УТВЕРЖДЕН

**\_\_\_\_\_\_\_\_\_\_\_\_**

**Автоматизированная система   
“Испытательный стенд автоматизации и интеллектуального контроля ручных операций”**

**Документация разработчика**

Москва

2024

**АННОТАЦИЯ**

Настоящий документ описывает функциональную структуру АС, функции и задачи ее ключевых компонент. Приведена схема функциональной структуры, описана логика работы системы на уровне взаимодействия её компонент. Особое внимание уделено программным подсистемам, т.к. именно они реализуют основные функций АС — автоматический контроль ручных операций.

**Содержание**

[1 Информационные связи между элементами системы и связи системы с внешней средой 5](#_Toc137118431)

[2 Элементы функциональной структуры 5](#_Toc137118432)

[3 Функции и задачи подсистем АС 8](#_Toc137118433)

[3.1 Программная подсистема сбора, обработки и сохранения данных 8](#_Toc137118434)

[3.2 Программная подсистема хранения данных 9](#_Toc137118435)

[3.3 Программная подсистема отрисовки вспомогательных элементов 9](#_Toc137118436)

[3.4 Программная подсистема обработки входного потока данных 10](#_Toc137118437)

[3.5 Аппаратная компьютерная платформа 10](#_Toc137118438)

[3.6 Рабочее поле стола и станина 11](#_Toc137118439)

[3.7 Видеокамеры и датчики 12](#_Toc137118440)

[3.8 Информационное оборудование и осветительные приборы 12](#_Toc137118441)

## Информационные связи между элементами системы и связи системы с внешней средой

На диаграмме, представленной ниже, отображены связи между элементами системы:

