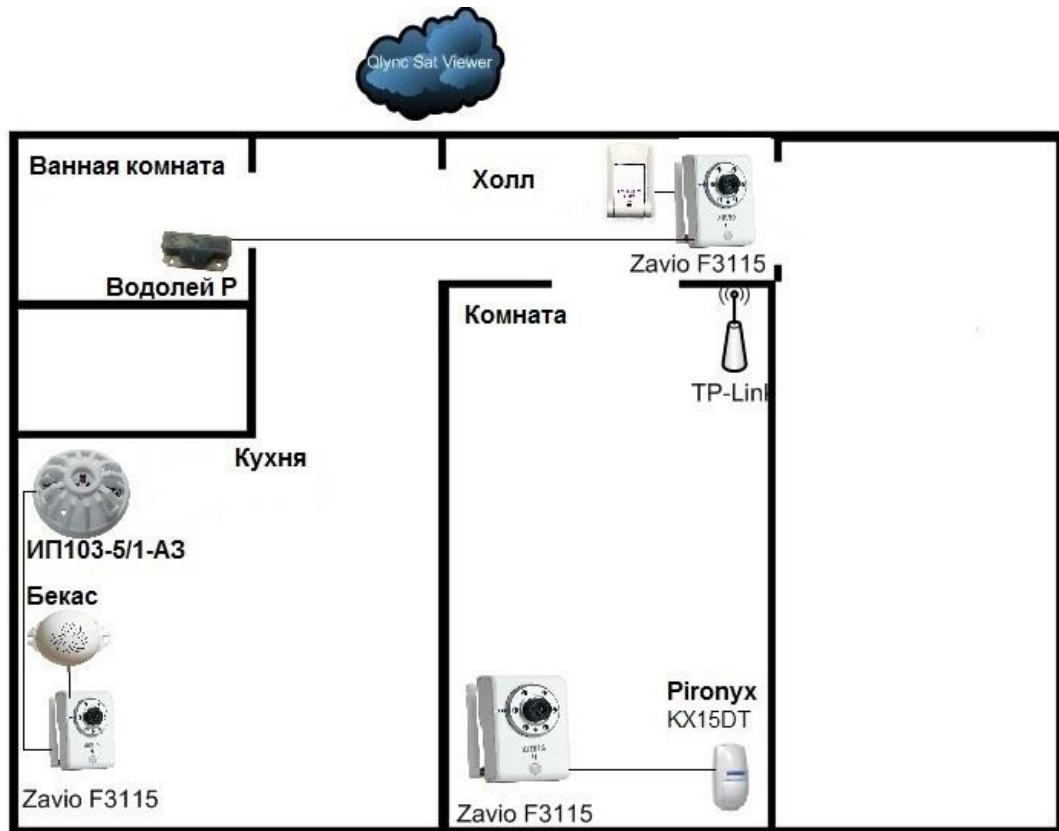
Модем 3G / 4G від 1 Мбіт / с

Подієва суперечачи кадрів і відео

### 3.3. Збереження об'єктів

Розширення функціоналу IP-відеоспостереження для застосування в побуті

У цій статті я хочу розповісти про те, як можна вичавити максимум з IP-камери, організувавши на її базі не просто відеоспостереження, а вже щось, наближене до пожежно-охоронного комплексу, по засобом підключення різних датчиків (PIR, тепла, води і т . Д.). Мова піде про підключення, налаштування ір-камер і датчиків, про те, наскільки це може бути корисно в забезпеченні безпеки побуту.



Зовнішній вигляд, характеристики, призначення

Почнемо з опису кожного з пристроїв.

Zavio F3115

Основою виступає Zavio F3115 - це бездротова IP-камера Тайванської походження, в моєму запасі 3 шт.





### Технічні характеристики

Особливості Zavio полягають в Plug & Play і безкоштовному софті:

хмарний сервіс Qlync Sat Viewer

SNVR CamGrab для ПК.

Детальну інформацію про камеру можна подивитися, перейшовши по цьому посиланню.

Далі у нас підуть датчики. Датчики поділяються на 2 основні типи: Нормально Замкнуті (НЗ) і Нормально Розімкнені (НР) - дуже важливо це знати для правильного підключення датчиків.

### Pironyx KX15DT

Це - Pir-sensor, в простолюдді називають «Датчик руху». Поєднаний датчик (пасивний ІЧ + мікрохвильовий) KX15DT виробник Pyronix:

Оснащений так званим Tampering Detection, він же «датчик саботажу», «датчик розтину», тобто при розтині самого датчика він відправляє Tampering-сигнал на хост-пристрій (в моєму випадку це Zavio F3115).

Широка область застосування - від звичайних сигналізують сповіщень, до управління вмикання ламп.

**УВАГА, ЗАХОДИ УНИКНЕННЯ помилкових спрацьовувань:**

Познайомиться більш детально з ним можна перейшовши по цьому посиланню.

Спосіб застосування та принцип роботи дуже прості, його я встановив у ванній кімнаті. Призначений він для оповіщення в разі витoku води. «Штука» проста і елементарна, основна електронна частина датчика заізолювана, відкритими залишаються Анод і Катод, які замикаються при взаємодії з водою.

Сповіщувач пожежний тепловий ПП 103-5 / 1-А3



Досить простий датчик, при нагріванні до температури спрацювання сповіщає про «пожежу».

Оповіщувач охоронно-пожежний звуковий Бекас (ПКІ-МБ)

Комутується разом з ПП 103-5 / 1-А3, але в якості виходу, маленька «сирена»:



При спрацюванні теплового датчика, Бекас починає сигналізувати до того моменту, поки не буде усунена «проблема», вважав за потрібне підключити такий вихід, щоб оповістити домашніх в разі пожежі. Дуже гучний оповіщувач,

який здатний розбудити «мертвого». Пройшли тисячі років, а найнебезпечнішим стихійним ворогом людства є Вогонь, вважаю, що дана зв'язка повинна бути у кожного в будинку. Я вже мав такий інцидент і тому вирішив убезпечити свій будинок таким способом.

Так само в «системі» використовується Emergency Alarm ( «Кнопка Паніки»). Являє собою «ключ», що спрацьовує на розмикання ланцюга. На жаль, інформації в інтернеті про нього немає. Дана «кнопка паніки» потрібна для передачі тривожного сигналу на камеру, дуже корисна Кнопка для установки в квартирі літніх людей, наприклад.



Установка в тестовому режимі

У планах ремонт квартири, але хочеться бути у всеозброєнні, перейдемо на Кухню.

кухня

Сильно з монтажу не возився, тому що до кожного датчика є інструкція по підключенню. Для підключення датчиків до камери досить було «розкрити» виту пару і вилучити звідти одну пару проводів і підключити її в клемнічек камери, схема підключення датчиків так само додається в User Manual. 1-pin - Харчування 5В, 2-pin - GND, 3-pin Digital Input (Вхід), 4-pin - Digital Output (Вихід).



