# Протокол обмена между ВС тренажера и СКГИ

# Введение

Данный документ описывает протокол и дисциплину обмена информацией между ведущей вычислительной системой и ведомым генератором изображения в тренажере-симуляторе железнодорожной станции.

Как само описание протокола, так и его рекоммендуемая реализация делится на несколько частей (уровней).

Отдельно описаны и реализованы:

* оборудование (физическая часть)
* транспортная часть, втч механизм установления соединения
* общий формат пакета
* смысловое наполнение пакета (информационная часть)

Предполагается, что они лишь минимально зависимы друг от друга, что позволяет модифицировать одну подсистему не затрагивая существенно функционирование остальных, а также реализацию разными исполнителями.

Данный протокол может уточняться по обоюдному согласию заинтересованных сторон.

Серым фоном в настоящем документе помечены нереализуемые в данной версии части протокола.

# Термины

*ВС Тренажера* – вычислительная система тренажера. Моделирующая система. Включает в себя компьютер (компьютеры) и программное обеспечение, а также дополнительное аппаратное обеспечение (пульты, рукоятки итп) задействованные в расчете модели функционирования *моделируемой сцены* (железнодорожной станции, сортировочной горки итд).

*КГИ – компьютерный генератор изображения. Тоже самое СВО (Система отображения Визуальной Обстановки)*. Включает в себя компьютер (компьютеры) и программное обеспечение, и производит визуальное отображение моделируемой сцены.

ВС Тренажера и КГИ объединены каналом связи (локальной сетью), по которому они обмениваются данными по описываемому *Протоколу.*

*Абонент* – другая сторона в соединении. Для КГИ это ВС Тренажера и наоборот

*Моделируемая сцена* – весь в совокупности моделируемый объект (жд станция, сортировочная горка), или его модель, в зависимости от контекста.

*Объект* – часть моделируемой сцены, состояние которой передается в СКГИ посредством протокола. Это, например, стрелочный перевод или железнодорожный вагон.

*Сеанс тренировки* – Период от инициализации всех систем до следующей переинициализации, в течение которого происходит собственно работа тренажера.

# Физическая часть (оборудование)

Для обмена данными используется локальная сеть на базе Ethernet с пропускной способностью не менее 100 мегабит или аналогичная с не худшим временем отклика.

Для надежной работы тренажера необходимо, чтобы ВС Тренажера и КГИ находились на одном и том же логическом сегменте локальной сети, иначе говоря, нужно, чтобы друг от друга они были отделены только Ethernet Switch (Hub). Не допускается наличие роутеров, построенных на базе PC, медленных (< 100 мегабит) сегментов итп.

Это требование является критичным для обеспечения минимального времени отклика тренажера.

Во время тренировки не допускается использование задействованного сегмента локальной сети для передачи не связанных с тренировкой данных, копирования файлов итп.

Это требование является критичным, как для обеспечения безсбойной работы тренажера, так и для обеспечения гарантированного времени отклика.

В случае, если в состав ВС Тренажера и/или СКГИ входят несколько компьютеров, допускается нахождение их на том же самом сегменте и использование этого сегмента для обмена внутренними данными между ними в ходе тренировки.

Однако, разработчики как ВС Тренажера, так и СКГИ, должны уделить внимание оценке объемов передаваемых данных и их минимизации.

# Формат пакета

*Пакет* – блок данных, передаваемый между ВС Тренажера и КГИ.

Порядок байт в числовых значениях в пакете – Little Endian, иначе говоря, наименее значащий байт числа имеет наименьший адрес, как принято в архитектуре x86.

Ниже используются следующие обозначения для базовых типов данных:

WORD – 2 bytes unsigned integer

DWORD – 4 bytes unsigned integer

double – 8 bytes IEEE floating point

int – 4 bytes signed integer

char – 1byte signed integer

BYTE – 1 byte unsigned integer

ASCIIZ – 0-terminated character string

Пакет состоит из записей.

