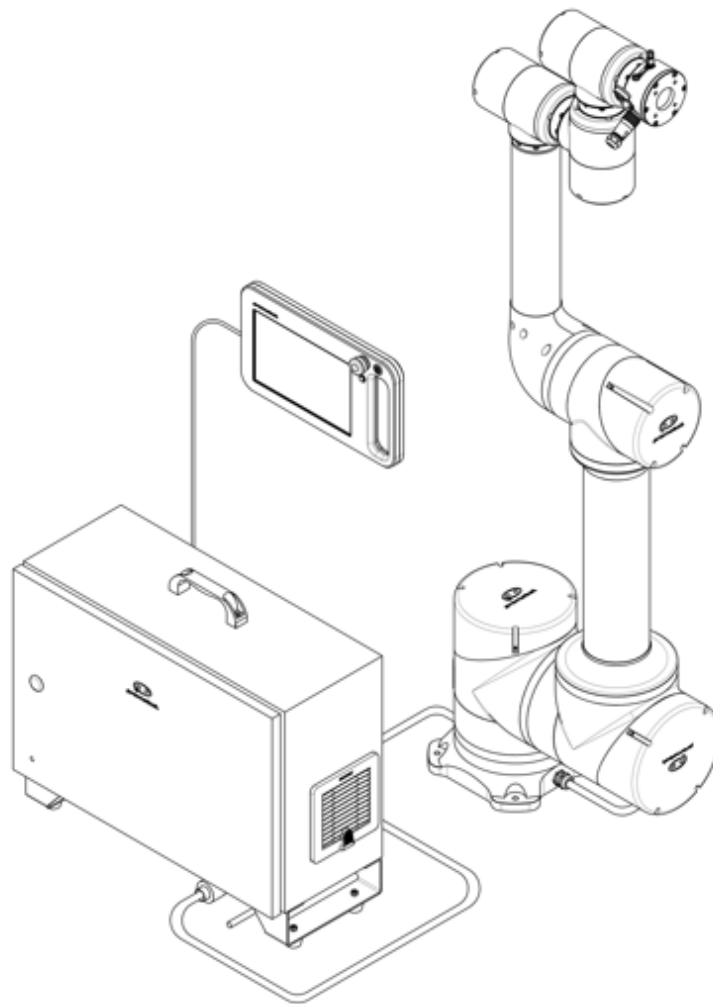


ПРОГРАММНО – АППАРАТНЫЙ КОМПЛЕКС PROMOBOT M13

Руководство по эксплуатации



Редакция: 27.08.2026

Версия релиза: 0.25.1

Информация для пользователя

Внешний вид комплекса и пользовательского интерфейса может отличаться от изображений, представленных в данном документе. Предприятие-изготовитель постоянно улучшает продукт, и данное руководство систематически актуализируется. Предприятие-изготовитель оставляет за собой право вносить изменения без снижения качества и технических характеристик.

Содержание

I. ЭКСПЛУАТАЦИЯ, ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ И БЕЗОПАСНОСТЬ	
PROMOBOT M13	I-4
1 ОПИСАНИЕ И РАБОТА	I-4
1.1 Описание и работа Promobot M13	I-4
1.2 Описание и работа составных частей Promobot M13	I-13
2 ИСПОЛЬЗОВАНИЕ ПО НАЗНАЧЕНИЮ	I-19
2.1 Эксплуатационные ограничения Promobot M13	I-19
2.2 Подготовка Promobot M13 к использованию	I-21
2.3 Использование Promobot M13	I-46
2.4 Демонтаж Promobot M13	I-55
3 ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ	I-57
3.1 Регламентное техническое обслуживание Promobot M13.....	I-57
3.2 Диагностика неисправностей Promobot M13.....	I-59
3.3 Рекомендации по ремонту Promobot M13.....	I-60
3.4 Хранение запасных частей Promobot M13.....	I-60
3.5 Регистрация обслуживания и ремонта Promobot M13.....	I-61
4 МЕРЫ БЕЗОПАСНОСТИ ПРИ ИСПОЛЬЗОВАНИИ	I-61
4.1 Общие требования безопасности	I-61
4.2 Требования к оператору.....	I-62
4.3 Рабочее окружение манипулятора.....	I-63
4.4 Действия в аварийных ситуациях	I-64
5 БЕЗОПАСНОСТЬ	I-66
5.1 Индикация Promobot M13.....	I-66
6 ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА.....	I-67
7 ХРАНЕНИЕ.....	I-69
7.1 Условия хранения Promobot M13	I-69
7.2 Подготовка к хранению Promobot M13	I-70
7.3 Периодическое обслуживание во время хранения Promobot M13	I-70
7.4 Подготовка к использованию после хранения Promobot M13	I-70
7.5 Хранение запасных частей и аксессуаров Promobot M13	I-70

8	ТРАНСПОРТИРОВАНИЕ	I-71
9	УТИЛИЗАЦИЯ	I-72
II.	ПРИЛОЖЕНИЕ PROMOVOT M CONTROL	II-1
1	ВВЕДЕНИЕ	II-1
2	АВТОРИЗАЦИЯ, ПОДКЛЮЧЕНИЕ И ВЫКЛЮЧЕНИЕ МАНИПУЛЯТОРА В ПРИЛОЖЕНИИ	II-2
3	РУЧНОЕ УПРАВЛЕНИЕ	II-6
4	СОЗДАНИЕ ПРОГРАММ	II-12
4.1	Blockly	II-14
4.2	Python	II-33
5	НАСТРОЙКА	II-34
5.1	Общие и системные настройки	II-35
5.2	Инструмент	II-36
5.3	Рабочая зона	II-38
5.4	Положение и пределы	II-41
5.5	Сервисные функции	II-45
6	МОНИТОРИНГ	II-49
7	ОБНОВЛЕНИЕ ПРИЛОЖЕНИЯ	II-56
III.	PROMOVOT M CONTROL SDK	III-1
1	Назначение и область применения	III-1
2	Архитектура и принципы работы SDK	III-2
2.1	Компоненты	III-2
2.2	Модель работы	III-3
3	SDK Python	III-3
3.1	Требования и подготовка среды Python	III-3
3.2	Получение и установка SDK	III-5
3.3	Настройка сети и первое подключение	III-7
3.4	Модель выполнения команд	III-9
3.5	Единицы измерения и структуры данных	III-11
3.6	Получение управления и первый сценарий	III-12
3.7	Движение манипулятора	III-13

3.8	Состояние и диагностика.....	III-18
3.9	ТСР и инструмент.....	III-19
3.10	GPIO, выходы и интерфейсы фланца	III-20
3.11	Запуск программ.....	III-23
3.12	Потоковое управление	III-25
3.13	Состояния и программный останов.....	III-26
3.14	События и подписки	III-28
3.15	Modbus RTU и RS-485.....	III-29
3.16	Дополнительные устройства	III-31
3.17	Обработка ошибок и поиск неисправностей	III-32
3.18	Корректное завершение программы.....	III-35
3.19	Карта методов по задачам	III-35
3.20	Поддерживаемый указатель методов M13.....	III-36
3.21	Алфавитный указатель.....	III-49
3.22	Контрольный список перед запуском	III-1
3.23	Минимальный шаблон приложения	III-1
4	SDK C++	III-3
4.1	Требования и использование C++	III-3
4.2	Типы команд	III-5
4.3	Программные блоки (структура программы).....	III-7
4.4	Заключение.....	III-25

Настоящее руководство по эксплуатации предназначено для ознакомления пользователей с устройством, принципами работы, правилами эксплуатации, технического обслуживания и ремонта программно-аппаратного комплекса Promobot M13.

Руководство содержит основные сведения о комплексе, инструкции по монтажу, демонтажу, использованию, обслуживанию, описание режимов работы, рекомендации по устранению неисправностей и правила безопасности.

Руководство по эксплуатации предназначено для следующих категорий пользователей:

- операторов, работающих непосредственно с Promobot M13;
- инженеров, отвечающих за настройку, программирование и диагностику Promobot M13;
- технического персонала, выполняющего техническое обслуживание, ремонт и проверку работоспособности Promobot M13.

Документ также может быть использован иными категориями пользователей для обучения персонала и в качестве справочного материала при работе с Promobot M13.

Функциональные возможности манипулятора нацелены на использование в следующих сферах промышленности: пищевой, химической, деревообрабатывающей, автомобилестроении, металлургии и металлообработке, других сферах.

Promobot M13 представляет собой программно-аппаратный комплекс, состоящий из манипулятора, шкафа управления и пульта управления. В состав комплекса также входят соединительные элементы и управляющее программное обеспечение.

Promobot M13 изготовлен по ТУ 28.99.39-006-40897141-2024. По электробезопасности Promobot M13 удовлетворяет требованиям аппаратуры 1 класса по ГОСТ IEC 60204-1.

Качество и безопасность Promobot M13 подтверждается сертификатами ТР ТС 004/2011 «О безопасности низковольтного оборудования», ТР ТС 010/2011 «О безопасности машин и оборудования», ТР ТС 020/2011 «Электромагнитная совместимость технических средств».

Сведения о производителе

Производитель: ООО «ПРОМОБОТ»

Юридический адрес: 614066, Россия, г. Пермь, Шоссе Космонавтов, стр.111А,
пом. 6

Фактический адрес: 614066, Россия, г. Пермь, Шоссе Космонавтов, стр.111А,
пом. 6

Телефон: +7 (342) 257-80-85

E-mail: info@promo-bot.ru

Сайт: <https://promo-bot.ru/>

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящем документе применяют следующие сокращения, обозначения и термины с соответствующими определениями:

Promobot M13 / Комплекс	– робот промышленный адаптивный с контурным управлением электромеханический Promobot M13
ПО	– программное обеспечение
Приложение Promobot M Control/ Приложение	– веб-интерфейс для управления, программирования и мониторинга работы манипулятора Promobot M13
ПУ	– пульт управления
СП	– сервопривод
ЦТИ	– центральная точка инструмента
ЧПУ	– числовое программное управление

I. ЭКСПЛУАТАЦИЯ, ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ И БЕЗОПАСНОСТЬ PROMOBOT M13

1 ОПИСАНИЕ И РАБОТА

1.1 Описание и работа Promobot M13

1.1.1 Назначение

Робот промышленный адаптивный с контурным управлением электромеханический Promobot M13 предназначен для выполнения задач автоматизации в промышленных и лабораторных условиях. Манипулятор с шестью степенями свободы обеспечивает высокоточное перемещение, позиционирование и обработку объектов в пространстве.

Promobot M13 интегрируется в условия рабочей среды с применением дополнительного оборудования и оснастки.

Области применения:

- сварка (перемещение сварочного инструмента по заданной траектории);
- перемещение готовой продукции, полуфабрикатов и товарно-материальных ценностей;
- перемещение заготовок и деталей (автоматизированная загрузка или выгрузка деталей в станки с ЧПУ, литейные машины, прессы; работа с тяжелыми заготовками);
- механическая обработка деталей: зачистка, полировка, шлифовка (использование абразивных инструментов для финишной обработки поверхностей и минимизация дефектов благодаря точному повторению профиля изделия), фрезеровка (выполнение черновой и чистовой обработки металлических или пластиковых заготовок);
- сканирование объектов (3D-сканирование для контроля геометрии деталей, выявления дефектов; интеграция с системами компьютерного зрения для автоматической коррекции производственных процессов; применение в реверс-инжиниринге и цифровом моделировании);

- нанесение жидкостей и материалов: клей, краска, герметик и прочее;
- использование в качестве образовательного оборудования для обучения студентов основам робототехники, программирования и автоматизации, для отработки реальных производственных сценариев в лабораториях;
- использование в качестве демонстрационного оборудования на выставках, конференциях, в шоурумах.

1.1.2 Технические характеристики

Основные технические характеристики Promobot M13 приведены в таблице 1.

Таблица 1 – Технические характеристики Promobot M13

Параметр	Значение
Основные характеристики	
Количество степеней свободы манипулятора	6
Грузоподъемность манипулятора, кг	13
Линейная скорость перемещения центральной точки инструмента (ЦТИ), м/с	1
Средняя повторяемость, не хуже, мм	0,05
Рабочий радиус от оси основания (рабочая зона), мм	от 150 до 1300
Рабочий диапазон суставов*, градус	±358
Степень защиты корпуса манипулятора	IP56
Габаритные размеры манипулятора, не более	
Длина, мм	1696
Ширина, мм	546
Высота, мм	262
Габаритные размеры манипулятора (транспортировочное положение), не более	
Длина, мм	540
Ширина, мм	1008
Высота, мм	403

Параметр	Значение
Габаритные размеры манипулятора в упаковке, не более	
Длина, мм	1139
Ширина, мм	660
Высота, мм	605
Масса манипулятора, не более, кг	66
Масса манипулятора с пультом управления в упаковке, не более, кг	75
Степень защиты корпуса шкафа управления	IP44
Габаритные размеры шкафа управления, не более	
Длина, мм	693
Ширина, мм	279
Высота, мм	592
Габаритные размеры шкафа управления в упаковке, не более	
Длина, мм	747
Ширина, мм	362
Высота, мм	649
Масса шкафа управления, не более, кг	25
Масса шкафа управления в упаковке, не более, кг	27
Степень защиты корпуса пульта управления	IP54
Габаритные размеры пульта управления, не более	
Длина, мм	392
Ширина, мм	227
Высота, мм	52
Масса пульта управления, не более, кг	3
Температура эксплуатации, °C	от +5 до +50
Относительная влажность максимальная (без конденсации), %	80
Номинальная потребляемая мощность, Вт	2500
Максимальная потребляемая мощность манипулятора в сборе 230 В АС, не более, Вт	3600

Параметр	Значение
Максимальная потребляемая мощность манипулятора 48 В DC, Вт	3200
Максимальная потребляемая мощность пульта управления 12 В DC, Вт	10
Ресурс, не менее, часов	10000
Относительная влажность при 25 °С (без конденсации), не более, %	80
Разъем для подключения инструмента	SF812B/S8
Интерфейс механический	фланец круглый в соответствии с ГОСТ Р 60.3.4.1-2017/ИСО 9409-1:2004, исполнение 63-4-M6
Шкаф управления	
Состав аппаратных компонентов	- вычислительный блок; - плата безопасности; - плата компенсации; - блоки питания
Характеристики вычислительного блока	
Процессор	4-ядерный с тактовой частотой 3,3/4,3 ГГц
Оперативная память, не менее, Гб	16
Стандарты Wi-Fi	IEEE 802.11ac, IEEE 802.11n, IEEE 802.11g, IEEE 802.11b, IEEE 802.11a, Wi-Fi Direct
Частота беспроводной связи, ГГц	2,4; 5
Объем памяти для хранения данных, не менее, Гб	120

Параметр	Значение
Сервисная панель	- 1 разъем входного питания 230 В 50 Гц; - 1 разъем для подключения пульта управления; - 1 разъем для подключения манипулятора
Пользовательская панель	- 11 разъемных клеммников 15EDGVC-3.5-08P-14; - 1 разъемный клеммник 15EDGVC-3.5-04P-14; - 2 разъем USB тип А для подключения периферийных устройств; - 1 разъем Ethernet для подключения к внешней сети; - 1 разъем HDMI для подключения внешнего монитора
Пульт управления	
Разрешение сенсорного экрана, точек	1280×800
Диагональ экрана, дюйм	10,1
* В значении параметра указан полный механический диапазон движения сервопривода. Доступный диапазон может быть меньше из-за ограничений, предотвращающих самоколлизии манипулятора	

Габаритные размеры составных частей Promobot M13 (см. Рисунок I-1).

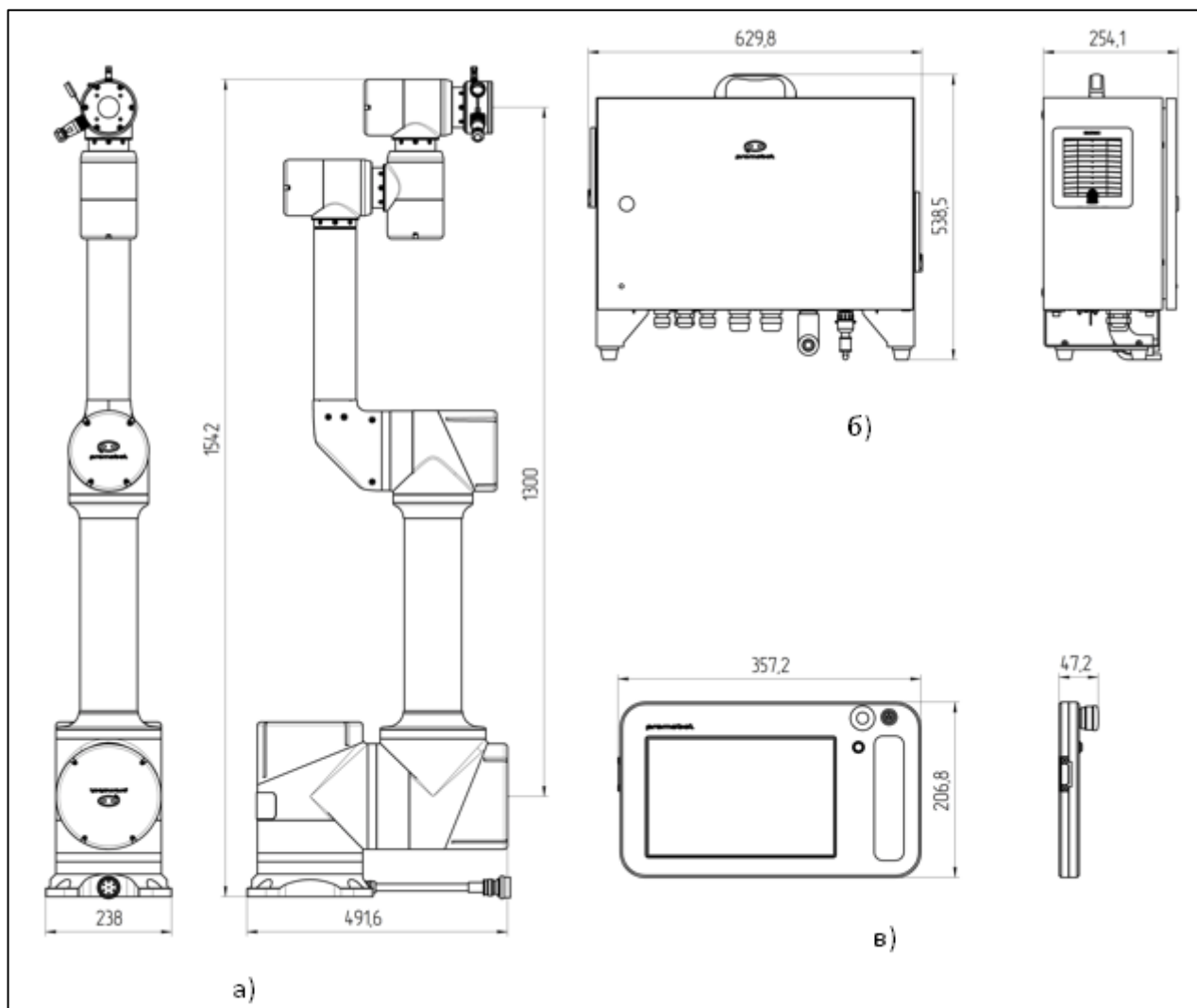


Рисунок I-1 – Габаритные размеры составных частей Promobot M13, где:
а) манипулятор, б) шкаф управления, в) пульт управления

1.1.3 Состав

Общий вид Promobot M13 (см. Рисунок I-2).

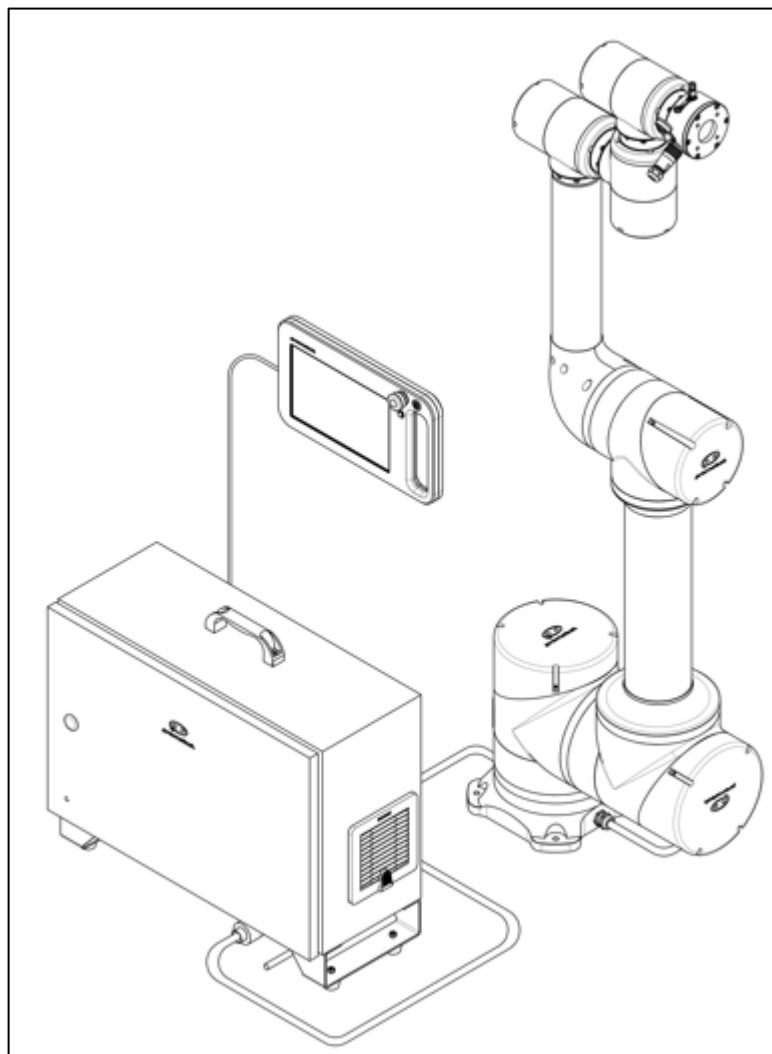


Рисунок I-2 – Общий вид Promobot M13

В комплект Promobot M13 входит:

- манипулятор Promobot M13;
- пульт управления Promobot M13;
- шкаф управления Promobot M13;
- встроенное программное обеспечение Promobot M Control;
- комплект кабелей;
- паспорт;
- руководство по эксплуатации;
- лист описи поставки.

1.1.4 Маркировка

Маркировка выполнена в виде этикеток, расположенных на упаковке Promobot M13, а также на манипуляторе, шкафе управления и пульте управления.

Транспортировочные этикетки (см. Рисунок I-3) располагаются на боковых поверхностях соответствующих упаковок.

а)


Программно-аппаратный комплекс Promobot M13
№: _____
Дата пр-ва _____ г.
Изготовитель:
ООО «Промобот», Россия,
г. Пермь, ул. Шоссе Космонавтов,
стр. 111а, помещ. 6

☎	+ 7 (342) 257-80-85
✉	info@promo-bot.ru
🗣	Promobot_Support_bot

Сделано в России

б)


Манипулятор Promobot M13
Пульт управления Promobot M13
№: _____ Произведено: _____ г.

Основные технические данные	Наименование изделия	Манипулятор	Пульт-управления
	Габаритные размеры	≤600x300x1550 мм	≤400x300x100 мм
	Масса изделия	≤75 кг	≤2 кг
Электрические характеристики	Номинальное напряжение питания	48±3 В	12±0,5 В
	Макс. потребляемая мощность изделия	3200 Вт	10 Вт

Изготовитель:
ООО «Промобот»,
Россия, г. Пермь,
ул. Шоссе Космонавтов,
стр. 111а, помещ. 6

Произведен в соответствии с ТУ 28.99.39-001-40897141-2025
Содержит: манипулятор и пульт-управления
Срок службы: 5 лет
Гарантийный срок: 12 месяцев

Сделано в России

☎ + 7 (342) 257-80-85

✉ info@promo-bot.ru

🗣 Promobot_Support_bot

в)


Шкаф управления Promobot M13
№: _____ Произведено: _____ г.

Основные технические данные	Наименование изделия	Шкаф управления
	Габаритные размеры	≤650x300x540 мм
	Масса изделия	≤35 кг
Электрические характеристики	Номинальное напряжение питания	230±20 В, 55±8 Гц
	Макс. потребляемая мощность изделия	3600 Вт

Изготовитель:
ООО «Промобот»,
Россия, г. Пермь,
ул. Шоссе Космонавтов,
стр. 111а, помещ. 6

Произведен в соответствии с ТУ 28.99.39-001-40897141-2025
Содержит: шкаф управления Promobot M13
Срок службы: 5 лет
Гарантийный срок: 12 месяцев

Сделано в России

☎ + 7 (342) 257-80-85

✉ info@promo-bot.ru

🗣 Promobot_Support_bot

Рисунок I-3 – Транспортные этикетки, где: а) этикетка на общей упаковке Promobot M13, б) этикетка на упаковке манипулятора с пультом управления, в) этикетка на упаковке шкафа управления

Этикетки комплекса (см. Рисунок I-4) имеют общий формат и располагаются: у манипулятора – на основании, у шкафа управления – на боковой панели, у пульта управления – на задней стенке.



Рисунок I-4 – Маркировка составных частей комплекса, где: а) манипулятор, б) пульт управления, в) шкаф управления

1.1.5 Упаковка

Упаковка манипулятора и пульта управления с комплектом кабелей (см. Рисунок I-5) выполнена из гофрокартона с ложементами. Шкаф управления содержится в отдельной упаковке из гофрокартона с ложементами.

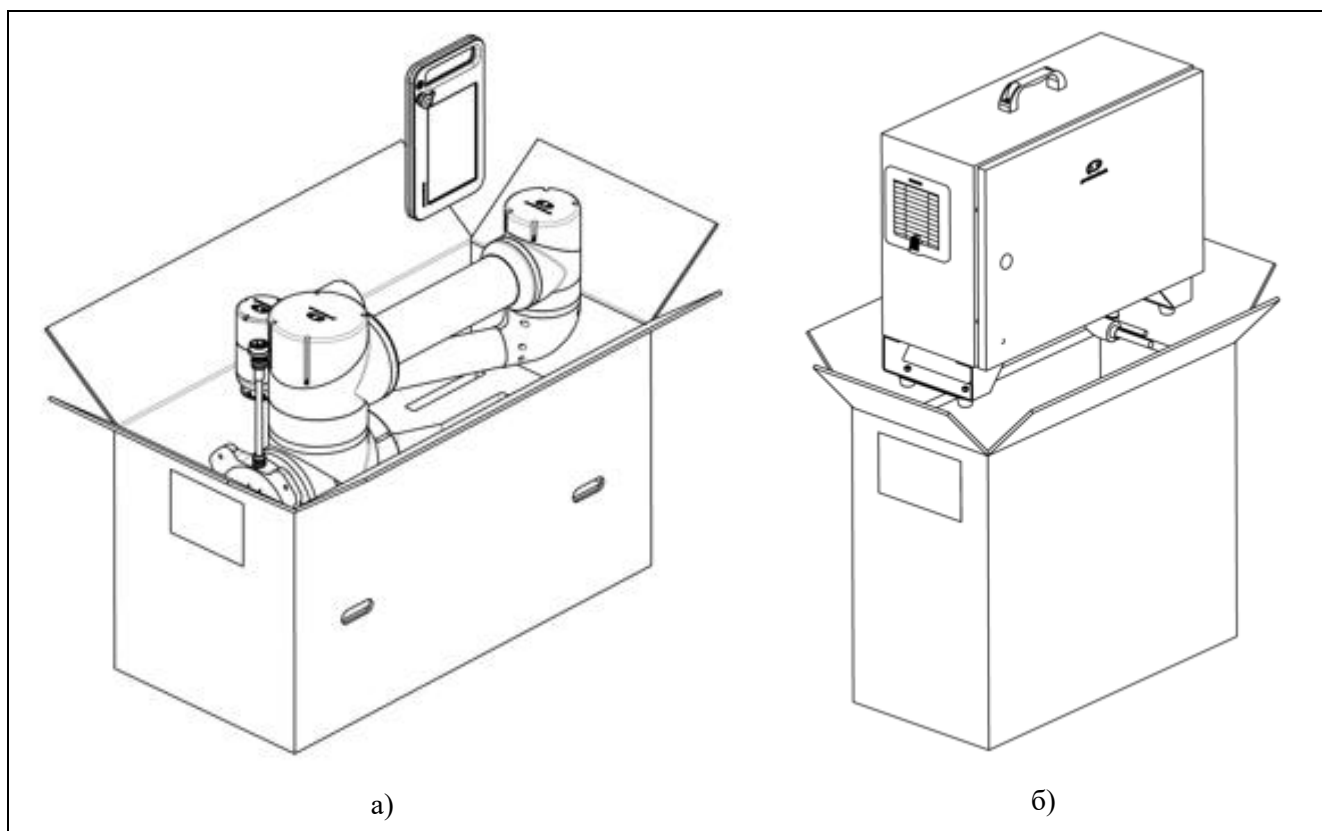


Рисунок I-5 – Упаковка, где: а) упаковка с манипулятором и пультом управления, б) упаковка со шкафом управления

1.2 Описание и работа составных частей Promobot M13

1.2.1 Манипулятор

Манипулятор Promobot M13 (см. Рисунок I-6) представляет собой электромеханическое устройство, состоящее из следующих частей:

- основание, обеспечивающее устойчивость манипулятора; внутри основания расположены элементы коммутации;
- СП1 – поворотное основание, обеспечивающее вращение манипулятора вокруг вертикальной оси;
- СП2 – плечо, обеспечивающее движение в вертикальной плоскости;
- СП3 – предплечье, соединенное с плечом и обеспечивающее дополнительное движение в вертикальной плоскости;
- СП4, СП5 и СП6 – запястья, обеспечивающие ориентацию инструмента или захватного устройства в пространстве;

- фланец инструмента, предназначенный для установки захватного устройства или инструмента.

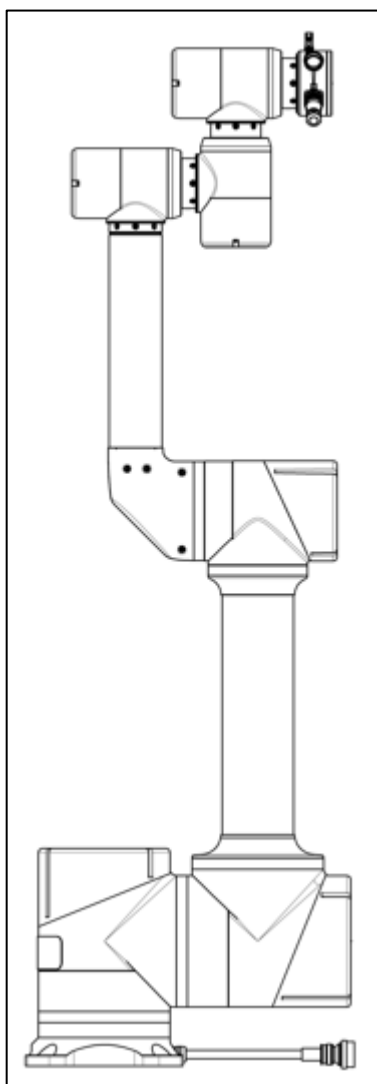


Рисунок I-6 – Манипулятор Promobot M13

Составные части манипулятора (см. Рисунок I-7).

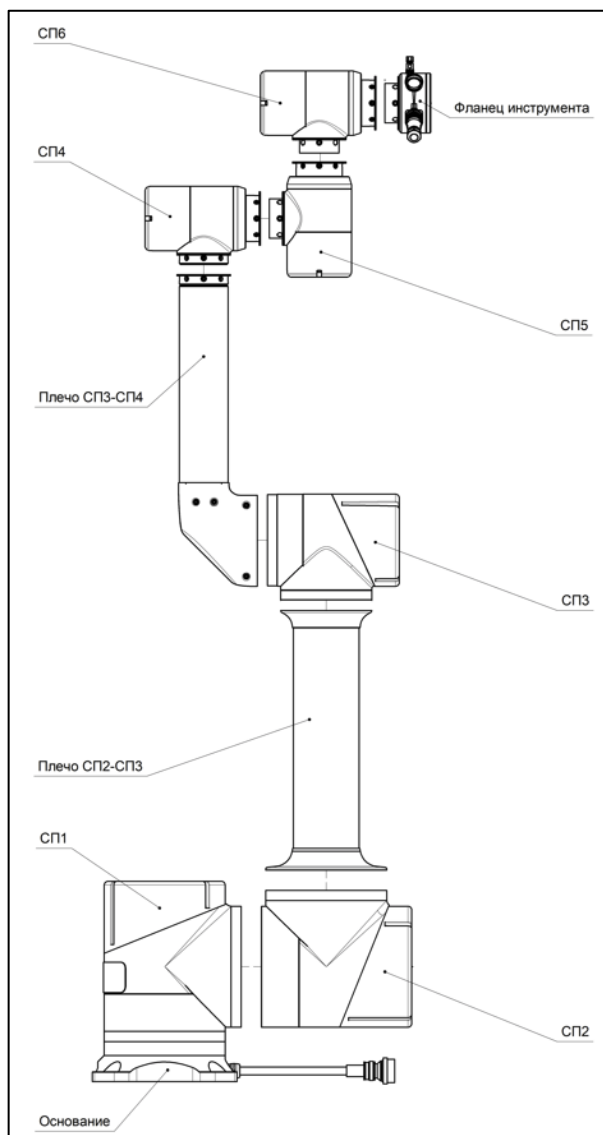


Рисунок I-7 – Составные части манипулятора

Каждый СП представляет собой узел в корпусе, содержащий:

- электромотор;
- редуктор;
- универсальную систему пассивного торможения;
- управляющий вычислительный модуль;
- систему определения положения СП;

Механическая часть манипулятора работает на основе взаимодействия сервоприводов. Каждая ось управляется отдельным бесколлекторным электромотором, который передает движение через редуктор на соответствующий элемент конструкции.

Высокая точность движений обеспечивается за счет использования волновых редукторов и системы обратной связи, состоящей из энкодеров и датчиков тока.

Рабочая зона манипулятора (см. Рисунок I-8), достигает 1300 мм от оси СП2 до оси фланца инструмента.

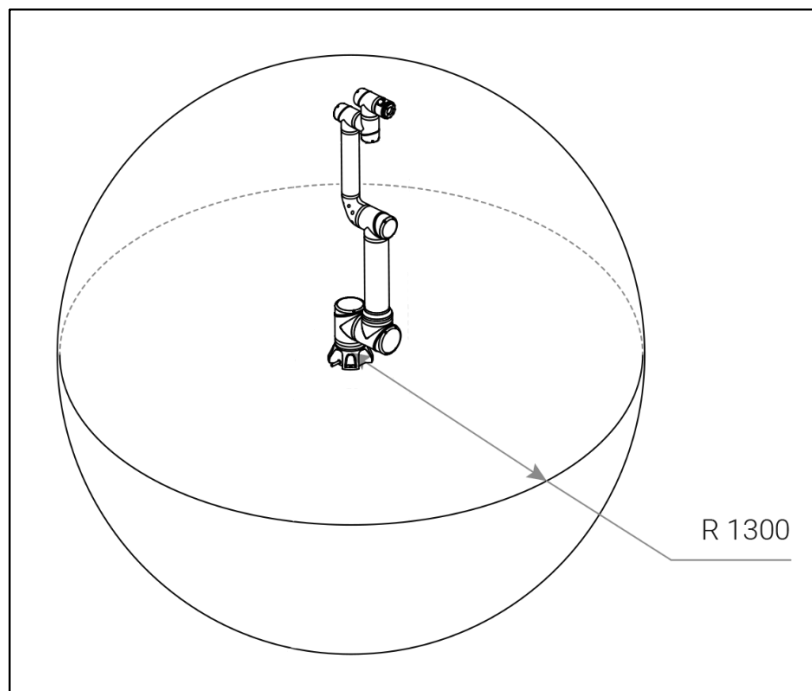


Рисунок I-8 – Рабочая зона манипулятора

1.2.2 Шкаф управления

Шкаф управления Promobot M13 (см. Рисунок I-9) выполняет функции электропитания, управления, обеспечивает безопасность работы и позволяет манипулятору интегрироваться с внешними системами автоматизации. Для обеспечения данного функционала шкаф управления содержит группы электрических интерфейсов – входы и выходы шкафа управления. Все внешние подключения должны осуществляться исключительно через гермовводы разных диаметров, обеспечивающие защиту корпуса шкафа от проникновения пыли и жидкостей (см. Рисунок I-15).

Примечание – Положение разъемов и гермовводов может отличаться от представленного на рисунке (см. Рисунок I-15).

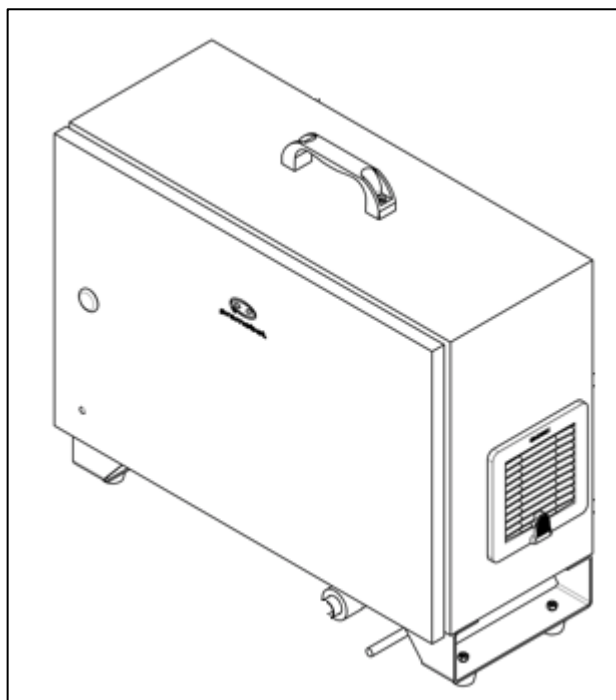


Рисунок I-9 – Шкаф управления Promobot M13

1.2.3 Пульт управления

ПУ Promobot M13 (см. Рисунок I-10) предназначен для взаимодействия оператора с манипулятором. ПУ обеспечивает управление движением манипулятора, настройку параметров работы, ввод программ, диагностику и мониторинг состояния системы.

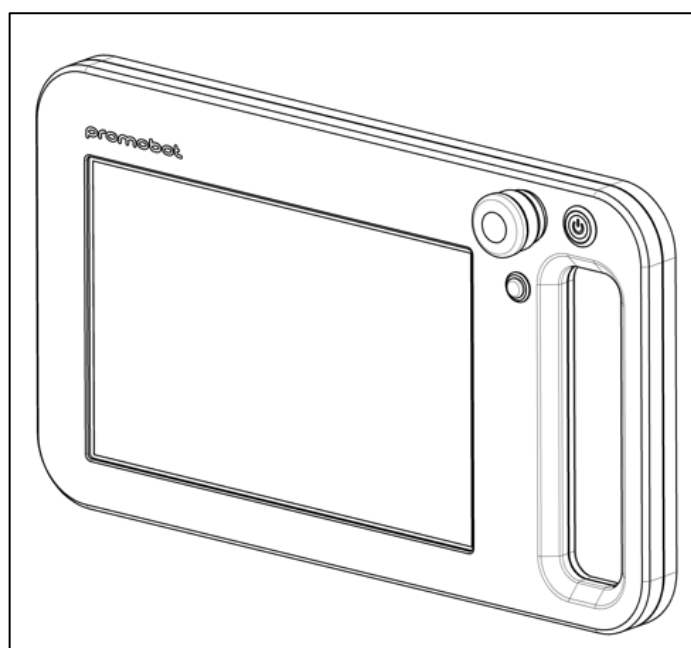


Рисунок I-10 – Пульт управления Promobot M13

ПУ состоит из следующих компонентов (см. Рисунок I-11):

- алюминиевый корпус, который обеспечивает защиту от механических повреждений: имеет класс защиты IP54 (защита от пыли и влаги);
- сенсорный дисплей для отображения информации о состоянии манипулятора, параметрах работы и меню управления;
- функциональные кнопки: кнопка включения/выключения комплекса, кнопка аварийного останова манипулятора, кнопка FreeDrive;
- микроконтроллерный модуль для обработки сигналов сенсорного дисплея.

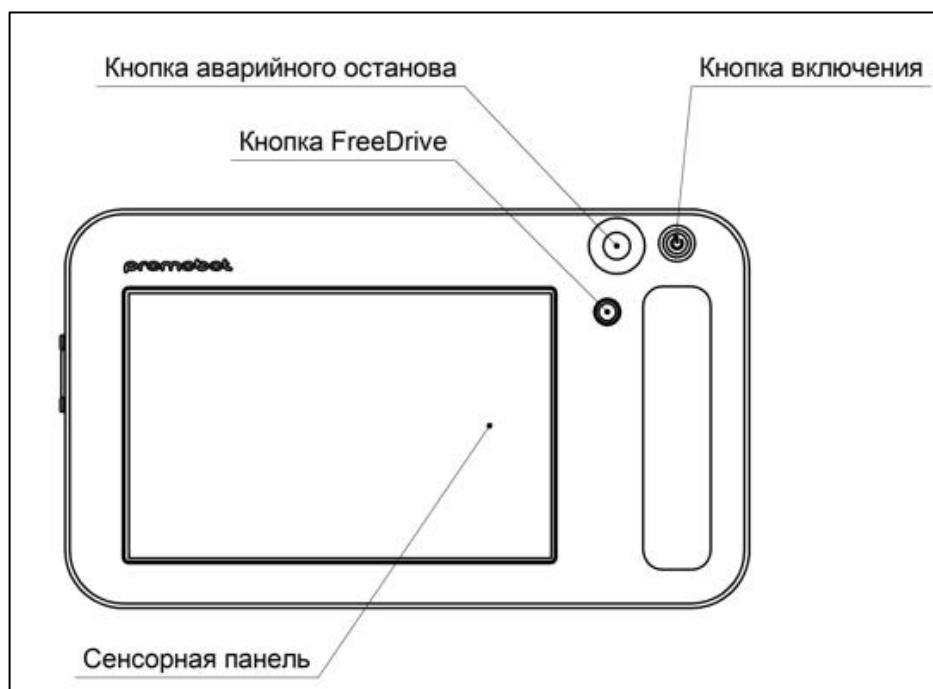


Рисунок I-11 – Составные части ПУ

1.2.4 Встроенное ПО Promobot M Control

Программное обеспечение Promobot M Control предназначено для обеспечения взаимодействия оператора с манипулятором.

Основные функции ПО:

- управление движением манипулятора в ручном режиме;
- создание, редактирование и сохранение программ движения;
- визуализация рабочей зоны и траекторий движения;
- мониторинг состояния и диагностика неисправностей;
- интеграция с внешними системами управления.

Подробное описание программного обеспечения представлено в томе II. ПРИЛОЖЕНИЕ PROMOBOT M CONTROL.

2 ИСПОЛЬЗОВАНИЕ ПО НАЗНАЧЕНИЮ

2.1 Эксплуатационные ограничения Promobot M13

Эксплуатация устройства должна осуществляться в строгом соответствии с техническими характеристиками и рекомендациями, указанными в данном руководстве. Несоблюдение эксплуатационных ограничений может привести к повреждению Promobot M13, травмам персонала и другим негативным последствиям.

Для обеспечения безопасной и эффективной работы Promobot M13, необходимо строго соблюдать эксплуатационные ограничения, приведенные в таблице 2.

Таблица 2 – Эксплуатационные ограничения

Параметр	Значение	Примечание
Максимальная грузоподъемность манипулятора, не более, кг	13	Превышение нагрузки может привести к механическим повреждениям, снижению точности или созданию аварийных ситуаций
Допустимый диапазон температур, °C	от +5 до +50	Работа за пределами указанного диапазона может привести к снижению производительности или выходу комплекса из строя
Относительная влажность воздуха, %	80	Не допускается эксплуатация манипулятора в условиях повышенной влажности или в помещениях с образованием конденсата. Это может вызвать коррозию, короткое замыкание или повреждение электронных компонентов

Параметр	Значение	Примечание
Линейная скорость перемещения центральной точки инструмента (ЦТИ), не более, м/с	1	Превышение этого параметра может привести к потере точности, повреждению Promobot M13 или травмам персонала

Манипулятор должен подключаться только к источнику электропитания, соответствующему техническим характеристикам (напряжение, частота, мощность).

Строго запрещается эксплуатировать манипулятор в условиях, характеризующихся повышенной запыленностью, а также в средах, содержащих агрессивные химические вещества, такие как кислоты, щелочи, растворители и так далее. Кроме того, недопустимо подвергать манипулятор ударным нагрузкам, воздействию вибраций или иным механическим воздействиям.

Запрещается использовать манипулятор для работы со взрывоопасными, легковоспламеняющимися или токсичными материалами без дополнительных мер защиты, предусмотренных предприятием-изготовителем.

Регулярно обновляйте программное обеспечение Promobot M Control, чтобы обеспечить стабильную работу и безопасность комплекса.

Запрещается проводить самостоятельный ремонт без согласования с предприятием-изготовителем или авторизованным сервисным центром. Неквалифицированное вмешательство может нарушить штатную работу устройства и сделать его использование небезопасным, а также аннулирует гарантию.

Также не допускается самостоятельная модификация механических узлов, электроники или программного обеспечения манипулятора.

Манипулятор должен использоваться только для задач, предусмотренных его назначением. Запрещается применять комплекс для выполнения работ, не указанных в технической документации.

Соблюдение указанных ограничений обеспечивает безопасность специалистов, продлевает срок службы манипулятора и гарантирует его корректную работу. В случае возникновения нештатных ситуаций или сомнений в правильности эксплуатации

прекратите использование устройства и обратитесь в сервисную службу предприятия-изготовителя.

2.2 Подготовка Promobot M13 к использованию

Перед началом эксплуатации Promobot M13 необходимо выполнить ряд подготовительных мероприятий, чтобы обеспечить безопасную и корректную работу устройства.

Действия перед установкой:

- убедитесь, что условия эксплуатации соответствуют требованиям, описанным в разделе 2.1 «Эксплуатационные ограничения»;
- убедитесь в отсутствии на упаковке повреждений, полученных при транспортировке. При обнаружении дефектов на упаковке требуется обратиться к предприятию-изготовителю и не приступать к распаковке комплекса.

В н и м а н и е!

- 1 Не допускается подключение и включение манипулятора до его монтажа.
- 2 Установка комплекса должна проходить в условиях, отвечающих требованиям класса защиты.

2.2.1 Монтаж

При установке манипулятора, ввиду большого веса, необходимо использовать стропы:

- используйте круглопрядные текстильные стропы длиной 2 метра;
- при обхвате стропами используйте захват мертвой петлей за плечо СП2–СП3 (см. Рисунок I-12);
- нельзя подвергать усилиям запрещенные места обхвата (СП4, СП5, СП6) и пластиковые защитные крышки (см. Рисунок I-12).

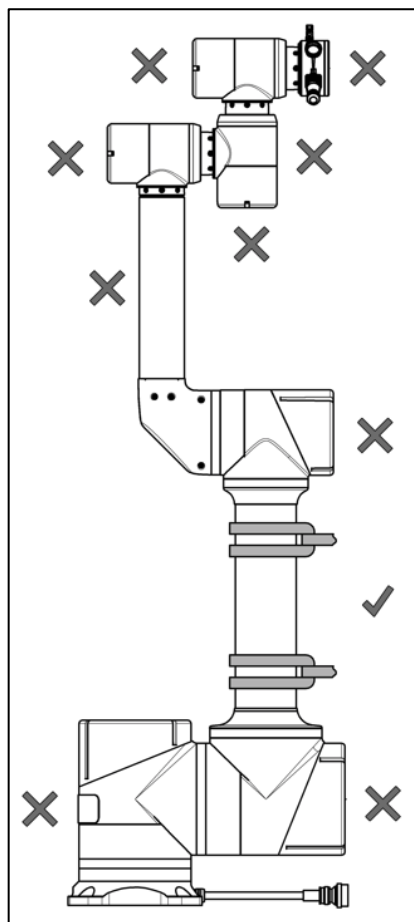


Рисунок I-12 – Разрешенные и запрещенные места обхвата при монтаже/демонтаже манипулятора

Для установки Promobot M13 выполните следующие действия:

- а) извлеките манипулятор, шкаф управления, пульт управления и комплектующие из упаковки (будьте аккуратны при распаковке, чтобы не повредить упаковочный материал – он может понадобиться для хранения и дальнейшей транспортировки Promobot M13);
- б) проверьте комплектность согласно паспорту комплекса;
- в) убедитесь в отсутствии повреждений составных частей Promobot M13 при транспортировке;
- г) определите место и способ установки: напольный, настенный, подвесной, на движущейся платформе или на линейной оси:
 - при установке необходимо учитывать рабочий диапазон манипулятора и обеспечить достаточное пространство для движения согласно пункту 4.3 «Рабочее окружение манипулятора»;

- место установки ровное, устойчивое и свободное от внешних вибраций, способное выдержать нагрузку с воздействием силы в 2 кН;
- требование к напольному покрытию: бетон не хуже марки Б20 и толщиной не менее 200 мм;
- требования к режиму скорости линейной оси или движущейся платформе: минимальное время ускорения и замедления – 0,6 с., максимальная линейная скорость – 1 м/с;

Внимание! Несоблюдение режима скорости может привести к срабатыванию аварийного останова и повреждению манипулятора.

д) установите манипулятор:

- просверлите 4 монтажных отверстия по окружности диаметром $\varnothing 254$ мм под крепления основания манипулятора согласно схеме разметки (см. Рисунок I-13);

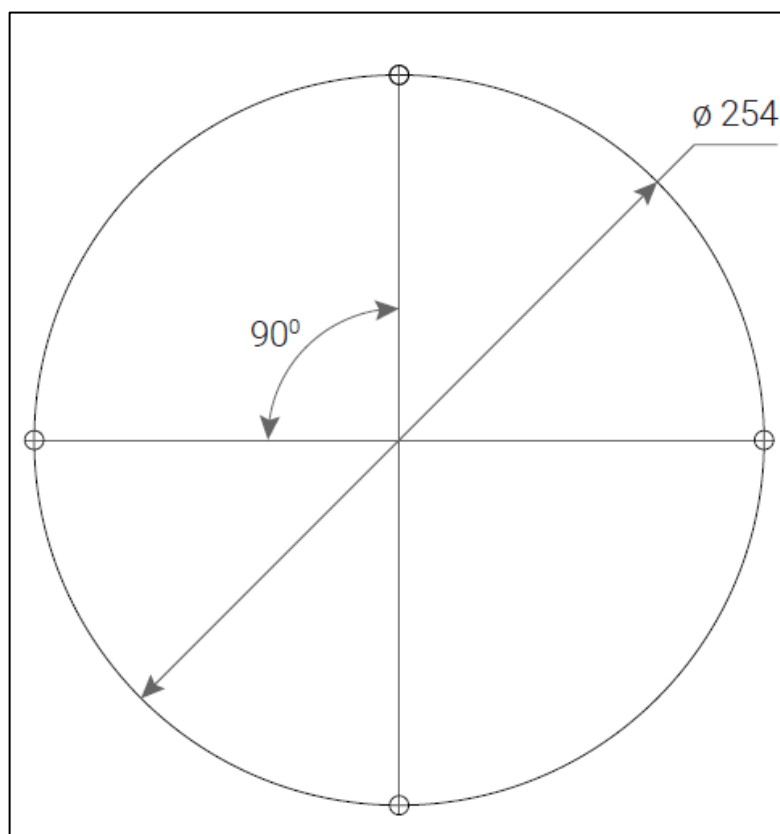


Рисунок I-13 – Схема разметки отверстий для монтажа манипулятора

- установите манипулятор основанием на подготовленную поверхность и закрепите крепежными элементами: на химический анкер со шпилькой не менее M8x100 или на винты M8 (не входят в комплект поставки,

приобретаются отдельно), рекомендуемый стандарт DIN 912, класс прочности не ниже 8.8;

В н и м а н и е! После монтажа осмотрите манипулятор и убедитесь, что он надежно закреплен и устойчив. Неустойчивая установка может привести к некорректной работе, повреждениям манипулятора и травмам персонала.

- е) разместите на полу шкаф управления:
 - допускается монтаж шкафа управления на стену с использованием четырех монтажных отверстий ø6 мм на корпусе шкафа (отверстия закрыты заглушками);
 - для обеспечения циркуляции воздуха с каждой стороны шкафа управления необходимо оставить свободное пространство шириной 100 мм;
 - условия расположения шкафа управления должны соответствовать его классу защиты;
- ж) убедитесь, что при эксплуатации кабель пульта управления не создает опасности споткнуться; для удобства хранения пульта можно использовать специальные кронштейны (не входят в комплект поставки, приобретаются отдельно);
- з) закрепите рабочий инструмент на фланец (см. Рисунок I-14):
 - фланец инструмента выполнен в соответствии с ГОСТ Р 60.3.4.1–2017/ИСО 9409-1:2004 исполнение 63-4-M6;
 - при необходимости подключения инструмента возможно использование разъема SF812B/S8 (см. раздел «2.2.2.2 Подключение инструмента»).

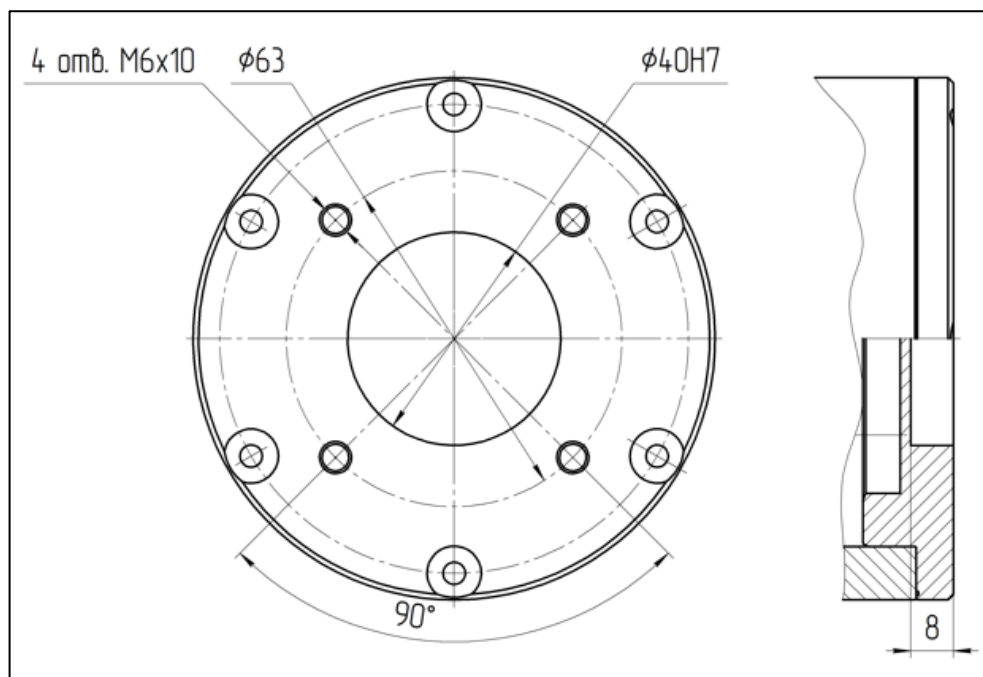


Рисунок I-14 – Фланец инструмента

- и) убедитесь, что рабочая зона свободна от посторонних предметов;
- к) установите ограждения или маркеры для обозначения зоны работы манипулятора (не входят в комплект поставки, приобретаются отдельно).