На кухні я встановив датчик Тепла і оповещатель Бекас, мета: при спрацьовуванні датчика тепла, потрібно задіяти оповещатель Бекас. Для цього я підключив до входу камери ІП 103-5 / 1-А3, а на вихід оповіщувач. Харчування підключив як, ви вже напевно здогадалися - до 1-му піну, щоб замкнути ланцюг - підключив оповещатель і датчик до GND.

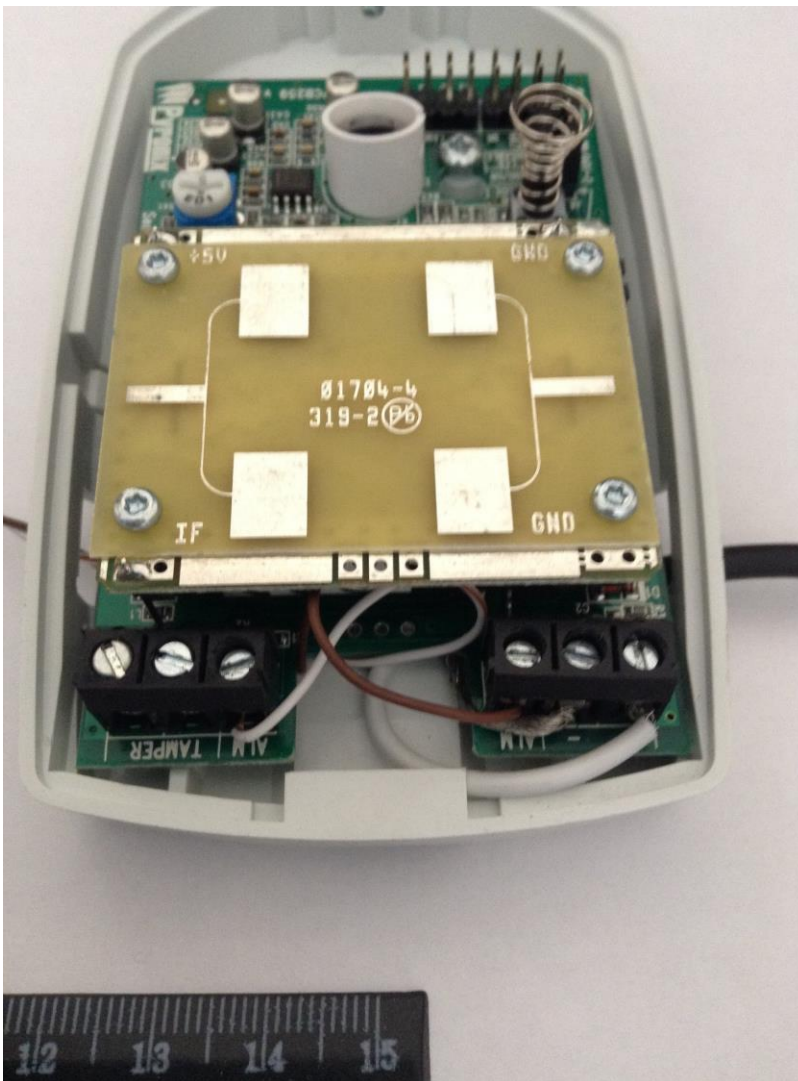
Розташування: Діоди на камері можна відключити, щоб не привертати уваги. "Посміхаємося і махаємо".

Датчик тепла правильніше ставити в центрі приміщення на стелі, але поки що нехай висить над раковиною.

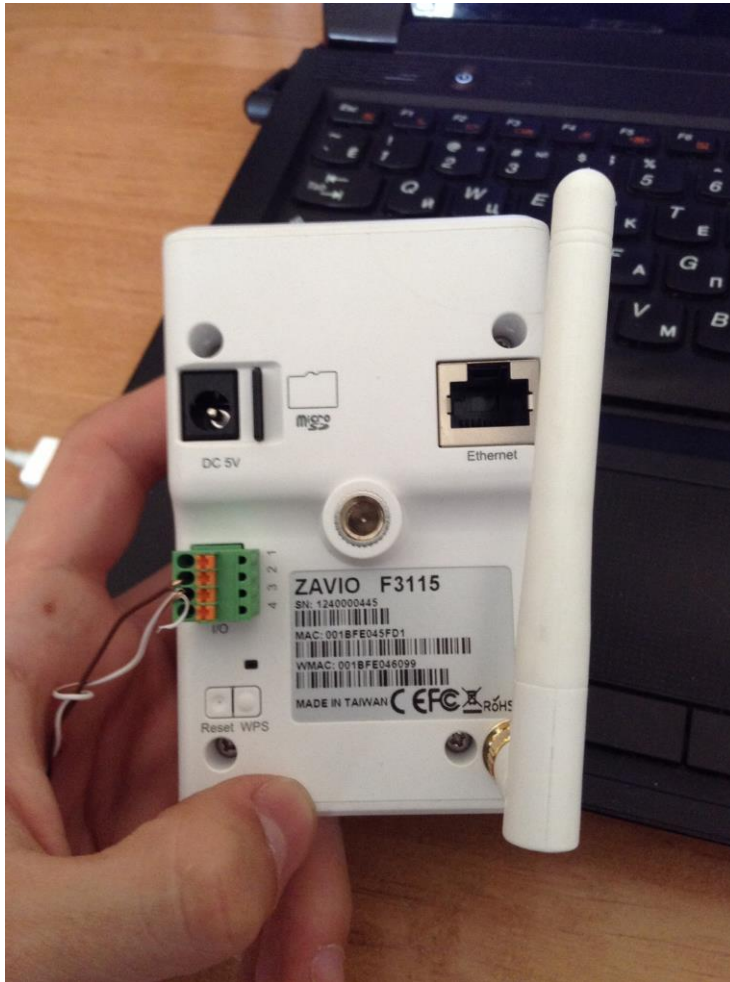


кімната

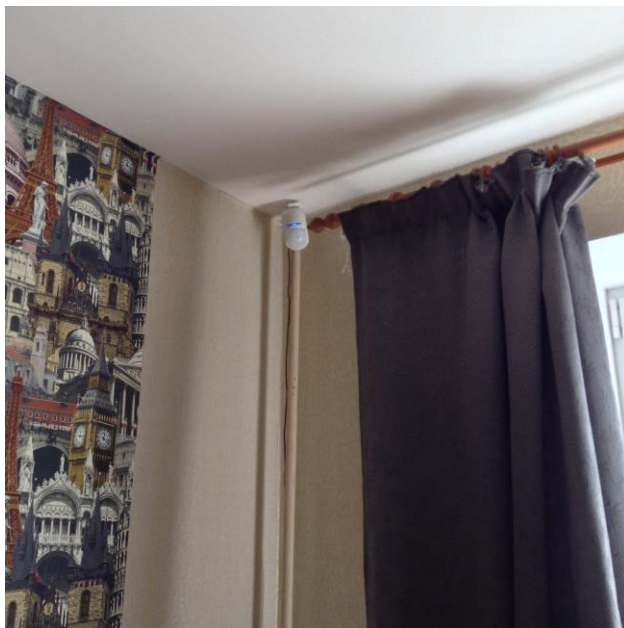
У своїй кімнаті я встановив Руніх КХ15DT і підключив до такої ж камері Zавіо F3115.



