

«ТОО «НПО «Группа Компаний «DOSTI»

ROTRICS DEX ARM

Rotrics Dex ARM – многофункциональный универсальный робот-манипулятор, с возможностью гравировки по дереву, рисованию и 3D-печати. С помощью языка программирования Python достигается возможность автоматизации промышленных процессов с использованием машинного зрения



Мольганов А.А.
18.7.2022

«Rotrics DexArm. Энциклопедия пользователя», 2023. – 50 с.: ил.

Данная книга знакомит читателя с роботом Rotrics DexArm, который помогает обучаться в различных дисциплинах и компетенциях – программирование, робототехника, мехатроника, 3D-печать и электроника.

Издание предназначено для школьников, студентов, разработчиков и преподавателей, желающих обучиться программированию, робототехнике, мехатронике и моделированию, а также обучать других с помощью данной книги.

Благодаря модульной конструкции роботизированная рука Rotrics может легко переключаться между различными функциями, такими как писание и рисование, лазерная гравировка, 3D-печать и сбор, и размещение.

Благодаря простому в использовании программному обеспечению Rotrics предоставляет вам интуитивно понятный и интерактивный способ управления и воплощения ваших идей всего за несколько кликов.

Компания Rotrics также предоставляет разработчикам свободный API на нескольких языках, таких как Python и G-Code (Marlin). Вы можете легко управлять роботом с помощью языков программирования и SDK, а также с легкостью интегрировать в свой существующий проект.

СОДЕРЖАНИЕ

	стр.
Предисловие	3
Глава 1. Механика робота	4
1.1. Первоначальная настройка робота	5
1.2. Конструкция робота	11
1.3. Пространственная модель перемещения робота	13
Глава 2. Программирование робота	17
2.1. Программирование на языке Scratch	18
2.1.1. Модуль «Держатель»	18
2.1.2. Модуль «2.5W Лазер»	20
2.1.3. Модуль «3D-принтер»	23
2.1.4. Модуль «Пневматика: Мягкий держатель»	25
2.1.5. Модуль «Пневматика: Твердый держатель»	27
2.1.6. Модуль «Механика: Конвейерная лента»	29
2.1.7. Модуль «Механика: Направляющая»	31
2.1.8. Основы программирования на Scratch	33
2.2. Программирование на языке Python	41
2.2.1. Модуль «2.5W Лазер»	41
2.2.2. Модуль «Пневматика: Мягкий держатель»	42
2.2.3. Модуль «Пневматика: Твердый держатель»	
2.2.4. Модуль «Механика: Конвейерная лента»	
2.2.5. Модуль «Механика: Направляющая»	
2.2.6. Основы промышленной автоматизации	
Эпилог	
Дополнительные материалы	
Список использованных материалов	
Список использованных графических материалов	

ПРЕДИСЛОВИЕ

Робототехника – очень сложная наука, находящаяся на стыке математики, физики и информатики. Для реализации самого простого робота надо знать несколько законов из электротехники, иметь опыт программирования микроконтроллеров и знать математические основы движения робота в пространстве.

Но с помощью Rotrics Dex ARM занятия по робототехнике станут увлекательнее, проще и намного разнообразнее в плане создания творческой атмосферы для личных и образовательных сфер.

Rotrics Dex ARM снижает порог вхождения в робототехнику, с помощью упрощения некоторых незначительных, но одного из самых трудоемких процессов – программирования микроконтроллера.

Высокоточный манипулятор программируется на языке Scratch, понятным даже для детей школьного возраста с помощью простых и удобных визуальных конструкций. Язык Scratch является полностью программируемым, то есть на нем можно реализовать почти все программные теории и методы, которые применяется при разработке полнофункционального программного обеспечения.

Но самое главное преимущество робота – модульность. Робот был сконструирован с упором на модульность – когда пользователь можно поменять любой модуль на абсолютно другой. Это действительно очень просто и увлекательно!

Данная книга расширяет горизонты применения робота, и использует в своем арсенале такой мощный, гибкий и простой в использовании язык программирования – Python. С помощью данного языка программирования можно автоматизировать робот и придумать самые различные творческие сценарии.

Например – автоматизация производства. Вы никогда не сортировали ящики по цвету? Это просто, круто и что самое главное – доступно для всех. И все это в этой книге.

Удачного пути!

ГЛАВА 1. МЕХАНИКА РОБОТА

Rotrics Dex ARM – универсальный настольный робот-манипулятор для обучения различным компетенциям и направлениям – программирование, робототехника, мехатроника, электроника и промышленная автоматизация.

Благодаря модульной конструкции роботизированная рука может легко переключаться между различными функциями, такими как роспись, рисование, лазерная гравировка, 3D-печать, сбор, перемещение и сортировка предметов.

Благодаря простому в использовании программному обеспечению, Rotrics Dex ARM предоставляет вам интуитивно понятный и интерактивный способ управления и воплощения ваших идей всего за несколько кликов мыши.



Рисунок 1.0.1 – Общий вид Rotrics Dex ARM



Рисунок 1.0.2 – Дополнительные элементы для Rotrics Dex ARM

§ 1.1. Первоначальная настройка робота

Перед использованием робота в учебных и творческих целях, следует провести первоначальную базовую настройку робота для правильной работоспособности и исключения аппаратных и программных неполадок. Перед работой с роботом, необходимо ознакомиться со спецификациями робота. Они включают в себя следующие характеристики и массогабаритные свойства:

- Точность операций робота: ~ 0.05 мм;
- Размер упаковки робота: 220x155x160 мм;
- Размер полностью собранного робота: 175x128x315мм;
- Вес: ~ 2.4* кг (Вес робота указан без дополнительных модулей)

Робот в стандартной комплектации включает в себя следующие составные компоненты:

1. Rotrics Dex Arm;
2. Мобильное пульт управления (Touchscreen);
3. Кабеля и сервисные инструкции;
4. Модуль «Мягкий держатель»;
5. Модуль «Твердый держатель»;
6. Модуль «2.5W Лазер»;
7. Модуль «3D-принтер»;

Конструкция робота представляет из себя 4DOF-манипулятор (Манипулятор с поддержкой 4-х зон пространственного перемещения). Сменные модули позволяют переключаться с одной операции на другую без дополнительной аппаратной и программной настройки робота или управляющего компьютера.

Благодаря открытому API, роботом можно управлять с помощью языка программирования Python, и не использовать программу для управления роботом от Rotrics, под названием «Rotrics Studio». API представляет собой набор функций и переменных, уже определенных в программе, необходимых для управления роботом.

Алгоритм настройки робота состоит из следующих действий:

- Извлеките белую коробку с роботом из транспортировочной упаковки;
- Извлеките черную матовую упаковку с роботом из белой коробки;
- Переведите защелки на черной матовой упаковке с роботом в положение «Open», затем извлеките и поставьте робот на заранее подготовленную чистую, ровную и матовую поверхность;



Рисунок 1.1.1 – Распаковка робота из транспортировочной упаковки

- Далее, скачайте программное обеспечение «Rotrics Studio» с официального сайта – <https://rotrics.com/pages/downloads>;

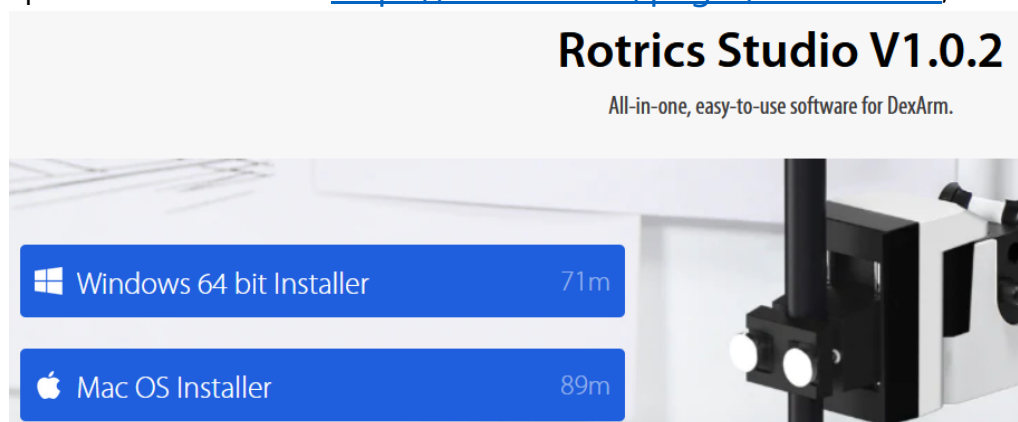


Рисунок 1.1.2 – Страница загрузки Rotrics Studio

- После установки необходимого программного обеспечения «Rotrics Studio», подключите к роботу питание с помощью кабеля

«Power Adapter». Порт USB Type-C предназначен для подключения к роботу, а порт USB Type-A для подключения к компьютеру.

ВНИМАНИЕ! Не используйте двойной кабель USB Type-C для подключения к персональному компьютеру. Данный кабель используется только для подключения дополнительных модулей к роботу;

- После подключения робота к персональному компьютеру, убедитесь, что подключение прошло успешно. Перейдите в меню «Пуск», и найдите утилиту «Управление компьютером», как показано ниже (Рис 1.1.2):

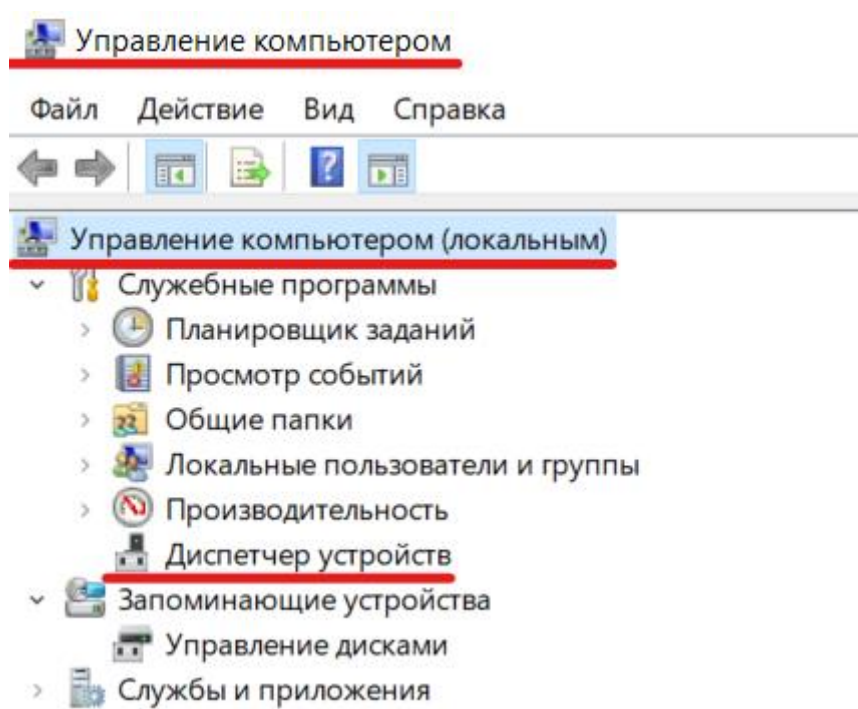


Рисунок 1.1.3 – Диспетчер устройств

- В открывшемся окне, найдите подменю «Диспетчер устройств»;
- В открывшемся подменю будет показан подключенный робот с номером порта к которому он подключен. (COMN, где N – номер порта);

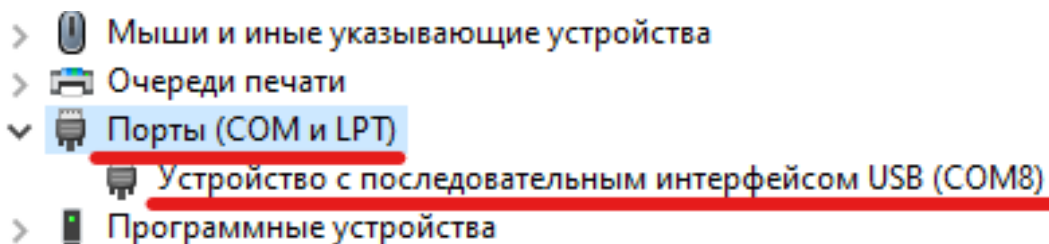


Рисунок 1.1.4 – Статус подключенного робота

- После успешного подключения робота, откройте установленную программу «Rotrics Studio» и на верхней панели инструментов выберите кнопку, изображенную ниже:

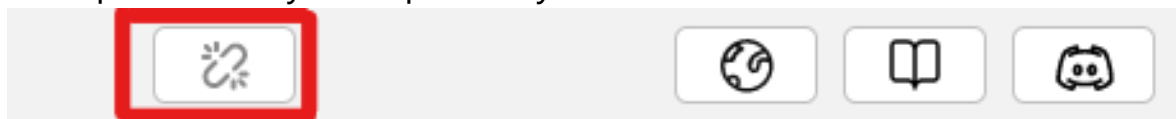


Рисунок 1.1.5 – Подключение робота в Rotrics Studio

- В открывшемся диалоговом окне, в поле «Status», выберите доступный в данный момент COM-порт для соединения с роботом:

Connect DexArm

Status:

Choose a port ▼

Connect

Рисунок 1.1.6 – Выбор COM-порта для подключения

- После выбранного COM-порт для соединения, нажмите кнопку «Connect» для подключения:

Connect DexArm

Status: Disconnected

COM8 ▼

Connect

Рисунок 1.1.7 – Статус подключения

- После успешного подключения к роботу, поле «Status» изменится с «Disconnected» на «Connected»;

- В главном меню «Rotrics Studio», в правой части найдите панель управления, и нажмите кнопку «Home» для перемещения робота в начальную позицию.



Рисунок 1.1.8 – Управление подключенным роботом

После выполнения всех пунктов данного параграфа, попробуйте начать взаимодействовать с роботом через панель управления, а также с помощью «Мобильного пульта управления». Для дальнейшей работы, необходимо установить дополнительные программные компоненты. Для этого требуется сделать следующие действия:

- Ознакомьтесь со следующими системными требованиями:

Таблица 1.1.9 – Системные требования

Характеристика	Рекомендуемое
Операционная система	Windows 7/8/8.1/10/11, MacOS 10.16+, Linux 5.11+
Python (Язык программирования)	3.10+ (Рекомендуется версия Python не ниже 3.6, однако результат может отличаться, так как все действия проводились на тестовом стенде с Python 3.10, обратите внимание, что Python 3.10+ не поддерживает Windows 7/8/8.1)
Git (Система контроля версий)	2.20+
IDE (Интегрированная среда разработки)	Microsoft Visual Studio Code JetBrains PyCharm

- После установки необходимого программного обеспечения (Python, Git, IDE), скачайте следующий файл по ссылке: <https://bootstrap.pypa.io/get-pip.py>. Данный файл поместите в директорию Локального Диска (C:).

