HORUS - Универсальная система контроля дефектов производственных линий с функционалом автоматического контроля дефектов продукции, сканирования кодов маркировки, управления производственной линией и визуализации результатов работы

Описание программы

2023

АННОТАЦИЯ

В данном документе приведено описание программно-аппаратного комплекса, предназначенного для автоматического контроля дефектов продукции, сканирования кодов маркировки, управления производственной линией и визуализации результатов работы. Комплекс может быть использован в промышленности, использующей конвейерные линии с целью повысить и контролировать качество выпускаемой продукции.

СОДЕРЖАНИЕ

Оглавление

АННОТАЦИЯ....2
СОДЕРЖАНИЕ....3
1. ОБЩИЕ СВЕДЕНИЯ....5
1.1. Обозначение и наименование программы....5
1.2. Программное обеспечение, необходимое для функционирования программы....5
1.3. Языки программирования, использованные при разработке....5
2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ....6
3. ОПИСАНИЕ СТРУКТУРЫ, ЛОГИЧЕСКОЙ СХЕМЫ и интерфейса программы....7
3.1. СТРУКТУРА ПРОГРАММЫ....7
3.2. АЛГОРИТМ ПРОГРАММЫ....9
3.2.1. Общий алгоритм....9
3.2.2. Считывание маркировки....11
3.2.3. Контроль геометрических отклонений....12
3.2.4. Контроль визуальных дефектов....13
3.2.5. Начальная настройка системы....14
3.3. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС ПРОГРАММЫ....18
3.3.1. Ролевая модель....18
3.3.2. Экран администратора....19
3.3.3. Экран оператора....24
4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА....26
5. ВЫЗОВ, ЗАГРУЗКА, ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ....27
5.1. Вызов и загрузка....27
5.2. Входные данные....27
5.3. Выходные данные....27

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Обозначение и наименование программы

Программно-аппаратный комплекс (далее - Horus), предназначен для автоматического контроля дефектов продукции (как 2D, так и 3D), сканирования кодов маркировки, управления производственной линией и визуализации результатов работы. Комплекс может быть использован в промышленности, использующей конвейерные линии с целью повысить и контролировать качество выпускаемой продукции.

1.2. Программное обеспечение, необходимое для функционирования программы

- Необходим предустановленный интерпретатор Python3.8. в папку по умолчанию «/usr/bin/python3.8»;
- Необходим установленный Docker Engine;
- Установленная библиотека работы с маркировкой libdmtx-dev;
- Запущенный контейнер mongo;
- Для выполнения вычислений на GPU рекомендовано использование архитектуры Nvidia CUDA. В противном случае дополнительное ПО не требуется, все необходимые зависимости поставляются в рамках дистрибутива.

1.3. Языки программирования, использованные при разработке

- Для реализации основных структур данных, алгоритмов и элементов пользовательского интерфейса использовался язык Python 3.8.

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Программно-аппаратный комплекс Horus предназначен для определения визуальных и геометрических дефектов изделий, а также управления производственной линией.

Комплекс может быть использован в промышленности, использующей конвейерные линии с целью повысить и контролировать качество выпускаемой продукции.

3. ОПИСАНИЕ СТРУКТУРЫ, ЛОГИЧЕСКОЙ СХЕМЫ И ИНТЕРФЕЙСА ПРОГРАММЫ

3.1. СТРУКТУРА ПРОГРАММЫ

Общая структура программного комплекса Horus соответствует схеме, приведенной на рисунке Рисунок 1.

Комплекс включет в себя следующие блоки:

- сервис вычислительной аналитики;
- сервис взаимодействия камер;
- сервис взаимодействия с внешними устройствами;
- базы данных;
- графический интерфейс.

Блоки сервисов вычислительной аналитики, взаимодействия камер и взаимодействия с внешними устройствами реализованы в соответствии с архитектурным стилем REST API.

Блок сервиса вычислительной аналитики обеспечивает выполнение функций анализа изделия в соответствии с настройками, заданными пользователем. Для выполнения каждой из функций анализа (считывание маркировки, контроль геометрических отклонений и визуальных дефектов) предусмотрен отдельный модуль.

Блок сервиса взаимодействия камер обеспечивает возможность подключения 2D и 3D камер, контроля их статуса («в работе» \ «не в работе») и настройки камер с помощью ПО производителя. Кроме того, в рамках этого модуля реализован цикл постоянного обновления буфера изображений и 3D-сцен.

Блок сервиса взаимодействия с внешними устройствами обеспечивает возможность подключения внешних устройств, контроля их статуса («в работе» \ «не в работе») и настройки данных устройств с помощью ПО производителя.

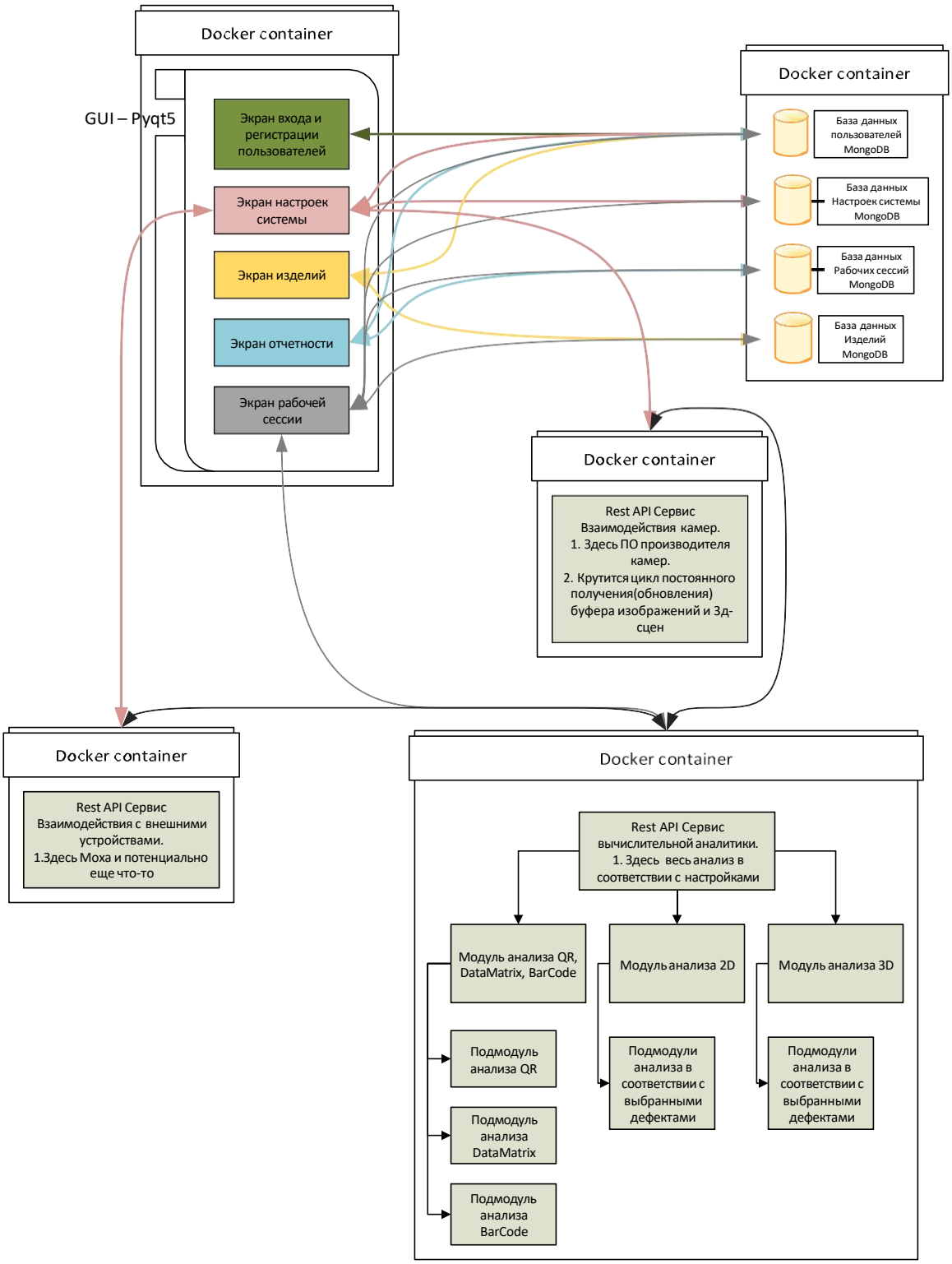


Рисунок 1 — Структура программного комплекса Horus

Графический интерфейс пользователя должен реализован с помощью средств пакета PyQt5. Интерфейс должен включает в себя:

- экраны входа и регистрации пользователей;
- экран настроек системы;

- экран изделий;
- экран формирования отчетности;
- экран рабочей сессии.

3.2. АЛГОРИТМ ПРОГРАММЫ

3.2.1. Общий алгоритм

Программно-аппаратный комплекс обеспечивает выполнение следующего алгоритма (рисунок Рисунок 2).

После начальной настройки системы комплекс выполняет автоматический анализ изделия в зоне обзора. Вхождение изделия в зону обзора определятся по сигналу от барьерного инфракрасного датчика. Факт выхода изделия из зоны обзора определяется по сигналу о преодолении второго барьера.

За время нахождения изделия в зоне обзора камеры комплекс выполняет фотографирование сцены, на полученном изображении обнаруживает и идентифицирует изделие, и далее выполняет его анализ в соответствии с заданными настройками. В рамках анализа комплекс обеспечивает возможность считывания маркировки, контроля геометрических отклонений и визуальных дефектов.

По результатам анализа комплекс автоматически формирует необходимое воздействие в соответствии с настройками и оповещает оператора в случае ошибки детекции изделия, идентификации изделия, анализа изделия, а также при обнаружении дефекта.

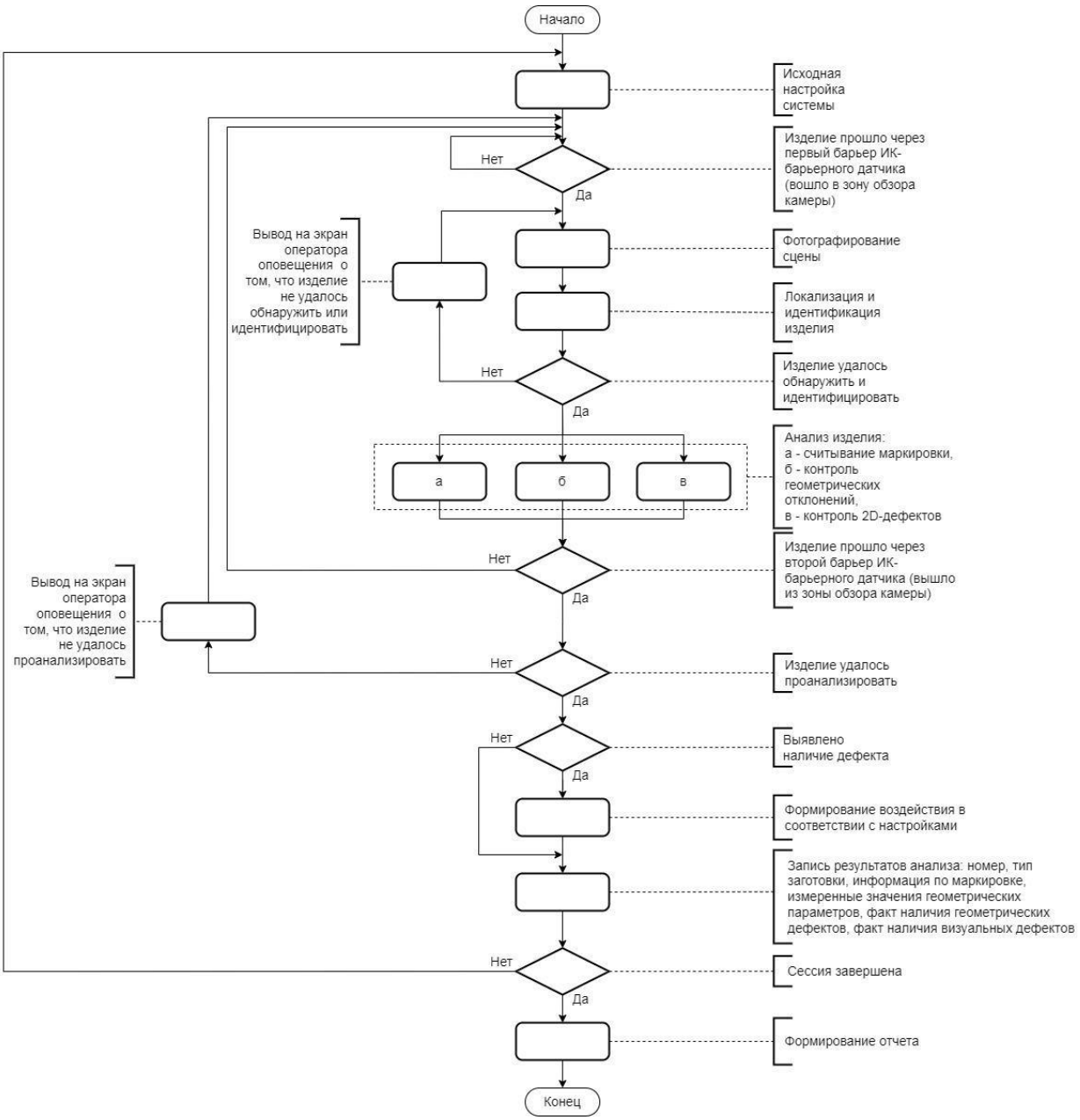


Рисунок 2 — Общий алгоритм работы комплекса Horus

Под воздействием в общем случае понимается некоторое настраиваемое пользователем действие над изделием, выполняемое с помощью внешнего устройства (например, перемещение или удаление с конвейера при обнаружении дефекта).

Работа комплекса организована в формате сессий, запустить и завершить которые может оператор комплекса. Предусмотрена возможность автоматического формирования отчетности по каждой сессии.

Программно-аппаратный комплекс обеспечивает возможность выполнения следующих видов анализа:

- считывание маркировки на изделии (QR-код, DataMatrix, штрихкод);
- контроль геометрических отклонений;
- контроль визуальных дефектов.

В зависимости от начальных настроек, комплекс обеспечивает возможность выполнения в рамках одной сессии для каждого изделия одного, двух или всех трех видов анализа.

3.2.2. Считывание маркировки

В рамках данного модуля обеспечивается возможность настройки параметров, необходимых для точного распознавания маркировок продукции. В специализированном разделе модуля реализован функционал, позволяющий определить расположение искомой маркировки на заданном изделии, а также определить ее тип.

Обеспечивается возможность работы со следующими типами маркировок:

- Barcode (штрих-код);
- DataMatrix (Честный знак);
- QR-код.

Функция считывания маркировки реализована в соответствии со следующим алгоритмом (рисунок Рисунок 3).

