# Параметры командной строки

**-port0:** указывается номер сетевого порта для управления, по умолчанию 8888

**-logfile:** указывается путь и имя файла для записи лога, по умолчанию - *simulation3dlog.txt*

Пример:

**simulation3d.exe -port:7777 logfile:d:/folder/log.txt**

# Управление программой

F10 – выход

Управление камерой: сдвиги - движение мышью удерживая левую кнопку, повороты - движение мышью удерживая правую кнопку, наезд-отъезд камеры – прокручиванием колёсика мыши. По умолчанию существует камера с именем *maincamera*

# Сетевой протокол

Cлушается порт управления.

**Диаграммы последовательности протокола управления.**

соединение

ready

Клиент

Сервер

**. . .**

команда

ready + результат

**. . .**

рассоединение

end

Пакеты передаются в формате **json**. Перед каждым пакетом находится заголовочная строка, например: 'json123:', где цифры '123' - размер в байтах пакета следующего после символа ':'

В скрипте *client0.py* – примеры управление через сокеты.

**Формат пакетов управления.**

Сообщение о готовности сервера, также приходит от сервера с результатом выполнения команды.

*ready*

{

"packet":"ready"

}

У пакета ready возможны дополнительные поля с результатами выполнения команд.

Команда завершения сеанса, ответ не приходит после этой команды.

*end*

{

"packet":"end"

}

Команда закрытия приложения, ответ не приходит после этой команды.

*close*

{

"packet":"close"

}

Команда *create camera,* создаёт и устанавливает позицию камеры.

Размеры подразумеваются в метрах.

{

"packet":"create",

"type":"camera",

"idname":"<string>", *(имя-идентификатор)*

"x0":<float>, *(координата x точки, на которую направлена камера)*

"y0":< float >, *(координата y точки, на которую направлена камера)*

"z0":< float >, *(координата z точки, на которую направлена камера)*

"x1":< float >, *(координата x камеры)*

"y1":< float >, *(координата y камеры)*

"z1":< float >, *(координата z камеры)*

}

Ответ для команды *create camera*: пакет *ready* с параметром.

{

"packet":"ready",

"ok":<целое> *(1 - успешно, 0 – если ошибка)*

}

Команда *create manipulator1*

Масса подразумевается в килограммах.

Размеры подразумеваются в метрах, углы в градусах.

Смысл размеров - в чертежике для манипулятора1.

Для звеньев, которые управляются моторами задаются коэф. САР.