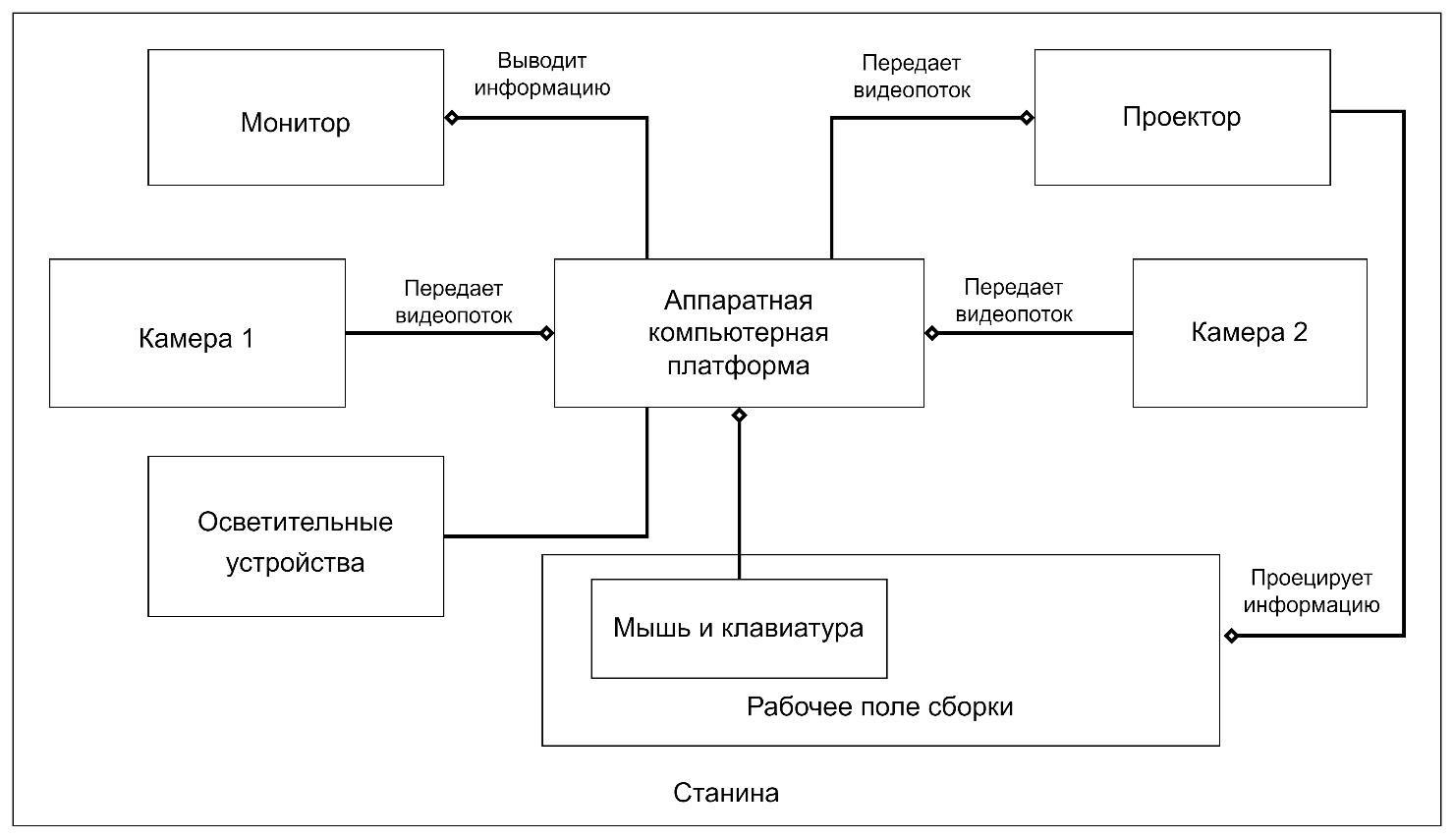


Рисунок 1 – Связи между элементами системы

## Основные элементы функциональной структуры

Логика работы и ключевой функционал системы реализована в виде программного обеспечения, работающего на компьютерной аппаратной платформе. Входные данные, как показано на рисунках 1 и 2, поступают от видео камер.

Также на вход программной части системы поступают данные из конфигурационных файлов, данные файлов с моделями ИИ.

На выходе комплекса программных подсистем:

* видео данные с изображением зоны сборки;
* данные с визуальной разметкой и подсказками для сборщика, выводимые на проектор, для визуализации процесса на рабочей зоне;
* записанное видео для последующей обработки, разметки и оценки эффективности работы стенда и оператора;
* текущие данные о работе стенда (опционально);
* итоговые данные о работе стенда с временными метками наступления событий различного рода.