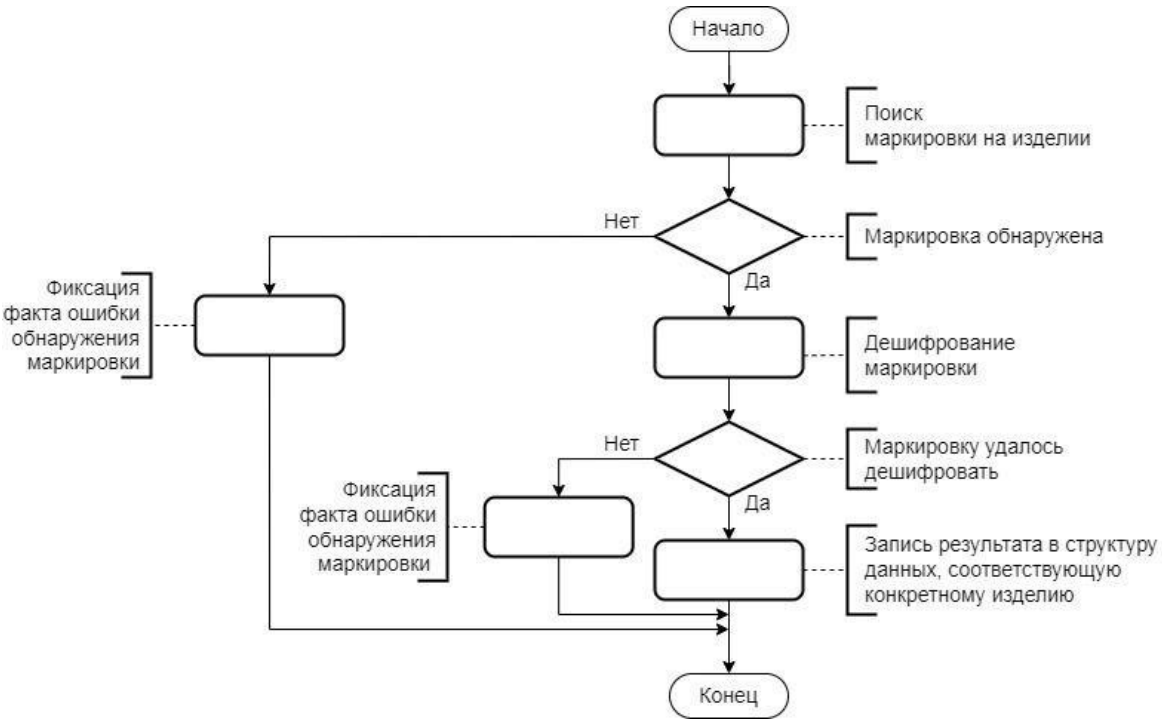


Рисунок 3 — Алгоритм считывания маркировки на изделии

3.2.3. Контроль геометрических отклонений

При реализации контроля геометрических отклонений предусмотрена возможность ввода и выбора контролируемых размеров изделия, а также возможность ввода их нормативных значений и допусков.

Под контролируемым размером понимается габаритный размер изделия (длина, ширина, высота для изделия типа «куб» или радиус, высота для изделия типа «цилиндр»), факт отклонения от которого должна фиксировать система при реализации функции контроля геометрических отклонений.

Функция контроля геометрических отклонений реализована в соответствии со следующим алгоритмом (рисунок Рисунок 4).



Рисунок 4 — Алгоритм контроля геометрических отклонений

В рамках комплекса предусмотрено наличие 3D-камеры и возможность создания с ее помощью облака точек сцены. После съемки сцены из общего облака точек выделяется область, описывающая изделие. По полученной части облака измеряются контролируемые размеры изделия, после чего результаты измерения сравниваются с заданными в процессе настройки нормативными значениями.

3.2.4. Контроль визуальных дефектов

При реализации контроля визуальных дефектов предусмотрена возможность выбора и добавления типов изделий и типов контролируемых дефектов.

Под контролируемым дефектом понимается вид визуального дефекта изделия (усадочные раковины, трещины, сколы и т.д.), факт наличия которого должна фиксировать система при реализации функции контроля визуальных дефектов.

Функция контроля визуальных дефектов реализована в соответствии со следующим алгоритмом (рисунок Рисунок 5).

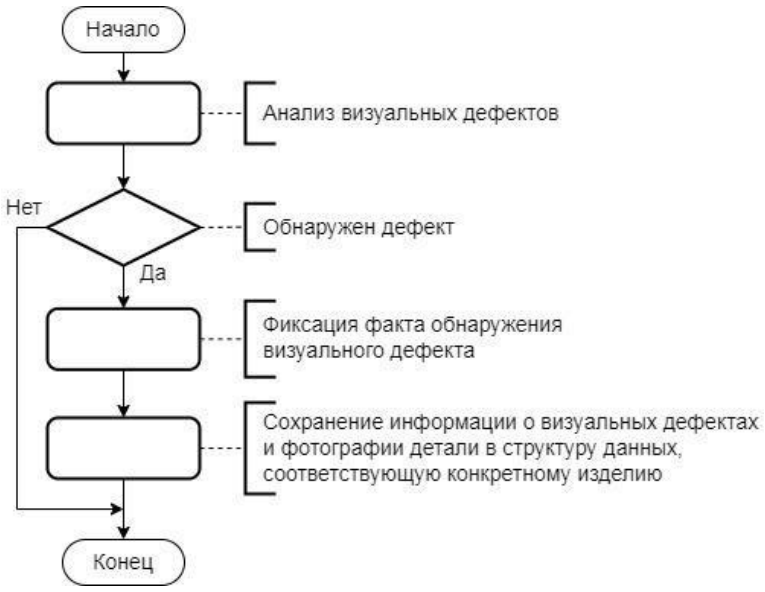


Рисунок 5 — Алгоритм контроля визуальных дефектов

Контроль осуществляется методом анализа 2D-изображения сцены. При обнаружении визуального дефекта предусмотрено возможность сохранения информации о дефекте и фотографии детали в структуру данных, соответствующую данному изделию, а также возможность задания и настройки воздействия на изделие с обнаруженным дефектом.

3.2.5. Начальная настройка системы

При разработке комплекса предусмотрена возможность начальной настройки системы в соответствии с последовательностью, показанной на рисунке Рисунок 6.

В рамках комплекса камеры, датчики должны быть подключены в единую систему, для чего реализована возможность добавления и настройки устройств, входящих в состав программно-аппаратного комплекса (2D камеры, 3D камеры, а также устройств расширения) (рисунок Рисунок 7).



Рисунок 6 — Последовательность начальной настройки системы



Рисунок 7 — Определение конфигурации системы

Модуль предоставляет возможность добавления новой камеры при помощи указания следующих ее параметров:

- название;
- модель;
- разрешение;

- IP-адрес.

Подключение устройств реализовано путем ввода IP-адреса устройства в программном обеспечении комплекса Horus, настройку устройств — через программное обеспечение производителя.

В рамках ПО комплекса Horus также реализован раздел «Библиотека изделий» и обеспечивающий возможность добавления в него изделий, которые предстоит анализировать. Добавление изделия реализовано в соответствии с алгоритмом, представленным на рисунке Рисунок 8.

При добавлении в библиотеку нового изделия предусмотрена возможность ввода наименования изделия и номера чертежа (конструкторской документации), в соответствии с которой производится данное изделие. Далее предусмотрена опция выбора возможностей контроля, а также опция добавления и настройки воздействия в рамках выбранных возможностей контроля.

Под воздействием в данном случае подразумевается перемещение или удаление с конвейера конкретного изделия с помощью внешнего устройства.