Підключаємо до камери:



На жаль, через мою неуважність Ругоніх вимагає 12V, а камера віддає всього 5V і мені довелося підключити до нього окремий БП (Mean Well), благо в моїх запасах їх виявилось кілька штук.





Все працює, подивимося відео.

Спрацювання датчика:

Push-повідомлення:

ХоллЗдесь я встановив кнопку «Паніки» і датчик «затоплення», що розташовується у ванній кімнаті. Підключення теж дуже просте, але в цьому випадку я маю 2 датчика на ВХІД, комутуючі обидва датчика до 3-піню камери.

Налаштування

Почнемо з настройки DI / DO, в веб-інтерфейсі потрібно створити Подія, робиться це дуже легко і просто, труднощів не виникало, так само є локалізація для RU користувачів в веб-інтерфейсі камери. Передбачена настройка розкладів для «Події», наприклад датчик Ругоніх працює виключно в той час, коли вдома нікого немає, крім кота (датчик має особливість в чутливості не спрацьовує на вихованців, але при всьому при цьому датчик спрацює на рух дитини).

Вікно додавання Подій:



## Налаштовуємо Подія:

**Основные**

Название

Событие  Вкл  Выкл

---

**Событие**

Тип события

Интервал между событиями  Сек

Цифровой вход1

---

**Действие**

Отправка кадра

Сервер событий

Название	Тип	Описание
<input checked="" type="checkbox"/> SD Card	SD	Видео, Пред:7, Пост:7

Настройте функцию [Сервер событий](#) или [Карта памяти](#)

Отправка уведомления

Задействовать цифровой выход

PTZ

Ночной режим

Аудиоклип

---

**Расписание**

Всегда

Расписание

Настройте функцию [Расписание](#)

## Підключення хмарного сервісу і настройка push-повідомлень

Тут все дуже просто, я поставив на свій iPad додаток (так само є клієнт під Android, а для користувачів ПК досить перейти і залягання на [sat.qlync.com](http://sat.qlync.com)) .Лічно у мене труднощів установка не викликала, але є інструкція в базі знань

Zavio для користувачів, у яких можуть виникнути проблеми. Сам хмарний сервіс вміє переглядати Архів з SD-носія, а так же відправляти Event ролики в Google Drive, тобто після заповнення картки в 16Гб я буду мати архів (безкоштовно до 15Гб) на своєму Гугл-диску.

Qlunc вміє відправляти push щодо подій, в моєму випадку події - спрацювання датчиків. Ось так вони виглядають:

При свайпа Push, перенаправляє на LiveView:

Ви можете переглянути список подій на Камері, імена роликів містять дату, тобто дуже зручно знайти подія в потрібний час:

На рис. 3.2 представлена діаграма класів, що відображає структуру даного механізму.

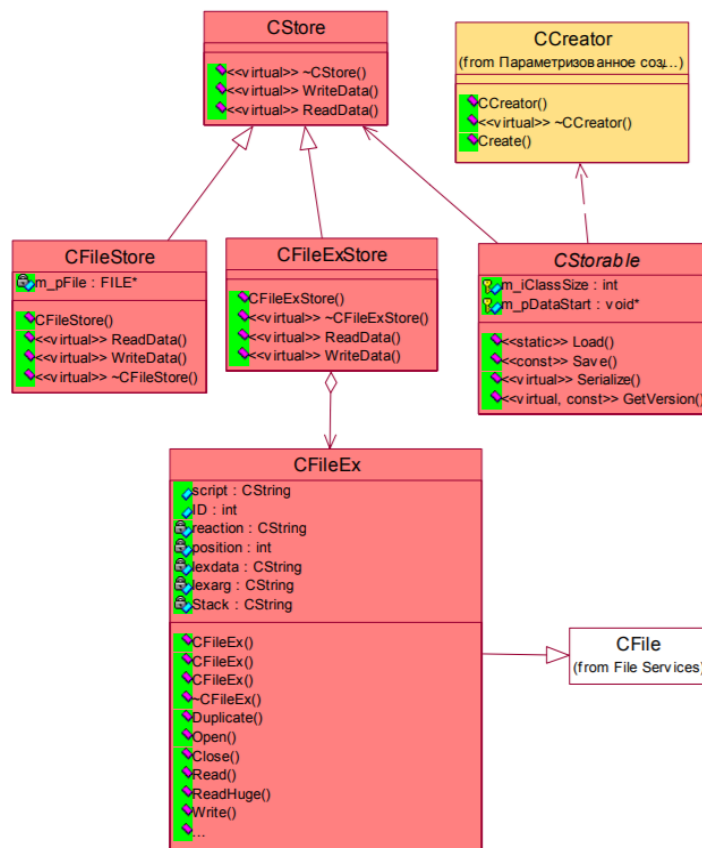


Рис. 3.2. Діаграма класів механізму серіалізації

При додаванні об'єкта, викликається функція перетворення (*Transf*), яка за аналогією з функцією *Serialize* (рис. 3.2).