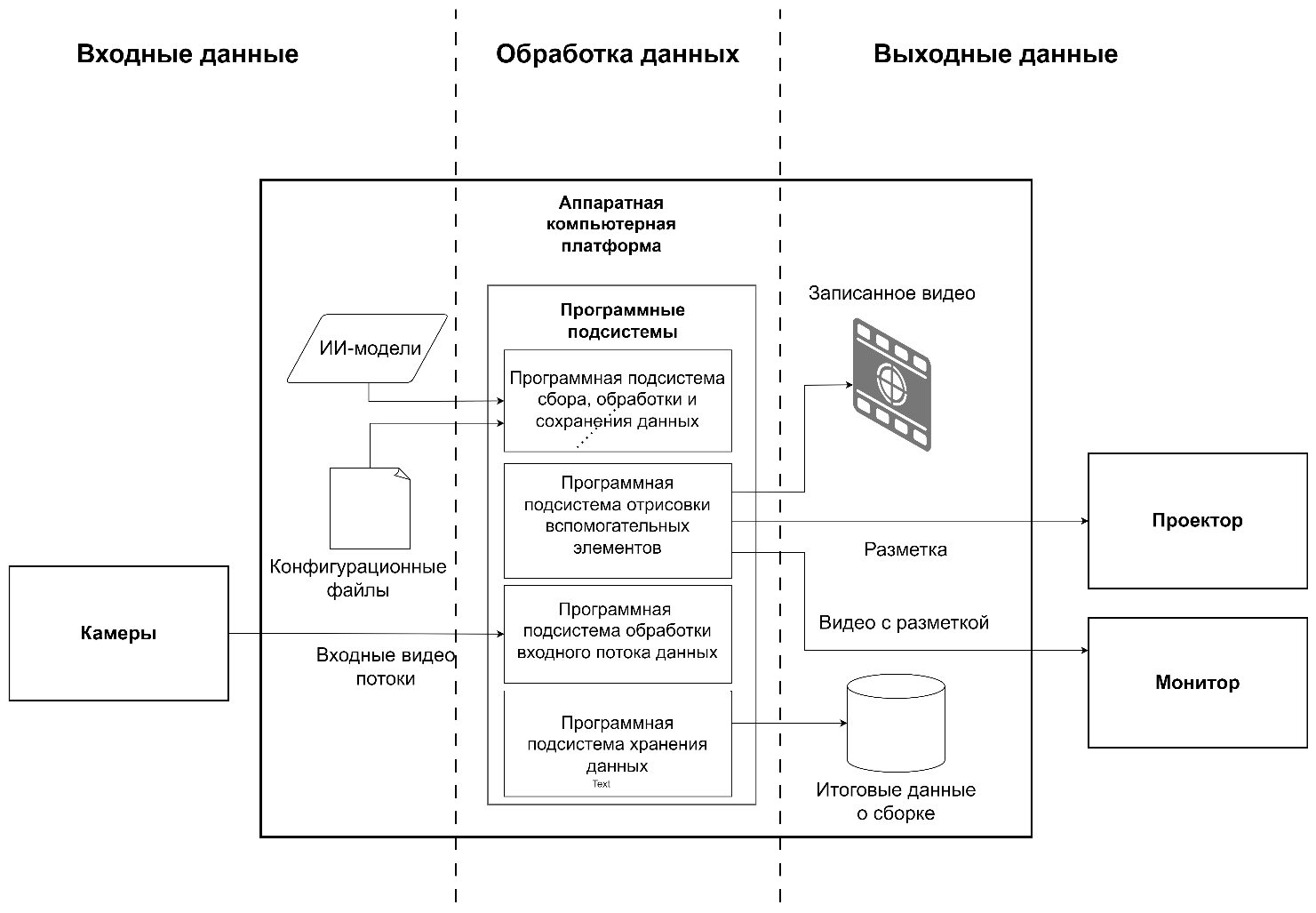


Рисунок 2 – Схема функциональной структуры взаимодействия подсистем, входной и выходной информации

В целом, в составе Системы выделяются следующие функциональные подсистемы:

1. программная подсистема сбора и предварительной обработки данных - предназначена для реализации процессов сбора данных (фотографий деталей) и приведения указанных данных к виду, необходимому для наполнения подсистемы хранения данных (датасетов для обучения моделей);
2. программная подсистема хранения данных - предназначена для хранения данных в структурах, предназначенных для обучения и использования моделей компьютерного зрения;
3. программная подсистема отрисовки вспомогательных элементов - предназначена для отображения таких визуальных элементов как границы рабочей области, ограничительные рамки основной области сборки, а также областей хранения деталей в рабочей области для поддержки оператора;
4. программная подсистема обработки входного потока данных - предназначена для анализа действий оператора-сборщика в рабочей области с целью выявления ошибок сборки, а также предоставления соответствующих подсказок/сопроводительной информации для поддержки нормального (удовлетворяющего всем правилам техники безопасности и любым другим техническим требованиям на производстве) процесса сборки
5. аппаратная компьютерная платформа;
6. рабочее поле (сборки) и станина;
7. видеокамеры и датчики.

## Функции и задачи подсистем АС

Для каждой подсистемы приведем перечень выполняемых ею функций и задач:

### Программная подсистема сбора, обработки и сохранения данных

Таблица 1 – Программная подсистема сбора, обработки и сохранения данных

|  |  |
| --- | --- |
| **Функция** | **Задачи** |
| Выполнение процессов сбора, обработки и загрузки данных | 1. Сбор данных (съемка всех деталей в различных положениях) 2. Обработка данных (разметка данных, создание искусственных датасетов для обучения моделей компьютерного зрения) 3. Загрузка данных в область временного/постоянного хранения |
| Протоколирует результаты сбора, обработки и загрузки данных | 1. Ведение журналов результатов сбора, обработки и загрузки данных 2. Предоставление информации о процессе работы подсистемы пользователям |

### Программная подсистема хранения данных

Таблица 2 – Программная подсистема хранения данных

|  |  |
| --- | --- |
| **Функция** | **Задачи** |
| Запись, хранения и модификация данных | 1. Сохранение значений ранее загруженных данных в случае их изменения 2. Составление файлов JSON для дальнейшей работы с данными |

### Программная подсистема отрисовки вспомогательных элементов

Таблица 3 – Программная подсистема отрисовки вспомогательных элементов

|  |  |
| --- | --- |
| **Функция** | **Задачи** |
| Хранение информации о всех вспомогательных визуальных элементах | Хранение информации о цвете, размере, местоположении на экране, толщине линии и других характеристиках граничных прямоугольников для рабочей области и областей хранения деталей |
| Анализ и расчет характеристик каждого сопроводительного элемента | 1. Сбор данных о текущих размерах и положении рабочей области в кадре камеры 2. Вычисление корректных координат каждого вспомогательного элемента для отображения в рабочей области 3. Своевременное обновление характеристик |
| Визуализация сопроводительных элементов поддержки | 1. Вывод элементов на мониторе вычислительной машины 2. Вывод элементов в рабочей области благодаря проектору |

