

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	4
2	Элементы функциональной структуры	4
3	Функции и задачи подсистем АС	7
3.1	Программная подсистема сбора, обработки и сохранения данных ...	7
3.2	Программная подсистема хранения данных	8
3.3	Программная подсистема отрисовки вспомогательных элементов ...	8
3.4	Программная подсистема обработки входного потока данных	9
3.5	Аппаратная компьютерная платформа.....	9
3.6	Рабочее поле стола и станина	10
3.7	Видеокамеры и датчики	11
3.8	Информационное оборудование и осветительные приборы.....	11

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

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

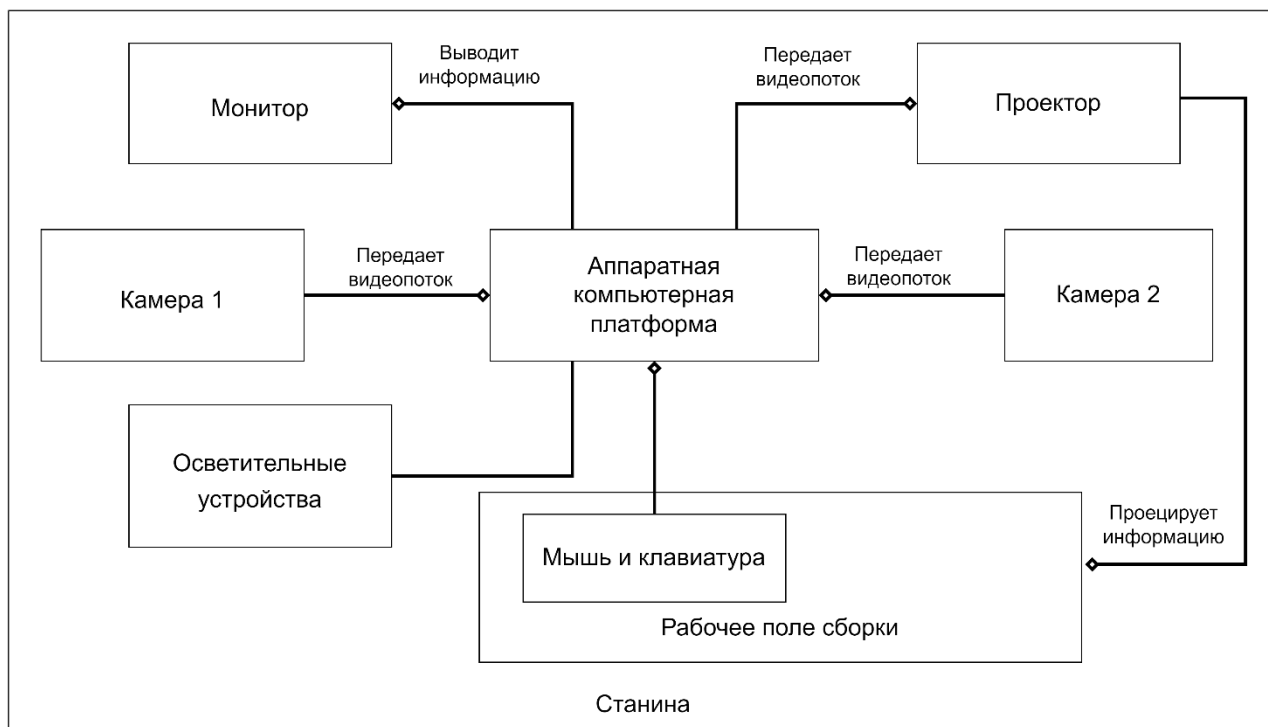


Рисунок 1 – Связи между элементами системы

2 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 станда и оператора;
- текущие данные о работе станда (опционально);
- итоговые данные о работе станда с временными метками наступления событий различного рода.