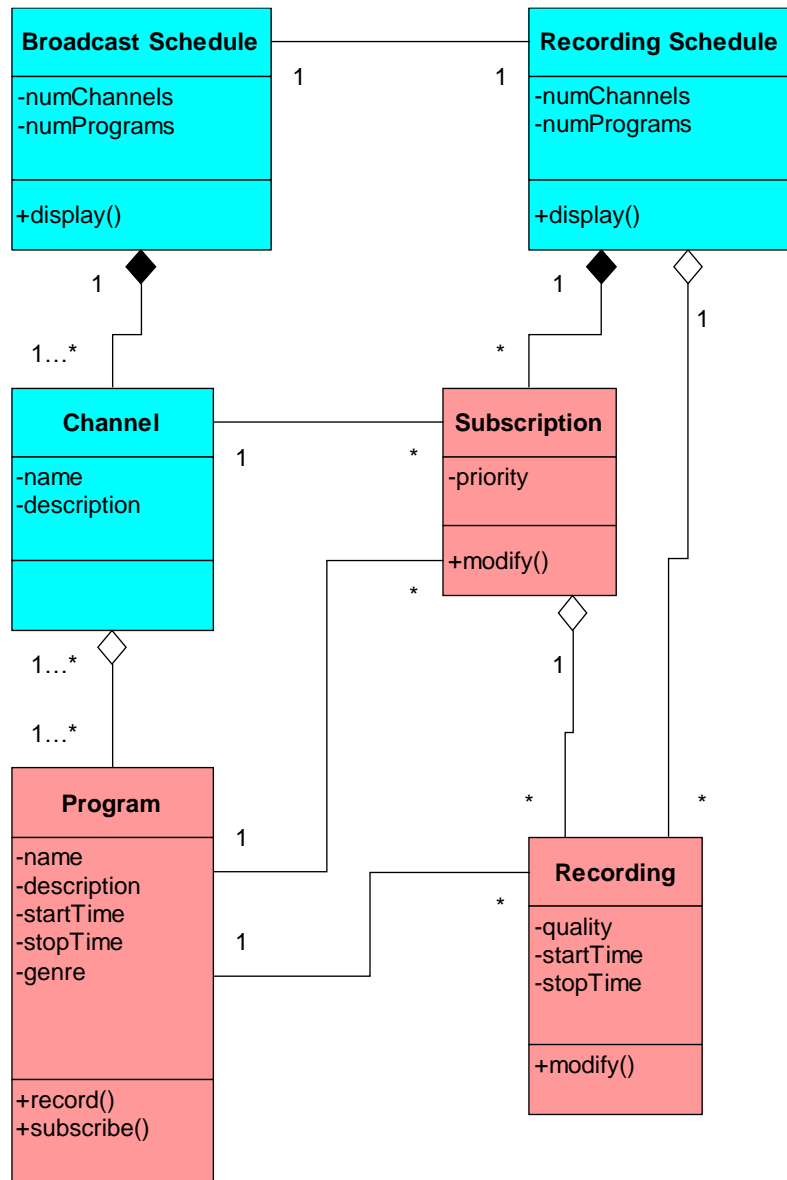


Рис. 3.3. Діаграма класів механізму складування об'єктів

Діаграма класів представлена на рис. 3.5.

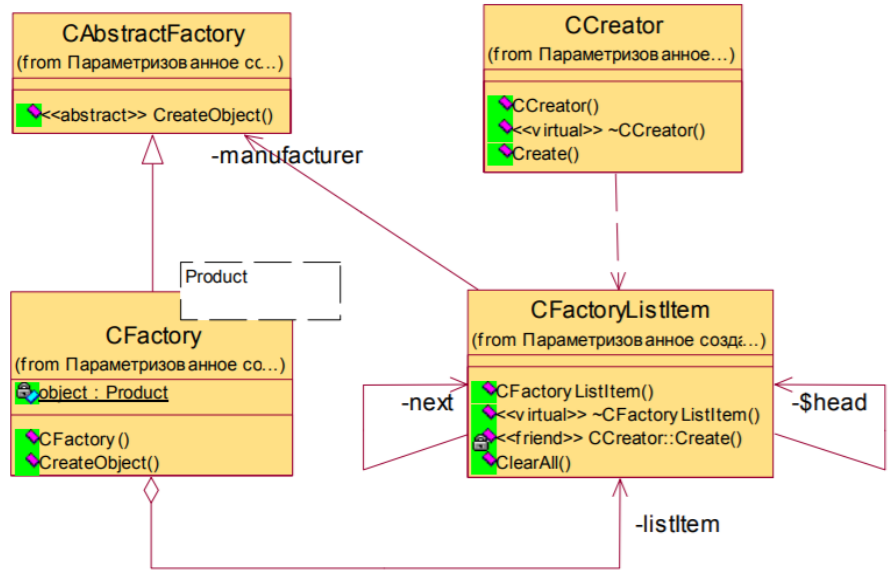


Рис. 3.5. Діаграма класів механізму параметризовані створення об'єктів

### 3.4. Висновки до розділу

Ось таким чином мені вдалося вичавити максимум з домашнього відеоспостереження, надаючи можливість контролювати те, що відбувається в квартирі, за рахунок підключення додаткових «сенсорних органів». Самі датчики можуть бути різними, будь цим «органом» квартирний дзвінок, або ж спеціалізований датчик газу або звукова сирена.



## ВИСНОВКИ

В дипломному проекті було розроблено програмні модулі для інтеграції відеоспостереження в систему охорони приватного будинку.

Найскладніше була оптимізація алгоритмів, які не придумування гарної системи, а боротьба з load average. Уявіть, у вас є 30 потоків камер, які пишуть на диск, диски десктопні в рейді на FreeNas через NFS, навіть якщо б без NFS, то все одно просадка системи колосальна.

Відеоспостереження - не новинка, а частина повсякденного життя. Його використовують і поліцейські, і домогосподарки. У домашніх умовах хмарне відеоспостереження, тобто таке, яке не вимагає окремих комп'ютерів, серверів і іншої апаратури, крім камер, допомагає контролювати роботу прибиральниць і нянечок, стежити за тим, чи вчасно вийшов дитина в школу або ж за улюбленим вихованцем. Розповімо докладніше, як за 15 хвилин створити власну систему відеоспостереження в домашніх умовах.

З одного боку, камера - це той відеодатчик, без якого система відеоспостереження поки немислима. З іншого боку, краще за все використовувати IP-камеру. Це використовує протокол IP сучасний цифровий прилад, особливість якого - передача відеопотоку в цифровому форматі по мережі Ethernet.

Щоб підключити таку камеру, не потрібні додаткові кабелі, перехідники, відеокасети. Оскільки IP-камери використовують цифровий спосіб передачі сигналу, це гарантує якість зображення, проте варто пам'ятати, що у відповідності зі стандартами локальних обчислювальних мереж обмеження в застосуванні IP-камер становить 90 метрів. Це актуально для великих квартир або заміських ділянок, проте, пам'ятати про це необхідно на стадії підготовки.

Отже, крок перший. Переконайтеся, що тримаєте в руках IP-камеру, роздрукуйте коробку і підготуйте прилад до використання. Якщо коробка промаркована такими аббревіатурами як AHD, TVI, SDI або CVI, перед вами аналогова камера, яка зажадає трохи більших витрат при установці.

Крок другий. Підключіть IP-камеру до вашого маршрутизатора, або роутера. Це та сама коробочка, яка роздає в квартирі інтернет. Для підключення вам знадобиться ethernet-кабель, який зазвичай йде в комплекті з роутером. Якщо його немає, можна його придбати в будь-якому магазині або знайти в пакеті з непотрібними проводами. Підключіть камеру до роутера за допомогою ethernet-кабелю.