### Программная подсистема обработки входного потока данных

Таблица 4 – Программная подсистема обработки входного потока данных

|  |  |
| --- | --- |
| **Функция** | **Задачи** |
| Обработка данных с камер | 1. Сбор входного видеопотока с камер высокого разрешения 2. Передача каждого кадра в модель компьютерного зрения для обработки и получения характеристик деталей и рабочей области |
| Обработка данных с внешних устройств | Обработка внешних сигналов пользователя, переданных системе от клавиатуры и мыши |

### Аппаратная компьютерная платформа

Таблица 5 – Аппаратная компьютерная платформа

|  |  |
| --- | --- |
| **Функция** | **Задачи** |
| Запуск и работа всех программных подсистем | Запуск, функционирование всех программных подсистем |
| Аппаратное сопряжение с сетевой инфраструктурой (при необходимости) | Подключение к информационно-вычислительной сети (по умолчанию, по интерфейсу UTP/Ethernet) |
| Подключение другого оборудования стенда | Подключение видеокамер и др. устройств |

### Рабочее поле стола и станина

Таблица 6 – Рабочее поле стола и станина

|  |  |
| --- | --- |
| **Функция** | **Задачи** |
| Пространство для работы с деталями и объектом сборки | Создание пространства на рабочем столе для расположения деталей сборки и работы с ними |
| Несущая функция для расположенных на станине и рабочей зоне объектов | Обеспечение стабильности непоколебимости и стабильности закрепленных и расположенных  на станине частей стенда: камер, осветительных устройств, деталей и т.п. |
| Место размещения оператора | Стул, с опорой для спины |

### Видеокамеры и датчики

Таблица 7 – Рабочее поле стола и станина

|  |  |
| --- | --- |
| **Функция** | **Задачи** |
| Получение последовательностей изображений (видео) рабочей зоны | Получение (съемка) последовательностей изображений (видео) рабочей зоны оператора в режиме реального времени |
| Передача данных на вычислительную аппаратную платформу | Передача видео данных на вычислительную аппаратную платформу в режиме реального времени |

### Информационное оборудование и осветительные приборы

Таблица 8 – Рабочее поле стола и станина

|  |  |
| --- | --- |
| **Функция** | **Задачи** |
| Освещение области сборки | Освещение области сборки до заданных значений |
| Отображение (проецирование) данных на рабочую область  сборки | Отображение информации в режиме реального времени на область сборки: разметка зоны сборки, информирование о ходе сборки и ошибках и т.п. |
| Отображение (визуализация) информации | Отображение информации в режиме реального времени на экране монитора |
| Озвучивание информации (опционально) | Аудио-информирование оператора  (по акустическому каналу ) о ходе сборки и др. событиях при работе стенда |