- Откройте терминал, командную строку или любой другой терминал с правами администратора и перейдите в Локальный Диск (C:), а затем выполните следующую команду:

```
python get-pip.py
```

- После установки служебной утилиты pip, установите дополнительные пакеты для корректной работы:

```
pip install opencv-python  
pip install pyyaml  
pip install pyserial
```

- С помощью утилиты Git клонируйте репозиторий по следующему адресу <https://github.com/0x600/dex-arm>. Репозиторий имеет следующую структуру:

```
archive – архивные файлы  
book – учебное пособие  
docs – дополнительные материалы  
pic – изображения  
src – программный код  
CHANGELOG – история изменений  
LICENSE – лицензия  
README – о проекте
```

- После клонирования репозитория, откройте папку с репозиторием в IDE, и в открывшемся окне, заходим в директорию src. В этой директории содержится весь программный код. На это полноценная первоначальная настройка робота и программного окружения завершена.

§ 1.2. Конструкция робота

Rotrics Dex ARM является 4DOF-манипулятором (с англ. – манипулятор с поддержкой четырех зон пространственного перемещения). Благодаря сменным модулям, появляется возможность быстро и «на лету» переключаться с одной операции на другую без дополнительной аппаратной, программной или любой другой предварительной настройки робота или управляющего компьютера.

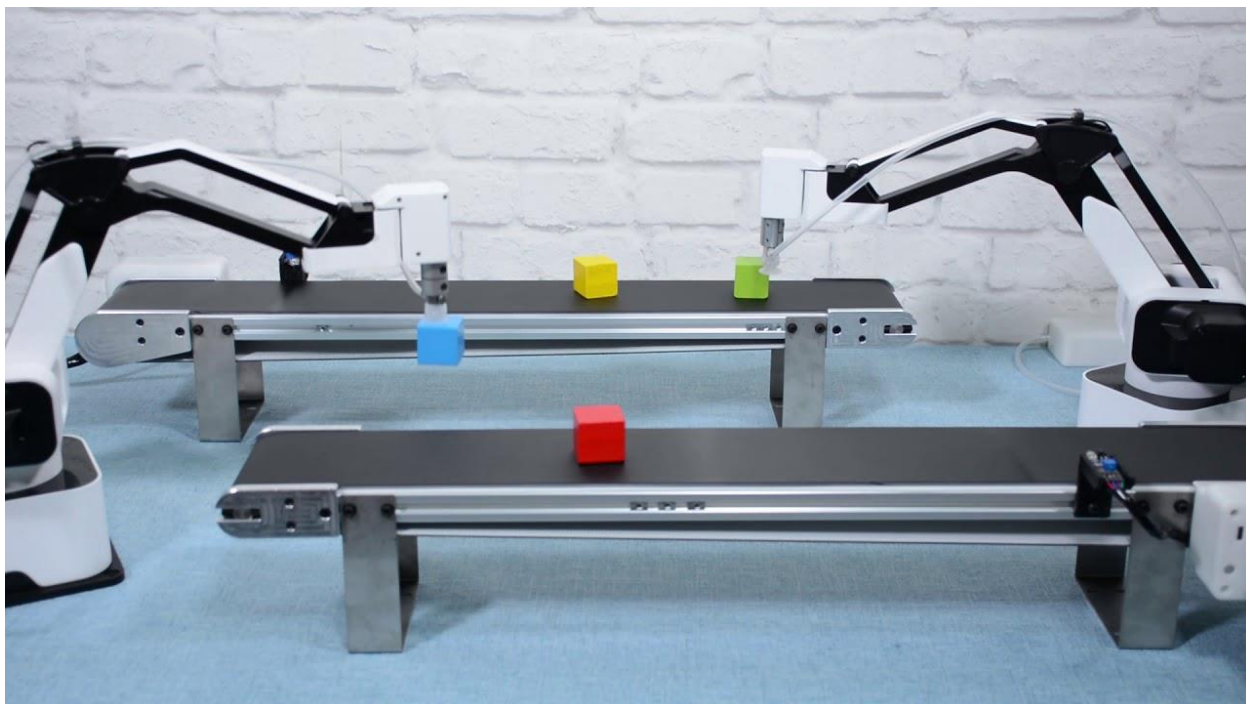


Рисунок 1.2.1 – Программирование робота

Благодаря открытому API, роботом можно управлять с помощью языка программирования Python, и не использовать программу для управления роботом от Rotrics, под названием «Rotrics Studio». API представляет собой набор функций и переменных, уже определенных в программе, необходимых для управления роботом.

Программирование робота на языках Python и C++ сводится к написанию программ-функций, которые управляют роботом с помощью команд, исходящих от управляющего компьютера (пользователя). Роботом управляет микроконтроллер от компании STMicroelectronics серии STM32F4.

Благодаря модульному подключению, расположенному на манипуляторе, появляется возможность использовать различные

сменные модули без необходимой дополнительной аппаратной и программной пред настройки.

- Модуль «Держатель». Модуль предназначен для создания графических элементов с помощью пишущего элемента (ручка, карандаш, фломастер) на офисной бумаге любого стандарта, при этом размер рисунка ограничен возможностями робота, чаще всего это рисунок размером 10 см на 15 см;
- Модуль «2.5W Лазер». Модуль предназначен для создания графических элементов с помощью лазера 2.5W-мощности на деревянной или пластиковой подложке, при этом размер рисунка ограничен возможностями робота, чаще всего это рисунок размером 10 см на 15 см;
- Модуль «3D-принтер». Модуль предназначен для создания твердотельных объектов на основе смоделированной 3D-модели в САПР любой сложности поддерживающий вывод файлов в формате .stl для удобной печати объектов.
- Модуль «Пневматика: Мягкий держатель». Модуль предназначен для перемещения объектов весом не более 150 грамм, на любое расстояние, при этом расстояние ограничено возможностями робота. Главное отличие — это использование одного воздушного канала и одной присоски;
- Модуль «Пневматика: Твердый держатель». Модуль предназначен для перемещения объектов весом не более 150 грамм, на любое расстояние, при этом расстояние ограничено возможностями робота. Главное отличие — это использование трех воздушных каналов и трех клешней;
- Модуль «Механика: Конвейерная лента». Модуль предназначен для перемещения объектов с помощью конвейерной ленты, поддерживающей плавное изменение скорости, остановку и движение в различных направлениях (вперед и назад);
- Модуль «Механика: Направляющая». Модуль предназначен для перемещения робота на значительные расстояния с помощью, направляющей рельсы и моторов;

§ 1.3. Пространственная модель перемещения робота

Изучив конструкцию и проведя первоначальную настройку робота, давайте рассмотрим основные правила задания движения робота по координатным осям OX , OY , OZ . На Рис. 1.3.1, изображены координатные оси, а именно:

1. Ось X
2. Ось Y
3. Ось Z
4. Вектор U
5. Углы α , β и γ

Вектор U – это точка в пространстве которая была задана роботу с помощью трех осей, например: (100, 300, -75). Данные координаты расшифровываются так:

1. 100 – точка на координатной оси X
2. 300 – точка на координатной оси Y
3. -75 – точка на координатной оси Z

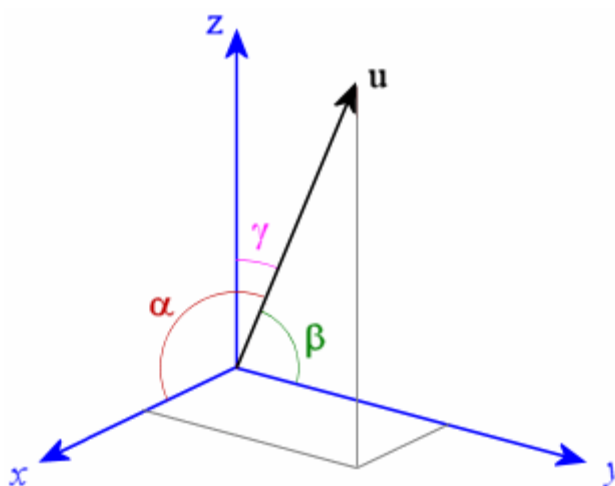


Рисунок 1.3.1 – Координатные векторы робота Rotrics Dex Arm

Для следующего примера установите модуль «Держатель» и надежно закрепите в данном модуле канцелярскую ручку. С помощью данного модуля будет показана логика ориентирования робота в пространстве с помощью осевых координат.

```
from pydexarm import Dexarm
dexarm = Dexarm("COM8") # Подключение к Windows
# device = Dexarm("/dev/tty.usbmodem3086337A34381") #
Подключение к MacOS/Linux

dexarm.go_home()
dexarm.move_to(135, 225, -95) # A
dexarm.move_to(120, 210, -95) # B
dexarm.move_to(105, 225, -95) # C
dexarm.move_to(120, 240, -95) # D
dexarm.move_to(135, 225, -95) # D to A
dexarm.go_home ()
```

Рассмотрим повнимательнее Листинг 1.3.2:

1. `from pydexarm import Dexarm` - включает в основной код библиотеку-интерфейс для общения с роботом при помощи связи между Python и G-code;
2. `dexarm = Dexarm("COM8")` - отправляет скомпилированный код в робот на указанный COM-порт, а именно COM8;
3. `dexarm.go_home()` - дает команду роботу вернуться в начальное или исходное положение. Функция не требует аргументов, так как робот настроен на начальное положение в (0; 0; 0);
4. `dexarm.move_to(135, 225, -95)` - определяет координаты по трем осям, и отправляет роботу команду для перемещения манипулятор робота в данную точку, а именно – по оси X – 135; по оси Y – 225; по оси Z – -95;

Математическое объяснение листинга 1.3.2 и операций робота:

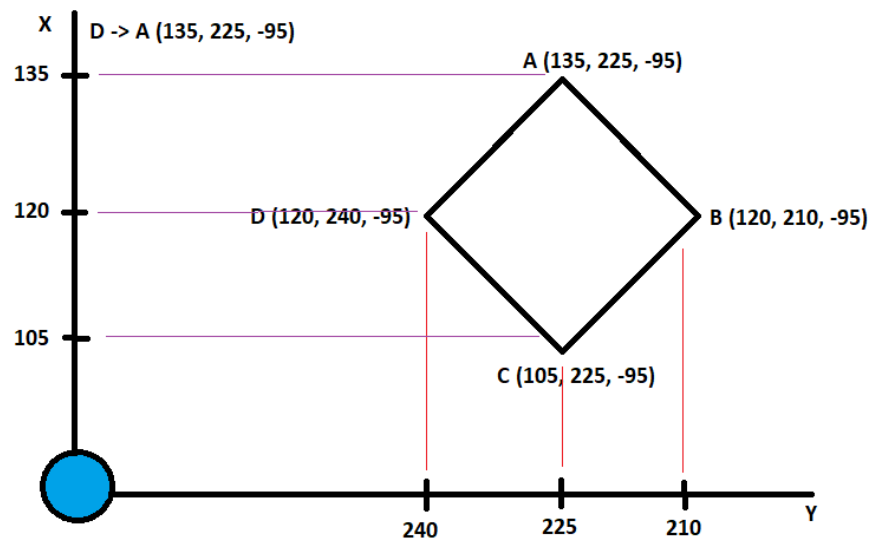


Рисунок 1.3.3 – Математическое объяснение операций робота в листинге 1.3.2

- Для того чтобы нарисовать квадрат или любую другую фигуру на бумаге, дереве или другом материале, следует помнить то, что для модуля «Ручка» установлено минимальная высота, а именно -95 (минус 95), а для модуля «Лазер» минимальная высота составляет -70 (минус 70);
- Координатные оси распределены следующим образом:

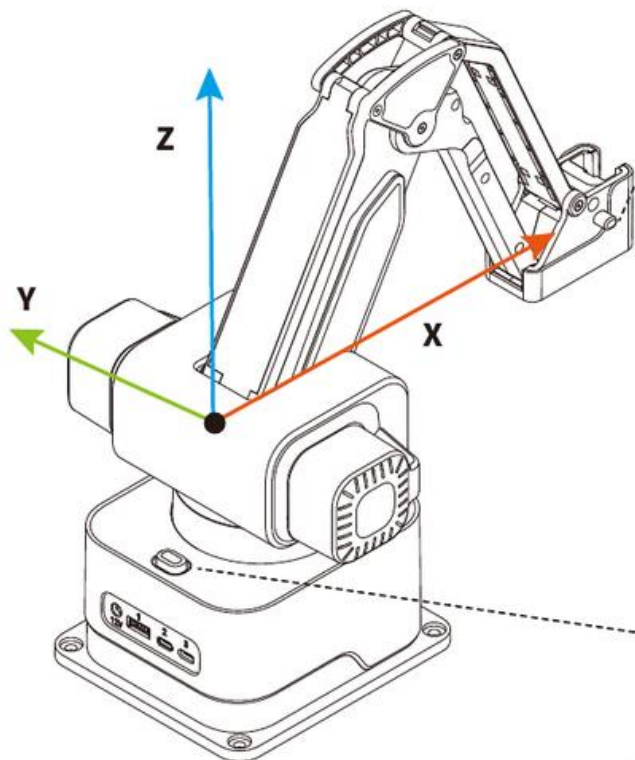


Рисунок 1.3.4 – Координатные оси робота в 3D-проекции

- Таким образом, оси X и Y поменялись местами в отличие от математической модели осей, где вверху ось Y, а внизу ось X.
- При работе с роботом, следует не забывать про конструктивные ограничения, а именно:

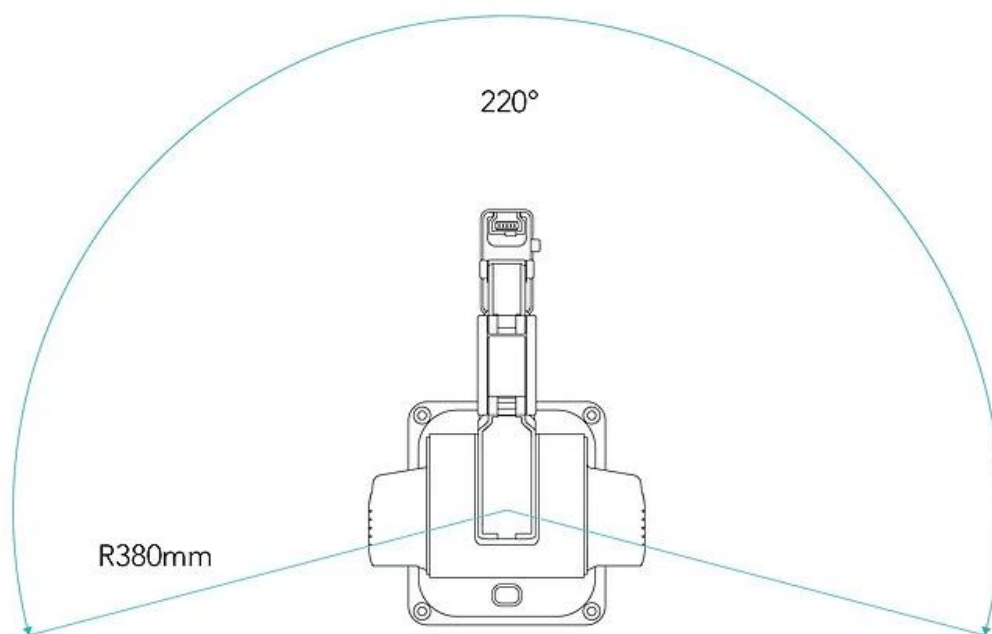


Рисунок 1.3.5 – Максимальные угловые и радиальные операции доступные для робота

- Максимальный угол для движения по оси X и Y составляет 220 градусов;
- При этом максимальный радиус для операций составляет 380 мм;
- Причем, координаты в нашей программе мы принимаем за 1 мм, то есть (100, 300, -75), означает что робот по оси X перейдет в точку 100 (100 мм от начала координат 0; 0; 0), по оси Y в точку 300 (300 мм от начала координат 0; 0; 0) и по оси Z в точку -75 (-75 мм от начала координат 0; 0; 0);

ГЛАВА 2. ПРОГРАММИРОВАНИЕ РОБОТА

Программирование робота доступно с помощью двух языков программирования – графический язык Scratch и высокоуровневый язык Python.