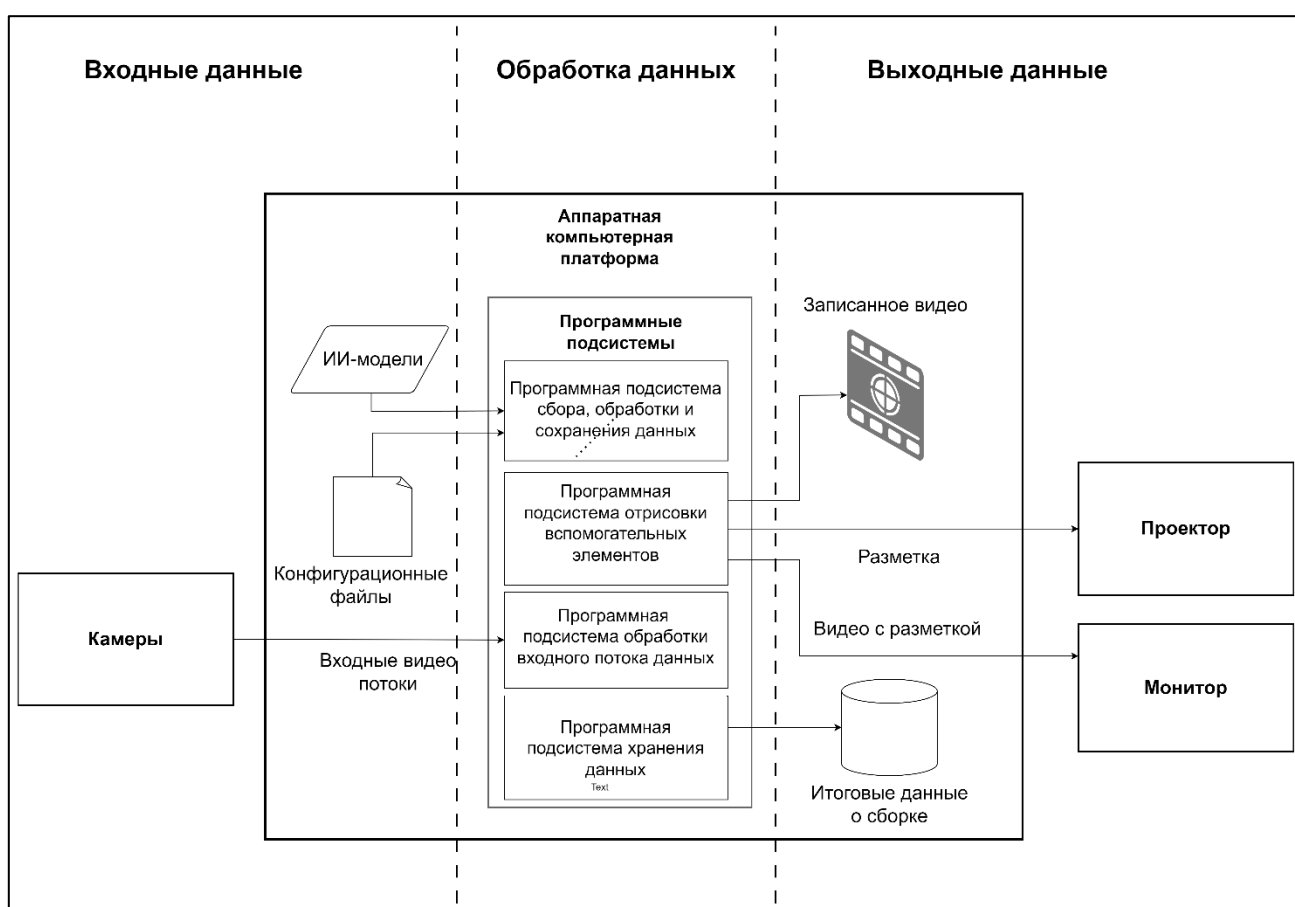


Рисунок 2 – Схема функциональной структуры взаимодействия подсистем, входной и выходной информации

В целом, в составе Системы выделяются следующие функциональные подсистемы:

- 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);
- программная подсистема хранения данных - предназначена для хранения данных в структурах, предназначенных для обучения и использования моделей компьютерного зрения;
- 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;
- 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
- аппаратная компьютерная платформа;
- рабочее поле (сборки) и станина;
- видеокамеры и датчики.

