

**Rotrics DexArm**

*Руководство пользователя Rotrics DexArm*

Алматы, 2021 (v1.2-20211101-1514)

СОДЕРЖАНИЕ:

1. Введение
2. Подготовка робота
3. Движение робота
4. Лазер
5. Пневматика
6. Приложение
7. Литература

**ВВЕДЕНИЕ (1)**

***Rotrics DexArm*** – универсальный настольный робот-манипулятор для лазерной гравировки, 3D-печати и машинного зрения. Робот обладает повышенной по сравнению с другими настольными роботами точностью операций – до 0,005 мм. Данное свойство позволяет реализовать множество кропотливых операций, в которых важна точность исполнения. ******

Рисунок 1.1 – Основные и дополнительные компоненты Rotrics Dex Arm, а также сменные модули и насадки

Робот программируется на языках - G-code, C++, Python. Два последних языка программирования описывают интерфейс-библиотеку для работы с роботом. Робот ориентируется в трех плоскостях – X, Y, Z. При этом, у робота существуют ограничения в координатных плоскостях из-за конструктивных особенностей (*см. Таб. 1.1*).

|  |  |  |
| --- | --- | --- |
| *Ось X* | *Ось Y* | *Ось Z* |
| (-330: - 120) U (120: 330) | (230; 380) | (165; 127) |

Таблица 1.1 – Координатные ограничения робота Rotrics Dex Arm

*Спецификации:*

1. Точность операций: 0,005 мм;
2. Сборочные габариты: 220 x 155 x 160 мм;
3. Габариты робота: 175 x 128 x 315 мм;
4. Вес робота: 2,4 кг.

*Комплектация:*

1. Робот;
2. Модуль «Держатель для ручки»;
3. Модуль «Лазер 2,5 Вт»;
4. Модуль «3D-печать»;
5. Сенсорный экран.

*Первоначальная настройка робота:*

**Шаг 1** – Скачайте с официального сайта Rotrics программное обеспечение Rotrics Studio для взаимодействия с роботом – <https://rotrics.com>;

**Шаг 2** – Подключите робота с помощью разъема *Power Adapter* к сети 220В, и с помощью кабеля *USB Type-C <-> USB Type A*, подключите робота к компьютеру;

**Шаг 3** – После подключения робота, зайдите в меню «*Пуск*», и найдите утилиту «*Управление компьютером*» как показано на изображении ниже (*см. Рис. 1.2*);

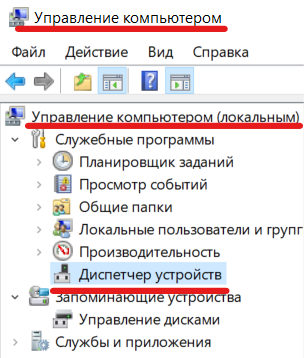


Рисунок 1.2 – Основной вид утилиты «Управление компьютером»

**Шаг 4** – В подменю «*Диспетчер устройств*», нажмите на вкладку «*Порты (COM и LPT)*». В выпадающем меню будет показан робот с последовательным портом и номером этого порта. Например: «*Устройство с последовательным портом USB (COM8)*», где *8* – номер вашего порта (*см. Рис. 1.3*);

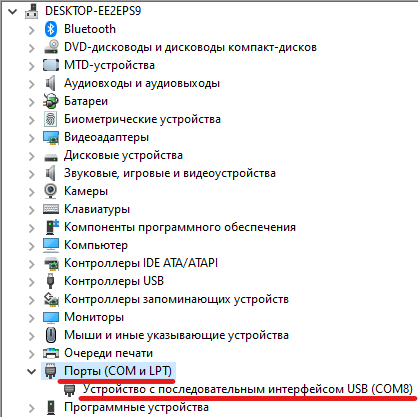


Рисунок 1.3 – Основной вид подменю «Порты (COM и LPT)»

**ПОДГОТОВКА РОБОТА (2)**

Перед взаимодействием с роботом ознакомьтесь со следующими системными требованиями:

*Системные требования*

* *Операционная система* – Windows, MacOS, Linux;
* *Python* – 3.10 и выше (Рекомендуется версия Python не ниже 3.6, однако результат может отличаться, т.к. все действия проводились на ПК с Python 3.10 (Python 3.10 не поддерживает Windows 7);
* *Git* – 2.20 и выше;
* *IDE* – VS Code, PyCharm, Emacs, Vim, … (Рекомендуется использовать Visual Studio Code или PyCharm).