САР – система автоматического регулирования

Коэффициенты: ACSProportional, ACSIntegral, ACSDifferential

В режиме кинематики позиционир. не САР, а KinematicAngularVelocity

{

"packet":"create",

"type":"manipulator1",

"idname":"<string>", *(имя-идентификатор)*

"x":<float>, *(необязательный, по умолчанию 0.0)*

"y":< float >, *(необязательный, по умолчанию 0.0)*

"z":< float >, *(необязательный, по умолчанию 0.0)*

"angle":< float >, *(необязательный, по умолчанию 0.0)*

"Kinematic":< bool >, *(необязательный, по умолчанию false)*

"KinematicAngularVelocity":< float >, *(необязательный, по умолчанию 100.0)*

"UseGravity":< bool >, *(необязательный, по умолчанию false)*

"ChassisHeight":< float >, *(необязательный, по умолчанию 1.0)*

"ChassisWidth":< float >, *(необязательный, по умолчанию 0.5)*

"RotatingplatformMass":< float >, *(необязательный, по умолчанию 1.0)*

"RotatingplatformACSProportional":< float >, *(необяз., по умолчанию 1.5)*

"RotatingplatformACSIntegral":< float >, *(необяз., по умолчанию 0.0)*

"RotatingplatformACSDifferential":< float >, *(необяз., по умолчанию 1.1)*

"RotatingplatformAngle0":< float >, *(необязательный, по умолчанию 0.0)*

"RotatingplatformAngle1":< float >, *(необязательный, по умолчанию 360.0)*

"RotatingplatformDiameter":< float >, *(необязательный, по умолчанию 0.5)*

"RotatingplatformWidth":< float >, *(необязательный, по умолчанию 0.02)*

"HolderMass":< float >, *(необязательный, по умолчанию 1.0)*

"HolderWidth":< float >, *(необязательный, по умолчанию 0.03)*

"HolderLength":< float >, *(необязательный, по умолчанию 0.14)*

"LeverMass":< float >, *(необязательный, по умолчанию 1.0)*

"LeverACSProportional":< float >, *(необязательный, по умолчанию 1.5)*

"LeverACSIntegral":< float >, *(необязательный, по умолчанию 0.0)*

"LeverACSDifferential":< float >, *(необязательный, по умолчанию 1.1)*

"LeverAngle0":< float >, *(необязательный, по умолчанию 10.0)*

"LeverAngle1":< float >, *(необязательный, по умолчанию 60.0)*

"LeverhingeMass":< float >, *(необязательный, по умолчанию 1.0)*

"LeverhingeDiameter":< float >, *(необязательный, по умолчанию 0.08)*

"LeverhingeWidth":< float >, *(необязательный, по умолчанию 0.023)*

"LeverWidth":< float >, *(необязательный, по умолчанию 0.03)*

"LeverLength":< float >, *(необязательный, по умолчанию 0.6)*

"ArmMass":< float >, *(необязательный, по умолчанию 1.0)*

"ArmACSProportional":< float >, *(необязательный, по умолчанию 1.5)*

"ArmACSIntegral":< float >, *(необязательный, по умолчанию 0.0)*

"ArmACSDifferential":< float >, *(необязательный, по умолчанию 1.1)*

"ArmAngle0":< float >, *(необязательный, по умолчанию 60.0)*

"ArmAngle1":< float >, *(необязательный, по умолчанию 120.0)*

"ArmhingeMass":< float >, *(необязательный, по умолчанию 1.0)*

"ArmhingeDiameter":< float >, *(необязательный, по умолчанию 0.08)*

"ArmhingeWidth":< float >, *(необязательный, по умолчанию 0.023)*

"ArmWidth":< float > *(необязательный, по умолчанию 0.03)*

"ArmLength":< float > *(необязательный, по умолчанию 0.6)*

}

Ответ для команды *create manipulator1* : пакет *ready* с параметром.

{

"packet":"ready",

"ok":<целое> *(1 - успешно, 0 – если ошибка)*

}

Команда *create manipulator2*

Масса подразумевается в килограммах.

Размеры подразумеваются в метрах, углы в градусах.

Смысл размеров - в чертежиках для манипулятора2, захвата1-пальца1.

Для звеньев, которые управляются моторами задаются коэф. САР.