## **Таблица ключевых функций программы**

Таблица 1 – Ключевые функции программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название файла | Название функции / метода | Входные данные | Выходные данные | Описание |
| draw.py | set\_colors\_and\_description\_to\_default | **param** None | **return:** список цветов зон, список описаний зон | Установка цвета и описания зон по умолчанию |
| draw.py | init\_zones | **param** shifts: (x\_shift, y\_shift, table\_width, table\_height), сдвиги зон | **return:** два списка зон, для монитора и проектора | Первичная инициализация зон, применение сдвигов |
| draw.py | fill\_zone\_rectangle | **param** array: полотно, на котором отрисовываются прямоугольники **param** zone: зона для отрисовки  **param** color: цвет зоны  **param** thickness: толщина сторон прямоугольника | **return:** None | Добавление зоны (прямоугольника) нужного цвета и добавление к нему подписи, если она есть |
| draw.py | stage\_tracking | param stream: входной кадр  **param** stage\_number: номер stage, который хотим задетектировать  **param** stages\_model\_load: модель детекции соединений (stages)  **param** current\_st\_sequence: текущая последовательность детектирования stages | **return:** (флаг найден ли необходимый stage, сообщение для печати на экран) | Трекинг соединений (stages) |
| draw.py | one\_object\_display | **param** array: полотно для вывода  **param** cur\_obj: координаты задетектированного объекта  **param** des: название задетектированного объекта | **return:** None | Отрисовка прямоугольника задетектированного объекта |
| draw.py | is\_rectangle\_in\_inside | **param** zone\_for\_check: зона  **param** obj: прямоугольник, внутки которого находится объект | **return:** bool (True, если объект внутри зоны, иначе False) | Проверка находится ли объект в зоне |
| draw.py | update\_all\_zones | **param** array: полотно, на котором отрисовываются прямоугольники **param** zone\_list: список всех зон **param** current\_col\_dict: цвета каждой зоны  **param** current\_desc\_list: подпись к каждой зоне | **return:** None | Обновление цветов и подписей у всех зон |
| draw.py | display\_to\_message\_box | **param** array: полотно, куда выводить текст  **param** text\_to\_print: список словарей {ключ: значение}, где значение -- цвет надписи, ключ, сама надпись  **param** vertical\_text\_shift: для вывода в столбик, расстояние между соседними строчками **param** font\_size: размер шрифта  **param** thickness: толщина шрифта  **param** x0: расположение текста, x координата начала фразы  **param** y0: расположение текста, y координата начала фразы | **return:** None | Вывод сообщений на экран |
| draw.py | usecase\_step\_0\_detail\_in\_zone | **param** obj: прямоугольник, внутки которого находится объект  **param** detail\_name: имя детали, координаты положение которой будет происходить проверка  **param** target\_zone: зона, в которой должна находиться данная деталь  **param** target\_detail\_number: номер детали, которую проверяем  **param** cur\_matrix: текущая матрица расположения всех объектов по зонам  **param** messages: сообщения, которые отрисовываются под зонами и отражают текущее состояние выполнение кейса | **return:** (current\_matrix, messages) - текущая матрица, сообщения | Говорит находится ли задектированный объект в нужной зоне или нет |
| draw.py | usecase\_zone\_tracker | **param** array: массив для вывода на монитор  **param** array\_for\_projector: массив для вывода на проектор  **param** to\_do: инструкции, что нужно сделать на текущем шаге  **param** step\_number: номер шага  **param** target\_matrix: целевая матрица расположения деталей по зонам (-1 -- не следим за деталью)  **param** current\_matrix: текущая матрица расположения деталей по зонам  **param** orig\_stream: входной исходный кадр для детекции, изображение без дополнительных надписей  **param** stages\_model: модель для детектирования соединений между деталями (stages)  **param** stages\_needed\_flag: флаг необходимости детектирования соединений (stages) на данном этапе  **param** current\_color\_list: текущий набор цветов зон  **param** current\_description\_list: текущий набор подписей зон  **param** step\_list\_flags: список флагов этапов (отработал этап или нет)  **param** list\_of\_zones: зоны, отображающиеся на мониторе  **param** list\_of\_zones\_projector: зоны, отображающиеся на проекторе  **param** target\_detail\_places: исходная раскладка деталей (связка название детали -- номер детали)  **param** stage\_number: какое содинение проверяется на текущем этапе (номер stage, если -1 -- stage не нужен) | **return:** None | Основная функция реализации usecase |
| draw.py | display\_objects | **param** array: полотно для вывода найденных объектов на экран  **param** item: задектированные нейросетью объекты  **param** all\_found\_objects\_description: список названий объектов  **param** debug\_mode\_flag: флаг режима дебага (по умолчанию False) | **return:** None | Перебор и вывод задетектированных объектов |
| draw.py | usecase\_hands\_tracking | **param** array: полотно для вывода результатов  **param** show\_connections: флаг отрисовки точек и связей на кистях рук | **return:** None | Usecase, который работает параллельно с основной последовательностью сборки, отвечает за слежение за количеством рук в кадре (детектирование постороннего человека в кадре) |
| image\_captor.py | update | **param**  self | **return:** None | Захват текущего кадра с камеры |
| image\_captor.py | get\_image | **param** self | **return:** None | Получение изображения и его поворот |
| window.py | show | **param** self | **return:** None | Отрисовка изображения на экране |
| window.py | set\_image | **param** self | **return:** None | Установка изображения, которое будет выведено на экран и его вывод |
| main\_workflow.py | download\_file | **param** url: ссылка на веса  **param** path: путь для сохранения весов  **param** url: имя файла с весами | **return:** None | Скачивание предобученных весов модели по указанной ссылке |
| main\_workflow.py | main | — | — | Отвечает за запуск программы в многопоточном режиме |
| seqmark.py | — | — | — | Модуль, предназначенный для анализа последовательности задетектированных соединений для уменьшения числа ложных срабатываний |