2.2.2 Подключение

2.2.2.1 Подключение электрических соединений

Подключите шкаф управления и пульт управления кабелями из комплекта поставки Promobot M13 следующим образом:

- а) подключите кабель питания к входному разъему шкафа управления;
- б) подключите кабель манипулятора к шкафу управления и к манипулятору (см. Рисунок I-15);

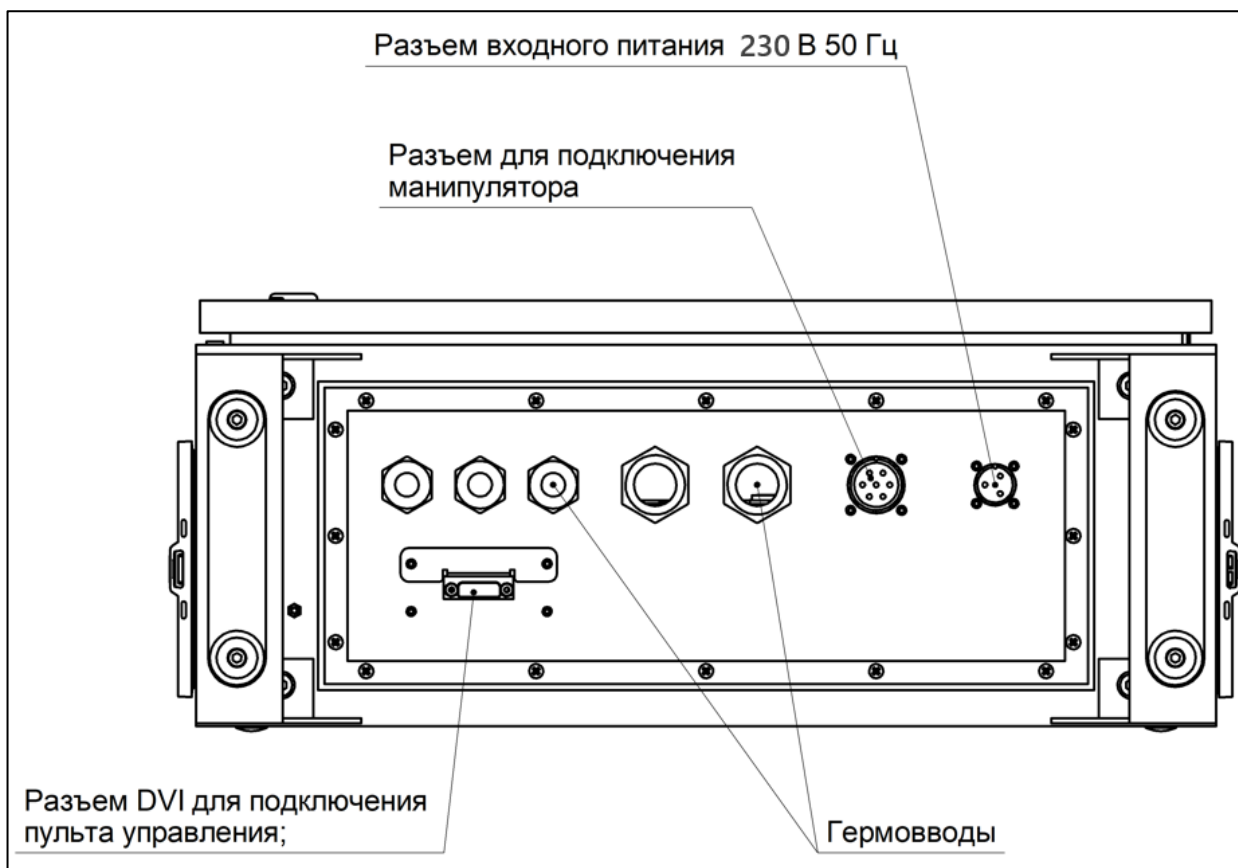


Рисунок I-15 – Расположение разъема подключения манипулятора и гермоввода для прохода кабеля пульта управления

- в) откройте шкаф управления, убедитесь, что автоматический выключатель находится в положении «ВЫКЛ»;
- г) подключите пульт управления в разъем DVI;
- д) подключите необходимые периферийные устройства и модули (при наличии) (см. раздел 2.2.2.3 «Подключение внешних устройств»);
- е) убедитесь, что дополнительные кнопки аварийного останова установлены и переведены в положение «ВКЛ» либо в соответствующие разъемы установлены перемычки (см. раздел 4.4.1 «Категории останова»);
- ж) подключите кабель питания к питающей сети.

Примечание – Если комплекс был перемещен из зоны с отрицательной температурой в теплое помещение, обязательно выдержите его в новых температурных условиях не менее 12 часов, прежде чем подключить к электросети.

- з) переведите автоматический выключатель шкафа управления в положение «ВКЛ»;

- и) нажмите кнопку включения на пульте управления: после подачи питания шкаф управления издаст один длинный звуковой сигнал, на пульте управления будет отображаться процесс загрузки управляющего ПО;
- к) закройте шкаф управления.

Внимание! Убедитесь, что шкаф управления и кабели не контактируют с водой. Последствия контакта могут привести к смертельной травме.

2.2.2.2 Подключение инструмента

На фланце находятся группы электрических интерфейсов, необходимые для подключения инструмента (см. Рисунок I-16).

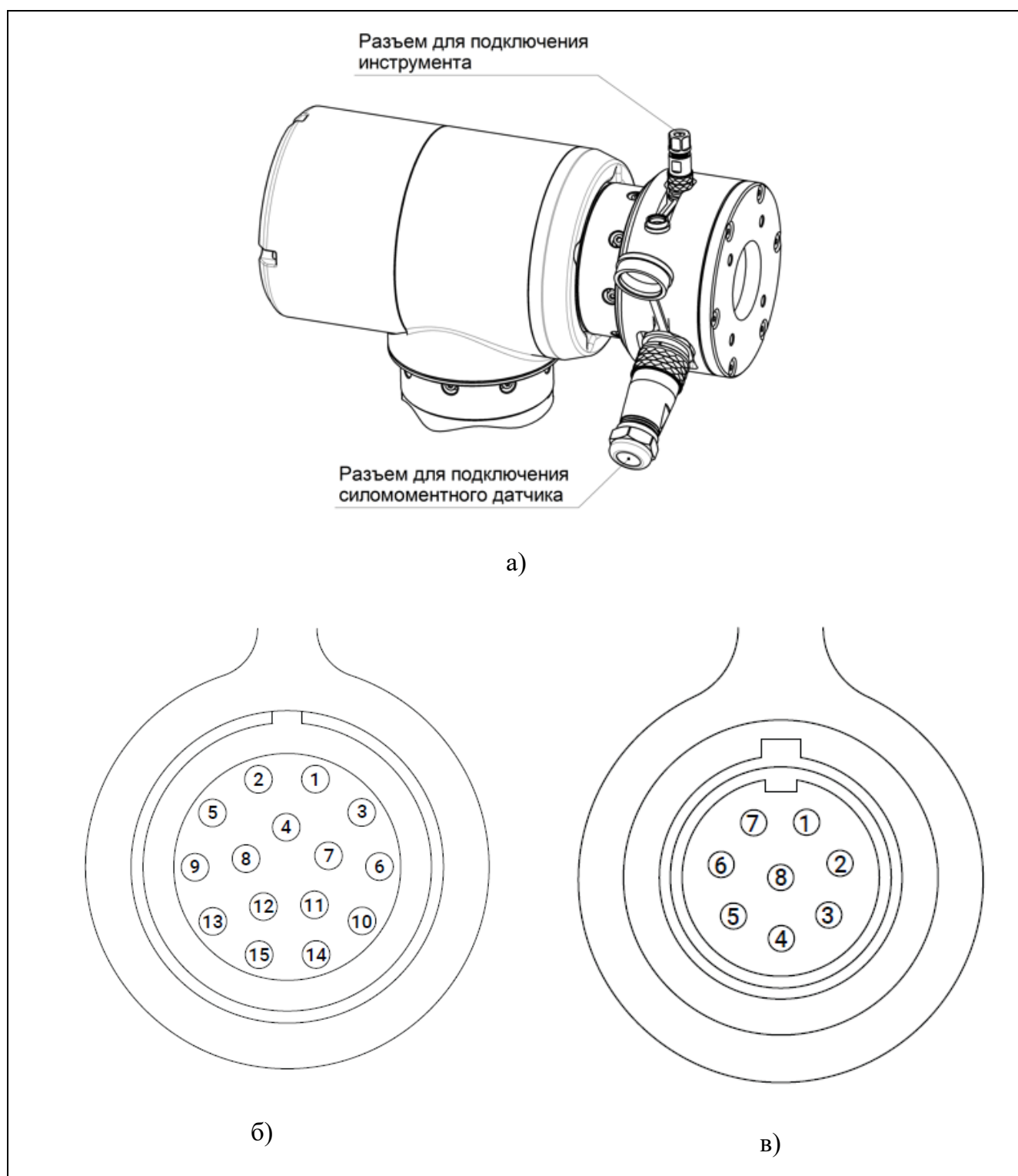


Рисунок I-16 – Группы электрических интерфейсов на фланце, где: а) входы и выходы на фланце инструмента, б) разъем SF2012/S151 для подключения силомоментного датчика, в) разъем SF812B/S8 для подключения инструмента

Порядок расположения контактов разъема инструмента (см. Рисунок I-17).

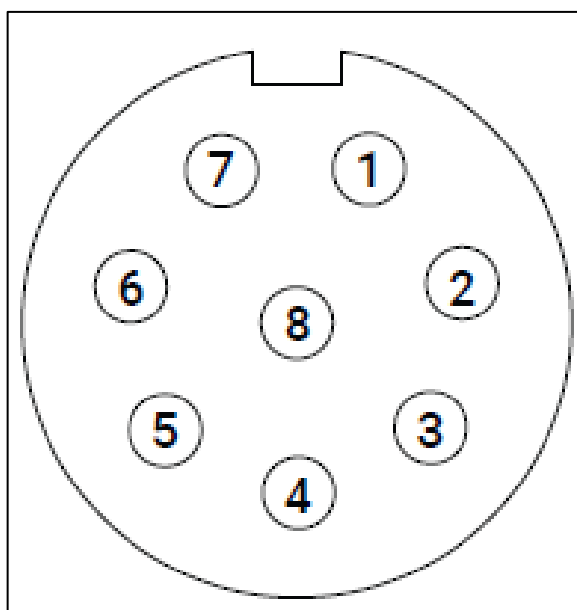


Рисунок I-17 – Порядок расположения контактов разъема инструмента

Описание и характеристики отличаются в зависимости от установленной версии платы инструмента V2 или платы инструмента V3. Версия платы, установленная на вашем манипуляторе, отображается в интерфейсе мониторинга, в разделе «Программное обеспечение» (см. раздел II.6 «МОНИТОРИНГ»).

Описание и характеристики контактов разъема инструмента при установленной версии платы инструмента V2 представлены в таблице 3.

Таблица 3 – Описание и характеристики контактов разъема инструмента платы инструмента V2

№	Краткое наименование	Описание	Характеристики
1	GADGET_1	Аналоговый вход 2 (AIN2) / RS-485 (B)	Режим измерения напряжения:

№	Краткое наименование	Описание	Характеристики
2	GADGET_2	Аналоговый вход 1 (AIN1) / RS-485 (A)	- Максимальное измеряемое напряжение – 24 В; - Входное сопротивление – 11,5 кОм. Режим токовой петли: - Максимальное входное напряжение – 5 В; - Максимальный входной ток – 25 мА; - Входное сопротивление – 182 Ом
3	GADGET_3	Цифровой вход 2 (DIN2)	- Внутренняя подтяжка: к земле 50 кОм; - Логический «0»: <2 В; - Логическая «1»: >5 В; - Максимальный рабочий диапазон 0–24 В; - Макс допустимое напряжение 26 В
4	GADGET_4	Цифровой вход 1 (DIN1)	
5	GADGET_5	Цифровой выход 2 (DOUT2)	24 В, максимум 1 А Возможные режимы: - Sinking (Low/open); - Sourcing (High/open); - Push/Pull (High/Low)
6	GADGET_6	Цифровой выход 1 (DOUT1)	12 В, максимум 1 А Возможные режимы: - Sinking (Low/open); - Sourcing (High/open); - Push/Pull (High/Low)
7	GADGET_7	Питание (0 В) / 24 В	24 В, максимум 1 А
8	GADGET_8	Земля (GND)	Максимум 1 А, объединена с корпусом устройства

Описание и характеристики контактов разъема инструмента при установленной версии платы инструмента V3 представлены в таблице 4.

Таблица 4 – Описание и характеристики контактов разъема инструмента платы инструмента V3

№	Краткое наименование	Описание	Характеристики
1	GADGET_1	Аналоговый вход 2 (AIN2) / RS-485 (B)	Режим измерения напряжения: - Максимальное измеряемое напряжение – 24 В; - Входное сопротивление – 11,5 кОм.
2	GADGET_2	Аналоговый вход 1 (AIN1) / RS-485 (A)	Режим токовой петли: - Максимальное входное напряжение – 5 В; - Максимальный входной ток – 25 мА; - Входное сопротивление – 182 Ом
3	GADGET_3	Цифровой вход 2 (DIN2)	- Входное сопротивление – 50 кОм; - Логический «0»: <2 В; - Логическая «1»: >5 В; - Рабочий диапазон – 0–24 В; - Максимально допустимое напряжение – 26 В
4	GADGET_4	Цифровой вход 1 (DIN1)	
5	GADGET_5	Цифровой выход 2 (DOUT2)	Доступные напряжения: - 24 В, максимально 1 А. Возможные режимы: - Sinking (Low/open); - Sourcing (High/open); - Push/Pull (High/Low)

6	GADGET_6	Цифровой выход 1 (DOUT1)	Доступные напряжения: - 12 В, максимально 1 А; - 24 В, максимально 1 А. Возможные режимы: - Sinking (Low/open); - Sourcing (High/open); - Push/Pull (High/Low)
7	GADGET_7	Питание (VOUTPUT)	Доступные напряжения: - 24 В, максимально 1 А; - 12 В, максимально 1 А; - 0 В (GND), максимально 1 А
8	GADGET_8	Земля (GND)	Максимально 2 А, объединена с корпусом устройства
Примечания - Для индуктивных нагрузок рекомендуется использовать защитный диод. - Цифровые входы с высокоомными подтягивающими резисторами – означает, что плавающий вход всегда имеет низкий уровень. - Аналоговые входы не дифференциальные и могут быть настроены на напряжение (0–24 В) или ток (4–20 мА). - Короткое замыкание цифровых выходов или порта питания может привести к повреждению M13			

2.2.2.3 Подключение внешних устройств

Для обеспечения интеграции с внешними устройствами шкаф управления Promobot M13 содержит группы электрических интерфейсов, расположенных на пользовательской панели – входы и выходы шкафа управления (см. Рисунок I-18).

Группы электрических интерфейсов имеют световую индикацию, сигнализирующую об их активном состоянии.

Все внешние подключения осуществлять исключительно через гермовводы, обеспечивающие защиту корпуса шкафа от проникновения пыли и жидкостей.

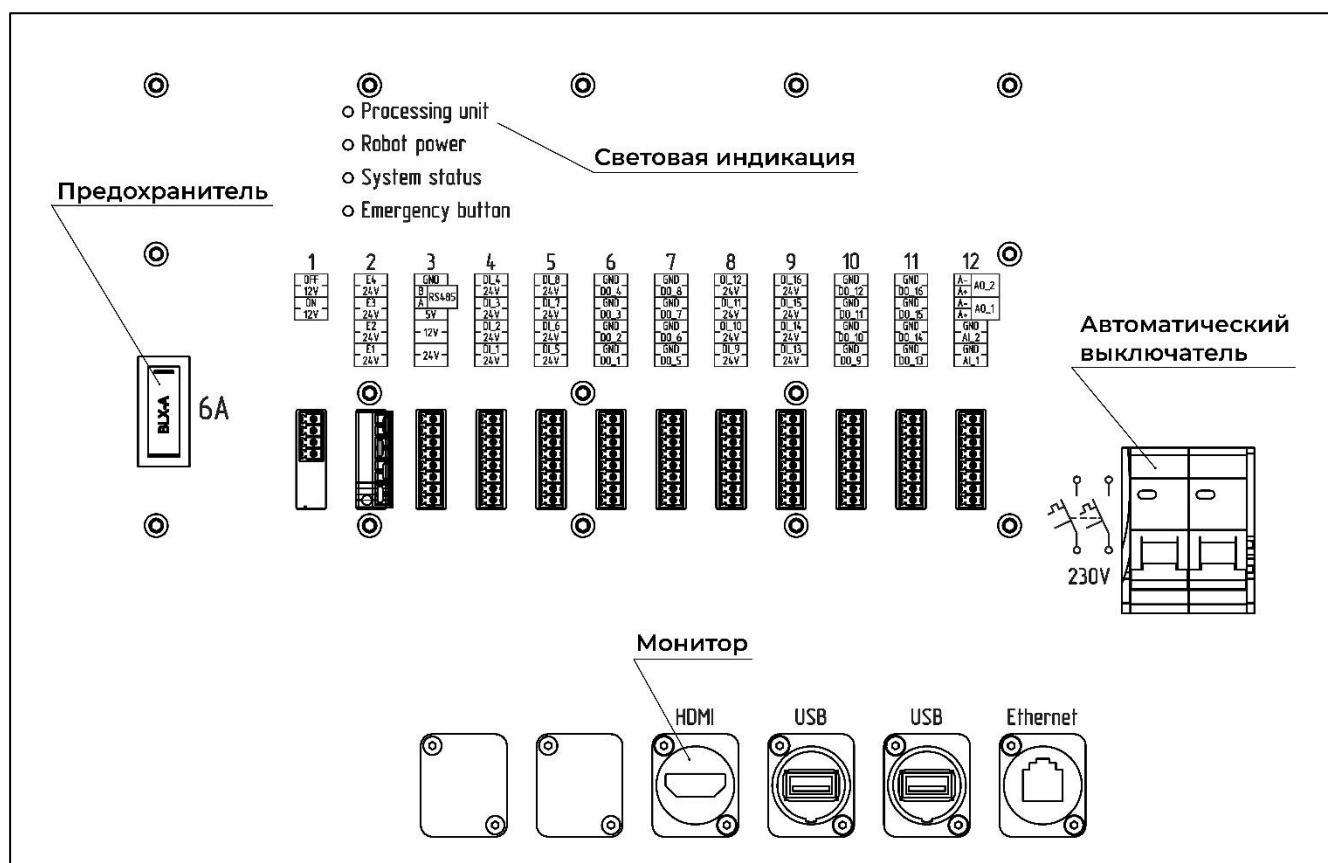


Рисунок I-18 – Общий вид пользовательской панели шкафа управления

Описание и характеристики отличаются в зависимости от установленной версии платы безопасности V2 или V3. Версию платы, установленную на вашем манипуляторе, вы можете уточнить, обратившись в сервисный центр предприятия-изготовителя.

Описание и характеристики интерфейсов, при установленной версии платы безопасности V2, представлены в таблице 5.

Таблица 5 – Описание и характеристики интерфейсов платы безопасности V2

№ разъема	Наименование разъема	Описание	Характеристики
1	Входы включения манипулятора	Разъемы для удаленного включения манипулятора	Разъем: 15EDGVC-3.5-04P-14; Нумерация контактов снизу вверх: - 1, 3 – 12 В, суммарно макс. 0,3 А; - 2 – контакт вкл.; - 4 – контакт выкл. Напряжение: - пределы напряжения для области «ВКЛ.»: мин. 7 В, ном. 12 В, макс. 14 В; - пределы напряжения для области «ВЫКЛ.»: мин. -3 В, ном. 0 В, макс. 3 В. Ток: мин. 1 мА, ном. 2 мА, макс. 20 мА
2	Входы Emergency	Группы дискретных входов для подключения кнопок и датчиков	Нумерация контактов снизу вверх: - 1, 3, 5, 7: - напряжение: 24 В; - макс. суммарный ток 0,7 А; - 2, 4 – аппаратные входы (для нормальной работы должны быть замкнуты на 24 В); - 6, 8 – цифровые входы. Параметры цифровых входов: - Напряжение: - пределы напряжения для области «ВКЛ.»: мин. 11 В, ном. 24 В, макс. 30 В; - пределы напряжения для области «ВЫКЛ.»: мин. -3 В, ном. 0 В, макс. 11 В. - Ток: мин. 2 мА, ном. 5 мА, макс. 15 мА

№ разъема	Наименование разъема	Описание	Характеристики
3	Выходы 24 В, 12 В, 5 В и интерфейс RS-485	Источники постоянного напряжения для внешних устройств. Примечание: В процессе эксплуатации интерфейса RS-485 при необходимости установите внешний терминатор	Нумерация контактов снизу вверх: - 1,2 – 24 В ($\pm 5\%$), 0,7 А; - 3,4 – 12 В ($\pm 5\%$), 0,7 А; - 5 – 5 В ($\pm 5\%$), 0,7 А; - 6 – RS-485 (А); - 7 – RS-485 (В); - 8 – GND
4, 5, 8, 9	Цифровые входы DI_1–DI_4, DI_5–DI_8, DI_9–DI_12, DI_13–DI_16	Группы дискретных входов для подключения датчиков	Нумерация контактов снизу вверх. Напряжение: - пределы напряжения для области «ВКЛ.»: мин. 11 В, ном. 24 В, макс. 30 В; - пределы напряжения для области «ВЫКЛ.»: мин. -3 В, ном. 0 В, макс. 11 В. Ток: мин. 2 мА, ном. 5 мА, макс. 15 мА
6, 7, 10, 11	Цифровые выходы DO_1–DO_4, DO_5–DO_8, DO_9–DO_12, DO_13–DO_16	Группы дискретных выходов для управления внешними устройствами	Нумерация контактов снизу вверх: - 1, 3, 5, 7 – выходы: - напряжение: 24 В; - ток макс. 0,3 А; - 2, 4, 6, 8 – GND

№ разъема	Наименование разъема	Описание	Характеристики
12	Аналоговые входы (AIN) 1, 2 и аналоговые выходы (AOUT) 1, 2	Входы и выходы для аналоговых сигналов	Нумерация контактов снизу вверх: - 1, 3 – аналоговые входы: - в режиме измерения напряжения 0...10 В, 11,5 кОм - или - в режиме токовой петли 4...20 мА; 150 Ом; - 5, 7 – аналоговые выходы: - в режиме напряжения 0...10 В, 0...20 мА - или - в режиме токовой петли 4...20 мА, макс. 12 В; - 2, 4, 6, 8 – GND
–	Предохранитель	Предназначен для защиты портов GND от суммарного превышения тока 6 А	Сменный предохранитель 6 А, корпус 5×20

№ разъема	Наименование разъема	Описание	Характеристики
–	Световая индикация	Светодиоды для передачи информации о состоянии или процессе работы манипулятора	Назначение снизу вверх: - Processing unit – индикатор состояния вычислительного блока (постоянное свечение зеленым – нормальное функционирование вычислительного блока манипулятора; отсутствует свечение – вычислительный блок отключен); - Robot power – в процессе: - запуска системы: индикатор успешной инициализации (постоянное свечение зеленым – вычислительный блок запущен, инициализация периферии прошла успешно); - работы комплекса: индикатор наличия напряжения на манипуляторе (прерывистое свечение раз в секунду – напряжение подано на манипулятор); - System status – индикатор состояния платы безопасности (постоянное свечение зеленым – загрузчик запущен на плате; прерывистое свечение – наличие ошибок на плате безопасности; отсутствует свечение – отсутствие ошибок на плате безопасности или запущен процесс обновления ПО на плате безопасности); - Emergency button – индикатор состояния аварийной кнопки на пульте управления (постоянное свечение зеленым – кнопка деактивирована (состояние нормально замкнутое); отсутствует свечение – кнопка активирована (состояние разомкнутое))
–	HDMI Монитор	Разъем для подключения внешнего дисплея	Тип: А; Стандарт: 1.4a

№ разъема	Наименование разъема	Описание	Характеристики
—	USB	Порт для подключения внешних периферийных устройств	Стандарт: USB 2.0; Тип: Type-A; Ток: 500 мА
—	Ethernet	Сетевой интерфейс для подключения к внешней локальной сети	Стандарт: 100BASE-TX; Скорость: 100 Мбит/с; Разъем: 8P8C

Описание и характеристики интерфейсов, при установленной версии платы безопасности V3, представлены в таблице 6.

Таблица 6 – Описание и характеристики интерфейсов плата безопасности V3

№ разъема	Наименование разъема	Описание	Характеристики
1	Входы включения манипулятора	Разъемы для удаленного включения манипулятора	Разъем: 15EDGVC-3.5-04P-14; Нумерация контактов снизу вверх: - 1, 3 – 12 В, суммарно макс. 0,3 А; - 2 – контакт вкл.; - 4 – контакт выкл. Напряжение: - пределы напряжения для области «ВКЛ.»: мин. 7 В, ном. 12 В, макс. 14 В; - пределы напряжения для области «ВЫКЛ.»: мин. -3 В, ном. 0 В, макс. 3 В. Ток: мин. 1 мА, ном. 2 мА, макс. 20 мА

№ разъема	Наименование разъема	Описание	Характеристики
2	Входы Emergency	Группы дискретных входов для подключения кнопок и датчиков	Нумерация контактов снизу вверх: - 1, 3, 5, 7: - напряжение: 24 В; - макс. суммарный ток 0,7 А; - 2, 4 – аппаратные входы (для нормальной работы должны быть замкнуты на 24 В); - 6, 8 – цифровые входы. Параметры цифровых входов: - Напряжение: - пределы напряжения для области «ВКЛ.»: мин. 11 В, ном. 24 В, макс. 30 В; - пределы напряжения для области «ВЫКЛ.»: мин. -3 В, ном. 0 В, макс. 11 В. - Ток: мин. 2 мА, ном. 5 мА, макс. 15 мА
3	Выходы 24 В, 12 В, 5 В и интерфейс RS-485	Источники постоянного напряжения для внешних устройств. Примечание: В процессе эксплуатации интерфейса RS-485 при необходимости установите внешний терминатор	Нумерация контактов снизу вверх: - 1, 2 – 24 В ($\pm 5\%$), 1,5 А; - 3, 4 – 12 В ($\pm 5\%$), 0,7 А; - 5 – 5 В ($\pm 5\%$), 0,7 А; - 6 – RS-485 (A); - 7 – RS-485 (B); - 8 – GND

№ разъема	Наименование разъема	Описание	Характеристики
4, 5, 8, 9	Цифровые входы DI_1–DI_4, DI_5–DI_8, DI_9–DI_12, DI_13–DI_16	Группы дискретных входов для подключения датчиков	Нумерация контактов снизу вверх. Напряжение: - пределы напряжения для области «ВКЛ.»: мин. 11 В, ном. 24 В, макс. 30 В; - пределы напряжения для области «ВЫКЛ.»: мин. -3 В, ном. 0 В, макс. 11 В. Ток: мин. 2 мА, ном. 5 мА, макс. 15 мА
6, 7, 10, 11	Цифровые выходы DO_1–DO_4, DO_5–DO_8, DO_9–DO_12, DO_13–DO_16	Группы дискретных выходов для управления внешними устройствами	Нумерация контактов снизу вверх: - 1, 3, 5, 7 – выходы: - напряжение: 24 В; - ток макс. 1,9 А; суммарный макс. ток всех 16 портов вместе с выходами 24 В - «1», «2» – 3,5 А; - 2, 4, 6, 8 – GND
12	Аналоговые входы (AIN) 1, 2 и аналоговые выходы (AOUT) 1, 2	Входы и выходы для аналоговых сигналов	Нумерация контактов снизу вверх: - 1, 3 – аналоговые входы: - в режиме измерения напряжения 0...10 В, 11,5 кОм или - в режиме токовой петли 4...20 мА; 150 Ом; - 5, 7 – аналоговые выходы: - в режиме напряжения 0...10 В, 0...20 мА или - в режиме токовой петли 4...20 мА, макс. 12 В; - 2, 4, 6, 8 – GND
–	Предохранитель	Предназначен для защиты портов GND от суммарного превышения тока 6 А	Сменный предохранитель 6 А, корпус 5×20

№ разъема	Наименование разъема	Описание	Характеристики
—	Световая индикация	Светодиоды для передачи информации о состоянии или процессе работы манипулятора	Назначение снизу вверх: - Processing unit – индикатор состояния вычислительного блока (постоянное свечение зеленым – нормальное функционирование вычислительного блока манипулятора; отсутствует свечение – вычислительный блок отключен); - Robot power – в процессе: - запуска системы: индикатор успешной инициализации (постоянное свечение зеленым – вычислительный блок запущен, инициализация периферии прошла успешно); - работы комплекса: индикатор наличия напряжения на манипуляторе (прерывистое свечение раз в секунду – напряжение подано на манипулятор); - System status – индикатор состояния платы безопасности (постоянное свечение зеленым – загрузчик запущен на плате; прерывистое свечение – наличие ошибок на плате безопасности; отсутствует свечение – отсутствие ошибок на плате безопасности или запущен процесс обновления ПО на плате безопасности); - Emergency button – индикатор состояния аварийной кнопки на пульте управления (постоянное свечение зеленым – кнопка деактивирована (состояние нормально замкнутое); отсутствует свечение – кнопка активирована (состояние разомкнутое))
—	HDMI Монитор	Разъем для подключения внешнего дисплея	Тип: А; Стандарт: 1.4a

№ разъема	Наименование разъема	Описание	Характеристики
—	USB	Порт для подключения внешних периферийных устройств	Стандарт: USB 2.0; Тип: Type-A; Ток: 500 мА
—	Ethernet	Сетевой интерфейс для подключения к внешней локальной сети	Стандарт: 100BASE-TX; Скорость: 100 Мбит/с; Разъем: 8P8C

2.2.2.4 Схемы подключения датчиков и исполнительных механизмов к портам

Для подключения готовых датчиков или исполнительных механизмов используются:

- порты (входы и выходы) на пользовательской панели шкафа управления (см. Рисунок I-19);

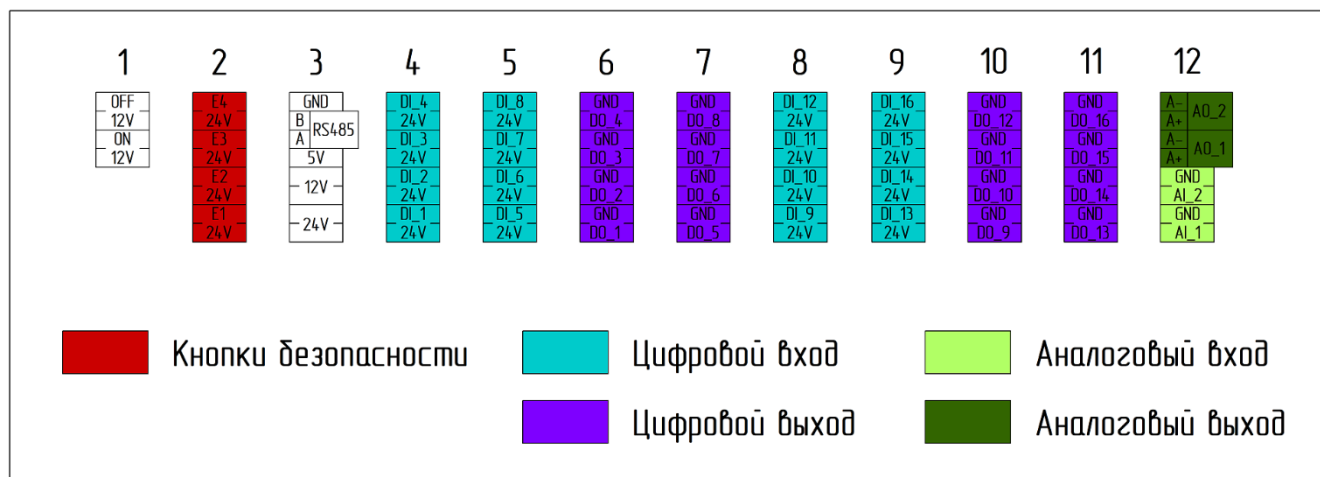


Рисунок I-19 – Порты на пользовательской панели шкафа управления

- порты (входы и выходы) на фланце инструмента (см. Рисунок I-20);

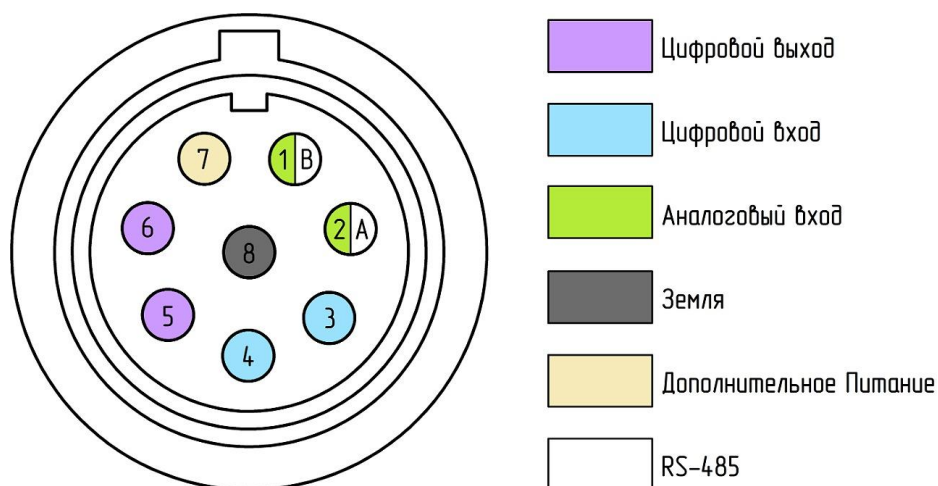


Рисунок I-20 – Порты на фланце инструмента

Типовая схема подключения к цифровому входу:

- Подключение датчика типа «Сухой контакт» (см. Рисунок I-21):

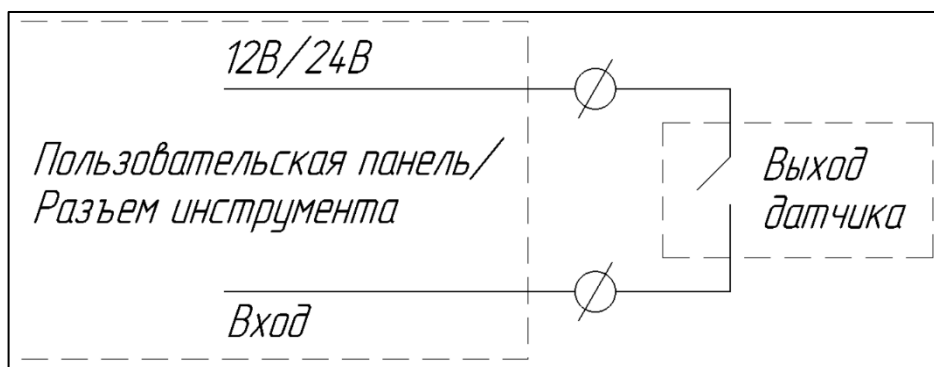


Рисунок I-21 – Подключение датчика типа «Сухой контакт»

Типовые схемы подключения к аналоговому входу:

- Подключение датчика с выходом по напряжению (см. Рисунок I-22).

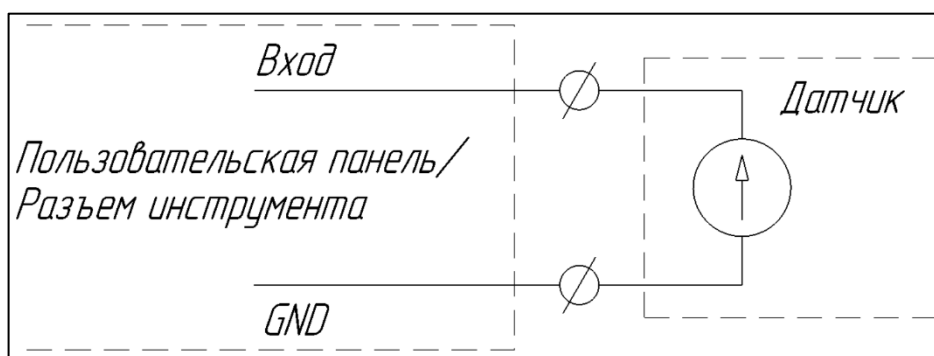


Рисунок I-22 – Схема подключения датчика с выходом по напряжению

Подключение датчика с выходом типа «токовая петля» (см. Рисунок I-23).

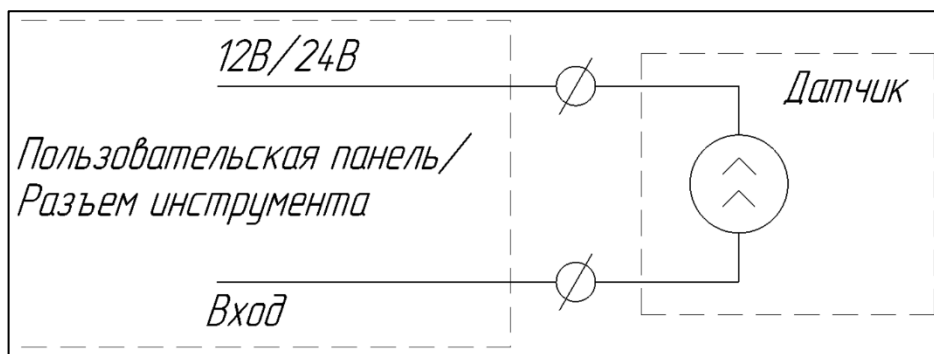


Рисунок I-23 – Схема подключения датчика с выходом типа «токовая петля»

Типовые схемы подключения к цифровым выходам:

- Подключение исполнительного механизма с цифровым дискретным интерфейсом (см. Рисунок I-24).

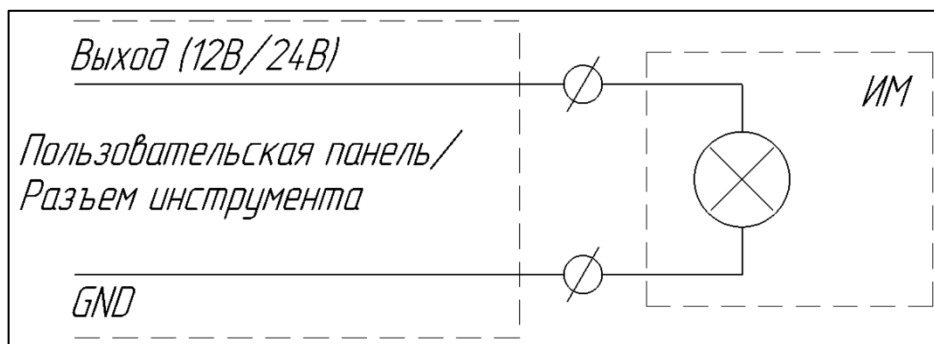


Рисунок I-24 – Схема подключения исполнительного механизма с цифровым дискретным интерфейсом

- Подключение исполнительного механизма с пневмоинтерфейсом (см. Рисунок I-25)

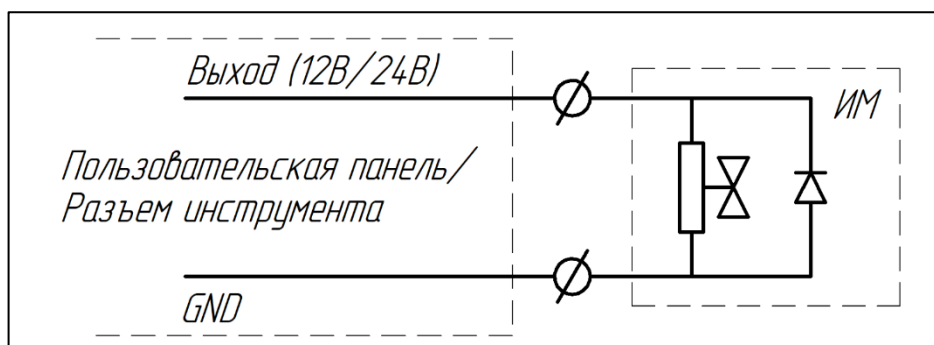


Рисунок I-25 – Схема подключения исполнительного механизма с пневмоинтерфейсом

Примечание – Рекомендуется использовать защитный диод при подключении механизмов с пневмоинтерфейсом (индуктивная нагрузка).

Типовые схемы подключения к аналоговым выходам:

- Подключение исполнительного механизма со входом типа «токовая петля» (см. Рисунок I-26).

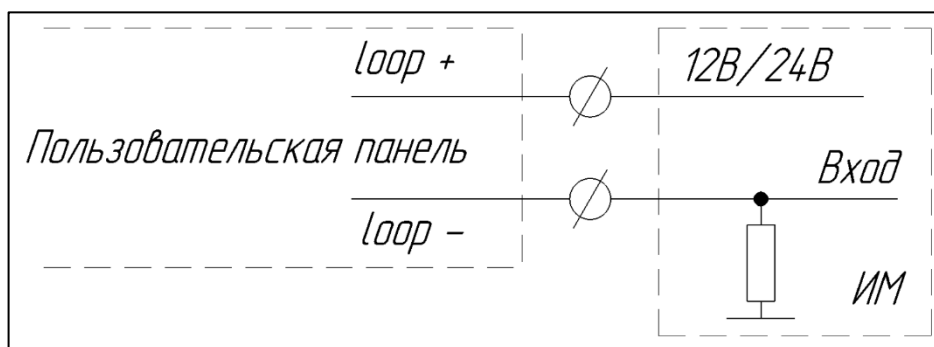


Рисунок I-26 – Схема подключения исполнительного механизма со входом типа «токовая петля»

- Подключение исполнительного механизма со входом по напряжению (см. Рисунок I-27)

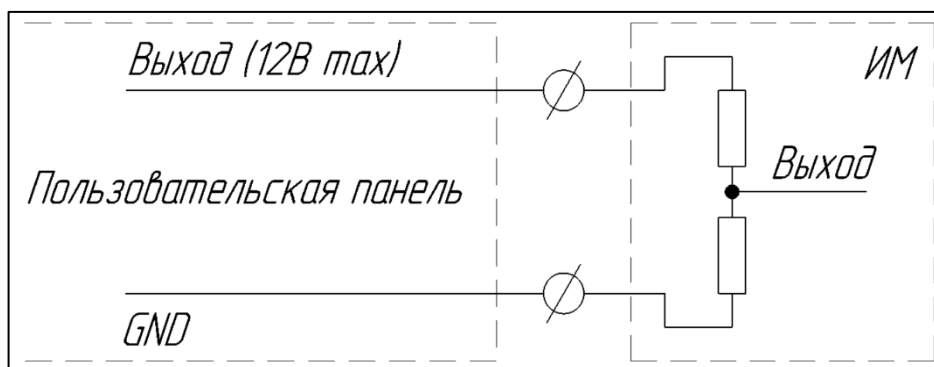


Рисунок I-27 – Схема подключения исполнительного механизма со входом по напряжению

Типовая схема подключения к интерфейсу RS-485 (см. Рисунок I-28):

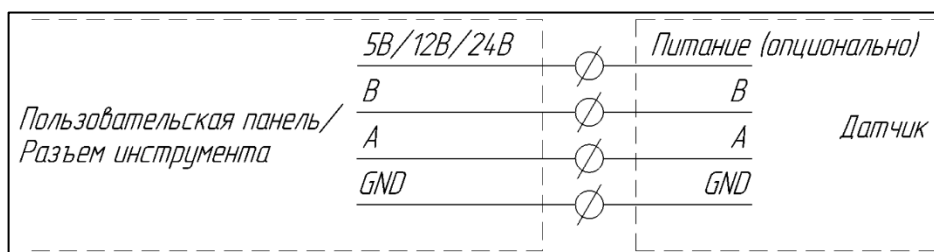


Рисунок I-28 – Схема подключения интерфейса RS-485

2.3 Использование Promobot M13

2.3.1 Включение

Перед включением Promobot M13 выполните следующие действия:

- а) убедитесь, что в рабочем окружении манипулятора отсутствуют посторонние предметы, которые могут помешать манипулятору выйти из транспортного положения и/или калибровке манипулятора;
- б) проверьте, что в рабочем окружении манипулятора находится только оператор, отсутствуют посторонние;
- в) убедитесь, что кнопка аварийного останова на пульте управления и кнопки аварийного останова, подключенные к шкафу управления, находятся в положении «ВКЛ» (см. раздел 4.4.2 «Кнопки аварийного останова»);
- г) убедитесь, что к шкафу управления подключены кабель питания и кабель манипулятора.

При штатном включение Promobot M13 выполните следующие действия:

- а) подключите кабель питания к сети;
- б) откройте шкаф управления и переведите автоматический выключатель в положение «ВКЛ»;
- в) приведите кнопку аварийного останова на пульте управления в рабочее положение;
- г) нажмите кнопку включения на пульте управления:
 - шкаф управления издаст звуковой сигнал;
 - на пульте управления отобразится процесс загрузки управляющего ПО;
 - после загрузки управляющего ПО отобразится окно для входа в приложение (см. Рисунок I-29);

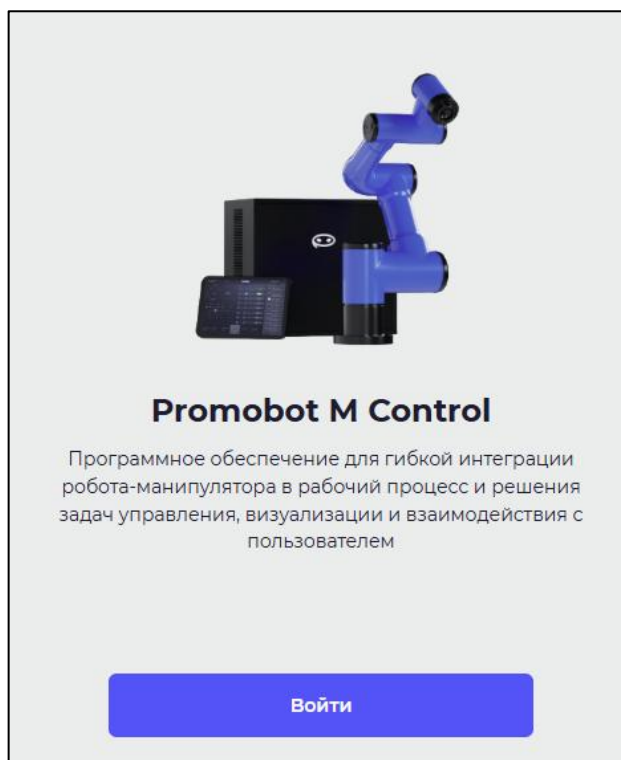


Рисунок I-29 – Окно для входа в приложение

- д) нажмите кнопку «Войти» и авторизуйтесь в приложении;
- е) подключите манипулятор (см. раздел II.2 «АВТОРИЗАЦИЯ, ПОДКЛЮЧЕНИЕ И ВЫКЛЮЧЕНИЕ МАНИПУЛЯТОРА В ПРИЛОЖЕНИИ»).

2.3.2 Режимы работы

Promobot M13 предусматривает следующие режимы работы:

- ручное перемещение с пульта управления;
- автоматическое перемещение по заданному алгоритму.

Манипулятор предназначен для осуществления движения и операций инструмента по заданному алгоритму (траектория движения, события). В качестве источника событий могут выступать цифровые, аналоговые входы шкафа управления и фланца инструмента манипулятора (см. раздел 2.2.2.3 «Подключение внешних устройств»).

Перед началом работы по заданному алгоритму пользователь должен настроить рабочее окружение манипулятора для исключения столкновений с рабочим окружением.

В н и м а н и е! Информация об изменении рабочего окружения манипулятора автоматически не обновляется.

Манипулятор предназначен для безопасной работы в одном рабочем пространстве с человеком, однако не имеет возможности для перестроения, заданной алгоритмом траектории. При столкновении с посторонним предметом или человеком манипулятор совершает аварийную остановку.

Для оптимизации производительности (максимизации скорости работы) рекомендуется использование контура (клетки) безопасности.

2.3.3 Сингулярности

2.3.3.1 Общие сведения

Сингулярность – особая конфигурация кинематики манипулятора, при которой движение центральной точки инструмента (ЦТИ) и/или ориентации инструмента ограничивается (см. Рисунок I-30), а для выполнения требуемого перемещения могут потребоваться непропорционально большие скорости отдельных суставов.

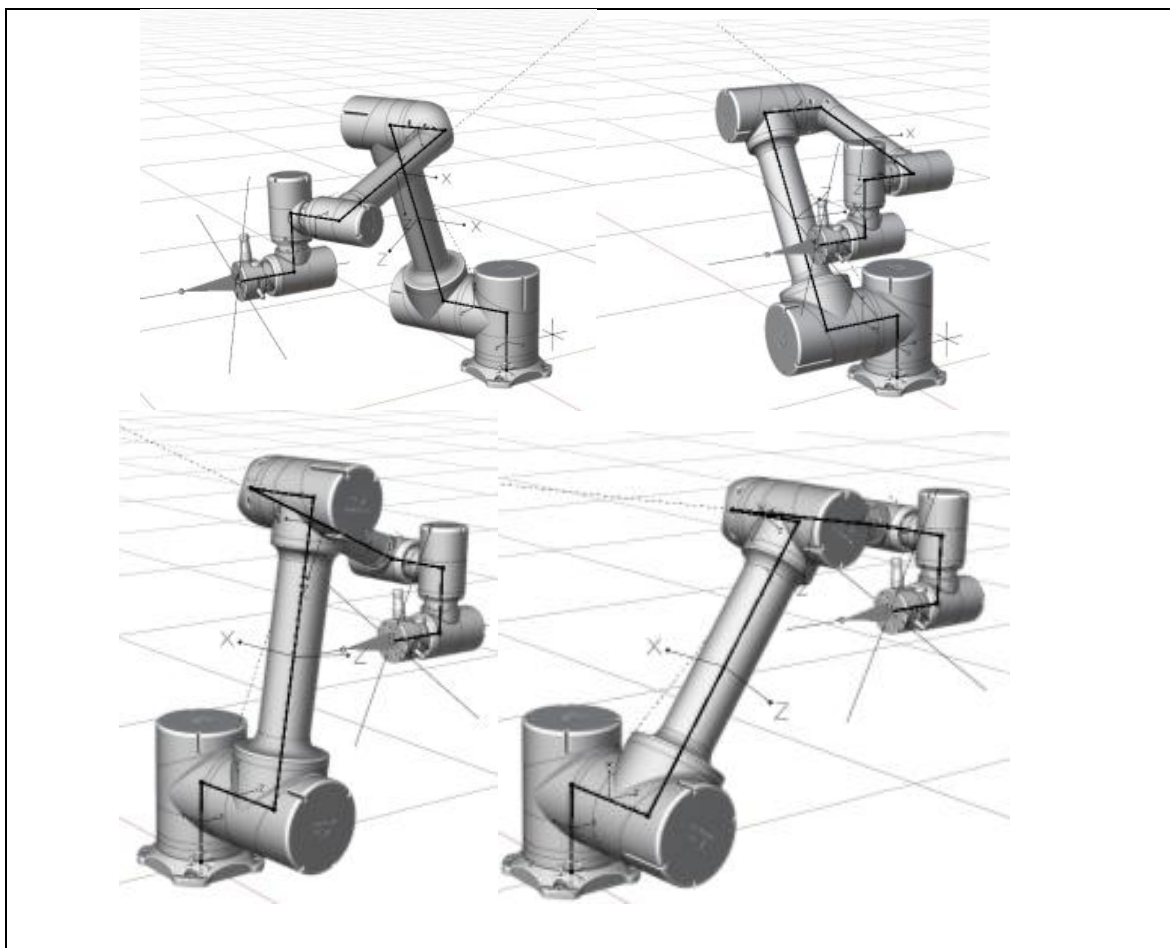


Рисунок I-30 – Пример конфигурации кинематики манипулятора, при приближении к сингулярности

При приближении к сингулярности или при выходе из нее возможны:

- резкое увеличение угловых скоростей отдельных суставов;
- затруднение или невозможность достижения требуемой позы и ориентации инструмента;
- прерывание перемещения по заданной траектории;
- непредсказуемое с точки зрения оператора изменение характера движения манипулятора.

Сингулярность не является неисправностью манипулятора, однако относится к потенциально опасным режимам движения и должна учитываться при ручном управлении, наладке и создании программ.

2.3.3.2 Типовые сингулярные положения

Для Promobot M13 наиболее характерны следующие случаи приближения к сингулярности:

- сингулярность внешней границы рабочей зоны (см. Рисунок I-31) – возникает при работе вблизи предельного вылета манипулятора;

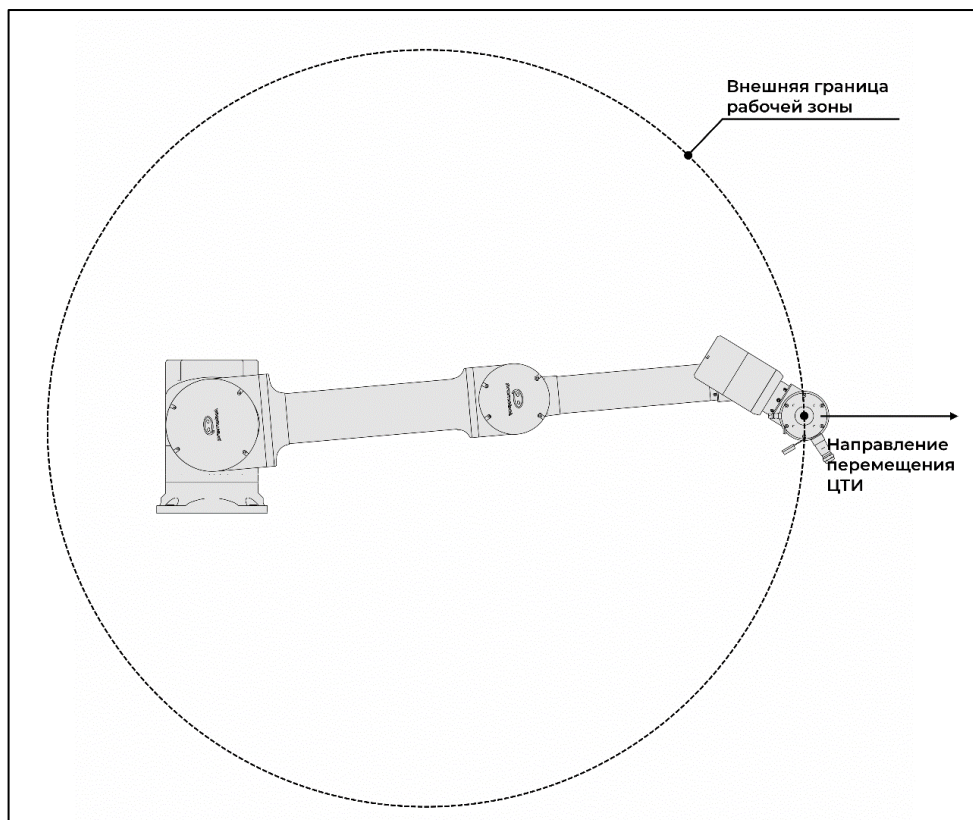


Рисунок I-31 – Сингулярность внешней границы рабочей зоны

- сингулярность внутренней зоны у основания (см. Рисунок I-32) – возникает при работе вблизи оси основания манипулятора, в том числе при перемещениях непосредственно над или под основанием, когда часть поз и ориентаций становится недостижимой или требует резкого перераспределения движений по суставам;

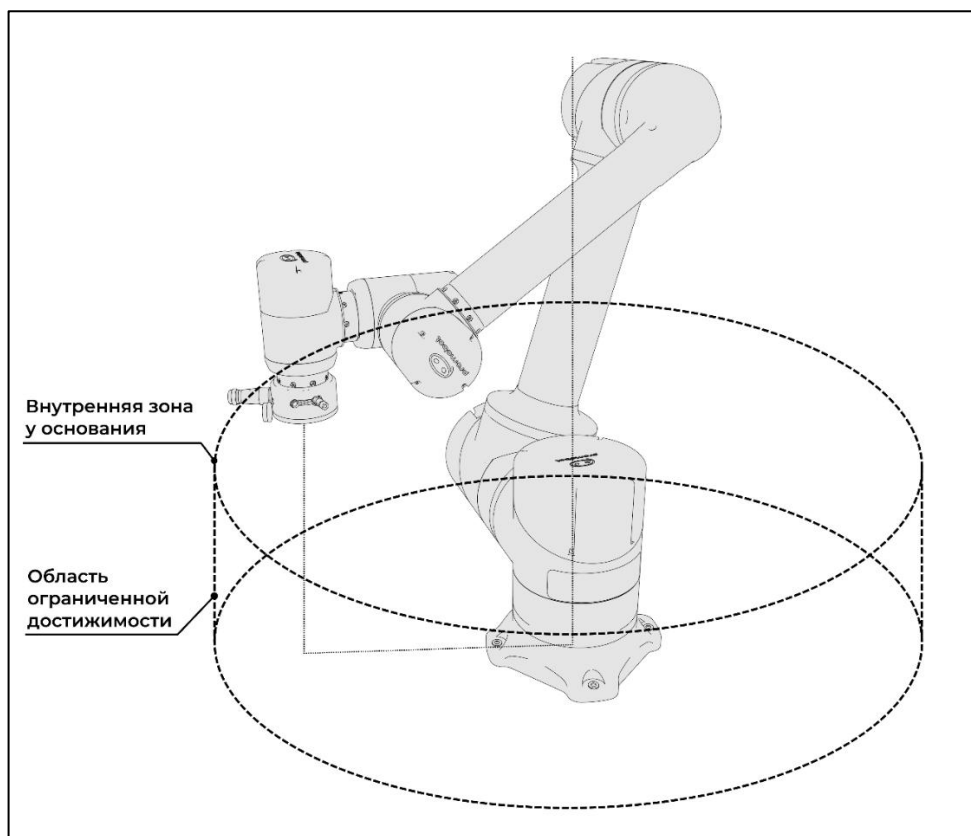


Рисунок I-32 – Сингулярность внутренней зоны у основания

- сингулярность запястья (см. Рисунок I-33) – возникает при таком взаимном положении запястий, при котором ориентация инструмента требует выравнивания осей СП и приводит к резкому росту скоростей СП4–СП6.