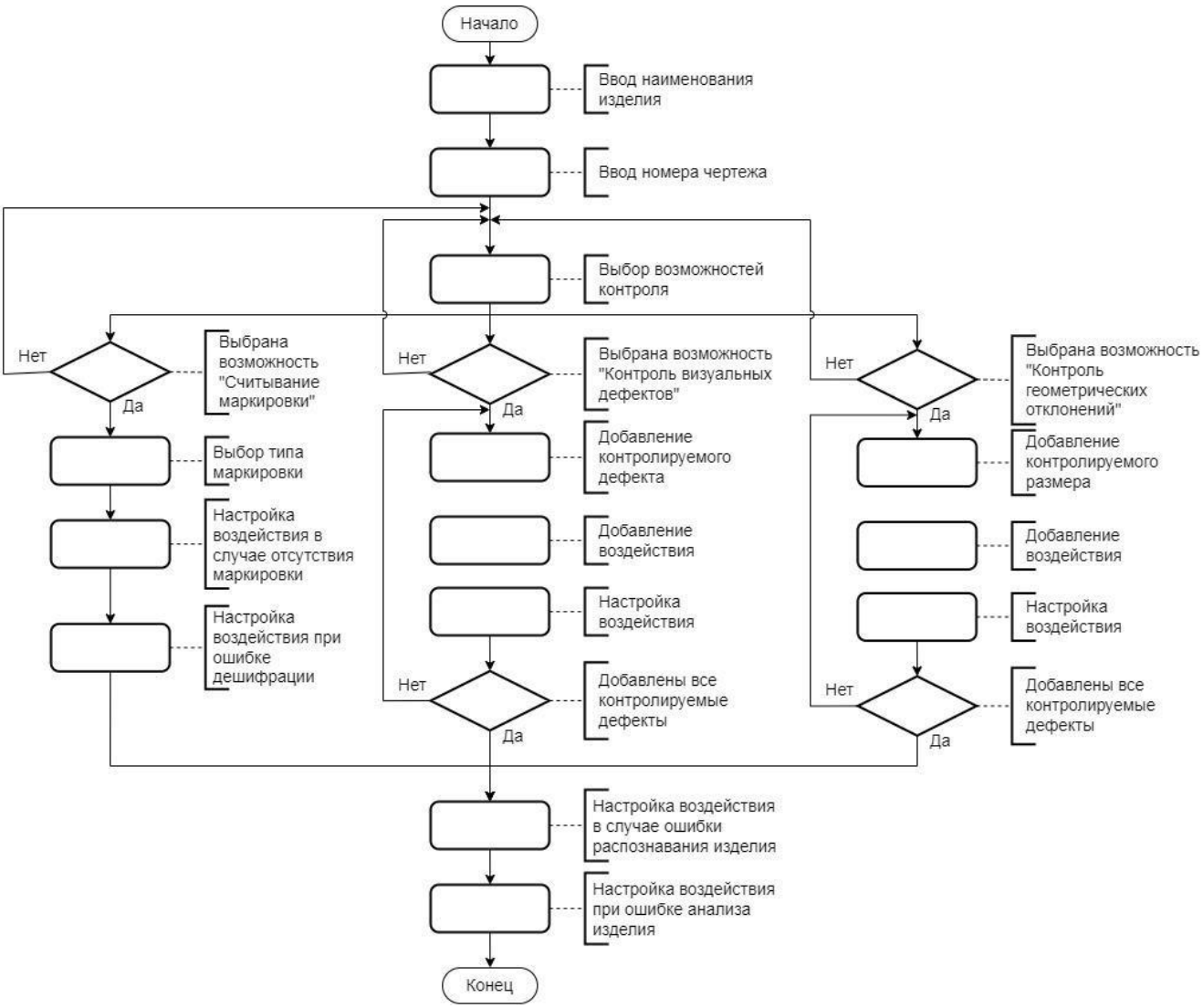


Рисунок 8 — Добавление изделия в библиотеку

Процесс настройки воздействия реализован в соответствии с алгоритмом, приведенном на рисунке Рисунок 9, и включает в себя:

- выбор устройства, с помощью которого воздействие будет реализовано;
- выбор конкретного выхода данного устройства;
- ввод задержки воздействия в мс;
- ввод длительности воздействия в мс;
- ввод количества повторений воздействия.

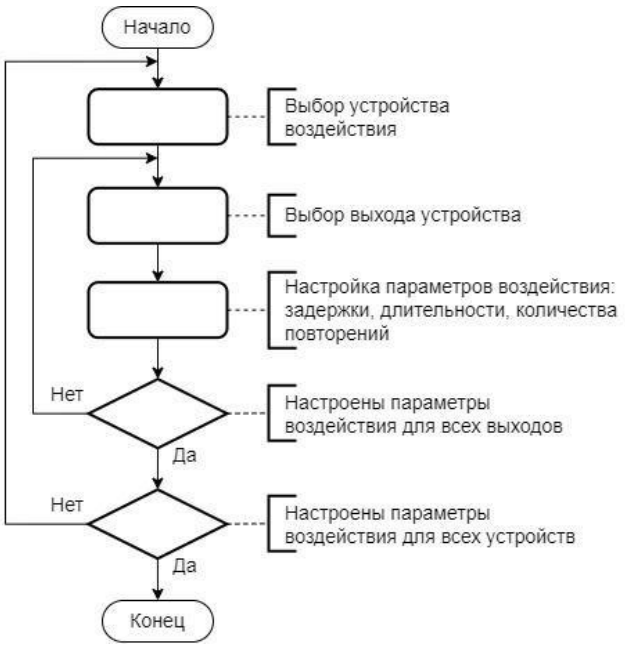


Рисунок 9 — Алгоритм настройки воздействия

3.3. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС ПРОГРАММЫ

3.3.1. Ролевая модель

В рамках комплекса реализована ролевая модель, включающая в себя роли «администратор» и «оператор».

Для пользователя с ролью «администратор» обеспечена возможность изменения всех параметров системы через панель администратора, а также обеспечен доступ к модулю визуализации.

Для пользователя с ролью «оператор» обеспечен доступ к панели оператора и отчетности, а также обеспечена возможность ввода ряда установочных параметров продукции.

Общим для всех пользователей системы является интерфейс входа (рисунок 10).

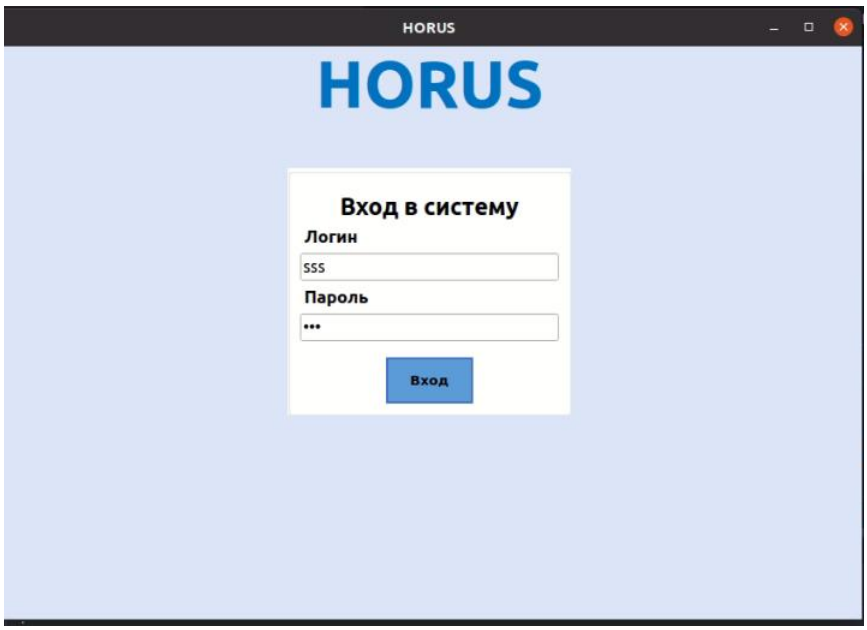


Рисунок 10 — Экран входа

3.3.2. Экран администратора

В рамках данного модуля разработана логическая модель, которая позволит пользователю с ролью «администратор» настраивать реакцию на внешние и внутренние события. Под реакцией в данной случае подразумевается настройка управляющего действия на конкретном выходе или интерфейсе.