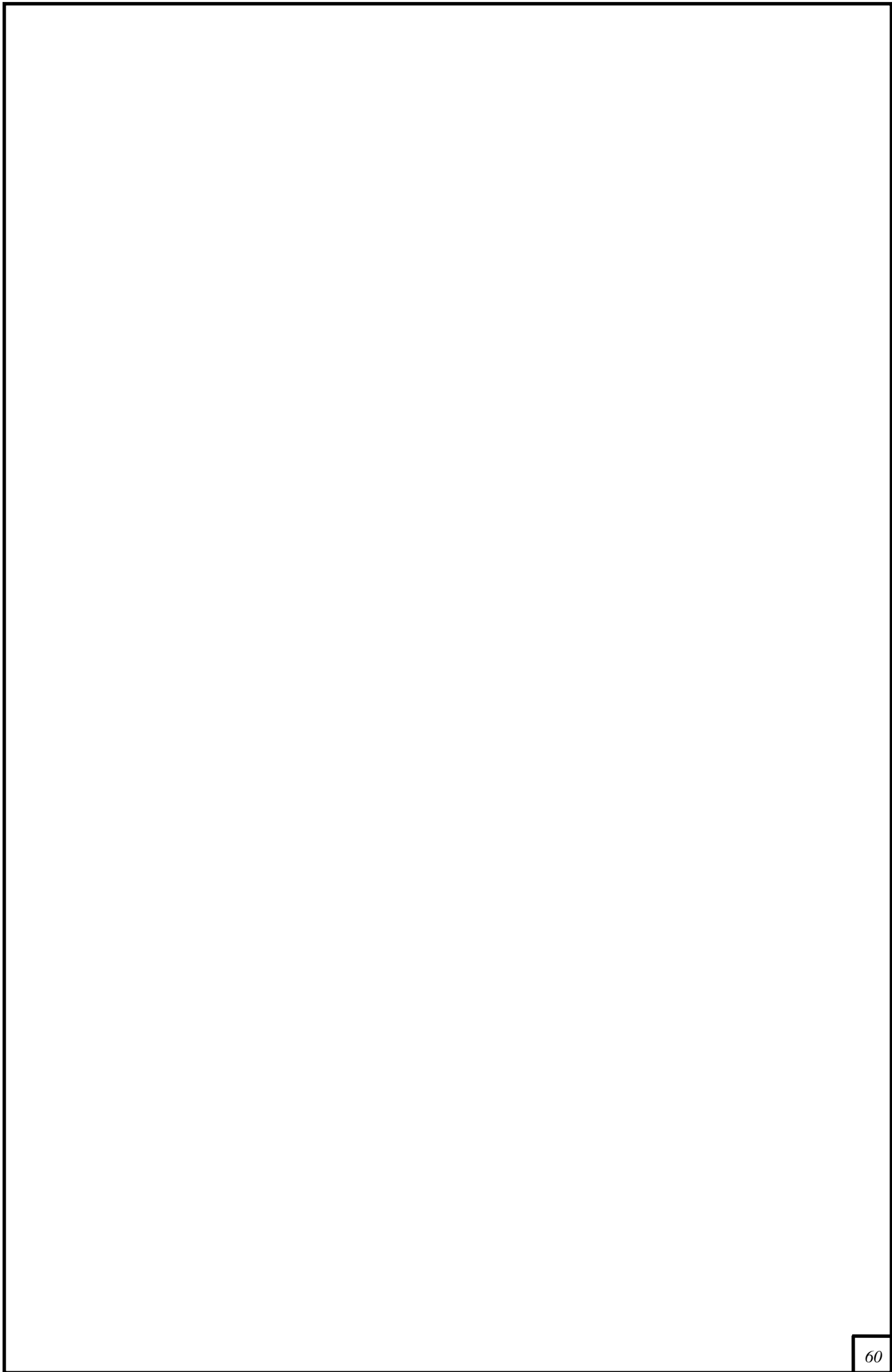
Крок третій. Настав час використовувати ваш домашній комп'ютер. Передбачається, що він вже підключений до домашнього інтернету. Тепер потрібно перевірити, чи працює камера в локальній мережі. З цим все просто.

Відкрийте на комп'ютері ваш браузер, щоб зайти через нього в інтерфейс налаштування роутера. Спробуйте набрати в адресному рядку браузера 192.168.1.1 або 192.168.0.1 - найчастіше саме за цими адресами можна «увійти в роутер». Якщо ви не змінювали пароль, то використовуйте поєднання admin / admin в якості логіна і пароля. Якщо ж при підключенні виникли складності, подивіться на спеціальну наклейку на корпусі роутера - всі дані також вказуються тут.

Після того, як ви перейшли в розділ налаштування роутера, знайдіть в списку підключених пристроїв вашу відеокамеру і з відповідного рядка скопіюйте її локальний IP-адресу. Він може виглядати, наприклад, так: 192.168.1.2-255. Тепер вставте цю адресу в сусідню вкладку браузера. Якщо відкриється сторінка камери, вона успішно підключена до роутера.

Крок четвертий. Тепер потрібно підключити сервіс хмарного відеоспостереження. Це ще простіше, ніж попередній крок.

## СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ



## Додаток А

### Лістинг основного програмного модуля

```
import wx
import socket
import threading
import time
import wx.lib.masked      as masked
import wx.lib.scrolledpanel  as scroll
class FuncThread(threading.Thread):
    def __init__(self, target, *args):
        self._target = target
        self._args = args
        threading.Thread.__init__(self)
    def run(self):
        self._target(*self._args)
def ByteToHex(byteStr):
    return ''.join([ "%02x " % ord(x) for x in byteStr ]).strip()
class TestTool(wx.Frame):
    def __init__(self, *args, **kwds):
        # begin wxGlade: MyFrame.__init__
        kwds["style"] = wx.DEFAULT_FRAME_STYLE
        wx.Frame.__init__(self, *args, **kwds)
        self.SetFont(wx.Font(10, wx.DEFAULT, wx.NORMAL, wx.NORMAL, 0,
""))
        self.l_ip = wx.StaticText(self, -1, "IP")
        self.tc_ip = masked.Ctrl(self, -1, controlType =
masked.controlTypes.IPADDR, style=wx.TE_CENTRE)
        self.l_port = wx.StaticText(self, -1, "Port")
```

```
self.tc_port = masked.NumCtrl (self, -1, integerWidth = 5, groupDigits =
False, style=wx.TE_CENTRE)
self.tc_port.SetBounds(0,65535)
self.button_send = wx.Button(self, -1, u"Начать тестирование")
#self.button_reset = wx.Button(self, -1, u"Сброс")
self.l_speed_header = wx.StaticText(self, -1, u"Скорость подключения")
self.l_open_header = wx.StaticText(self, -1, u"Открытие")
self.l_send_header = wx.StaticText(self, -1, u"Запись")
self.l_send_control_header = wx.StaticText(self, -1, u"Запись[проверка]")
self.l_close_header = wx.StaticText(self, -1, u"Закрытие")
self.l_speed110 = wx.StaticText(self, -1, u"110 бод")
self.l_open_speed110 = wx.StaticText(self, -1, u"...")
self.l_send_speed110 = wx.StaticText(self, -1, u"...")
self.l_send_control_speed110 = wx.StaticText(self, -1, u"...")
self.l_close_speed110 = wx.StaticText(self, -1, u"...")
self.l_speed300 = wx.StaticText(self, -1, u"300 бод")
self.l_open_speed300 = wx.StaticText(self, -1, u"...")
self.l_send_speed300 = wx.StaticText(self, -1, u"...")
self.l_send_control_speed300 = wx.StaticText(self, -1, u"...")
self.l_close_speed300 = wx.StaticText(self, -1, u"...")
self.l_speed600 = wx.StaticText(self, -1, u"600 бод")
self.l_open_speed600 = wx.StaticText(self, -1, u"...")
self.l_send_speed600 = wx.StaticText(self, -1, u"...")
self.l_send_control_speed600 = wx.StaticText(self, -1, u"...")
self.l_close_speed600 = wx.StaticText(self, -1, u"...")
self.l_speed1200 = wx.StaticText(self, -1, u"1200 бод")
self.l_open_speed1200 = wx.StaticText(self, -1, u"...")
self.l_send_speed1200 = wx.StaticText(self, -1, u"...")
self.l_send_control_speed1200 = wx.StaticText(self, -1, u"...")
self.l_close_speed1200 = wx.StaticText(self, -1, u"...")
self.l_speed2400 = wx.StaticText(self, -1, u"2400 бод")
```

```
self.l_open_speed2400 = wx.StaticText(self, -1, u"...")
self.l_send_speed2400 = wx.StaticText(self, -1, u"...")
self.l_send_control_speed2400 = wx.StaticText(self, -1, u"...")
self.l_close_speed2400 = wx.StaticText(self, -1, u"...")
self.l_speed4800 = wx.StaticText(self, -1, u"4800 бод")
self.l_open_speed4800 = wx.StaticText(self, -1, u"...")
self.l_send_speed4800 = wx.StaticText(self, -1, u"...")
self.l_send_control_speed4800 = wx.StaticText(self, -1, u"...")
self.l_close_speed4800 = wx.StaticText(self, -1, u"...")
self.l_speed9600 = wx.StaticText(self, -1, u"9600 бод")
self.l_open_speed9600 = wx.StaticText(self, -1, u"...")
self.l_send_speed9600 = wx.StaticText(self, -1, u"...")
self.l_send_control_speed9600 = wx.StaticText(self, -1, u"...")
self.l_close_speed9600 = wx.StaticText(self, -1, u"...")
self.l_speed19200 = wx.StaticText(self, -1, u"19200 бод")
self.l_open_speed19200 = wx.StaticText(self, -1, u"...")
self.l_send_speed19200 = wx.StaticText(self, -1, u"...")
self.l_send_control_speed19200 = wx.StaticText(self, -1, u"...")
self.l_close_speed19200 = wx.StaticText(self, -1, u"...")
self.l_speed38400 = wx.StaticText(self, -1, u"38400 бод")
self.l_open_speed38400 = wx.StaticText(self, -1, u"...")
self.l_send_speed38400 = wx.StaticText(self, -1, u"...")
self.l_send_control_speed38400 = wx.StaticText(self, -1, u"...")
self.l_close_speed38400 = wx.StaticText(self, -1, u"...")
self.l_speed57600 = wx.StaticText(self, -1, u"57600 бод")
self.l_open_speed57600 = wx.StaticText(self, -1, u"...")
self.l_send_speed57600 = wx.StaticText(self, -1, u"...")
self.l_send_control_speed57600 = wx.StaticText(self, -1, u"...")
self.l_close_speed57600 = wx.StaticText(self, -1, u"...")
self.l_speed115200 = wx.StaticText(self, -1, u"115200 бод")
self.l_open_speed115200 = wx.StaticText(self, -1, u"...")
```

```

self.l_send_speed115200 = wx.StaticText(self, -1, u"...")
self.l_send_control_speed115200 = wx.StaticText(self, -1, u"...")
self.l_close_speed115200 = wx.StaticText(self, -1, u"...")
self.label_array = []
self.label_array.append([u"Скорость 110", self.l_speed110,
self.l_open_speed110, self.l_send_speed110,
self.l_send_control_speed110, self.l_close_speed110])
self.label_array.append([u"Скорость 300", self.l_speed300,
self.l_open_speed300, self.l_send_speed300,
self.l_send_control_speed300, self.l_close_speed300])
self.label_array.append([u"Скорость 600", self.l_speed600,
self.l_open_speed600, self.l_send_speed600,
self.l_send_control_speed600, self.l_close_speed600])
self.label_array.append([u"Скорость 1200", self.l_speed1200,
self.l_open_speed1200, self.l_send_speed1200,
self.l_send_control_speed1200, self.l_close_speed1200])
self.label_array.append([u"Скорость 2400", self.l_speed2400,
self.l_open_speed2400, self.l_send_speed2400,
self.l_send_control_speed2400, self.l_close_speed2400])
self.label_array.append([u"Скорость 4800", self.l_speed4800,
self.l_open_speed4800, self.l_send_speed4800,
self.l_send_control_speed4800, self.l_close_speed4800])
self.label_array.append([u"Скорость 9600", self.l_speed9600,
self.l_open_speed9600, self.l_send_speed9600,
self.l_send_control_speed9600, self.l_close_speed9600])
self.label_array.append([u"Скорость 19200", self.l_speed19200,
self.l_open_speed19200, self.l_send_speed19200,
self.l_send_control_speed19200,
self.l_close_speed19200])
self.label_array.append([u"Скорость 38400", self.l_speed38400,
self.l_open_speed38400, self.l_send_speed38400,

```



```

        self.l_send_control_speed38400, self.l_close_speed38400])
        self.label_array.append([u"Скорость 57600", self.l_speed57600,
self.l_open_speed57600, self.l_send_speed57600,
        self.l_send_control_speed57600,
self.l_close_speed57600])
        self.label_array.append([u"Скорость 115200", self.l_speed115200,
self.l_open_speed115200, self.l_send_speed115200,
        self.l_send_control_speed115200,
self.l_close_speed115200])
        self.tcp_data_array = []
        self.tcp_data_array.append([u"110", "\x81\x00\x00\x10"])
        self.tcp_data_array.append([u"300", "\x81\x00\x00\x11"])
        self.tcp_data_array.append([u"600", "\x81\x00\x00\x12"])
        self.tcp_data_array.append([u"1200", "\x81\x00\x00\x13"])
        self.tcp_data_array.append([u"2400", "\x81\x00\x00\x14"])
        self.tcp_data_array.append([u"4800", "\x81\x00\x00\x15"])
        self.tcp_data_array.append([u"9600", "\x81\x00\x00\x16"])
        self.tcp_data_array.append([u"19200", "\x81\x00\x00\x17"])
        self.tcp_data_array.append([u"38400", "\x81\x00\x00\x18"])
        self.tcp_data_array.append([u"57600", "\x81\x00\x00\x19"])
        self.tcp_data_array.append([u"115200", "\x81\x00\x00\x1a"])
        self.sb = self.CreateStatusBar(1, 0)
        self.tc_ip.SetValue('192.168.10.10')
        self.tc_port.SetValue('8090')
        self.__set_properties()
        self.__do_layout()
        self.__attach_events()
        #self.demo()
def __attach_events(self):
        self.Bind(wx.EVT_BUTTON, self.button_start, self.button_send)
        #self.Bind(wx.EVT_BUTTON, self.label_reset, self.button_reset)