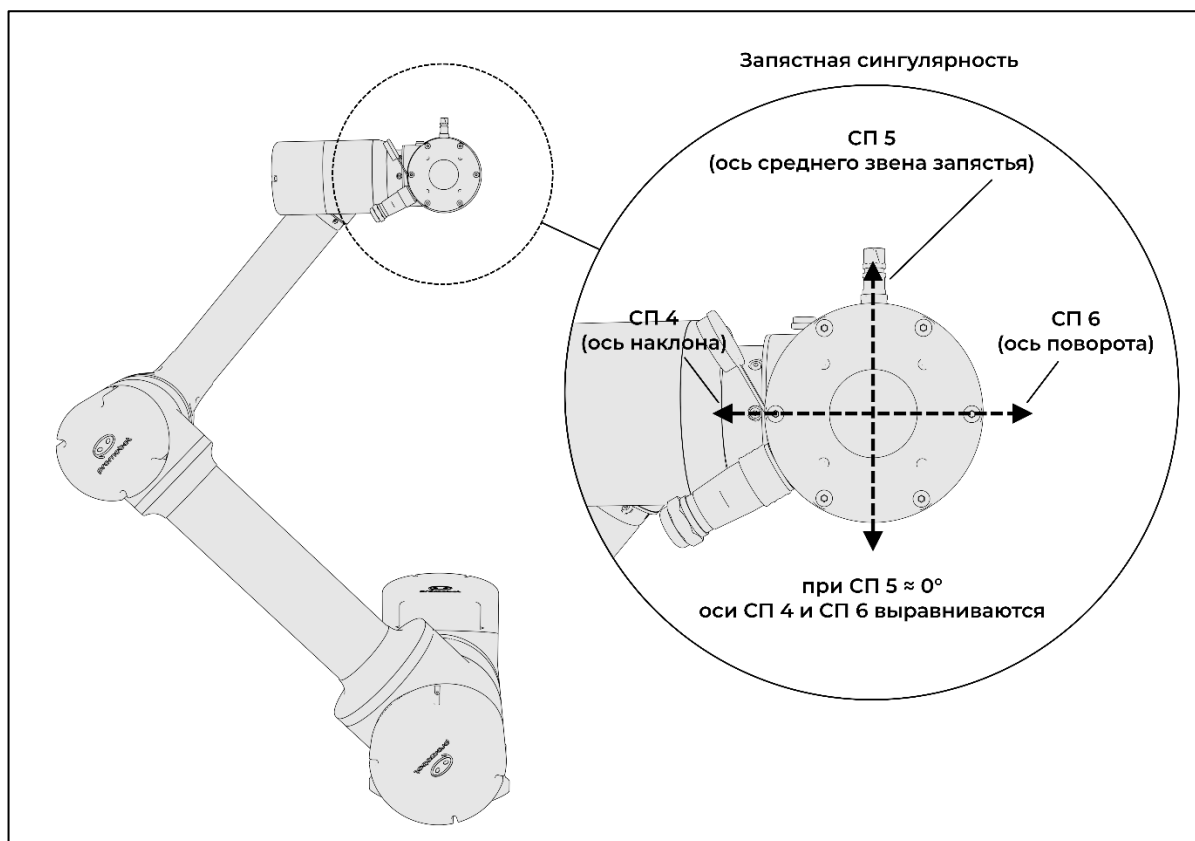


Рисунок I-33 – Сингулярность запястья

2.3.3.3 Признаки приближения к сингулярности

Признаками приближения к сингулярности являются:

- необходимость значительного перемещения одного или нескольких суставов при малом смещении ЦТИ;
- затруднение достижения манипулятором требуемой ориентации инструмента;
- появление предупреждений в приложении Promobot M Control.

При ручном управлении в приложении Promobot M Control отображаются следующие сообщения:

- «Манипулятор находится в сингулярности»;
- «Манипулятор приближается к сингулярности»;
- «Манипулятор отдаляется от сингулярности».

2.3.3.4 Требования безопасности при ручном управлении

При ручном управлении манипулятором соблюдайте следующие требования безопасности:

- перед началом движения убедитесь, что рабочая зона свободна от людей, инструмента, заготовок и иных посторонних предметов;
- выполняйте перемещения с минимально достаточной скоростью и минимальным шагом, особенно при изменении ориентации инструмента;
- при приближении к сингулярности предпочтительно использовать управление по суставам СП1–СП6, а не продолжать перемещение только по координатам ЦТИ;
- не допускайте нахождения частей тела в зоне возможного резкого поворота звеньев манипулятора, инструмента и заготовки;
- не прикладывайте усилие к манипулятору вне предусмотренных режимов ручного перемещения.

2.3.3.5 Требования безопасности при создании программ

При создании и отладке программ соблюдайте следующие требования безопасности:

- по возможности исключайте траектории, проходящие вблизи предельного вылета манипулятора, вблизи оси основания и в положениях выравнивания запястья;
- если линейное перемещение не является обязательным, используйте переход через безопасную промежуточную позу по суставам;
- при необходимости изменяйте ориентацию инструмента или добавляйте промежуточные точки, чтобы исключить прохождение через сингулярную область;
- после изменения траектории обязательно повторно проверяйте движение манипулятора в безопасном режиме;
- первый запуск новой или измененной программы выполняйте на пониженной скорости и при свободной рабочей зоне.

2.3.3.6 Действия оператора при предупреждении о сингулярности

При появлении предупреждения «Манипулятор приближается к сингулярности» оператору необходимо выполнить следующие действия:

- снизить скорость перемещения или остановить текущее движение;
- оценить положение манипулятора и убедиться в отсутствии людей и препятствий в рабочей зоне;
- изменить траекторию движения, ориентацию инструмента или выполнить отвод манипулятора в безопасное положение;
- при ручном управлении перейти к управлению по суставам и вывести манипулятор из опасной конфигурации малыми перемещениями.

При появлении сообщения «Манипулятор находится в сингулярности» оператору необходимо выполнить следующие действия:

- немедленно остановить текущее перемещение;
- вывести манипулятор из сингулярного положения с помощью последовательных малых перемещений по суставам;
- не возобновлять выполнение автоматического цикла до устранения причины возникновения сингулярности и проверки измененной траектории.

2.3.3.7 Запрещается

При предупреждении о сингулярности запрещается:

- игнорировать предупреждения и сообщения о сингулярности;
- многократно повторять одно и то же перемещение без изменения траектории или ориентации инструмента после возникновения предупреждения или ошибки;
- находиться в рабочей зоне манипулятора при выводе его из сингулярного положения.

2.3.3.8 Остаточный риск

Даже при соблюдении требований настоящего руководства вблизи сингулярных положений возможно быстрое изменение скоростей отдельных суставов и изменение

характера движения манипулятора. Оператор обязан учитывать данный остаточный риск при ручном управлении, наладке и программировании.

2.3.4 Выключение

При штатном выключении Promobot M13 выполните следующие действия:

- а) выключите манипулятор в приложении (см. Рисунок II-8);
- б) нажмите кнопку аварийного останова на пульте управления;
- в) откройте шкаф управления и переведите автоматический выключатель в положение «ВЫКЛ»;
- г) при необходимости полного обесточивания манипулятора – отсоедините кабель питания от питающей сети.

2.4 Демонтаж Promobot M13

Перед демонтажем Promobot M13 выполните следующие действия:

- а) демонтируйте инструмент и оснастку, закрепленную на манипуляторе.
- б) переведите манипулятор в транспортное положение (см. раздел II.5.5.1 «Транспортное положение»).
- в) выключите Promobot M13 (см. раздел 2.3.4 «Выключение»);

При демонтаже Promobot M13 выполните следующие действия:

- а) зафиксируйте манипулятор, чтобы предотвратить падение и повреждение оборудования;
- б) демонтируйте манипулятор с места установки, используя круглопрядные текстильные стропы длиной 2 метра:
 - 1) обхватите стропами плечо СП2–СП3 мертвой петлей как указано на рисунке (см. Рисунок I-12) (нельзя подвергать усилиям запрещенные места обхвата (СП4, СП5, СП6) и пластиковые защитные крышки (см. Рисунок I-12));
 - 2) открутите крепление основания манипулятора и снимите манипулятор с места установки;
 - 3) поставьте манипулятор основанием на ровный пол;

- 4) тщательно очистите манипулятор от пыли, грязи, используйте мягкие ткани и чистящие средства (см. раздел 3.1 «Регламентное техническое обслуживание Promobot M13»);
- 5) переместите манипулятор на место упаковки, используя подкатную тележку.

При отправке Promobot M13 на хранение соблюдайте инструкцию, приведенную в разделе 7.2 «Подготовка к хранению».

3 ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

Техническое обслуживание и ремонт Promobot M13 предназначены для обеспечения заявленных технических характеристик комплекса на протяжении всего срока эксплуатации.

3.1 Регламентное техническое обслуживание Promobot M13

Регулярное техническое обслуживание манипулятора представлено в таблице 7.

Таблица 7 – Техническое обслуживание манипулятора

Периодичность	Параметр проверки	Способ проверки
Ежедневно	Визуальный осмотр манипулятора	Обойдите манипулятор со всех сторон, проведите визуальный осмотр на наличие трещин, повреждений, дефектов и загрязнений
Ежедневно	Проверка кнопки аварийной остановки	Нажмите кнопку аварийной остановки на пульте управления во время паузы, должно отобразиться уведомление на экране
Ежедневно	Тест безопасного запуска в режиме низкой скорости	Выполните любой скрипт на медленной скорости (коэффициент 0,1)
Ежемесячно	Проверка креплений и основания манипулятора	При обесточенном манипуляторе попробуйте вручную сдвинуть манипулятор за основание. При наличии люфтов протяните крепления опоры
Ежемесячно	Осмотр кабелей и разъемов на износ	Визуально проведите контроль кабелей и проводки. Не допускается наличие трещин, порезов, пропусков изоляции
Ежемесячно	Очистка и проверка захвата/инструмента	Отсоедините фланец инструмента и проведите его осмотр и очистку от механических загрязнений

Периодичность	Параметр проверки	Способ проверки
Ежемесячно	Проверка калибровки энкодеров и датчиков положения	Выполните полный запуск манипулятора, должны отсутствовать ошибки
Раз в 3–6 месяцев	Диагностика редукторов на люфт и шум	При обесточенном манипуляторе поочередно попробуйте вручную сдвинуть манипулятор за плечо каждого СП и проверьте на наличие люфтов. При наличии люфтов работа на манипуляторе запрещается
		Запустите любой скрипт на манипуляторе. При наличии нехарактерных шумов, локализируйте место возникновения шума, обесточьте манипулятор и обратитесь в сервисную службу. Работа на манипуляторе с данным дефектом запрещается
Раз в 3–6 месяцев	Проверка состояния приводов (температура, вибрации, шумы)	При обнаружении нехарактерных вибраций, шумов и прерывистости движений, остановите работу манипулятора. Замерьте пирометром температуру узлов манипулятора, зафиксируйте данные и передайте в сервисную службу. Работа на манипуляторе с вышеперечисленными дефектами запрещается
Ежегодно	Замена изношенных деталей захвата и уплотнителей	Детали, имеющие характерный износ, отрицательно влияющие на работоспособность необходимо заменить у предприятия-изготовителя или сертифицированного сервиса
Ежегодно	Обновление прошивки контроллера и ПО	См. раздел II.7 «ОБНОВЛЕНИЕ ПРИЛОЖЕНИЯ»
Ежегодно	Калибровка всех датчиков и проверка точности позиционирования	Обратитесь в сервисную службу предприятия-изготовителя

3.2 Диагностика неисправностей Promobot M13

Возможные неисправности, их причины и методы решения представлены в таблице 8.

Таблица 8 – Диагностика неисправностей

Неисправность	Возможная причина	Методы устранения
Электромеханические неисправности		
Не включается экран пульта управления	Отключен автоматический выключатель	Включить автоматический выключатель в шкафу управления
	Не подсоединен кабель пульта к шкафу управления	Проверить подключения
	Неисправность кабеля пульта	Заменить кабель
	Не запускается шкаф управления манипулятором	Обратиться в сервисную службу
	Не поступает напряжение сети 230 В	Заменить неисправный кабель питания
При работе манипулятора слышно посторонние звуки при работе сервоприводов	Неисправность редуктора, драйвера мотора	Обратиться в сервисную службу
Визуальное замедление интерфейса на пульте управления, зависание	Перегрев шкафа управления	Проверить работоспособность электрического вентилятора

виртуальных переключателей, открытых диалоговых окон интерфейса управления. Самопроизвольное выключение манипулятора. Зависание исполнительной программы	Воздушный фильтр шкафа управления забит пылью	Продуть вкладыш либо заменить на новый
	Шкаф установлен неправильно, вентиляционные отверстия закрыты	Освободить место вокруг шкафа
	Температура окружающей среды превышает рабочую температуру	Снизить температуру окружающей среды до допустимых значений
	Неисправен вентилятор материнской платы	Заменить вентилятор
Щелкающие звуки на одном из сервоприводов во время работы	Аварийный останов категории 0. Неисправность тормоза	Обратиться в сервисную службу
Уведомление об ошибке в ПО Promobot M Control	—	Обратиться в сервисную службу

3.3 Рекомендации по ремонту Promobot M13

В случае сложных неисправностей или отсутствия необходимых навыков обратитесь в сервисный центр предприятия-изготовителя.

Не пытайтесь ремонтировать электронные компоненты или программное обеспечение самостоятельно.

Если манипулятор находится на гарантии, ремонт должен выполняться только уполномоченными специалистами предприятия-изготовителя.

Сохраните паспорт Promobot M13 и документы, подтверждающие дату покупки.

3.4 Хранение запасных частей Promobot M13

Для оперативного устранения неисправностей рекомендуется иметь набор запасных частей: вкладыши для фильтров (12 шт.), резиновые уплотнительные кольца.

Храните запасные части в сухом и чистом месте, защищенном от влаги и пыли.

3.5 Регистрация обслуживания и ремонта Promobot M13

Все проведенные работы по техническому обслуживанию и ремонту должны регистрироваться в журнале учета. Журнал должен содержать следующие графы:

- дата проведения работ;
- фактическая наработка изделия с начала эксплуатации на день проведения технического обслуживания или ремонта, ч;
- вид технического обслуживания или ремонта и краткое описание выполненных работ;
- использованные материалы и запасные части;
- подпись и ФИО лица, производившего работы по обслуживанию;
- подпись и ФИО лица, разрешающего пуск отремонтированного/обслуженного оборудования, дата пуска.

В случае возникновения вопросов или сложностей обратитесь в сервисный центр предприятия-изготовителя.

4 МЕРЫ БЕЗОПАСНОСТИ ПРИ ИСПОЛЬЗОВАНИИ

4.1 Общие требования безопасности

Работа с Promobot M13 требует строгого соблюдения мер безопасности. Несоблюдение мер может привести к травмам персонала, повреждению комплекса и другим негативным последствиям. Следуйте приведенным ниже требованиям безопасности, чтобы минимизировать риски:

- убедитесь, что манипулятор и инструмент смонтированы согласно инструкции;
- убедитесь, что рабочее окружение манипулятора соответствует алгоритму движения;
- убедитесь, что проводные подключения манипулятора, шкафа и пульта управления выполнены согласно инструкции;

- убедитесь, что к шкафу управления подключены все датчики и исполнительные механизмы, от которых зависит алгоритм движения манипулятора;
- убедитесь, что кнопки аварийного останова расположены в доступных для персонала местах;
- убедитесь, что персонал осведомлен о расположении кнопок аварийного останова и методах их активации в случае возникновения нештатных ситуаций;
- убедитесь, что крышка шкафа управления и крышки сервоприводов манипулятора закрыты;
- убедитесь, что соблюдены все меры и техника безопасности при перемещении манипулятора, если управление происходит с помощью пульта управления;
- при возникновении в процессе эксплуатации сообщений об ошибках следуйте инструкциям управляющего программного обеспечения;
- избегайте столкновений с манипулятором;
- запрещается носить свободную одежду или ювелирные изделия при работе с манипулятором;
- запрещается использовать комплекс при наличии механических повреждений корпусов манипулятора, шкафа управления, пульта управления, соединительных кабелей, крепежей;
- запрещается вносить изменения в Promobot M13. Любые изменения могут привести к появлению опасностей, не предусмотренных предприятием-изготовителем.

4.2 Требования к оператору

Оператор, работающий с Promobot M13, должен:

- пройти обучение по эксплуатации комплекса и ознакомиться с руководством по эксплуатации;
- иметь навыки работы с промышленным оборудованием и системами управления;

- соблюдать требования безопасности, указанные в разделе 4 «МЕРЫ БЕЗОПАСНОСТИ ПРИ ИСПОЛЬЗОВАНИИ» настоящего руководства.

При эксплуатации Promobot M13 оператор обязан:

- выполнять требования эксплуатационной документации, правил электро- и пожарной безопасности;
- не допускать расположения кабелей питания и интерфейсных кабелей в местах, где им могут быть нанесены механические повреждения;
- при появлении посторонних шумов прекратить работу и обесточить комплекс;
- при возникновении в процессе эксплуатации сообщений об ошибках следовать инструкциям управляющего программного обеспечения;
- своевременно проводить техническое обслуживание комплекса согласно инструкции.

Запрещается:

- производить действия, противоречащие эксплуатационной документации на комплекс;
- работать во взрывоопасной среде, рядом с легковоспламеняющимися предметами;
- производить монтаж и демонтаж электрооборудования без выключения комплекса;
- открывать и разбирать корпус манипулятора;
- модифицировать манипулятор (изменять или удалять элементы конструкции);
- производить ремонт комплекса самостоятельно;
- использовать комплекс не по назначению.

4.3 Рабочее окружение манипулятора

Рабочее окружение манипулятора должно соответствовать алгоритму движения манипулятора.

Перед началом работы убедитесь, что в рабочем окружении манипулятора отсутствуют изменения, которые могут привести к столкновениям и сбоям в алгоритме движения.

При наладке устройства используйте режим с ограниченной скоростью: это позволит избежать резких движений и снизит риск травмирования.

При программировании манипулятора убедитесь, что вы можете быстро остановить устройство в случае необходимости: пути отступления не должны быть заблокированы.

Перед запуском программы на полной скорости проведите тестирование на пониженной скорости. Убедитесь, что манипулятор движется по заданной траектории без отклонений.

Запрещается подлезать под движущиеся оси манипулятора, поправлять заготовки руками во время работы программы и находиться в рабочей зоне без необходимости.

4.4 Действия в аварийных ситуациях

4.4.1 Категории останова

В соответствии с ГОСТ МЭК 60204-1 категории останова, регламентирующие алгоритмы действий персонала при возникновении нештатных ситуаций в соответствии с классом риска:

- Категория 0: аварийный останов. Применяется при возникновении опасных и/или аварийных ситуаций. Сопровождается снятием питания с манипулятора и блокировкой узлов сервоприводов механическим тормозом.

Внимание! Выход из аварийного останова возможен только с последующей рекалибровкой манипулятора

- Категория 1: контролируемый останов. Применяется при возникновении некоторых категорий ошибок, а также при пусконаладочных работах. Сопровождается блокировкой узлов сервоприводов механическим тормозом.

Внимание! Выход из контролируемого останова возможен только с последующей рекалибровкой манипулятора

- Категория 2: пауза. Применяется в качестве планового или внепланового останова при нормальной работе манипулятора. Не сопровождается блокировкой узлов сервоприводов механическим тормозом.

4.4.2 Кнопки аварийного останова

Кнопки аварийного останова являются важным элементом системы безопасности и предназначены для экстренной остановки оборудования в аварийных ситуациях:

Аппаратные кнопки аварийного останова

Основная кнопка аварийного останова расположена на пульте управления. Кнопка является нормально замкнутой: в отжатом состоянии цепь скоммутирована. При нажатии на кнопку цепь размыкается.

К шкафу управления можно подключить до двух нормально замкнутых дополнительных кнопок аварийного останова.

Используйте для данного типа кнопок аварийного останова контакты 1–2, 3–4.

Все аппаратные кнопки аварийного останова подключены последовательно. В случае срабатывания (размыкания) хотя бы одной кнопки, с манипулятора снимается питание и узлы сервопривода манипулятора блокируются механическими тормозами.

Цифровые кнопки аварийного останова

К шкафу управления можно подключить до двух цифровых кнопок аварийного останова.

Используйте для данного типа кнопок аварийного останова контакты 5–6, 7–8.

Внимание! Если кнопки аварийного останова, подключаемые к шкафу управления, не используются – требуется установить четыре перемычки в разъемы подключения кнопок между контактами 1–2, 3–4, 5–6 и 7–8 (см. Рисунок I-34).

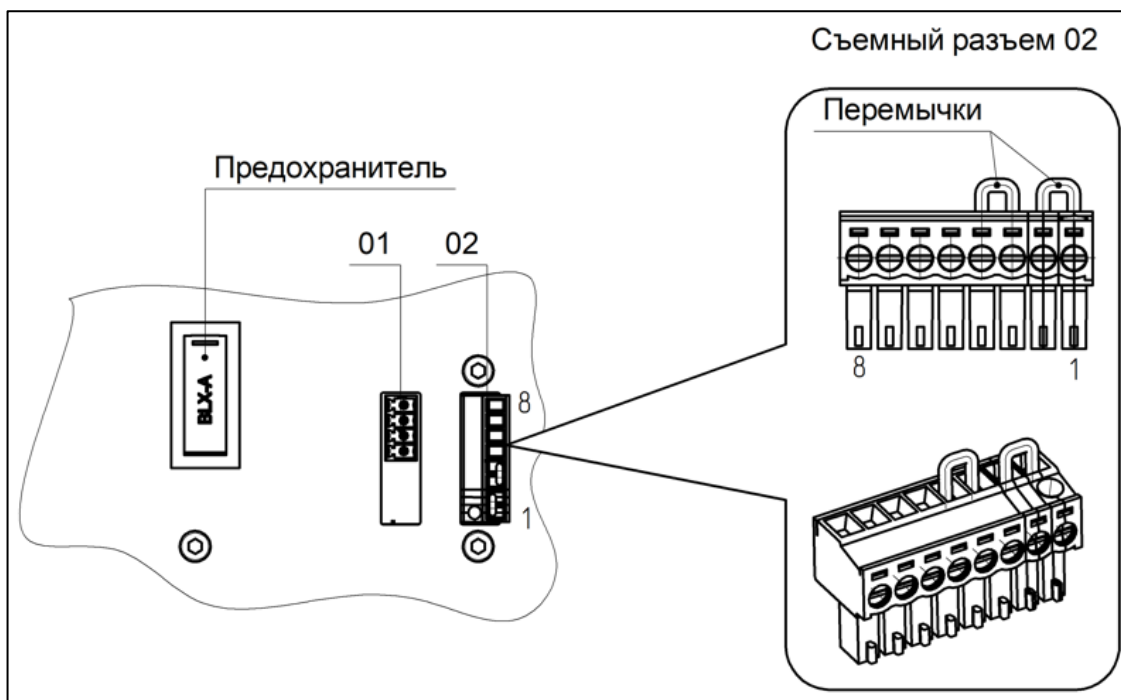


Рисунок I-34 – Установка перемычек между контактами 1–2, 3–4, 5–6, 7–8

4.4.3 Аварийный останов

Аварийный останов сопровождается отключением питания и блокировкой манипулятора. В случае возникновения аварийного останова следуйте инструкциям управляющего ПО на экране пульта управления.

5 БЕЗОПАСНОСТЬ

5.1 Индикация Promobot M13

Шкаф управления оснащен встроенным звуковым излучателем для индикации состояний Promobot M13.

Элементы индикации также могут быть подключены к цифровым выходам шкафа управления. Использование данных элементов индикации будет определяться пользовательским алгоритмом.

Внимание! Встроенная звуковая индикация является элементом безопасной эксплуатации Promobot M13 и не может быть отключена.

Описание встроенной звуковой индикации состояний приведено в таблице 9.

Таблица 9 – Встроенная звуковая индикация состояний

Ситуация	Цель сигнала	Звуковой сигнал	Повторяемость
Включение шкафа управления	Оповещение: нормальная работа	Один длинный	Однократно
Выключение шкафа управления	Оповещение: нормальная работа	Один длинный	Однократно
Включение питания манипулятора	Предупреждение	Три длинных, три коротких	Однократно
Выключение питания манипулятора	Предупреждение	Три коротких, три длинных	Однократно
Останов категории 2	Предупреждение	Нет	Однократно
Останов категории 1	Предупреждение	Один длинный	Однократно
Останов категории 0	Авария	Один длинный	Однократно

6 ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА

Гарантийный срок предприятия- изготовителя составляет 12 (двенадцать) месяцев. Гарантийный срок исчисляется по общему правилу с даты приобретения Promobot M13 у предприятия-изготовителя. Гарантийный срок на Promobot M13, приобретенный розничным потребителем (гражданином, приобретающим Promobot M13 исключительно для личных или иных нужд, не связанных с осуществлением предпринимательской деятельности) исчисляется с даты приобретения Promobot M13 розничным потребителем.

При обнаружении в Promobot M13 недостатков в период гарантийного срока владелец обязуется незамедлительно письменно сообщить об этом предприятию-изготовителю, описав подробно признаки выявленного недостатка, и в течение одного рабочего дня обеспечить по требованию предприятия-изготовителя дистанционный доступ к Promobot M13 для дистанционной диагностики.

В период гарантийного срока предприятие-изготовитель обязуется при получении претензии обеспечить безвозмездное устранение недостатков Promobot M13 в срок не более 60 рабочих дней с момента получения предприятием-изготовителем претензии и, в случае необходимости, предоставления дистанционного доступа к Promobot M13. Предприятие-изготовитель вправе по своему выбору осуществить замену Promobot M13 ненадлежащего качества.

В случае существенного нарушения требований к качеству Promobot M13 (обнаружения неустраняемых недостатков, которые не могут быть устранены без несоразмерных расходов или затрат времени, или выявляются неоднократно, либо неоднократно проявляются вновь после их устранения) владелец вправе потребовать замены Promobot M13 ненадлежащего качества.

Гарантийный срок продлевается на время, в течение которого Promobot M13 не мог использоваться из-за обнаруженных в нем недостатков, а именно на период времени, равный периоду с момента получения уполномоченным лицом претензии о недостатках Promobot M13 до момента устранения недостатков.

Гарантийное обслуживание осуществляется на выбор предприятия-изготовителя по месту его нахождения или по месту нахождения Promobot M13.

Гарантийному ремонту (замене) не подлежит Promobot M13:

- эксплуатировавшийся способом, не соответствующим требованиям Руководства по эксплуатации;
- имеющий по вине пользователей механические повреждения, явившиеся причиной обращения за гарантийным ремонтом;
- эксплуатировавшийся или хранившийся в условиях (среде), не соответствующей требованиям, установленным документацией на Promobot M13;

- прошедший модификацию (изменения) или ремонт без участия компании-производителя.

Гарантийные обязательства распространяются на Promobot M13 в той комплектации, в которой он находился на момент поставки предприятием-изготовителем, и не распространяются на недостатки, возникшие в результате неверной работы ПО Promobot M Control, разработанного (доработанного) пользователем самостоятельно без согласования с предприятием-изготовителем.

Порядок осуществления гарантийного обслуживания (ремонта) Promobot M13 установлен Положением о гарантийном ремонте и проведении сервисного обслуживания оборудования ООО «ПРОМОБОТ», размещенного по ссылке: <https://promo-bot.ru/warranty-repair-and-service-provision/>.

В гарантийное обслуживание (ремонт) не включены дополнительные услуги, в том числе, загрузка информации на Promobot M13, доработка ПО, функционала, мониторинг состояния Promobot M13, не связанный с исправлением недостатков. Дополнительные услуги оказываются на основании отдельно заключенного возмездного соглашения, в частности Соглашения об уровне сервиса (SLA).

7 ХРАНЕНИЕ

Правильное хранение Promobot M13 обеспечивает сохранность его работоспособности и предотвращает повреждения в период, когда устройство не используется. В данном разделе описаны основные требования и рекомендации по хранению манипулятора.

7.1 Условия хранения Promobot M13

Для обеспечения сохранности манипулятора во время хранения необходимо соблюдать следующие условия:

- температура окружающей среды: от -10°C до +40°C;
- относительная влажность: не более 80%;
- Promobot M13 должен храниться в чистом помещении, защищенном от пыли, грязи и агрессивных химических веществ;

- место хранения Promobot M13 должно быть безопасным, без риска падения предметов или повреждения манипулятора;
- помещение должно быть защищено от прямых солнечных лучей, которые могут вызвать выцветание или повреждение материалов.

7.2 Подготовка к хранению Promobot M13

Подготовка к хранению включает в себя следующие действия:

- а) демонтаж комплекса согласно инструкции (см. раздел 2.4 «Демонтаж») (если манипулятор будет храниться длительное время, то рекомендуется демонтировать съемные компоненты и хранить их отдельно в упаковке).
- б) упаковка манипулятора в оригинальную упаковку для предотвращения повреждений;
- в) Проверка, что устройство в упаковке надежно защищено от влаги и пыли.

7.3 Периодическое обслуживание во время хранения Promobot M13

Периодическое обслуживание во время хранения включает в себя следующие действия:

- а) проведение осмотра манипулятора не реже одного раза в месяц для проверки его состояния;
- б) проверка упаковки на целостность – упаковка не должна быть повреждена и должна обеспечивать надежную защиту.

7.4 Подготовка к использованию после хранения Promobot M13

Перед возвращением манипулятора в эксплуатацию выполните действия, описанные в разделе 2 «ИСПОЛЬЗОВАНИЕ ПО НАЗНАЧЕНИЮ».

7.5 Хранение запасных частей и аксессуаров Promobot M13

Запасные части и аксессуары должны храниться в отдельных упаковках в сухом и чистом месте. Они должны быть защищены от влаги, пыли и механических повреждений.

8 ТРАНСПОРТИРОВАНИЕ

Правильная транспортировка Promobot M13 является важным этапом, обеспечивающим сохранность устройства и предотвращающим повреждения.

Транспортировка должна осуществляться в соответствии с правилами перевозки грузов, действующими на территории страны, а также с учетом рекомендаций предприятия-изготовителя:

- а) условия транспортирования и хранения Promobot M13, сроки сохраняемости до ввода в эксплуатацию должны соответствовать данным, указанным в таблице 10;
- б) расстановка и крепление упакованных Promobot M13 в транспортных средствах должны обеспечивать их устойчивое положение, исключать возможность смещения и ударов о стенки транспортных средств;
- в) транспортирование Promobot M13 следует производить в соответствии с правилами перевозки грузов, действующими на каждом виде транспорта;
- г) транспортирование Promobot M13 должно производиться любым видом транспорта, при условии соблюдения требований, установленных манипуляционными знаками, нанесенными на транспортной таре;
- д) в случае кратковременного транспортирования на открытых автомашинах упаковка Promobot M13 должна быть накрыта водонепроницаемым брезентом.

Таблица 10 – Условия транспортирования, хранения, сроки сохраняемости

Обозначение условий транспортирования в части воздействия		Обозначение условий хранения по ГОСТ 15150	Сроки сохраняемости в упаковке изготовителя, годы
механических факторов по ГОСТ 23216	климатических факторов по ГОСТ 15150		
С	УХЛ4	1Л	5

9 УТИЛИЗАЦИЯ

Утилизация Promobot M13 должна производиться в соответствии с Законами РФ № 96-ФЗ «Об охране атмосферного воздуха», № 89-ФЗ «Об отходах производства и потребления», № 52-ФЗ «О санитарно-эпидемиологическом благополучии населения», а также другими российскими и региональными нормами, актами, правилами, распоряжениями и прочими нормативными документами, принятыми в использование в указанных законах и эксплуатирующей организации.

II. ПРИЛОЖЕНИЕ PROMOBOT M CONTROL

1 ВВЕДЕНИЕ

Настоящий раздел руководства по эксплуатации описывает порядок работы с приложением Promobot M Control манипулятора Promobot M13. Приложение предназначено для управления, программирования и мониторинга работы манипулятора Promobot M13 через веб-интерфейс. Оно обеспечивает ручное управление в реальном времени и создание автоматизированных программ различной сложности.

Приложение обладает функционалом обновления, который позволяет пользователям всегда иметь актуальную версию с улучшенной производительностью, новыми функциями и исправленными ошибками.

В верхней части располагается главная панель вкладок, которая содержит (см. Рисунок II-1):



Рисунок II-1 – Главная панель вкладок

- Кнопки:
 - «Подключить» – кнопка подключения/отключения манипулятора.
 - «Инструкции» – кнопка для открытия руководства Promobot M13 на портале производителя.
 - «Выход» – кнопка для выключения M13.
- Вкладки:
 - «Ручное управление» – интерфейс для ручного управления манипулятором.
 - «Создание программ» – интерфейс для создания программ с помощью языков программирования Blockly, Python.
 - «Настройка» – интерфейс для настроек работы системы и инструмента.
 - «Мониторинг» – интерфейс для мониторинга работы Promobot M13.

В нижней части приложения располагается фиксированная нижняя панель, которая отображает наименование устройства и версию ПО (см. Рисунок II-2).



Рисунок II-2 – Фиксированная нижняя панель

Информация на панели используется для быстрой проверки состояния программной части. После подключения к реальному манипулятору отображается версия системного ПО. В режиме симуляции вместо номера версии отображается значение virtual, а для команд, которые нельзя выполнить без реального контроллера, выводится сообщение «Недоступно в режиме симуляции».

Перед выполнением любых действий убедитесь, что в нижней панели указано ожидаемое устройство «Promobot M13», а состояние кнопки подключения соответствует фактическому соединению с манипулятором.

Также по нажатию кнопки  открывается экранная клавиатура (см. Рисунок II-3).

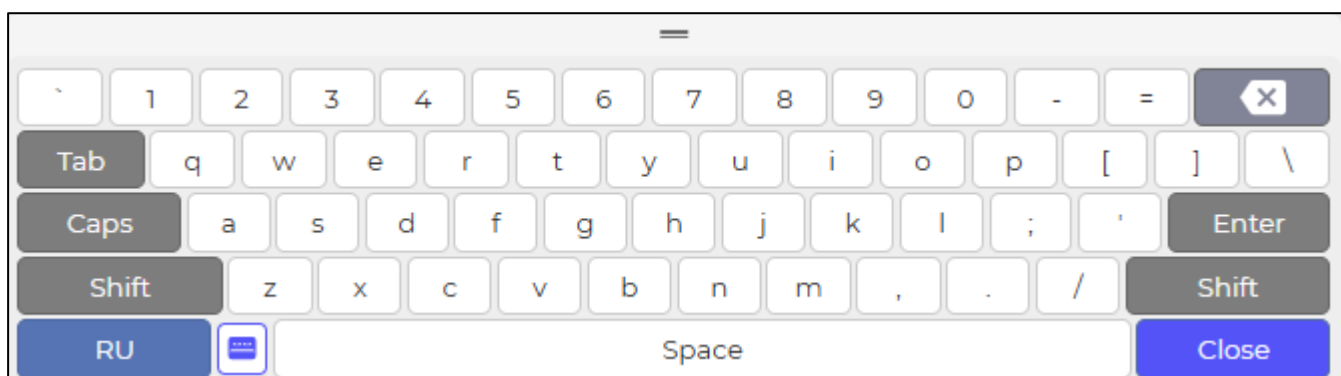


Рисунок II-3 – Экранная клавиатура

2 АВТОРИЗАЦИЯ, ПОДКЛЮЧЕНИЕ И ВЫКЛЮЧЕНИЕ МАНИПУЛЯТОРА В ПРИЛОЖЕНИИ

Для авторизации в приложении нажмите кнопку «Войти» (см. Рисунок I-29). Отобразится окно авторизации в приложении Promobot M Control. В окне авторизации введите логин и пароль и нажмите кнопку «Войти» (см. Рисунок II-4).

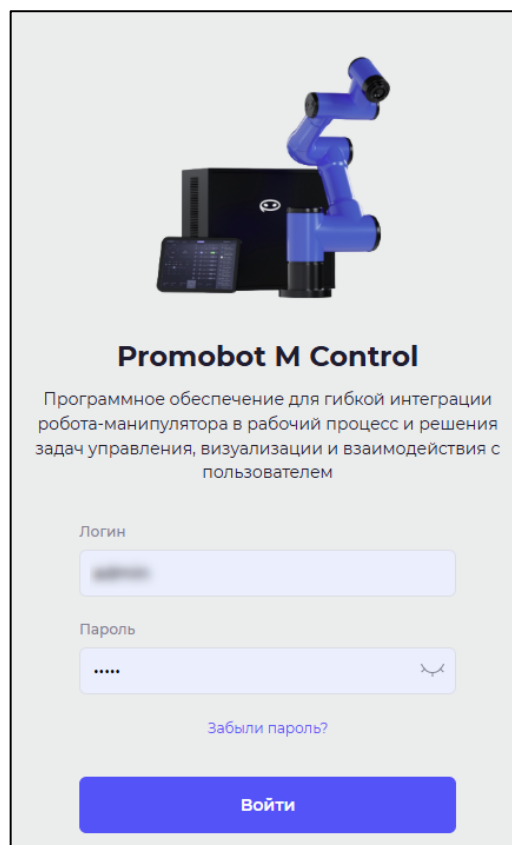


Рисунок II-4 – Окно авторизации

Отобразится интерфейс ручного управления манипулятором (см. Рисунок II-5).

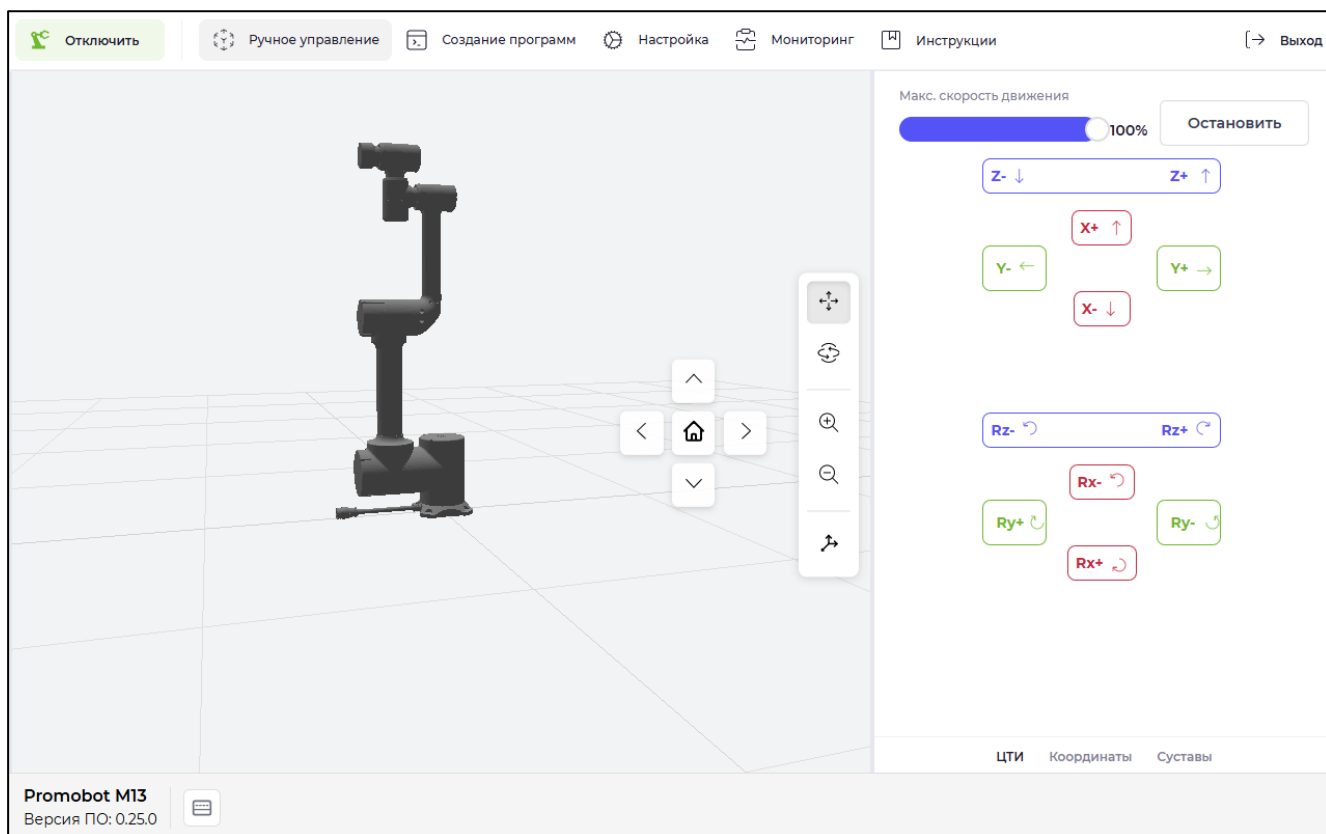


Рисунок II-5 – Интерфейс ручного управления манипулятором

Для подключения манипулятора нажмите кнопку «Подключить» в верхнем левом углу (см. Рисунок II-6).

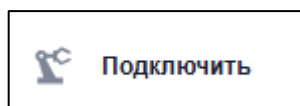


Рисунок II-6 – Кнопка «Подключить»

На манипуляторе запустится калибровка. В приложении отобразится уведомление «Поиск манипулятора». При успешном подключении кнопка изменит цвет на зеленый и отобразится как «Отключить» (см. Рисунок II-7).

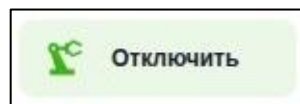


Рисунок II-7 – Кнопка «Отключить»

Для выключения манипулятора и системы нажмите кнопку «Выход» и выберите «Выключить манипулятор» (см. Рисунок II-8). Либо воспользуйтесь кнопкой включения/выключения на пульте управления.

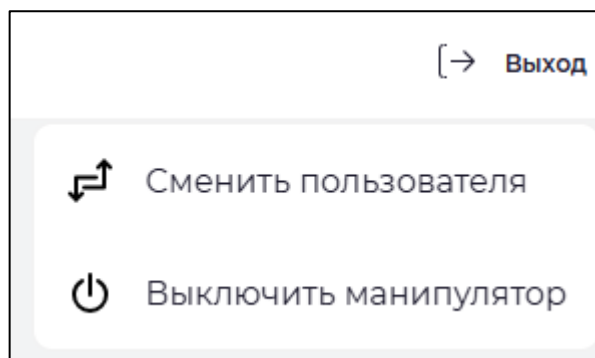


Рисунок II-8 – Диалоговое окно кнопки «Выход»

Меню «Выход» содержит две команды:

- «Сменить пользователя» – завершить текущий сеанс и вернуться к авторизации без выключения манипулятора;
- «Выключить манипулятор» – завершить работу Promobot M13.

Примечание – Перед сменой пользователя требуется отключить манипулятор в приложении.

Команду «Сменить пользователя» применяют, в частности, для входа под сервисной учетной записью. Сервисные функции доступны только пользователям с соответствующими правами.

Если при подключении манипулятора на пульте управления кнопка аварийного останова в рабочем положении, то в приложении отобразится уведомление «Нажата кнопка аварийного останова. Отожмите красную кнопку и подключитесь заново» (см. Рисунок II-9).

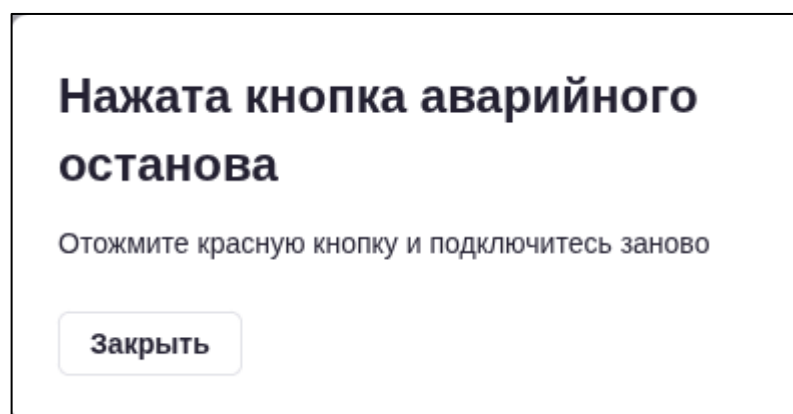


Рисунок II-9 – Уведомление

3 РУЧНОЕ УПРАВЛЕНИЕ

Ручное управление манипулятором (вкладка «Ручное управление») применяется для:

- проверки работы инструмента;
- диагностики приводов и датчиков;
- вывода манипулятора в исходные позиции перед автоматическим циклом.

Важно помнить, что манипулятор может оказаться в сингулярности. В приложении предусмотрено 3 типа ошибки сингулярности, описанные в таблице 11.

Таблица 11 – Типы ошибок сингулярности

Текст ошибки для пользователя	Уровень критичности
Манипулятор находится в сингулярности	ERROR
Манипулятор приближается к сингулярности	WARNING
Манипулятор отдаляется от сингулярности	WARNING

Требования безопасности и порядок действий приведены в разделе I.2.3.3 «Сингулярности».

Интерфейс ручного управления манипулятором (вкладка «Ручное управление») содержит виртуальное пространство с 3D-моделью манипулятора и панели ручного управления манипулятором.

Виртуальное пространство с 3D-моделью манипулятора (см. Рисунок II-10) воспроизводит движения настоящего манипулятора. Обзор 3D-модели манипулятора можно регулировать касанием пальца руки либо интерактивными кнопками.

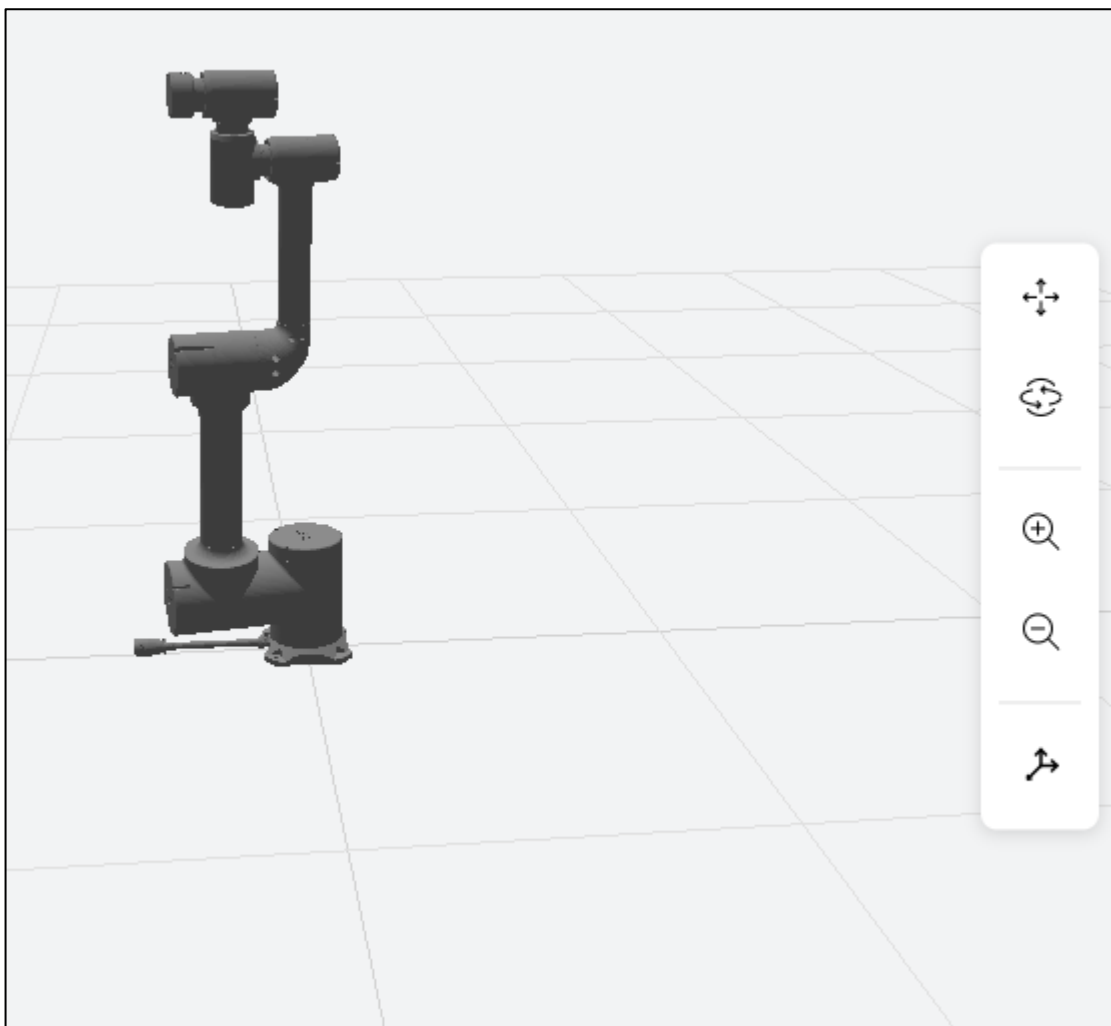


Рисунок II-10 – Виртуальное пространство с 3D-моделью манипулятора

Интерактивные кнопки:

	– перемещение сцены
	– вращение сцены

	– изменение масштаба
	– отображение систем координат основания манипулятора («База»), мировой системы координат («Мир») и инструмента («ЦТИ») с кнопкой «simple» – отображение ЦТИ в виде

Панели ручного управления манипулятором содержит:

- Панель ограничения скорости и останов движения (см. Рисунок II-11) содержит:
 - ползунок «Макс. скорость движения» – задает ограничение скорости для перемещения манипулятора (при первом движении, после изменения инструмента, крепления, рабочей зоны или пределов используйте пониженную скорость);
 - кнопку «Остановить» – прекращает текущее программное движение, но не заменяет аппаратную кнопку аварийного останова.
- При опасной ситуации применяйте аппаратный аварийный останов.



Рисунок II-11 – Панель ограничения скорости и останов движения

- «ЦТИ» – управление центральной точкой инструмента (фланцем инструмента) с помощью кнопок со стрелками направления (см. Рисунок II-15).

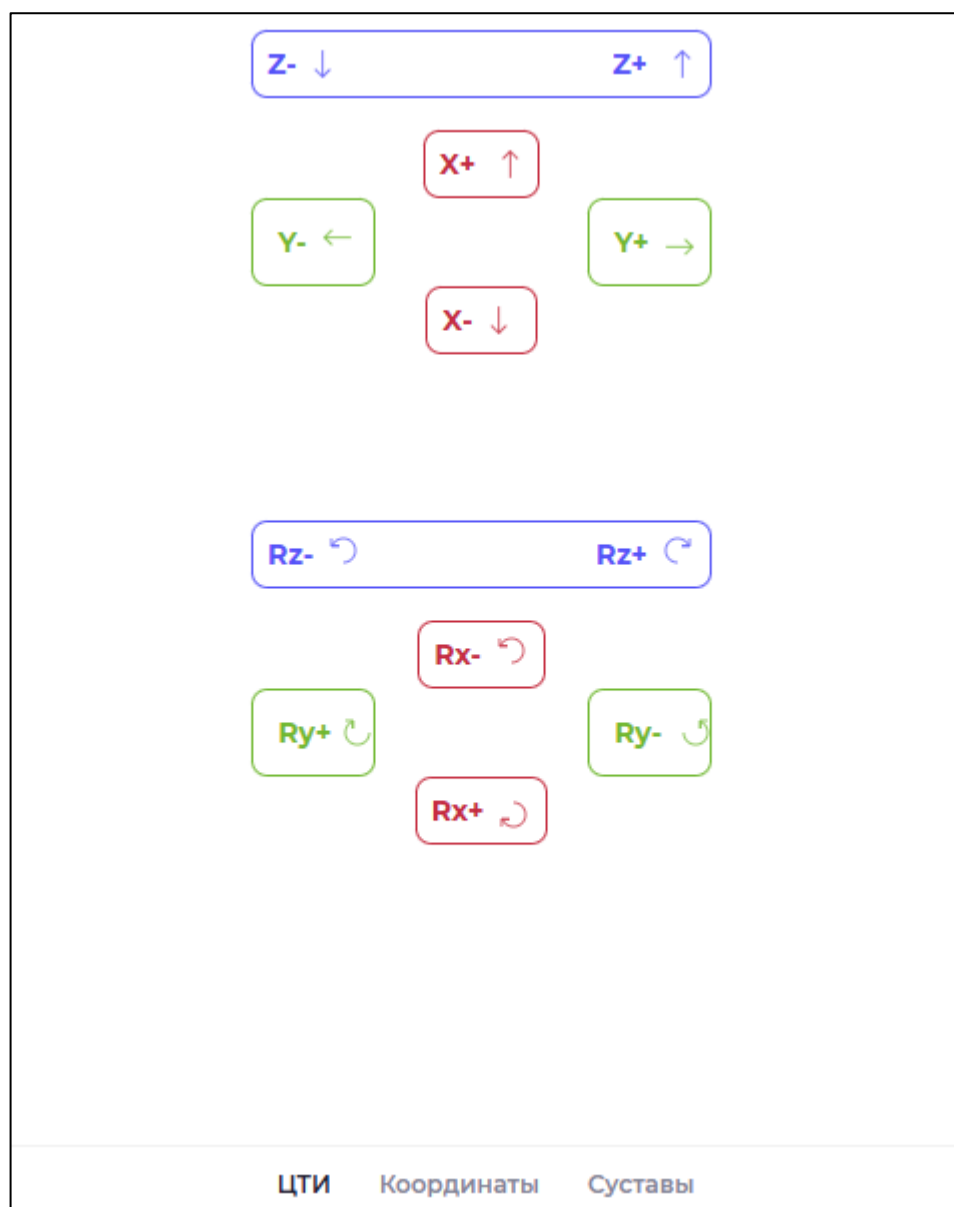


Рисунок II-12 – Управление центральной точкой инструмента

- «Координаты» – управление манипулятором в трехмерной системе координат, значения отображаются в сантиметрах и управление движением сферически вокруг ЦТИ значения отображаются в градусах (см. Рисунок II-13).