Модуль обладает следующим функционалом:

- управление данными о пользователях (добавление новых пользователей, редактирование и удаление информации о зарегистрированных пользователях);
- добавление и настройка устройств (2D и 3D камеры, устройства управления);
- настройка типов считываемой маркировки изделий;
- настройка типов контролируемых дефектов;
- настройка типов контролируемых размеров;
- настройка портов и логики управления внешними устройствами.

Интерфейс модуля представляет собой набор экранов, каждый из которых обеспечивает необходимый функционал системы (рисунки 11-20). Доступ к модулю имеет только пользователь с ролью «администратор».

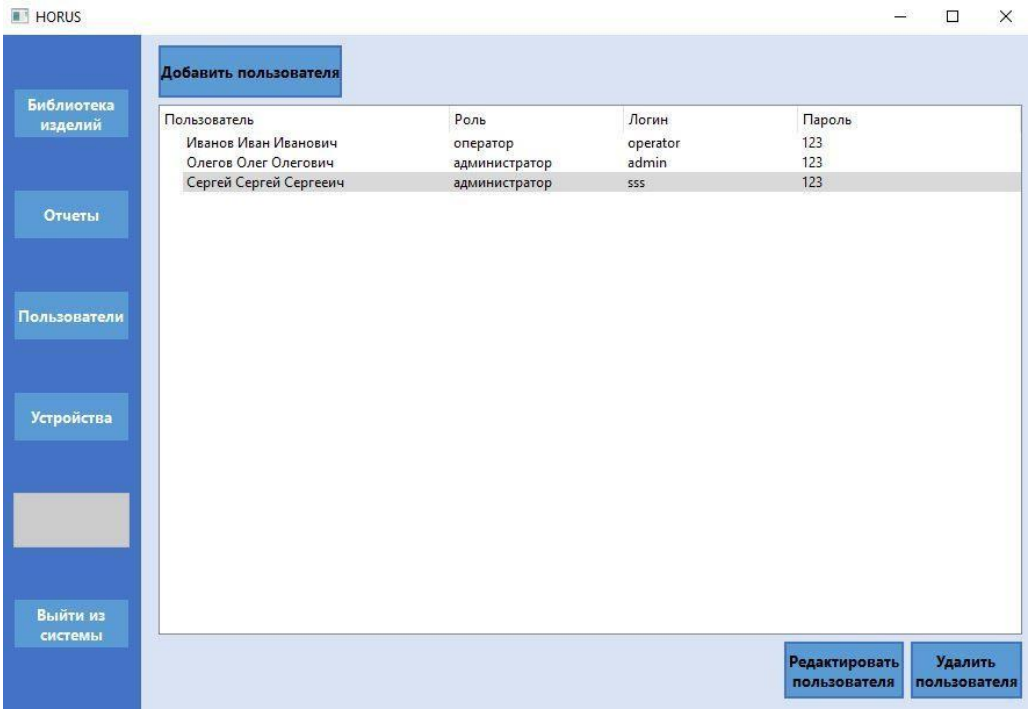


Рисунок 11 — Экран вкладки «Пользователи»

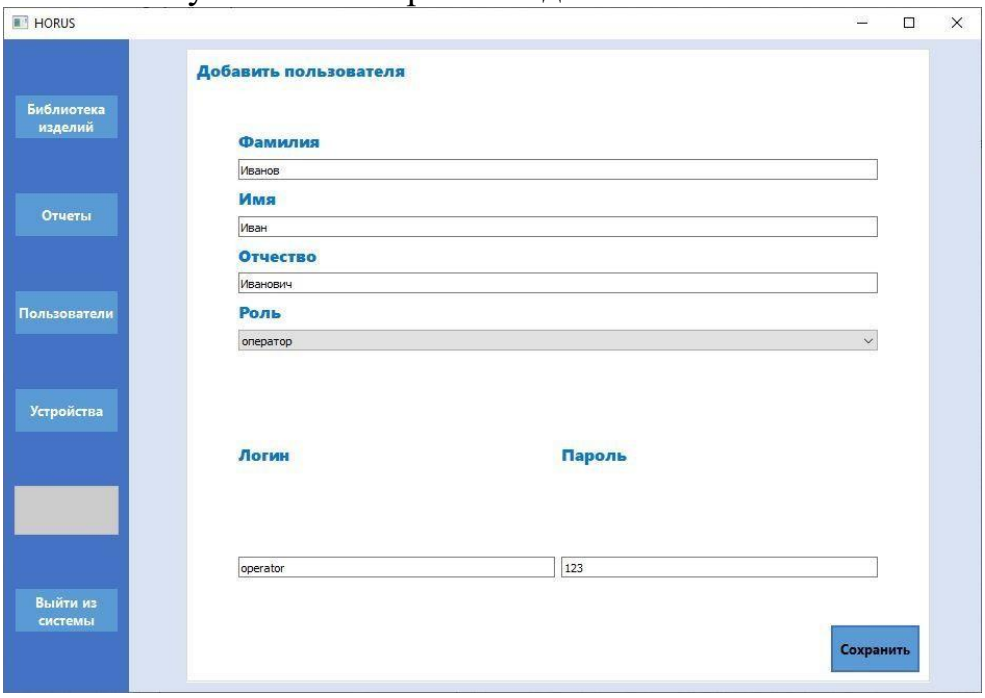


Рисунок 12 — Экран «Добавление пользователя»

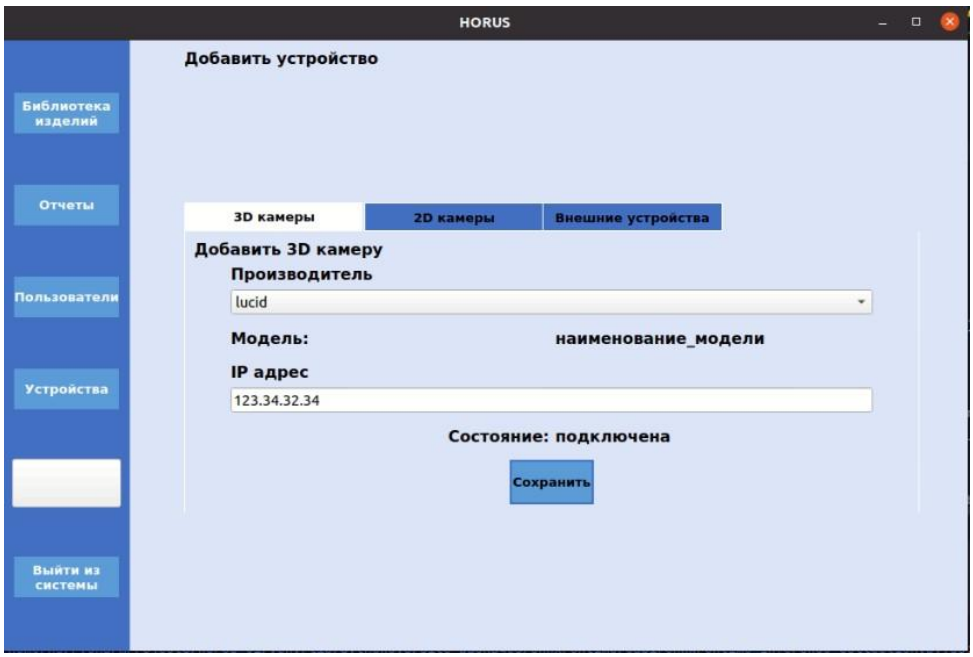


Рисунок 13 — Экран «Добавление устройств», вкладка «3D камеры»

