МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

**ОТЧЁТ**

**о прохождении преддипломной практики**

**студентом 4 курса, 143 учебной группы**

Ивановым Иваном Ивановичем

*(Ф.И.О. студента)*

Направление подготовки: 09.03.04 «Программная инженерия»

*(шифр и наименование)*

Направленность:

Бакалавр .

*(наименование образовательной программы)*

Тема ВКР: Разработка программного обеспечения для взаимодействия с камерой, предназначенной для распознавания двумерных кодов.

Сроки прохождения практики:

с «21» апреля 2025 г. по «21» мая 2025 г.

Место проведения преддипломной практики

ОКБ «Спектр»

Руководитель практики: к.т.н., ст. пр. кафедры ВПМ, Шадрин Максим Владимирович

(должность, ФИО)

Рязань 2025

**Основные итоги преддипломной практики:**

В процессе прохождения преддипломной практики было сделано следующее:

- осуществлён анализ требований;

- произведено проектирование ПО;

- осуществлена программная реализации ПО;

- осуществлено тестирование ПО.

Студент Рябинин Сергей Александрович ммм

*(Ф.И.О. студента, подпись)*

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2025 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики Шадрин Максим Владимирович сссссссссссссс

*(Ф.И.О., подпись)*

# 

# Введение

Современные производственные, логистические и торговые системы всё чаще используют автоматизированные средства идентификации и учёта продукции. Одним из самых эффективных способов представления информации на компактной площади является двумерный штрихкод (QR, DataMatrix и др.). Для корректной работы таких систем требуется надёжное программное обеспечение, обеспечивающее стабильное взаимодействие с аппаратурой — в частности, с камерами, осуществляющими захват изображения и распознавание кодов.

В рамках производственной практики была поставлена задача разработки программного обеспечения, предназначенного для взаимодействия с камерой, передающей данные по протоколу UDP, и осуществляющего получение, интерпретацию и отображение закодированной информации из двумерных кодов.

# Цель и задачи практики

**Цель практики** — разработка и отладка программного обеспечения, обеспечивающего стабильное соединение с промышленной камерой, получение и обработку данных, а также визуализацию результатов распознавания двумерных кодов.

Для достижения этой цели были поставлены следующие **задачи**:

* изучить протокол обмена данными между камерой и компьютером;
* определить структуру передаваемых байтовых команд;
* реализовать модуль приёма данных с использованием протокола UDP;
* встроить поддержку декодирования двумерных кодов;
* реализовать простой графический интерфейс для отображения результатов;
* провести тестирование и анализ корректности работы программы.

# Объект и предмет разработки

**Объектом разработки** является система автоматизированного распознавания двумерных кодов с использованием камеры и персонального компьютера.

**Предметом разработки** является программное обеспечение, реализующее взаимодействие с камерой через сетевой протокол и выполняющее распознавание двумерных кодов на полученных изображениях.

# Актуальность

## Применение 2D-кодов в современных системах

Двумерные штрихкоды (QR-коды, DataMatrix и другие форматы) получили широкое распространение в различных сферах: логистика, розничная торговля, здравоохранение, производство, биллинговые и маркетинговые системы. По сравнению с линейными штрихкодами, 2D-коды обладают рядом преимуществ — они компактны, вмещают больше информации и устойчивы к частичным повреждениям благодаря встроенным алгоритмам коррекции ошибок.

Современные системы учёта и автоматизации всё чаще полагаются на идентификацию через 2D-коды, что требует надёжного аппаратного и программного обеспечения, обеспечивающего быструю и точную обработку данных. На многих предприятиях камеры фиксируют коды в реальном времени, передают данные по сетевым протоколам, и далее они интерпретируются программным обеспечением на серверной или пользовательской стороне.

## Применение 2D-кодов в современных системах

Хотя на рынке присутствует ряд готовых решений для распознавания 2D-кодов, зачастую они являются:

* слишком универсальными и неэффективными в узкоспециализированных сценариях;
* платными или ограниченными по лицензии;
* сложными в настройке и интеграции с нестандартными камерами или протоколами передачи данных.

Кроме того, промышленные камеры, применяемые на предприятиях, нередко работают по нестандартным схемам обмена данными (например, передают байтовые команды по UDP), что требует точной настройки программной части.

Разработка собственного программного обеспечения позволяет:

* учесть специфику используемой камеры и формата данных;
* адаптировать интерфейс под требования оператора;
* обеспечить высокую скорость и надёжность распознавания;
* снизить затраты на приобретение и поддержку сторонних решений.

Таким образом, разработка индивидуального программного обеспечения представляет собой актуальную и практическую задачу, особенно в условиях, когда требуется высокая степень интеграции с производственным оборудованием.