```

```

        self.Bind(masked.EVT_NUM, self.port_correction, self.tc_port)
def __set_properties(self):
    self.SetTitle("Test tool")
    self.SetMinSize((750, 400))
    self.SetSize((750,400))
    self.SetMaxSize((950, 650))
    self.sb.SetStatusWidths([-1])
    self.sb.SetStatusText(u"",0)

self.SetBackgroundColour(wx.SystemSettings_GetColour(wx.SYS_COLOUR_BTNFACE))

def __do_layout(self):
    # begin wxGlade: MyFrame.__do_layout
    sizer_top = wx.BoxSizer(wx.VERTICAL)
    sizer_send = wx.BoxSizer(wx.HORIZONTAL)
    sizer_header = wx.BoxSizer(wx.HORIZONTAL)
    sizer_speed110 = wx.BoxSizer(wx.HORIZONTAL)
    sizer_speed300 = wx.BoxSizer(wx.HORIZONTAL)
    sizer_speed600 = wx.BoxSizer(wx.HORIZONTAL)
    sizer_speed1200 = wx.BoxSizer(wx.HORIZONTAL)
    sizer_speed2400 = wx.BoxSizer(wx.HORIZONTAL)
    sizer_speed4800 = wx.BoxSizer(wx.HORIZONTAL)
    sizer_speed9600 = wx.BoxSizer(wx.HORIZONTAL)
    sizer_speed19200 = wx.BoxSizer(wx.HORIZONTAL)
    sizer_speed38400 = wx.BoxSizer(wx.HORIZONTAL)
    sizer_speed57600 = wx.BoxSizer(wx.HORIZONTAL)
    sizer_speed115200 = wx.BoxSizer(wx.HORIZONTAL)
    sizer_send.Add(self.l_ip, 0, wx.ALL, 5)
    sizer_send.Add(self.tc_ip, 1, wx.ALL, 5)
    sizer_send.Add(self.l_port, 0, wx.ALL, 5)
    sizer_send.Add(self.tc_port, 1, wx.ALL, 5)

```

```
sizer_send.Add(self.button_send, 1, wx.ALL, 5)
#sizer_send.Add(self.button_reset, 1, wx.ALL, 5)
sizer_header.Add(self.l_speed_header, 2, wx.ALL, 5)
sizer_header.Add(self.l_open_header, 1, wx.ALL, 5)
sizer_header.Add(self.l_send_header, 1, wx.ALL, 5)
sizer_header.Add(self.l_send_control_header, 1, wx.ALL, 5)
sizer_header.Add(self.l_close_header, 1, wx.ALL, 5)
sizer_speed110.Add(self.l_speed110, 2, wx.ALL, 5)
sizer_speed110.Add(self.l_open_speed110, 1, wx.ALL, 5)
sizer_speed110.Add(self.l_send_speed110, 1, wx.ALL, 5)
sizer_speed110.Add(self.l_send_control_speed110, 1, wx.ALL, 5)
sizer_speed110.Add(self.l_close_speed110, 1, wx.ALL, 5)
sizer_speed300.Add(self.l_speed300, 2, wx.ALL, 5)
sizer_speed300.Add(self.l_open_speed300, 1, wx.ALL, 5)
sizer_speed300.Add(self.l_send_speed300, 1, wx.ALL, 5)
sizer_speed300.Add(self.l_send_control_speed300, 1, wx.ALL, 5)
sizer_speed300.Add(self.l_close_speed300, 1, wx.ALL, 5)
sizer_speed600.Add(self.l_speed600, 2, wx.ALL, 5)
sizer_speed600.Add(self.l_open_speed600, 1, wx.ALL, 5)
sizer_speed600.Add(self.l_send_speed600, 1, wx.ALL, 5)
sizer_speed600.Add(self.l_send_control_speed600, 1, wx.ALL, 5)
sizer_speed600.Add(self.l_close_speed600, 1, wx.ALL, 5)
sizer_speed1200.Add(self.l_speed1200, 2, wx.ALL, 5)
sizer_speed1200.Add(self.l_open_speed1200, 1, wx.ALL, 5)
sizer_speed1200.Add(self.l_send_speed1200, 1, wx.ALL, 5)
sizer_speed1200.Add(self.l_send_control_speed1200, 1, wx.ALL, 5)
sizer_speed1200.Add(self.l_close_speed1200, 1, wx.ALL, 5)
sizer_speed2400.Add(self.l_speed2400, 2, wx.ALL, 5)
sizer_speed2400.Add(self.l_open_speed2400, 1, wx.ALL, 5)
sizer_speed2400.Add(self.l_send_speed2400, 1, wx.ALL, 5)
sizer_speed2400.Add(self.l_send_control_speed2400, 1, wx.ALL, 5)
```