Запись состоит из тега (4 байта), длины (4 байта) и тела записи:

DWORD tag;

DWORD length

………… (тело записи)

Тег идентифицирует запись.

Длина включает в себя длину заголовка, поэтому минимальное значение поля длины составляет 8.

Записи могут быть вложены, то есть тело записи, помимо других данных может включать другие записи.

Сам пакет являтся записью, содержащей все остальные.

При описании формата пакета следующая нотация обозначает массив, несколько повторяющихся однородных элементов:

DWORD n;

Record records[n];

Некоторые данные в пакете умышленно сделаны избыточными, ~~данные,~~ что позволяет:

* сделать некоторые операции над пакетом не зависящими от конкретных данных содержащихся в нем
* проверить пакет на корректность при отладке
* сделать формат легко расширяемым

Так, например, все записи содержат поле «длина записи», хотя для некоторых из них это значение однозначно определяется тегом записи.

Максимальная длина пакета теоретически ограничена 4 гигабайтами, но на практике должна быть ограничена и оговорена таким образом, чтобы пакет гарантированно вмещал в себя полное состояние моделируемой сцены.

Следует приложить усилия к тому, чтобы подсистемы как ВС Тренажера, так и СКГИ по возможности не использовали буфера фиксированного размера для хранения и передачи пакета.

Следует заметить, что как информационная, так и транспортная часть оперируют понятием пакета. Это разные пакеты, вернее информационный пакет содержится в транспортном, или, более подробно, формирование пакетов и обмен ими происходит следующим образом.

Информационная часть формирует пакет, который передается транспортной части для отправки. Транспортная часть добавляет к информационному пакету необходимые ей данные, получая таким образом транспортный пакет, после чего отправляет пакет по сети абоненту. Транспортная часть абонента разворачивает пакет в обратном порядке, передавая информационной части абонента полученный информационный пакет.

Однако, как информационный пакет, так и транспортный пакет, имеют один и тот же общий формат, описанный выше.

# Транспортная часть

В качестве транспортного протокола используется протокол TCP.

ВС Тренажера является клиентом TCP, а СКГИ – сервером ТСР.

При этом допускается произвольный порядок запуска ВС Тренажера и СКГИ.

IP-адреса и порты для подсоединения задаются в конфигурационных файлах ВС Тренажера и СКГИ.

Как ВС Тренажера, так и КГИ должны автоматически детектировать отсоединение абонента, и пытаться восстановить соединение при его разрыве, чем бы не был вызван разрыв соединения, перезагрузкой абонента или сбоями в сети.

Это позволяет оперативно перезагружать программное обеспечение, перезапускать компьютеры итп без синхронного и согласованного перезапуска ВСЕЙ системы.

Предполагается, что как у клиента, так и у сервера (ВС Тренажера и КГИ) должен быть реализован режим установления соединения, в который , как сервер так и клиент, автоматически переходит во-первых, сразу после запуска, а во-вторых, при разрыве соединения. Находясь в этом режиме сервер ожидает соединения, а клиент периодически, раз в несколько секунд, пытается установить соединение с сервером вплоть до успешного соединения. За счет этого реализуется возможность произвольного порядка запуска и возможность восстановления соединения при его разрыве.

Информация о наличии соединения должна передаваться в приложение (в информационную часть) с тем, чтобы обеспечить полную передачу состояния при восстановлении соединения.

Разработчикам ВС Тренажера предлагается либо реализовать транспортную часть своими силами, или же воспользоваться библиотекой, предоставляемой СофтЛаб-НСК.

## Записи

### Транспортный пакет

Весь пакет является записью следующего формата

DWORD tag=0x20574C54;

DWORD length;

THeader header

. . . data . . .

length - длина записи, по совместительству равен длине всего пакета

THeader – транспортный заголовок (см ниже)

data – данные для передачи, полученные от информационной части, являющиеся, на самом деле, информационным пакетом.