САР – система автоматического регулирования

Коэффициенты: ACSProportional, ACSIntegral, ACSDifferential

В режиме кинематики позиционир. не САР, а KinematicAngularVelocity

{

"packet":"create",

"type":"manipulator1",

"idname":"<string>", *(имя-идентификатор)*

"x":<float>, *(необязательный, по умолчанию 0.0)*

"y":< float >, *(необязательный, по умолчанию 0.0)*

"z":< float >, *(необязательный, по умолчанию 0.0)*

"angle":< float >, *(необязательный, по умолчанию 0.0)*

"Kinematic":< bool >, *(необязательный, по умолчанию false)*

"KinematicAngularVelocity":< float >, *(необязательный, по умолчанию 100.0)*

"UseGravity":< bool >, *(необязательный, по умолчанию false)*

"ChassisHeight":< float >, *(необязательный, по умолчанию 1.0)*

"ChassisWidth":< float >, *(необязательный, по умолчанию 0.5)*

"RotatingplatformMass":< float >, *(необязательный, по умолчанию 1.0)*

"RotatingplatformACSProportional":< float >, *(необяз., по умолчанию 1.5)*

"RotatingplatformACSIntegral":< float >, *(необяз., по умолчанию 0.0)*

"RotatingplatformACSDifferential":< float >, *(необяз., по умолчанию 1.1)*

"RotatingplatformAngle0":< float >, *(необязательный, по умолчанию 0.0)*

"RotatingplatformAngle1":< float >, *(необязательный, по умолчанию 360.0)*

"RotatingplatformDiameter":< float >, *(необязательный, по умолчанию 0.5)*

"RotatingplatformWidth":< float >, *(необязательный, по умолчанию 0.02)*

"LeverhingeMass":< float >, *(необязательный, по умолчанию 1.0)*

"LeverhingeDiameter":< float >, *(необязательный, по умолчанию 0.08)*

"LeverMass":< float >, *(необязательный, по умолчанию 1.0)*

"LeverACSProportional":< float >, *(необязательный, по умолчанию 1.5)*

"LeverACSIntegral":< float >, *(необязательный, по умолчанию 0.0)*

"LeverACSDifferential":< float >, *(необязательный, по умолчанию 1.1)*

"LeverWidth":< float >, *(необязательный, по умолчанию 0.03)*

"LeverLength":< float >, *(необязательный, по умолчанию 0.6)*

"ArmMass":< float >, *(необязательный, по умолчанию 1.0)*

"ArmACSProportional":< float >, *(необязательный, по умолчанию 1.5)*

"ArmACSIntegral":< float >, *(необязательный, по умолчанию 0.0)*

"ArmACSDifferential":< float >, *(необязательный, по умолчанию 1.1)*

"ArmAngle0":< float >, *(необязательный, по умолчанию 60.0)*

"ArmAngle1":< float >, *(необязательный, по умолчанию 120.0)*

"ArmhingeMass":< float >, *(необязательный, по умолчанию 1.0)*

"ArmhingeDiameter":< float >, *(необязательный, по умолчанию 0.08)*

"ArmhingeWidth":< float >, *(необязательный, по умолчанию 0.023)*

"ArmWidth":< float >, *(необязательный, по умолчанию 0.03)*

"ArmLength":< float >, *(необязательный, по умолчанию 0.25)*

"HolderMass":< float >, *(необязательный, по умолчанию 1.0)*

"HolderWidth":< float >, *(необязательный, по умолчанию 0.03)*

"HolderLength":< float >, *(необязательный, по умолчанию 0.14)*

"HoldersDistance":< float >, *(необязательный, по умолчанию 0.179)*

"WheelhingeMass":< float >, *(необязательный, по умолчанию 1.0)*

"WheelhingeWidth":< float >, *(необязательный, по умолчанию 0.046)*

"WheelMass":< float >, *(необязательный, по умолчанию 1.0)*

"WheelDiameter":< float >, *(необязательный, по умолчанию 0.32)*

"WheelWidth":< float >, *(необязательный, по умолчанию 0.023)*

"WheelLever":< float >, *(необязательный, по умолчанию 0.12)*

"LeverAngle0":< float >, *(необязательный, по умолчанию 10.0)*

"LeverAngle1":< float >, *(необязательный, по умолчанию 60.0)*

"CaptureConnectorMass":< float >, *(необязательный, по умолчанию 0.2)*

"CaptureConnectorDiameter":< float >, *(необязательный, по умолчанию 0.1)*

"CaptureConnectorWidth":< float >, *(необязательный, по умолчанию 0.02)*

"CaptureArmMass":< float >, *(необязательный, по умолчанию 0.2)*

"CaptureArmKinematicAngularVelocity":< float >, *(необяз., по умолч. 100.0)*

"CaptureArmACSProportional":< float >, *(необязательный, по умолчанию 1.5)*

"CaptureArmACSIntegral":< float >, *(необязательный, по умолчанию 0.0)*

"CaptureArmACSDifferential":< float >, *(необязательный, по умолчанию 1.1)*

"CaptureArmAngle0":< float >, *(необязательный, по умолчанию 0.0)*

"CaptureArmAngle1":< float >, *(необязательный, по умолчанию 360.0)*

"CaptureArmDiameter":< float >, *(необязательный, по умолчанию 0.03)*