3 Функции и задачи подсистем АС

Для каждой подсистемы приведем перечень выполняемых ею функций и задач:

3.1 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	<ol style="list-style-type: none">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	<ol style="list-style-type: none">1. Ведение журналов результатов сбора, обработки и загрузки данных2. Предоставление информации о процессе работы подсистемы пользователям

3.2 Программная подсистема хранения данных

Таблица 2 – Программная подсистема хранения данных

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

3.3 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	<ol style="list-style-type: none">1. Сбор данных о текущих размерах и положении рабочей области в кадре камеры2. Вычисление корректных координат каждого вспомогательного элемента для отображения в рабочей области

	3. Своевременное обновление характеристик
Визуализация сопроводительных элементов поддержки	Вывод элементов на мониторе вычислительной машины Вывод элементов в рабочей области благодаря проектору

3.4 Программная подсистема обработки входного потока данных

Таблица 4 – Программная подсистема обработки входного потока данных

Функция	Задачи
Обработка данных с камер	<ol style="list-style-type: none"> 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

3.5 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

3.6 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
----------------------------	--------------------------

3.7 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

3.8 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

4 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	return: None	Добавление зоны (прямоугольника) нужного цвета и

		param color: цвет зоны param thickness: толщина сторон прямоугольника		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, внутри 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	return: None	Вывод сообщений на экран

		<p>param x0: расположение текста, x координата начала фразы</p> <p>param y0: расположение текста, y координата начала фразы</p>		
draw.py	usecase_step_0 _detail_in_zone	<p>param obj: прямоугольник, внутри которого находится объект</p> <p>param detail_name: имя детали, координаты положение которой будет происходить проверка</p> <p>param target_zone: зона, в которой должна находиться данная деталь</p> <p>param target_detail_number: номер детали, которую проверяем</p> <p>param cur_matrix: текущая матрица расположения всех объектов по зонам</p> <p>param messages: сообщения, которые отрисовываются под зонами и отражают текущее состояние выполнение кейса</p>	<p>return: (current_matrix, messages) - текущая матрица, сообщения</p>	Говорит находится ли задетированный объект в нужной зоне или нет
draw.py	usecase_zone_tracker	<p>param array: массив для вывода на монитор</p> <p>param array_for_projector: массив для вывода на проектор</p> <p>param to_do: инструкции, что нужно сделать на текущем шаге</p> <p>param step_number: номер шага</p> <p>param target_matrix: целевая матрица расположения деталей по зонам (-1 -- не следим за деталью)</p> <p>param current_matrix: текущая матрица расположения деталей по зонам</p> <p>param orig_stream: входной исходный кадр для детекции, изображение без дополнительных надписей</p>	<p>return: None</p>	Основная функция реализации usecase

		<p>param stages_model: модель для детектирования соединений между деталями (stages)</p> <p>param stages_needed_flag: флаг необходимости детектирования соединений (stages) на данном этапе</p> <p>param current_color_list: текущий набор цветов зон</p> <p>param current_description_list: текущий набор подписей зон</p> <p>param step_list_flags: список флагов этапов (отработал этап или нет)</p> <p>param list_of_zones: зоны, отображающиеся на мониторе</p> <p>param list_of_zones_projector: зоны, отображающиеся на проекторе</p> <p>param target_detail_places: исходная раскладка деталей (связка название детали -- номер детали)</p> <p>param stage_number: какое содинение проверяется на текущем этапе (номер stage, если - 1 -- stage не нужен)</p>		
draw.py	display_objects	<p>param array: полотно для вывода найденных объектов на экран</p> <p>param item: задетированные нейросетью объекты</p> <p>param all_found_objects_description: список названий объектов</p> <p>param debug_mode_flag: флаг режима дебага (по умолчанию False)</p>	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
--	--	--	--	---