*Начало работы:*

**Шаг 1** – После установки необходимого программного обеспечения, установите утилиту pip для дополнительной установки необходимых пакетов Python:

python get-pip.py

**Шаг 2** – После установки утилиты pip, установите дополнительные пакеты для корректной работы с роботом:

pip install opencv-python

pip install pyyaml

pip install pyserial

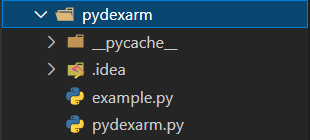
**Шаг 3** – С помощью утилиты git клонируйте репозиторий по следующему адресу:

git clone https://github.com/AndreM07/dex-arm-book

**Шаг 4** – Открываем IDE, во вкладке на верхней панели, щелкаем на меню «Открыть», выбираем папку с нашим клонированным репозиторием;

Изображение 2.1 – Иерархия проекта pydexarm

**Шаг 5** – В открывшемся дереве директории, увидим нашу папку с проектом, в котором есть два файла – «example.py» и «pydexarm.py»



Изображение 2.1 – Иерархия проекта pydexarm

**Шаг 6** – Файл «example.py» - содержит своеобразный hello-world для робота, а файл «pydexarm.py» - содержит библиотеку-интерфейс для взаимодействия с роботом на языке G-code.

**ДВИЖЕНИЕ РОБОТА (3)**

Рассмотрим основные правила задания движения робота по координатным осям OX, OY, OZ. На *Рис. 2.1,* 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

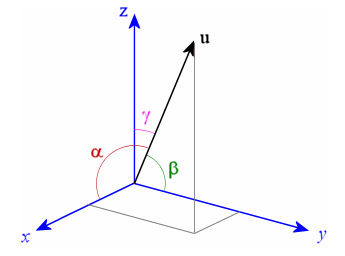


Рисунок 3.1 – Координатные векторы робота Rotrics Dex Arm

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.

*Листинг 3.1:*

**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 *листинг 3.1:*

1. Строка **« from** pydexarm **import** Dexarm **»** - включает в основной код библиотеку-интерфейс для общения с роботом при помощи связи между Python и G-code;
2. Строка **«** dexarm = Dexarm("COM8") **»** - отправляет скомпилированный код в робот на указанный COM-порт, а именно COM8. Номер порта можно узнать в главе *«Введение», Шаг 4*
3. Строка **«** 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. Строка **«** 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 2.1 и операций робота:*

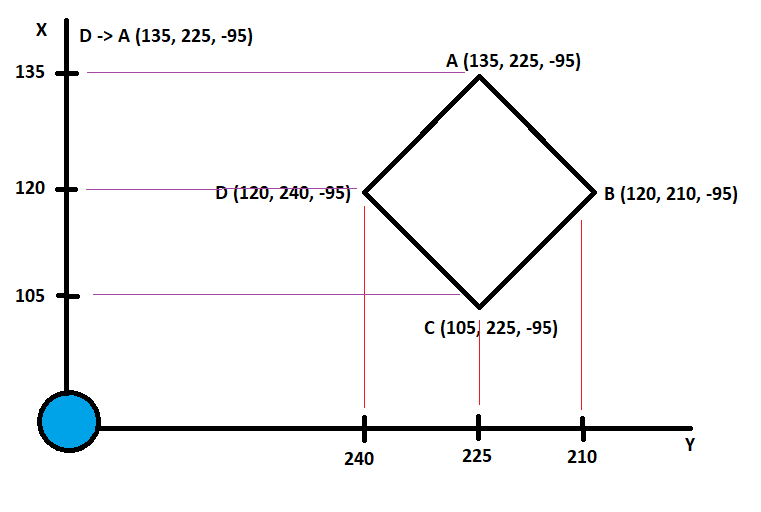


Рисунок 3.2 – Математическое объяснение операций робота в листинге 3.1

1. 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);
2. Координатные оси распределены следующим образом:

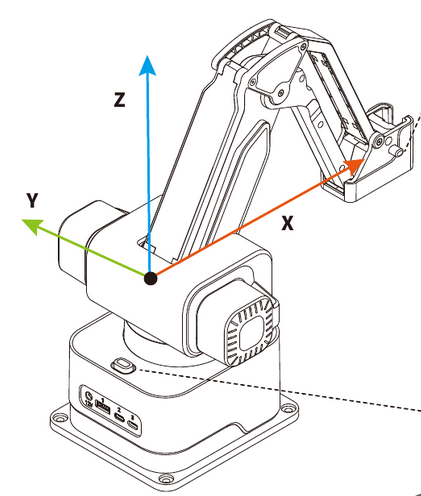


Рисунок 3.3 – Координатные оси робота в 3D-проекции

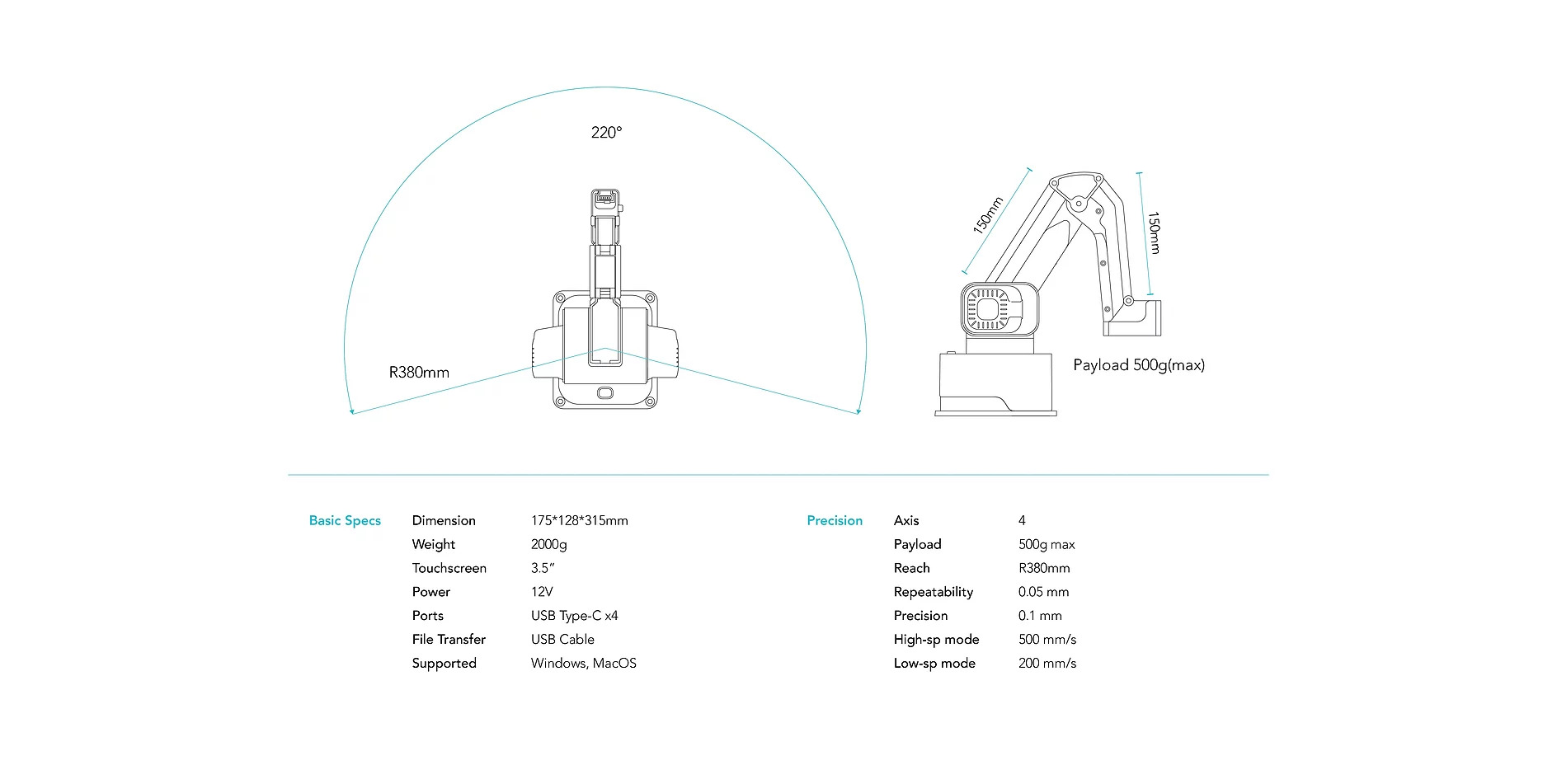
1. Таким образом, оси X и Y поменялись местами в отличие от математической модели осей, где вверху ось Y, а внизу ось X.
2. При работе с роботом, следует не забывать про конструктивные ограничения, а именно:

Рисунок 3.4 – 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);

1. При всем этом, мы должны обязательно помнить то, что нельзя выходить за рамки координат!

**ЛАЗЕР (4)**

***ВНИМАНИЕ! ПРИ РАБОТЕ С МОДУЛЕМ «ЛАЗЕР», СОБЛЮДАЙТЕ НЕОБХОДИМУЮ ТЕХНИКУ БЕЗОПАСНОСТИ, А ИМЕННО ПРИ ВКЛЮЧЕНИИ ЛАЗЕРА ОДЕВАЙТЕ СПЕЦИАЛЬНЫЕ ОТРАЖАЮЩИЕ ОЧКИ, ПОСЛЕ ВЫКЛЮЧЕНИЯ ЛАЗЕРА, ОБЯЗАТЕЛЬНО ПРОВЕТРИВАЙТЕ ОБЛАСТЬ И НЕ СОЗДАВАЙТЕ ПОЖАРООПАСНЫХ СИТУАЦИЙ! БУДЬТЕ ОСТОРОЖНЫ!***

После установки модуля «Лазер» и его надежном закреплении в месте установки, включится маленький вентилятор, предназначенный для охлаждения лазерного модуля. Введите *Листинг 4.1* в вашу IDE и посмотрите за результатом выполнения программы.