### Транспортный заголовок (THeader)

DWORD tag=0x1;

DWORD length;

double localTime; // Локальное время отправки пакета, в миллисекундах

Локальное время отправки пакета используется КГИ для синхронизации часов между КГИ и ВС Тренажера.

Понятно, что невозможно точно определить понятие «отправки», так как пакет «отправляется» из одной подсистемы в другую несколько раз, пока фактически попадет в сетевую карту, при чем на этот момент он может быть разбит на несколько кадров Ethernet.

В этом смысле предлагается считать временем отправки пакета время его передачи из информационной части в транспортную. При отсутствии искусственной очереди между этими двумя подсистемами можно снимать показания часов прямо при формировании пакета, в противном случае лучше, если таковые будут сняты при вынимании пакета из очереди, для отправки.

Сформированное таким образом время отправки вполне пригодно для коррекции взаимного ухода часов КГИ и ВС Тренажера.

Локальное время задается в миллисекундах, причем за ноль отсчета времени принимается любой произвольный момент, например, момент инициализации транспортной подсистемы моделирующей программы. Такая произвольность допустима, поскольку в алгоритмах синхронизации часов фигурируют разности времен, и любой произвольный константный сдвиг по времени «съедается» алгоритмом.

Транспортный заголовок должен идти первым в транспортном пакете.

# Информационная часть

Информационная часть может расширяться, по мере введения в модель новых типов объектов.

В настоящей версии протокола пакеты с информацией передаются только в одном направлении – от ВС Тренажера к СКГИ, но протокол не исключает возможности расширения таким образом, чтобы информация передавалась и в обратном направлении, если это будет необходимо.

## Модельное время.

Записи, соответствующие передаче моделируюемых параметров, всегда содержат в своем составе модельное время.

Модельное время – это внутреннее время модели, то время, для которого ВС тренажера рассчитала данный параметр. Оно может соответствовать показаниям часов компьютера, на котором производится расчет модели, или быть сдвинуто относительно него на произвольную константную величину.

Большинство параметров, передаваемых по протоколу, содержат модельное время, для которого параметр расчитан, если этот параметр расчетный, или то модельное время, которому соответствует момент снятия с датчика, если этот параметр съемный.

Следует подчеркнуть, что время, для которого параметр расчитан, совсем не обязательно совпадает с моментом расчета параметра, и вообще говоря, не может абсолютно точно совпадать. Расчет параметра занимает протяженное время, в то время как производится для какого-то конкретного момента. Кроме того, например, моделирующая программа может расчитать одновременно несколько значений на несколько тактов вперед, и каждое из этих значений будет иметь разное модельное время, причем не совпадающее с моментом расчета.

Модельное время задается в миллисекундах.

## Объекты

Железнодорожные вагоны, а также другие объекты, для которых по протоколу передается состояние, идентифицируются в протоколе индексами.

Объекты бывают динамическими и статическими.

*Динамические* объекты создаются и уничтожаются в ходе тренировки, к ним относятся вагоны. Соответствие между самим объектом и его индексом устанавливается при создании объекта.

Для такого объекта передается команда на создание, затем произвольное количество команд состояния, затем, возможно, команда на уничтожение.

*Статические* объекты создаются при моделировании сцены (жд станции) и для них не существует команд создания или уничтожения. Соответствие между такими объектами и их индексами устанавливается конфигурационными файлами.

Для разных объектов одного типа (например, для разных вагонов) индексы должны быть разными (уникальными). Для объектов разных типов индексы могут совпадать, иначе говоря, пространства индексов для разных типов объектов являются раздельными.

Для динамических объектов индексы должны быть уникальными в течение всей тренировки. Так, не может быть двух вагонов с одинаковыми индексами, даже если их существование не пересекается в модельном времени.

Иначе говоря, не допускается повторное использование индексов при уничтожении динамического объекта, то есть, например, при уничтожении вагона любым способом (убить) не допускается создание вагона с тем же индексом.