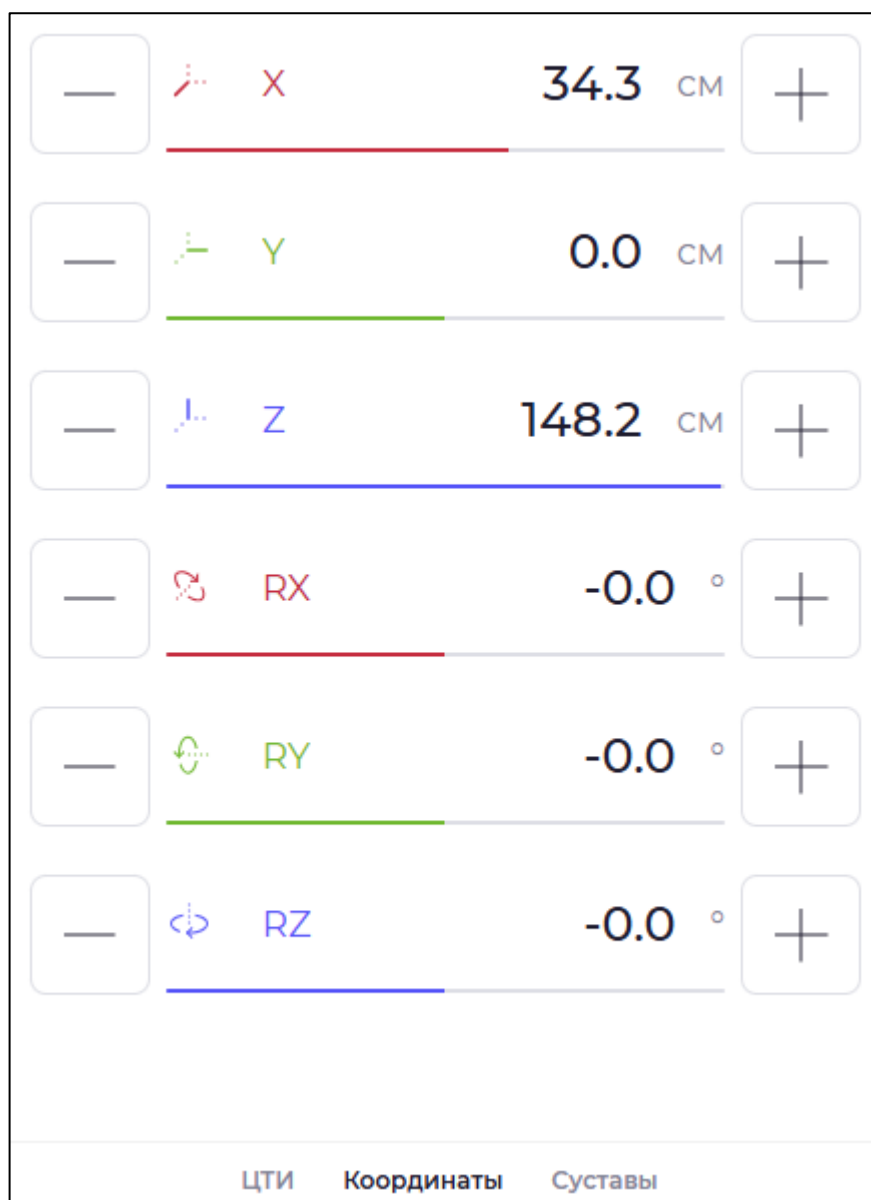


Рисунок II-13 – Управление манипулятором в трехмерной системе координат и движением сферически вокруг ЦТИ

- «Суставы» – управление поворотными узлами манипулятора, значения отображаются в градусах:
 - «Плечевой поддон» – изменяет угол СП1;
 - «Подъем плеча» – изменяет угол СП2;
 - «Локоть» – изменяет угол СП3;
 - «Запястье 1» – изменяет угол СП4;
 - «Запястье 2» – изменяет угол СП5;
 - «Запястье 3» – изменяет угол СП6.

Макс. скорость движения

50%

Остановить

—	Плечевой поддон	170.4 °	+
—	Подъём плеча	186.5 °	+
—	Локоть	182.0 °	+
—	Запястье 1	182.0 °	+
—	Запястье 2	182.0 °	+
—	Запястье 3	182.0 °	+

ЦТИ

Координаты

Суставы

Рисунок II-14 – Управление поворотными узлами манипулятора

Координаты ЦТИ и углы суставов можно вводить в соответствующие поля. Экранная клавиатура открывается автоматически при нажатии на значение.

Перед подтверждением значения проверьте единицы измерения: координаты задаются в сантиметрах, углы – в градусах (см. Рисунок II-15).

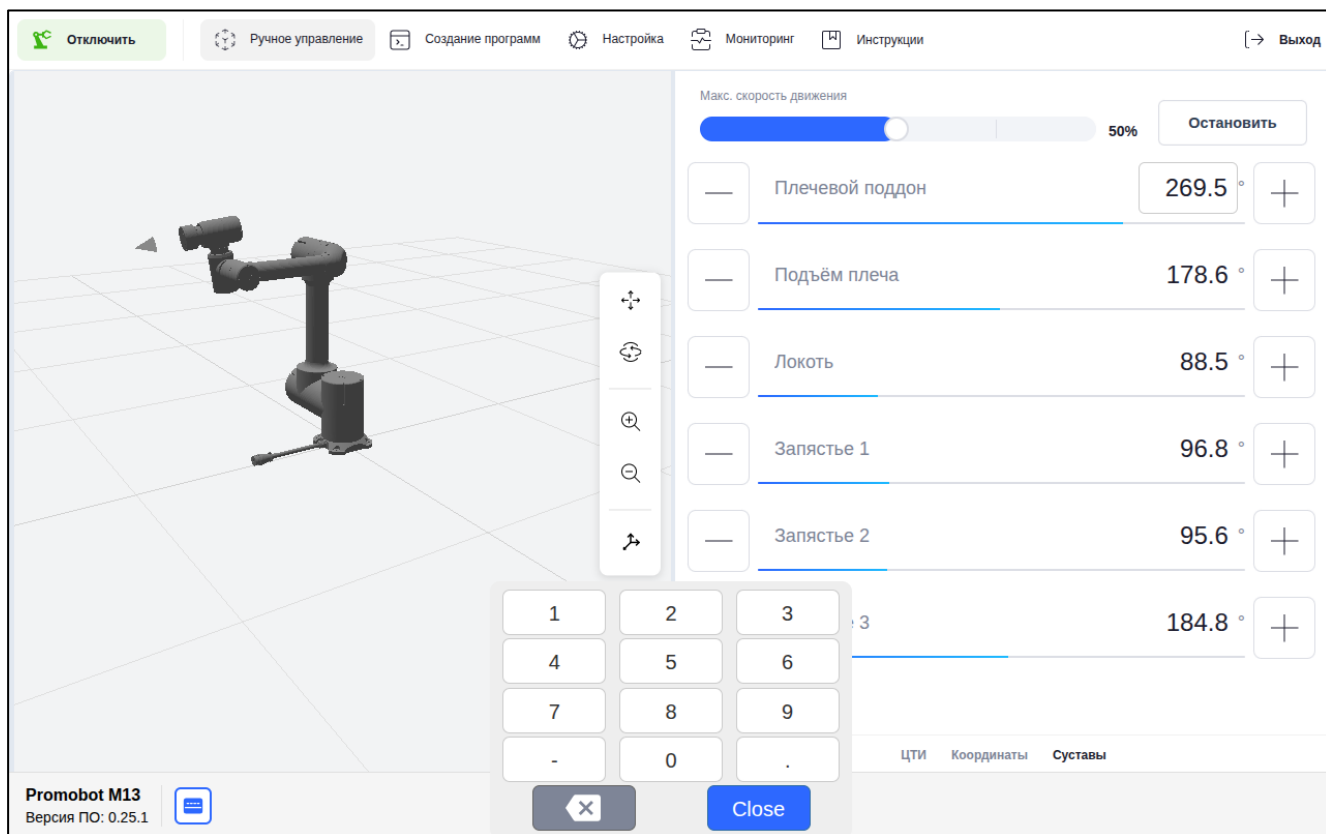


Рисунок II-15 – Поле для ручного ввода значения

Примечание – В приложении значения углов отображаются в градусах, а при программировании через SDK – в радианах.

4 СОЗДАНИЕ ПРОГРАММ

Приложение Promobot M Control позволяет создавать программы движения манипулятора (вкладка «Создание программ») с помощью языков программирования Blockly, Python.

Интерфейс создания программ имеет верхнее меню (см. Рисунок II-16) с кнопками:

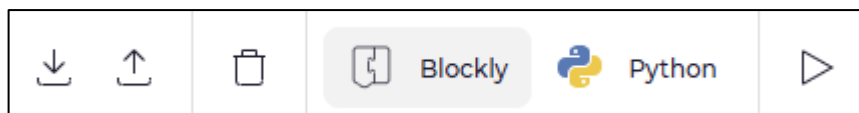


Рисунок II-16 – Верхнее меню создания программ



«Скачать» – кнопка для скачивания созданного скрипта в текстовом формате.



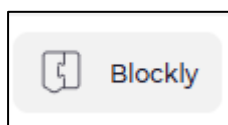
«Загрузить» – кнопка загрузки готового скрипта.



«Удалить все» – кнопка для очистки редактора.



«Распиновка» – кнопка для открытия схемы расположения электрических интерфейсов на фланце и пользовательской панели, активна только при конфигурировании портов.



«Blockly» – переход в графический редактор Blockly.



«Python» – переход в редактор кода Python.



«Запуск на манипуляторе» – кнопка запуска скрипта на манипуляторе.

В интерфейсе предусмотрено поле «Результат выполнения программы» – отображает результат выполнения программы. Для увеличения поля щелкните по нему левой кнопкой мыши и поле увеличится за счет виртуального пространства (см. Рисунок II-17).



Рисунок II-17 – Поле «Результат выполнения программы»

Интерфейс создания программ позволяет создавать программы на двух языках программирования: Blockly, Python.

4.1 Blockly

Blockly в приложении Promobot M Control представляет собой набор графических блоков (библиотека) определенной формы и графический редактор (см. Рисунок II-18). Соединяя одни блоки с другими, можно создать программу движений для манипулятора.

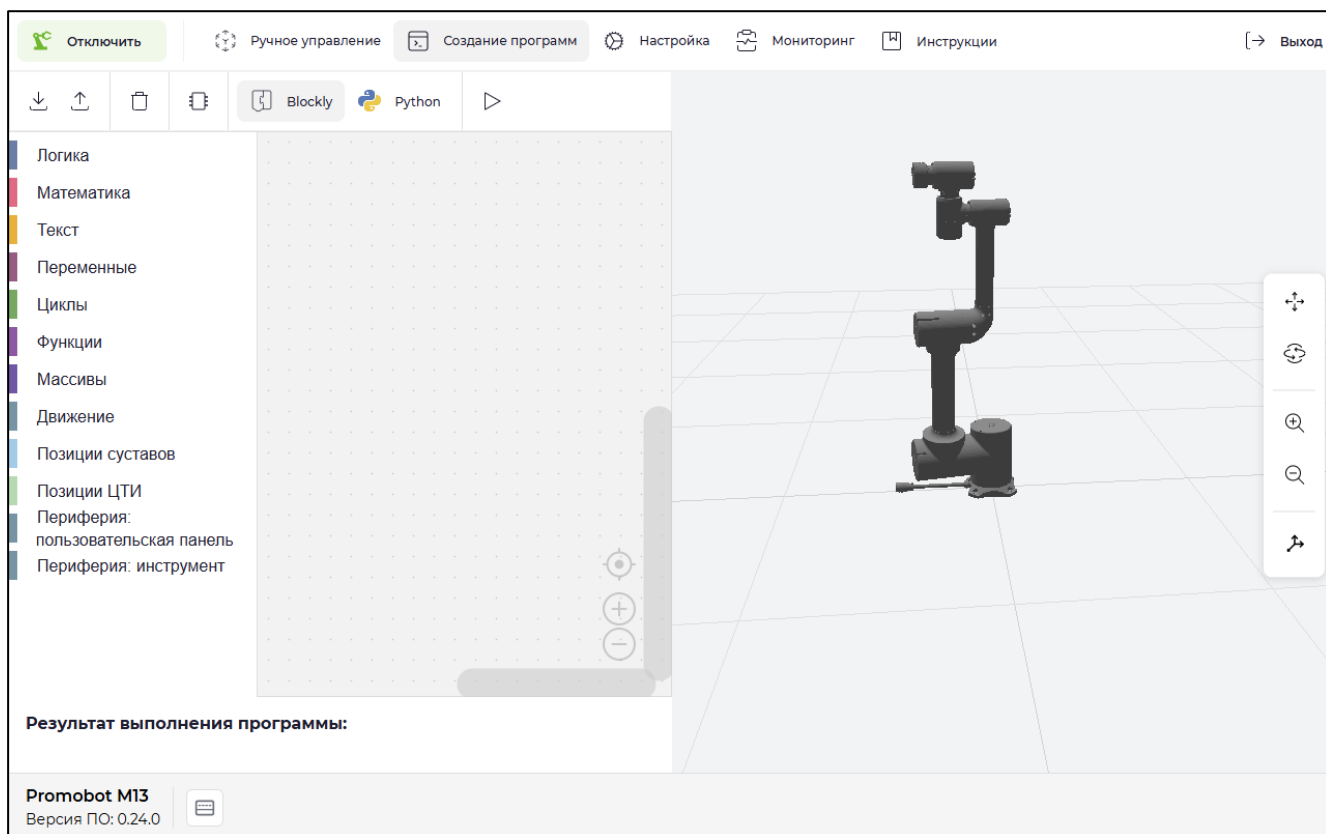


Рисунок II-18 – Blockly в приложении Promobot M Control

4.1.1 Описание графических блоков (библиотека Blockly)

Библиотека Blockly в приложении Promobot M Control имеет следующие разделы блоков:

- а) блоки «Логика» – условия, сравнения (см. Рисунок II-19);

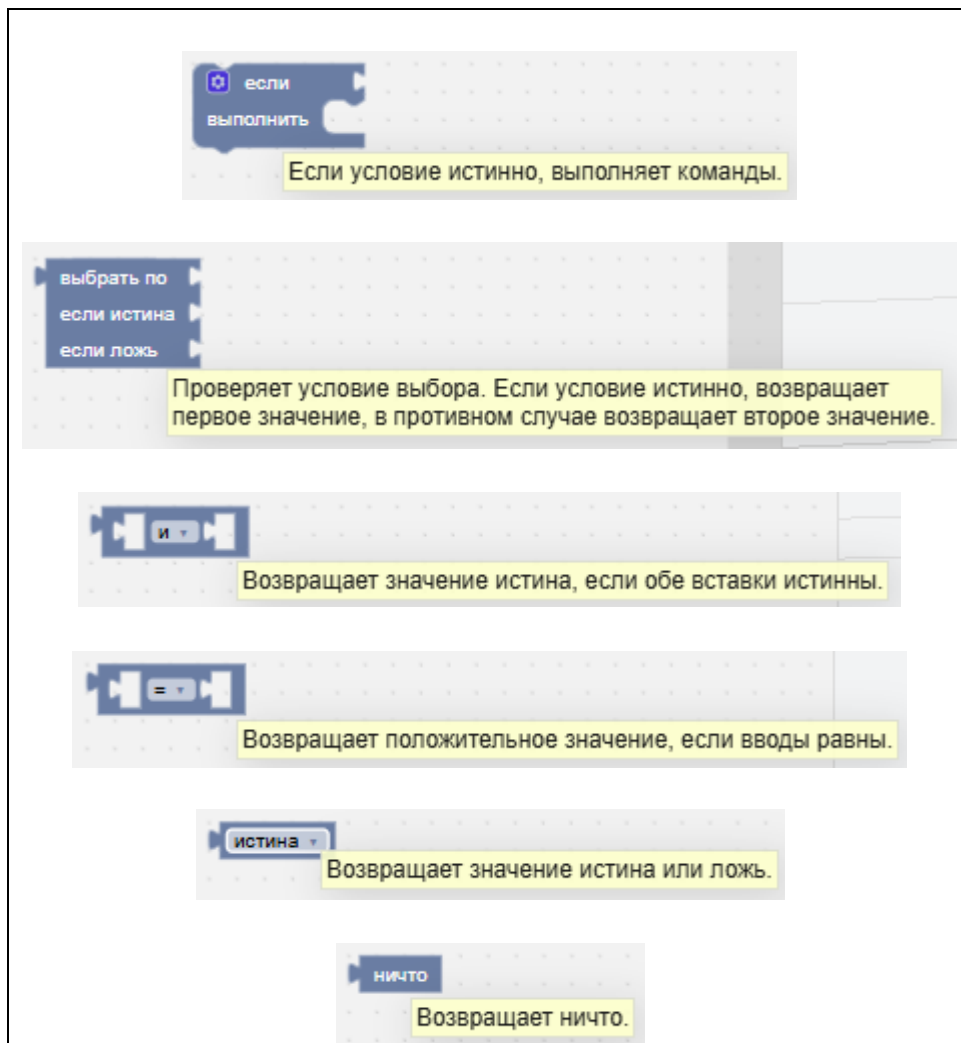


Рисунок II-19 – Блоки «Логика»

б) блоки «Математика» – вычисления, формулы (см. Рисунок II-20);

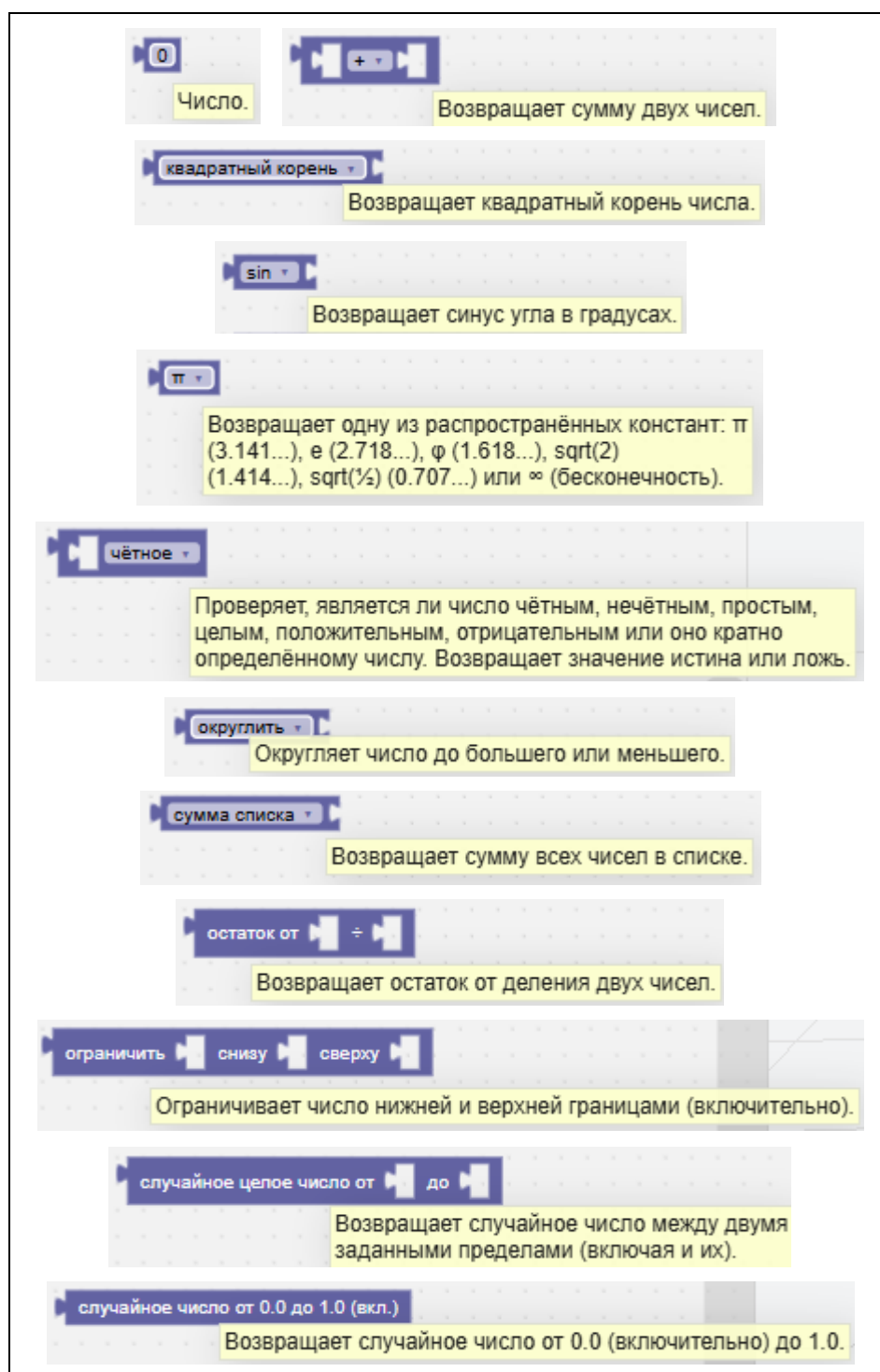


Рисунок II-20 – Блоки «Математика»

в) блоки «Текст» – работа с текстом (см. Рисунок II-21);

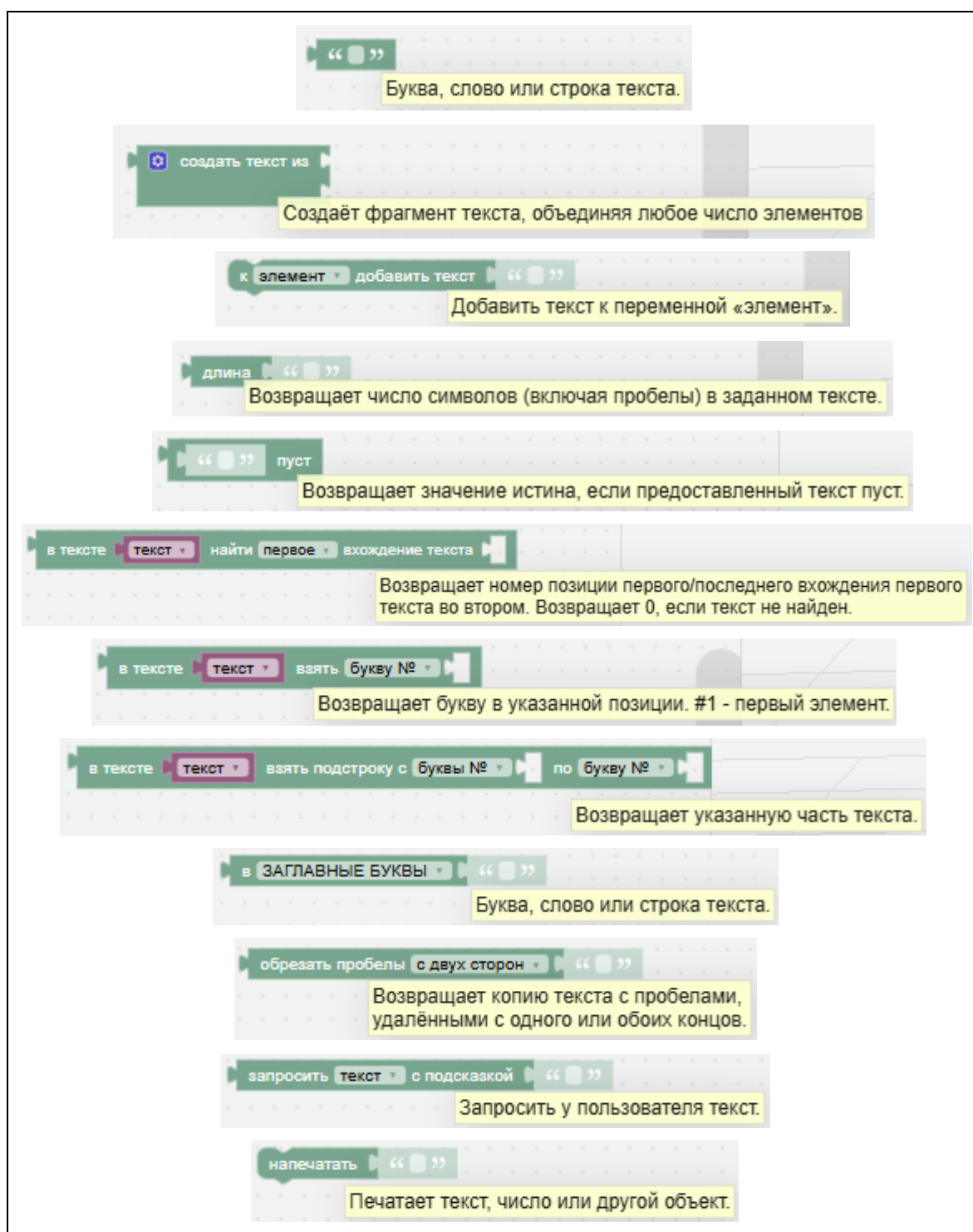


Рисунок II-21 – Блоки «Текст»

г) блоки «Переменные» – создание и редактирование переменных (см. Рисунок II-22);

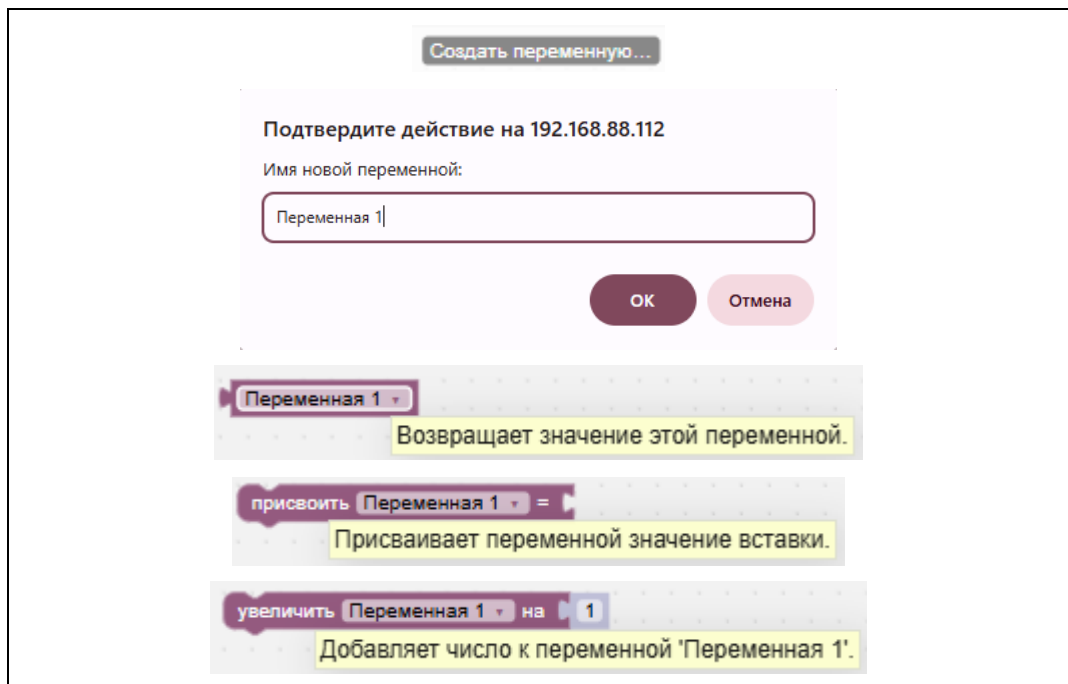


Рисунок II-22 – Блоки «Переменные»

- д) блоки «Циклы» – многократное повторение определенных действий (команд) (см. Рисунок II-23);

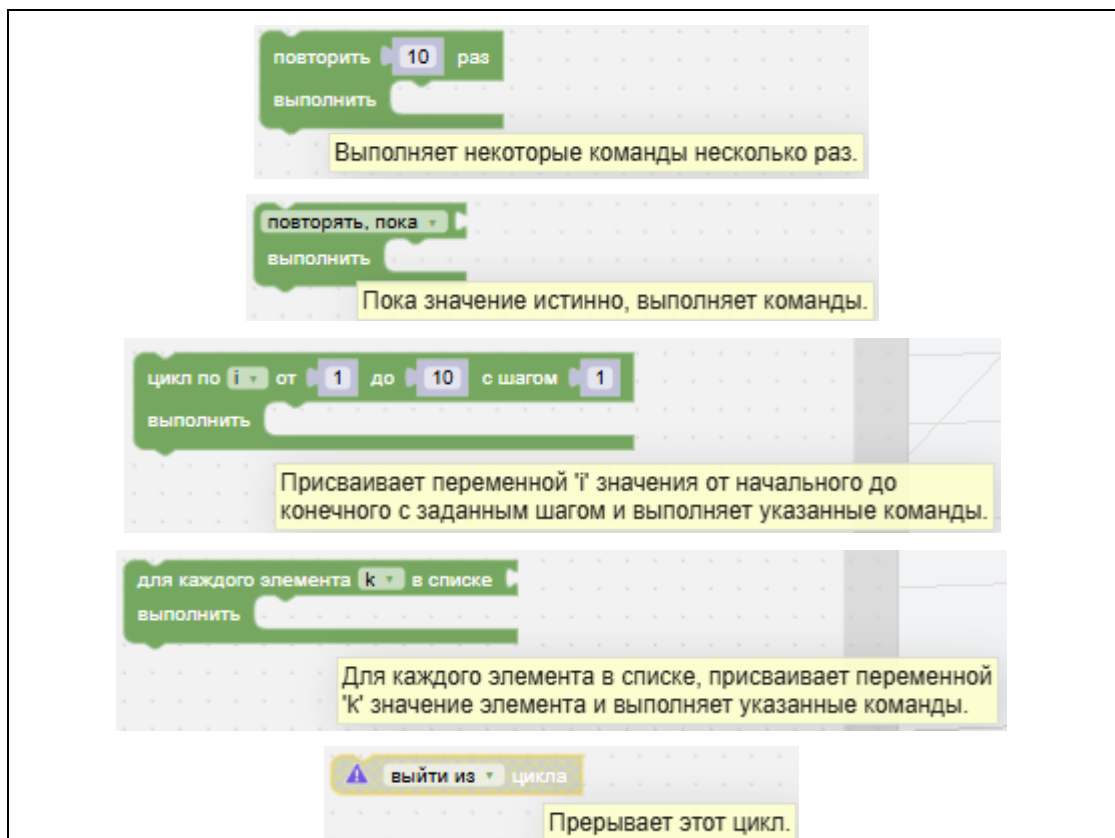


Рисунок II-23 – Блоки «Циклы»

- е) блоки «Функции» – создание собственных процедур и именованных подпрограмм (см. Рисунок II-24);

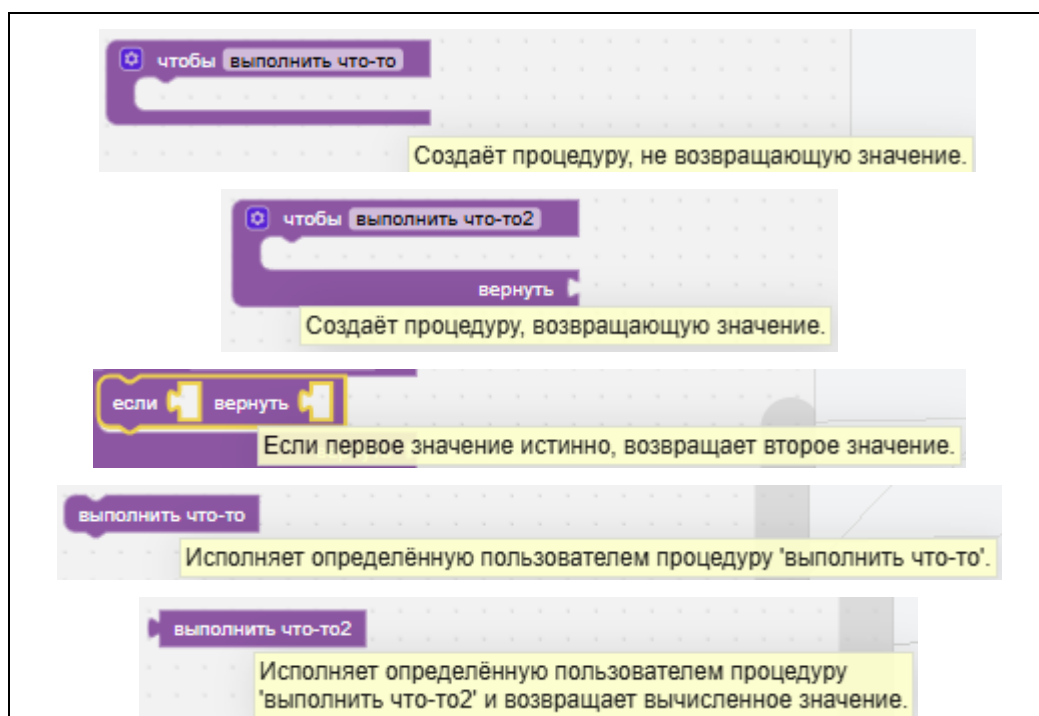


Рисунок II-24 – Блоки «Функции»

- ж) блоки «Массивы» – работа с элементами списков (см. Рисунок II-25);

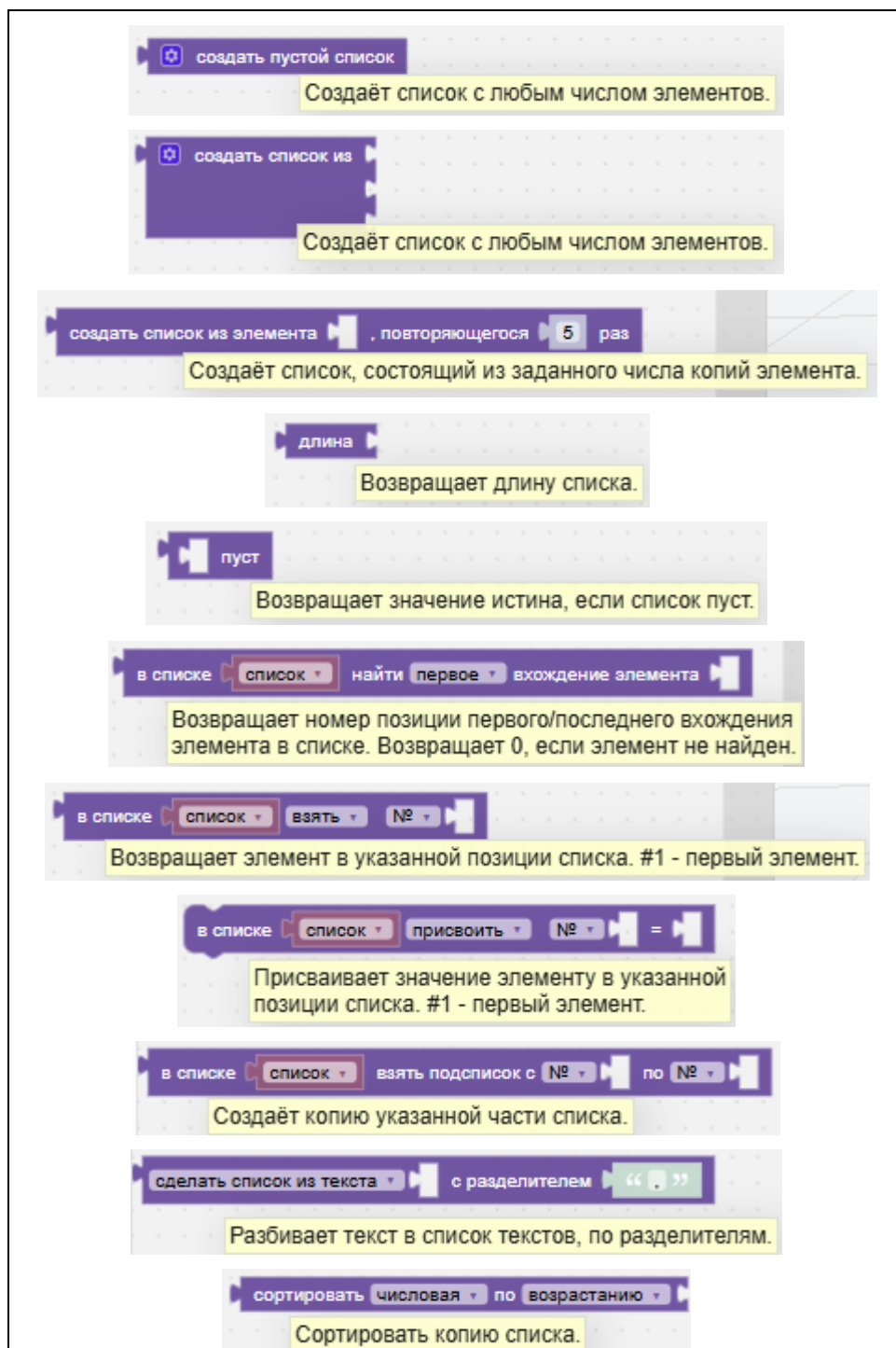


Рисунок II-25 – Блоки «Массивы»

з) блоки «Движение» – команды перемещения (см. Рисунок II-26);

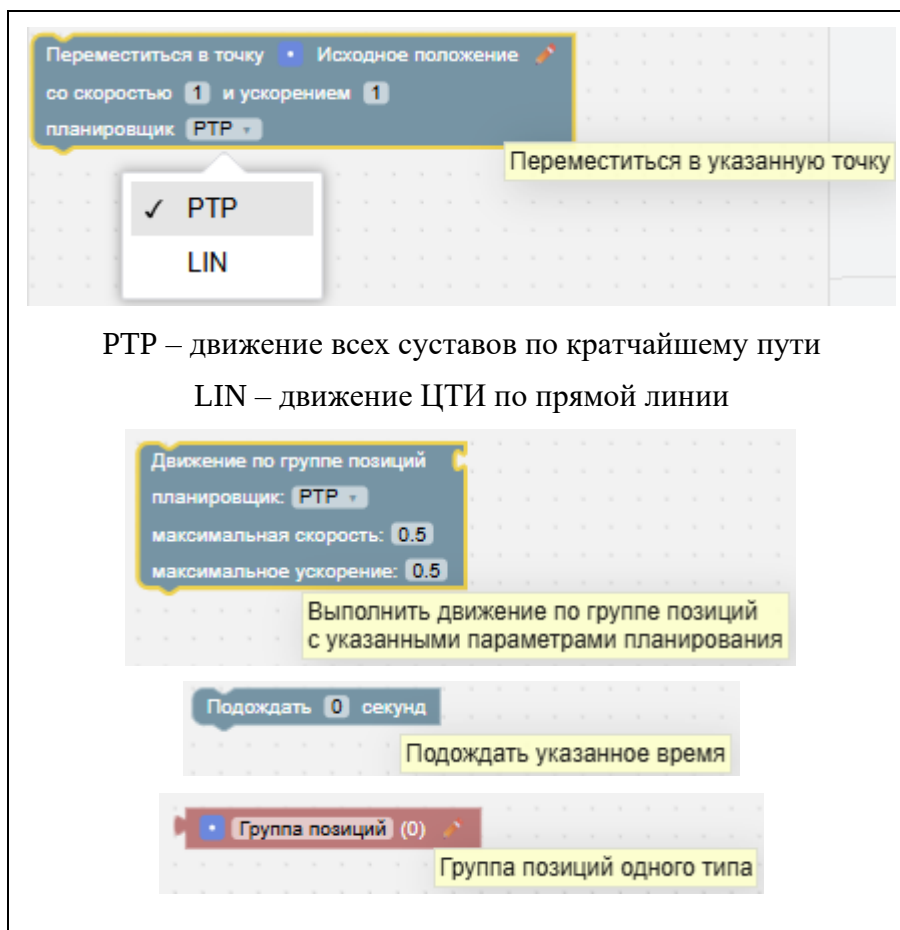


Рисунок II-26 – Блоки «Движение»

Параметры скорости и ускорения являются коэффициентами от скорости, указанной в разделе «Положение и пределы».

Блок «группа позиций» (см. Рисунок II-27) формирует упорядоченный набор сохраненных позиций.

Примечание – Блок «Движение по группе позиций» выполняет этот набор с выбранным планировщиком и установленными ограничениями скорости и ускорения.

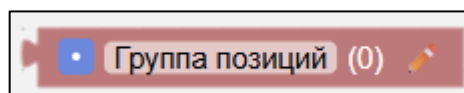


Рисунок II-27 – Блок «группа позиций»

Для редактирования или добавления позиций нажмите , отобразится форма «Группа позиций» (см. Рисунок II-28). Выберите уже имеющиеся позиции или создайте новую.

Группа позиций: Группа позиций

Позиции суставов

Создать позицию

Исходное положение

Выбрать

Редактировать

Рабочая позиция

Выбрать

Редактировать

Позиции ЦТИ

Создать позицию

Каталог позиций ↑

Группа пуста. Добавьте позиции, нажав кнопку "+".

Рисунок II-28 – Форма «Группа позиций»

После редактирования группы повторно проверьте все переходы на пониженной скорости.

- и) блоки «Позиции суставов» – сохранение и использование точек положения СП манипулятора относительно вертикальной плоскости (см. раздел I.1.2.1 «Манипулятор»); для добавления и сохранения позиций нажмите «Добавить позицию суставов», отобразится форма для создания (см. Рисунок II-29);

Добавить позицию суставов

Исходное положение

Рабочая позиция

Группа позиций (0)

Позиция по суставам

×

Наименование

Введите имя позиции

Макс. скорость движения

50%

—

Плечевой поддон

170.4 °

+

—

Подъём плеча

186.5 °

+

—

Локоть

182.0 °

+

Суставы

Сохранить

Остановить движение

Удалить

Рисунок II-29 – Форма для создания позиции по суставам

По умолчанию добавлены две позиции «Исходное положение» и «Рабочая позиция», при нажатии на них отобразится форма, для изменения нажмите кнопку «Разблокировать редактирование».

• Исходное положен

• Рабочая позиц

Позиция по суставам

×

Разблокировать редактирование

(!) Редактирование сбросит все координаты до текущего положения манипулятора

Наименование

Исходное положение

Сохранить

Остановить движение

Удалить

Рисунок II-30 – Форма «Позиция по суставам»

- к) блоки «Позиции ЦТИ» – сохранение и использование точек положения центральной точки инструмента (ЦТИ). Для добавления и сохранения позиций нажмите «Добавить позицию ЦТИ», отобразится форма для создания позиции ЦТИ (см. Рисунок II-31).

Добавить позицию ЦТИ

Группа позиций (0)
Группа позиций одного типа

Позиция по ЦТИ

✕

Наименование

Введите имя позиции

Макс. скорость движения

50%

Z- ↓

Z+ ↑

X+ ↑

Y- ←

Y+ →

ЦТИ
Координаты

Сохранить

Остановить движение

Удалить

Рисунок II-31 – Форма для создания позиции ЦТИ

4.1.1.1 Блоки «Периферия»

Программирование групп электрических интерфейсов на пользовательской панели и на фланце (см. Рисунок II-32).

Для правильного определения расположения электрических интерфейсов при использовании блоков периферии, можно использовать кнопку  для открытия схемы их расположения (см. Рисунок II-32).

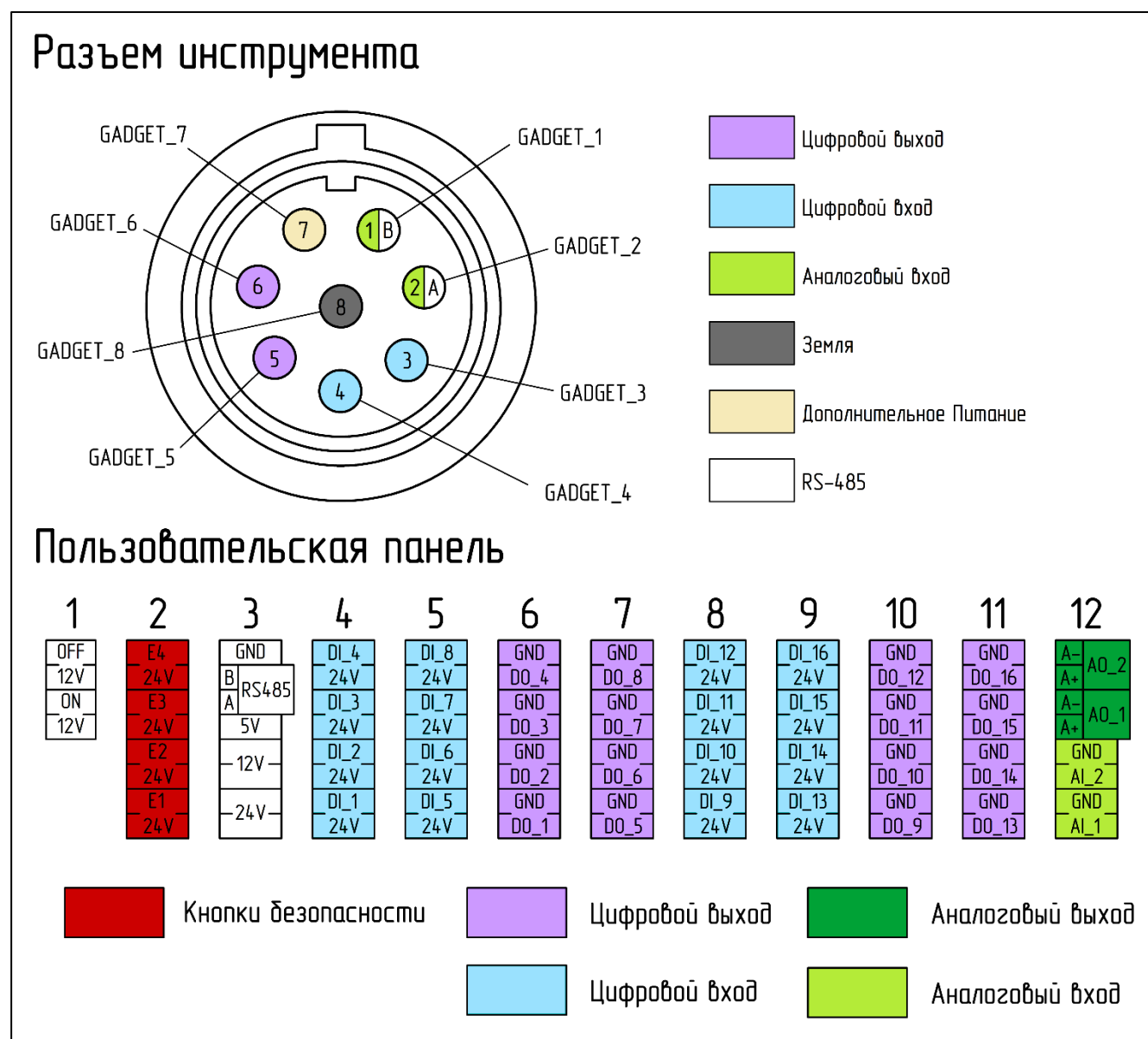


Рисунок II-32 – Схема расположения электрических интерфейсов

Логика работы с аналоговыми портами: сначала порт должен быть сконфигурирован отдельным блоком далее в программе можно использовать порт с выбранными настройками.

Работа с цифровыми портами на пользовательской панели не требует их конфигурирования, для цифровых выходов разъема инструмента имеется возможность выбора режима работы.

4.1.1.1. Блоки «Периферия: пользовательская панель»

Программирование групп электрических интерфейсов на пользовательской панели (см. Рисунок II-33).



Рисунок II-33 – Блоки «Периферия: пользовательская панель»

Раздел «Периферия: пользовательская панель» состоит из следующих блоков:

а) аналоговый вход:

1) блок конфигурирования (см. Рисунок II-34):

- параметр: «Аналоговый вход» – поле: «AI_1» ... «AI_2»;
- параметр «режим» – поле: «Напряжение» / «Ток».

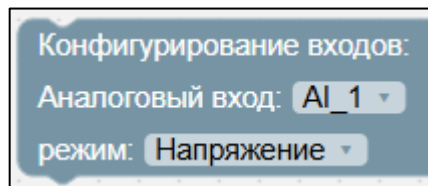


Рисунок II-34 – Блок конфигурирования

2) блоки измерения (см. Рисунок II-35):

- параметр: «Измерить напряжение»;
- параметр: «Измерить ток».

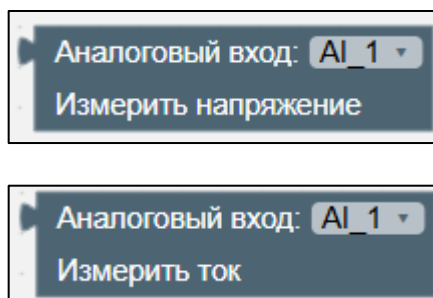


Рисунок II-35 – Блоки измерения

б) аналоговый выход:

1) блок конфигурирования (см. Рисунок II-36):

- параметр: «Аналоговый выход» – поле: «АО_1» ... «АО_2»;
- параметр «режим» – поле: «Напряжение»/ «Ток».

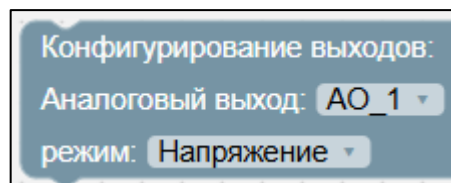


Рисунок II-36 – Блок конфигурирования

2) блоки задания выходного сигнала (см. Рисунок II-37):

- параметр: «Установить напряжение»;
- параметр: «Установить ток».

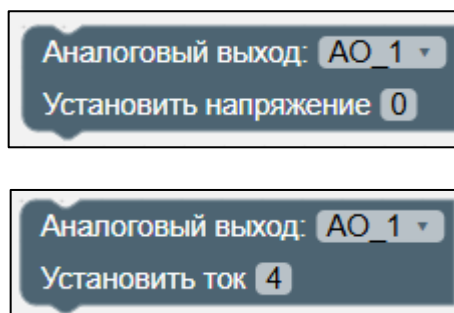


Рисунок II-37 – Блоки задания выходного сигнала

Примечание – Ограничения по входным/выходным характеристикам портов в соответствии с разделом I.2.2.2.3 «Подключение внешних устройств».

в) цифровой вход (см. Рисунок II-38 – Цифровой вход):

1) параметр: «Цифровой вход» – поле: «DI_1» ... «DI_16».

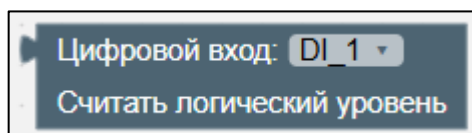


Рисунок II-38 – Цифровой вход

г) цифровой выход (см. Рисунок II-39):

1) параметр: «Цифровой выход» – поле: «DO_1» ... «DO_16»;

2) параметр: «Установить значение» – поле: «0» / «1».

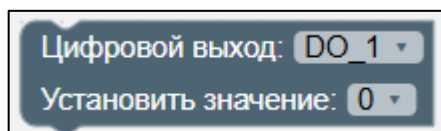


Рисунок II-39 – Цифровой выход

4.1.1.2. Блоки «Периферия: инструмент»

Программирование групп электрических интерфейсов на фланце (см. Рисунок II-40).

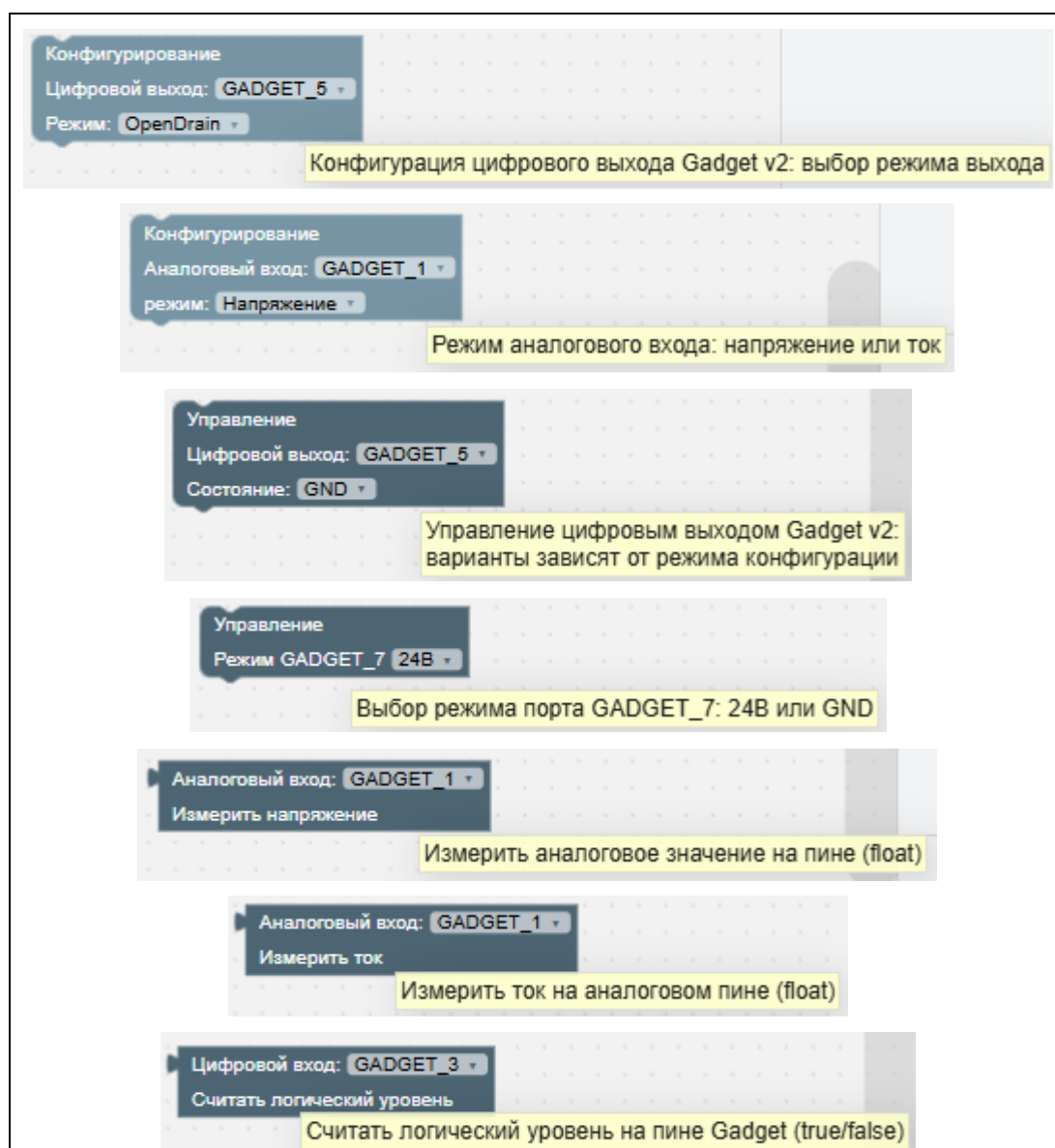


Рисунок II-40 – Блоки «Периферия: инструмент»

Раздел «Периферия: инструмент» состоит из следующих блоков:

а) аналоговый вход:

1) блок конфигурирования (см. Рисунок II-41):

- параметр: «Аналоговый вход» – поле: «GADGET_1»/ «GADGET_2»;
- параметр: «режим»: поле: «Напряжение» / «Ток».

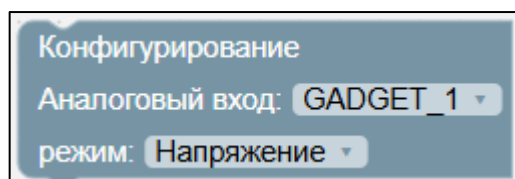


Рисунок II-41 – Блок конфигурирования

2) блоки измерений (см. Рисунок II-42):

- параметр: «Измерить напряжение»;
- параметр: «Измерить ток».

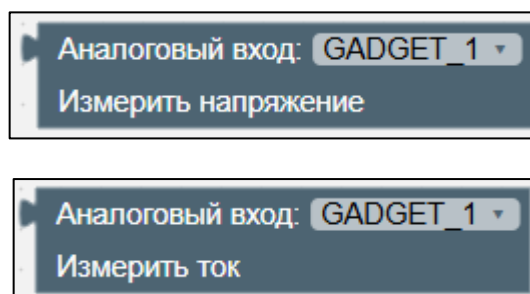


Рисунок II-42 – Блоки измерений

б) питание инструмента (см. Рисунок II-43):

1) блок задания выходного сигнала (см. Рисунок II-43):

- параметр: «Режим GADGET_7»:
- поле: «24В»;
- поле: «GND».

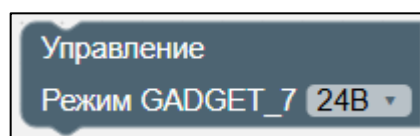


Рисунок II-43 – Блок задания выходного сигнала

Примечание – Ограничения по входным/выходным характеристикам портов в соответствии с разделом I.2.2.2.3 «Подключение внешних устройств».

в) цифровой вход (см. Рисунок II-44):

- параметр: «Цифровой вход» – поле: «GADGET_3» / «GADGET_4».

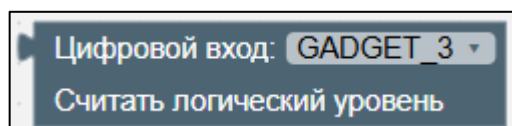


Рисунок II-44 – Цифровой вход

г) цифровой выход:

1) блок конфигурирования (см. Рисунок II-45):

- параметр: «Цифровой вход» – поле: «GADGET_5»/ «GADGET_6»;
- параметр: «Режим» – поле: «OpenDrain» / «OpenSource» / «Push/Pull»;

- параметр: «Цифровой выход» – поле: «GADGET_5»/ «GADGET_6»;
- параметр: «Состояние» (зависит от версии платы и параметра «Режим» в блоке «Конфигурирование. Цифровой выход»):
 - версия платы gadget v3:
 - GADGET_5:
 - режим «OpenDrain»: состояния «HighZ», «GND»;
 - режим «OpenSource»: состояния «V24», «HighZ»;
 - режим «Push/Pull»: состояния «V24», «GND»;
 - GADGET_6:
 - режим «OpenDrain»: состояния «HighZ», «GND»;
 - режим «OpenSource»: состояния «V24», «V12», «HighZ»;
 - режим «Push/Pull»: состояния «V24», «V12», «GND»;
 - версия платы gadget v2:
 - GADGET_5:
 - режим «OpenDrain»: состояния «HighZ», «GND»;
 - режим «OpenSource»: состояния «V24», «HighZ»;
 - режим «Push/Pull»: состояния «V24», «GND».
 - GADGET_6:
 - режим «OpenDrain»: состояния «HighZ», «GND»;
 - режим «OpenSource»: состояния «V12», «HighZ»;
 - режим «Push/Pull»: состояния «V24», «V12», «GND».