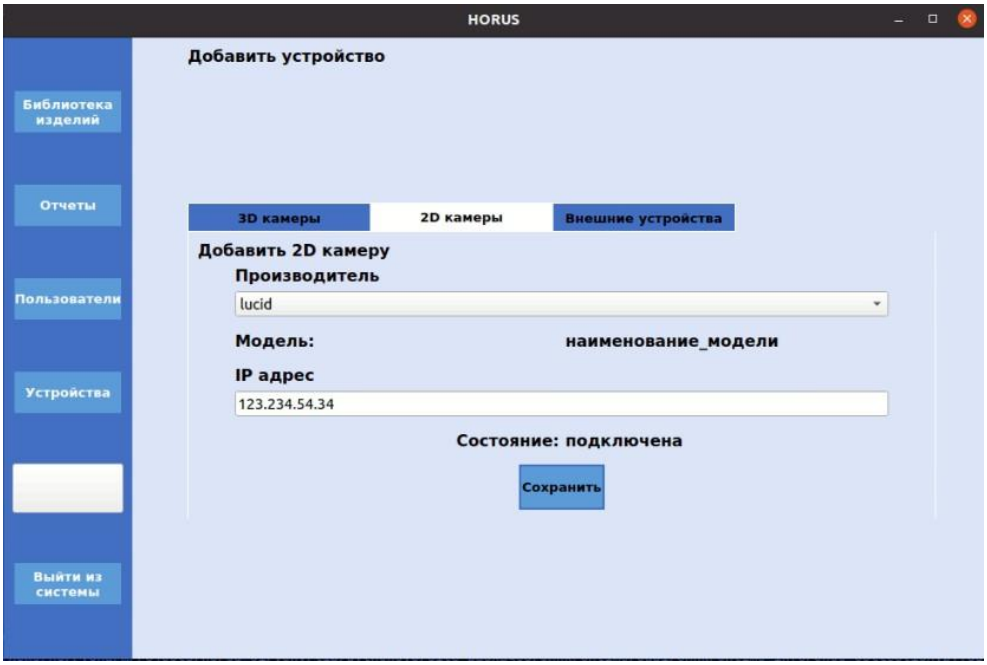


Рисунок 14 — Экран «Добавление устройств», вкладка «2D камеры»

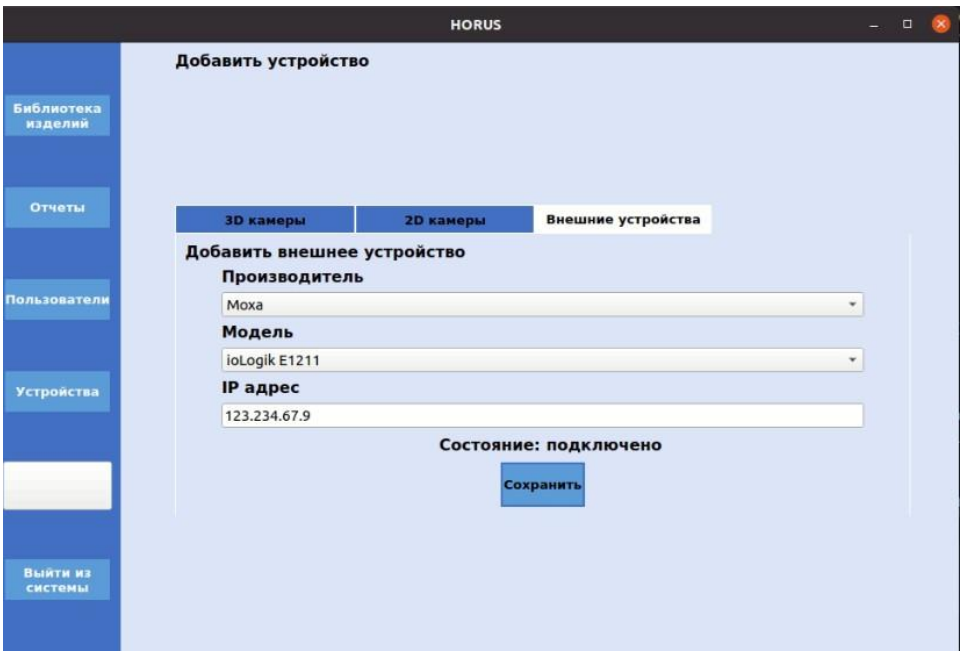


Рисунок 15 — Экран «Добавление устройств», вкладка «Внешние устройства»