## Частота передачи данных

ВС Тренажера должна передавать пакеты в СКГИ каждые 100 миллисекунд, даже если нет необходимых для передачи данных.

Это необходимо, в частности, для синхронизации часов.

При этом ВС Тренажера должна обеспечить максимальную регулярность выдачи пакетов.

Это обеспечивается наличием отдельного потока (thread), отвечающего за формирование и отправку пакетов.

## Сеанс тренировки, сброс состояния

После инициализации модели, перед выдачей полного состояния (см “Передача состояния”), ВС тренажера должна выдать в СКГИ команду Reset (см. ниже). По этой команде ВС тренажера сбрасывает хранимое состояние о динамических объектах (таблицы, итд) в исходное.

Если восстановление связи не связано с переинициализацией модели (а только с потерей связи), то выдавать команду Reset не нужно.

Иначе говоря, Reset выдается ВС тренажера в том случае, если она сама сбрасывает свои счетчики, таблицы итд, чтобы СКГИ сделала тоже самое.

## Передача состояния

После установления соединения ВС Тренажера должна передать в СКГИ полное текущее состояние всех параметров, необходимое для восстановления изображения на экране.

Для этого транспортная часть должна обеспечивать детектирование наличия соединения.

В этом смысле первичное соединение ничем не отличается от соединения, восстановленного после разрыва соединения.

Так, передаются команды состояния тормозных упоров, стрелочных переводов итп для каждого стрелочного перевода, тормозного упора итп, существующих в модели; затем «создать вагон», «координаты вагона» для всех существующих в модели в данный момент вагонов.

## Записи

### Информационный пакет (RLW)

Весь пакет является записью следующего формата

DWORD tag=0x20574C52;

DWORD length;

//DWORD n; - отсутствует в текущей реализации

Record children[n];

length - длина записи, по совместительству равен длине всего пакета

n – количество вложенных записей

### Сброс (Reset)

DWORD tag=0x80001;

DWORD length;

Команда на сброс состояния.

После получения этой команды СКГИ уничтожает все внутренние структуры данных, соответствующие динамическим объектам.

Выдача этой команды начинает новый сеанс тренировки.

### Создать вагон

DWORD tag=0x2001;

DWORD length;

double modelTime; // модельное время

DWORD index; // уникальный индекс вагона

DWORD type; // тип вагона

DWORD id; // бортовой номер вагона

//голова

// координата первой колёсной оси в первой тележке со стороны головы вагона

DWORD index;  // индекс рельса

double t; // расстояние от первой точки рельса

//координата последней колёсной оси в первой тележке со стороны головы вагона на граферельсов

DWORD index; //индексрельса

double t; //расстояние от первой точкирельса

//хвост

//координата первой колёсной оси впоследней тележке со стороны головы вагона на граферельсов

DWORD index; //индексрельса

double t; //расстояние от первой точкирельса

// координата последней колёсной оси в последней тележке со стороны головы, то есть вообще последней оси, которая находится в самом хвосте

DWORD index; // индекс рельса

double t; // расстояние от первой точки рельса

Порождённый вагон считается «замороженным» (см. ниже).

### Уничтожить вагон

DWORD tag=0x2004;

DWORD length;

double modelTime; // модельное время

DWORD index; // индекс вагона

Указанный вагон уничтожается, все связанные с ним структуры данных освобождаются.