*sizer\_speed2400.Add(self.l\_close\_speed2400, 1, wx.ALL, 5)*  
*sizer\_speed4800.Add(self.l\_speed4800, 2, wx.ALL, 5)*  
*sizer\_speed4800.Add(self.l\_open\_speed4800, 1, wx.ALL, 5)*  
*sizer\_speed4800.Add(self.l\_send\_speed4800, 1, wx.ALL, 5)*  
*sizer\_speed4800.Add(self.l\_send\_control\_speed4800, 1, wx.ALL, 5)*  
*sizer\_speed4800.Add(self.l\_close\_speed4800, 1, wx.ALL, 5)*  
*sizer\_speed9600.Add(self.l\_speed9600, 2, wx.ALL, 5)*  
*sizer\_speed9600.Add(self.l\_open\_speed9600, 1, wx.ALL, 5)*  
*sizer\_speed9600.Add(self.l\_send\_speed9600, 1, wx.ALL, 5)*  
*sizer\_speed9600.Add(self.l\_send\_control\_speed9600, 1, wx.ALL, 5)*  
*sizer\_speed9600.Add(self.l\_close\_speed9600, 1, wx.ALL, 5)*  
*sizer\_speed19200.Add(self.l\_speed19200, 2, wx.ALL, 5)*  
*sizer\_speed19200.Add(self.l\_open\_speed19200, 1, wx.ALL, 5)*  
*sizer\_speed19200.Add(self.l\_send\_speed19200, 1, wx.ALL, 5)*  
*sizer\_speed19200.Add(self.l\_send\_control\_speed19200, 1, wx.ALL, 5)*  
*sizer\_speed19200.Add(self.l\_close\_speed19200, 1, wx.ALL, 5)*  
*sizer\_speed38400.Add(self.l\_speed38400, 2, wx.ALL, 5)*  
*sizer\_speed38400.Add(self.l\_open\_speed38400, 1, wx.ALL, 5)*  
*sizer\_speed38400.Add(self.l\_send\_speed38400, 1, wx.ALL, 5)*  
*sizer\_speed38400.Add(self.l\_send\_control\_speed38400, 1, wx.ALL, 5)*  
*sizer\_speed38400.Add(self.l\_close\_speed38400, 1, wx.ALL, 5)*  
*sizer\_speed57600.Add(self.l\_speed57600, 2, wx.ALL, 5)*  
*sizer\_speed57600.Add(self.l\_open\_speed57600, 1, wx.ALL, 5)*  
*sizer\_speed57600.Add(self.l\_send\_speed57600, 1, wx.ALL, 5)*  
*sizer\_speed57600.Add(self.l\_send\_control\_speed57600, 1, wx.ALL, 5)*  
*sizer\_speed57600.Add(self.l\_close\_speed57600, 1, wx.ALL, 5)*  
*sizer\_speed115200.Add(self.l\_speed115200, 2, wx.ALL, 5)*  
*sizer\_speed115200.Add(self.l\_open\_speed115200, 1, wx.ALL, 5)*  
*sizer\_speed115200.Add(self.l\_send\_speed115200, 1, wx.ALL, 5)*  
*sizer\_speed115200.Add(self.l\_send\_control\_speed115200, 1, wx.ALL, 5)*  
*sizer\_speed115200.Add(self.l\_close\_speed115200, 1, wx.ALL, 5)*

```

sizer_top.Add(sizer_send, 1, wx.EXPAND, 0)
sizer_top.Add(sizer_header, 1, wx.EXPAND, 0)
sizer_top.Add(sizer_speed110, 1, wx.EXPAND, 0)
sizer_top.Add(sizer_speed300, 1, wx.EXPAND, 0)
sizer_top.Add(sizer_speed600, 1, wx.EXPAND, 0)
sizer_top.Add(sizer_speed1200, 1, wx.EXPAND, 0)
sizer_top.Add(sizer_speed2400, 1, wx.EXPAND, 0)
sizer_top.Add(sizer_speed4800, 1, wx.EXPAND, 0)
sizer_top.Add(sizer_speed9600, 1, wx.EXPAND, 0)
sizer_top.Add(sizer_speed19200, 1, wx.EXPAND, 0)
sizer_top.Add(sizer_speed38400, 1, wx.EXPAND, 0)
sizer_top.Add(sizer_speed57600, 1, wx.EXPAND, 0)
sizer_top.Add(sizer_speed115200, 1, wx.EXPAND, 0)
self.SetSizer(sizer_top)
#sizer_top.Fit(self)
self.Layout()
def button_start(self,event):
    test_thread = FuncThread(self.start_test)
    test_thread.start()
def start_test(self):
    self.label_reset(None)
    """
    i=0
    for data in self.tcp_data_array:
        print data[0]
        print ByteToHex(data[1])
        self.label_array[i][2].SetLabel("OK")
        i=i+1
    for data in self.tcp_data_array:
        print ByteToHex(data[1][2])
        print ByteToHex(data[1][3])

```

```

"""
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
TCP_IP = self.tc_ip.GetValue()
TCP_PORT = int(self.tc_port.GetValue())
s.settimeout(10)
try:
    s.connect((TCP_IP, TCP_PORT))
except socket.error:
    self.sb.SetStatusText(u"Подключение не удалось",0)
else:
    s.settimeout(None)
    self.sb.SetStatusText(u"Успешное подключение. Идем
тестирование...",0)
    i=0
    for data in self.tcp_data_array:
        s.send(data[1])
        print data[0]
        answer = s.recv(4)
        if answer[1] == "\x01" and answer[2]== data[1][2] and
answer[3]== data[1][3]:
            self.label_array[i][2].SetLabel("OK")
        #write
        s.send("\x02\x00\x00\x01\x55")
        write_answer = s.recv(4)
        if write_answer[1] == "\x01":
            self.label_array[i][3].SetLabel("OK")
        #write and check
        s.send("\x02\x01\x00\x01\xAA")
        write_answer = s.recv(4)
        if write_answer[1] == "\x01":
            self.label_array[i][3].SetLabel("OK")

```

```

        elif write_answer[1] == "\x03":
            self.label_array[i][3].SetLabel("Error")
#close
        s.send("\x82\x00\x00\x00")
        close_answer = s.recv(4)
        if close_answer[1] == "\x00" and answer[2]== data[1][2] and
answer[3]== data[1][3]:
            self.label_array[i][5].SetLabel("OK")
        elif answer[1] == "\x01" and answer[3]!= data[1][3]:
            for datax in self.tcp_data_array:
                if answer[3] == datax[1][3]:
                    self.label_array[i][2].SetLabel(datax[0])
        elif answer[1] == "\x00":
            for datax in self.tcp_data_array:
                if answer[3] == datax[1][3]:
                    self.label_array[i][2].SetLabel("Can't open")
        i=i+1
        time.sleep(0.1)
    s.close()
    self.sb.SetStatusText(u"Тестирование завершено",0)
    self.Layout()
def label_reset(self,event):
    for label in self.label_array:
        label[2].SetLabel("...")
        label[3].SetLabel("...")
        label[4].SetLabel("...")
        label[5].SetLabel("...")
    self.Layout()
def port_correction(self,event):
    num_ctrl = event.GetEventObject()
    if num_ctrl.GetValue()>num_ctrl.GetMax():

```

```

        num_ctrl.SetValue(num_ctrl.GetMax())
def demo(self):
    for label in self.label_array:
        if label[0]==u"Скорость 110":
            label[2].SetLabel("Can't open")
            label[3].SetLabel("...")
            label[4].SetLabel("...")
            label[5].SetLabel("...")
        elif label[0]==u"Скорость 57600" or label[0]==u"Скорость
115200":
            label[2].SetLabel("38400")
            label[3].SetLabel("...")
            label[4].SetLabel("...")
            label[5].SetLabel("...")
        else:
            label[2].SetLabel("OK")
            label[3].SetLabel("OK")
            label[4].SetLabel("OK")
            label[5].SetLabel("OK")
    self.Layout()
if __name__ == "__main__":
    app = wx.PySimpleApp(0)
    wx.InitAllImageHandlers()
    frame_1 = TestTool(None, -1, "")
    app.SetTopWindow(frame_1)
    frame_1.Show()
    app.MainLoop()

```