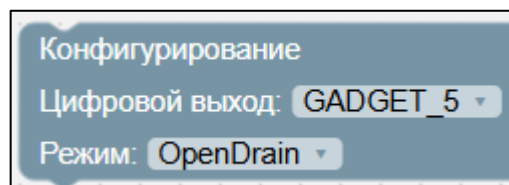


Рисунок II-45 – Блок конфигурирования

4.2 Python

Python в приложении Promobot M Control представляет собой редактор кода Python для написания, редактирования и сохранения кода, оптимизированный под синтаксис языка Python (см. Рисунок II-46).

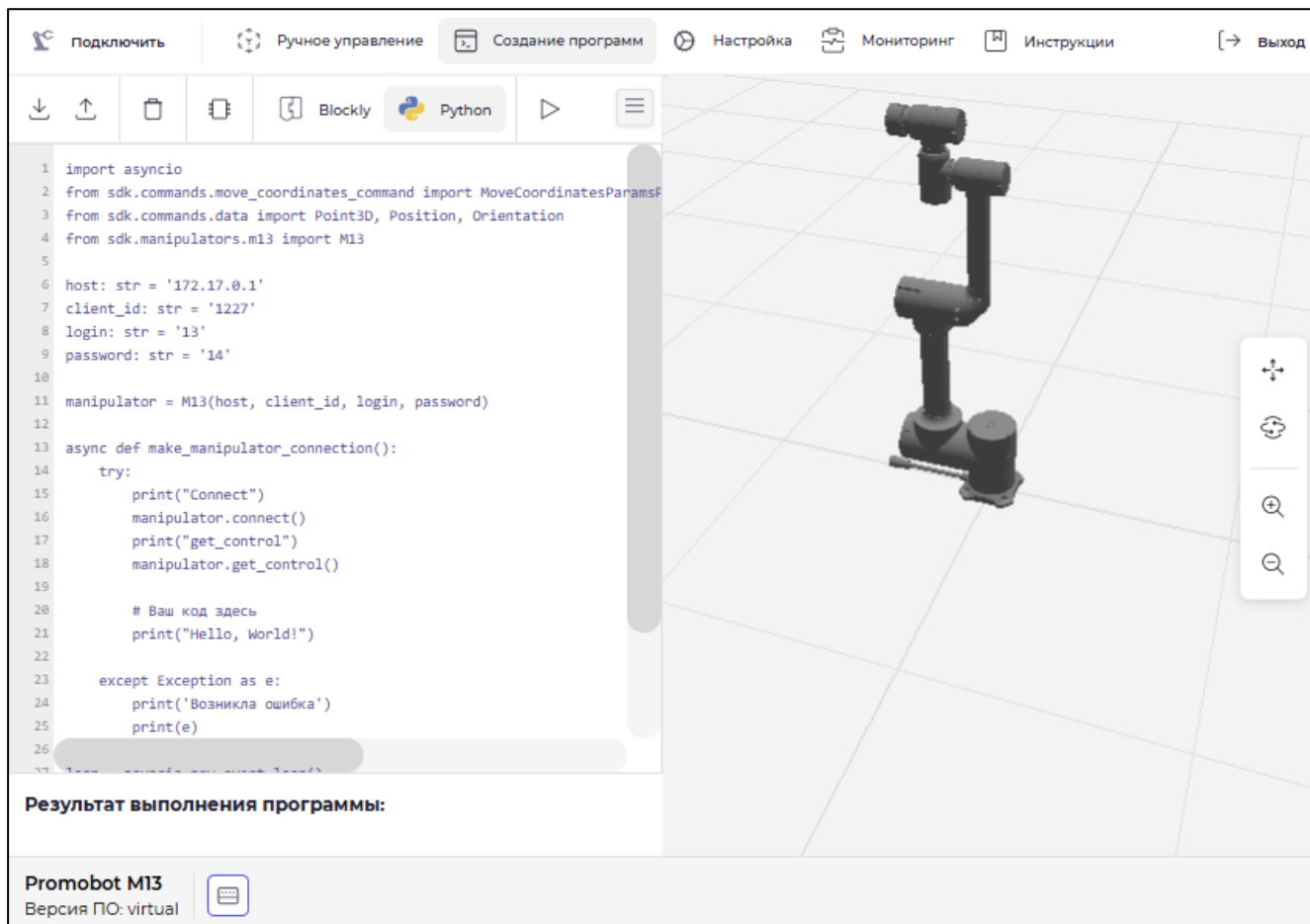


Рисунок II-46 – Python в приложении Promobot M Control

Если ранее в Blockly была создана программа, то ее код может автоматически генерироваться в Python. Далее можно редактировать существующие строки.

При написании собственного кода используйте библиотеку SDK, описанную в разделе III «PROMOBOT M CONTROL SDK».

При загрузке готового файла Python учтите, что программный код в файле должен быть написан по правилам PEP8, сам файл должен иметь расширение .py или .json.

В соответствии со структурой JSON формат сохранения программы такой – в JSON-объекте присутствует единственное поле "program": "/* текст программы */",

содержащее всю программу в типе данных "строка". Каждая новая строка кода отделяется от предыдущей при помощи символа "\n" (без использования переноса строки).

Пример программного кода в формате .json:

```
{
    "program": "import asyncio\nfrom sdk.commands.move_coordinates_command import\nMoveCoordinatesParamsPosition, MoveCoordinatesParamsOrientation\nfrom\nsdk.manipulators.m13 import M13\n\nhost: str = '172.17.0.1'\nclient_id: str = '1227'\nlogin: str =\n'13'\npassword: str = '14'\n\nmanipulator = M13(host, client_id, login, password)\n\nasync def\nmake_manipulator_connection():\n    try:\n        print(\"Connect\")\n        manipulator.connect()\n\n        print(\"get_control\")\n        manipulator.get_control()\n\n        # Ваш код\n        #\n\n        print(\"Hello, World!\")\n\n    except Exception as e:\n        print('Возникла\nошибка')\n        print(e)\n\nloop\n= asyncio.new_event_loop()\nasyncio.set_event_loop(loop)\nasyncio.get_event_loop().run_until_\ncomplete(make_manipulator_connection())\n"
```

5 НАСТРОЙКА

Интерфейс для настройки (вкладка «Настройка») содержит разделы (см. Рисунок П-47):

- «Общие и системные настройки» – сброс кэша и перезапуск интерфейса;
- «Инструмент» – добавление, редактирование, активация центральной точки инструмента;
- «Положение и пределы» – ограничение скоростей и сочленений манипулятора;
- «Рабочая зона» – защита от столкновения с полом.
- «Сервисные функции» – перевод манипулятора в транспортное положение и движение без инициализации (раздел доступен только для пользователя с доступом к сервисным функциям).

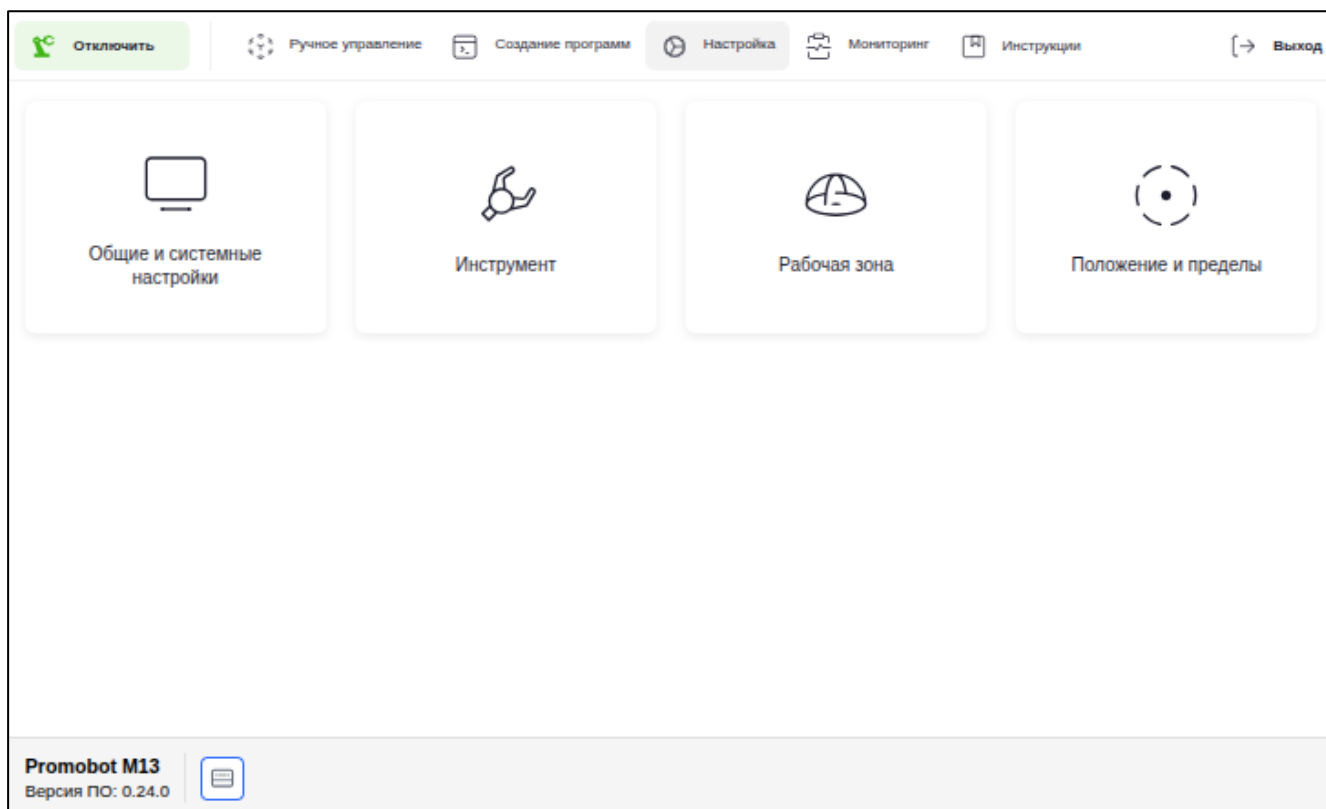


Рисунок II-47 – Интерфейс для настройки

5.1 Общие и системные настройки

В разделе «Общие и системные настройки» (см. Рисунок II-48) находится кнопка «Сбросить кэш и перезапустить интерфейс», которая может быть использована для освобождения памяти и исправления ошибок, возникающих при работе с приложением.

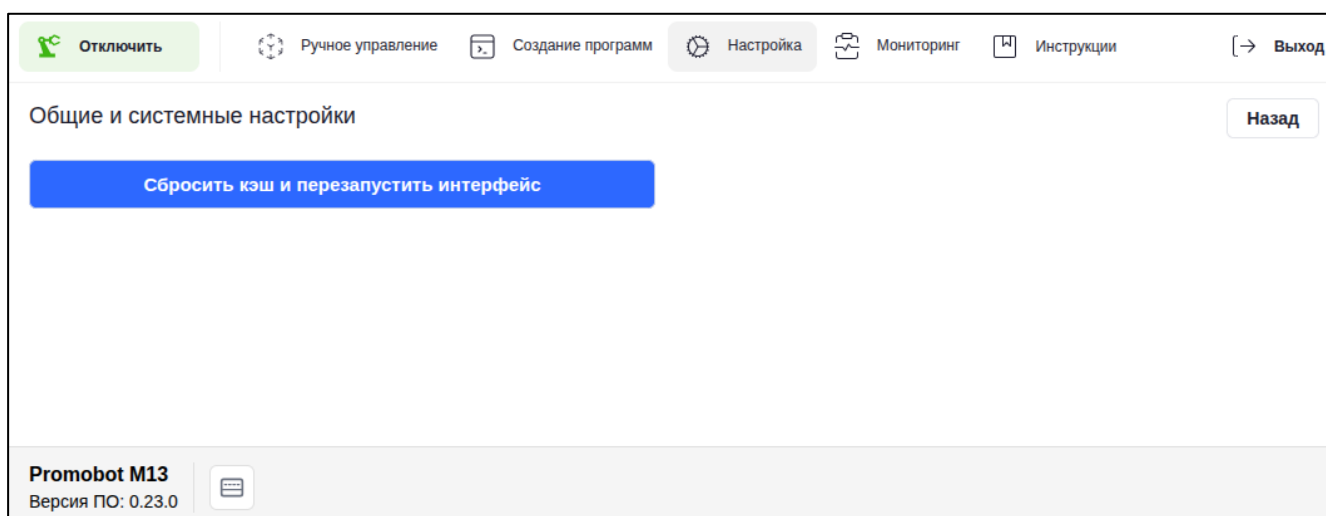


Рисунок II-48 – Раздел «Общие и системные настройки»

Для возврата к другим настройкам нажмите кнопку «Назад».

5.2 Инструмент

Раздел «Инструмент» (см. Рисунок II-49) позволяет добавлять центральную точку инструмента (ЦТИ), установленного на фланец.

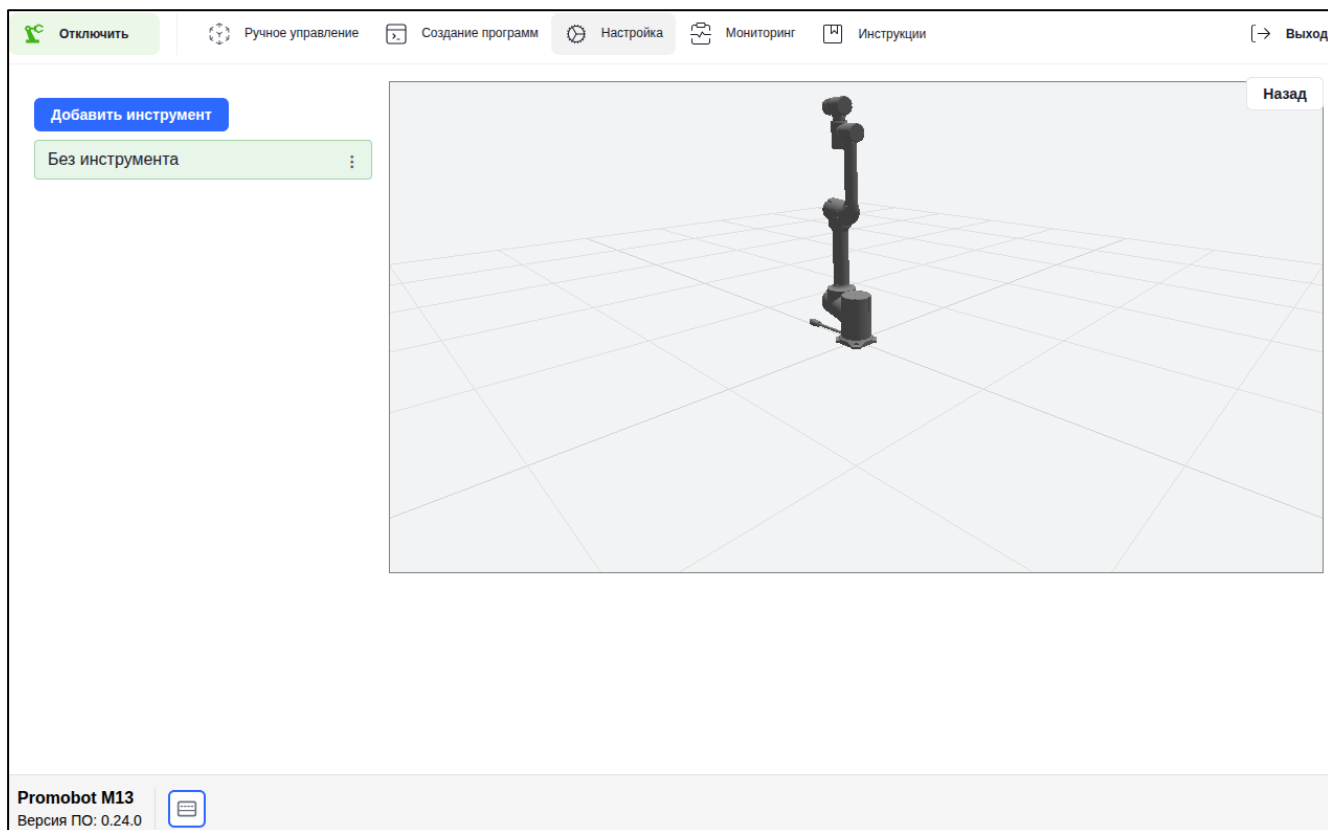


Рисунок II-49 – Раздел «Инструмент»

Для добавления инструмента выполните следующие действия:

- а) нажмите кнопку «Добавить инструмент», отобразится форма для добавления новой точки (см. Рисунок II-50);

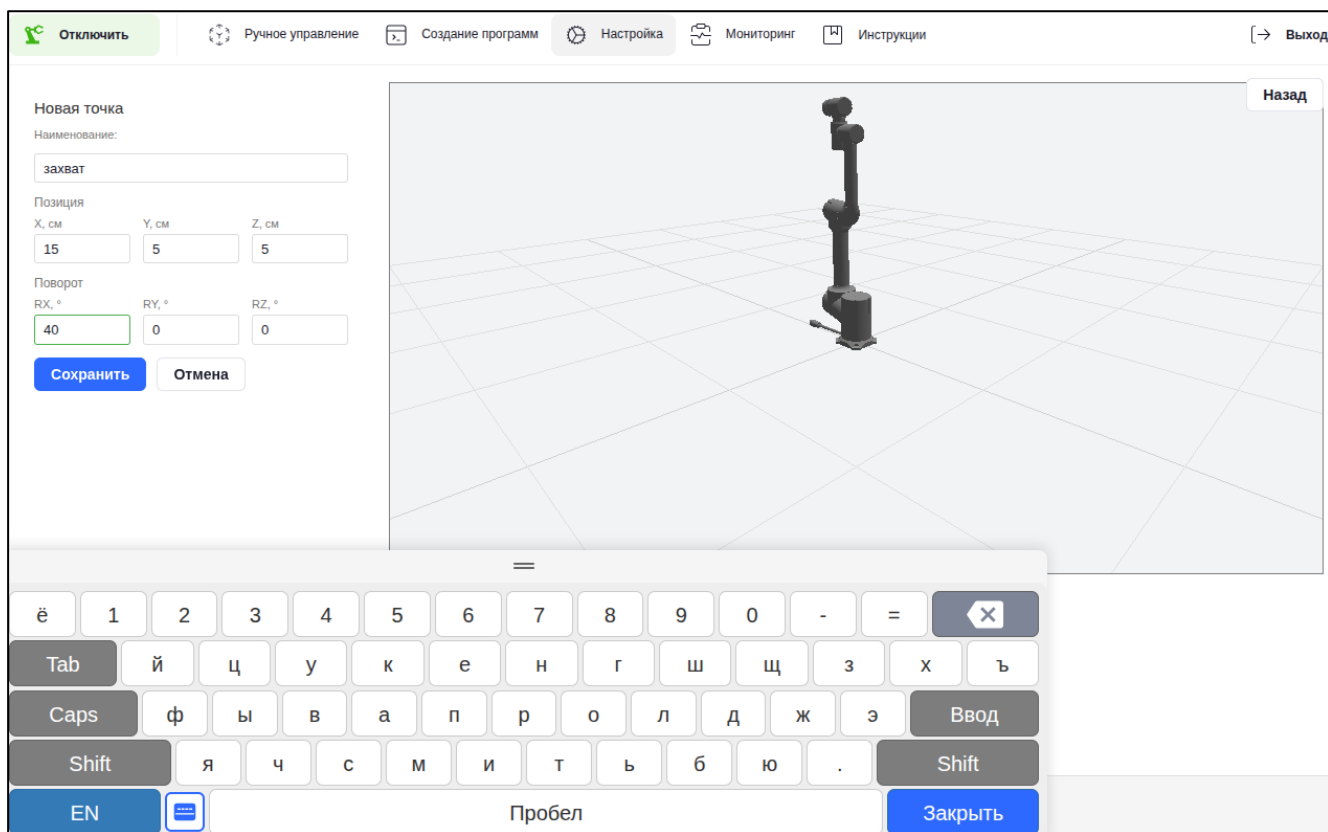


Рисунок II-50 – Форма для добавления новой точки

- б) введите значения позиции в трехмерной системе координат (в сантиметрах) и значения расположения сферически вокруг ЦТИ (в градусах);
- в) нажмите кнопку «Сохранить».

Для отмены добавления инструмента нажмите кнопку «Отмена».

Редактирование созданного инструмента доступно двойным нажатием левой кнопки мыши по наименованию инструмента.

Для созданного инструмента по контекстному меню (см. Рисунок II-51) доступны следующие действия:

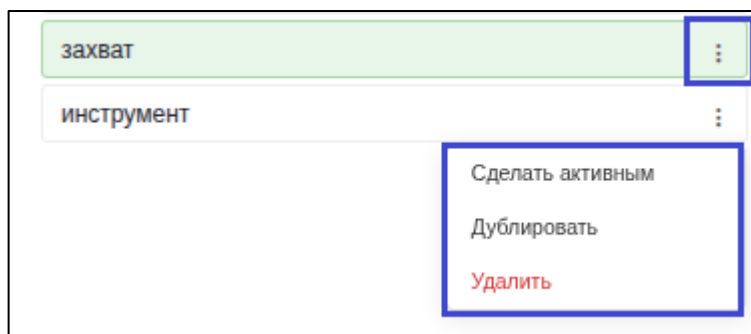


Рисунок II-51 – Контекстное меню

- «Сделать активным» – активировать инструмент;
- «Дублировать» – создать копию инструмента (не доступно при активированном инструменте);
- «Удалить» – удалить инструмент (не доступно при активированном инструменте).

Для деактивации инструмента активируйте инструмент «Без инструмента» либо другой созданный инструмент.

5.3 Рабочая зона

Раздел «Рабочая зона» предназначен для настройки отображения рабочей зоны и крепления манипулятора в виртуальном пространстве.

Раздел содержит 3 подраздела: «Крепление и монтаж», «Границы», «Объекты».

5.3.1 Крепление и монтаж

Подраздел «Крепление и монтаж» позволяет выбрать вариант крепления манипулятора.

Выберите вариант крепления («Напольное», «Потолочное», «Настенное» (см. Рисунок II-52)), либо установите пользовательские значения в полях X, Y, Z в метрах и RX, RY, RZ в градусах («Пользовательское»). Нажмите кнопку «Сохранить» либо «Отменить» для отмены действия.

Внимание! Изменение крепления меняет направление осей основания относительно мировой системы координат. После сохранения обязательно проверьте рабочие зоны, объекты, сохраненные позиции и программы движения.

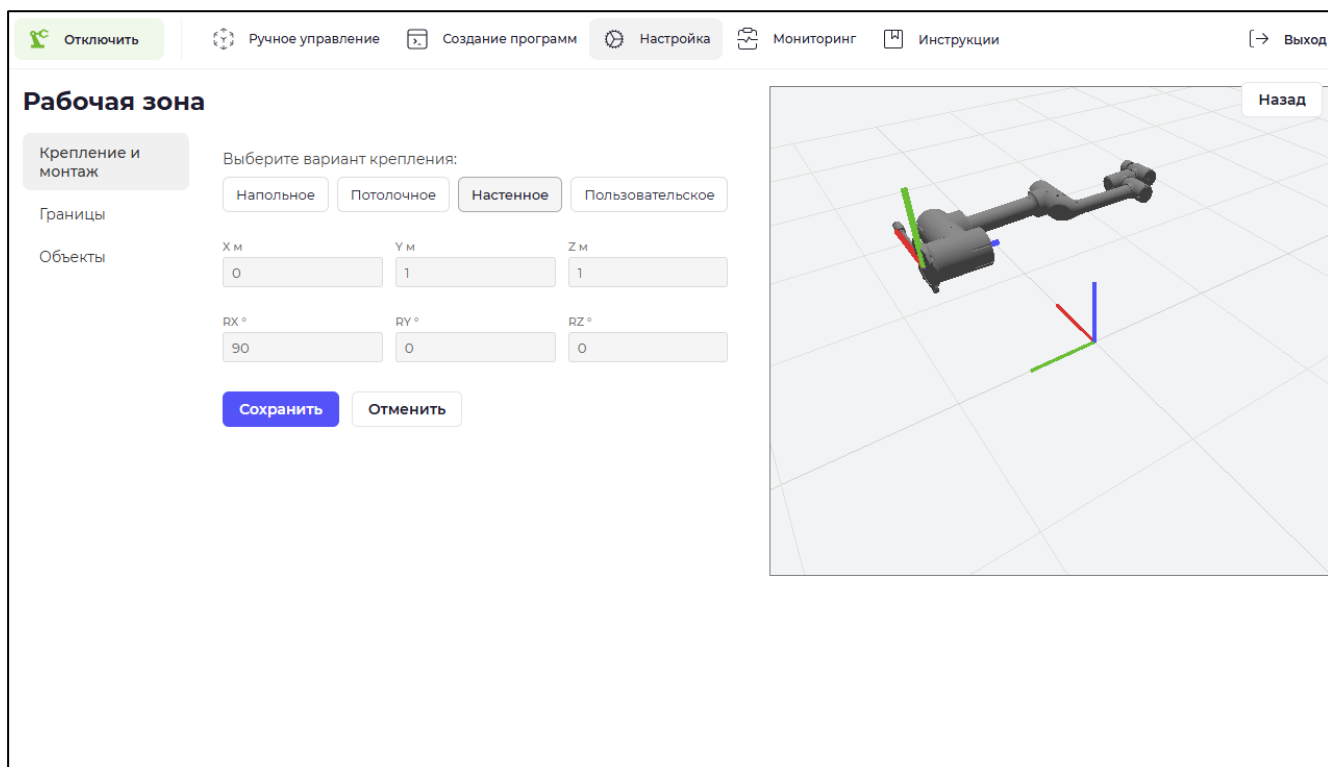


Рисунок II-52 – Подраздел «Крепление и монтаж»

Для возврата к другим настройкам нажмите кнопку «Назад».

5.3.2 Границы

Подраздел «Границы» позволяет настроить границы рабочего окружения манипулятора и отобразить их в виртуальном пространстве, что обеспечивает защиту от столкновения манипулятора с полом и стенами.

Для отображения опорной плоскости манипулятора активируйте слайдер «Включить опорную плоскость». Для отображения стены нажмите кнопку «Добавить стену», установите значения в полях X, Y, Z в метрах и RX, RY, RZ в градусах и нажмите кнопку «Сохранить» (см. Рисунок II-53) либо «Отменить» для отмены действия.

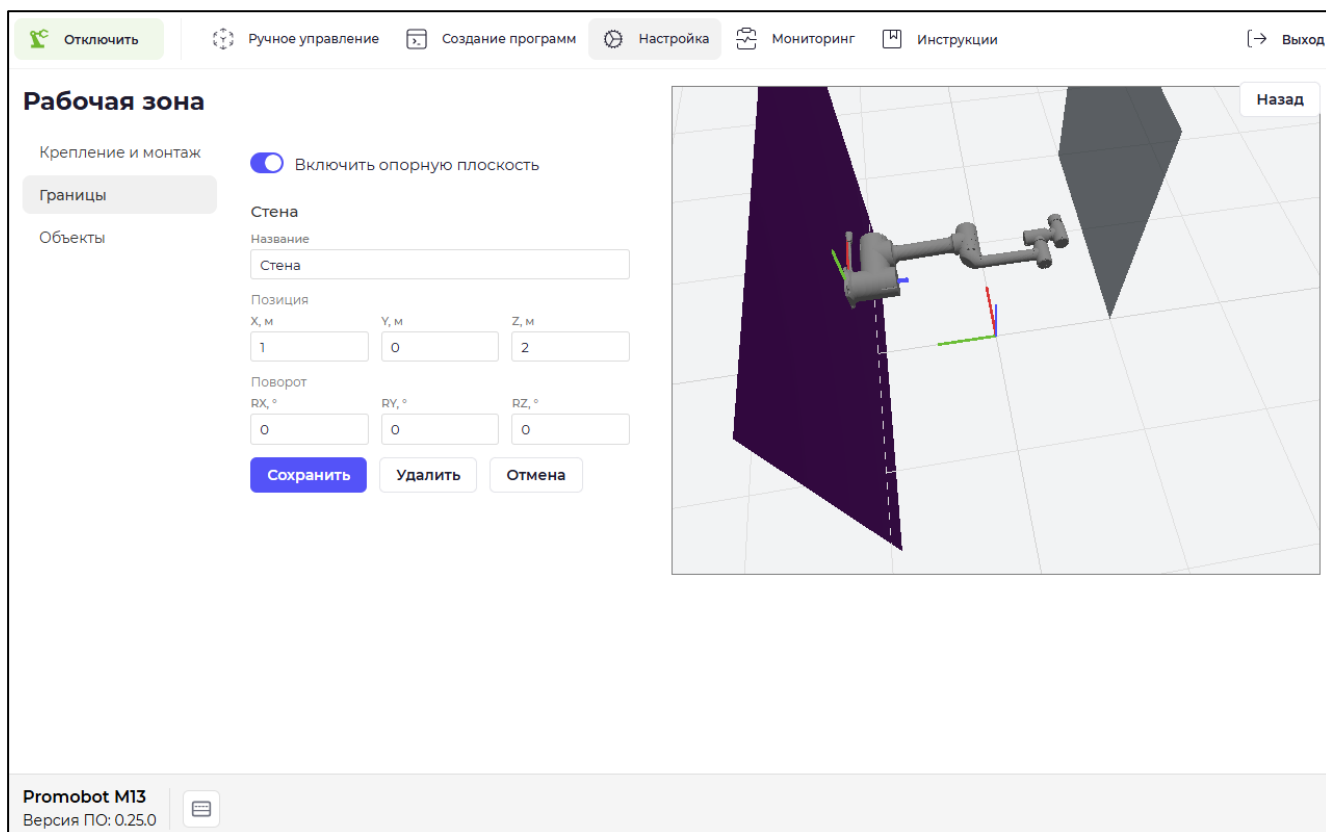


Рисунок II-53 – Подраздел «Границы»

После сохранения проверьте, что виртуальная граница соответствует реальной ячейке.

Для удаления стены нажмите на ее наименование и нажмите кнопку «Удалить».

Для возврата к другим настройкам нажмите кнопку «Назад».

5.3.3 Объекты

Подраздел «Объекты» позволяет добавить новый объект в рабочую зону манипулятора. Объекты используются при проверке коллизий и должны соответствовать фактической геометрии рабочей зоны.

Для добавления объекта нажмите кнопку «Добавить объект», заполните поле «Название», выберите форму объекта («Куб», «Цилиндр», «Сфера»), установите значения в полях X, Y, Z в метрах и RX, RY, RZ в градусах и укажите размер объекта. Нажмите кнопку «Сохранить» (см. Рисунок II-54) либо «Отменить» для отмены действия.

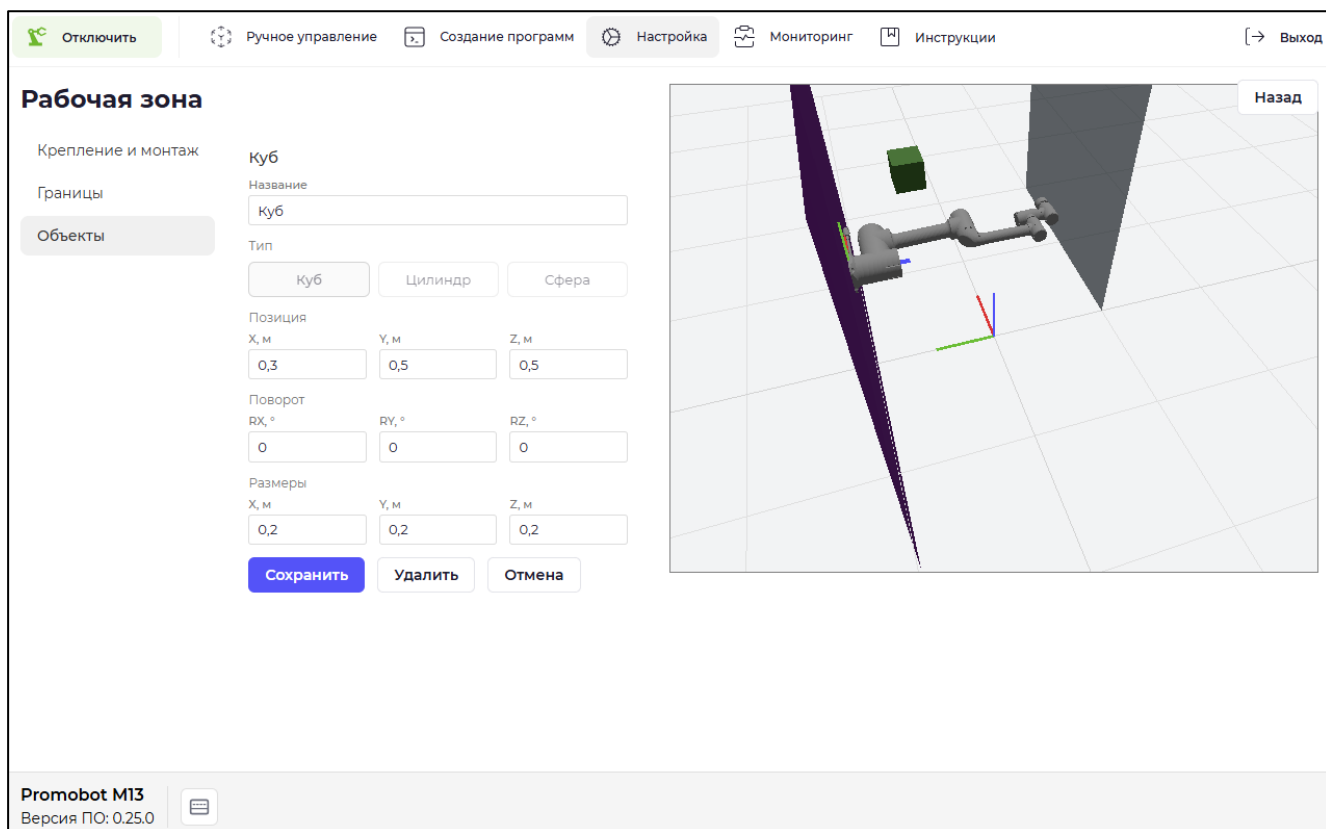


Рисунок II-54 – Подраздел «Объекты»

Для возврата к другим настройкам нажмите кнопку «Назад».

5.4 Положение и пределы

Раздел «Положение и пределы» (см. Рисунок II-55) позволяет задать пределы скоростей и сочленений, чтобы ограничивать рабочий диапазон манипулятора для повышения безопасности в стесненных условиях.

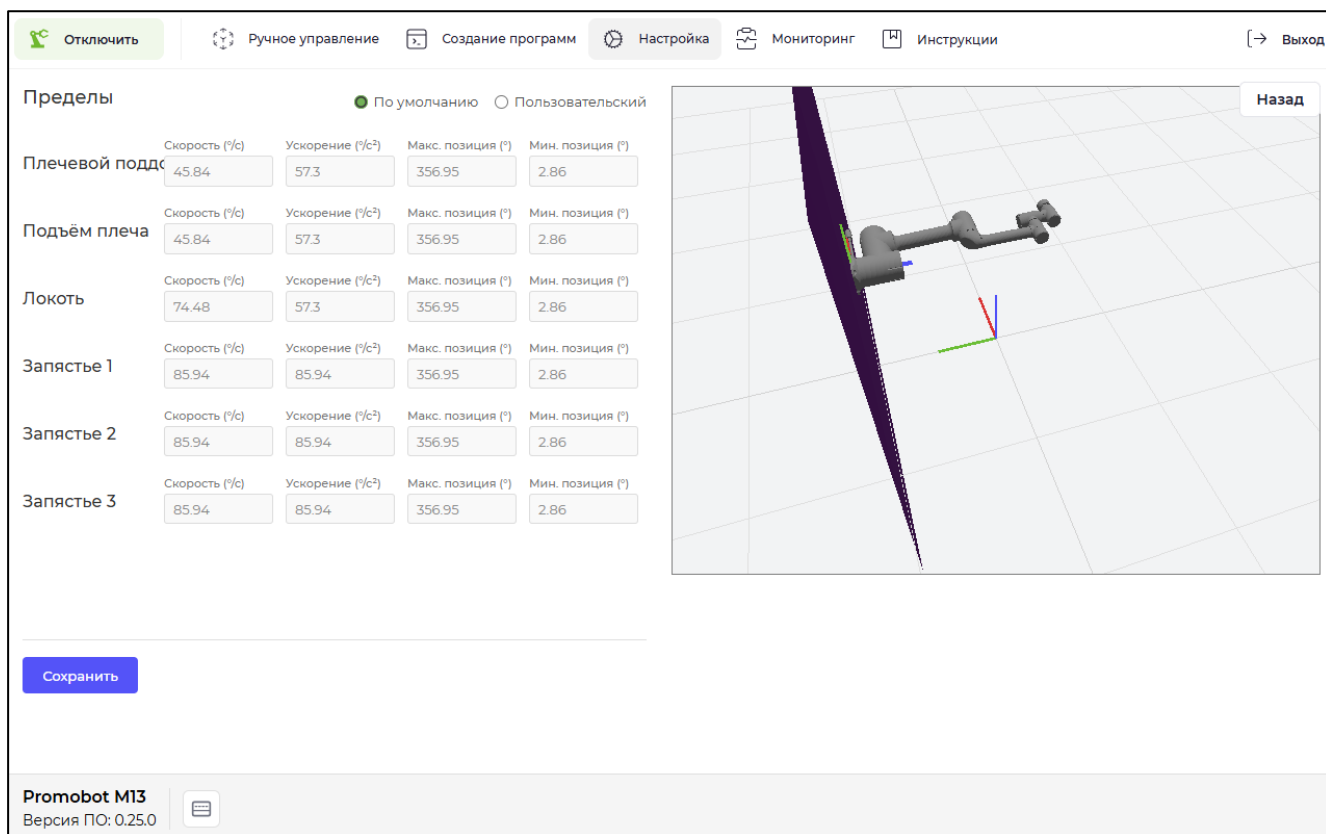


Рисунок II-55 – Раздел «Положение и пределы»

Настройки пределов «По умолчанию» – это максимально допустимые значения. Для настройки новых пределов установите флаг «Пользовательский» – настройки пределов отобразятся в редактируемом формате. Введите значения новых пределов (см. Рисунок II-56) и нажмите кнопку «Сохранить».

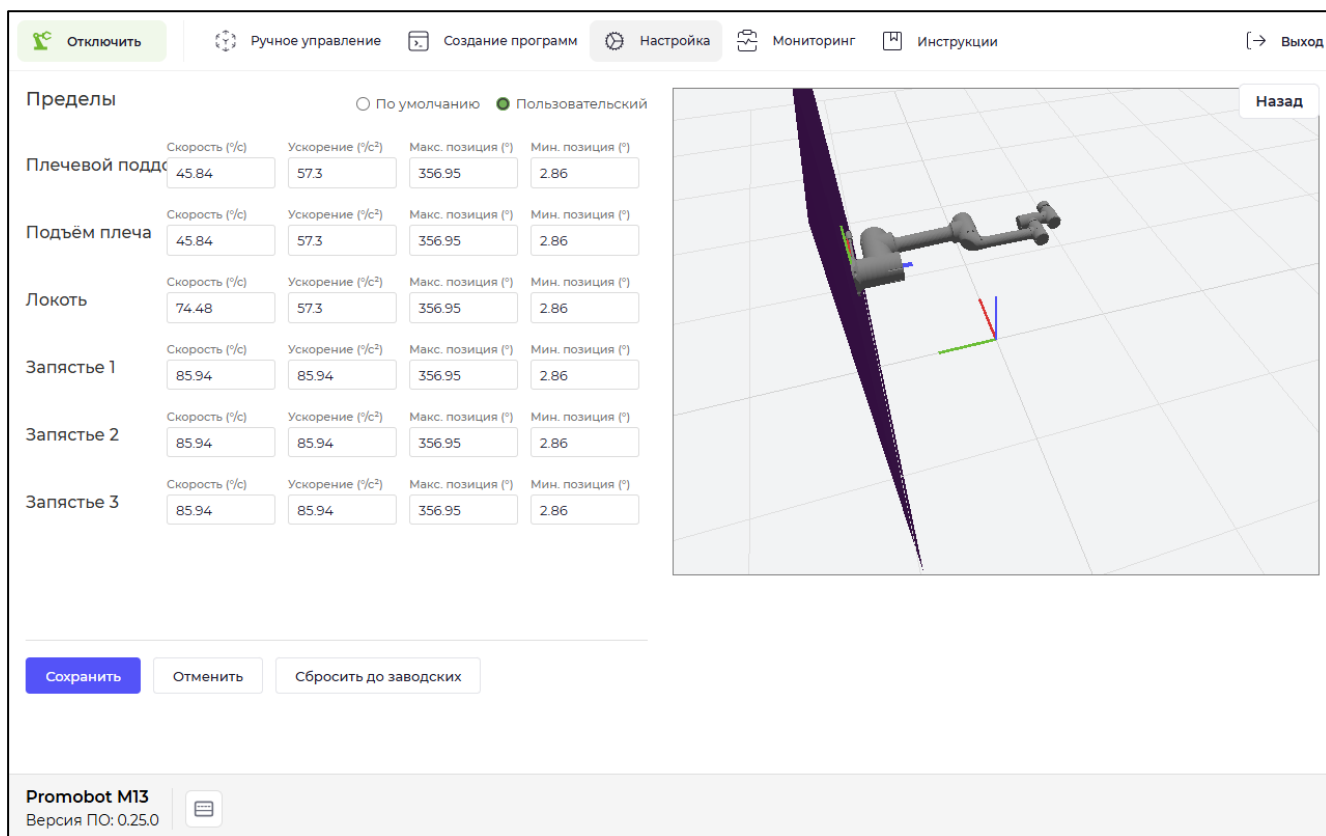


Рисунок II-56 – Настройка новых пределов

Отобразится уведомление применения настроек (см. Рисунок II-57).

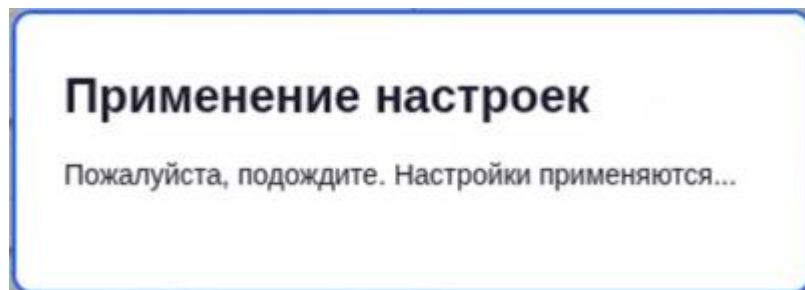


Рисунок II-57 – Уведомление «Применение настроек»

Дождитесь появления уведомления «Настройки применены успешно» (см. Рисунок II-58). Для отмены установки новых пределов нажмите «Отмена».

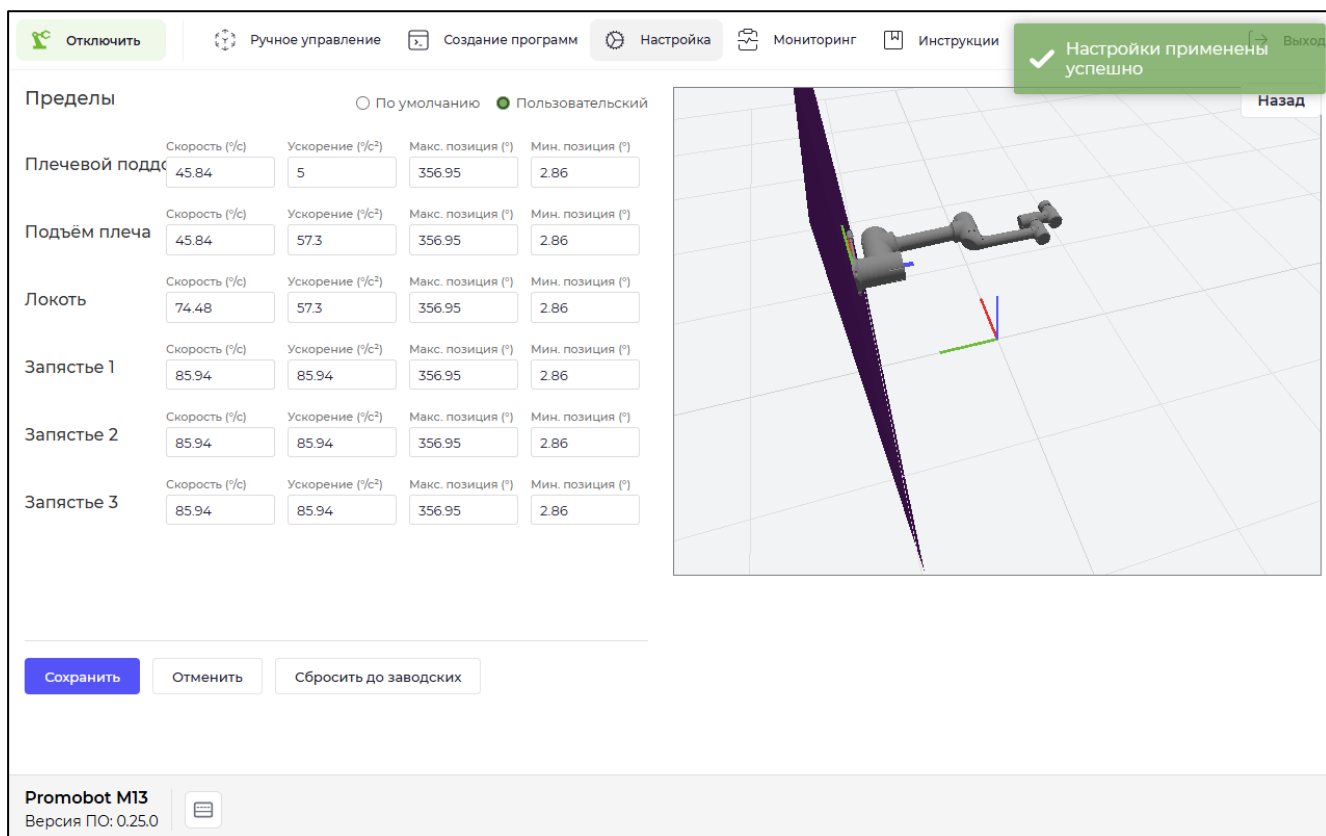


Рисунок II-58 – Уведомление «Настройки применены успешно»

При попытке увести манипулятор за установленные пользовательские пределы, при ручном управлении или при создании программы, отобразится предупреждение (см. Рисунок II-59).

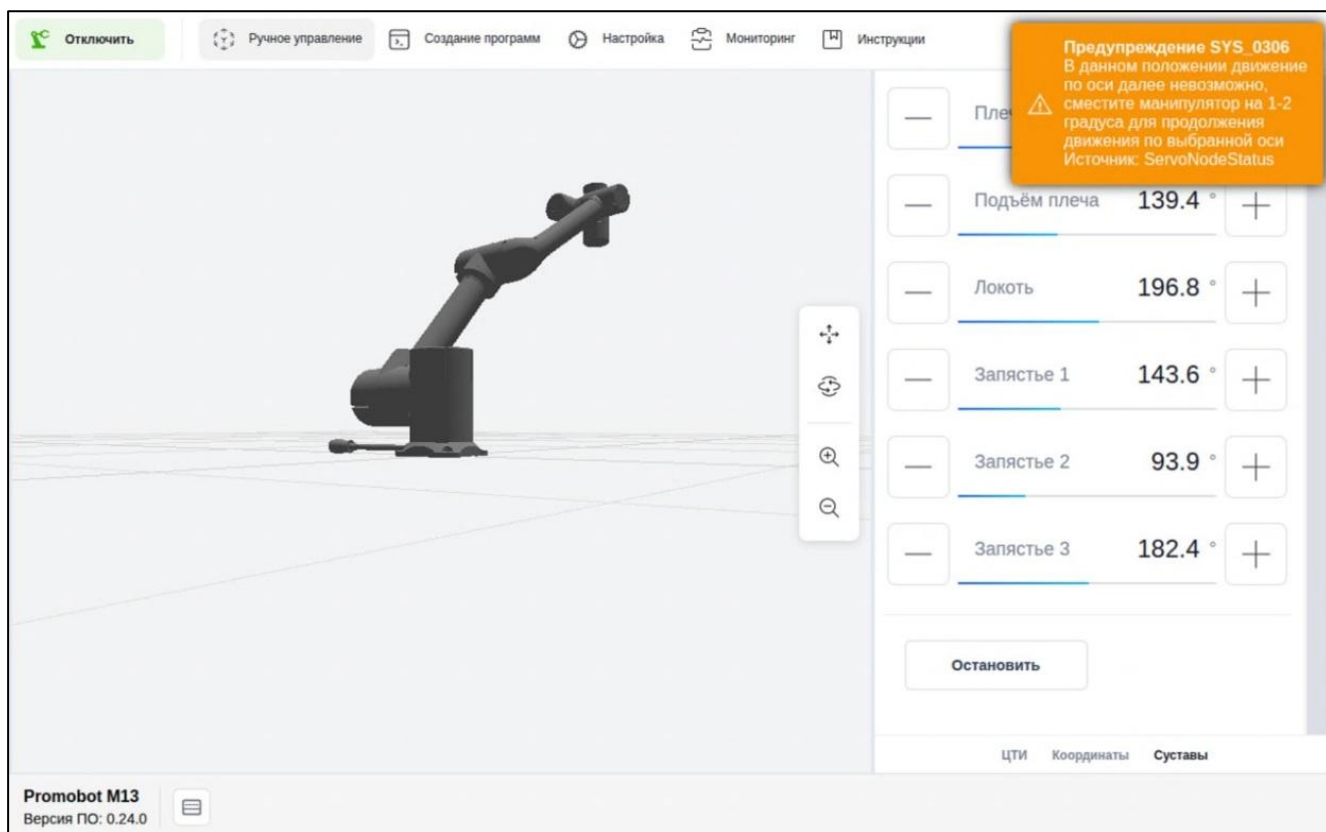


Рисунок II-59 – Предупреждение о невозможности дальнейшего движения манипулятора

Для возврата к настройкам пределов «По умолчанию» перейдите на вкладку «Настройки» – раздел «Положение и пределы», нажмите кнопку «Сбросить до заводских», а затем кнопку «Сохранить», либо установите флаг «По умолчанию» и нажмите кнопку «Сохранить».

5.5 Сервисные функции

Примечание – Раздел доступен только для пользователя с доступом к сервисным функциям.

Раздел «Сервисные функции» позволяет переводить манипулятор в транспортное положение и производить движения манипулятора вручную без инициализации.

Раздел содержит два подраздела: «Транспортное положение» и «Движение без инициализации».

5.5.1 Транспортное положение

Подраздел «Транспортное положение» предназначен для перевода манипулятора в транспортное положение, для этого выполните следующие действия:

- а) отключите манипулятор в приложении (см. Рисунок II-7);
- а) смените пользователя в приложении на пользователя с доступом к сервисным функциям (см. Рисунок II-8);
- б) подключите манипулятор в приложении (см. Рисунок II-6);
- в) откройте вкладку «Настройка» – раздел «Сервисные функции». Отобразится интерфейс сервисных функций (см. Рисунок II-60);

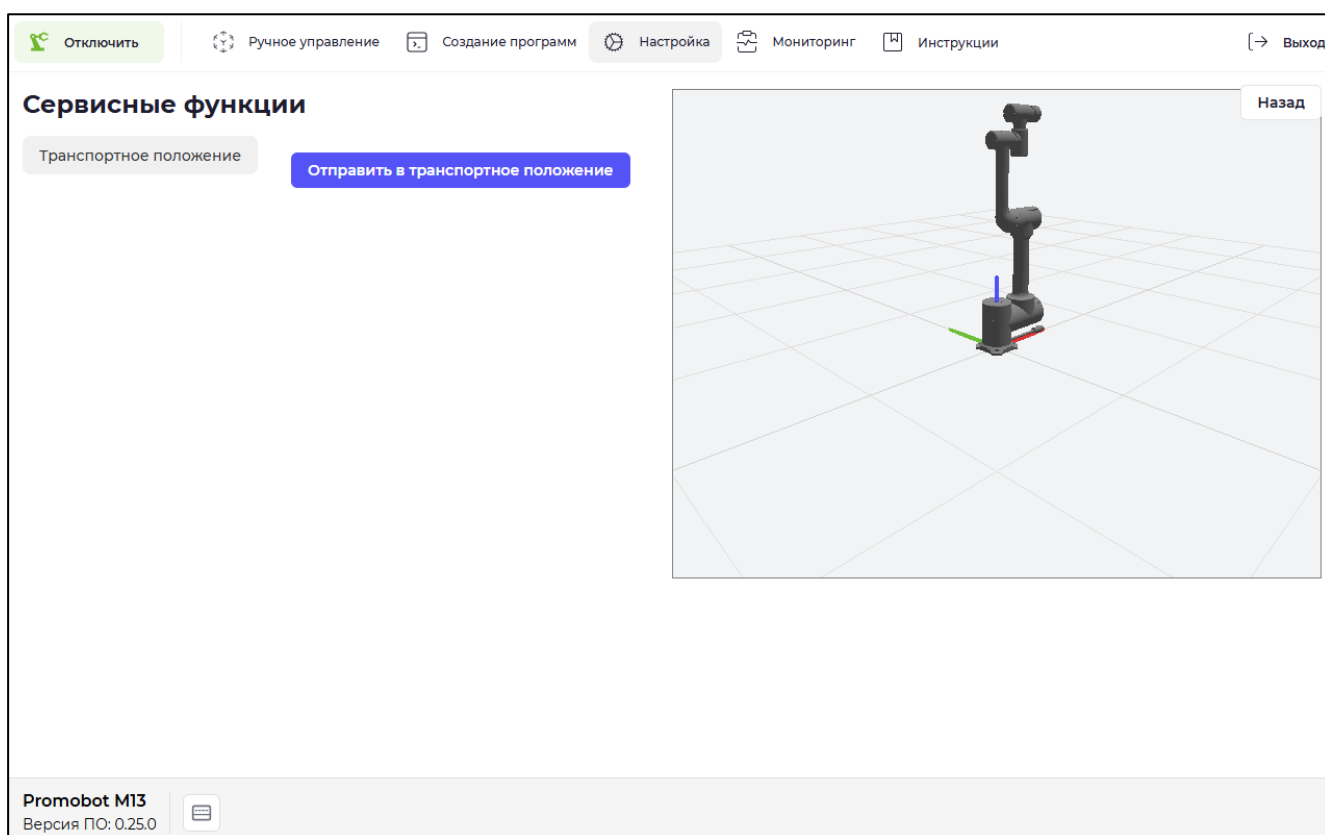


Рисунок II-60 – Вкладка «Настройки»

- г) нажмите кнопку «Отправить в транспортное положение»;
- д) дождитесь пока манипулятор примет транспортное положение и на экране отобразится уведомление «Программа выполнена успешно» (см. Рисунок II-61+).