*Листинг 4.1*:

dexarm.go\_home()

dexarm.move\_to(135, 225, -70) # A

dexarm.laser\_on(1000)

dexarm.move\_to(120, 210, -70) # B

dexarm.move\_to(105, 225, -70) # C

dexarm.move\_to(120, 240, -70) # D

dexarm.move\_to(135, 225, -70) # D to A

dexarm.laser\_off()

dexarm.go\_home()

*Рассмотрим поподробнее код приведенный в Листинге 4.1:*

1. Строка «dexarm.laser\_on(1000)» - предназначена для включения лазера, в скобках функций указывается мощность, с которой лазер будет работать, мощность указывается в диапазоне от 0 до 1000;
2. Строка «dexarm.laser\_off()» - предназначена для выключения лазера.

**ПНЕВМАТИКА (5)**

После установки модуля «Пневматика» и его надежном закреплении в месте установки, включится маленький пневматический генератор воздуха, предназначенный для работы присоски и пневматической системы. После установки пневматического генератора воздуха произведите установку модуля «Присоска». («Soft Gripper» – щупальца с мягким захватом, «Air Picker» - присоска)

***Air Picker***



Введите *Листинг 5.1* в вашу IDE и посмотрите за результатом выполнения программы.

*Листинг 5.1*:

dexarm.go\_home()

dexarm.move\_to(135, 225, -15)

dexarm.air\_picker\_pick()

dexarm.dealy\_ms(1000)

dexarm.go\_home()

dexarm.air\_picker\_place()

dexarm.dealy\_ms(1000)

dexarm.air\_picker\_stop()

dexarm.air\_picker\_nature()

dexarm.go\_home()

*Рассмотрим поподробнее код приведенный в Листинге 5.1:*

1. Строка «dexarm.air\_picker\_pick()» - предназначена для откачки воздуха из присоски;
2. Строка «dexarm.air\_picker\_place()» - предназначена для закачивания воздуха из присоски;
3. Строка «dexarm.air\_picker\_stop()» - предназначена для остановки подачи воздуха;
4. Строка «dexarm.air\_picker\_nature()» - предназначена для возврата в исходное положение или откачки воздуха;

***Soft Gripper***



После установки пневматического генератора воздуха произведите установку модуля «Присоска». («Soft Gripper» – щупальца с мягким захватом, «Air Picker» - присоска)

Введите *Листинг 5.2* в вашу IDE и посмотрите за результатом выполнения программы.

*Листинг 5.2*:

dexarm.go\_home()

dexarm.fast\_move\_to(135, 225, -40) # A

dexarm.soft\_gripper\_place() # разжать щупальцу

dexarm.dealy\_ms(1000)

dexarm.soft\_gripper\_pick() # зажать щупальцу

dexarm.dealy\_ms(1000)

dexarm.go\_home()

dexarm.soft\_gripper\_place() # разжать щупальцу

dexarm.dealy\_ms(1000)

dexarm.soft\_gripper\_stop() # остановка подачи воздуха

*Рассмотрим поподробнее код приведенный в Листинге 5.1:*

1. Строка «dexarm.soft\_gripper\_pick()» - предназначена для зажатия щупальцы;
2. Строка «dexarm.soft\_gripper\_place ()» - предназначена для разжатия щупальцы;
3. Строка «dexarm.soft\_gripper\_stop()» - предназначена для остановки подачи воздуха;
4. Строка «dexarm.soft\_gripper\_nature()» - предназначена для возврата в исходное положение или откачки воздуха;

**ПРИЛОЖЕНИЕ (6)**

Приложение № 1 – Полное описание библиотек-интерфейса «pydexarm.py»:

|  |  |  |
| --- | --- | --- |
| ***Группа команд*** | ***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 |

**ЛИТЕРАТУРА (7)**

1. <https://rotrisc.com> – официальный сайт разработка робота
2. <https://manual.rotrics.com/> - онлайн-обновляемый справочник по роботу и его командам
3. <https://www.avanti-edu.tech/product/RotricsDexARM> - официальный сайт дистрибьютора робота в странах СНГ и ЕАЭС
4. <https://github.com/AndreM07/dex-arm-book> - репозиторий №1 с примерами для робота
5. <https://github.com/Rotrics-Dev/DexArm_API> - репозиторий №2 с API для робота
6. <https://github.com/Rotrics-Dev> - официальный Github профиль компании Rotrics с репозиториями для работы с роботом