"CaptureArmLength":< float >, *(необязательный, по умолчанию 0.4)*

"CaptureClamphingeMass":< float >, *(необязательный, по умолчанию 0.2)*

"CaptureClamphingeDiameter":< float >, *(необязательный, по умолчанию 0.1)*

"CaptureClamphingeWidth":< float >, *(необязательный, по умолчанию 0.03)*

"CaptureClampMass":< float >, *(необязательный, по умолчанию 0.1)*

"CaptureClampKinematicAngularVelocity":< float >, *(необяз., по умолч. 100.0)*

"CaptureClampACSProportional":< float >, *(необяз., по умолчанию 1.5)*

"CaptureClampACSIntegral":< float >, *(необязательный, по умолчанию 0.0)*

"CaptureClampACSDifferential":< float >, *(необяз., по умолчанию 1.1)*

"CaptureClampAngle0":< float >, *(необязательный, по умолчанию 240.0)*

"CaptureClampAngle1":< float >, *(необязательный, по умолчанию 120.0)*

"CaptureClampDiameter":< float >, *(необязательный, по умолчанию 0.06)*

"CaptureClampWidth":< float >, *(необязательный, по умолчанию 0.028)*

"CaptureClampConnectorKinematicAngularVelocity":< float >, *(н. по ум. 100.0)*

"CaptureClampConnectorACSProportional":< float >, *(необяз., по умолч. 1.5)*

"CaptureClampConnectorACSIntegral":< float >, *(необяз., по умолч. 0.0)*

"CaptureClampConnectorACSDifferential":< float >, *(необяз., по умолч. 1.1)*

"CaptureClampConnectorAngle0":< float >, *(необяз., по умолчанию 0.0)*

"CaptureClampConnectorAngle1":< float >, *(необяз., по умолчанию 360.0)*

"FingerKinematic":< bool >, *(необязательный, по умолчанию true)*

"FingerConnectorMass":< float >, *(необязательный, по умолчанию 0.005)*

"FingerConnectorHeight":< float >, *(необязательный, по умолчанию 0.001)*

"FingerRibHeight":< float >, *(необязательный, по умолчанию 0.005)*

"FingerSectionMass":< float >, *(необязательный, по умолчанию 0.005)*

"FingerSectionHeight":< float >, *(необязательный, по умолчанию 0.015)*

"FingerSectionWidth":< float >, *(необязательный, по умолчанию 0.025)*

"FingerSectionThick":< float >, *(необязательный, по умолчанию 0.005)*

"FingerSectionKinematicAngularVelocity":< float >, *(необяз., по умолч. 100.0)*

"FingerSectionACSProportional":< float >, *(необяз., по умолчанию 1.5)*

"FingerSectionACSIntegral":< float >, *(необязательный, по умолчанию 0.0)*

"FingerSectionACSDifferential":< float >, *(необяз., по умолчанию 1.1)*

"FingerSectionAngle0":< float >, *(необязательный, по умолчанию 340.0)*

"FingerSectionAngle1":< float >, *(необязательный, по умолчанию 0.0)*

"FingerSectionCount":< int >, *(необязательный, по умолчанию 4)*

"FingerDown":< bool >, *(необязательный, по умолчанию true)*

}

Ответ для команды *create manipulator2* : пакет *ready* с параметром.

{

"packet":"ready",

"ok":<целое> *(1 - успешно, 0 – если ошибка)*

}

Команды *create thing, create table -* предметы и столы (конвейры)

{

"packet":"create",

"type":"thing",

"model":< string > , *(имеется список имён возможных предметов и столов)*

"idname":"<string>", *(имя-идентификатор)*

"x":< float >, *(координата x)*

"y":< float >, *(координата y)*

"z":< float >, *(координата x)*

"ex":< float >, *(угол Эйлера для x)*

"ey":< float >, *(угол Эйлера для y)*

"ez":< float >, *(угол Эйлера для z)*

"scale":<float>, *(коэффициент масштабирования)*

"kinematic":< bool > *(если false предмет падает вниз до поверхности)*

}

Ответ для команд *create thing, create table*: пакет *ready* с параметром.

{

"packet":"ready",

"ok":<целое> *(1 - успешно, 0 – если ошибка)*

}

Команда *delete*, удаляет камеру, устройство или предмет*.*

{

"packet":"delete",

"idname":"<string>", *(имя-идентификатор)*

}

Ответ для команды *delete* : пакет *ready* с параметром.

{

"packet":"ready",

"ok":<целое> *(1 - успешно, 0 – если ошибка)*

}

Команда *clear*, удаляет все камеры, устройства и предметы*,* инициализирует главную камеру *(maincamera)*

{

"packet":"clear",

}

Ответ для команды *clear* : пакет *ready* с параметром.

{

"packet":"ready",

"ok":<целое> (1 - успешно, 0 – если ошибка)

}

Команда *setpos* для *manipulator1* и *manipulator2* устанавливает позиции плечей*.* Углы – в градусах. Смысл - в чертежиках для манипуляторов 1 и 2. "a3", "a4", “a5” не игнорируются только для манипулятора2 и при установке параметра FingerDown = False.