Язык графического программирования Scratch был спроектирован для детей подросткового возраста для обучения их основам программирования и применения различных информационно-коммуникационных технологий как в образовании, так и в качестве хобби. Язык имеет широкую поддержку различных устройств, например воспроизведение звуковых композиций, работу с сенсорами и двигателями, а также поддерживает различные расширения с различным практическим уклоном.

Язык программирования Python был спроектирован как замена более сложных языков программирования C, C++, Java, и может похвастаться простой установкой и настройкой, читабельным кодом, быстрой скоростью написания кода. В данном случае язык программирования Python использует специальный интерфейс для общения с роботом и отправки команды.

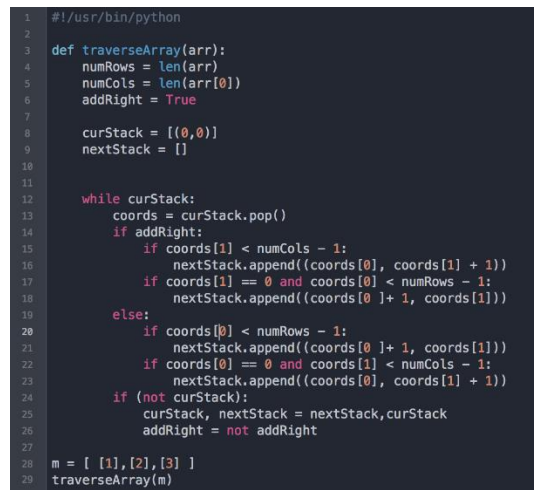
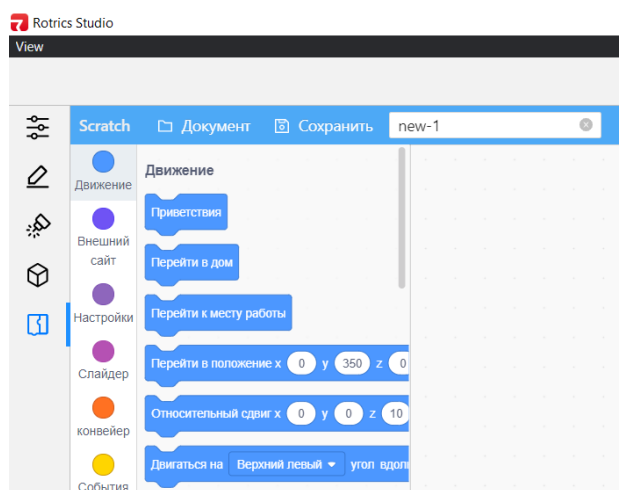


Рисунок 2.0.1 и 2.0.2 – Языки программирования Scratch и Python

§ 2.1. Программирование на языке Scratch

§ 2.1.1. Модуль «Держатель»

После установки модуля «Держатель», удостоверись что он надежно и крепко закреплен в механических пазах. Далее, запустите «Rotrics Studio» и следуйте следующим шагам:

- Откройте вкладку «Write & Draw»;

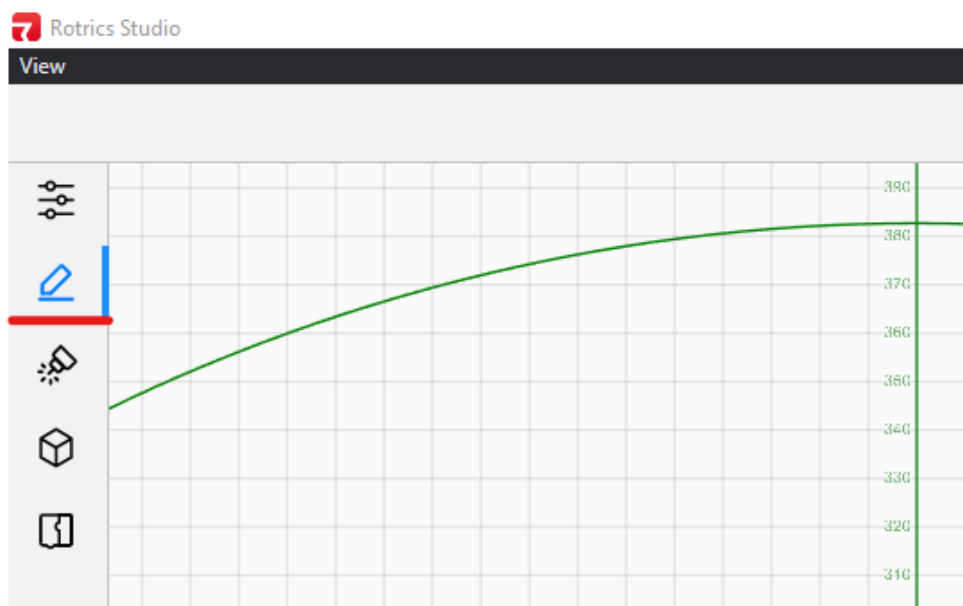


Рисунок 2.1.1.1 – Write and Draw

- В правой части программы располагается управляющая панель, выберите интересующую вас пиктограмму для изображения различных фигур, текста или рисунка (Рис 2.1.1.2);

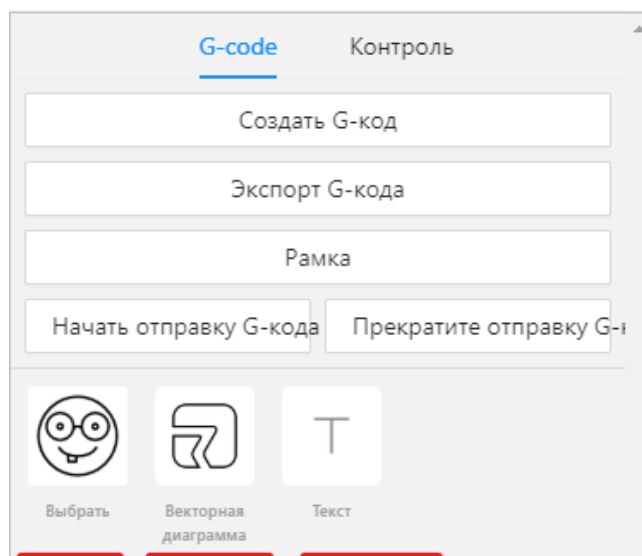


Рисунок 2.1.1.2 – Панель управления

- После выбора любой пиктограммы, откроются дополнительные свойства в панели управления, где можно выбрать необходимые свойства рисунка, такие как размер, заполнение, ускорение и так далее (Рис 2.1.1.3);

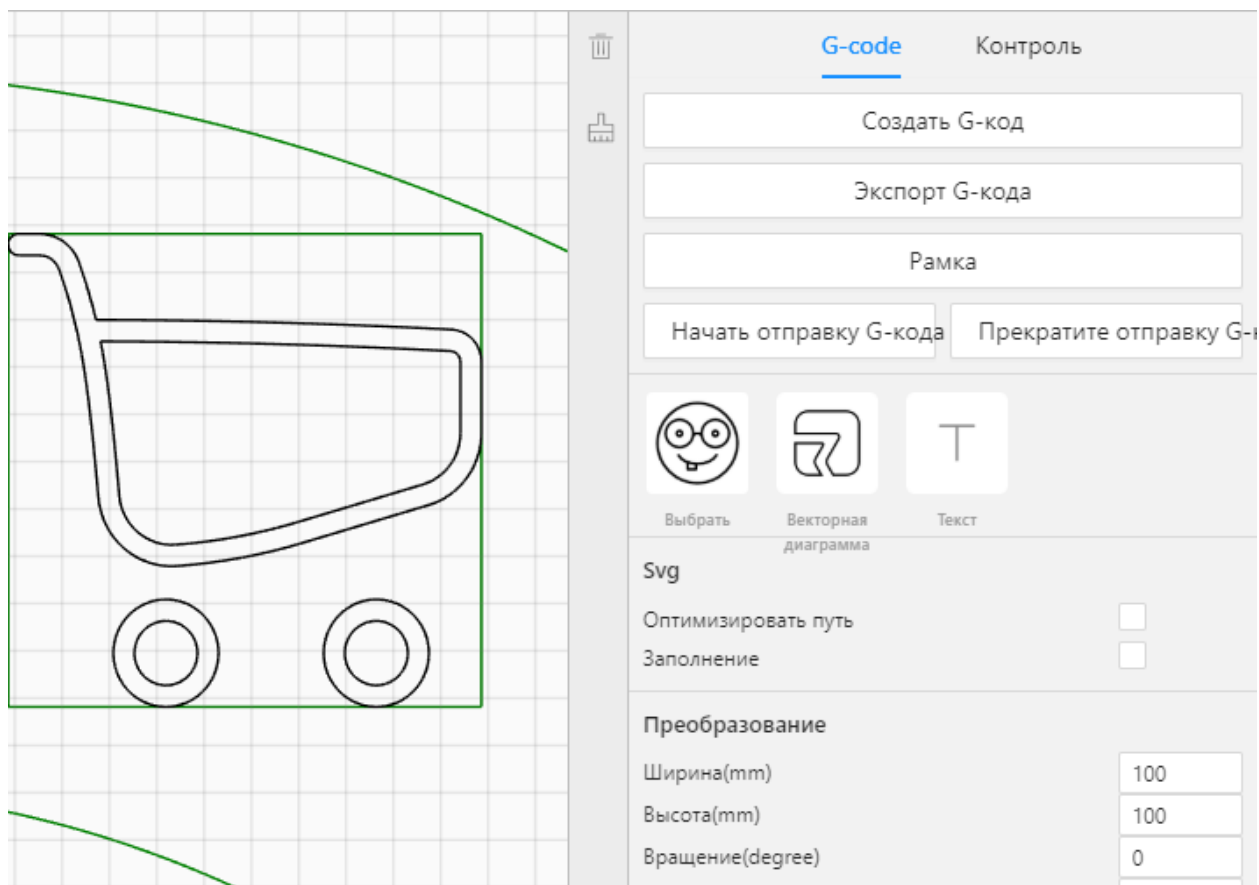


Рисунок 2.1.1.3 – Свойства рисунка в панели управления

- Далее, нажмите кнопку «Рамка», робот предварительно покажет вам границы вашего рисунка. При необходимости измените расположение листа бумаги или рисунка с помощью мыши;
- После того, как ваш рисунок удовлетворяет всем условиям, можно нажимать кнопку «Создать G-код», что создать специальный программный код для робота, по которым он будет ориентироваться в процессе выполнения работы. После успешного создания G-кода, нажмите на кнопку «Начать отправку G-кода» для начала рисования графического объекта на поверхности бумаги.
- Поздравляем, вам первый графический рисунок готов! Такие же манипуляции можно будет проводить и с SVG-картинками и текстом.

§ 2.1.2. Модуль «2.5W Лазер»

Внимание! При работе с модулем «2.5W Лазер», соблюдайте необходимую технику безопасности, а именно при включении лазера одевайте специальные отражающие очки; после выключения лазера, не притрагивайтесь к нему в течении 15 минут, обязательно проветривайте рабочую область и не создавайте пожароопасных ситуаций! Будьте осторожны при работе с лазером!

После установки модуля «2.5W Лазер», удостоверитесь что он надежно и крепко закреплен в механических пазах. При правильной установке, автоматически включится маленький охлаждающий вентилятор расположенный в верхней части лазерного модуля. Далее, запустите «Rotrics Studio» и следуйте следующим шагам:

- Откройте вкладку «Laser» (Рис 2.1.2.1);

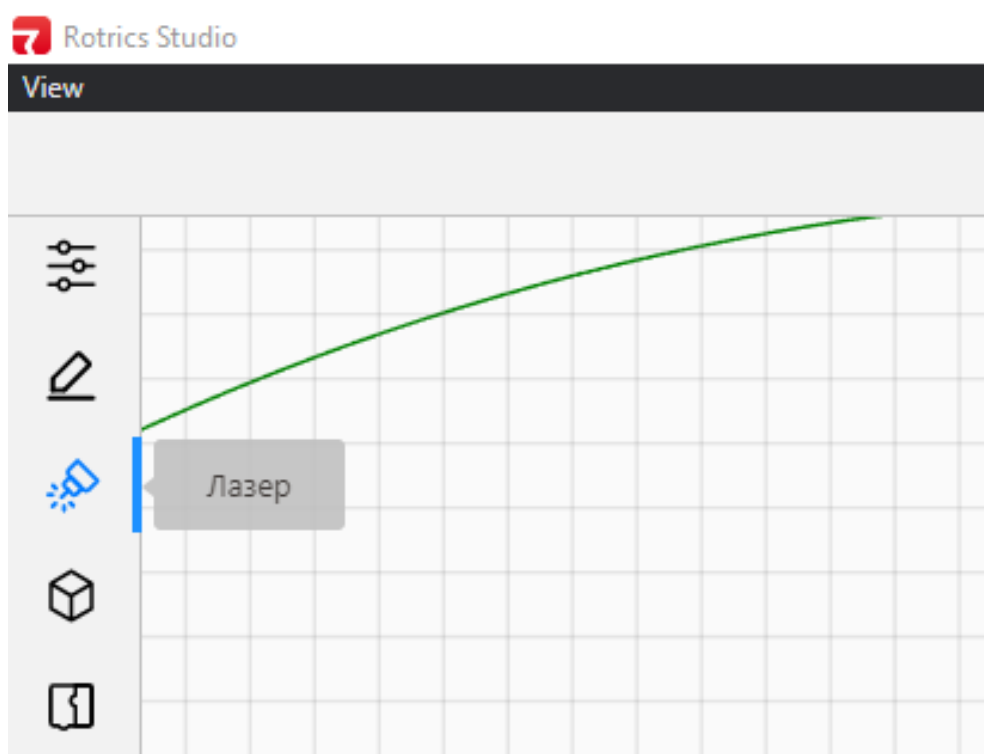


Рисунок 2.1.2.1 – Laser

- В правой части программы располагается управляющая панель, выберите интересующую вас пиктограмму для изображения различных фигур, текста или рисунка (Рис 2.1.2.2);