# Анализ аналогов

## Обзор существующих аналогов

В настоящее время существует множество программных решений, предназначенных для распознавания двумерных кодов. Ниже приведены некоторые из наиболее известных решений:

* ZBar — кроссплатформенная библиотека для сканирования штрихкодов, поддерживает QR, DataMatrix и другие форматы. Может быть интегрирована с Python, C/C++, Java. Основной недостаток — устаревший интерфейс и ограниченные возможности настройки под специфическое оборудование.
* OpenCV + pyzbar / OpenCV + ZXing — популярный подход, сочетающий мощные возможности компьютерного зрения с библиотеками для декодирования. Обеспечивает гибкость, но требует дополнительных усилий по обработке сетевых протоколов и взаимодействию с нестандартными камерами.
* Dynamsoft Barcode Reader — коммерческое решение с высокой скоростью и точностью распознавания. Поддерживает множество платформ, но требует приобретения лицензии, что может быть критично для учебных и мелких промышленных проектов.
* Программное обеспечение производителей камер (например, Hikvision, Basler) — предоставляет SDK и приложения для работы с изображениями и видеопотоками с камер. Однако такие решения, как правило, "заточены" под конкретные модели устройств и часто не подходят для интеграции с кастомными системами.

## Выводы по результатам сравнения

На основе проведённого анализа можно выделить следующие проблемы существующих решений:

* ограниченная поддержка специфических протоколов передачи данных (например, UDP с собственным форматом команд);
* избыточность функционала при решении конкретной задачи;
* высокая стоимость при использовании коммерческих библиотек;
* отсутствие интеграции с нужным оборудованием или невозможность быстрой адаптации.

В связи с этим разработка собственного программного обеспечения позволяет получить решение, полностью соответствующее требованиям конкретного проекта, с гибкой архитектурой, поддержкой нужного протокола и минимальной зависимостью от сторонних библиотек.

# Реализация программного обеспечения

## Выбор протокола обмена (UDP)

Для обмена данными между компьютером и интеллектуальной камерой выбран протокол **UDP**. Такой выбор обусловлен следующими причинами:

* **Минимальная задержка передачи данных**, что критически важно при работе с потоковыми видеоданными и при высокой частоте считывания кодов.
* **Отсутствие необходимости в подтверждении получения пакетов**, что снижает нагрузку на процессор и ускоряет обмен.
* **Простота реализации** на уровне прикладного программного обеспечения.

UDP позволяет реализовать быстрый и прямой обмен короткими командами управления, а также передачу изображения и результатов распознавания.

## Формат байтовых команд между ПО и камерой

Обмен информацией между программным обеспечением и камерой реализован с помощью строго определённых байтовых команд. Команды делятся на две группы: команды **отправки** и **получения**.

**Команды, отправляемые камере:**

1. **Установка параметров сенсора (COP = 0xB0)**

**Структура пакета**: COP | SUBCOP | Pad (2 байта) | [Param]

* SUBCOP = 0x00 — инициализация сенсора.
* SUBCOP = 0x01 — установка экспозиции. Значение передаётся в param (2 байта: младший, затем старший).
* SUBCOP = 0x04 — установка значения чувствительности яркости (TAI). Передаётся аналогично.

1. **Запись/чтение всех параметров устройства (COP = 0x04)**

**Структура пакета**: COP | SUBCOP | Pad (2 байта) | [Param]

* SUBCOP = 0x00 — запись всех параметров (используется поле Param)
* SUBCOP = 0x01 — чтение текущих параметров (поле Param не используется)

**Содержимое поля Data (начиная с байта 4):**

|  |  |
| --- | --- |
| **Байты** | **Назначение** |
| 4-7 | udp\_delay — задержка передачи строк |
| 8-11 | DY — координата Y верхнего левого угла ROI |
| 12-15 | DX — координата X верхнего левого угла ROI |
| 16-19 | im\_height — высота изображения или ROI |
| 20-23 | im\_width — ширина изображения или ROI |
| 24-27 | count\_code — макс. количество распознаваемых кодов (0 = все) |
| 28 | live\_video — включение получения кадров с сенсора |
| 29 | frame\_eth\_tx — включение передачи кадров по UDP |
| 30-31 | Значение экспозиции (младший/старший байт) |
| 32 | AEC — автоэкспозиция: 0x00 – выкл., 0x01 – вкл. |
| 33 | Длина импульса подсветки для ШИМ |
| 34 | Режим триггера |
| 35 | Флаг активации алгоритма декодирования |
| 36-39 | delay\_cnt — значение задержки для ШИМ |
| 40-41 | div\_cnt — делитель частоты для ШИМ |
| 42-43 | Зарезервированы |
| 44-45 | Значения чувствительности яркости TAI |
| 46-47 | trashB — верхний порог алгоритма локализации |
| 48-49 | trashL — нижний порог алгоритма локализации |
| 50 | black\_lev — уровень черного |
| 51 | minW\_length — минимум уровня белого |
| 52-53 | maxW\_length — максимум уровня белого |
| 54 | Алгоритм декодирования (см. ниже) |
| 55-59 | Настройки альтернативного алгоритма локализации |