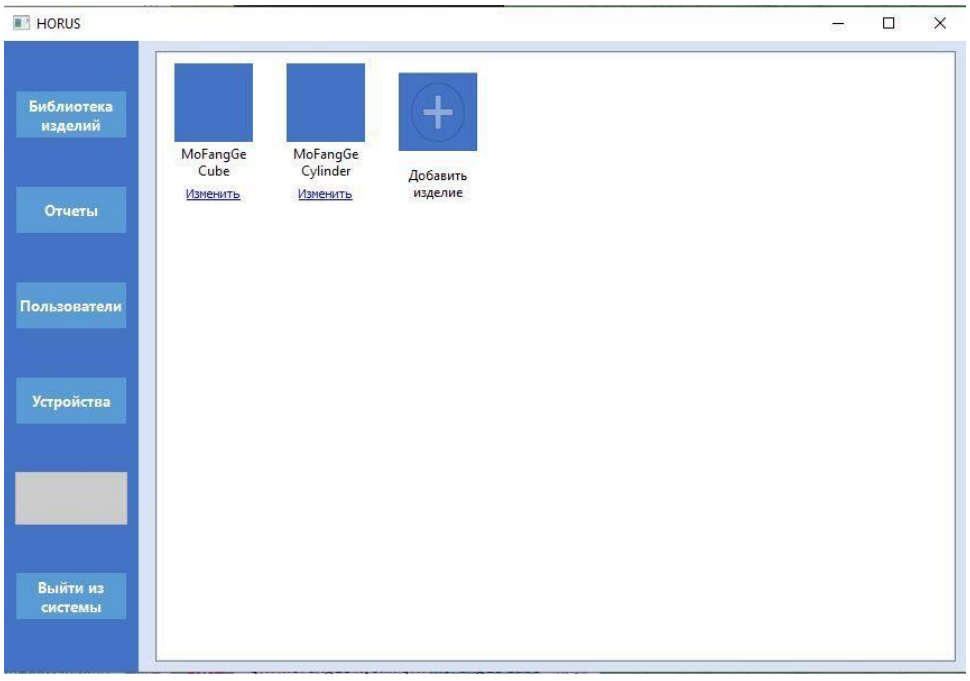


Рисунок 16 — Экран библиотеки изделий

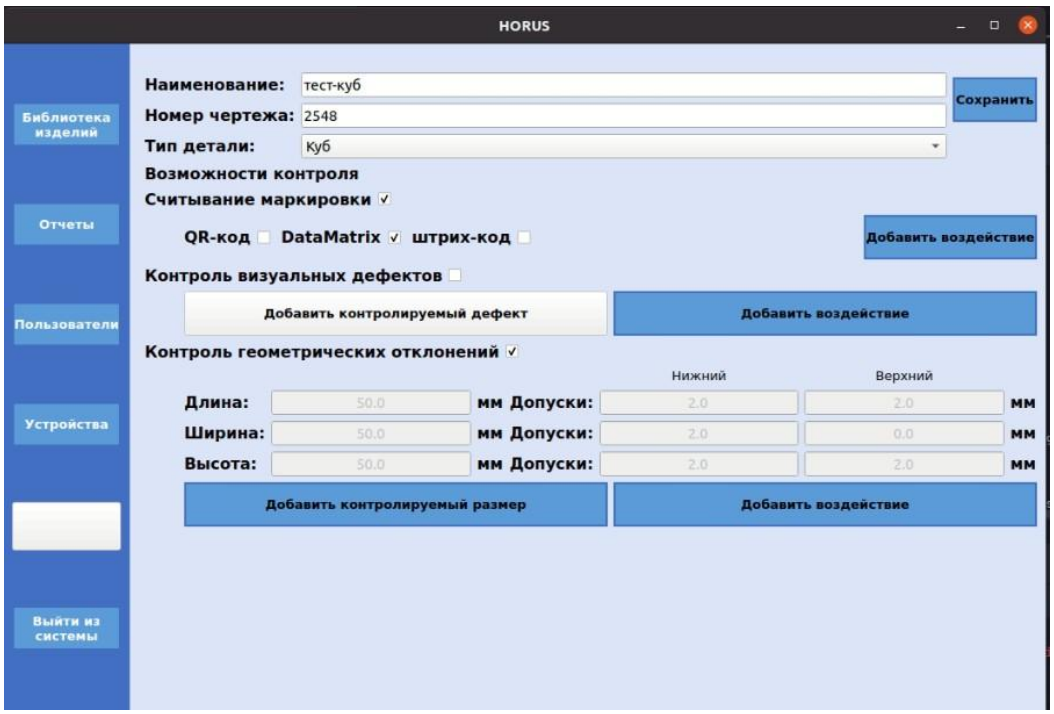


Рисунок 17 — Экран карточки изделия типа «куб»

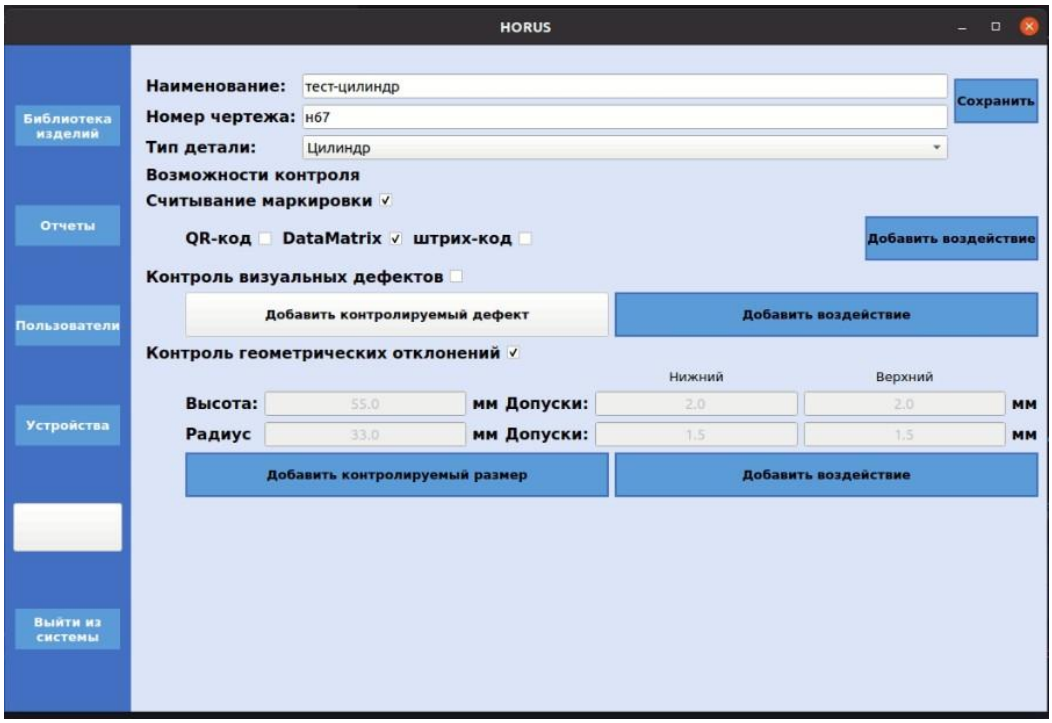


Рисунок 18 — Экран карточки изделия типа «цилиндр»

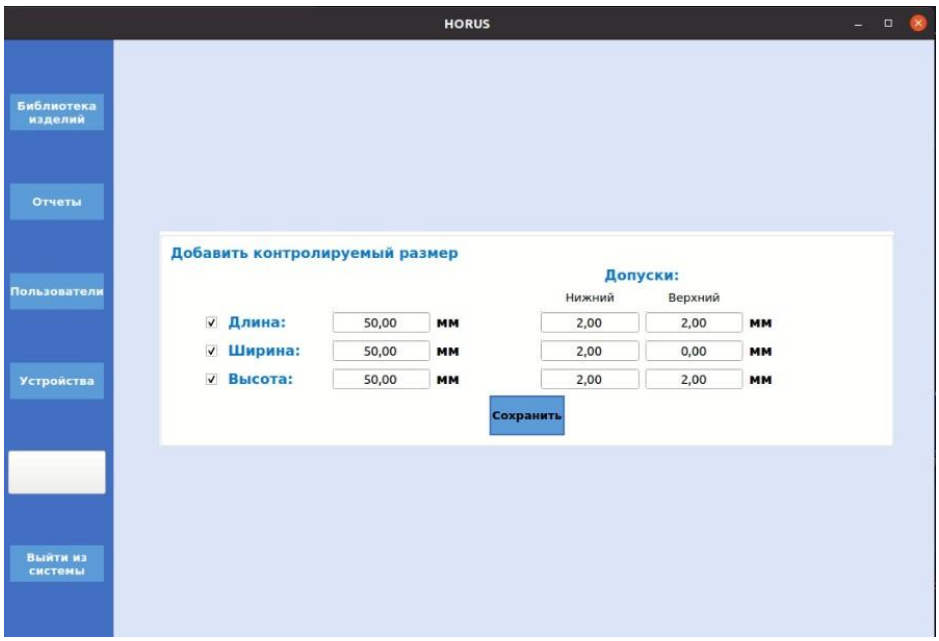


Рисунок 19 — Экран «Добавление контролируемого размера»