### Координаты вагона

DWORDtag=0x2005;

DWORD length;

double modelTime;//модельноевремя

DWORD index;//индексвагона

//голова

// координата первой колёсной оси в первой тележке со стороны головы вагона

DWORD index;  // индекс рельса

double t; // расстояние от первой точки рельса

//координата последней колёсной оси в первой тележке со стороны головы вагона на граферельсов

DWORD index; //индексрельса

double t; //расстояние от первой точкирельса

//хвост

//координата первой колёсной оси впоследней тележке со стороны головы вагона на графе рельсов

DWORD index; //индексрельса

double t; //расстояние от первой точкирельса

// координата последней колёсной оси в последней тележке со стороны головы, то есть вообще последней оси, которая находится в самом хвосте

DWORD index; // индекс рельса

double t; // расстояние от первой точки рельса

double velocity;  // скорость вагона м/с, относится к самой последней координате, координате самой последней колёсной оси.

скорость используется для имитации гладкого движения вагона в условиях несовпадения частоты команд «координаты вагона» и кадровой частоты СКГИ

### 3D Координаты вагона

DWORDtag=0x2007;

DWORD length;

double modelTime;//модельное время

DWORD index;//индекс вагона

double xh,yh,zh, // 3d координаты головы

xt,yt,zt, // 3d координаты хвоста

bank, // крен вагона

courseH,courseT,// углы поворота тележек

velocity;  // скорость вагона м/с

### Заморозить вагон

DWORD tag=0x2006;

DWORD length;

double modelTime; // модельное время

DWORD index; // индекс вагона

Указание, что вагон не движется. Используется для оптимизации, например движение указанного вагона не интерполируется СКГИ.

Вагон остается замороженным до получения команды «Координаты вагона».

### Состояние стрелочного перевода

DWORD tag=0x3001;

DWORD length;

double modelTime; // модельное время

DWORD index; // индекс стрелки

int direction; // 0 прямо 1 отклонение (в сторону)

// (-1 промежуточные, нешт. ситуация)

### Состояние стрелочного перевода

DWORD tag=0x3002;

DWORD length;

double modelTime; // модельное время

DWORD index; // индекс стрелки

int direction; // 0 влево 1 вправо

// (-1 промежуточные, нешт. ситуация)

### Состояние тормозного упора:

DWORD tag=0x4001;

DWORD length;

double modelTime; // модельное время

DWORD index; //   индекс

double state; // 0.0 - крайнее нижнее 1.0 крайнее

// верхнее  ] 0...1[ - промежуточные

### Состояние замедлителя

DWORD tag=0x5001;

DWORD length;

double modelTime; // модельное время

DWORD index; // индекс

double state;  // -1.0 - крайнее опущенное

// 0.0 готовое к торможению

// 1.0 крайнее верхнее - торможение

// [-1.0 ..0.0] [0.0 .. 1.0] - промежуточные

### Состояние светофора

DWORD tag=0x6001;

DWORD length;

double modelTime; // модельное время

DWORD index; // индекс

DWORD state;   // битовое поле.

// 1-красный

// 2-зелёный

// 4-синий

// 8-лунно-белый

// 16-жёлтый верхний

// 32-жёлтый нижний

//

// 256 – красный мигающий

// 512 – зелёный мигающий

// 1024 - синий мигающий

// 2048 - лунно-белый мигающий

// 4096 - жёлтый верхний мигающий

// 8192 - жёлтый нижний мигающий

Биты, задающие мигания, интерпретируются только при наличии основных.

### Далее, в командах задания параметров наблюдателя, будет использованы следующие структуры данных:

### Положение в пространстве

struct Vector6

{

double x; // x-координата, метр

double y; // у-координата, метр

double z; // z-координата, метр

double course; // курс, радиан

double pitch; // тангаж, радиан

double bank; // крен, радиан

}

Вспомогательная структура данных для построения записей пакета, задающих положение объекта (наблюдателя) в 3D-пространстве.

Задаёт положение и ориентацию объекта (наблюдателя) в пространстве.

Может использоваться для задания скорости, в этом случае линейная скорость задается в метрах в секунду, а угловая – в радианах в секунду.

### Имя наблюдателя

BYTE name[8];

Используется для указания имени наблюдателя в записях пакета, задающих параметры наблюдателя.

Имя наблюдателя соответствует записи group\_name в файле train.ini.

