

# OXD\_TCP

Описание протокола «Open XD TCP»  
v. 0.0.05a

## Цель протокола

Организовать двустороннюю передачу данных между приложением на базе игрового движка (ниже, условно называем как **клиент**) и приложением, которое управляет динамической платформой (далее условно называем как **сервер**).

Обмен данных осуществляется по средствам TCP протокола. Адрес и порт соединения должен быть настраиваемый. По умолчанию используется IP 127.0.0.1 и порт 11000.

Протокол состоит из набора сообщений, которые могут распознаются как команды, отчеты, поток данных.

## Команды получаемые клиентом

Команда	Параметры	Описание
Управление основным треком анимации (камера или транспорт)		
<b>START</b>		Запуск проигрывания
<b>STOP</b>		Остановка с возвратом на начало проигрывания
<b>PAUSE</b>		Остановка
<b>QUIT</b>		Закрытие клиента, выход из программы
Управление уровнями		
<b>LEVELS</b>		Возвращает список доступных уровней в формате JSON, пример: <b>{'Super Bomb','Fly Fire','Mad Attack'}</b>
<b>LOAD_LEVEL</b>	Имя уровня	Загружает указанный уровень. Имя уровня можно получить при помощи <b>LEVELS_GET</b> , если уровень не найден,

		возвращает <b>LEVEL name NOT_FOUND</b>
Справочная информация		
<b>NAME</b>		Возвращает имя приложения
<b>TITLE</b>		Возвращает заголовок приложения
<b>DESCRIPTION</b>		Возвращает описание приложения
<b>AUTHOR</b>		Возвращает имя автора приложения
<b>ENGINE</b>		Возвращает имя игрового движка, на базе которого построено приложение
<b>OXD_TCP</b>		Возвращает версию протокола OXD_TCP
<b>ID</b>		Идентификационный номер или контрольная сумма
<b>OCULOS_SDK</b>		Возвращает номер версии SDK Oculus Rift, если не поддерживает, возвращает <b>NONE</b>
Настройки клиента		
<b>DEBUG</b>		Устанавливает или отменяет режим отладки TCP клиента. В этом режиме на экран выводятся строки, получаемые или отправляемые серверу. В некоторых случаях функция может быть не доступной, например если приложение на базе UE4, собрано без параметра -Debug.
<b>CMD</b>	Любая поддерживаемая консольная команда, например запрос для UE4: <b>CMD quit</b> , приведет к закрытию приложения.	Запуск консольных команд клиента. В некоторых случаях часть консольных команд может быть не доступна или приводить к нежелательным последствиям. Например клиент на базе UE4, собранный без команды -Debug, может зависнуть, получив команду <b>CMD Stat FPS</b>
<b>MODE_SET</b>	<b>SYNCH_FRAME STREAM</b> (возможно расширение списка режимов)	Установка режима работы клиента. В режиме <b>SYNCH_FRAME</b> клиент будет возвращать опорные сигналы, для синхронизации с уже прописанным треком движения.

		<p>В режиме <b>STREAM</b>, клиент будет возвращать координаты перемещения, и в первом и во втором случае интервал передачи данных можно настраивать (см. ниже <b>TIMER_GET</b>, <b>TIMER_SET</b>)</p> <p>Ответом на изменение режима будет строка <b>MODE SYNCH_FRAME</b> или <b>MODE STREAM</b>, а если режим не поддерживается возвращает <b>MODE NOT_SUPPORT</b>. Подробнее о данных передаваемых в этих режимах см. ниже.</p>
<b>MODE_GET</b>		Команда возвращает строку <b>MODE SYNCH_FRAME</b> или <b>MODE STREAM</b> , для того, чтобы проверить, какой режим активен в данный момент.
<b>SCALE_GET</b>		Возвращает переменную масштабирования, по умолчанию — 1.0. Пример ответа: <b>SCALE 1.0</b> . Масштабирование актуально только для <b>MODE STREAM</b>
<b>SCALE_SET</b>	Переменная <b>float</b>	Устанавливает размер масштаба. Параметр влияет на размер передаваемых координат.
<b>TIMER_GET</b>		Возвращает значение интервала отправки данных, пример ответа: <b>TIMER 1.0</b> , по умолчанию интервал равен 1.0 (секунд)
<b>TIMER_SET</b>	Переменная <b>float</b>	Устанавливает интервал отправки данных, по умолчанию значение = 1.0
<b>SAVE_CONFIG</b>		Сохраняет значения измененных параметров: <b>TIMER, SCALE, MODE, DEBUG</b>
<b>FULLSCREEN</b>		Переход в полноэкранный режим и обратно
<b>DIMENSION</b>	<b>[width]x[height][w   f]</b>	<p>Изменяет разрешение экрана, пример:</p> <p><b>DIMENSION 1920x1080f</b> – установит разрешение 1920x1080 с полноэкранным режимом,</p> <p><b>DIMENSION 800x600w</b> – установит разрешение 800x600 в</p>