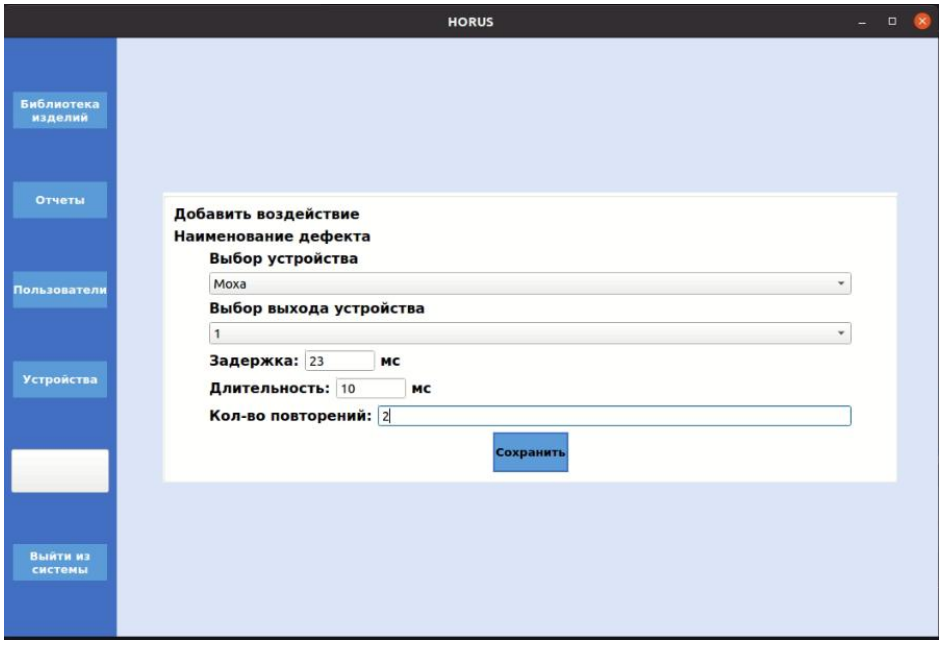


Рисунок 20 — Экран «Добавление воздействия»

3.3.3. Экран оператора

В рамках данного модуля разработана логическая модель, которая обеспечивает пользователю с ролью «оператор» возможность ввода ряда установочных параметров, возможность отслеживания промежуточных результатов текущей сессии, а также возможность формирования отчетности. Важным условием

является отсутствие у пользователя с ролью «оператор» доступа к другим модулям программы, за исключением панели оператора и панели отчетности.

Интерфейс модуля представляет собой набор экранов, каждый из которых обеспечивает необходимый функционал системы (рисунки 21, 22).

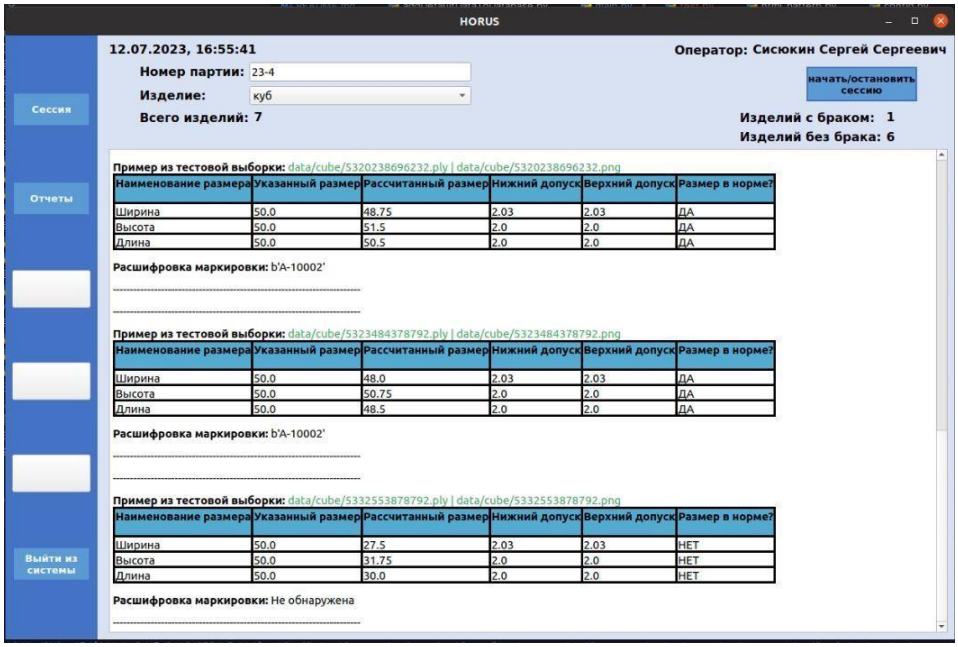


Рисунок 21 — Экран рабочей сессии

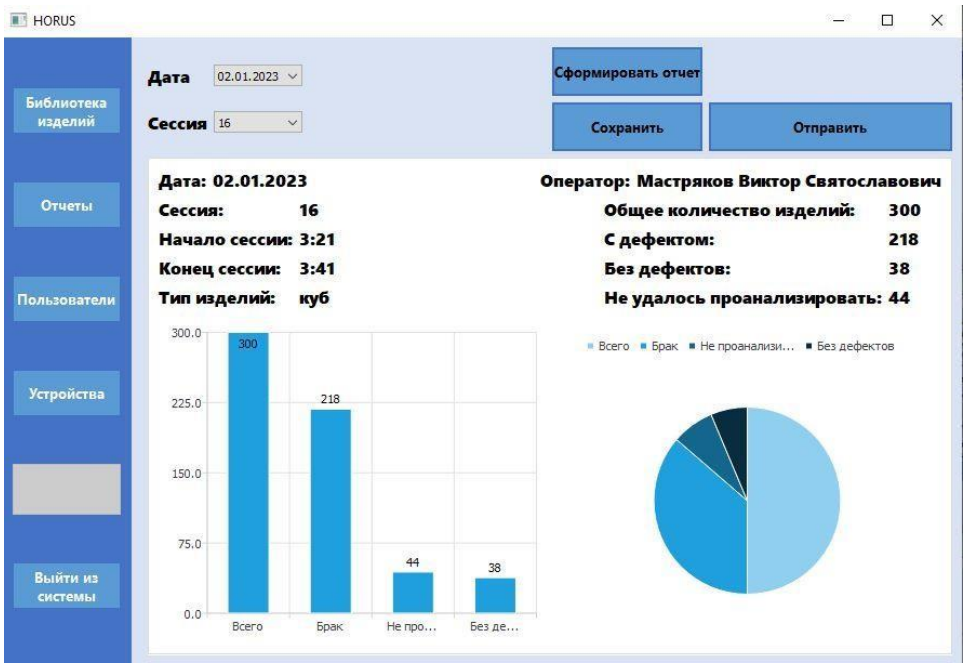


Рисунок 22 — Экран отчетности

4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Для функционирования программы МВЦУБ требуется ПЭВМ со следующей конфигурацией:

- ОС Ubuntu 20.04;
- Процессор i5 и выше, с частотой от 2.5ГГц и выше, с 4 ядрами и выше;
- Оперативная память - 8ГБ;
- SSD - 1ТБ;
- Видеокарта Nvidia с CUDA 4 ГБ;
- Дисплей 21 дюйм;
- Установленный python3.8;
- Установленный Docker Engine;
- Установленная библиотека работы с маркировкой libdmtx-dev;
- Наличие устройств ввода (клавиатура, сенсорный экран/манипулятор «мышь»).

5. ВЫЗОВ, ЗАГРУЗКА, ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ.

5.1. Вызов и загрузка

Вызов программы осуществляется через консоль с помощью команды «./venv/bin/python main.py».

5.2. Входные данные

Входными данными программы является облако точек и фотография, описывающее сцену конвейера в момент времени;

5.3. Выходные данные

Выходными данными программы являются количественные характеристики прошедших сессий:

- *Общее количество прошедших контроль изделий*;
- *Количество изделий с браком*;
- *Количество изделий без брака*;
- *Количество изделий, которые не удалось проанализировать*;
- *ФИО оператора, проводившего сессию*;
- *Время начала и окончания сессии*;
- *Наименование изделия*.