Неиспользуемые символы заполнены нулями.

### Положение наблюдателя

DWORD tag=0x7001;

DWORD length;

double modelTime; // модельное время

BYTE name[8]; // имя наблюдателя.

Vector6 pos; // положение

Ставит наблюдателя в новое положение.

### Положение и скорость наблюдателя

DWORD tag=0x7002;

DWORD length;

double modelTime; // модельное время

BYTE name[8]; // имя наблюдателя.

Vector6 pos; // положение

Vector6 vel; // скорость

Ставит наблюдателя в новое положение, при этом производится сглаживание его движения. Скорость используется для имитации гладкого движения наблюдателя в условиях несовпадения частоты команд «координаты наблюдателя» и кадровой частоты СКГИ

### Присоединить наблюдателя к локомотиву (вагону)

DWORD tag=0x7004;

DWORD length;

double modelTime; // модельное время

BYTE name[8]; // имя наблюдателя.

DWORD index; // индекс вагона

Vector6 pos; // положение относительно системы

// координат вагона

После выдачи этой команды соответствующий наблюдатель будет перемещаться вместе с указанным вагоном.

### Массив предопределённых позиций наблюдателя

DWORD tag=0x7005;

DWORD length;

double modelTime; // модельное время

DWORD number; // количество позиций

Vector6 pos[number];

Задает набор положений наблюдателя, из которого в дальнейшем можно выбирать то или иное положение командой “Выбор положения наблюдателя”.

Сама по себе команда **не** приводит к изменению фактического положения наблюдателя, даже если ранее была выдана команда “Выбор положения наблюдателя”.

### Выбор положения наблюдателя из массива позиций

DWORD tag=0x7006;

DWORD length;

double modelTime; // модельное время

BYTE name[8]; // имя наблюдателя.

DWORD number; // номер позиции

Ставит наблюдателя в новое положение.

### Горизонтальный угол обзора наблюдателя

DWORD tag=0x7007;

DWORD length;

double modelTime; // модельное время

double fovX; // горизонтальный угол обзора в градусах

### *Звуки – на согласовании*

### Издавать скрежет тормозных замедлителей о колёса вагона

DWORD tag=0x8001;

DWORD length;

double modelTime; // модельное время

DWORD index; // индекс тормозного замедлителя

### Остановить скрежет тормозных замедлителей о колёса вагона

DWORD tag=0x8002;

DWORD length;

double modelTime; // модельное время

DWORD index; // индекс тормозного замедлителя

### Издать звук удара вагона о вагон

DWORD tag=0x8003;

DWORD length;

double modelTime; // модельное время

double x,y,z; // точка удара в мировой системе координат

### Задать время суток

DWORD tag=0x9001;

DWORD length;

double modelTime;    // модельное время

double timeOfDay;    // 0-24

### Задать небо

DWORD tag=0x9002;

DWORD length;

double modelTime;    // модельное время

double skyState;    // 0 - пасмурно, 1 - ясно

### Задать наличие осадков

DWORD tag=0x9003;

DWORD length;

double modelTime;         // модельное время

DWORD isPrecipitationOn;  // 0 - нет, 1 - есть

### Задать наличие тумана

DWORD tag=0x9004;

DWORD length;

double modelTime;    // модельное время

DWORD isFogOn;       // 0 - нет, 1 - есть

### Задать время года

DWORD tag=0x9005;

DWORD length;

double modelTime;    // модельное время

DWORD season;        // 0 - лето, 1 - зима

### Задать мокрое стекло

DWORD tag=0x9006;

DWORD length;

double modelTime;    // модельное время

DWORD state;        // 0 - нет, 1 - есть

### Показать/скрыть пост

DWORD tag=0x9007;

DWORD length;

double modelTime;    // модельное время

DWORD flag;

Переключение объектов сцены.

flag - флаги объектов по битам:

0 - основной пост

1 - дополнительный пост