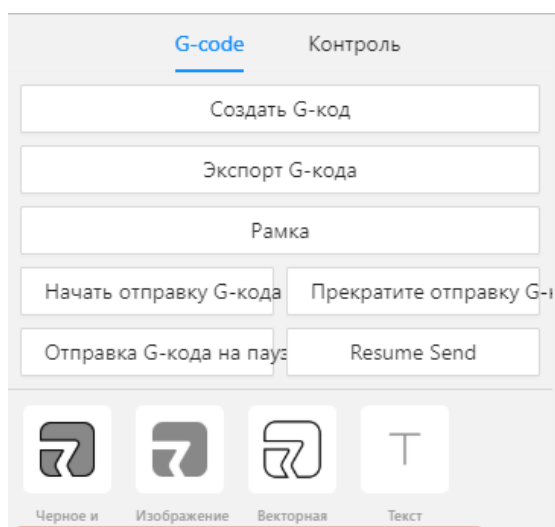


Рисунок 2.1.2.2 – Панель управления

- После выбора любой пиктограммы, откроются дополнительные свойства в панели управления, где можно выбрать необходимые свойства рисунка, такие как размер, заполнение, ускорение и так далее (Рис 2.1.2.3);

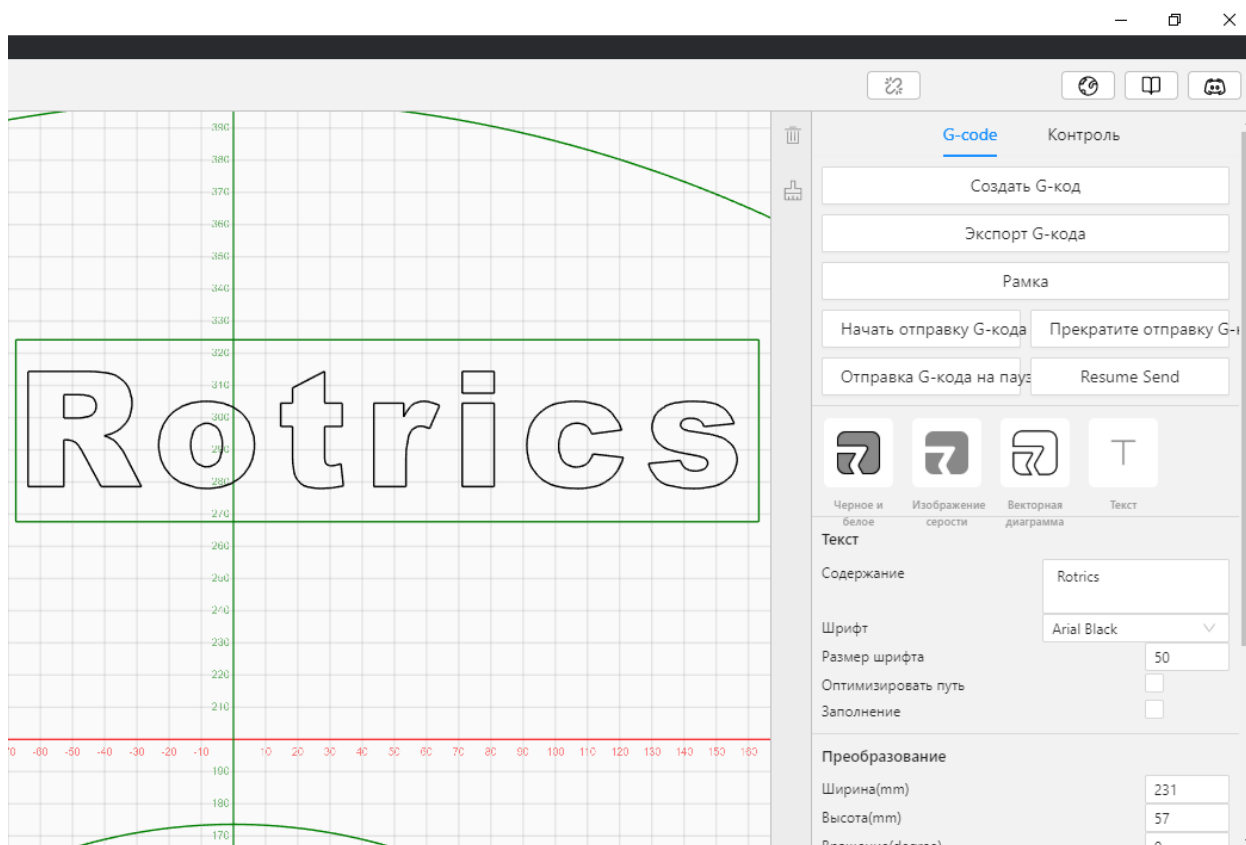


Рисунок 2.1.2.3 – Свойства рисунка в панели управления

- Далее, нажмите кнопку «Рамка», робот предварительно покажет вам границы вашего рисунка. При необходимости измените

расположение деревянной плоскости или рисунка с помощью мыши;

- После того, как ваш рисунок удовлетворяет всем условиям, можно нажимать кнопку «Создать G-код», что создать специальный программный код для робота, по которым он будет ориентироваться в процессе выполнения работы. После успешного создания G-кода, нажмите на кнопку «Начать отправку G-кода» для начала рисования графического объекта на деревянной поверхности.
- Поздравляем, вам первый графический рисунок готов! Такие же манипуляции можно будет проводить и с SVG-картинками и текстом.

§ 2.1.3. Модуль «3D-принтер»

После установки модуля «3D-принтер», удостоверитесь что он надежно и крепко закреплен в механических пазах. При правильной установке, автоматически включится маленький охлаждающий вентилятор расположенный в верхней части лазерного модуля. Далее, запустите «Rotrics Studio» и следуйте следующим шагам:

- Откройте вкладку «3D-Print» (Рис 2.1.3.1);

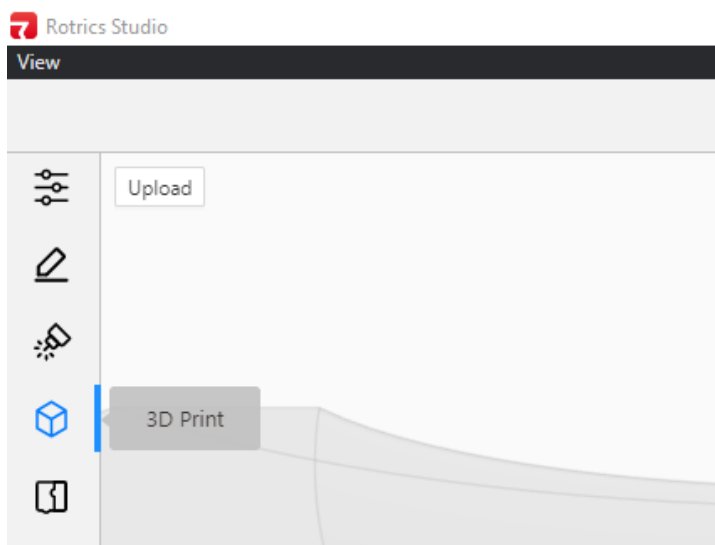


Рисунок 2.1.3.1 – 3D-Print

- В правой части программы располагается управляющая панель, выберите интересующую вас пиктограмму для изображения различных фигур, текста или рисунка (Рис 2.1.3.2);

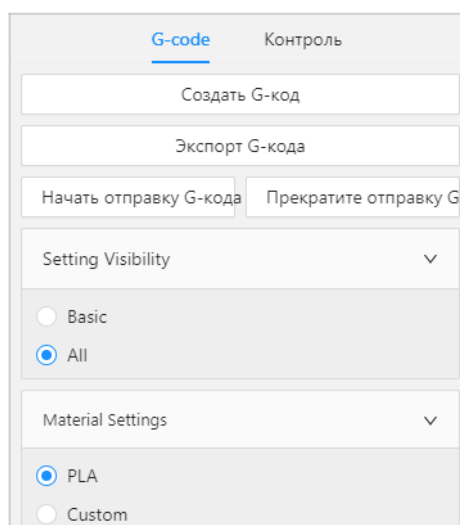


Рисунок 2.1.3.2 – Панель управления

- После выбора любой пиктограммы, откроются дополнительные свойства в панели управления, где можно выбрать необходимые свойства 3D-модели, такие как размер, заполнение, ускорение и так далее (Рис 2.1.3.3);

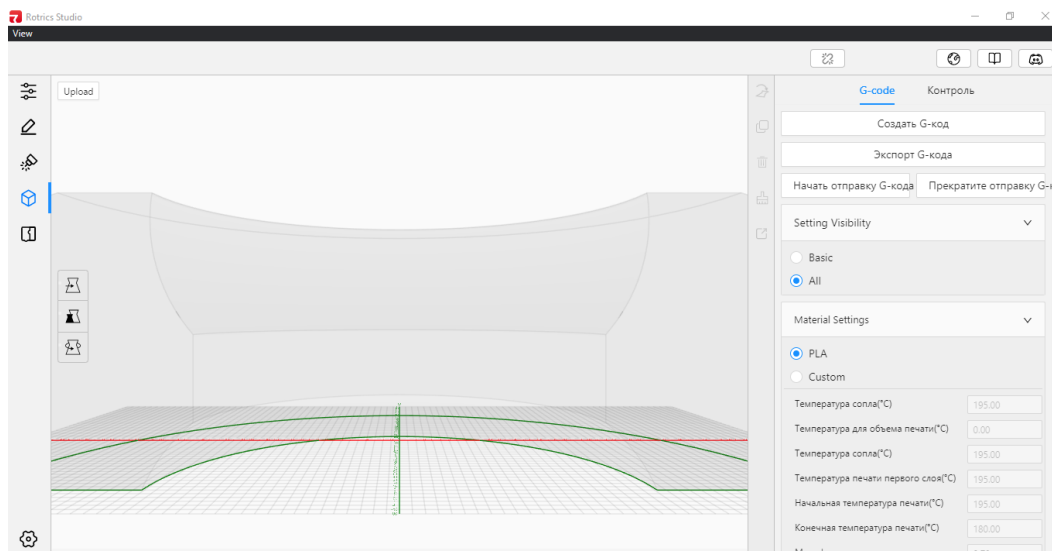


Рисунок 2.1.3.3 – Свойства рисунка в панели управления

- При необходимости измените расположение печатной плоскости или 3D-модели с помощью мыши;
- После того, как ваш рисунок удовлетворяет всем условиям, можно нажимать кнопку «Создать G-код», что создать специальный программный код для робота, по которым он будет ориентироваться в процессе выполнения работы. После успешного создания G-кода, нажмите на кнопку «Начать отправку G-кода» для начала печати объекта на поверхности.
- Поздравляем, ваш первый 3D-объект готов! После 3D-печати, обязательно прочистите все отверстия, где остался расплавленный пластик, потому что затвердевание пластика происходит очень быстро и для качественной печати в будущем следует очищать от пластика все узлы модуля сразу же после печати.

§ 2.1.4. Модуль «Пневматика: Мягкий держатель»

После установки модуля «Мягкий держатель», удостоверитесь что он надежно и крепко закреплен в механических пазах. При запуске операций, запустится пневматический компрессор, будьте аккуратны и осторожны. Далее, запустите «Rotrics Studio» и следуйте следующим шагам:

- Откройте вкладку «Basics» (Рис 2.1.4.1);

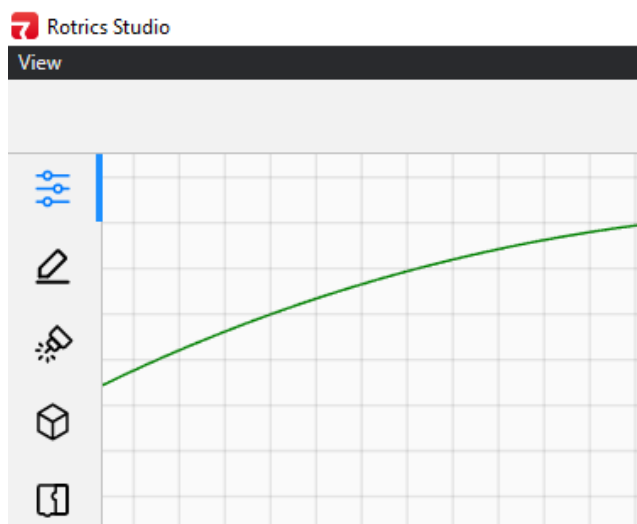


Рисунок 2.1.4.1 – Basics

- В правой части программы располагается управляющая панель, выберите модуль Air Picker для мягкого держателя (Рис 2.1.4.2);

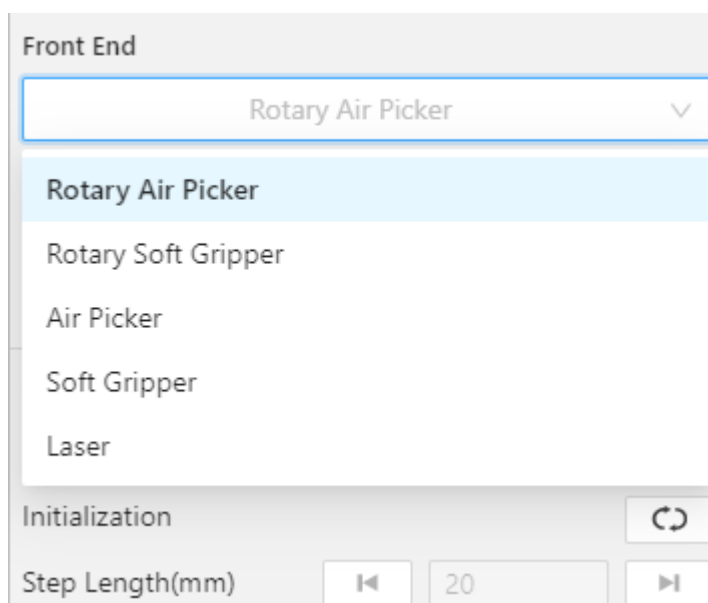


Рисунок 2.1.4.2 – Панель управления

- После выбора необходимого оборудования, откроются дополнительные свойства в панели управления, где можно выбрать необходимые свойства держателя, такие как действие, положение, ускорение и так далее (Рис 2.1.4.3);

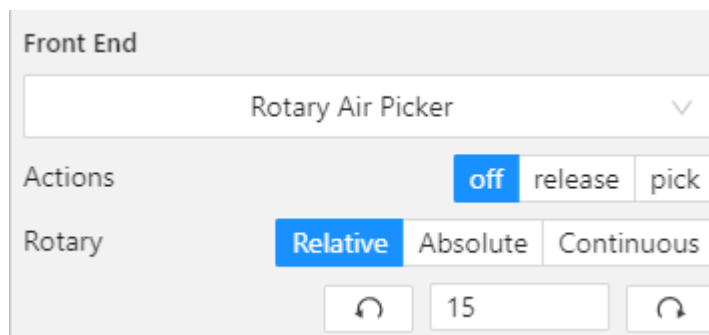


Рисунок 2.1.4.3 – Свойства рисунка в панели управления

- При необходимости измените положение манипулятора с помощью управляющих клавиш (Рис 2.1.4.4);

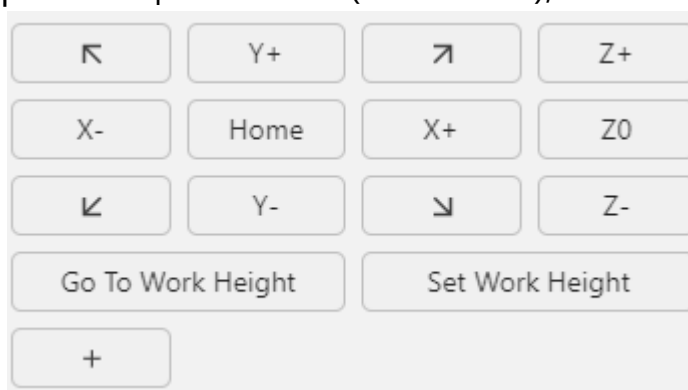


Рисунок 2.1.4.4 – Управляющие клавиши

- Поздравляем, ваше первое действие с пневматической системой выполнено! После работы с пневматической системой, прочистите все шланги, отверстия и технологические каналы. Храните элементы пневматической системы в условиях, при которых достигается наименьшее количество пыли и частиц в воздухе.