{

"packet":"setpos",

"idname":"<string>", *(имя-идентификатор)*

"a0":< float >, *(угол поворота платформы)*

"a1":< float >, *(угол поворота нижней секции)*

"a2":< float >, *(угол поворота верхней секции)*

"a3":< float >, *(угол вращения верхней секции)*

"a4":< float >, *(угол поворота захвата)*

"a5":< float >, *(угол вращения площадки захвата)*

"a6":< float >, *(признак сжатия захвата)*

"a7":< float >, *(игнорируется)*

}

Команда *setpos* для *thing* устанавливает позицию.

{

"packet":"setpos",

"idname":"<string>", *(имя-идентификатор)*

"a0":< float >, *(координата x)*

"a1":< float >, *(координата y)*

"a2":< float >, *(координата x)*

"a3":< float >, *(угол Эйлера для x)*

"a4":< float >, *(угол Эйлера для y)*

"a5":< float >, *(угол Эйлера для z)*

"a6":<float>, *(коэффициент масштабирования)*

"a7":< float >, *(0 или 1, если 0 предмет падает вниз до поверхности)*

}

***Для команды setpos ответ не приходит!***

Команда *setcamera*, устанавливает позицию камеры.

Размеры подразумеваются в метрах.

{

"packet":"setcamera",

"idname":"<string>", *(имя-ид., необяз.по умоланию. – активная камера)*

"x0":<float>, *(координата x точки, на которую направлена камера)*

"y0":< float >, *(координата y точки, на которую направлена камера)*

"z0":< float >, *(координата z точки, на которую направлена камера)*

"x1":< float >, *(координата x камеры)*

"y1":< float >, *(координата y камеры)*

"z1":< float >, *(координата z камеры)*

}

Ответ для команды *setcamera*: пакет *ready* с параметром.

{

"packet":"ready",

"ok":<целое> *(1 - успешно, 0 – если ошибка)*

}

Команда *activecamera*, устанавливает вывод с камеры.

{

"packet":"activecamera",

"idname":"<string>", *(имя-идентификатор)*

}

Ответ для команды *activecamera*: пакет *ready* с параметром.

{

"packet":"ready",

"ok":<целое> *(1 - успешно, 0 – если ошибка)*

}

Команда *shoot*, делает снимок экрана (вида активной камеры).

{

"packet":"shoot",

}

Ответ для команды *shoot*: пакет *ready* с параметром.

{

"packet":"ready",

"ok":<целое> *(1 - успешно, 0 – если ошибка)*

"base64jpg":<строка> *файл jpg закодированный в base64*

}

Команда *gripped*, запрашивает у манипулятора есть ли захват предмета*.*

{

"packet":"gripped",

"idname":"<string>", *(имя-идентификатор)*

}

Ответ для команды *gripped*: пакет *ready* с параметрами.

{

"packet":"ready",

"ok":<целое> *(1 - успешно, 0 – если ошибка)*

"gripped":<bool> *(true – есть захват предмета)*

}

Команда *transform*, запрашивает позицию, поворот и габариты предмета или стола(конвейра)*.*

{

"packet":"transform",

"idname":"<string>", *(имя-идентификатор)*

}

Ответ для команды *transform*: пакет *ready* с параметрами.

Позиция соответствует геометрическому центру. Габариты – ориентировочные т.е. не соответствуют абсолютно точно реальным.

{

"packet":"ready",

"ok":<целое> *(1 - успешно, 0 – если ошибка)*

"x":<float> *(координата x)*

"y":< float >, *(координата y)*

"x":< float >, *(координата x)*

"ex":< float >, *(угол Эйлера для x)*

"ey":< float >, *(угол Эйлера для y)*

"ez":< float >, *(угол Эйлера для z)*

"sx":< float >, *(габарит по оси x без учёта поворота, с учётом масштаба)*

"sy":< float >, *(габарит по оси y без учёта поворота, с учётом масштаба)*

"sz":< float >, *(габарит по оси z без учёта поворота, с учётом масштаба)*

"scale":<float>, *(коэффициент масштабирования)*

}

Имеющиеся предметы - модели для команды create thing:

|  |
| --- |
| Cube  bottle1  juice1  coke1  box1 |

Имеющиеся столы - модели для команды create table:

|  |
| --- |
| conveyor1  conveyor2 |