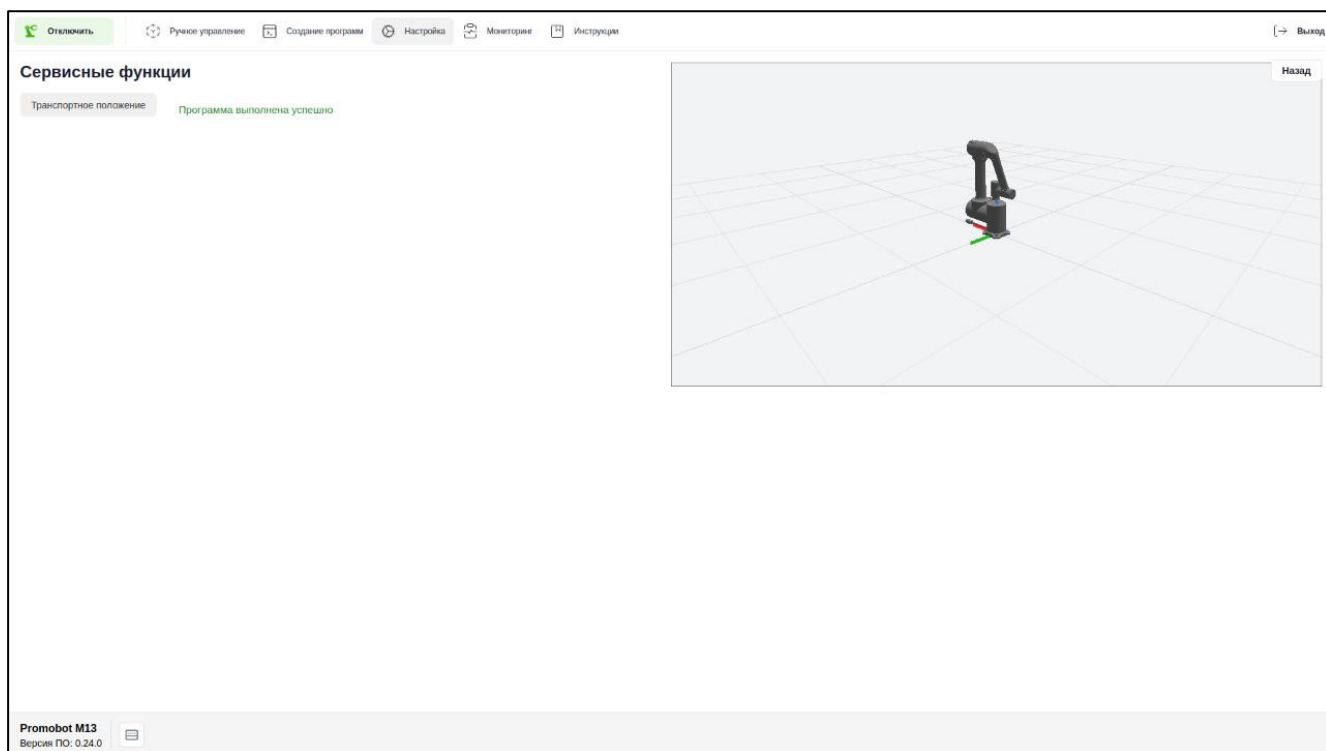


Рисунок II-61 – Раздел «Сервисные функции»

5.5.2 Движение без инициализации

Подраздел «Движение без инициализации» – это ручной режим перемещения на манипуляторе, предназначенный для того, чтобы вывести манипулятор из зоны коллизии, либо вернуть его в исходное положение при сбое датчиков. Чтобы осуществить движение без инициализации выполните следующие действия:

- а) отключите манипулятор в приложении (см. Рисунок II-7);
- б) смените пользователя в приложении на пользователя с доступом к сервисным функциям (см. Рисунок II-8);
- в) откройте вкладку «Настройка» – раздел «Сервисные функции». Отобразится интерфейс сервисных функций;
- г) выберите подраздел «Движение без инициализации», отобразится интерфейс управления движением без инициализации (см. Рисунок II-62);

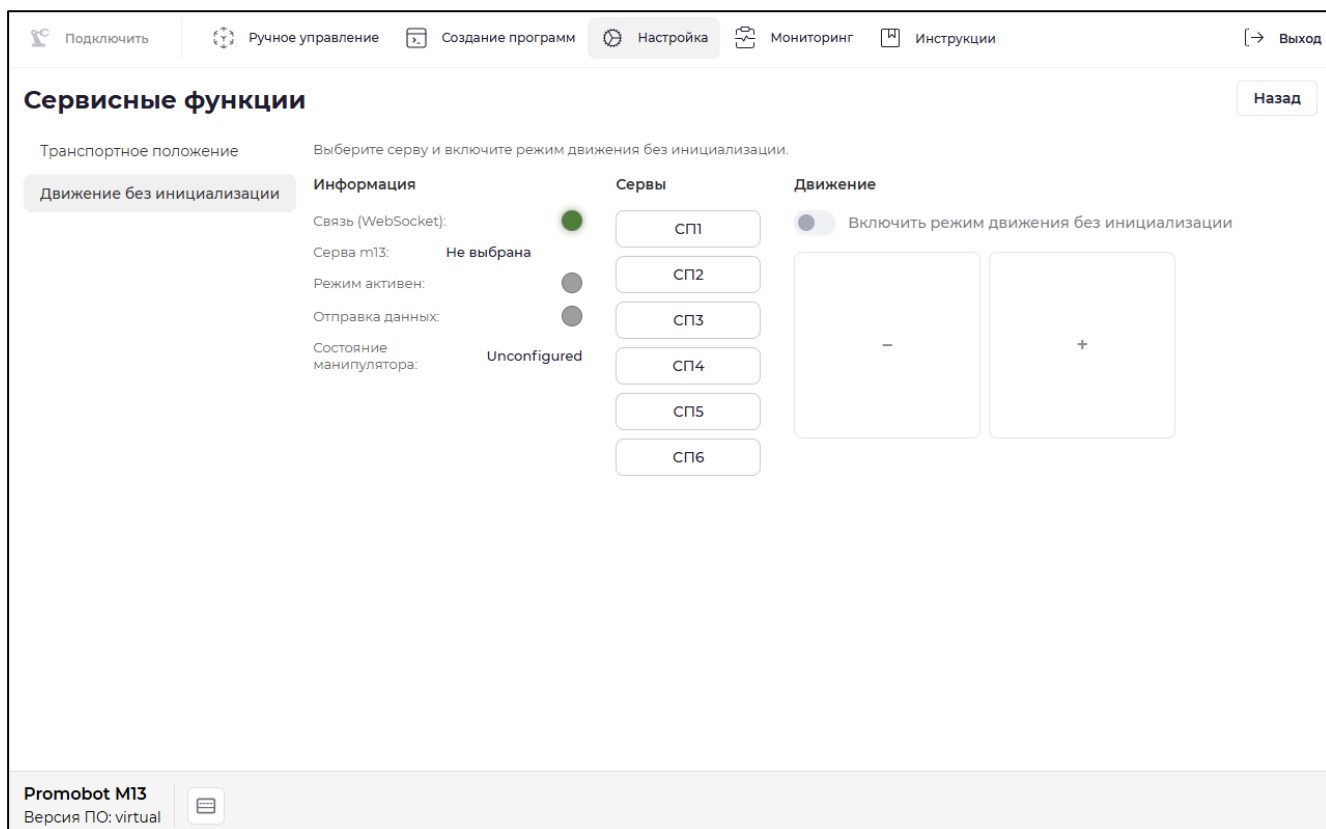


Рисунок II-62 – Подраздел «Движение без инициализации»

- д) выберите СП для движения, активируйте слайдер «Включить режим движения без инициализации» и управляйте движением с помощью кнопок «-» и «+» (см. Рисунок II-63).

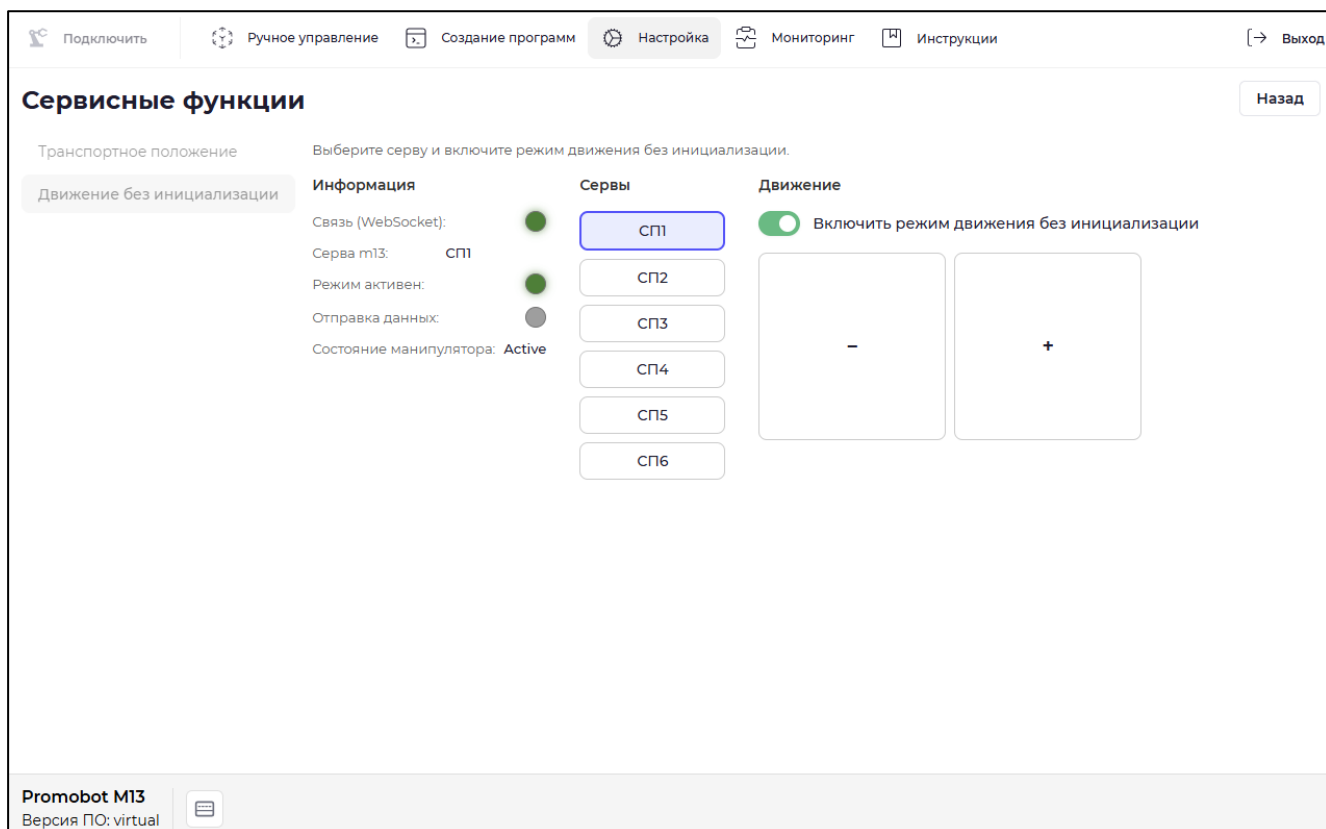


Рисунок II-63 – Включение режима движения без инициализации

6 МОНИТОРИНГ

Интерфейс мониторинга (вкладка «Мониторинг») предназначен для журналирования событий, отображения показателей работы манипулятора, информирования о версии релиза ПО и актуальных прошивок плат. Актуальные данные отображаются только при подключенном манипуляторе.

Интерфейс мониторинга включает в себя следующие разделы:

- а) раздел «Журнал» (см. Рисунок II-64) предназначен для ведения журнала следующих событий:
 - срабатывание кнопки аварийного останова;
 - предупреждения, присылаемые с платы безопасности, платы инструмента, плат СП.
 - ошибки, присылаемые с платы безопасности, платы инструмента, плат СП.
 - потеря связи с приложением Promobot M Control;

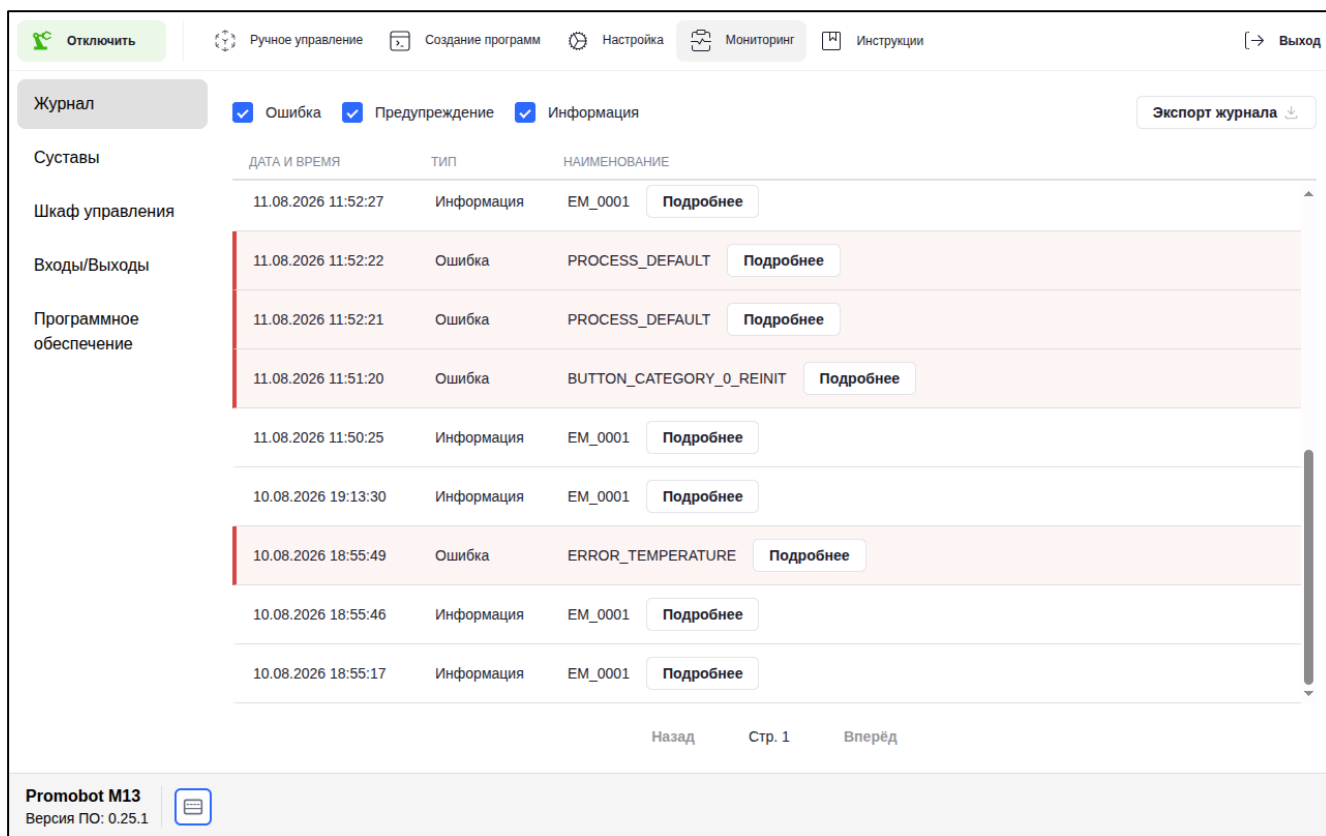


Рисунок II-64 – Раздел «Журнал»

Журнал отображает на экране 50 последних записей с возможностью прокрутки и постраничного перехода. Пользователь имеет возможность фильтровать записи по типу сообщения (ошибки, предупреждения, информация). Записи имеют цветовую маркировку: для ошибок – красный, для предупреждений – желтый, для информации – белый. При двойном нажатии на строку записи (или отдельной кнопкой «Подробнее») открывается окно с описанием ошибки/события (см. Рисунок II-65).

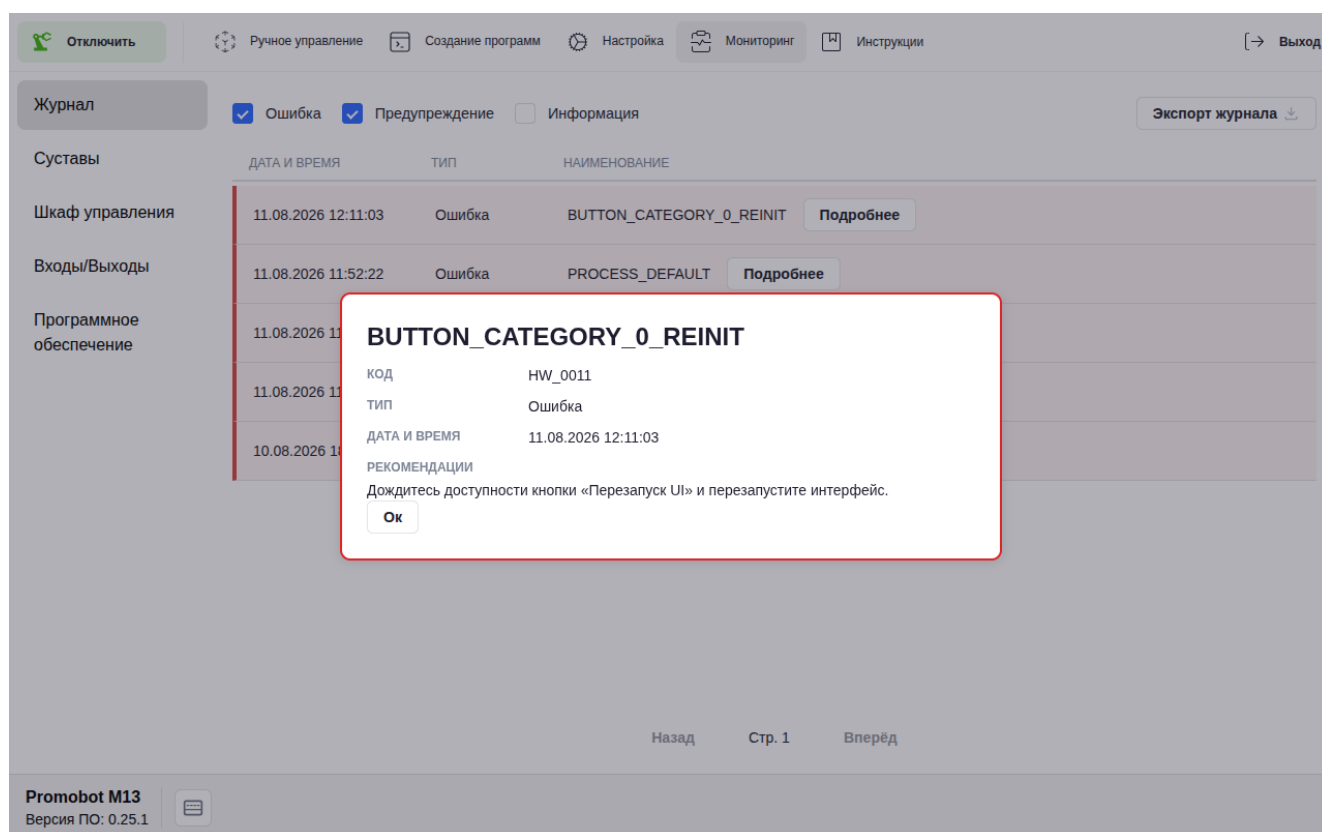


Рисунок II-65 – Окно с описанием ошибки

Журнал можно экспортировать в формате *.csv с помощью кнопки «Экспорт журнала».

- б) раздел «Суставки» (см. Рисунок II-66) предназначен для мониторинга температуры и напряжения каждого из шести суставов. Значения следует оценивать совместно с предупреждениями и условиями работы робота. Нулевые, отсутствующие или не обновляющиеся значения означают отсутствие актуальной телеметрии либо неподключенный узел и не должны трактоваться как нормальное состояние;

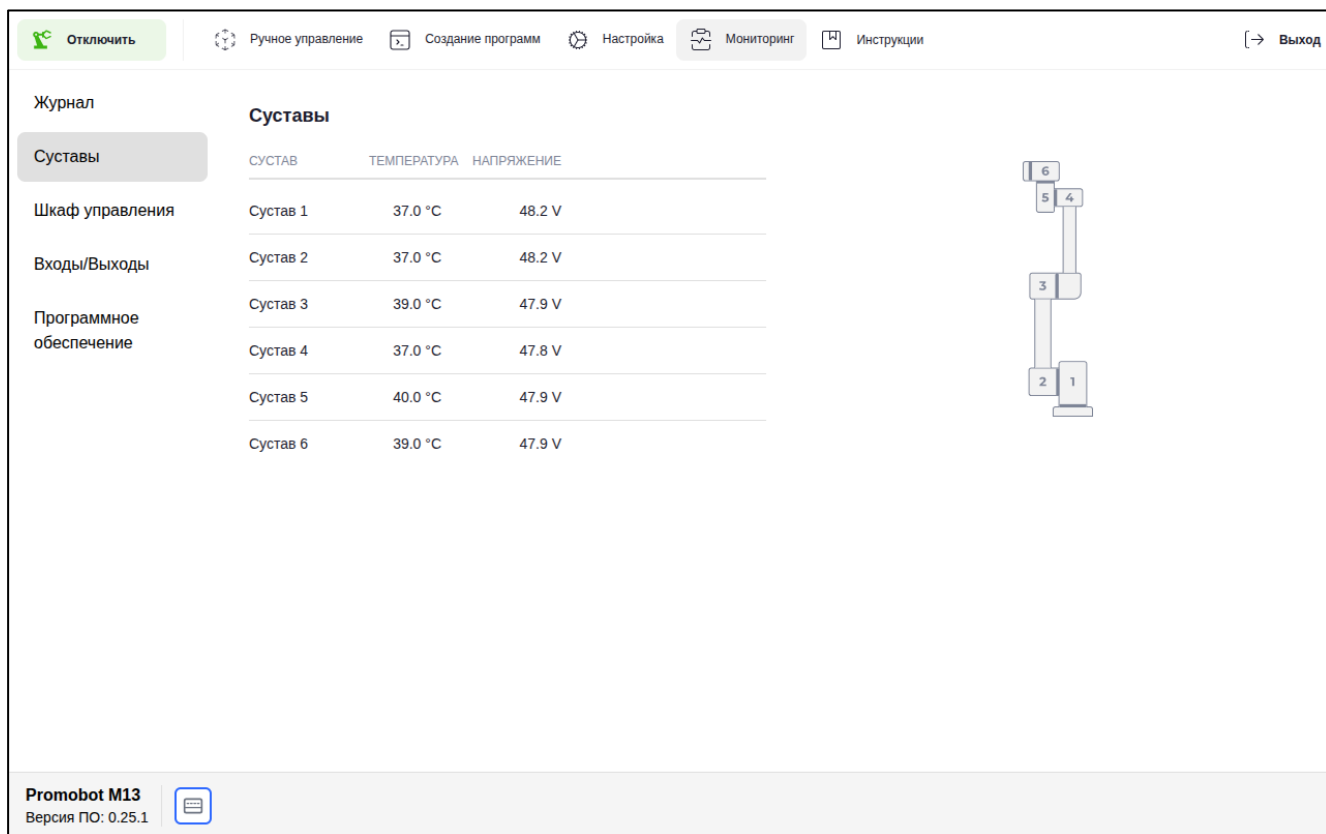


Рисунок II-66 – Раздел «Суставы»

- в) раздел «Шкаф управления» предназначен для мониторинга значений системного тока и напряжения в шкафу во время выполнения программы;

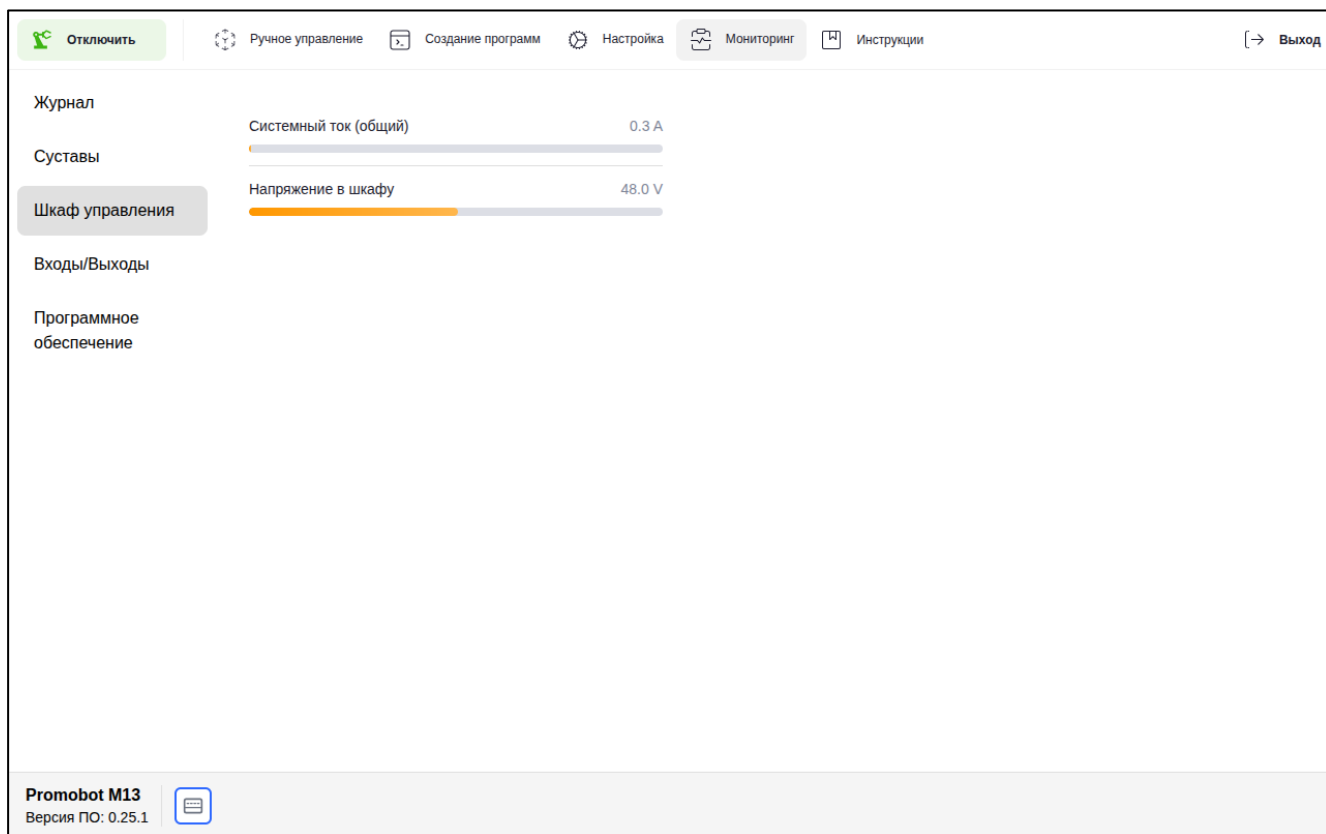


Рисунок II-67 – Раздел «Шкаф управления»

- г) раздел «Входы/Выходы» предназначен для мониторинга:
- сигналов пользовательской панели (подраздел «Пользовательская панель»)
(см. Рисунок II-68);

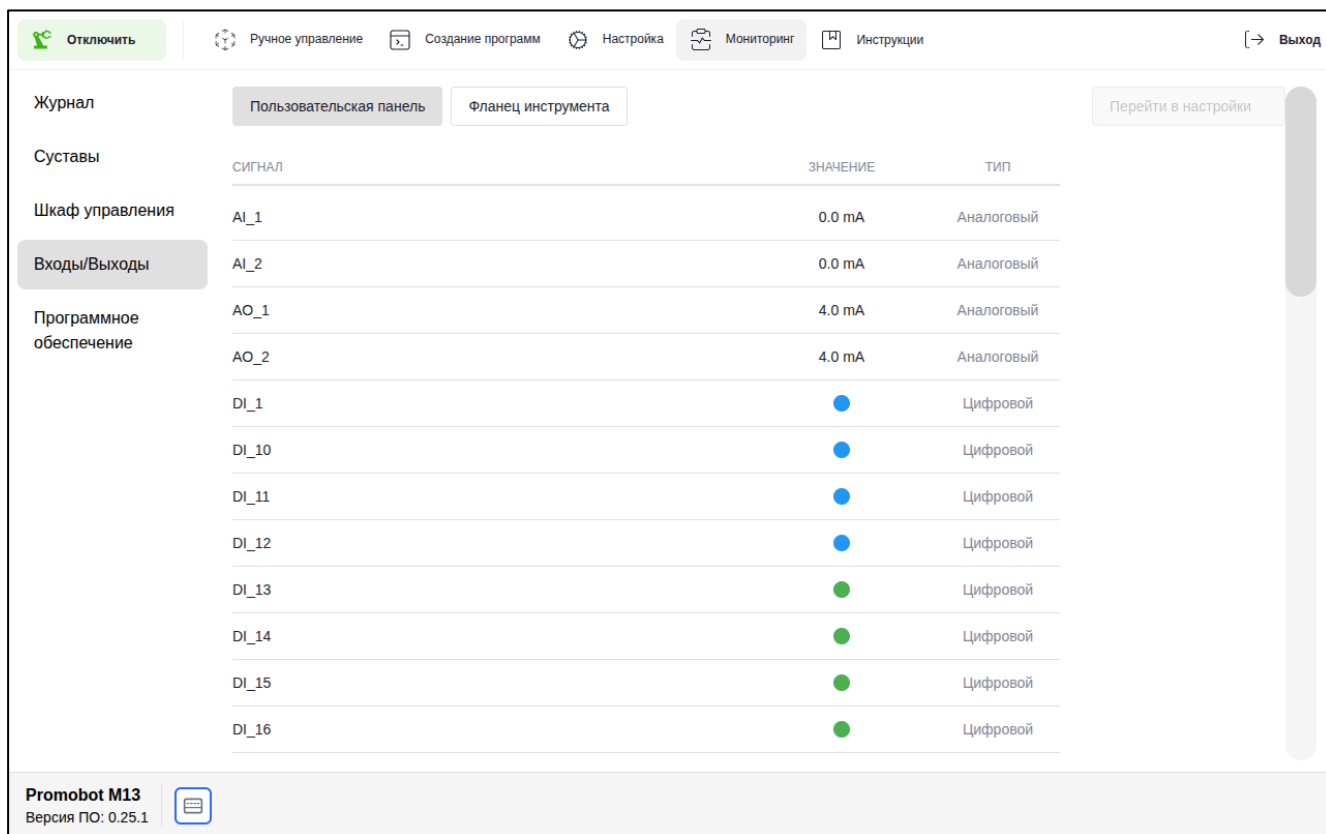


Рисунок II-68 – Подраздел «Пользовательская панель»

- сигналов разъемов инструмента (подраздел «Фланец инструмента») (см. Рисунок II-69);

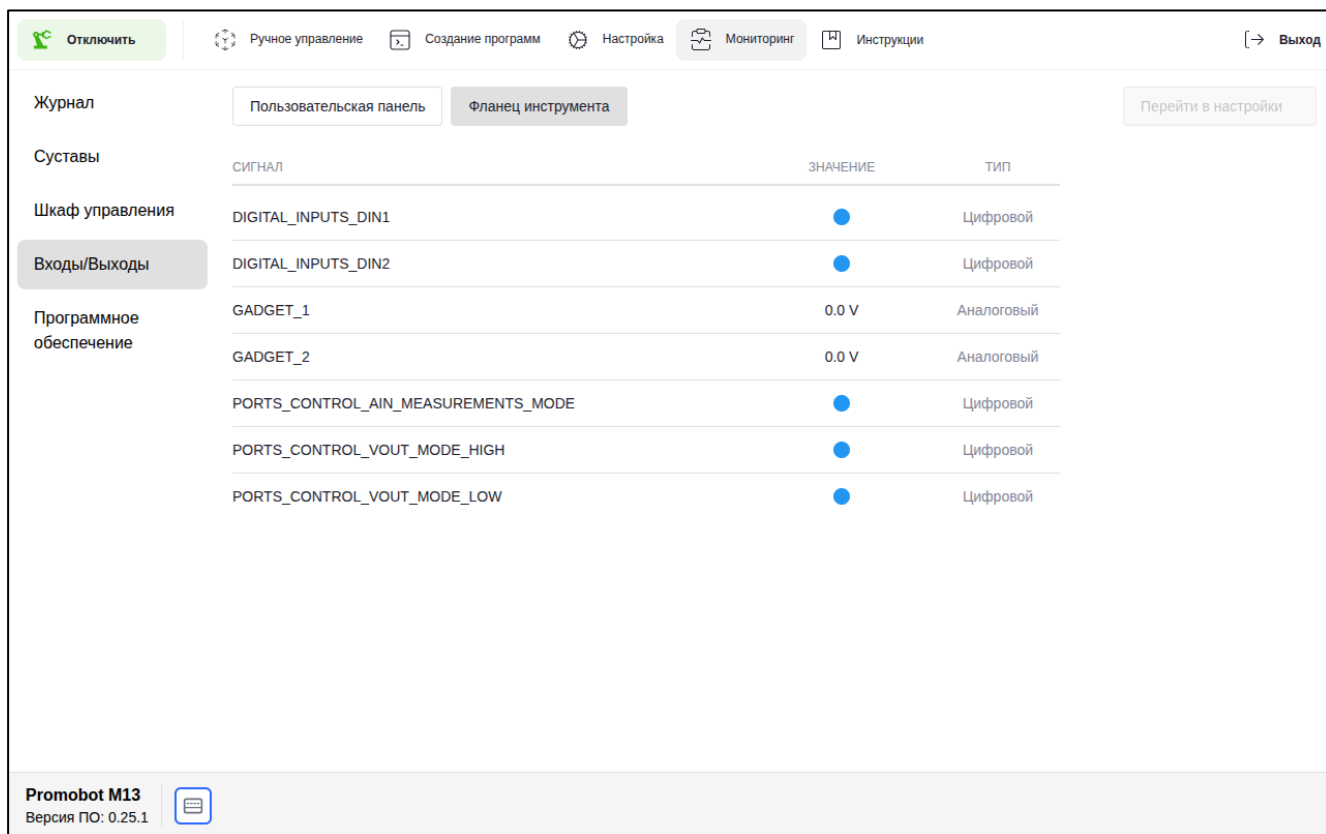


Рисунок II-69 – Подраздел «Фланец инструмента»

Значения сигналов:

- цветовая индикация на цифровых сигналах: зеленый – включен, синий – выключен, серый – неактивен;
 - числовое значение для аналоговых сигналов – назначается пользователем вручную при создании программ.
- д) раздел «Программное обеспечение» (см. Рисунок II-70) – предназначен для отображения версий системного ПО (релиз), веб-интерфейса, SDK, прошивки шкафа управления (Safety), платы безопасности (PCB), прошивки инструмента (Gadget), платы инструмента (PCB), прошивки СП.

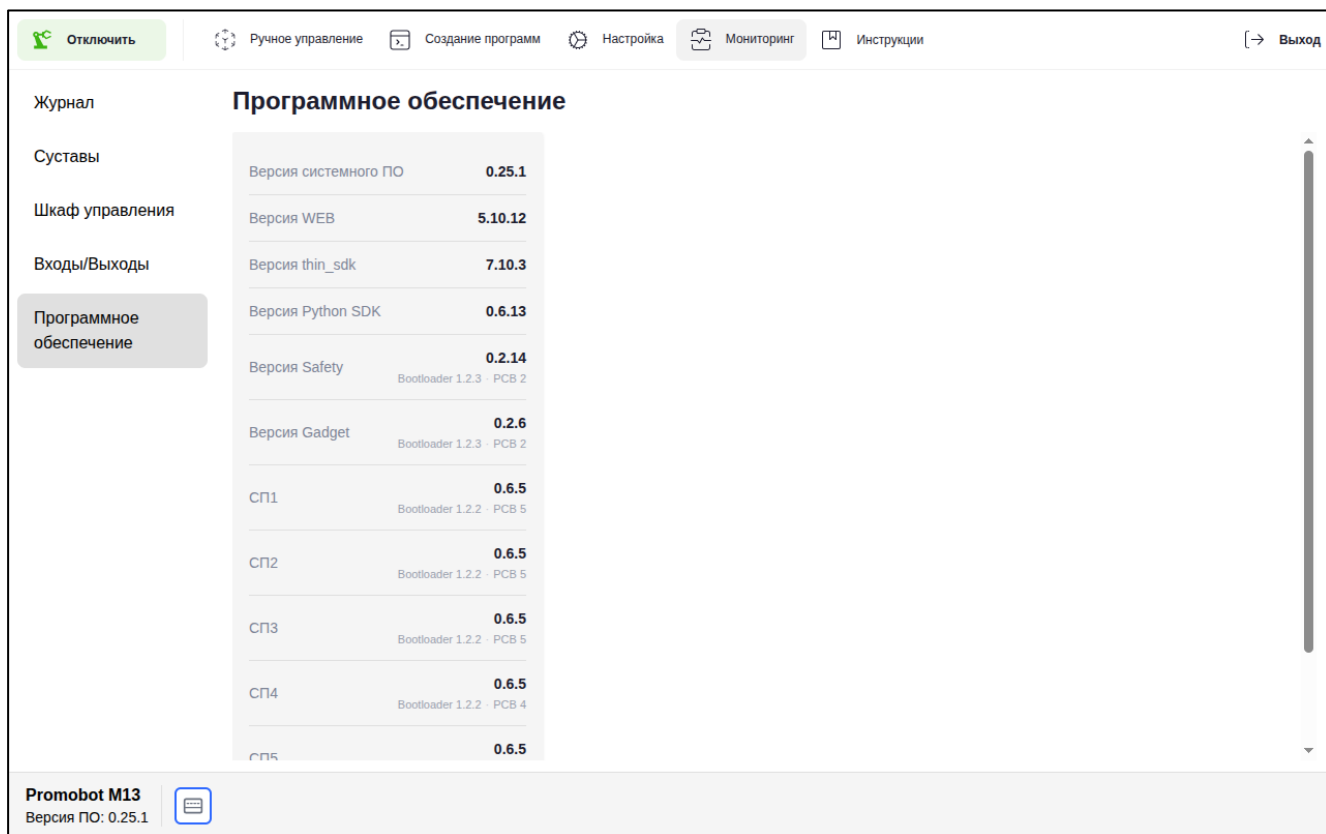


Рисунок II-70 – Раздел «Программное обеспечение»

7 ОБНОВЛЕНИЕ ПРИЛОЖЕНИЯ

Перед обновлением сравните версии интерфейса, релиза и прошивки вашего Promobot M13 с версиями в последнем релизе. Информацию о последнем релизе вы можете узнать в инструкции (в приложении нажмите кнопку «Инструкции» и перейдите в раздел 8 «РЕЛИЗ»).

При несоответствии, приведите манипулятор в исходное положение, красная кнопка должна быть отжата, подключите Promobot M13 к интернету и выполните все обновления по очереди:

- а) обновите ПО:
 - 1) закройте приложение;
 - 2) запустите файл «Update_Mcontrol» на рабочем столе (см. Рисунок II-71);

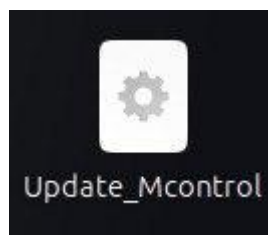


Рисунок II-71 – Файл «Update_Mcontrol»

- 3) откроется программный интерфейс «Terminal». Отобразится диалоговое окно для выбора типа обновления (см. Рисунок II-72);

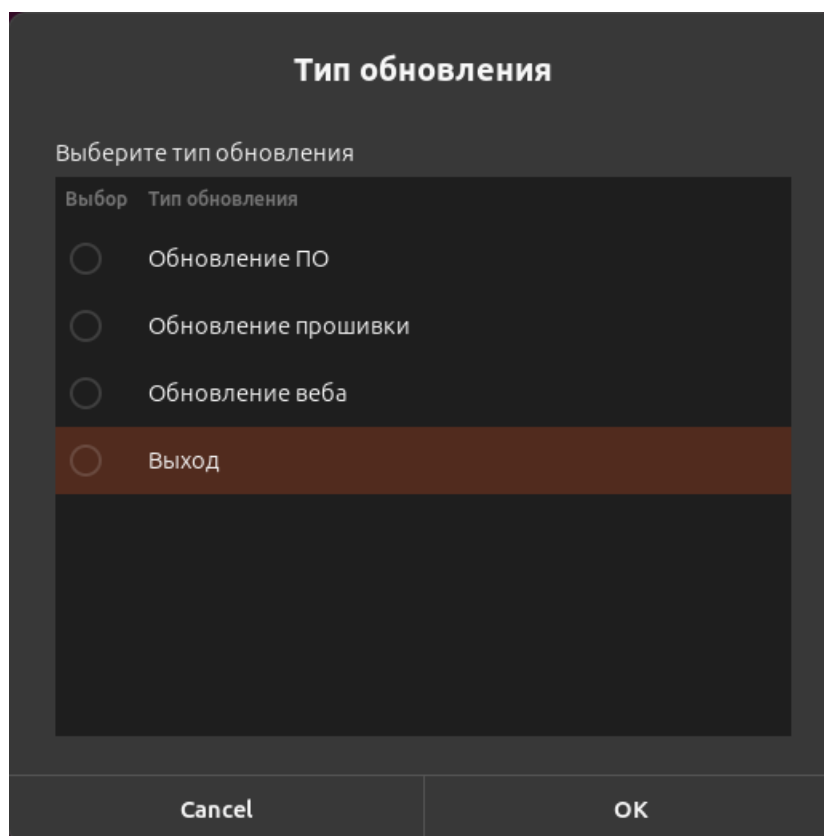


Рисунок II-72 – Диалоговое окно для выбора типа обновления

- 4) выберите «Обновление ПО» и нажмите «ОК». Отобразится уведомление о запуске обновления;
- 5) по завершению обновления отобразится уведомление о завершении. Нажмите «ОК»;

Дождитесь появления диалогового окна для выбора типа обновления (см. Рисунок II-72).

- б) обновите веб-интерфейс:

- 1) выберите «Обновление веба» и нажмите «ОК»;

2) по завершению обновления отобразится уведомление о завершении.

Нажмите «ОК»;

Дождитесь появления диалогового окна для выбора типа обновления (см. Рисунок II-72).

в) обновите прошивку:

1) выберите «Обновление прошивки» и нажмите «ОК». Отобразится диалоговое окно для выбора устройства для прошивки (см. Рисунок II-73);

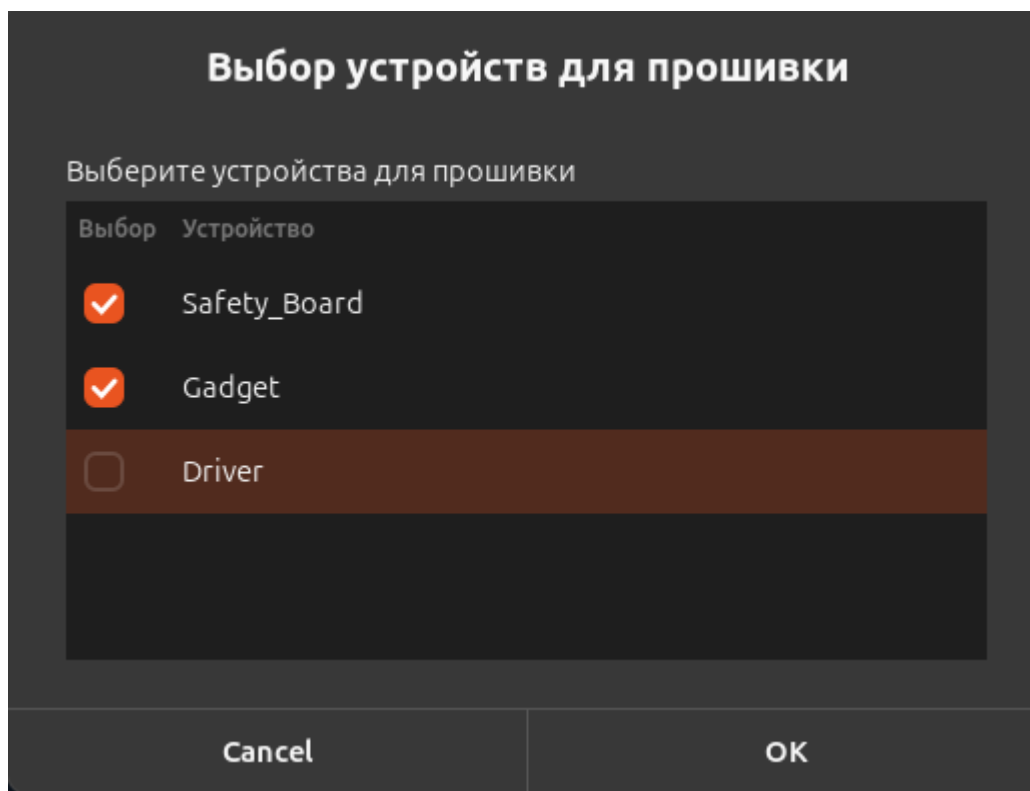


Рисунок II-73 – Диалоговое окно для выбора устройства для прошивки

2) выберите одно или несколько устройств и нажмите «ОК». Отобразится уведомление «Найдена прошивка: ... Продолжить прошивку?» (см. Рисунок II-74);

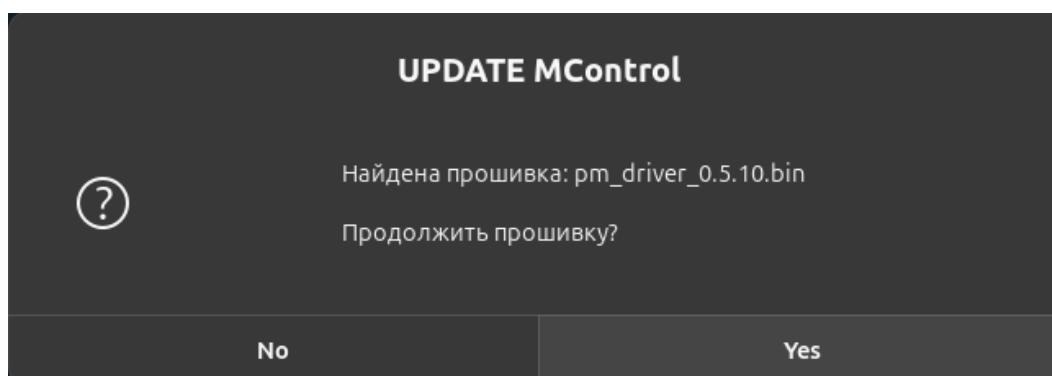


Рисунок II-74 – Уведомление прошивки

3) нажмите «Yes». При обновлении:

- «Gadget», «Driver» – отобразится предупреждение «Перед началом процедуры обновления драйверов: 1) Манипулятор в исходном положении. 2) Кнопка аварийной остановки ОТЖАТА. Будет обновлено ... драйверов суставов. Продолжить?» (см. Рисунок II-75);

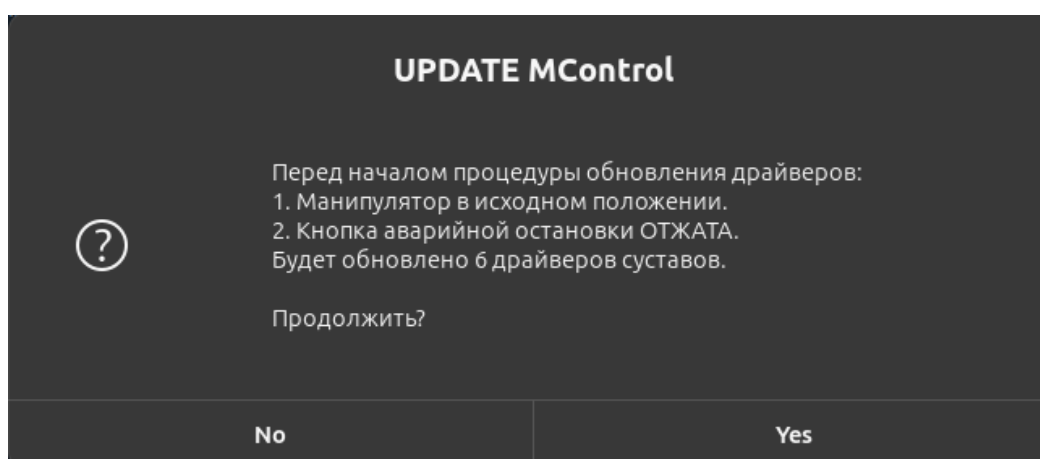


Рисунок II-75 – Предупреждение

- «Safety Board» – отобразится предупреждение «Перед началом процедуры обновления Safety Board: 1) Манипулятор в исходном положении. 2) Кнопка аварийной остановки НАЖАТА. Продолжить?» (см. Рисунок II-76);

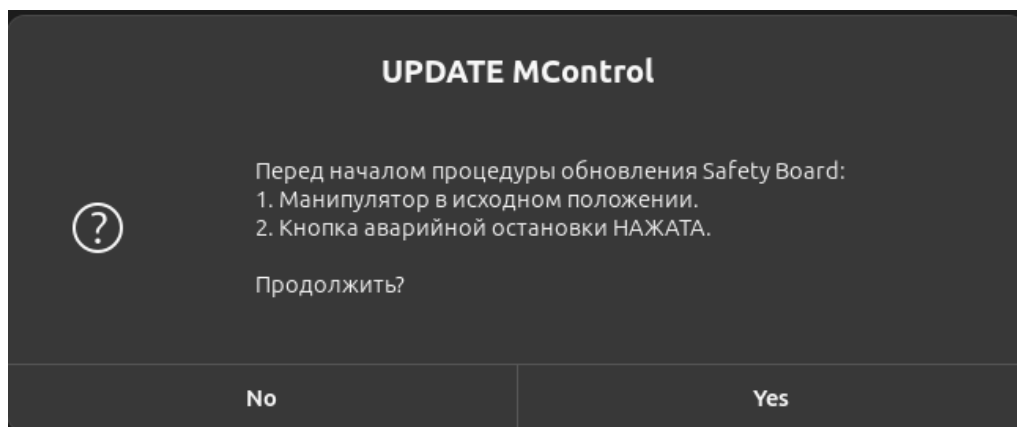


Рисунок II-76 – Предупреждение

- 4) если все условия выполнены, то нажмите «Yes». Отобразится уведомление о начале обновления (см. Рисунок II-77);

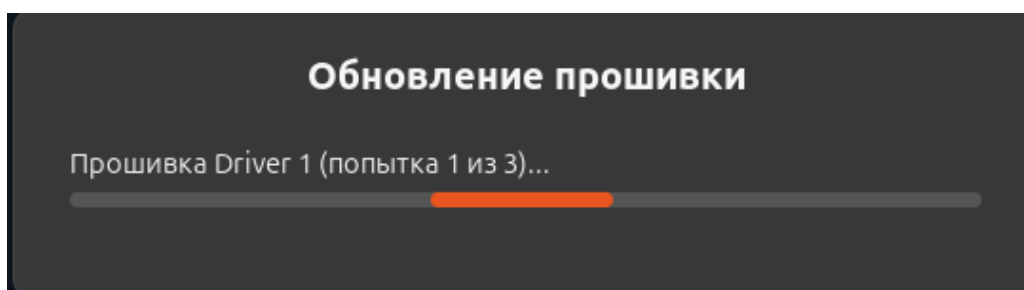


Рисунок II-77 – Уведомление об обновлении

- 5) при обновлении:
- успешном – отобразится уведомление о завершении, нажмите «ОК»; закройте диалоговое окно, нажав кнопку «Cancel»;
 - неуспешном – запустится восстановление устройства (см. Рисунок II-78).

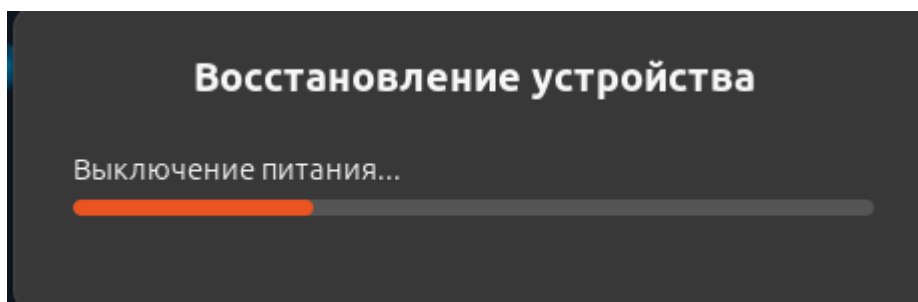


Рисунок II-78 – Уведомление о восстановлении устройства

- 6) после восстановления устройства обновление прошивки запустится автоматически (см. Рисунок II-79);

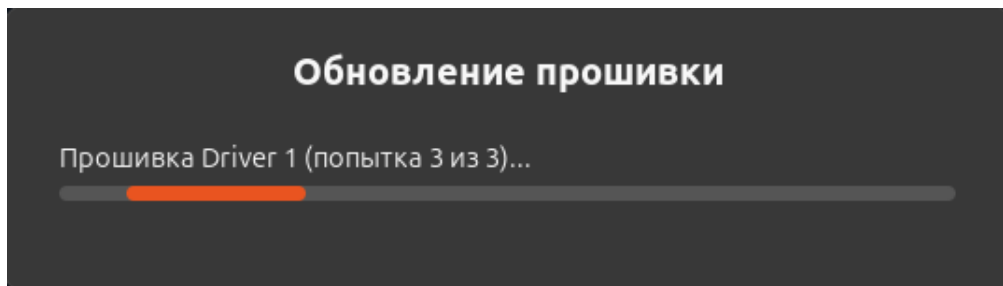


Рисунок II-79 – Уведомление об обновлении

- 7) если повторное обновление прошивки пройдет неуспешно, то отобразится уведомление о критической ошибке (см. Рисунок II-80);

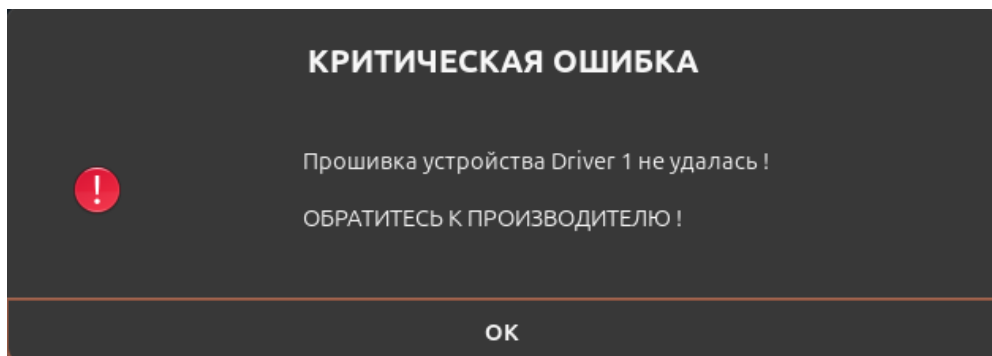


Рисунок II-80 – Уведомление о критической ошибке

- 8) нажмите «ОК». Отобразится лог ошибки (см. Рисунок II-81);

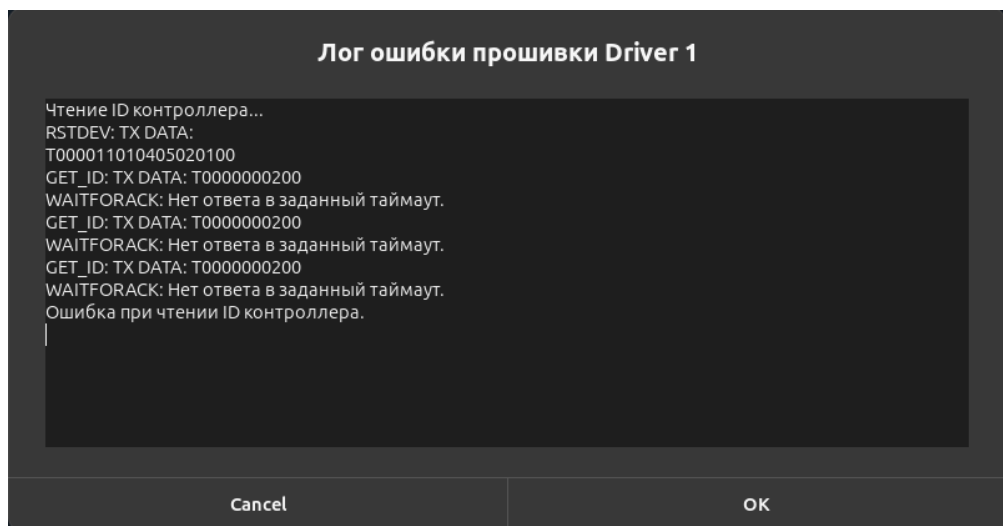


Рисунок II-81 – Лог ошибки прошивки

- 9) сообщите информацию о логе ошибки при обращении в сервисный центр предприятия-изготовителя.

На данный момент одновременное обновление версий ПО, прошивки и веб-интерфейса недоступно. Поочередное обновление гарантирует стабильную работу приложения.

В некоторых случаях может потребоваться перезагрузка устройства после обновления, при этом отобразится уведомление о требовании перезагрузки.

После обновления в интерфейсе мониторинга отобразятся версии, соответствующие последнему релизу.

III. PROMOBOT M CONTROL SDK

1 Назначение и область применения

Promobot M Control SDK предоставляет программный интерфейс для управления роботами-манипуляторами, работающими под управлением Promobot M Control, в частности промышленным роботом-манипулятором Promobot M13. SDK предназначен для автоматизации задач и интеграции манипулятора с внешними системами. Поддерживаются синхронные операции, coroutine-вызовы, команды без ожидания результата и потоковое управление.