§ 2.1.5. Модуль «Пневматика: Твердый держатель»

После установки модуля «Твердый держатель», удостоверитесь что он надежно и крепко закреплен в механических пазах. При запуске операций, запустится пневматический компрессор, будьте аккуратны и осторожны. Далее, запустите «Rotrics Studio» и следуйте следующим шагам:

- Откройте вкладку «Basics» (Рис 2.1.5.1);



Рисунок 2.1.5.1 – Basics

- В правой части программы располагается управляющая панель, выберите модуль Soft Gripper для мягкого держателя (Рис 2.1.5.2);

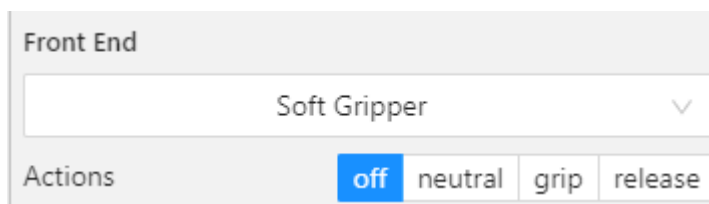


Рисунок 2.1.5.2 – Панель управления

- После выбора необходимого оборудования, откроются дополнительные свойства в панели управления, где можно выбрать необходимые свойства держателя, такие как действие, положение, ускорение и так далее. При необходимости измените положение манипулятора с помощью управляющих клавиш (Рис 2.1.5.3);



Рисунок 2.1.5.3 – Управляющие клавиши

- Поздравляем, ваше первое действие с пневматической системой выполнено! После работы с пневматической системой, прочистите все шланги, отверстия и технологические каналы. Храните элементы пневматической системы в условиях, при которых достигается наименьшее количество пыли и частиц в воздухе.

§ 2.1.6. Модуль «Механика: Конвейерная лента»

После установки модуля «Конвейерная лента», удостоверитесь что он надежно и крепко закреплен в механических пазах. Далее, запустите «Rotrics Studio» и следуйте следующим шагам:

- Откройте вкладку «Basics» (Рис 2.1.6.1);

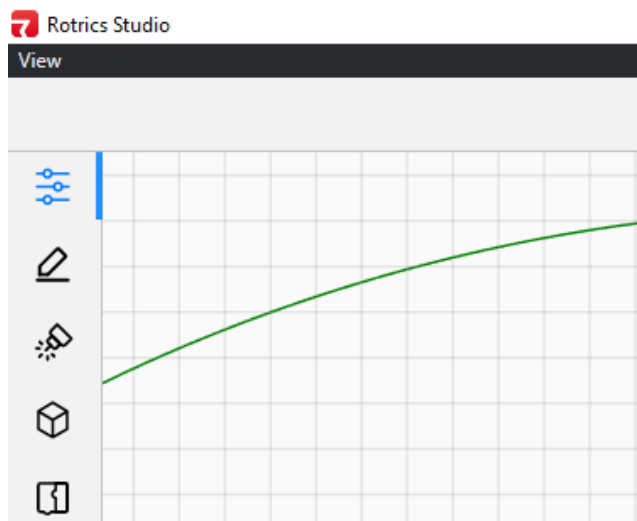


Рисунок 2.1.6.1 – Basics

- В правой части программы располагается управляющая панель, выберите модуль Conveyor Belt для конвейерной ленты (Рис 2.1.6.2);

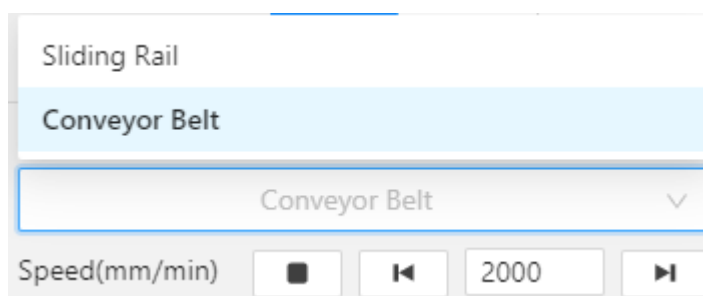


Рисунок 2.1.6.2 – Панель управления

- После выбора необходимого оборудования, откроются дополнительные свойства в панели управления, где можно выбрать необходимые свойства конвейерной ленты, такие как остановка, скорость, ускорение и так далее (Рис 2.1.6.2);
- При необходимости измените положение манипулятора с помощью управляющих клавиш (Рис 2.1.6.3);



Рисунок 2.1.6.3 – Управляющие клавиши

- Поздравляем, ваше первое действие с конвейерной лентой выполнено! После работы с конвейерной лентой, прочистите все отверстия, технологические каналы и саму ленту. Храните элементы конвейерной ленты в условиях, при которых достигается наименьшее количество пыли и частиц в воздухе.

§ 2.1.7. Модуль «Механика: Направляющая»

После установки модуля «Направляющая», удостоверитесь что он надежно и крепко закреплен в механических пазах. Далее, запустите «Rotrics Studio» и следуйте следующим шагам:

- Откройте вкладку «Basics» (Рис 2.1.7.1);



Рисунок 2.1.7.1 – Basics

- В правой части программы располагается управляющая панель, выберите модуль Soft Gripper для мягкого держателя (Рис 2.1.5.2);

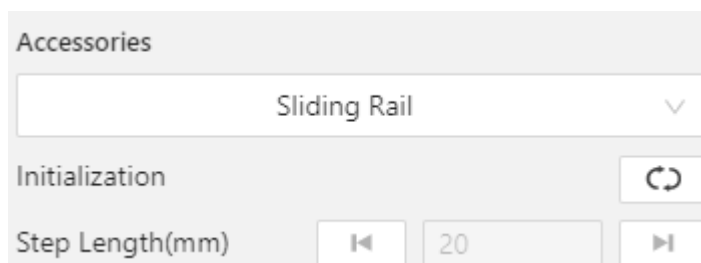


Рисунок 2.1.7.2 – Панель управления

- После выбора необходимого оборудования, откроются дополнительные свойства в панели управления, где можно выбрать необходимые свойства направляющей, такие как скорость, ускорение и так далее (Рис 2.1.7.2). При необходимости измените положение манипулятора с помощью управляющих клавиш (Рис 2.1.7.3);



Рисунок 2.1.7.3 – Управляющие клавиши

- Поздравляем, ваше первое действие с направляющей выполнено! После работы с направляющей, прочистите все отверстия, технологические каналы и саму направляющую. Храните элементы направляющей в условиях, при которых достигается наименьшее количество пыли и частиц в воздухе.

§ 2.1.8. Основы программирования на Scratch

Используя язык программирования Scratch, можно добиться существенной автоматизации обычных рутинных действий. Для того, чтобы начать программировать на Scratch, необходимо перейти на вкладку «Code» (Рис. 2.1.8.1)

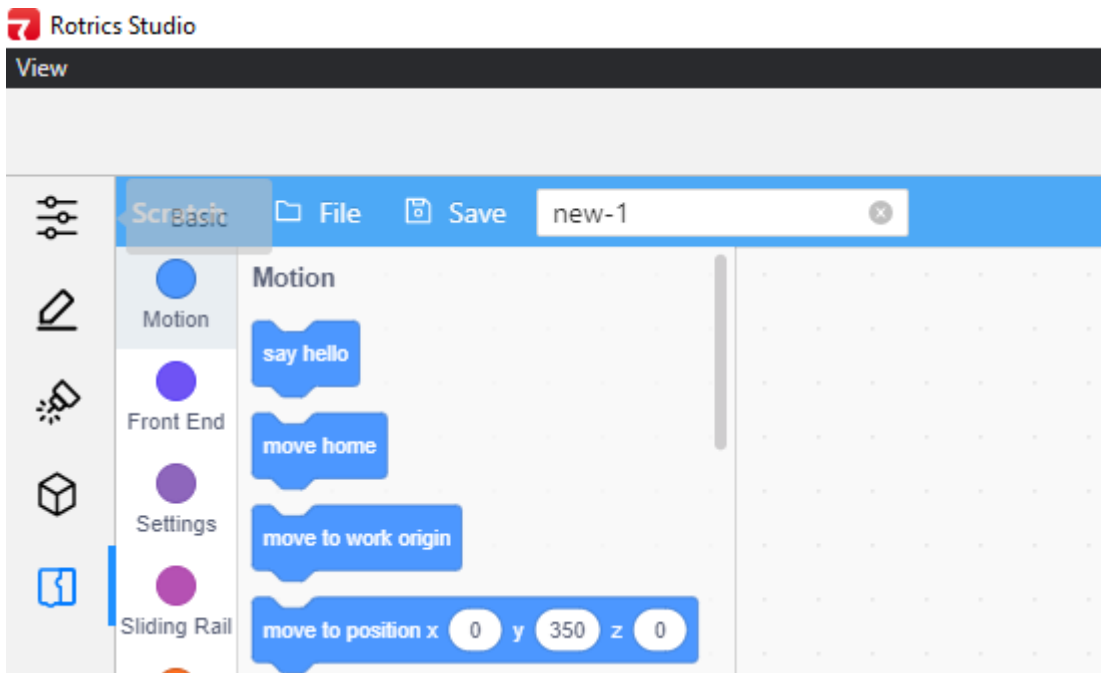
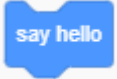
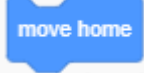
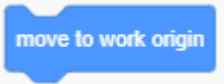



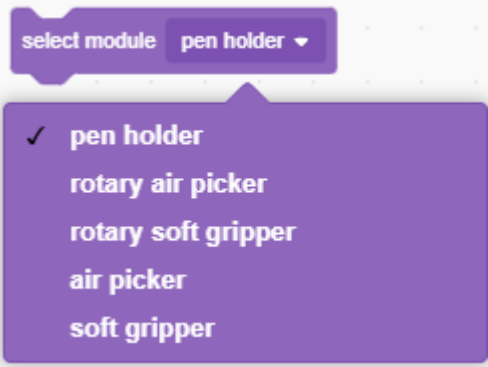

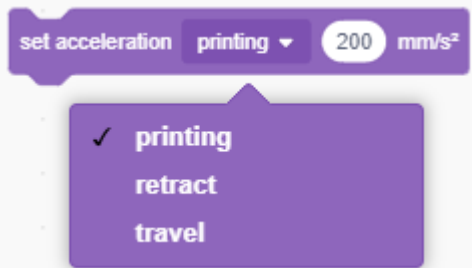
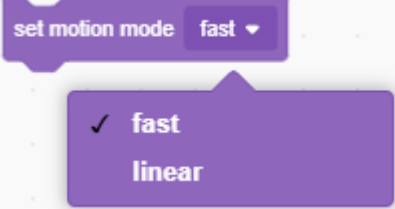
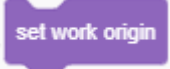
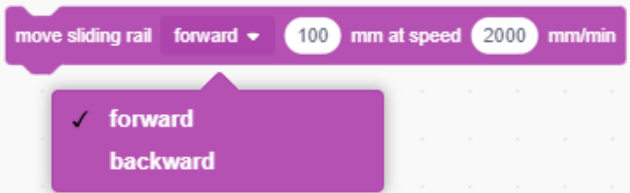


Рисунок 2.1.8.1 – Вкладка «Code»

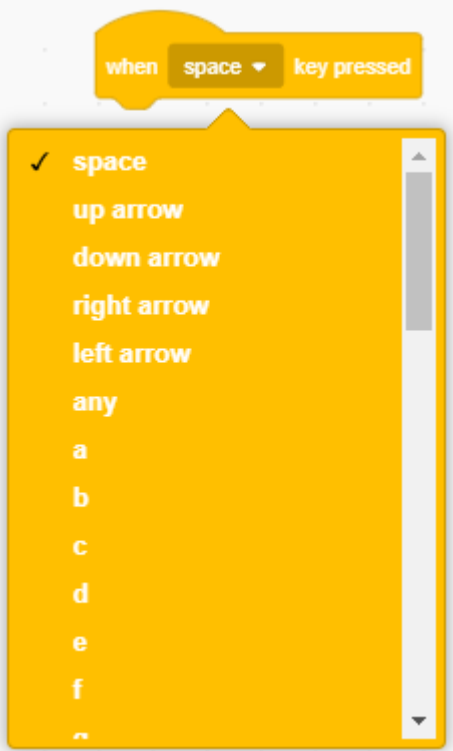
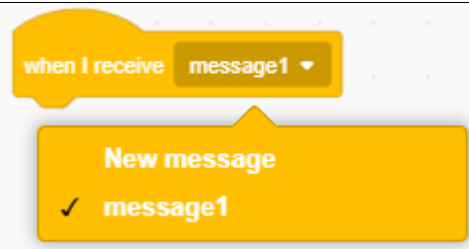
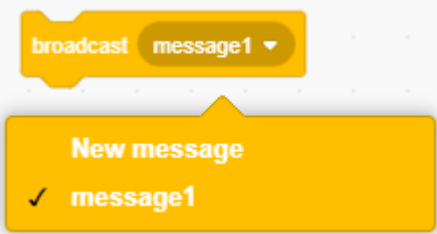


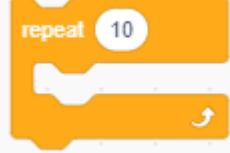

Ниже приведена специальная таблица с описанием всех функций и конструкций языка программирования Scratch и приложений к нему (Таблица 2.1.8.2)



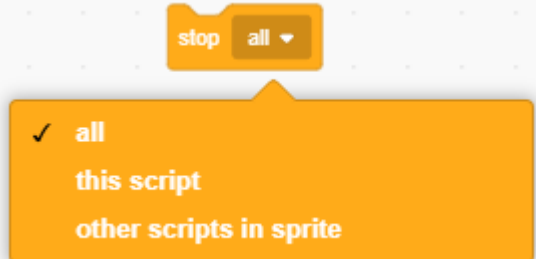
Таблица 2.1.8.2 – Описание функций языка Scratch