		оконном режиме.
Данные в режиме <b>SYNCH_FRAME</b>		
Данные отправляются клиентом через промежуток времени, установленный при помощи <b>TIMER_SET</b> . Передача начинается после отправки команды <b>START</b> и заканчивается после передачи команд <b>STOP, PAUSE</b>  <b>{F:[int],DT:[float],T:[float]}</b>		Набор величин в формате JSON, где: <ul style="list-style-type: none"> <li>• <b>F</b> – номер кадра (frame)</li> <li>• <b>DT</b> – промежуток времени между пред идущим вызовом (delta time) в секундах.</li> <li>• <b>T</b> – общее время с момента запуска проигрывания в секундах</li> </ul> Пример: <b>{F:125,DT:1.002856,T:5.678946}</b>
Данные в режиме <b>STREAM</b>		
Данные отправляются клиентом через промежуток времени, установленный при помощи <b>TIMER_SET</b> . Передача начинается после отправки команды <b>START</b> и заканчивается после передачи команд <b>STOP, PAUSE</b>  <b>{DT:[float],X:[float],Y:[float],Z:[float],R:[float],P:[float],W:[float]}</b>		Набор величин в формате JSON, где по умолчанию: <ul style="list-style-type: none"> <li>• <b>DT</b> – промежуток времени между пред идущим вызовом (delta time)</li> <li>• <b>X</b> – изменение координаты X за промежуток <b>DT</b></li> <li>• <b>Y</b> – изменение координаты Y за промежуток <b>DT</b></li> <li>• <b>Z</b> – изменение координаты Z за промежуток <b>DT</b></li> <li>• <b>R</b> – текущее значение Roll в градусах</li> <li>• <b>P</b> - текущее значение Pitch в градусах</li> <li>• <b>W</b> – текущее значение Yaw в градусах</li> </ul> Пример: <b>{DT:1.005895,X:1.556446,Y:0.523495,Z:0.413259,R:5.123495,P:20.493178,W:1.379545}</b> см. ниже, как перенастроить вывод для <b>STREAM</b>
Настройка режима <b>STREAM</b>		
<b>LOCATION</b>	<b>DELTA GLOBAL VECTOR</b>	TODOO
<b>ROTATION</b>	<b>DELTA</b>	TODOO

	<b>GLOBAL VECTOR</b>	
Управление эффектами		
<p>Данные отправляются клиентом в те моменты, которые прописаны сценарием приложения.</p> <p>Передача возможна только после отправки команды <b>START</b> и заканчивается после передачи команд <b>STOP, PAUSE</b></p> <p><b>{FX:[int],P:[float]:T:[float]}</b></p> <p><b>Пример.</b> По пути следования попадаем в водопад, триггеры расставленные в приложении срабатывают и отправляют серверу строку: <b>{FX:3,P:1.0:T:3.0}</b> Что должно интерпретироваться сервером как: «Включить эффект брызги на 3 секнды на полной мощности». Если устройство создающее эффекты, не управляет мощностью, то поступить можно несколькими способами:</p> <ol style="list-style-type: none"> <li>1. Игнорировать мощность.</li> <li>2. Использовать дискредитацию, давать серию импульсов с определенной длительностью и частотой.</li> </ol>		<p>Строка в формате JSON, где:</p> <ul style="list-style-type: none"> <li>• <b>FX</b> — номер эффекта (от 0 до 255) <ol style="list-style-type: none"> <li>1. <b>WIND</b> (ветер)</li> <li>2. <b>LIGHT</b> (стробоскоп)</li> <li>3. <b>SPRAY</b> (брызги)</li> <li>4. <b>SMOKE</b> (дым)</li> <li>5. <b>SNAKE</b> (змеи, «мышинные хвостики»)</li> <li>6. <b>SOAP</b> (пузыри)</li> <li>7. <b>SNOW</b> (снег)</li> <li>8. <b>SHAK</b> (вибрация)</li> </ol> </li> </ul> <p>Эффекты от 9 номера до 255 не зарезервированы и могут быть использованы как угодно. Для того, чтобы пояснить как работать с новыми эффектами, или изменить номера эффектов, используйте команду <b>FX_MAP</b></p> <ul style="list-style-type: none"> <li>• <b>P</b> — мощность (от 0 до 1)</li> <li>• <b>T</b> — промежуток времени (секунды) в котором работает эффект</li> </ul>
<b>FX_MAP</b>		<p>Возвращает список используемых эффектов и их карту (map) в формате JSON</p> <p>Например: <b>{WIND:1,SPRAY:3,SMOKE:23,LASER:5}</b></p> <p>Из этой строки видно, что используется 4 эффекта.</p>

		<p>Из них «ветер» и «брызги» не изменились, «дым» занял номер 23 (что не приветствуется, но возможно), и появился новый эффект <b>LASER</b>.</p> <p>В случае, если производитель контента грубо путает установленные номера (необоснованный remapping), издатель может высказать претензии.</p> <p>В случае, если производители контента добавляют новые эффекты, они обязаны предоставить их описание вместе с приложением, чтобы все понимали, что такое в данном случае <b>LASER</b>, можно ли его использовать и как. В противном случае эффект игнорируется.</p>
--	--	---