Основные возможности SDK:

- а) подключение манипулятора;
- б) перемещение и управление движением:
 - задание целевых координат в декартовом пространстве или целевых углов каждого узла для точного позиционирования манипулятора;
 - управление параметрами движения с помощью коэффициентов скорости и ускорения;
 - реализация сложных траекторий, включая движение по дуге и последовательные перемещения по точкам;
 - отслеживание положения манипулятора и его узлов в режиме реального времени;
 - управление состоянием манипулятора (активный режим, режим ожидания, остановка и т.д.);
 - смена режимов управления сервоприводами;
- в) работа с инструментом и сменными модулями:
 - управление питанием инструмента манипулятора;
 - управление гриппером и вакуумным насосом при наличии совместимого сменного модуля;
 - настройка питания и цифровых выходов фланца инструмента;
 - воспроизведение аудиофайлов при наличии и настройке аудиоподсистемы;
- г) взаимодействие с внешними устройствами – работа с датчиками и исполнительными устройствами через GPIO, I2C, Modbus RTU и RS-485 в пределах возможностей конкретной комплектации Promobot M13;

- д) автоматизация:
 - выполнения программ по указанному имени;
 - выполнение Python-скриптов;
 - выполнение программ с передачей JSON-данных;
 - тайм-ауты операций.
- е) диагностика – обработка сбоев и ошибок манипулятора.

2 Архитектура и принципы работы SDK

2.1 Компоненты

Ключевые компоненты (см. Рисунок III-1):

- клиент: компьютер пользователя / приложение с SDK;
- контроллер манипулятора – принимает команды SDK, проверяет их и передает исполнительным подсистемам;
- Promobot M Control – приложение настройки, программирования и диагностики манипулятора.

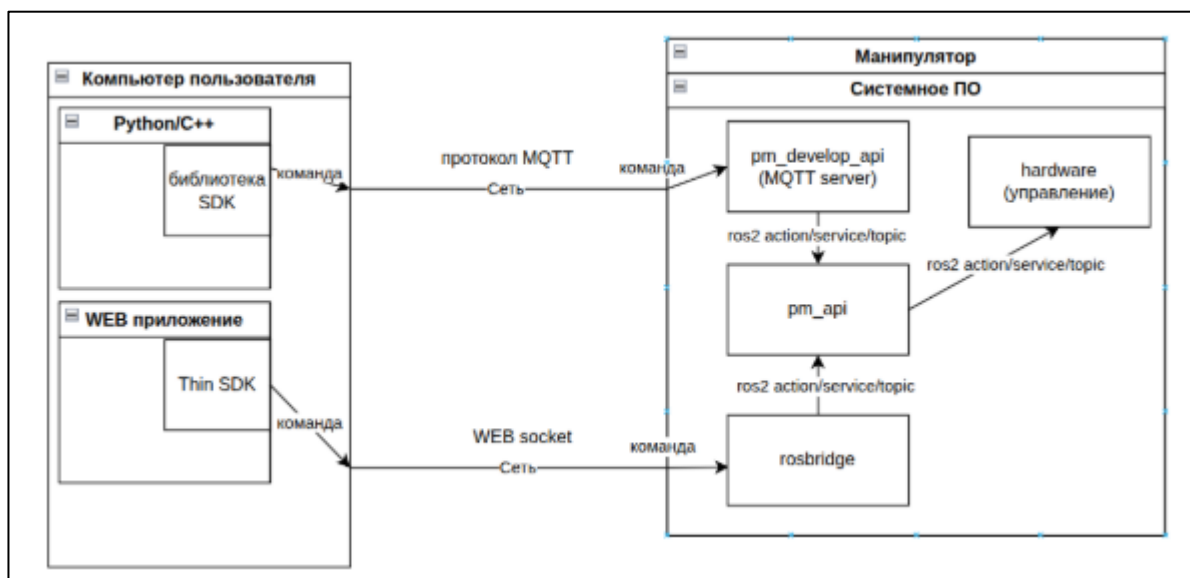


Рисунок III-1 – Компоненты

Взаимодействие построено по модели клиент – сервер с использованием протокола MQTT.

SDK работает по модели клиент – сервер. Клиентское приложение передает контроллеру команды управления и получает результаты выполнения, телеметрию и диагностические сообщения. Внутренняя программная архитектура контроллера не является

частью пользовательского API и может изменяться между версиями программного обеспечения.

2.2 Модель работы

Модель работы включает в себя следующие этапы:

- а) установка и настройка SDK на стороне клиента;
- б) подключение к контроллеру манипулятора Promobot M Control по IP-адресу и параметрам авторизации;
- в) захват управления манипулятором (`get_control`) и резервирование канала управления движением;
- г) выполнение команд движения, управления периферией или запуск программ/сценариев;
- д) при необходимости – получение состояния манипулятора и диагностических данных;
- е) освобождение управления и завершение работы приложения.

3 SDK Python

3.1 Требования и подготовка среды Python

SDK Python предназначен для разработки внешних приложений, взаимодействующих с контроллером Promobot M13 по сети. В пользовательский интерфейс входят подключение, получение управления, движение, чтение состояния, выбор сохраненного TCP, работа с вводом-выводом, запуск программ, события и поддерживаемые дополнительные интерфейсы.

Рекомендуемый порядок разработки приложения:

- а) установить Python и SDK в отдельное виртуальное окружение;
- б) проверить сетевую доступность контроллера и TCP-порта 1883 в доверенной сети;
- в) выполнить первое подключение и прочитать координаты без движения;
- г) получить управление методом `get_control()`;
- д) проверить TCP, полезную нагрузку, ограничения и состояние оборудования;
- е) использовать синхронные методы для последовательных сценариев либо явно указанные `coroutine`-методы для `asyncio`;
- ж) обрабатывать тайм-ауты и ошибки SDK;
- з) в блоке `finally` снять подписки и вызвать `disconnect()`.

Параметры SDK Python описаны в таблице 12.

Таблица 12 – Параметры SDK Python

Параметр	Значение
Применяемая версия SDK	0.6.13
Поддерживаемая версия Python	3.12.x
Основной класс	sdk.manipulators.m13.M13
Транспорт	MQTT по TCP; порт 1883 по умолчанию
Назначение	Подключение, управление движением, чтение состояния, ввод-вывод, запуск программ и работа с дополнительными интерфейсами Promobot M13

Примечание – Раздел описывает поддерживаемый пользовательский интерфейс SDK 0.6.13. Используйте SDK из комплекта поставки или из утвержденного репозитория пакетов, соответствующий версии Promobot M Control на контроллере. Команды движения разрешается выполнять только после проверки рабочей зоны, TCP, инструмента, полезной нагрузки, ограничений, состояния оборудования и доступности аварийного останова.

Требования к компьютеру:

- операционная система Windows или Linux с Python 3.12.x;
- сетевой интерфейс, имеющий маршрут до контроллера Promobot M13;
- доступ к утвержденному дистрибутиву SDK или репозиторию пакетов;
- учетные данные MQTT, предоставленные для конкретной установки;
- уникальный client_id для каждого одновременно подключённого приложения.
- Порт 1883 должен быть доступен только внутри доверенной производственной сети или через утвержденный защищенный канал. Не публикуйте MQTT-порт контроллера напрямую в Интернет и не храните учетные данные в исходном коде приложения.

Подготовка манипулятора:

- манипулятор установлен и введен в эксплуатацию в соответствии с руководством по эксплуатации.
- рабочая зона освобождена от людей и посторонних предметов.
- аварийный останов доступен оператору и проверен установленным порядком.
- в Promobot M Control настроены актуальные TCP, инструмент и полезная нагрузка.
- проверены программные и механические ограничения движения.

- для первых испытаний используются минимальные практически допустимые коэффициенты скорости и ускорения.

Примечание – Метод `stop_movement()` является программной командой. Он не заменяет аварийный останов, STO и иные аппаратные функции безопасности.

3.2 Получение и установка SDK

3.2.1 Проверка версии Python

Для SDK рекомендуется Python 3.12.x. Перед созданием виртуального окружения проверьте установленную версию. Если Python отсутствует, установите Python 3.12 с официального сайта python.org для Windows либо из репозитория дистрибутива для Linux.

```
py -3.12 --version  
python3.12 --version
```

3.2.2 Создание виртуального окружения в Windows PowerShell

Команды ниже создают изолированное виртуальное окружение `.venv`, активируют его, обновляют `pip` и устанавливают зависимости и пакет SDK из репозитория, указанного в актуальной документации Promobot. Выполняйте команды в каталоге вашего проекта; после активации в приглашении PowerShell обычно отображается префикс `(.venv)`.

```
py -3.12 --version  
py -3.12 -m venv .venv  
.\.venv\Scripts\Activate.ps1  
python -m pip install --upgrade pip  
python -m pip install --index-url https://pypi.org/simple pydantic==2.11.7 paho-mqtt==2.1.0  
--extra-index-url https://test.pypi.org/simple/ pm-python-sdk
```

3.2.3 Создание виртуального окружения в Linux

Команды ниже создают и активируют отдельное окружение Python, после чего устанавливают зависимости и пакет SDK. В Ubuntu/Debian при отсутствии Python установите пакеты `python3.12` и `python3.12-venv`. При открытии нового терминала виртуальное окружение необходимо активировать повторно.

```
python3.12 --version
```

```
python3.12 -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip
python -m pip install --index-url https://pypi.org/simple pydantic==2.11.7 paho-mqtt==2.1.0
--extra-index-url https://test.pypi.org/simple/ pm-python-sdk
```

Примечание – Команда `--break-system-packages` не требуется и не должна использоваться внутри виртуального окружения. Если репозиторий пакетов недоступен, установите wheel-файл SDK, предоставленный изготовителем.

3.2.4 Установка из локального дистрибутива

Если SDK поставлен в виде wheel-файла, поместите файл в каталог проекта, активируйте виртуальное окружение и выполните:

```
python -m pip install pm_python_sdk-<version>-py3-none-any.whl
```

3.2.5 Проверка установки

Проверка подтверждает, что пакет установлен в активное виртуальное окружение, основной класс Promobot M13 доступен для импорта, а версия установленного пакета определена. Подключение к манипулятору при этом не выполняется.

```
from importlib.metadata import version
from sdk.manipulators.m13 import M13
print("Импорт SDK выполнен, версия:", version("pm-python-sdk"))
```

Ожидаемый результат:

Импорт SDK выполнен, версия: <установленная версия>

Если установленная версия отличается от версии, указанной в таблице параметров SDK, используйте документацию соответствующего выпуска либо установите пакет из комплекта поставки контроллера.

3.3 Настройка сети и первое подключение

3.3.1 Параметры подключения

Параметры сети и учетные данные должны соответствовать настройкам конкретного контроллера, а `client_id` – быть уникальным для каждого активного клиентского процесса. Аргументы конструктора M13 приведены в следующей таблице 13.

Таблица 13 – Параметры сети и учетные данные

Параметр	Назначение	Требование
host	IP-адрес или сетевое имя контроллера	Должен быть доступен с компьютера пользователя. localhost применяется только при запуске программы на самом контроллере
client_id	Идентификатор MQTT-клиента	Должен быть уникальным среди одновременно активных клиентов
login	Имя пользователя MQTT	Использовать данные конкретной установки
password	Пароль MQTT	Не хранить в открытом репозитории; в рабочем приложении читать из защищённой конфигурации

3.3.2 Проверка доступности контроллера

Перед запуском SDK проверьте IP-связность и доступность MQTT-порта. Команды Windows и Linux приведены отдельно.

```
ping 192.168.0.10
```

```
Test-NetConnection 192.168.0.10 -Port 1883
```

```
ping -c 4 192.168.0.10
```

```
nc -vz 192.168.0.10 1883
```

3.3.3 Первый запуск без движения

Первый тест должен только подключиться к контроллеру и получить телеметрию. Получение управления и команды движения в этом примере не используются.

```
import json
```

```

from sdk.errors import CommandTimeout, ConnectionError as SdkConnectionError,
SdkError

from sdk.manipulators.m13 import M13

ROBOT_HOST = "192.168.0.10"
CLIENT_ID = "m13-sdk-check-01"
LOGIN = "<login>"
PASSWORD = "<password>"

robot = M13(ROBOT_HOST, CLIENT_ID, LOGIN, PASSWORD)

try:
    robot.connect()

    cartesian = json.loads(robot.get_cartesian_coordinates(timeout_seconds=10.0))
    joints = json.loads(robot.get_joint_state(timeout_seconds=10.0))

    print("Декартовы координаты:", cartesian)
    print("Суставы:", joints)

except SdkConnectionError as exc:
    print(f"Ошибка подключения: {exc}")
except CommandTimeout as exc:
    print(f"Тайм-аут команды: {exc}")
except SdkError as exc:
    print(f"Ошибка SDK: {exc}")
finally:
    robot.disconnect()

```

Примечание – Методы `get_cartesian_coordinates()` и `get_joint_state()` возвращают JSON-строки. Для доступа к полям строки преобразуются функцией `json.loads()`. Состав полей зависит от версии контроллера; приложение должно проверять наличие требуемых ключей.

3.4 Модель выполнения команд

В SDK применяются пять моделей вызова. Суффикс `_async` сам по себе не гарантирует, что метод является coroutine и может быть передан оператору `await`. Используйте только варианты, явно приведенные в пользовательском справочнике. Описание моделей представлено в таблице 14.

Таблица 14 – Описание моделей вызова

Модель	Как распознать	Поведение	Рекомендация
Синхронная	<code>method(...)</code>	Отправляет команду и блокирует текущий поток до результата или тайм-аута.	Основной вариант для линейных сценариев.
Coroutine	<code>async def</code> ; обычно <code>*_async_await</code> или отдельные методы <code>*_async</code>	Вызывается через <code>await</code> и не блокирует цикл <code>asyncio</code> .	Использовать только для методов, явно отмеченных как coroutine.
Объект команды	Часть методов <code>*_async</code> возвращает <code>Command</code> или <code>Promise</code>	Создаёт низкоуровневый объект. Оператор <code>await</code> к такому методу неприменим.	Для прикладного кода предпочтителен синхронный или coroutine-вариант.
Без ожидания	<code>*_no_wait(...)</code>	Отправляет запрос без ожидания результата выполнения.	Применять, когда приложение самостоятельно контролирует состояние и ошибки.
Потоковая	<code>stream_*</code>	Передаёт целевые значения без подтверждения каждой точки.	Требует предварительного выбора серво-режима и постоянной передачи данных.

3.4.1 Рекомендуемые пары методов

Для выбора варианта одного действия для последовательной программы, приложения на `asyncio` или отправки без ожидания результата воспользуйтесь данными, приведенными в таблице 15. Не все операции имеют полный набор вариантов, поэтому используйте только фактически указанные пары.

Таблица 15 – Действия для последовательной программы

Операция	Синхронный метод	Coroutine для await	Без ожидания
Получить управление	get_control()	get_control_async_await()	get_control_no_wait()
Движение по координатам	move_to_coordinates()	move_to_coordinates_async_await()	move_to_coordinates_no_wait()
Движение по суставам Promobot M13	move_to_angles()	move_to_angles_async()	move_to_angles_no_wait()
Останов движения	stop_movement()	stop_movement_async_await()	stop_movement_no_wait()
Запуск программы по имени	run_program_by_name()	run_program_by_name_async_await()	run_program_by_name_no_wait()
Запуск Python-кода	run_python_program()	run_python_program_async_await()	run_python_program_no_wait()

Внимание! Метод `move_to_angles_async()` является coroutine. В отличие от него `move_to_coordinates_async()` возвращает объект команды и не должен вызываться через `await`. Для координатного движения применяйте `move_to_coordinates_async_await()`.

3.4.2 Общие параметры

Параметры, определяющие время ожидания, способ обработки ошибки и режим выполнения и повторяющиеся во многих командах SDK описаны в таблице 16. Перед изменением значений учитывайте длительность реальной операции и требования приложения к диагностике.

Таблица 16 – Параметры, определяющие время ожидания, способ обработки ошибки и режим выполнения

Параметр	Назначение
timeout_seconds	Максимальное время ожидания результата команды. Значение по умолчанию для большинства методов — 60 с.
throw_error	При True ошибка или тайм-аут передаются приложению исключением. Для прикладных программ рекомендуется сохранять True.
velocity_scaling_factor velocity_factor	/ Коэффициент скорости. Для первых запусков используйте минимальное валидированное значение.

Параметр	Назначение
acceleration_scaling_factor / acceleration_factor	Коэффициент ускорения. Должен соответствовать настройкам и ограничениям конкретной установки.
enable_feedback	Включает обработку промежуточных статусов для методов, которые поддерживают обратную связь.

SDK 0.6.13 не декларирует потокобезопасность параллельных команд. В рамках одного объекта M13 сериализуйте команды движения и не сочетайте одновременно обычное, `no_wait`- и потоковое управление.

Внимание! Истечение `timeout_seconds` прекращает ожидание результата клиентским приложением, но само по себе не подтверждает физическую остановку движения. После тайм-аута запросите состояние и текущие ошибки; дальнейшие действия выполняйте по утвержденной процедуре, при необходимости используя программный или аппаратный останов.

3.5 Единицы измерения и структуры данных

Единицы измерения и форматы, применяемые в пользовательском интерфейсе SDK указаны в таблице 17. Все исходные данные следует привести к указанным единицам до формирования объектов команд и отправки их контроллеру.

Таблица 17 – Единицы измерения и форматы, применяемые в пользовательском интерфейсе SDK

Величина	Единица/формат
Декартовы координаты x, y, z	метры
Углы суставов	радианы
Скорости суставов	радианы в секунду
Ориентация — кватернион	x, y, z, w; единичный кватернион
Ориентация — углы Эйлера	gx, gy, gz; радианы
GPIO-слово	целое значение или строковая маска в зависимости от метода

Примечание – Для декартовых движений применяются метры, радианы и единичный кватернион. Создание или редактирование TCP через SDK 0.6.13 в данном разделе не описывается, поскольку единицы полей Angles не зафиксированы публичным контрактом. Настраивайте TCP в Promobot M Control и используйте SDK для чтения и выбора сохраненного TCP.

3.5.1 Основные классы данных

Основные классы, из которых формируются параметры движения, TCP и интерфейсов фланца описаны в таблице 18. Импортируйте класс из указанного модуля и не заменяйте его одноименным классом из другого файла SDK.

Таблица 18 – Основные классы данных

Класс	Импорт	Назначение
MoveCoordinatesParams Position	sdk.commands.move_coordinate s_command	Позиция x, y, z для координатного движения
MoveCoordinatesParams Orientation	sdk.commands.move_coordinate s_command	Кватернион или углы Эйлера для координатного движения
PlannerType	sdk.commands.move_coordinate s_command	Тип траектории PTP или LIN
Position	sdk.commands.data	Позиция для TCP и расширенных команд
TCPInfo	sdk.commands.data	Описание сохраненного TCP
ManipulatorState	sdk.utils.enums	Состояние жизненного цикла манипулятора
ServoControlType	sdk.utils.enums	Режим потокового управления

3.6 Получение управления и первый сценарий

Метод connect() устанавливает транспортное соединение. Для команд, изменяющих состояние оборудования, после подключения необходимо запросить управление методом get_control().

```
from sdk.manipulators.m13 import M13
```

```
ROBOT_HOST = "192.168.0.10"
```

```
CLIENT_ID = "m13-sdk-control-01"
```

```
LOGIN = "<login>"
```

```
PASSWORD = "<password>"
```

```
robot = M13(ROBOT_HOST, CLIENT_ID, LOGIN, PASSWORD)
```

```
try:
```

```
    robot.connect()
```

```
    robot.get_control(timeout_seconds=10.0)
```

```
    # Целевая точка должна быть валидирована для конкретной установки.
```

```
    # Не снимайте комментарий до проверки рабочей зоны, TCP и полезной нагрузки.
```

```
    # robot.move_to_angles(
```

```
    #     sp1, sp2, sp3, sp4, sp5, sp6,
```

```
    #     velocity_factor=0.05,
```

```
    #     acceleration_factor=0.05,
```

```
    #     planner_id="PTP",
```

```
    # )
```

```
finally:
```

```
    robot.disconnect()
```

Внимание! В примере команда движения закомментирована намеренно. Значения суставов нельзя считать универсальной безопасной точкой: они должны быть получены из утвержденной конфигурации конкретного рабочего места.

3.7 Движение манипулятора

3.7.1 Движение по суставам (move_to_angles)

Суставное движение задает целевые углы для всех шести приводов Promobot M13 и выполняется с выбранными коэффициентами скорости и ускорения. Перед использованием замените значения на проверенную конфигурацию, получите управление манипулятором и убедитесь, что траектория не нарушает ограничения рабочей зоны.

```
move_to_angles(sp1, sp2, sp3, sp4, sp5, sp6, sp1_v=0.0, ..., sp6_v=0.0, velocity_factor=0.1,
acceleration_factor=0.1, planner_id="PTP", timeout_seconds=60.0, throw_error=True) -> None
```

Перемещает шесть суставов Promobot M13 в заданные угловые положения. Суставы передаются в порядке `shoulder_pan_joint`, `shoulder_lift_joint`, `elbow_joint`, `wrist_1_joint`, `wrist_2_joint`, `wrist_3_joint`.

Параметры команды описаны в таблице 19.

Таблица 19 – Параметры команды движения по суставам

Параметр	Описание
Предусловия	Установлено соединение, получено управление, проверены ограничения и траектория
Возвращаемое значение	None после подтвержденного завершения команды
Особенности	<code>sp1...sp6</code> задаются в радианах; <code>sp1_v...sp6_v</code> — в радианах в секунду. <code>planner_id</code> в текущем M13 применяется как строка, обычно «РТР»

Асинхронный вариант для `asyncio`:

```
async def move_to_validated_joint_target(
    robot: M13,
    target: tuple[float, float, float, float, float, float],
) -> None:
    await robot.move_to_angles_async(
    )
    acceleration_factor=0.05,
    velocity_factor=0.05,
    *target,
```

3.7.2 Движение по декартовым координатам

Декартово движение задает целевое положение и ориентацию ТСП относительно рабочей системы координат. В примере явно выбирается планировщик LIN или РТР; координаты, ориентация, ТСП и полезная нагрузка должны быть проверены до отправки команды.

```
from sdk.commands.move_coordinates_command import (
    MoveCoordinatesParamsOrientation,
    MoveCoordinatesParamsPosition,
    PlannerType,
```

```

)
from sdk.manipulators.m13 import M13

robot = M13("192.168.0.10", "m13-cartesian-01", "<login>", "<password>")

position = MoveCoordinatesParamsPosition(x=0.40, y=0.00, z=0.35)
orientation = MoveCoordinatesParamsOrientation.from_quaternion(
    x=0.0, y=0.0, z=0.0, w=1.0
)

try:
    robot.connect()
    robot.get_control(timeout_seconds=10.0)

    # Используйте только после проверки точки и всей траектории.
    # robot.move_to_coordinates(
    #     position=position,
    #     orientation=orientation,
    #     velocity_scaling_factor=0.05,
    #     acceleration_scaling_factor=0.05,
    #     planner_type=PlannerType.LIN,
    #     timeout_seconds=30.0,
    # )
finally:
    robot.disconnect()

```

Параметры команды описаны в таблице 20.

Таблица 20 – Параметры команды движения по декартовым координатам

Планировщик	Назначение
PlannerType.LIN	Линейное перемещение ТСП в декартовом пространстве
PlannerType.PTP	Перемещение к целевой позе по траектории, рассчитываемой планировщиком суставов

Примечание – Фактическое значение `planner_type` по умолчанию в SDK 0.6.13 – `PlannerType.LIN`. В производственном коде рекомендуется всегда передавать тип планировщика явно.

3.7.3 Асинхронное координатное движение

Асинхронный вариант применяется в приложениях на `asyncio`, когда во время ожидания завершения движения должны обрабатываться другие задачи. Для ожидания результата используйте именно `move_to_coordinates_async_await()`; метод `move_to_coordinates_async()` возвращает объект команды и не является `coroutine`.

```
import asyncio

from sdk.commands.move_coordinates_command import (
    MoveCoordinatesParamsOrientation,
    MoveCoordinatesParamsPosition,
    PlannerType,
)
from sdk.manipulators.m13 import M13

async def main() -> None:
    robot = M13("192.168.0.10", "m13-async-01", "<login>", "<password>")
    try:
        await robot.connect_async()
        await robot.get_control_async_await(timeout_seconds=10.0)

        position = MoveCoordinatesParamsPosition(0.40, 0.00, 0.35)
        orientation = MoveCoordinatesParamsOrientation.from_quaternion(0.0, 0.0, 0.0, 1.0)

        # Снимайте комментарий только для проверенных точки и траектории.
        # await robot.move_to_coordinates_async_await(
        #     position,
        #     orientation,
        #     velocity_scaling_factor=0.05,
```

```

# acceleration_scaling_factor=0.05,
# planner_type=PlannerType.LIN,
# )
finally:
    robot.disconnect()

```

```

if __name__ == "__main__":
    asyncio.run(main())

```

3.7.4 Останов движения

Доступные варианты программной команды останова и их модель ожидания результата сравниваются в таблице 21. Эти методы предназначены для управления программой и не заменяют аварийный останов, STO и другие аппаратные функции безопасности.

Таблица 21 – Варианты программной команды останова и их модель ожидания результата

Метод	Назначение
stop_movement()	Отправляет программную команду останова и ожидает ответ.
await stop_movement_async_await()	Coroutine-вариант программного останова.
stop_movement_no_wait()	Отправляет команду без ожидания результата.

3.7.5 Расширенные движения

Специализированные способы движения, которые не требуются для первого запуска представлены в таблице 22. Перед применением необходимо проверить геометрию траектории, совместимость используемых классов данных и ограничения конкретной версии контроллера.

Таблица 22 – Специализированные способы движения

Семейство	Назначение	Примечание
arc_motion*	Движение по дуге через целевую и центральную позы	Для моделей Pose импортировать классы из sdk.commands.arc_motion. Перед применением проверить геометрию дуги
paletizing_movement*	Паллетизирующее движение с дискретизацией траектории	Имя метода в SDK 0.6.13 содержит написание paletizing. Использовать синхронный paletizing_movement()
move_group*	Выполнение группы точек в декартовом или суставном представлении	Расширенный интерфейс; все допуски и радиус сглаживания задавать явно

3.8 Состояние и диагностика

Методы в таблице 23 позволяют получить текущую позу, состояние суставов, домашнее положение, ошибки и состояние манипулятора. Чтение состояния рекомендуется выполнять перед движением и после тайм-аута или отказа команды.

Таблица 23 – Методы чтения текущей позы, углов суставов, состояния, ошибок и домашнего положения

Метод	Результат	Применение
get_cartesian_coordinates()	JSON-строка	Текущая декартова поза
get_joint_state()	JSON-строка	Текущие положения и состояние суставов
get_home_position()	dict	Домашнее положение, полученное от контроллера
get_current_errors()	данные ошибки или None	Текущие диагностические сообщения
get_cobot_state()	идентификатор/значение состояния или None	Текущее состояние манипулятора
tcp_get_current()	TCPInfo	Активная система координат инструмента

```
import json
```

```
joints = json.loads(robot.get_joint_state())
```

```
coordinates = json.loads(robot.get_cartesian_coordinates())
```

```

errors = robot.get_current_errors()
state = robot.get_cobot_state()

print(joints)
print(coordinates)
print(errors)
print(state)

```

Примечание – В пользовательском API SDK 0.6.13 для состояния манипулятора поддерживается синхронный метод `get_cobot_state()`. Метод `get_cobot_state_async_await()` в руководство не включен. В приложении `asyncio` синхронный вызов можно выполнить через `asyncio.to_thread()`.

3.9 TCP и инструмент

TCP определяет рабочую точку инструмента относительно фланца. Создавайте и редактируйте TCP в Promobot M Control, где доступны проверка геометрии и единиц. Через SDK рекомендуется читать список TCP, получать активный TCP и выбирать ранее сохраненную конфигурацию.

```

from sdk.manipulators.m13 import M13

robot = M13("192.168.0.10", "m13-tcp-01", "<login>", "<password>")

try:
    robot.connect()
    robot.get_control()

    tcps, current_name = robot.tcp_get_list()
    print("Текущий TCP:", current_name)
    for tcp in tcps:
        print(tcp)

    # Выбирайте только ранее проверенный TCP.
    # robot.tcp_apply("tool_1")
finally:
    robot.disconnect()

```

Методы TCP описаны в таблице 24.

Таблица 24 – Методы TCP

Метод	Назначение
tcp_apply()	Выбрать сохранённый TCP
tcp_delete()	Удалить TCP. При удалении активного TCP требуется определить reset_current/apply_other
tcp_get_current()	Получить активный TCP как TCPInfo
tcp_get_list()	Получить список TCP и имя последнего примененного TCP

Примечание – Применяйте tcp_apply() только к TCP, созданному и проверенному в Promobot M Control. Удаление активного TCP выполняйте только при наличии утвержденной процедуры выбора, замещающего TCP.

3.10 GPIO, выходы и интерфейсы фланца

3.10.1 Работа с именованным GPIO-ресурсом

Именованный GPIO-ресурс позволяет читать или изменять конкретный сигнал контроллера по его системному имени. Перед записью уточните актуальное имя ресурса, направление сигнала и допустимый формат значения для используемой версии Promobot M Control.

```
from sdk.manipulators.m13 import M13

robot = M13("192.168.0.10", "m13-gpio-01", "<login>", "<password>")
GPIO_RESOURCE = "safety/digital_ports/digital_outputs/value"

try:
    robot.connect()
    robot.get_control()

    # Запись включайте только для проверенного ресурса и подключенной нагрузки.
    # robot.write_gpio(GPIO_RESOURCE, 1)
    value = robot.get_gpio_value(GPIO_RESOURCE)
    print("Значение GPIO:", value)
```

finally:

```
robot.disconnect()
```

Внимание! Имена ресурсов зависят от версии контроллера и конфигурации Promobot M Control. Перед записью получите утвержденный перечень ресурсов конкретного изделия. Не переносите имена из примера на другую конфигурацию без проверки.

3.10.2 Побитовая и масочная работа

Различие между операциями с отдельным битом и записью или чтением полной маски GPIO отображено в таблице 25. Номер бита, разрядность маски и допустимые значения должны соответствовать конкретному именованному ресурсу контроллера.

Таблица 25 – Различие между операциями с отдельным битом и записью или чтением полной маски GPIO

Метод	Параметр значения	Результат
write_gpio_bit(name, value)	0 или 1	Записывает отдельное значение
get_gpio_bit(name)	—	Возвращает значение или None
set_gpio_on_mask(name, value_mask)	Строковая маска в формате контроллера	Записывает маску
get_gpio_mask(name)	—	Возвращает mask_value как str или None

3.10.3 Аналоговые, цифровые и I2C-выходы

Низкоуровневые способы управления выходами и интерфейсом I2C перечислены в таблице 26. Электрические диапазоны, адреса каналов и допустимые нагрузки определяются аппаратной конфигурацией M13 и подключенного оборудования.

Таблица 26 – Низкоуровневые способы управления выходами и интерфейсом I2C

Метод	Назначение	Ограничение
write_analog_output(channel, value)	Запись значения аналогового выхода.	Канал, единицы и диапазон должны соответствовать разделу РЭ по электрическим интерфейсам конкретной комплектации. Не используйте метод без подтвержденной карты каналов.

Метод	Назначение	Ограничение
write_digital_output(channel, value)	Запись True/False в цифровой выход.	Номер канала должен быть сопоставлен с физическим выходом DO в документации конкретной версии контроллера.
write_i2c(name, value)	Передача целого значения в именованный I2C-ресурс.	Имя ресурса, формат и адресация определяются конфигурацией контроллера и подключенным устройством.

3.10.4 Фланец

Интерфейсы фланца используются для настройки цифровых выводов и питания подключённого инструмента. Перед выполнением примера проверьте назначение контактов, тип выхода, допустимое напряжение и электрические характеристики внешнего устройства.

```

from sdk.commands.data import FlangePinName, OutputType, VPWRType
from sdk.manipulators.m13 import M13

robot = M13("192.168.0.10", "m13-flange-01", "<login>", "<password>")

try:
    robot.connect()
    robot.get_control()

    # Изменение режима и питания выполняйте только для проверенной схемы.
    # robot.configure_flange_dout(
    #     pin_name=FlangePinName.GADGET5,
    #     output_type=OutputType.PushPull,
    # )
    # robot.configure_flange_vpwr(VPWRType.V24)

    result, output_type, voltage_level, value = robot.get_flange_dout(
        FlangePinName.GADGET5
    )
    print(result, output_type, voltage_level, value)

```

```
finally:
    robot.disconnect()
```

Значения для настройки цифровых выводов описаны в таблице 27.

Таблица 27 – Значения для настройки цифровых выводов

Перечисление	Допустимые значения
FlangePinName	GADGET5, GADGET6
OutputType	OpenSource, OpenDrain, PushPull
VoltageLevel	V12, V24, NONE
VPWRType	GND, V12, V24
BoardName	SAFETY, GADGET, FLANGE

Внимание! Изменение режима выхода или напряжения фланца допускается только при совместимости с подключенным оборудованием. Ошибочная конфигурация может повредить внешнее устройство.

3.11 Запуск программ

3.11.1 Запуск сохраненной программы

Метод запускает программу, заранее сохраненную и проверенную в Promobot M Control. В коде необходимо указать точное имя программы, существующее на контроллере; перед запуском приложение устанавливает соединение и получает управление.

```
from sdk.manipulators.m13 import M13

robot = M13("192.168.0.10", "m13-program-01", "<login>", "<password>")
PROGRAM_NAME = "m13/<program_name>"

try:
    robot.connect()
    robot.get_control()
    robot.run_program_by_name(PROGRAM_NAME, timeout_seconds=300.0)
finally:
    robot.disconnect()
```

Имя должно соответствовать программе, существующей в Promobot M Control. Строка m13/<program_name> в примере обозначает формат и должна быть заменена на фактическое имя программы.

3.11.2 Запуск JSON-программы

Метод `run_program_json(name, program_json)` принимает словарь Python. Структура JSON должна быть сформирована или экспортирована совместимой версией Promobot M Control. Руководство не задает универсальную схему JSON, поскольку она является частью формата программ контроллера.

```
import json
from pathlib import Path

program = json.loads(Path("program.json").read_text(encoding="utf-8"))
robot.run_program_json("program_1", program, timeout_seconds=300.0)
```

3.11.3 Запуск Python-кода на контроллере

Метод передает текст Python-программы контроллеру и ожидает завершения выполнения. Используйте только доверенный код из контролируемого репозитория. Не передавайте пользовательский ввод напрямую, не устанавливайте непроверенные зависимости и заранее задавайте достаточный `timeout_seconds`. Доступность пакетов и ограничения выполнения определяются конфигурацией контроллера.

```
from sdk.manipulators.m13 import M13

robot = M13("192.168.0.10", "m13-python-program-01", "<login>", "<password>")

try:
    robot.connect()
    robot.get_control()
    robot.run_python_program(
        python_code="print('Hello from M13!')",
        python_version="3.12",
        requirements=[],
```

```
)  
finally:  
    robot.disconnect()
```

Внимание! `run_python_program()` выполняет код на контроллере. Применяйте метод только при наличии разрешения, утвержденного перечня зависимостей, журналирования и процедуры восстановления контроллера.

3.12 Потокое управление

Потоковые методы применяются для последовательной передачи целевых положений или скоростей. Они не ожидают подтверждения каждой точки и требуют непрерывного цикла управления, контроля связи и заранее определенной реакции на прекращение потока. Подтверждение команды выбора серво-режима означает отправку команды контроллеру; перед началом потока приложение должно проверить фактическую реакцию системы в испытанном сценарии:

- а) Подключиться и получить управление.
- б) Выбрать режим `JOINT_JOG`, `TWIST` или `POSE`.
- в) Включить потоковый режим методом `enable_servo_streaming(True)`.
- г) Передавать точки с постоянным периодом, установленным проектом.
- д) В блоке `finally` выключить потоковый режим и разорвать соединение.

```
import time
```

```
from sdk.manipulators.m13 import M13
```

```
robot = M13("192.168.0.10", "m13-stream-01", "<login>", "<password>")
```

```
try:
```

```
    robot.connect()
```

```
    robot.get_control()
```

```
    robot.set_servo_joint_jog_mode()
```

```
    robot.enable_servo_streaming(True)
```

```
# Список должен быть сформирован приложением из валидированных точек.
```

```
validated_points: list[tuple[float, float, float, float, float, float]] = []
```

```

for point in validated_points:
    robot.stream_joint_angles(*point)
    time.sleep(0.1)
finally:
    robot.enable_servo_streaming(False)
    robot.disconnect()

```

Потоковые методы описаны в таблице 28.

Таблица 28 – Потоковые методы

Метод	Передаваемые данные
stream_joint_angles()	Шесть углов Promobot M13 и шесть скоростей
stream_joint_positions()	Словари positions и velocities с именами суставов
stream_cartesian_velocities()	Словари линейной и угловой скорости
stream_coordinates()	Позиция и ориентация TCP

Примечание – Частота 10 Гц в примере демонстрирует только механизм вызова и не является требованием системы. Рабочий период, допустимый пропуск пакетов и поведение при прекращении потока должны быть определены и испытаны в проекте конкретного приложения.

3.13 Состояния и программный останов

Методы ниже предназначены для чтения и изменения состояния жизненного цикла и для программного останова. Используйте change_state() только в переходах, разрешенных контроллером и эксплуатационной процедурой. stop_movement() не является функцией функциональной безопасности и не заменяет аварийный останов или STO.

Методы управления состоянием и программным остановом описаны в таблице 29.

Таблица 29 – Методы управления состоянием и программным остановом

Метод	Назначение	Уровень применения
change_state(state)	Изменить состояние через ManipulatorState.	Только в разрешенных переходах жизненного цикла.
get_cobot_state()	Получить текущее состояние	Пользовательский метод

Метод	Назначение	Уровень применения
stop_movement()	Программно остановить текущее движение	Не заменяет safety-функции

Перечисление ManipulatorState содержит идентификаторы состояний жизненного цикла SDK. Оно не задает граф допустимых переходов: перед вызовом change_state() проверьте текущее состояние, текущие ошибки и разрешенный переход для версии Promobot M Control, установленной на контроллере. Идентификаторы состояний ManipulatorState перечислены в таблице 30.

Таблица 30 – Состояния ManipulatorState

ManipulatorState	Значение
UNKNOWN	Неизвестное состояние
UNCONFIGURED	Не настроен
INACTIVE	Неактивен
ACTIVE	Активен
FINALIZED	Завершен
PAUSED	Пауза
CONFIGURING	Настройка
CLEANING_UP	Очистка
SHUTTING_DOWN	Выключение
ACTIVATING	Активация
DEACTIVATING	Деактивация
ERROR_PROCESSING	Обработка ошибки
PAUSING	Переход в паузу
RESUMING	Возобновление

Примечание – Перечисление отражает идентификаторы SDK, но не задает разрешенный граф переходов. Допустимые переходы должны соответствовать логике текущей версии контроллера и эксплуатационной процедуре.

3.14 События и подписки

События позволяют получать обновления без периодического вызова getter-методов. Обработчики следует регистрировать после connect(), чтобы подписка MQTT создавалась при активном соединении.

```
import time

from sdk.manipulators.m13 import M13

robot = M13("192.168.0.10", "m13-events-01", "<login>", "<password>")

try:
    robot.connect()

    @robot.on_joint_states()
    def handle_joint_states(data: dict) -> None:
        print("Joint states:", data)

    @robot.on_hardware_error()
    def handle_hardware_error(data: dict) -> None:
        print("Hardware error:", data)

    time.sleep(10.0)
finally:
    robot.clear_all_handlers()
    robot.disconnect()
```

Декораторы регистрируют обработчик события после установления соединения. Обработчик должен быстро завершаться и не выполнять блокирующие команды движения; длительную обработку передавайте в очередь приложения. Перед disconnect() удалите обработчики методом clear_all_handlers() или отпишитесь от отдельных топиков.

События и декораторы подписки перечислены в таблице 31.

Таблица 31 – События и декораторы подписки

Декоратор	Топик/событие
on_coordinates()	Текущие координаты
on_joint_states()	Состояния суставов
on_gpio_states()	Состояния GPIO
on_hardware_state()	Состояние аппаратной части
on_hardware_error()	Аппаратные ошибки
on_coordinate_limits()	Координатные ограничения
on_gamepad_info()	Данные устройства ручного управления
on_manipulator_info()	Общая информация манипулятора

Альтернативно можно передать функцию методами `set_coordinates_handler()`, `set_joint_states_handler()`, `set_gpio_states_handler()` и другими `set_*_handler()`. Для снятия подписки применяются `unsubscribe_from_topic()`, `unsubscribe_from_streaming_topics()` и `clear_all_handlers()`.

3.15 Modbus RTU и RS-485

Интерфейс Modbus RTU предназначен для конфигурирования устройств на линии RS-485 и получения результатов опроса. Параметры линии, адрес ведомого устройства, функции, адреса регистров, порядок байтов и масштабирование должны соответствовать документации подключенного оборудования.

```
import time

from sdk.commands.modbus_command import ModbusSignal
from sdk.manipulators.m13 import M13

robot = M13("192.168.0.10", "m13-modbus-01", "<login>", "<password>")
SLAVE_ID = 1

try:
    robot.connect()
    robot.get_control()
```

```

signals = [ModbusSignal.holding_registers(address=12, count=2, poll_ms=200)]
robot.modbus_add_device(SLAVE_ID, "device_1", signals)
robot.modbus_toggle_polling(True)

time.sleep(1.0)
print(robot.last_modbus_data)
finally:
    try:
        robot.modbus_toggle_polling(False)
        robot.modbus_remove_device(SLAVE_ID)
    finally:
        robot.disconnect()

```

Методы таблицы управляют общей конфигурацией Modbus, списком устройств и циклическим опросом. Перед добавлением устройства остановите опрос, проверьте уникальность `slave_id` и сформируйте сигналы `ModbusSignal` по карте регистров. Результат последнего корректно разобранный ответ доступен в `robot.last_modbus_data` как словарь с полями `type`, `slave_id` и `values` либо как `None` до получения данных.

Методы Modbus RTU описаны в таблице 32.

Таблица 32 – Методы Modbus RTU

Метод	Назначение
<code>modbus_set_parameters(params)</code>	Передать параметры последовательной линии и опроса
<code>modbus_add_device(slave_id, name, signals)</code>	Добавить устройство и список сигналов
<code>modbus_remove_device(slave_id)</code>	Удалить устройство
<code>modbus_list_devices()</code>	Получить список устройств
<code>modbus_toggle_polling(enable)</code>	Включить или выключить опрос

3.15.1 Формирование сигналов ModbusSignal

Фабричные методы формируют описание операции Modbus: код функции, адрес, количество регистров и период опроса. Выбирайте метод в соответствии с картой регистров устройства и проверяйте допустимые адреса и формат передаваемых данных.

Фабричные методы описаны в таблице 33.

Таблица 33 – Фабричные методы

Фабричный метод	Назначение
holding_registers(address, count, poll_ms)	Чтение holding-регистров, функция 0x03
input_registers(address, count, poll_ms)	Чтение input-регистров, функция 0x04
coils(address, count, poll_ms)	Чтение coils, функция 0x01
discrete_inputs(address, count, poll_ms)	Чтение дискретных входов, функция 0x02
write_single_coil(address, value)	Запись одной катушки, функция 0x05
write_single_register(address, value)	Запись одного регистра, функция 0x06
write_multiple_coils(...)	Запись нескольких катушек, функция 0x0F
write_multiple_registers(...)	Запись нескольких регистров, функция 0x10
mask_write_register(...)	Маскированная запись, функция 0x16
read_write_multiple_registers(...)	Совмещённое чтение и запись, функция 0x17

Внимание! Значения address, count, value и элементы data должны попадать в диапазоны, разрешенные подключенным устройством. SDK формирует описание операции, но не проверяет смысл адресов, порядок слов и коэффициенты масштабирования конкретного прибора.

3.16 Дополнительные устройства

Методы для опциональных устройств и подсистем, которые могут отсутствовать в конкретной комплектации Promobot M13 описаны в таблице 34. Используйте соответствующее семейство только после проверки наличия оборудования, его настройки и поддержки текущей версией контроллера.

Таблица 34 – Методы для опциональных устройств и подсистем

Семейство методов	Назначение	Условие применения
play_audio*	Воспроизведение аудиофайла	Аудиоподсистема установлена и файл доступен контроллеру
play_gcode_draw*	Выполнение G-code для рисования	Установлена совместимая оснастка и проверена рабочая система координат
play_gcode_3dprint*	Выполнение G-code для 3D-печати	Установлено и настроено соответствующее оборудование
move_linear_module*	Перемещение линейного модуля	Модуль подключен и сконфигурирован
set_conveyer_velocity*	Задание скорости конвейера	В API 0.6.13 имя содержит conveyer; устройство подключено

3.17 Обработка ошибок и поиск неисправностей

3.17.1 Исключения SDK

Сопоставление основных исключений SDK с типовыми причинами их возникновения описано в таблице 35. В обработчиках сначала перехватывают более конкретные исключения, а затем базовый SdkError, сохраняя диагностическое сообщение в журнале.

Таблица 35 – Сопоставление основных исключений SDK с типовыми причинами их возникновения

Исключение	Ситуация
SdkError	Базовый класс ошибок SDK
ConnectionError	Ошибка подключения к MQTT или отправки сообщения
CommandTimeout	Не получен ожидаемый результат в пределах timeout_seconds
CommandError	Ошибка выполнения команды, когда она передана через иерархию SDK

```
import json
import logging
```

```

from sdk.errors import CommandTimeout, ConnectionError as SdkConnectionError,
SdkError

from sdk.manipulators.m13 import M13

logging.basicConfig(level=logging.INFO)
log = logging.getLogger(__name__)

def main() -> None:
    robot = M13(
        host="192.168.0.10",
        client_id="m13-production-client-01",
        login="<login>",
        password="<password>",
    )

    try:
        robot.connect()
        robot.get_control(timeout_seconds=10.0)

        state = robot.get_cobot_state(timeout_seconds=10.0)
        joints = json.loads(robot.get_joint_state(timeout_seconds=10.0))
        log.info("State: %s", state)
        log.info("Суставы: %s", joints)

        # Здесь размещаются только валидированные команды приложения.

    except SdkConnectionError:
        log.exception("Не удалось подключиться к манипулятору")
    except CommandTimeout:
        log.exception("Манипулятор не ответил в пределах тайм-аута")
    except SdkError:
        log.exception("Команда SDK завершилась ошибкой")
    except Exception:

```

```

log.exception("Непредвиденная ошибка приложения")
finally:
    robot.clear_all_handlers()
    robot.disconnect()

if __name__ == "__main__":
    main()

```

3.17.2 Типовые проблемы

Диагностика построена по схеме «признак – проверка – действие» и описана в таблице 36. Выполняйте проверки последовательно и не переходите к командам движения, пока причина ошибки подключения, управления или параметров не устранена.

Таблица 36 – Диагностика типовых проблем

Признак	Проверка	Действие
ModuleNotFoundError: sdk	Активно ли виртуальное окружение; установлен ли пакет pm-python-sdk	Активировать .venv и повторно выполнить установку пакета
ModuleNotFoundError: pydantic или paho	Установлен ли пакет SDK из утвержденного репозитория в активное виртуальное окружение	Повторно выполнить команду установки SDK из раздела «Получение и установка SDK»
Тайм-аут подключения	IP, маршрут, порт 1883, учетные данные	Проверить ping/Test-NetConnection/nc и настройки MQTT
Подключение обрывается при втором клиенте	Совпадают ли client_id	Назначить каждому процессу уникальный client_id
Команда движения не выполняется	Получено ли управление; состояние; TCP; ограничения; ошибки	Вызвать get_control(), прочитать get_current_errors() и проверить Promobot M Control
TypeError: object ... cannot be used in await	Использован ли обычный *_async, возвращающий Command/Promise	Выбрать метод *_async_await либо указанный coroutine-вариант
Получена JSON-строка вместо dict	Метод get_joint_state/get_cartesian_coordinates	Выполнить json.loads(result)

3.18 **Корректное завершение программы**

Каждое приложение должно завершать потоковый режим, удалять обработчики событий и закрывать соединение независимо от результата основной операции. Если команда завершилась по тайм-ауту, сначала определите фактическое состояние манипулятора и только затем переходите к следующим действиям.

- а) При активном streaming вызвать `enable_servo_streaming(False)`.
- б) При необходимости остановить выполняемое движение установленным проектом способом.
- в) Удалить обработчики `clear_all_handlers()` или отписаться от отдельных топиков.
- г) Вызвать `disconnect()` в блоке `finally`.
- д) Не полагаться только на автоматический вызов деструктора Python.

```
robot = M13(host, client_id, login, password)
try:
    robot.connect()
    # логика приложения
finally:
    robot.clear_all_handlers()
    robot.disconnect()
```

Примечание — В пользовательский справочник включены только методы, поддерживаемые для прикладного использования в SDK 0.6.13. Внутренние объекты `Command/Promise`, служебные обработчики и методы с известными дефектами намеренно не публикуются как пользовательский API.

3.19 **Карта методов по задачам**

Карта предназначена для быстрого поиска метода по пользовательской задаче, без просмотра полного алфавитного списка. После выбора семейства перейдите к соответствующему разделу руководства и проверьте предусловия и ограничения метода. Карта методов по задачам описана в таблице 37.

Таблица 37 – Карта методов по задачам

Задача	Рекомендуемые методы
Подключение	connect(), connect_async(), disconnect(), get_control(), get_control_async_await(), get_control_no_wait()
Движение	move_to_angles(), move_to_angles_async(), move_to_angles_no_wait(), move_to_coordinates(), move_to_coordinates_async_await(), move_to_coordinates_no_wait(), stop_movement(), arc_motion(), paletizing_movement(), move_group()
Диагностика	get_cartesian_coordinates(), get_joint_state(), get_home_position(), get_current_errors(), get_cobot_state()
TCP	tcp_apply(), tcp_delete(), tcp_get_current(), tcp_get_list()
Ввод-вывод	get_gpio_value(), write_gpio(), write_gpio_bit(), get_gpio_bit(), set_gpio_on_mask(), get_gpio_mask(), write_analog_output(), write_digital_output(), write_i2c()
Фланец	configure_flange_dout(), get_flange_dout(), configure_flange_vpwr(), get_flange_vpwr(), set_gpio_enable(), get_gpio_enable(), reset_gpio()
Программы	run_program_by_name(), run_program_json(), run_python_program()
Streaming	set_servo_joint_jog_mode(), set_servo_twist_mode(), set_servo_pose_mode(), enable_servo_streaming(), stream_joint_angles(), stream_joint_positions(), stream_cartesian_velocities(), stream_coordinates()
События	on_coordinates(), on_joint_states(), on_gpio_states(), on_hardware_error(), on_hardware_state(), unsubscribe_from_topic(), clear_all_handlers()
Modbus/RS-485	modbus_set_parameters(), modbus_add_device(), modbus_remove_device(), modbus_list_devices(), modbus_toggle_polling()

3.20 Поддерживаемый указатель методов M13

Указатель содержит поддерживаемые пользовательские и расширенные методы класса M13 с учетом наследования от Manipulator. Внутренние объекты Command/Promise, служебные обработчики, сервисные методы и методы с известными дефектами исключены. Для каждого метода приведены фактическая сигнатура SDK 0.6.13 и статус применения.

3.20.1 Подключение и управление

Методы установления соединения, получения права управления и завершения сеанса собраны в таблице 38. Их следует использовать в начале и конце каждого приложения, соблюдая уникальность client_id и гарантированное отключение.

Таблица 38 – Методы установления соединения, получения права управления и завершения сеанса

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
connect	connect() -> None	Пользовательский
connect_async	connect_async() -> None	Пользовательский
disconnect	disconnect() -> None	Пользовательский
get_control	get_control(timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
get_control_async_wait	get_control_async_wait(timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
get_control_no_wait	get_control_no_wait() -> None	Расширенный: без ожидания

3.20.2 Состояние и диагностика

Методы получения телеметрии, текущих координат, состояний суставов, ошибок и диагностических данных приведены в таблице 39. Большинство из них не изменяют состояние оборудования и подходят для предварительных проверок и мониторинга.

Таблица 39 – Методы получения телеметрии, текущих координат, состояний суставов, ошибок и диагностических данных

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
get_cartesian_coordinates	get_cartesian_coordinates(timeout_seconds: float = 60.0, throw_error: bool = True) -> str	Пользовательский
get_cartesian_coordinates_async_wait	get_cartesian_coordinates_async_wait(timeout_seconds: float = 60.0, throw_error: bool = True) -> str	Пользовательский
get_cobot_state	get_cobot_state(timeout_seconds: float = 20.0, throw_error: bool = True) -> int str None	Пользовательский
get_current_errors	get_current_errors(timeout_seconds: float = 60.0, throw_error: bool = True)	Пользовательский
get_home_position	get_home_position(timeout_seconds: float = 60.0, throw_error: bool = True) -> dict	Пользовательский
get_home_position_async_wait	get_home_position_async_wait(timeout_seconds: float = 60.0, throw_error: bool = True) -> dict	Пользовательский

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
get_joint_state	get_joint_state(timeout_seconds: float = 60.0, throw_error: bool = True) -> str	Пользовательский
get_joint_state_async_await	get_joint_state_async_await(timeout_seconds: float = 60.0, throw_error: bool = True) -> str	Пользовательский

3.20.3 Состояния и останов

Методы управления состояниями жизненного цикла и программными остановами содержатся в таблице 40. Применяйте их только в допустимых переходах контроллера и не смешивайте программный останов с аппаратными функциями безопасности.

Таблица 40 – Методы управления состояниями жизненного цикла и программными остановами

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
change_state	change_state(state: ManipulatorState, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
change_state_async_await	change_state_async_await(state: ManipulatorState, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
change_state_no_wait	change_state_no_wait(state: ManipulatorState) -> None	Расширенный: без ожидания
stop_movement	stop_movement(timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
stop_movement_async_await	stop_movement_async_await(timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
stop_movement_no_wait	stop_movement_no_wait() -> None	Расширенный: без ожидания

3.20.4 Движение и планирование

Суставные, декартовы, дуговые и групповые движения, а также варианты ожидания результата собраны в таблице 41. Для каждого сценария необходимо определить планировщик, ограничения скорости и ускорения и проверенную целевую конфигурацию.

Таблица 41 – Суставные, декартовы, дуговые и групповые движения

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
arc_motion	arc_motion(target: Pose, center_arc: Pose, step: float = 0.05, count_point_arc: int = 50, max_velocity_scaling_factor: float = 0.5, max_acceleration_scaling_factor: float = 0.5, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
arc_motion_async_await	arc_motion_async_await(target: Pose, center_arc: Pose, step: float = 0.05, count_point_arc: int = 50, max_velocity_scaling_factor: float = 0.5, max_acceleration_scaling_factor: float = 0.5, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
move_group	move_group(points: Optional[List[Point3D]] = None, positions: Optional[List[JointPositions]] = None, move_group: str = 'main', planning_pipeline: str = 'pilz_industrial_motion_planner', planner_id: str = 'PTP', max_velocity: float = 0.5, max_acceleration: float = 0.5, position_tolerance: float = 0.01, orientation_tolerance: float = 0.01, blend_radius: float = 0.1, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
move_to_angles	move_to_angles(sp1: float, sp2: float, sp3: float, sp4: float, sp5: float, sp6: float, sp1_v: float = 0.0, sp2_v: float = 0.0, sp3_v: float = 0.0, sp4_v: float = 0.0, sp5_v: float = 0.0, sp6_v: float = 0.0, velocity_factor: float = 0.1, acceleration_factor: float = 0.1, planner_id: str = 'PTP', timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
move_to_angles_async	move_to_angles_async(sp1: float, sp2: float, sp3: float, sp4: float, sp5: float, sp6: float, sp1_v: float = 0.0, sp2_v: float = 0.0, sp3_v: float = 0.0, sp4_v: float = 0.0, sp5_v: float = 0.0, sp6_v: float = 0.0, velocity_factor: float = 0.1, acceleration_factor: float = 0.1, planner_id: str = 'PTP', timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
move_to_angles_no_wait	move_to_angles_no_wait(sp1: float, sp2: float, sp3: float, sp4: float, sp5: float, sp6: float, sp1_v: float = 0.0, sp2_v: float = 0.0, sp3_v: float = 0.0, sp4_v: float = 0.0, sp5_v: float = 0.0, sp6_v: float = 0.0, velocity_factor: float = 0.1, acceleration_factor: float = 0.1, planner_id: str = 'PTP') -> None	Расширенный: без ожидания

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
move_to_coordinates	move_to_coordinates(position: MoveCoordinatesParamsPosition, orientation: MoveCoordinatesParamsOrientation, velocity_scaling_factor: float, acceleration_scaling_factor: float, planner_type: PlannerType = PlannerType.LIN, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
move_to_coordinates_async_await	move_to_coordinates_async_await(position: MoveCoordinatesParamsPosition, orientation: MoveCoordinatesParamsOrientation, velocity_scaling_factor: float, acceleration_scaling_factor: float, planner_type: PlannerType = PlannerType.LIN, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
move_to_coordinates_no_wait	move_to_coordinates_no_wait(position: MoveCoordinatesParamsPosition, orientation: MoveCoordinatesParamsOrientation, velocity_scaling_factor: float, acceleration_scaling_factor: float) -> None	Расширенный: без ожидания
paletizing_movement	paletizing_movement(target_point: Pose, hold_orientation: Orientation = Orientation(), use_orientation: bool = False, step: float = 0.05, count_point: int = 50, max_velocity_scaling_factor: float = 0.1, max_acceleration_scaling_factor: float = 0.1, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский

3.20.5 TCP и инструмент

Чтение, выбор и удаление сохраненного TCP приведены в следующей таблице 42. Создание и редактирование TCP выполняйте в Promobot M Control.