Группа	Функция	Предназначение
Движение		Выводи любое слово/предложение
		Перемещает робота в начальное положение (Home)
		Перемещает робота в рабочее положение (Определяется на панели управления)
		Перемещает робота в заданные координаты


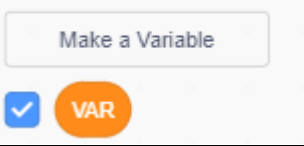
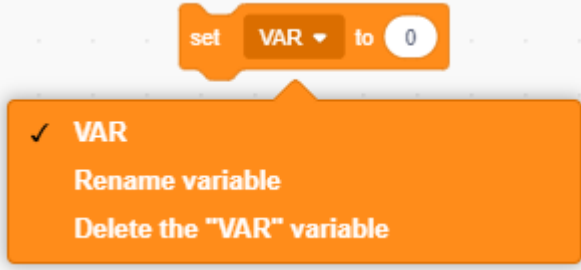
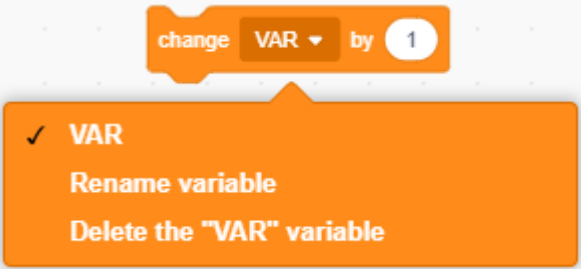
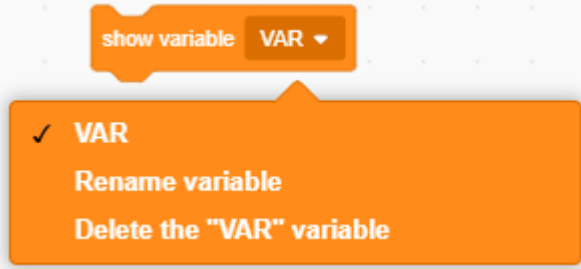
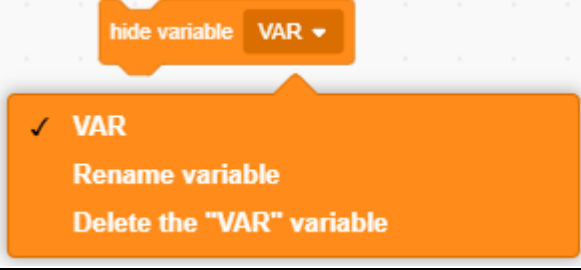
		Перемещает робота относительно заданных координат
		Перемещает прямоугольник в любой левый или правый угол с заданной шириной и высотой
		Перемещает круг влево, вправо, вверх или вниз с заданным радиусом
		Вращает манипулятор по часовой или против часовой стрелки на заданный угол
		Вращает манипулятор на заданный угол (без направления)
		Вращает манипулятор по часовой или против часовой стрелки с заданной скоростью
Пневматика		Использует мягкий пневматический захват с различными режимами работы
		Использует твердый пневматический захват с различными режимами работы

Модули		Позволяет выбрать необходимый для работы сменный модуль
		Задаёт скорость работы сменного модуля
		Задаёт ускорение для сменного модуля при работе с печатью/перемещением
		Задаёт режим перемещения (быстрый/линейный)
		Перемещает робота в рабочее положение
Направляющая		Активирует направляющую с заданным направлением, скоростью и пройденным расстоянием
		Возвращает направляющую в исходное положение
Конвейерная лента		Активирует конвейерную ленту с заданным направлением и скоростью
		Останавливает работу конвейерной ленты
События		Устанавливает логическое условие истинности, если флажок был нажат с помощью мыши

			Устанавливает логическое условие истинности, если определенная клавиша на компьютере или ноутбуке была нажата
			Когда будет принято какое-то сообщение, будет установлено логическое условие истинности
			Позволяет отслеживать статус сообщения
			Позволяет отслеживать статус сообщения и ожидать внешних действий от пользователя
Управление			Заставляет ожидать заданное количество секунд
			Заставляет повторять определенное действие заданное количество раз
			Стандартный цикл «for»

		Стандартная управляющая структура «if-then»
		Стандартная управляющая структура «if-then-else»
		Заставляет ждать пока не произойдет определенное действие
		Заставляет повторить до определенного действия или значения
		Останавливает этот Scratch-скрипт, все скрипты или скрипты, которые находятся только в этом файле
Сенсоры		Устанавливает логическое условие истинности, если определенная клавиша на компьютере или ноутбуке была нажата

		Устанавливает логическое условие истинности, если робот находится в текущей заданной позиции
		Устанавливает логическое условие истинности, если ускорение определенного типа включено
Операторы		Математические операторы сложения, вычитания, умножения и деления числовых значений или переменных
		Определяет случайное значение переменной в заданном диапазоне
		Математические операторы сравнения числовых значений или переменных
		Логические операторы, которые позволяют сравнивать значения переменных
		Оператор производит деление числе и берет остаток в качестве числового значения Оператор производит округление числового значения или переменной Оператор производит вычисление математической функции

Работа с переменными		<p>Логические операторы для взаимодействия со словами или предложениями</p>
	<p>Variables</p> 	<p>Создает переменную и позволяет включать и отключать их для использования в скетче</p>
		<p>Устанавливает начальное значение переменной заданным числом</p>
		<p>Меняет значение переменной на заданное число</p>
		<p>Выводит значение переменной на экран ноутбука или компьютера</p>
		<p>Позволяет скрывать переменную</p>

§ 2.2. Программирование на языке Python

§ 2.2.1. Модуль «2.5W Лазер»

После установки модуля «2.5W Лазер», удостоверитесь что он надежно и крепко закреплен в механических пазах. При правильной установке, автоматически включится маленький охлаждающий вентилятор расположенный в верхней части лазерного модуля. Далее, запустите «Microsoft Visual Studio Code» и следуйте следующим шагам:

- Откройте клонированный репозиторий в Microsoft Visual Code (Рис 2.2.1.1 и 2.2.1.2):

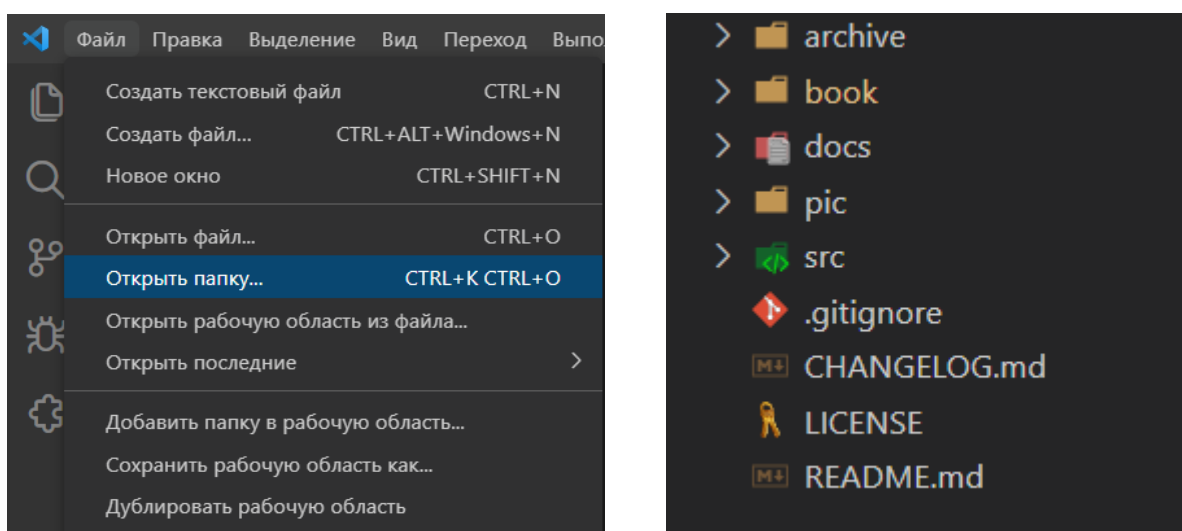


Рисунок 2.2.1.1 и 2.2.1.2 – Открытие репозитория

- Перейдите в директорию `src` и откройте файл «`laser-moving.py`». Вы увидите программный код, написанный на языке Python (Рис 2.2.1.3):

```
laser-moving.py U X
src > laser-moving.py > ...
1  from pydexarm import Dexarm
2  dexarm = Dexarm("COM8") # Подключение к Windows
3
4  dexarm.go_home()
5
6  dexarm.move_to(135, 225, -70) # точка A
7  dexarm.laser_on(255) # включение лазера
8  dexarm.move_to(120, 210, -70) # точка B
9  dexarm.move_to(105, 225, -70) # точка C
10 dexarm.move_to(120, 240, -70) # точка D
11 dexarm.move_to(135, 225, -70) # от точки D к точки A
12 dexarm.laser_off() # выключение лазера
13
14 dexarm.go_home()
```

Рисунок 2.2.1.3 – Программный код для запуска лазера

- Давайте разберем что означает каждая команда:
 - `from pydexarm import Dexarm` – включает в основной код библиотеку-интерфейс для общения с роботом при помощи связи между Python и G-code;
 - `dexarm = Dexarm("COM8")` – отправляет скомпилированный код в робот на указанный COM-порт, а именно COM8;
 - `dexarm.go_home()` – дает команду роботу вернуться в начальное или исходное положение. Функция не требует аргументов, так как робот настроен на начальное положение в (0; 0; 0);
 - `dexarm.move_to(135, 225, -95)` – определяет координаты по трем осям, и отправляет роботу команду для перемещения манипулятор робота в данную точку, а именно – по оси X – 135; по оси Y – 225; по оси Z – -95;
 - `dexarm.laser_on(1000)` – дает команду для запуска лазера, в скобках указывает мощность лазера от 1 до 255;
 - `dexarm.laser_off()` – дает команду для отключения лазера;

§ 2.2.2. Модуль «Пневматика: Мягкий держатель»

После установки модуля «Мягкий держатель», удостоверитесь что он надежно и крепко закреплен в механических пазах. При запуске операций, запустится пневматический компрессор, будьте аккуратны и осторожны. Далее, запустите «Microsoft Visual Studio Code» и следуйте следующим шагам:

- Откройте клонированный репозиторий в Microsoft Visual Code (Рис 2.2.2.1 и 2.2.2.2):

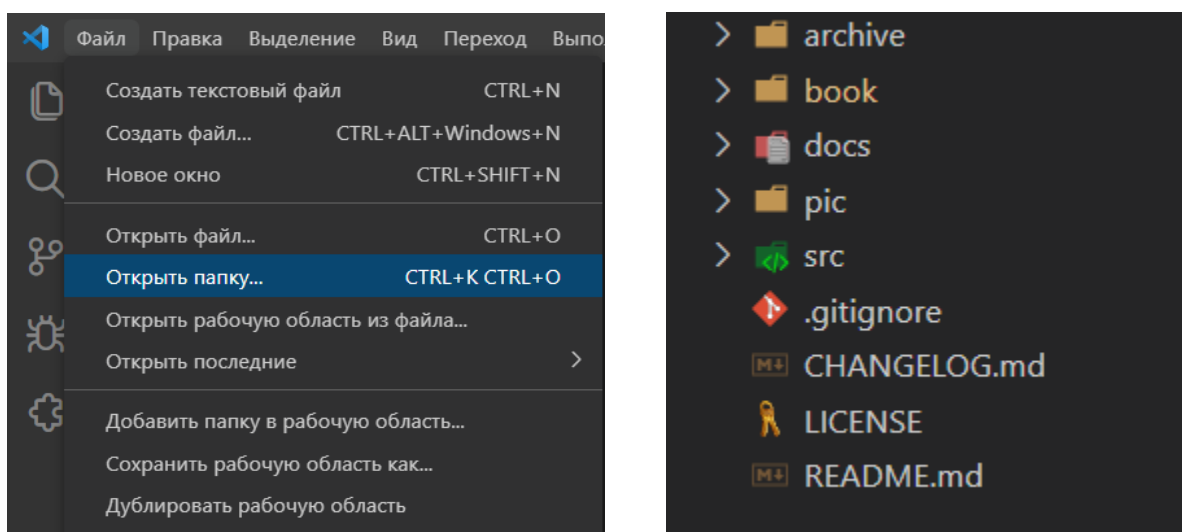


Рисунок 2.2.2.1 и 2.2.2.2 – Открытие репозитория

- Перейдите в директорию `src` и откройте файл «`air-picker.py`». Вы увидите программный код, написанный на языке Python (Рис 2.2.2.3):

```
laser-moving.py U  air-picker.py X
src > air-picker.py > ...
1  from pydexarm import Dexarm
2  dexarm = Dexarm("COM10") # Подключение к Windows
3
4  dexarm.go_home()
5  dexarm.move_to(135, 225, 0) # А
6
7  dexarm.air_picker_place()
8  dexarm.air_picker_pick() # вдох
9  dexarm.dealy_ms(1000)
10
11 dexarm.go_home()
12
13 dexarm.air_picker_place() # выдох
14 dexarm.dealy_ms(1000)
15
16 dexarm.air_picker_stop() # остановка подачи воздуха
```

Рисунок 2.2.2.3 – Программный код для запуска мягкого держателя

- Давайте разберем что означает каждая команда:
 - `from pydexarm import Dexarm` – включает в основной код библиотеку-интерфейс для общения с роботом при помощи связи между Python и G-code;
 - `dexarm = Dexarm("COM8")` – отправляет скомпилированный код в робот на указанный COM-порт, а именно COM8;
 - `dexarm.go_home()` – дает команду роботу вернуться в начальное или исходное положение. Функция не требует аргументов, так как робот настроен на начальное положение в (0; 0; 0);
 - `dexarm.move_to(135, 225, -95)` – определяет координаты по трем осям, и отправляет роботу команду для перемещения манипулятор робота в данную точку, а именно – по оси X – 135; по оси Y – 225; по оси Z – -95;
 - `dexarm.air_picker_pick()` – дает команду для запуска откачки воздуха из держателя;
 - `dexarm.air_picker_place()` – дает команду для подачи воздуха из пневматического блока в держатель;
 - `dexarm.air_picker_stop()` – дает команду для остановки подачи воздуха в любом направлении;
 - `dexarm.air_picker_nature()` – дает команду для возврата в исходное положение всей пневматической системы;

§ 2.2.3. Модуль «Пневматика: Твердый держатель»

После установки модуля «Твердый держатель», удостоверитесь что он надежно и крепко закреплен в механических пазах. При запуске операций, запустится пневматический компрессор, будьте аккуратны и осторожны. Далее, запустите «Microsoft Visual Studio Code» и следуйте следующим шагам:

- Откройте клонированный репозиторий в Microsoft Visual Code (Рис 2.2.3.1 и 2.2.3.2):