**Алгоритмы декодирования (байт 54):**

* 0x00 — Data\_matrix (Fast)
* 0x01 — AUTO
* 0x02 — AZTEC
* 0x03 — CODABAR
* 0x04 — CODE\_39
* 0x05 — CODE\_93
* 0x06 — CODE\_128
* 0x07 — медленный DATA\_MATRIX
* 0x08 — EAN\_8
* 0x09 — EAN\_13
* 0x0A — ITF
* 0x0C — PDF\_417
* 0x0D — QR\_CODE
* 0x10 — UPC\_A
* 0x11 — UPC\_E

**Команды, получаемые от камеры:**

* 1. **Изображение с камеры (COP = 0xB1)**

**Структура**: COP | id\_row (2 байта) | Pad | Pixels

* id\_row — номер строки изображения
* С 4 байта начинаются пиксели строки
  1. **Результаты распознавания (COP = 0xA9)**
* 2 байт — тип кода (значение - 1 соответствует типу из списка выше)
* 3 байт — количество найденных кодов
* 4 байт — номер текущего кода
* 5-6 байты — количество координат
* Далее — пары координат (по 2 байта)
* 22-23 — длина строки содержимого
* С 24 байта — строковое содержимое кода
  1. **Информация о DataMatrix-коде (COP = 0xAA)**
* Len — длина данных
* id — номер кода
* Pad — выравнивание
* Pos\_Y, Pos\_X — координаты верхнего левого угла кода
* BUF\_for\_Parsing — строка с содержимым кода

## Архитектура и структура ПО

Программное обеспечение разделено на две основные части: **библиотеку взаимодействия с камерой** и **интерфейсное приложение**. Такая архитектура обеспечивает модульность, повторное использование кода и удобную интеграцию в различные проекты.

#### Библиотека взаимодействия с камерой

**Описание классов библиотеки**:

1. **SmartCameraSystem** — основной класс, содержащий высокоуровневые методы взаимодействия.
2. **Camera** — класс, описывающий камеру, содержит:

* методы отправки/приёма UDP-пакетов;
* IP-адрес камеры;
* предопределённые пакеты: PackageInitCamera, PackageGetImageCamera, PackageStopVideoCamera.

1. **ImageOutput** — работа с изображением, включая:

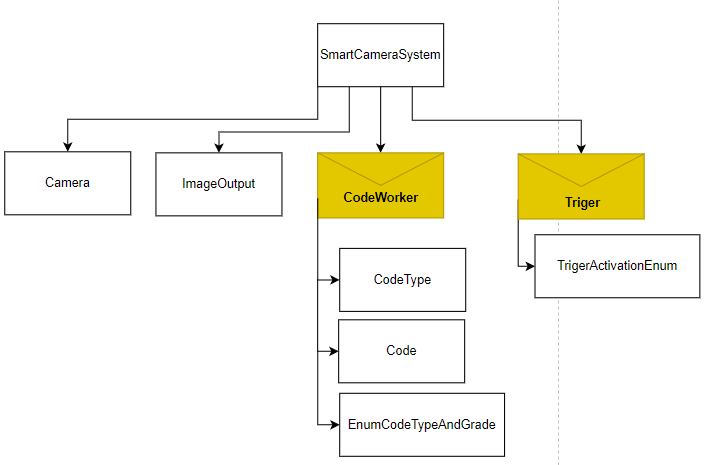
* DisplayImage — визуализация изображения;
* HighlightCode — выделение DataMatrix кода.

1. **Папка CodeWorker**:

* Code — структура распознанного кода;
* CodeType — сопоставление типов и байтовых значений кодов;
* EnumCodeTypeAndGrade — перечисления типов кодов и грейдов.

1. **Папка Triger**:

* TrigerActivationEnum — перечисления режимов триггера.



## Интерфейс пользователя

Графическое интерфейсное приложение обеспечивает удобную работу с камерой и распознаванием кодов. Библиотека подключается к проекту и управляется через элементы интерфейса.

**Возможности интерфейса:**

* Получение и отображение изображения в реальном времени с камеры, что позволяет оперативно оценить качество съёмки и корректность настроек.
* Гибкая настройка параметров съёмки, включая экспозицию, уровень яркости, область интереса (ROI) и другие параметры, критичные для надёжного распознавания.
* Запуск процесса декодирования штрихкодов по нажатию кнопки, а также возможность выполнения автоматического распознавания при включённом триггере.
* Отображение детальной информации по каждому распознанному коду:
* тип кода (например, DataMatrix, QR, EAN и др.);
* текстовое содержимое;
* координаты расположения кода на изображении (основные и дополнительные);
* грейд кода, отражающий качество его печати или контрастности.
* Визуальное выделение обнаруженного DataMatrix кода зелёной рамкой непосредственно на изображении, что позволяет быстро идентифицировать расположение и проверить корректность захвата.
* Управление режимом работы камеры: включение/отключение триггерного режима, настройка подсветки, активация автосъёмки и другие вспомогательные функции.

Таким образом, интерфейсное приложение не только обеспечивает полный цикл взаимодействия с камерой, но и делает этот процесс максимально простым, прозрачным и адаптируемым под различные сценарии использования.

# Интерфейс приложения