Таблица 42 – Чтение, выбор и удаление сохраненного TCP

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
tcp_apply	tcp_apply(name: str, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
tcp_delete	tcp_delete(name: str, reset_current: bool = True, apply_other: str = "", timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
tcp_get_current	tcp_get_current(timeout_seconds: float = 60.0, throw_error: bool = True) -> TCPInfo	Пользовательский
tcp_get_list	tcp_get_list(timeout_seconds: float = 60.0, throw_error: bool = True) -> tuple[list[TCPInfo], str]	Пользовательский

3.20.6 Ввод-вывод и фланец

Методы именованного GPIO, побитовых и масочных операций, аналоговых и цифровых выходов, I2C и интерфейсов фланца перечислены в таблице 43. Перед записью требуется проверить назначение ресурса и электрические ограничения подключенной цепи. Таблица 43 – Методы именованного GPIO, побитовых и масочных операций, аналоговых и цифровых выходов, I2C и интерфейсов фланца

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
configure_flange_dout	configure_flange_dout(pin_name: FlangePinName, output_type: OutputType, timeout_seconds: float = 60.0, throw_error: bool = True)	Пользовательский
configure_flange_vpwr	configure_flange_vpwr(mode: VPWRType, timeout_seconds: float = 60.0, throw_error: bool = True)	Пользовательский
get_flange_dout	get_flange_dout(pin_name: FlangePinName, timeout_seconds: float = 60.0, throw_error: bool = True)	Пользовательский
get_flange_vpwr	get_flange_vpwr(timeout_seconds: float = 60.0, throw_error: bool = True)	Пользовательский
get_gpio_bit	get_gpio_bit(name: str, timeout_seconds: float = 60.0, throw_error: bool = True) -> Optional[float]	Пользовательский
get_gpio_enable	get_gpio_enable(board_name: BoardName, timeout_seconds: float = 60.0, throw_error: bool = True)	Пользовательский
get_gpio_mask	get_gpio_mask(name: str, timeout_seconds: float = 60.0, throw_error: bool = True) -> str None	Пользовательский

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
get_gpio_value	get_gpio_value(name: str, timeout_seconds: float = 60.0, throw_error: bool = True) -> Optional[float]	Пользовательский
on_gpio_states	on_gpio_states()	Пользовательский
reset_gpio	reset_gpio(timeout_seconds: float = 60.0, throw_error: bool = True)	Пользовательский
set_gpio_enable	set_gpio_enable(board_name: BoardName, enable: bool, timeout_seconds: float = 60.0, throw_error: bool = True)	Пользовательский
set_gpio_on_mask	set_gpio_on_mask(name: str, value_mask: str, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
set_gpio_states_handler	set_gpio_states_handler(handler: Callable[[Dict[str, Any]], None]) -> None	Пользовательский
write_analog_output	write_analog_output(channel: int, value: float, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
write_analog_output_async_await	write_analog_output_async_await(channel: int, value: float, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
write_analog_output_no_wait	write_analog_output_no_wait(channel: int, value: float) -> None	Расширенный: без ожидания
write_digital_output	write_digital_output(channel: int, value: bool, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
write_digital_output_async_await	write_digital_output_async_await(channel: int, value: bool, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
write_digital_output_no_wait	write_digital_output_no_wait(channel: int, value: bool) -> None	Расширенный: без ожидания
write_gpio	write_gpio(name: str, value: int, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
write_gpio_async_await	write_gpio_async_await(name: str, value: int, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
write_gpio_bit	write_gpio_bit(name: str, value: int, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
write_i2c	write_i2c(name: str, value: int, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
write_i2c_async_await	write_i2c_async_await(name: str, value: int, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский

3.20.7 Потокное сервоуправление

Методы выбора режима сервоуправления, включения потоковой передачи и отправки последовательных целевых значений содержит таблица 44. Эти интерфейсы требуют устойчивого цикла приложения и заранее определённой реакции на потерю связи или прекращение потока.

Таблица 44 – Методы выбора режима сервоуправления, включения потоковой передачи и отправки последовательных целевых значений

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
enable_servo_streaming	enable_servo_streaming(enabled: bool = True, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный
enable_servo_streaming_async	enable_servo_streaming_async(enabled: bool = True, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный
set_servo_control_type	set_servo_control_type(control_type: ServoControlType, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный
set_servo_control_type_async_await	set_servo_control_type_async_await(control_type: ServoControlType, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный
set_servo_control_type_no_wait	set_servo_control_type_no_wait(control_type: ServoControlType) -> None	Расширенный: без ожидания
set_servo_joint_jog_mode	set_servo_joint_jog_mode(timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный
set_servo_joint_jog_mode_no_wait	set_servo_joint_jog_mode_no_wait() -> None	Расширенный: без ожидания
set_servo_pose_mode	set_servo_pose_mode(timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный
set_servo_pose_mode_no_wait	set_servo_pose_mode_no_wait() -> None	Расширенный: без ожидания

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
set_servo_twist_mode	set_servo_twist_mode(timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный
set_servo_twist_mode_no_wait	set_servo_twist_mode_no_wait() -> None	Расширенный: без ожидания
stream_cartesian_velocities	stream_cartesian_velocities(linear_velocities: Dict[str, float], angular_velocities: Dict[str, float]) -> None	Расширенный
stream_cartesian_velocities_async	stream_cartesian_velocities_async(linear_velocities: Dict[str, float], angular_velocities: Dict[str, float]) -> None	Расширенный
stream_coordinates	stream_coordinates(position: MoveCoordinatesParamsPosition, orientation: MoveCoordinatesParamsOrientation) -> None	Расширенный
stream_coordinates_async	stream_coordinates_async(position: MoveCoordinatesParamsPosition, orientation: MoveCoordinatesParamsOrientation) -> None	Расширенный
stream_joint_angles	stream_joint_angles(sp1: float, sp2: float, sp3: float, sp4: float, sp5: float, sp6: float, v1: float = 0.0, v2: float = 0.0, v3: float = 0.0, v4: float = 0.0, v5: float = 0.0, v6: float = 0.0) -> None	Расширенный
stream_joint_angles_async	stream_joint_angles_async(sp1: float, sp2: float, sp3: float, sp4: float, sp5: float, sp6: float, v1: float = 0.0, v2: float = 0.0, v3: float = 0.0, v4: float = 0.0, v5: float = 0.0, v6: float = 0.0) -> None	Расширенный
stream_joint_positions	stream_joint_positions(positions: Dict[str, float], velocities: Dict[str, float]) -> None	Расширенный
stream_joint_positions_async	stream_joint_positions_async(positions: Dict[str, float], velocities: Dict[str, float]) -> None	Расширенный

3.20.8 Выполнение программ

Методы запуска сохраненных, JSON-, Python- и G-code-программ, включая варианты без ожидания собраны в таблице 45. Перед вызовом убедитесь, что формат программы поддерживается контроллером и все используемые ресурсы доступны.

Таблица 45 – Методы запуска сохраненных, JSON-, Python- и G-code-программ, включая варианты без ожидания

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
play_gcode_3dprint	play_gcode_3dprint(gcode_file: str, temperature: float = 200, velocity_near: float = 0.01, velocity_far: float = 0.01, acceleration_near: float = 0.02, acceleration_far: float = 0.01, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный
play_gcode_3dprint_async_await	play_gcode_3dprint_async_await(gcode_file: str, temperature: float = 200, velocity_near: float = 0.01, velocity_far: float = 0.01, acceleration_near: float = 0.02, acceleration_far: float = 0.01, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный
play_gcode_draw	play_gcode_draw(gcode_file: str, velocity_near: float = 0.01, velocity_far: float = 0.01, acceleration_near: float = 0.04, acceleration_far: float = 0.02, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный
play_gcode_draw_async_await	play_gcode_draw_async_await(gcode_file: str, velocity_near: float = 0.01, velocity_far: float = 0.01, acceleration_near: float = 0.04, acceleration_far: float = 0.02, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный
run_program_by_name	run_program_by_name(program_name: str, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
run_program_by_name_async_await	run_program_by_name_async_await(program_name: str, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
run_program_by_name_no_wait	run_program_by_name_no_wait(program_name: str) -> None	Расширенный: без ожидания
run_program_json	run_program_json(name: str, program_json: dict, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
run_program_json_async_await	run_program_json_async_await(name: str, program_json: dict, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
run_program_json_no_wait	run_program_json_no_wait(name: str, program_json: dict) -> None	Расширенный: без ожидания
run_python_program	run_python_program(python_code: str, python_version: str = '3.12', requirements: dict) -> None	Пользовательский

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
	List[str] = None, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	
run_python_program_async_await	run_python_program_async_await(python_code: str, python_version: str = '3.12', requirements: List[str] = None, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
run_python_program_no_wait	run_python_program_no_wait(python_code: str, python_version: str = '3.12', requirements: List[str] = None) -> None	Расширенный: без ожидания

3.20.9 События и подписки

Декораторы и обработчики для получения событий контроллера без периодического опроса перечислены в таблице 46. Обработчики регистрируются после подключения и должны быть удалены при завершении приложения.

Таблица 46 – Декораторы и обработчики для получения событий контроллера без периодического опроса

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
on_coordinate_limits	on_coordinate_limits()	Пользовательский
on_coordinates	on_coordinates()	Пользовательский
on_gamepad_info	on_gamepad_info()	Пользовательский
on_hardware_error	on_hardware_error()	Пользовательский
on_hardware_state	on_hardware_state()	Пользовательский
on_joint_states	on_joint_states()	Пользовательский
on_manipulator_info	on_manipulator_info()	Пользовательский
set_coordinate_limits_handler	set_coordinate_limits_handler(handler: Callable[[Dict[str, Any]], None]) -> None	Пользовательский
set_coordinates_handler	set_coordinates_handler(handler: Callable[[Dict[str, Any]], None]) -> None	Пользовательский
set_gamepad_info_handler	set_gamepad_info_handler(handler: Callable[[Dict[str, Any]], None]) -> None	Пользовательский
set_hardware_error_handler	set_hardware_error_handler(handler: Callable[[Dict[str, Any]], None]) -> None	Пользовательский

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
set_hardware_state_handler	set_hardware_state_handler(handler: Callable[[Dict[str, Any]], None]) -> None	Пользовательский
set_joint_states_handler	set_joint_states_handler(handler: Callable[[Dict[str, Any]], None]) -> None	Пользовательский
set_manipulator_info_handler	set_manipulator_info_handler(handler: Callable[[Dict[str, Any]], None]) -> None	Пользовательский
subscribe_to_joint_state	subscribe_to_joint_state(callback: callable) -> None	Пользовательский
unsubscribe_from_joint_state	unsubscribe_from_joint_state() -> None	Пользовательский
unsubscribe_from_streaming_topics	unsubscribe_from_streaming_topics() -> None	Пользовательский
unsubscribe_from_topic	unsubscribe_from_topic(topic: str) -> None	Пользовательский
clear_all_handlers	clear_all_handlers() -> None	Пользовательский

3.20.10 Modbus RTU

Методы настройки устройств Modbus и управления циклическим опросом приведены в следующей таблице 47. Параметры линии, адрес ведомого устройства и карта регистров должны совпадать с документацией внешнего оборудования.

Таблица 47 – Методы настройки устройств Modbus и управления циклическим опросом

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
modbus_add_device	modbus_add_device(slave_id: int, name: str, signals: List[Dict[str, Any]], timeout_seconds: float = 60.0, throw_error: bool = True) -> Dict[str, Any]	Расширенный
modbus_list_devices	modbus_list_devices(timeout_seconds: float = 60.0, throw_error: bool = True) -> List[Dict[str, Any]]	Расширенный
modbus_remove_device	modbus_remove_device(slave_id: int, timeout_seconds: float = 60.0, throw_error: bool = True) -> Dict[str, Any]	Расширенный
modbus_set_parameters	modbus_set_parameters(params: Dict[str, Any], timeout_seconds: float = 60.0, throw_error: bool = True) -> Dict[str, Any]	Расширенный

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
modbus_toggle_polling	modbus_toggle_polling(enable: bool = True, timeout_seconds: float = 60.0, throw_error: bool = True) -> Dict[str, Any]	Расширенный

3.20.11 Периферия

Интерфейсы поддерживаемой опциональной периферии приведены в следующей таблице 48. Наличие метода в SDK не подтверждает наличие соответствующего оборудования в конкретной поставке.

Таблица 48 – Интерфейсы поддерживаемой опциональной периферии

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
move_linear_module	move_linear_module(distance: float, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный
move_linear_module_async_await	move_linear_module_async_await(distance: float, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный
play_audio	play_audio(file_name: str, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
play_audio_async_await	play_audio_async_await(file_name: str, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
play_audio_in_background	play_audio_in_background(file_name: str, timeout_seconds: float = 60.0, throw_error: bool = True) -> BackgroundAudioPlayback	Пользовательский
set_conveyer_velocity	set_conveyer_velocity(velocity: float, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный
set_conveyer_velocity_async_await	set_conveyer_velocity_async_await(velocity: float, timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Расширенный

3.20.12 Ограничения суставов

Методы настройки ограничений суставов предназначены для авторизованной конфигурации рабочего места. Значения должны соответствовать утвержденным ограничениям конкретного манипулятора; после записи требуется независимая проверка конфигурации в Promobot M Control.

Методы настройки ограничений суставов описаны в таблице 49.

Таблица 49 – Методы настройки ограничений суставов

Метод	Фактическая сигнатура SDK 0.6.13	Статус применения
set_joint_limits	set_joint_limits(limits: list[JointLimit], timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский
set_joint_limits_async_await	set_joint_limits_async_await(limits: list[JointLimit], timeout_seconds: float = 60.0, throw_error: bool = True) -> None	Пользовательский

3.21 Алфавитный указатель

Алфавитный указатель, представленный в таблице 50, позволяет найти метод по точному имени и определить его функциональный раздел и статус применения. После поиска используйте основной раздел или полный указатель методов для проверки сигнатуры и ограничений.

Таблица 50 – Алфавитный указатель

Метод	Функциональный раздел	Статус
arc_motion	Движение и планирование	Пользовательский
arc_motion_async_await	Движение и планирование	Пользовательский
change_state	Состояния и останов	Пользовательский
change_state_async_await	Состояния и останов	Пользовательский
change_state_no_wait	Состояния и останов	Расширенный: без ожидания
clear_all_handlers	События и подписки	Пользовательский
configure_flange_dout	Ввод-вывод и фланец	Пользовательский
configure_flange_vpwr	Ввод-вывод и фланец	Пользовательский

Метод	Функциональный раздел	Статус
connect	Подключение и управление	Пользовательский
connect_async	Подключение и управление	Пользовательский
disconnect	Подключение и управление	Пользовательский
enable_servo_streaming	Потоковое/сервоуправление	Расширенный
enable_servo_streaming_async	Потоковое/сервоуправление	Расширенный
get_cartesian_coordinates	Состояние и диагностика	Пользовательский
get_cartesian_coordinates_async_await	Состояние и диагностика	Пользовательский
get_cobot_state	Состояние и диагностика	Пользовательский
get_control	Подключение и управление	Пользовательский
get_control_async_await	Подключение и управление	Пользовательский
get_control_no_wait	Подключение и управление	Расширенный: без ожидания
get_current_errors	Состояние и диагностика	Пользовательский
get_flange_dout	Ввод-вывод и фланец	Пользовательский
get_flange_vpwr	Ввод-вывод и фланец	Пользовательский
get_gpio_bit	Ввод-вывод и фланец	Пользовательский
get_gpio_enable	Ввод-вывод и фланец	Пользовательский
get_gpio_mask	Ввод-вывод и фланец	Пользовательский
get_gpio_value	Ввод-вывод и фланец	Пользовательский
get_home_position	Состояние и диагностика	Пользовательский
get_home_position_async_await	Состояние и диагностика	Пользовательский
get_joint_state	Состояние и диагностика	Пользовательский
get_joint_state_async_await	Состояние и диагностика	Пользовательский
modbus_add_device	Modbus/RS-485	Расширенный
modbus_list_devices	Modbus/RS-485	Расширенный
modbus_remove_device	Modbus/RS-485	Расширенный
modbus_set_parameters	Modbus/RS-485	Расширенный
modbus_toggle_polling	Modbus/RS-485	Расширенный
move_group	Движение и планирование	Пользовательский
move_linear_module	Периферия	Расширенный

Метод	Функциональный раздел	Статус
move_linear_module_async_await	Периферия	Расширенный
move_to_angles	Движение и планирование	Пользовательский
move_to_angles_async	Движение и планирование	Пользовательский
move_to_angles_no_wait	Движение и планирование	Расширенный: без ожидания
move_to_coordinates	Движение и планирование	Пользовательский
move_to_coordinates_async_await	Движение и планирование	Пользовательский
move_to_coordinates_no_wait	Движение и планирование	Расширенный: без ожидания
on_coordinate_limits	События и подписки	Пользовательский
on_coordinates	События и подписки	Пользовательский
on_gamepad_info	События и подписки	Пользовательский
on_gpio_states	Ввод-вывод и фланец	Пользовательский
on_hardware_error	События и подписки	Пользовательский
on_hardware_state	События и подписки	Пользовательский
on_joint_states	События и подписки	Пользовательский
on_manipulator_info	События и подписки	Пользовательский
paletizing_movement	Движение и планирование	Пользовательский
play_audio	Периферия	Пользовательский
play_audio_async_await	Периферия	Пользовательский
play_audio_in_background	Периферия	Пользовательский
play_gcode_3dprint	Выполнение программ	Расширенный
play_gcode_3dprint_async_await	Выполнение программ	Расширенный
play_gcode_draw	Выполнение программ	Расширенный
play_gcode_draw_async_await	Выполнение программ	Расширенный
reset_gpio	Ввод-вывод и фланец	Пользовательский
run_program_by_name	Выполнение программ	Пользовательский
run_program_by_name_async_await	Выполнение программ	Пользовательский

Метод	Функциональный раздел	Статус
run_program_by_name_no_wait	Выполнение программ	Расширенный: без ожидания
run_program_json	Выполнение программ	Пользовательский
run_program_json_async_await	Выполнение программ	Пользовательский
run_program_json_no_wait	Выполнение программ	Расширенный: без ожидания
run_python_program	Выполнение программ	Пользовательский
run_python_program_async_await	Выполнение программ	Пользовательский
run_python_program_no_wait	Выполнение программ	Расширенный: без ожидания
set_conveyer_velocity	Периферия	Расширенный
set_conveyer_velocity_async_await	Периферия	Расширенный
set_coordinate_limits_handler	События и подписки	Пользовательский
set_coordinates_handler	События и подписки	Пользовательский
set_gamepad_info_handler	События и подписки	Пользовательский
set_gpio_enable	Ввод-вывод и фланец	Пользовательский
set_gpio_on_mask	Ввод-вывод и фланец	Пользовательский
set_gpio_states_handler	Ввод-вывод и фланец	Пользовательский
set_hardware_error_handler	События и подписки	Пользовательский
set_hardware_state_handler	События и подписки	Пользовательский
set_joint_limits	Прочее	Пользовательский
set_joint_limits_async_await	Прочее	Пользовательский
set_joint_states_handler	События и подписки	Пользовательский
set_manipulator_info_handler	События и подписки	Пользовательский
set_servo_control_type	Потоковое/сервоуправление	Расширенный
set_servo_control_type_async_await	Потоковое/сервоуправление	Расширенный
set_servo_control_type_no_wait	Потоковое/сервоуправление	Расширенный: без ожидания
set_servo_joint_jog_mode	Потоковое/сервоуправление	Расширенный

Метод	Функциональный раздел	Статус
set_servo_joint_jog_mode_no_wait	Потоковое/сервоуправление	Расширенный: без ожидания
set_servo_pose_mode	Потоковое/сервоуправление	Расширенный
set_servo_pose_mode_no_wait	Потоковое/сервоуправление	Расширенный: без ожидания
set_servo_twist_mode	Потоковое/сервоуправление	Расширенный
set_servo_twist_mode_no_wait	Потоковое/сервоуправление	Расширенный: без ожидания
stop_movement	Состояния и останов	Пользовательский
stop_movement_async_await	Состояния и останов	Пользовательский
stop_movement_no_wait	Состояния и останов	Расширенный: без ожидания
stream_cartesian_velocities	Потоковое/сервоуправление	Расширенный
stream_cartesian_velocities_async	Потоковое/сервоуправление	Расширенный
stream_coordinates	Потоковое/сервоуправление	Расширенный
stream_coordinates_async	Потоковое/сервоуправление	Расширенный
stream_joint_angles	Потоковое/сервоуправление	Расширенный
stream_joint_angles_async	Потоковое/сервоуправление	Расширенный
stream_joint_positions	Потоковое/сервоуправление	Расширенный
stream_joint_positions_async	Потоковое/сервоуправление	Расширенный
subscribe_to_joint_state	События и подписки	Пользовательский
tcp_apply	ТСР и инструмент	Пользовательский
tcp_delete	ТСР и инструмент	Пользовательский
tcp_get_current	ТСР и инструмент	Пользовательский
tcp_get_list	ТСР и инструмент	Пользовательский
unsubscribe_from_joint_state	События и подписки	Пользовательский
unsubscribe_from_streaming_topics	События и подписки	Пользовательский
unsubscribe_from_topic	События и подписки	Пользовательский
write_analog_output	Ввод-вывод и фланец	Пользовательский
write_analog_output_async_await	Ввод-вывод и фланец	Пользовательский

Метод	Функциональный раздел	Статус
write_analog_output_no_wait	Ввод-вывод и фланец	Расширенный: без ожидания
write_digital_output	Ввод-вывод и фланец	Пользовательский
write_digital_output_async_wait	Ввод-вывод и фланец	Пользовательский
write_digital_output_no_wait	Ввод-вывод и фланец	Расширенный: без ожидания
write_gpio	Ввод-вывод и фланец	Пользовательский
write_gpio_async_wait	Ввод-вывод и фланец	Пользовательский
write_gpio_bit	Ввод-вывод и фланец	Пользовательский
write_i2c	Ввод-вывод и фланец	Пользовательский
write_i2c_async_wait	Ввод-вывод и фланец	Пользовательский

3.22 Контрольный список перед запуском

Контрольный список, представленный в таблице 51, необходимо пройти перед первым подключением и повторять перед выполнением команд движения после изменения конфигурации. Невыполненный или неподтверждённый пункт является основанием не запускать программу до устранения замечания.

Таблица 51 – Контрольный список

№	Проверка
1	Python 3.12.x установлен, виртуальное окружение активно.
2	Зависимости и утвержденный пакет SDK установлены без ошибок; версия пакета соответствует документации.
3	Контроллер доступен по сети, TCP-порт 1883 открыт только в доверенной сети.
4	client_id уникален, учетные данные получены для данной установки.
5	Первый read-only тест успешно получил координаты и состояния суставов.
6	Рабочая зона освобождена, аварийный останов доступен.
7	TCP, инструмент, полезная нагрузка и ограничения проверены.
8	Передаваемые точки и вся траектория валидированы.
9	Ошибки и тайм-ауты обрабатываются; после тайм-аута проверяется фактическое состояние манипулятора.
10	disconnect(), удаление обработчиков и отключение streaming находятся в finally.

3.23 Минимальный шаблон приложения

Шаблон объединяет рекомендуемую структуру пользовательского приложения: настройку журналирования, создание Promobot M13, подключение, безопасное чтение состояния, обработку исключений и гарантированное отключение в блоке finally. Команды движения намеренно не включены и должны добавляться только после проверки конкретного сценария.

```

import json
import logging

from sdk.errors import CommandTimeout, ConnectionError as SdkConnectionError,
SdkError

from sdk.manipulators.m13 import M13

logging.basicConfig(level=logging.INFO)
log = logging.getLogger(__name__)

def main() -> None:
    robot = M13(
        host="192.168.0.10",
        client_id="m13-production-client-01",
        login="<login>",
        password="<password>",
    )

    try:
        robot.connect()
        robot.get_control(timeout_seconds=10.0)

        state = robot.get_cobot_state(timeout_seconds=10.0)
        joints = json.loads(robot.get_joint_state(timeout_seconds=10.0))
        log.info("State: %s", state)
        log.info("Систавы: %s", joints)

        # Здесь размещаются только валидированные команды приложения.

    except SdkConnectionError:

```

```

        log.exception("Не удалось подключиться к манипулятору")
except CommandTimeout:
    log.exception("Манипулятор не ответил в пределах тайм-аута")
except SdkError:
    log.exception("Команда SDK завершилась ошибкой")
except Exception:
    log.exception("Непредвиденная ошибка приложения")
finally:
    robot.clear_all_handlers()
    robot.disconnect()

if __name__ == "__main__":
    main()

```

4 SDK C++

4.1 Требования и использование C++

Манипулятор и компьютер должны быть доступны друг другу по сети (локально или через Интернет).

SDK требует компилятор и систему сборки CMake. Требование к компилятору: компилятор с поддержкой C++17 (GCC 9+, Clang 10+, MSVC 2019+). Библиотека распространяется в предварительно собранном виде.

4.1.1 Проверка компилятора

```

g++ --version
# ожидается версия 9 или выше

```

4.1.2 Структура установленного SDK

После получения SDK, в корневой папке находится каталог `install` со следующей структурой:

```
install/
├── include/
│   ├── pm_sdk/      # Заголовочные файлы SDK
│   │   ├── manipulator.hpp
│   │   ├── types.hpp
│   │   └── debug_logger.hpp
│   └── nlohmann/    # JSON библиотека
├── mqtt/            # Заголовочные файлы Paho MQTT
├── lib/
│   ├── libpm_sdk.a  # Статическая библиотека SDK
│   ├── libpaho-mqtttp3.so  # Paho MQTT C++ библиотека
│   └── libpaho-mqtt3*.so  # Paho MQTT C библиотеки
└── bin/
    └── MQTTVersion  # Утилита для проверки MQTT
```

4.1.3 Подключение SDK к проекту

Пример `CMakeLists.txt` для вашего проекта:

```
cmake_minimum_required(VERSION 3.10)
project(sdk_demo)

set(CMAKE_CXX_STANDARD 17)

include_directories(${CMAKE_SOURCE_DIR}/../install/include)
link_directories(${CMAKE_SOURCE_DIR}/../install/lib)

add_executable(${PROJECT_NAME} main.cpp)
```

```
target_link_libraries(${PROJECT_NAME}
    pm_sdk
    paho-mqttpp3
    paho-mqtt3as
)
```

4.2 Типы команд

SDK поддерживает несколько типов команд для гибкого управления:

- синхронные – выполняются по очереди и блокируют выполнение до завершения. Принимают параметр `throwException` (по умолчанию `true`).
- асинхронные – не блокируют выполнение; имеют суффикс `Async` и принимают `callback`-функцию.
- мстриминговые – для реального управления малыми шагами (скорости, позы, суставы).
- подписки – для асинхронного получения данных состояния через `Subscribe/Unsubscribe`.

Важно! Не все методы имеют асинхронные версии. Например:

- `GetControl()` – только синхронный;
- `Connect()` – только синхронный;
- `CancelCommand()` – только синхронный;
- `GetCartesianCoordinates()` – только синхронный (но есть асинхронная подписка `SubscribeCartesianCoordinates`).

Большинство команд движения (`MoveToAngles`, `MoveToCoordinates` и др.) имеют как синхронные, так и асинхронные версии. Всегда проверяйте наличие суффикса `Async` в заголовочных файлах.

4.2.1 Примеры вызовов

```
// Синхронный вызов (блокирующий)
manipulator.GetControl();
manipulator.MoveToAngles(angles);
```

```

// Асинхронный вызов (с callback)
manipulator.MoveToAnglesAsync(angles, [](CommandResult result) {
    if (result.GetState()) {
        std::cout << "Движение завершено" << std::endl;
    }
});

// асинхронная подписка на состояния
manipulator.SubscribeCartesianCoordinates([](const nlohmann::json& data) {
    std::cout << "Координаты обновлены: " << data.dump() << std::endl;
});

```

4.2.2 Управление логированием

Для отладки SDK предусмотрена система логирования:

```

// Подключить заголовочный файл
#include "pm_sdk/debug_logger.hpp"

// Включить все логи
DebugLogger::SetEnable(LogEntity::ALL, true);

// Включить только логи SDK
DebugLogger::SetEnable(LogEntity::SDK, true);

// Включить только логи MQTT
DebugLogger::SetEnable(LogEntity::MQTT, true);

// Отключить все логи
DebugLogger::SetEnable(LogEntity::ALL, false);

```

Далее в примерах будут использоваться синхронные методы, если не указано иное.

4.3 Программные блоки (структура программы)

Структуру программы удобно разбить на логические блоки:

- а) Подключение заголовочных файлов SDK.
- б) Создание объекта манипулятора и подключение к нему.
- в) Захват управления.
- г) Основной блок выполнения (движения, чтение данных).
- д) Обработка исключений.

Ниже – минимальный пример подключения и захвата управления.

```
#include "pm_sdk/manipulator.hpp"
#include <iostream>

int main() {
    using namespace pb_sdk;

    std::string host = "192.168.88.182"; // IP манипулятора
    std::string client_id = "122";      // ID клиента
    std::string login = "13";           // Логин
    std::string password = "14";        // Пароль

    try {
        M13 manipulator(host, client_id, login, password);
        manipulator.Connect();

        if (manipulator.GetControl()) {
            std::cout << "Управление получено! " << std::endl;
        }
    } catch (const std::exception& e) {
        std::cerr << "Ошибка: " << e.what() << std::endl;
    }

    return 0;
}
```

4.3.1 Основной блок выполнения (движения, чтение данных)

Основной блок выполнения (движения, чтение данных) описан в таблице 52.

Таблица 52 – Основной блок выполнения (движения, чтение данных)

Наименование	Описание	Команда	Параметры	Пример
Команды перемещения				
MoveToAngles	Перемещает манипулятор по углам шести суставов: shoulder_pan_joint, shoulder_lift_joint, elbow_joint, wrist_1_joint, wrist_2_joint, wrist_3_joint	manipulator.MoveToAngles(angles);	- angles.m_position – JointPositions для 6 суставов [рад]; - angles.m_velocities – JointVelocities для 6 суставов [рад/с]; - angles.m_velocityFactor – коэффициент скорости 0.0–1.0; - angles.m_accelerationFactor – коэффициент ускорения 0.0..1.0; - timeout_seconds – опциональный таймаут операции; - throw_error – опциональный флаг выбрасывания исключения	<pre>#include "pm_sdk/types.hpp" MoveAnglesM13Params angles(JointPositions(0.0, -1.57, 1.57, 0.0, 1.57, 0.0), // углы 6 суставов [рад] JointVelocities(0.0, 0.0, 0.0, 0.0, 0.0, 0.0) // скорости [рад/с]); angles.m_velocityFactor = 0.1; // коэффициент скорости angles.m_accelerationFactor = 0.1; // коэффициент ускорения manipulator.MoveToAngles(angles);</pre>
MoveToCoordinates	Движение по координатам	manipulator.MoveToCoordinates(coords);	- coords.position – объект Position(x, y, z);	<pre>#include "pm_sdk/types.hpp"</pre>

Наименование	Описание	Команда	Параметры	Пример
			- coords. orientation – объект Orientation(x, y, z, w); - coords.velocity_y_scaling_factor – коэффициент скорости; - coords.acceleration_scaling_factor – коэффициент ускорения; - coords.planner_type – тип планировщика, по умолчанию PlannerType::LIN; - coords.timeout_seconds – таймаут ожидания; - coords.throw_error – выбрасывать исключение при ошибке	<pre>MoveCoordinatesParams coords(Position(0.5, 0.0, 0.3), // x, y, z Orientation(0, 0, 0, 1.0), // x, y, z, w (кватернион) 0.1, // velocity_scaling_factor 0.1 // acceleration_scaling_factor); manipulator.MoveToCoordinates(coords);</pre>
MoveArc	Движение по дуге	manipulator.MoveArc(arc);	- arc.target – целевая точка дуги; - arc.center_arc – центр дуги; - arc.step – шаг между точками дуги; - arc.count_point_arc – число точек аппроксимации;	<pre>#include "pm_sdk/types.hpp" // Создаем целевую точку и центр дуги MoveArcParams::Target target(Position(0.25, -0.1, 0.25), // x, y, z целевой точки Orientation(0, 0, 0, 1) // ориентация);</pre>

Наименование	Описание	Команда	Параметры	Пример
			- arc.max_velocity_scaling_factor – максимальный масштаб скорости; - arc.max_acceleration_scaling_factor – максимальный масштаб ускорения	<pre> MoveArcParams::CenterArc center(Position(0.15, 0.0, 0.20), // x, y, z Orientation(0, 0, 0, 1)); // Создаем параметры дуги MoveArcParams arc(target, // целевая точка center, // центр дуги 0.05, // step - шаг между точками 50, // count_point_arc - количество точек 0.5, // max_velocity_scaling_factor 0.5 // max_acceleration_scaling_factor); manipulator.MoveArc(arc); </pre>
Стриминг				
SetServoTwistMode / SetServoPoseMode / SetServoJointJogMode	Переключени е режима стриминга: скорости (TWIST), поза (POSE), суставы (JOINT_JOG). Режим выбирается до отправки	manipulator.Set ServoTwistMod e(); // Для скорости manipulator.Set ServoPoseMode (); // Для позы manipulator.Set ServoJointJogM	—	// Режимы manipulator.SetServo TwistMode(); // Для скорости manipulator.SetServo PoseMode(); // Для позы manipulator.SetServoJ ointJogMode(); // Для суставов

Наименование	Описание	Команда	Параметры	Пример
	стриминговых команд	ode(); // Для суставов		
StreamJoint Positions	Стриминг суставов (6 осей)	manipulator.StreamJointPositions(positions, velocities);	- positions: JointPositions для 6 суставов; - velocities: JointVelocities для 6 суставов	// Включаем режим JOINT_JOG manipulator.SetServoJointJogMode(); JointPositionsM13 positions(0.5, 1.0, 0.8, 0.2, 0.1, 0.15); // углы 6 суставов JointVelocitiesM13 velocities(0.2, 0.1, 0.15, 0.05, 0.05, 0.05); // скорости manipulator.StreamJointPositions(positions, velocities);
StreamCartesian Velocities	Стриминг скоростей	manipulator.StreamCartesianVelocities(linear, angular);	- linear – объект LinearVelocity(x, y, z); - angular – объект AngularVelocity(rx, ry, rz)	// Включаем режим TWIST manipulator.SetServoTwistMode(); LinearVelocity linear{0.02, 0, 0}; AngularVelocity angular{0, 0, 0.01}; manipulator.StreamCartesianVelocities(linear, angular);
Stream Coordinates	Стриминг позы	manipulator.StreamCoordinates(pos, orient);	- pos – объект Position(x, y, z); - orient – объект Orientation(x, y, z, w)	// Включаем режим POSE manipulator.SetServoPoseMode(); Position pos(0.5, 0.0, 0.3); Orientation orient(0, 0, 0, 1); manipulator.StreamCoordinates(pos, orient);
Программы				

Наименование	Описание	Команда	Параметры	Пример
RunProgram	Запускает готовую программу на манипуляторе	manipulator.RunProgram("m13/default");	name – название программы;	manipulator.RunProgram("m13/default");
RunProgramJSON	Запускает программу, сформированную в виде JSON-сценария	manipulator.RunProgramJSON("program_1", program_json);	- name – название программы; - program – JSON-объект nlohmann::json с корневым ключом Root	<pre>nlohmann::json program_json = { /* ... */ }; manipulator.RunProgramJSON("program_1", program_json);</pre>
RunPythonProgram	Запускает Python-код прямо на манипуляторе	<pre>PythonProgram py("print('Hello from M13!') "); auto result = manipulator.RunPythonProgram (py);</pre>	- py.code – строка с Python-кодом. - py.env_name – имя виртуального окружения, опционально; - py.python_version – версия Python, опционально; - py.requirements – список зависимостей, опционально	<pre>PythonProgram py("print('Hello from M13!') "); auto result = manipulator.RunPythonProgram(py);</pre>
Остановка движений				
StopMovement	Остановка всего движения	manipulator.StopMovement();	–	manipulator.StopMovement();
Состояния и данные – методы для получения состояния манипулятора и подписок. Все методы возвращают nlohmann::json				
GetJointStates / GetHomePosition / GetCartesianCoordinates / GetManipulatorInfo	Синхронное получение данных	<pre>manipulator.GetJointStates(); manipulator.GetHomePosition();</pre>	–	<pre>// Текущие углы суставов (6 значений) auto joint_states = manipulator.GetJointStates();</pre>

Наименование	Описание	Команда	Параметры	Пример
GetCoordinateLimits / GetHardwareError		manipulator.Get CartesianCoordinates(); manipulator.Get ManipulatorInfo() manipulator.Get CoordinateLimits(); manipulator.Get HardwareError() ;		<pre>std::cout << "Углы суставов: " << joint_states.dump(2) << std::endl; // Домашняя позиция auto home_pos = manipulator.GetHome Position(); std::cout << "Домашняя позиция: " << home_pos.dump(2) << std::endl; // Текущие координаты X, Y, Z auto coords = manipulator.GetCartes ianCoordinates(); std::cout << "Координаты: " << coords.dump(2) << std::endl; // Информация о манипуляторе auto manip_info = manipulator.GetManip ulatorInfo(); std::cout << "Информация: " << manip_info.dump(2) << std::endl; // Ограничения координат auto limits = manipulator.GetCoord inateLimits(); std::cout << "Ограничения: " << limits.dump(2) << std::endl;</pre>

Наименование	Описание	Команда	Параметры	Пример
				<pre>// Ошибки оборудования auto errors = manipulator.GetHard wareError(); std::cout << "Ошибки: " << errors.dump(2) << std::endl;</pre>
SubscribeCartesianCoordinates / SubscribeJointStates / SubscribeGPIOStates / SubscribeCoordinateLimits / SubscribeManipulatorInfo / SubscribeHardwareError	Подписка на обновления	manipulator.SubscribeCartesianCoordinates manipulator.SubscribeJointStates manipulator.SubscribeGPIOStates manipulator.SubscribeCoordinateLimits manipulator.SubscribeManipulatorInfo manipulator.SubscribeHardwareError	—	<pre>// Подписка на координаты manipulator.Subscribe CartesianCoordinates([])(const nlohmann::json& data) { std::cout << "Координаты обновлены: " << data.dump() << std::endl; }); // Подписка на состояние суставов manipulator.Subscribe JointStates([])(const nlohmann::json& data) { std::cout << "Углы суставов обновлены: " << data.dump() << std::endl; }); // Подписка на состояние GPIO manipulator.Subscribe GPIOStates([])(const nlohmann::json& data) { std::cout << "GPIO обновлено: " << data.dump() << std::endl; });</pre>

Наименование	Описание	Команда	Параметры	Пример
				<pre>// Подписка на ограничения координат manipulator.Subscribe CoordinateLimits([](c onst nlohmann::json& data) { std::cout << "Ограничения: " << data.dump() << std::endl; }); // Подписка на информацию о манипуляторе manipulator.Subscribe ManipulatorInfo([](co nst nlohmann::json& data) { std::cout << "Информация о манипуляторе: " << data.dump() << std::endl; }); // Подписка на ошибки оборудования manipulator.Subscribe HardwareError([](con st nlohmann::json& data) { std::cout << "Ошибка: " << data.dump() << std::endl; });</pre>
UnsubscribeCartesianCoordinates / UnsubscribeJointStates / UnsubscribeGPI	Отписка от обновлений	manipulator.UnsubscribeCartesianCoordinates(); manipulator.UnsubscribeJointStates();	—	manipulator.UnsubscribeCartesianCoordinates(); manipulator.UnsubscribeJointStates();

Наименование	Описание	Команда	Параметры	Пример
OStates / UnsubscribeCoordinateLimits / UnsubscribeManipulatorInfo / UnsubscribeHardwareError		manipulator.UnsubscribeGPIOStates(); manipulator.UnsubscribeCoordinateLimits(); manipulator.UnsubscribeManipulatorInfo(); manipulator.UnsubscribeHardwareError();		manipulator.UnsubscribeGPIOStates(); manipulator.UnsubscribeCoordinateLimits(); manipulator.UnsubscribeManipulatorInfo(); manipulator.UnsubscribeHardwareError();
Важно! Для асинхронного получения данных программа должна продолжать работать (например, иметь цикл или ожидание), чтобы успевать обрабатывать входящие сообщения. Подписки работают в фоновом потоке и вызывают ваш callback при каждом обновлении данных на манипуляторе				
SetJointLimits	Установка пределов суставов. Ошибки возвращаются в формате JSON: {"type": id, "message": "..."}	manipulator.SetJointLimits(limits);	—	// Установка ограничений скорости и ускорения для суставов std::vector limits = { JointLimit("shoulder_pan_joint", 2.0, 3.0), JointLimit("shoulder_lift_joint", 1.5, 2.5), JointLimit("elbow_joint", 2.0, 3.0), JointLimit("wrist_1_joint", 2.5, 3.5), JointLimit("wrist_2_joint", 2.5, 3.5), JointLimit("wrist_3_joint", 2.5, 3.5) }; manipulator.SetJointLimits(limits);

Наименование	Описание	Команда	Параметры	Пример
				// Получение текущих ограничений auto current_limits = manipulator.GetJointLimits();

4.3.2 Управление GPIO

GPIO (General Purpose Input/Output) позволяет управлять цифровыми входами и выходами манипулятора. Это полезно для взаимодействия с внешними устройствами, датчиками, светодиодами и другими периферийными компонентами.

Управление GPIO описано в таблице 53.

Таблица 53 – Управление GPIO

Наименование	Описание	Команда	Пример
Основные методы			
GPIOPin	Конфигурация GPIO пина	bool result = manipulator.ConfigureGPIOPin("gadget_digital_outputs_dout2", false);	bool result = manipulator.ConfigureGPIOPin("gadget_digital_outputs_dout2", false); // false = выход, true = вход
GetGPIO	Чтение значения GPIO	manipulator.GetGPIO("flange/digital_ports/digital_inputs");	nlohmann::json value = manipulator.GetGPIO("flange/digital_ports/digital_inputs");
SetGPIO	Запись значения GPIO	SetGPIO("flange/digital_ports/digital_outputs", 1);	manipulator.SetGPIO("flange/digital_ports/digital_outputs", 1);
Работа с цифровыми портами			
WriteDigitalOutput	Запись цифрового выхода	manipulator.WriteDigitalOutput(1, true);	manipulator.WriteDigitalOutput(1, true); // Канал 1, включить manipulator.WriteDigitalOutput(2, false); // Канал 2, выключить

Наименование	Описание	Команда	Пример
			manipulator.WriteDigitalOutput(3, true); // Канал 3, ВКЛЮЧИТЬ

4.3.2.1 Примеры использования GPIO

Примеры использования GPIO описаны в таблице 54.

Таблица 54 – Управление GPIO

Наименование	Пример
Конфигурация GPIO пина	<pre>bool result = manipulator.ConfigureGPIOPin("gadget_digital_outputs_dout2", false); if (result) { std::cout << "Пин gadget_digital_outputs_dout2 успешно skonфигурирован" << std::endl; } else { std::cout << "Ошибка конфигурации пина" << std::endl; }</pre>
Чтение и запись GPIO	<pre>nlohmann::json input_value = manipulator.GetGPIO("flange/digital_ports/digital_inputs"); std::cout << "Состояние входа: " << input_value << std::endl; // Запись в порт manipulator.SetGPIO("flange/digital_ports/digital_outputs", 1); // Проверка записи nlohmann::json check_value = manipulator.GetGPIO("flange/digital_ports/digital_outputs"); std::cout << "Состояние выхода: " << check_value << std::endl;</pre>
Комплексный пример с автоматизацией	<pre>#include #include void AutomatedProcess(M13& manipulator, Position target_pos, Orientation target_orient) { // Конфигурация пинов manipulator.ConfigureGPIOPin("gadget_digital_outputs_dout1", false); manipulator.ConfigureGPIOPin("gadget_digital_outputs_dout2", true);</pre>

	<pre> std::this_thread::sleep_for(std::chrono::milliseconds(500)); // Проверка наличия объекта nlohmann::json sensor_value = manipulator.GetGPIO("flange/digital_ports/digital_inputs"); if (sensor_value == 1) { std::cout << "Объект обнаружен" << std::endl; // Перемещение к объекту MoveCoordinatesParams coords(target_pos, target_orient, 0.1, 0.1); manipulator.MoveToCoordinates(coords); // Активация захвата через цифровой выход manipulator.WriteDigitalOutput(1, true); std::this_thread::sleep_for(std::chrono::seconds(1)); // Деактивация устройства manipulator.WriteDigitalOutput(1, false); std::cout << "Операция завершена" << std::endl; } else { std::cout << "Объект не обнаружен" << std::endl; } } </pre>
--	--

4.3.2.2 Особенности работы с GPIO на M13

Предварительная конфигурация: перед использованием GPIO пинов необходимо выполнить их конфигурацию с помощью `ConfigureGPIOPin`.

Цифровые выходы: управляются по номерам каналов с булевыми значениями.

Важно! При работе с GPIO портами соблюдайте требования по напряжению и току. Неправильное использование может повредить оборудование.

4.3.3 Примеры использования команд

В составе SDK предоставлен демонстрационный проект в каталоге `demo`. Он показывает использование основных команд управления манипулятором и дополнительными модулями.

Для сборки демо:

```
cd demo
mkdir build
cd build
cmake ..
make
./sdk_demo
```

4.3.4 Modbus и RS-485

Манипулятор Promobot M13 поддерживает взаимодействие с внешними устройствами по интерфейсу RS-485 с использованием протокола Modbus RTU. Это позволяет подключать датчики, контроллеры, приводы и другое промышленное оборудование.

4.3.4.1 Архитектура и терминология

Взаимодействие строится вокруг следующих концепций:

- Устройство (ModbusDevice): логическое представление внешнего Modbus-устройства, подключенного к RS-485. Каждое устройство идентифицируется своим slave_id.
- Сигнал (ModbusSignal): описывает запрос к устройству: функцию (чтение/запись), адрес регистра, количество элементов и интервал опроса (poll_ms).
- Параметры шины (ModbusParameters): глобальные настройки RS-485 интерфейса (скорость, биты данных, четность и т.д.).
- Кадры (ModbusFrame): необработанные байты, которыми обменивается манипулятор. SDK позволяет подписываться на отправленные (TX) и полученные (RX) кадры для отладки.

SDK предоставляет как высокоуровневый API для работы с устройствами и сигналами, так и низкоуровневый для отправки сырых данных.

4.3.4.2

Примеры использования Modbus и RS-485

Примеры использования Modbus и RS-485 описаны в таблице 55.

Таблица 55 – Управление Modbus и RS-485

Наименование	Описание	Пример
Управление параметрами шины RS-485	Перед началом работы необходимо настроить параметры последовательно го интерфейса. Это делается один раз или при изменении конфигурации сети	<pre>// Подготовка параметров RS-485 nlohmann::json rs485_params = { {"modbus.baudrate", 9600}, // Скорость, бод {"modbus.data_bits", 8}, // Биты данных {"modbus.parity", "none"}, // Четность: "none", "even", "odd" {"modbus.stop_bits", 1.0}, // Стоп-биты {"modbus.response_timeout_ms", 500}, // Таймаут "ответа, мс" {"modbus.poll_period_ms", 100} // Период опроса "сигналов, мс" }; // Применение параметров manipulator.ModbusSetParameters(rs485_params);</pre>
Управление опросом	После настройки устройств и сигналов, опрос можно запустить или остановить. Это глобальная команда для всех добавленных устройств	<pre>// Включить циклический опрос сигналов manipulator.ModbusTogglePolling(true); // Отключить опрос manipulator.ModbusTogglePolling(false);</pre>
Конфигурация Modbus-устройств и сигналов	Устройство описывается структурой ModbusDevice, которая содержит slave_id, имя и список сигналов (ModbusSignal)	<pre>// Создание сигнала чтения 2-х удерживающих регистров с адреса 1000 ModbusSignal read_signal(ModbusFunction::READ_HOLDING_REGISTERS, // "функция чтения" 1000, // начальный адрес 2, // количество регистров 1000 // интервал опроса (мс)); // Создание сигнала записи в один регистр (аналоговый выход) ModbusSignal write_signal = ModbusSignal::CreateWriteSingleRegister(</pre>

Наименование	Описание	Пример
		<pre> 2000, // адрес регистра 1234, // значение для записи 0 // poll_ms = 0 (не опрашивать циклически)); // Сборка устройства с slave_id = 1 ModbusDevice device(1, "TemperatureSensor"); device.signals.push_back(read_signal); device.signals.push_back(write_signal); // Добавление устройства на манипулятор auto result = manipulator.ModbusAddDevice(device); std::cout << "Device added: " << result.dump() << std::endl; </pre>
Типы функций Modbus: SDK поддерживает основные функции: чтение (1,2,3,4), запись (5,6,15,16), маскированную запись (22) и чтение/запись (23). Используйте соответствующие значения из перечисления ModbusFunction		
Управление устройствами	SDK позволяет получать список всех настроенных устройств и удалять их	<pre> // Получение списка всех устройств std::vector<ModbusDevice> devices = manipulator.ModbusListDevices(); for (const auto& dev : devices) { std::cout << "Device ID: " << (int)dev.slave_id << ", Name: " << dev.name << std::endl; std::cout << " Signals count: " << dev.signals.size() << std::endl; } // Удаление устройства по slave_id manipulator.ModbusRemoveDevice(1); </pre>
Отправка произвольных данных по RS-485	Для максимальной гибкости или работы с нестандартными протоколами можно отправлять сырой массив байт. Можно дожидаться ответа от устройства или отправить асинхронно	<pre> // Подготовка данных для запроса (например, Modbus RTU кадр) std::vector<uint8_t> raw_data = { 0x01, // slave ID 0x03, // function code (read holding registers) 0x03, 0xE8, // start address (1000) 0x00, 0x02, // quantity of registers (2) 0xC5, 0x7E // CRC }; // Отправка и ожидание ответа (таймаут 500 мс) RS485RawResponse response = manipulator.RS485SendRaw(raw_data, // данные </pre>

Наименование	Описание	Пример
		<pre> 1, // device_id (логический идентификатор, не slave_id) true, // wait_response 500 // response_timeout_ms); if (response.success) { std::cout << "Получен ответ: "; for (auto byte : response.data) { std::cout << std::hex << (int)byte << " "; } std::cout << std::endl; } else { std::cout << "Ошибка: " << response.error << std::endl; } </pre>
Мониторинг Modbus-трафика	Для отладки можно подписаться на все исходящие (TX) и входящие (RX) кадры. SDK предоставляет для этого колбэки, которые получают структуру ModbusFrame с временной меткой, направлением и данными	<pre> // Подписка на исходящие кадры manipulator.SubscribeModbusTx([](const ModbusFrame& frame) { std::cout << " [TX] " << frame.ToString() << std::endl; }); // Подписка на входящие кадры manipulator.SubscribeModbusRx([](const ModbusFrame& frame) { std::cout << " [RX] " << frame.ToString() << std::endl; }); // Подписка на ошибки Modbus (например, ошибка CRC, таймаут) manipulator.SubscribeModbusError([](const std::string& error) { std::cerr << "Modbus Error: " << error << std::endl; }); // Отписка от трафика manipulator.UnsubscribeModbusTx(); manipulator.UnsubscribeModbusRx(); </pre>
Структура ModbusFrame предоставляет удобные методы для вывода: - frame.ToString() – возвращает строку с расшифровкой кадра (например, tx Slave 1: Read Holding Registers, Addr=1000, Count=2 [01 03 03 E8 00 02 C5 7E]); - frame.ToHex() – возвращает шестнадцатеричное представление данных. Важно для промышленного использования:		

Наименование	Описание	Пример
- убедитесь, что физическое подключение RS-485 выполнено правильно (провода A/A+, B/B-, GND); - Скорость, биты данных и четность должны совпадать для манипулятора и всех подключенных устройств; - интервалы опроса (poll_ms) и таймауты должны быть настроены адекватно времени отклика устройств, чтобы не перегружать шину; - для длительных сеансов рекомендуется использовать синхронные вызовы с обработкой исключений для надежности.		
Полный пример: опрос датчика температуры	В этом примере демонстрируется настройка параметров шины, добавление устройства с сигналом чтения температуры и последующий ручной опрос	<pre> #include "pm_sdk/manipulator.hpp" #include <iostream> #include <thread> int main() { using namespace pb_sdk; M13 robot("192.168.88.182", "sdk_client", "13", "14"); robot.Connect(); robot.GetControl(); // 1. Настройка RS-485 nlohmann::json params = { {"modbus.baudrate", 19200}, {"modbus.data_bits", 8}, {"modbus.parity", "none"}, {"modbus.stop_bits", 1}, {"modbus.response_timeout_ms", 300} }; robot.ModbusSetParameters(params); // 2. Создание сигнала (чтение 1 регистра с адреса 300) ModbusSignal temp_signal(ModbusFunction::READ_HOLDING_REGISTERS, 300, // адрес 1, // количество регистров 1000 // poll_ms (автоматический опрос каждую секунду)); // 3. Создание устройства и добавление сигнала ModbusDevice temp_device(10, "TemperatureSensor"); temp_device.signals.push_back(temp_signal); robot.ModbusAddDevice(temp_device); // 4. Включение циклического опроса </pre>

Наименование	Описание	Пример
		<pre> robot.ModbusTogglePolling(true); // 5. Подписка на полученные значения (через сигналы) // Примечание: полученные данные приходят через топики /modbus/... // Для этого примера подпишемся на трафик RX для наблюдения robot.SubscribeModbusRx([])(const ModbusFrame& frame) { // Парсим ответ, если это ответ от slave 10 с функцией 3 if (frame.data.size() >= 7 && frame.data[0] == 10 && frame.data[1] == 3) { // Объединяем два байта в значение регистра int temp_raw = (frame.data[3] << 8) frame.data[4]; float temperature = temp_raw / 10.0f; // Предположим, что датчик отдает десятые доли градуса std::cout << "Temperature: " << temperature << " °C" << std::endl; } }); // 6. Основной цикл std::this_thread::sleep_for(std::chrono::seconds(10)); // 7. Очистка robot.ModbusTogglePolling(false); robot.ModbusRemoveDevice(10); robot.UnsubscribeModbusRx(); return 0; } </pre>

4.4 Заключение

При возникновении проблем используйте отладочный вывод:

```

#include "pm_sdk/debug_logger.hpp"

// Включение всех логов

pb_sdk::DebugLogger::SetEnable(pb_sdk::LogEntity::ALL, true);

```

```
// Включение только SDK логов
```

```
pb_sdk::DebugLogger::SetEnable(pb_sdk::LogEntity::SDK, true);
```

```
// Включение только MQTT логов
```

```
pb_sdk::DebugLogger::SetEnable(pb_sdk::LogEntity::MQTT, true);
```

При возникновении подробных логов обратитесь в сервисный центр предприятия-изготовителя.

Рекомендация: для повышения надежности исполнения все вызовы команд следует оборачивать в конструкцию try/catch, чтобы корректно обрабатывать возможные ошибки.