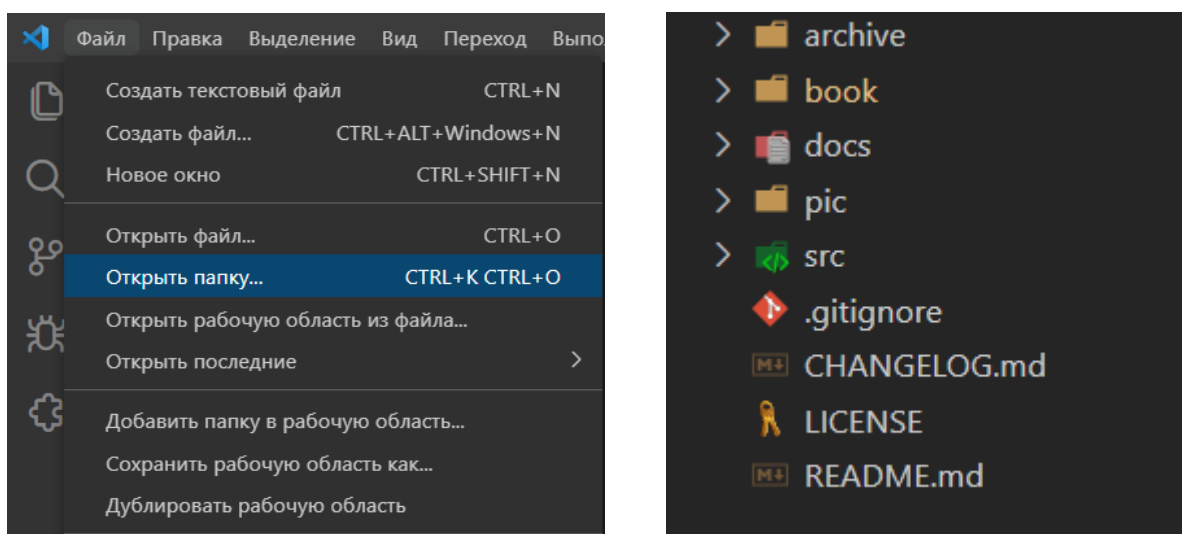


Рисунок 2.2.2.1 и 2.2.2.2 – Открытие репозитория

- Перейдите в директорию `src` и откройте файл «`soft-gripper.py`». Вы увидите программный код, написанный на языке Python (Рис 2.2.3.3):

```
soft-gripper.py U x
src > soft-gripper.py > ...
1 from pydexarm import Dexarm
2 dexarm = Dexarm("COM10") # Подключение к Windows
3
4 dexarm.go_home()
5 dexarm.fast_move_to(135, 225, -40) # A
6 dexarm.soft_gripper_place() # разжать щупальцу
7
8 dexarm.dealy_ms(1000)
9 dexarm.soft_gripper_pick() # зажать щупальцу
10
11 dexarm.dealy_ms(1000)
12 dexarm.go_home()
13
14 dexarm.soft_gripper_place() # разжать щупальцу
15 dexarm.dealy_ms(1000)
16
17 dexarm.soft_gripper_stop() # остановка подачи воздуха
18
```

Рисунок 2.2.3.3 – Программный код для запуска мягкого держателя

- Давайте разберем что означает каждая команда:
 - `from pydexarm import Dexarm` – включает в основной код библиотеку-интерфейс для общения с роботом при помощи связи между Python и G-code;
 - `dexarm = Dexarm("COM8")` – отправляет скомпилированный код в робот на указанный COM-порт, а именно COM8;
 - `dexarm.go_home()` – дает команду роботу вернуться в начальное или исходное положение. Функция не требует аргументов, так как робот настроен на начальное положение в (0; 0; 0);
 - `dexarm.move_to(135, 225, -95)` – определяет координаты по трем осям, и отправляет роботу команду для перемещения манипулятор робота в данную точку, а именно – по оси X – 135; по оси Y – 225; по оси Z – -95;
 - `dexarm.soft_gripper_pick()` – дает команду для зажатия щупальцы;
 - `dexarm.soft_gripper_place()` – дает команду для разжатия щупальцы;
 - `dexarm.soft_gripper_stop()` – дает команду для остановки подачи воздуха в любом направлении;
 - `dexarm.soft_gripper_nature()` – дает команду для возврата в исходное положение всей пневматической системы;

§ 2.2.4. Модуль «Механика: Конвейерная лента»

После установки модуля «Конвейерная лента», удостоверитесь что он надежно и крепко закреплен в механических пазах. Далее, запустите «Microsoft Visual Studio Code» и следуйте следующим шагам:

- Откройте клонированный репозиторий в Microsoft Visual Code (Рис 2.2.4.1 и 2.2.4.2):

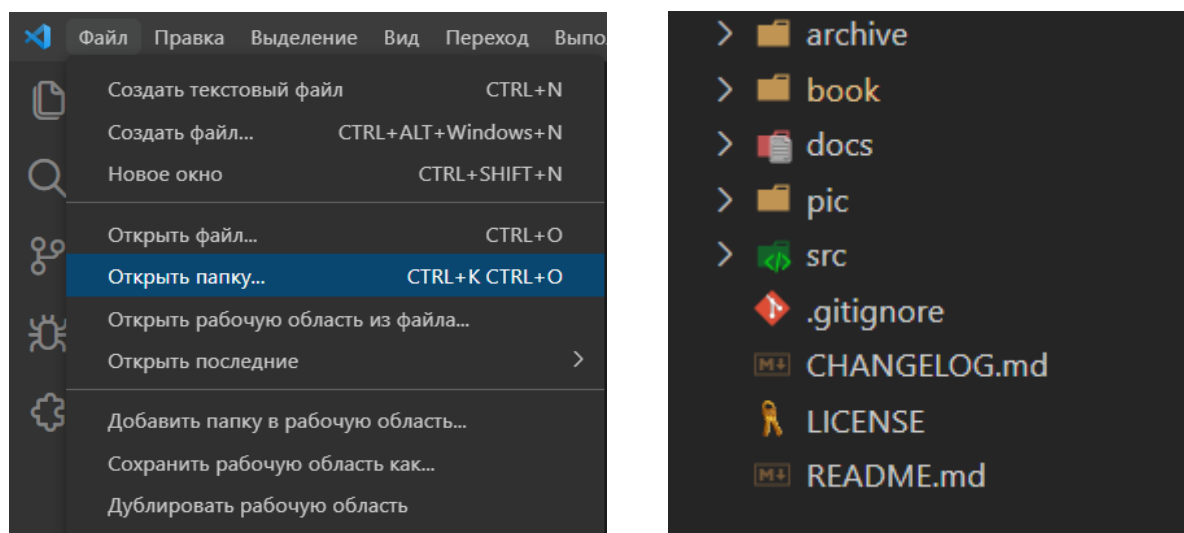


Рисунок 2.2.4.1 и 2.2.4.2 – Открытие репозитория

- Перейдите в директорию `src` и откройте файл «`conveyor.py`». Вы увидите программный код, написанный на языке Python (Рис 2.2.4.3):

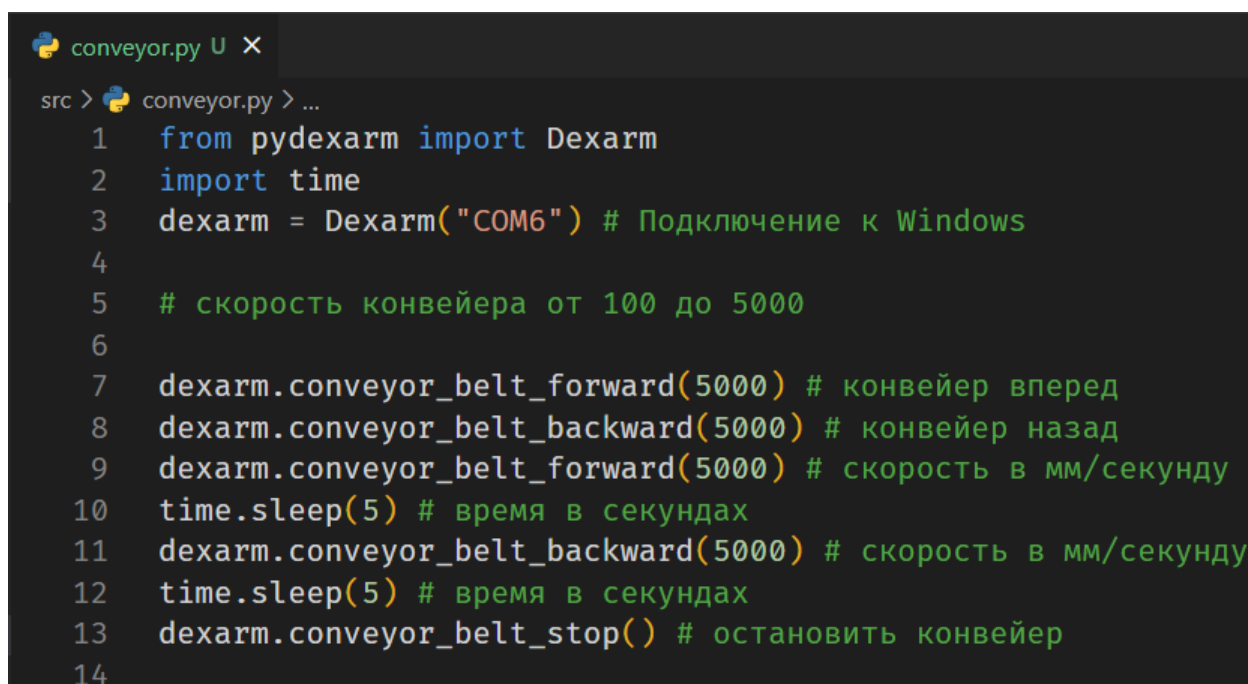


Рисунок 2.2.4.3 – Программный код для запуска мягкого держателя

- Давайте разберем что означает каждая команда:
 - `from pydexarm import Dexarm` – включает в основной код библиотеку-интерфейс для общения с роботом при помощи связи между Python и G-code;
 - `dexarm = Dexarm("COM8")` – отправляет скомпилированный код в робот на указанный COM-порт, а именно COM8;
 - `dexarm.go_home()` – дает команду роботу вернуться в начальное или исходное положение. Функция не требует аргументов, так как робот настроен на начальное положение в (0; 0; 0);
 - `dexarm.conveyor_belt_forward(5000)` – запускает конвейер вперед со скоростью 5000 мм/сек;
 - `dexarm.conveyor_belt_backward(5000)` – запускает конвейер назад со скоростью 5000 мм/сек;
 - `time.sleep(5)` – запускает временной интервал операции в течении заданного количества секунд;
 - `dexarm.conveyor_belt_stop()` – останавливает конвейер;

§ 2.2.5. Модуль «Механика: Направляющая»

После установки модуля «Конвейерная лента», удостоверитесь что он надежно и крепко закреплен в механических пазах. Далее, запустите «Microsoft Visual Studio Code» и следуйте следующим шагам:

- Откройте клонированный репозиторий в Microsoft Visual Code (Рис 2.2.5.1 и 2.2.5.2):

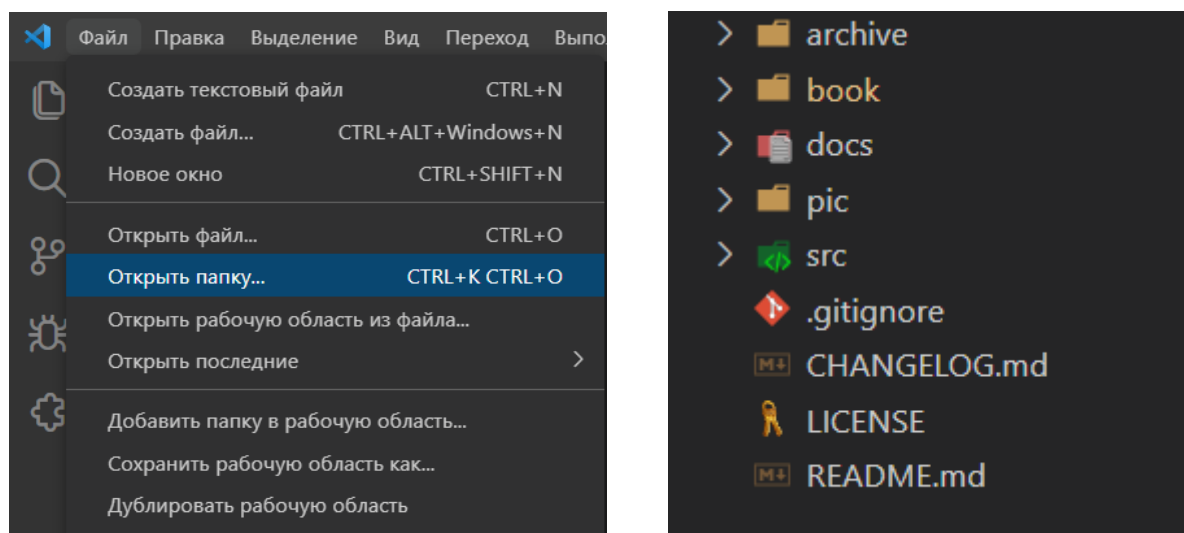


Рисунок 2.2.5.1 и 2.2.5.2 – Открытие репозитория

- Перейдите в директорию `src` и откройте файл «`railway.py`». Вы увидите программный код, написанный на языке Python (Рис 2.2.5.3):

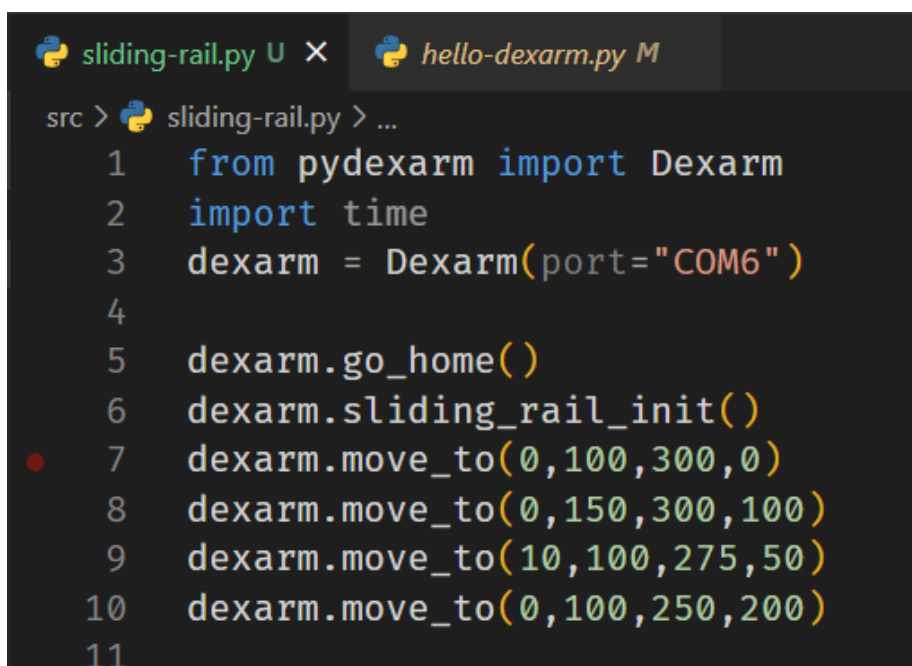


Рисунок 2.2.3.3 – Программный код для запуска мягкого держателя

- Давайте разберем что означает каждая команда:
 - `from pydexarm import Dexarm` – включает в основной код библиотеку-интерфейс для общения с роботом при помощи связи между Python и G-code;
 - `dexarm = Dexarm("COM8")` – отправляет скомпилированный код в робот на указанный COM-порт, а именно COM8;
 - `dexarm.go_home()` – дает команду роботу вернуться в начальное или исходное положение. Функция не требует аргументов, так как робот настроен на начальное положение в (0; 0; 0);
 - `dexarm.move_to(135, 225, -95, 100)` – определяет координаты по трем осям, и отправляет роботу команду для перемещения манипулятор робота в данную точку, а именно – по оси X – 135; по оси Y – 225; по оси Z – -95. Четвертое числовое значение определяет скорость перемещения по направляющей рельсе;
 - `dexarm.sliding_rail_inint()` – дает команду для инициализации направляющей рельсы;

§ 2.2.6. Основы промышленной автоматизации

После освоения всех примеров и получения необходимых первоначальных навыков по работе с Rotrics Dex ARM на языке программирования Python, рекомендуем вам еще раз обратиться к таблице со всеми возможными функциями, используемыми при программировании робота (Таблица 2.2.6.1):

Таблица 2.2.6.1 – Полное описание библиотек-интерфейса «pydexarm.py»:		
Группа	Команда	Описание
Служебные	def __init__()	Инициализация
	def _send_cmd()	Отправка команды
Движение	def go_home(self)	Перемещение в начальную позицию (X = 0, Y = 300, Z = 0)
	def set_workorigin()	Сброс рабочей высоты (Z = 0)
	def set_acceleration()	Включение акселерометра
	def move_to()	Перемещение в заданную позицию
	def fast_move_to()	Быстрое перемещение в заданную позицию
	def get_current_position()	Вычисление текущей позиции
Задержки	def dealy_ms()	Задержка в миллисекундах
	def dealy_s()	Задержка в секундах
Модули	def set_module_kind()	Инициализация модуля
	def get_module_kind()	Распознавание модуля
Щупальца	def soft_gripper_pick()	Сжать щупальцу
	def soft_gripper_place()	Разжать щупальцу
	def soft_gripper_nature()	Возврат в исходное состояние или разжатие щупальцы
	def soft_gripper_stop()	Остановка подачи воздуха
Присоска	def air_picker_pick()	Сжать присоску
	def air_picker_place()	Разжать присоску
	def air_picker_nature()	Возврат в исходное состояние или разжатие присоски
	def air_picker_stop()	Остановка подачи воздуха
Лазер	def laser_on()	Включить лазер
	def laser_off()	Выключить лазер
Конвейер	def conveyor_belt_forward()	Прокрутить конвейер вперед
	def conveyor_belt_backward()	Прокрутить конвейер назад
	def conveyor_belt_stop()	Остановить конвейер
Направляющая	def sliding_rail_init()	Инициализирует направляющую рельсу

В комплекте поставки с Rotrics Dex ARM поставляется также специальная цифровая камера, которая дает возможность автоматизации производственной ячейки с помощью технического

зрения и специального программного обеспечения. Выполните следующие шаги для корректной работы с Rotrics Vision Terminal:

- Скачайте Rotrics Vision Terminal по следующей ссылке - <https://rotrics.com/pages/downloads>;
- Для корректной работы с программным обеспечением также необходимо дополнительно установить следующие программные библиотеки для языка программирования Python:

```
pip install opencv-python  
pip install matplotlib
```

- После открытия «Rotrics Vision Terminal», подключитесь к Rotrics Dex Arm и камере. А именно выберите соответствующий COM-порт, и нажмите кнопку Open для открытия порта. Если свободных портов нет, нажмите кнопку «Refresh» для обновления списка доступных COM-портов. Соответствующие операции проделайте также и для камеры. Обратите внимание, что камера №0 – это встроенная камера ноутбука или компьютера (Рис 2.2.6.2).

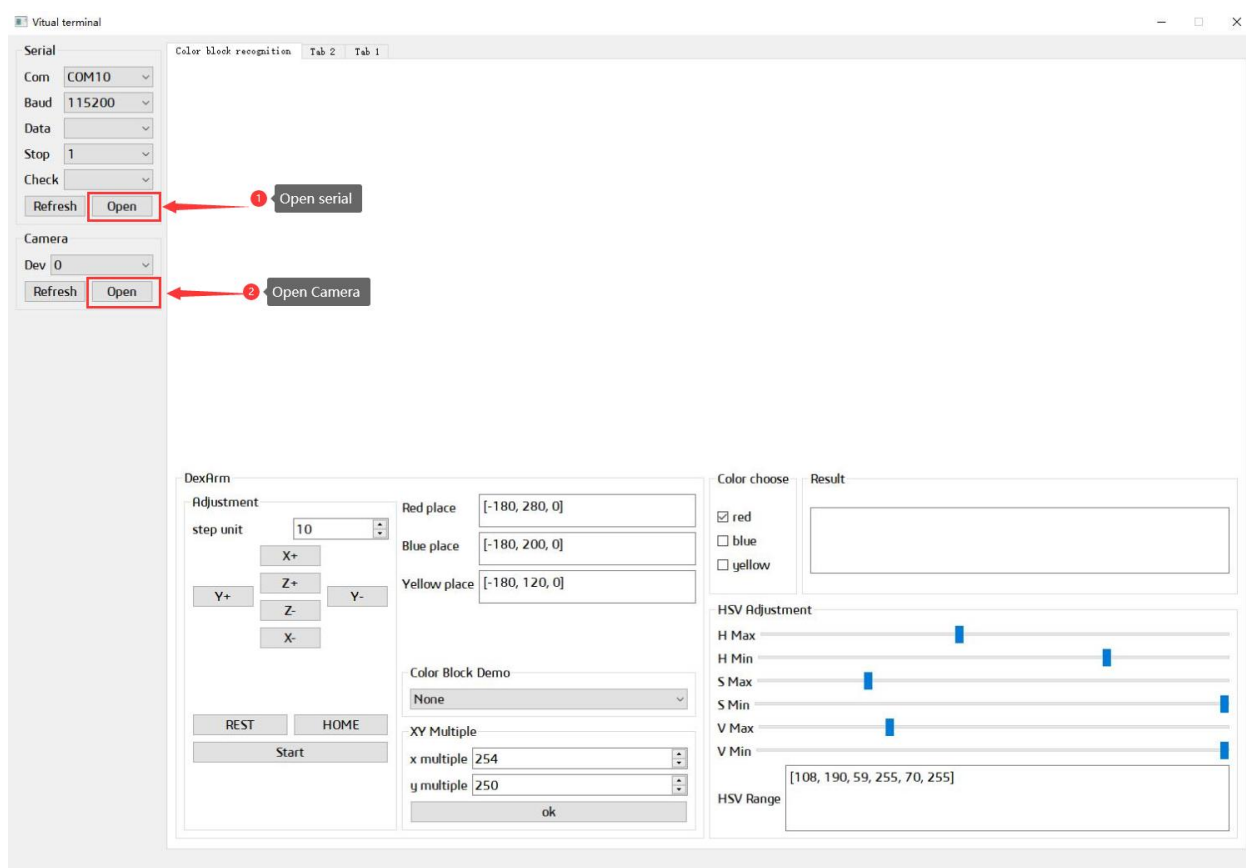


Рисунок 2.2.6.2 – Главное окно Rotrics Vision Terminal

- В нижнем левом углу, в поле «Color Block Demo», выберите функцию «Blocks». Переместите объекты для распознавания в область камеры. С помощью меню «Color choose», выберите необходимый цветовой диапазон и после сканирования области распознавания, координаты соответствующего цветового объекта будут изменены в ту или иную сторону (Рис 2.2.6.3).

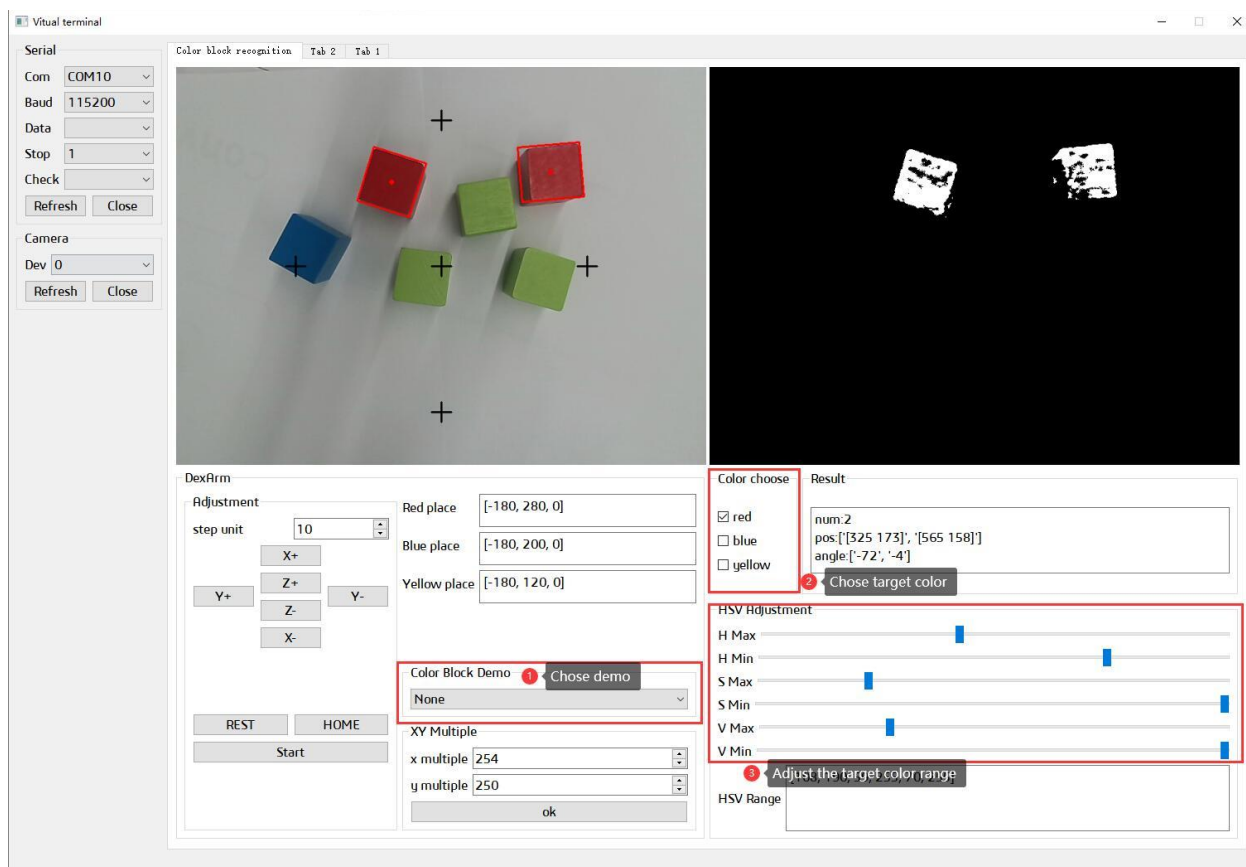


Рисунок 2.2.6.3 – Выбор режима работы

- После того, как вы выбрали цвет, используйте панель управления чтобы перемещать переместить робота в соответствующее место. После каждого перемещения, необходимо устанавливать конечную позицию для каждого цветового диапазона отдельно. При этом, после каждого распознавания объектов, необходимо вернуть робота в начальную позицию (Рис 2.2.6.4).

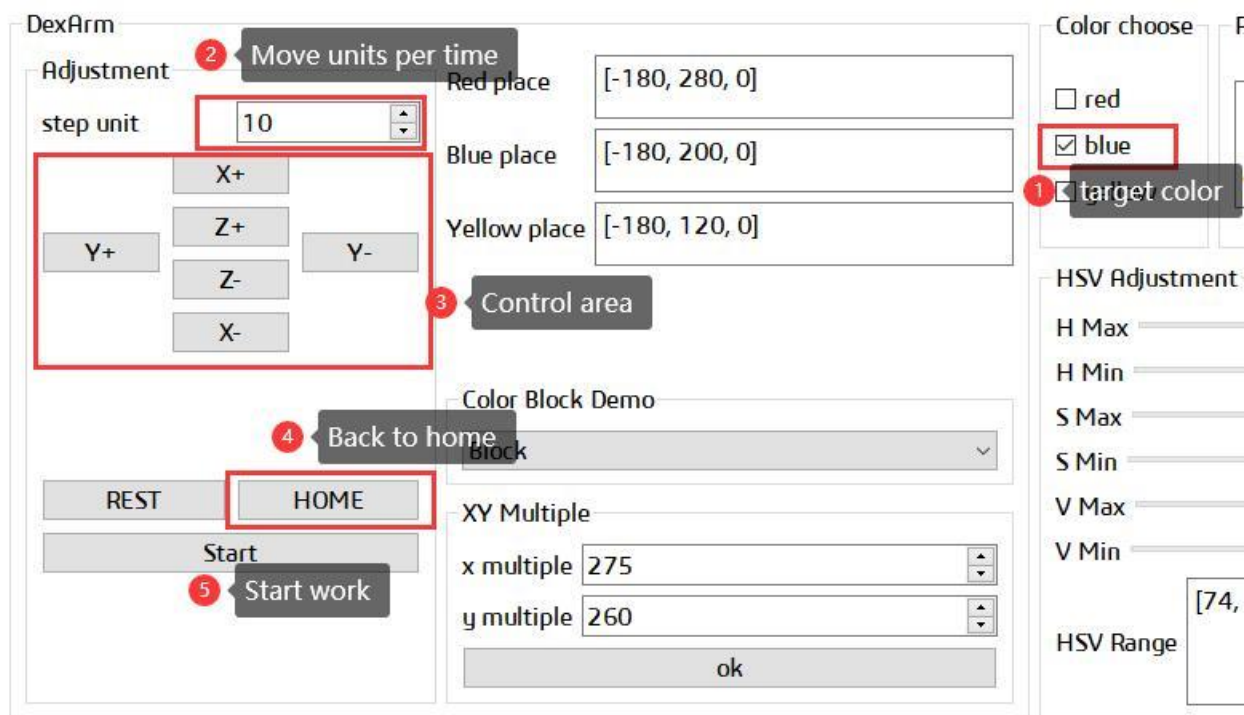


Рисунок 2.2.6.4 – Управление роботом

- После распознавания цветных объектов и диапазонов, нажмите кнопку «Start» для начала операции сортировки объектов по цвету. Если робот не смог отрегулировать центр элемента, попробуйте функцию «XY Multiple», для умножения векторов в декартовой плоскости (2D-режим, используется только оси X и Y).

ЭПИЛОГ

ДОПОЛНИТЕЛЬНЫЕ МАТЕРИАЛЫ

СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ

СПИСОК ИСПОЛЬЗОВАННЫХ ГРАФИЧЕСКИХ МАТЕРИАЛОВ