Протокол, версия 2.

Обмен информацией RC происходит с двумя типами устройств - камерами и SM02.

Обмен базовой информацией осуществляется по UDP протоколу с использованием порта 21075.

Обмен основной информацией осуществляется по TCP протоколу (порт 21074), где клиентом являются RC и камеры, а сервером - SM02.

Протокол UDP используется для оповещения устройств в сети о наличии работающего SM02 и сборе информации о клиентах.

Sm02 рассылает Broadcast сообщение, состоящее только из слова “PING”. RC при получении этой команды по адресу пакета открывает TCP соединение и отправляет команду AUTH (TCP).

Команда PING(UDP) рассылается раз в 2 секунды.

Протокол TCP.

После получения от RC команды AUTH в случае удовлетворения всем условиям (верный код и отсутствие уже работающего RC в сети) SM02 раз в 2 секунды отсылает PING и ждет в ответ HELLO. Если в течение 6 секунд не получает ответ, то считает, что устройство не в сети (верно и со стороны RC).

Команды TCP протокола имеют структуру:

[Заголовок][тело команды]

Заголовок имеет длину 4 байт. Первые 2 байта - длина сообщения без учета самих двух байт длины (напр. для пустого сообщения - 2 - идентификатор команды и версия протокола), третий - идентификатор команды, последний - версия протокола (см. первую строчку). Кодировка - UTF-8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Отправитель | Имя | Идентификатор | Поля тела команды | Описание |
| SM02 | PING | 0x01 | - | Команда проверки наличия устройства в сети. В ответ RC должен прислать команду HELLO (0x0E) |
| RC | SETNAME | 0x02 | [person] - 1 байт  [name length] - 1 байт  [name] - строка в UTF-8 | Установка имени бойца, где поле person - идентификатор роли:  int *PERSON\_TYPE\_LEFT* = 1;  int *PERSON\_TYPE\_RIGHT* = 2;  int *PERSON\_TYPE\_REFEREE* = 3; |
| RC | SETSCORE | 0x03 | [person] - 1 байт  [score] - 1 байт | Установка счет бойца, поле person - идентификатор роли:  int *PERSON\_TYPE\_LEFT* = 1;  int *PERSON\_TYPE\_RIGHT* = 2; |
| RC | SETCARD | 0x04 | [person] - 1 байт  [card] - 1 байт | Установка карточки, поле card:  byte *CARD\_STATUS\_NONE* = (byte) 0x01;  byte *CARD\_STATUS\_YELLOW* = (byte) 0x02;  byte *CARD\_STATUS\_RED* = (byte) 0x03;  byte *CARD\_STATUS\_BLACK* = (byte) 0x04;  byte *CARD\_P\_STATUS\_NONE* = (byte) 0x05;  byte *CARD\_P\_STATUS\_YELLOW* = (byte) 0x06;  byte *CARD\_P\_STATUS\_RED* = (byte) 0x07;  byte *CARD\_P\_STATUS\_BLACK* = (byte) 0x08; |
| RC | SETPRIORITY | 0x05 | [person] - 1 байт | Установка приоритета, , где поле person - идентификатор роли:  int *PERSON\_TYPE\_LEFT* = 1;  int *PERSON\_TYPE\_RIGHT* = 2;  int *PERSON\_TYPE\_NONE* = 0; |
| RC | SETPERIOD | 0x06 | [period] - 1 байт | Установка периода, поле period - номер (int) |
| RC | SETWEAPON | 0x07 | [weapon] - 1 байт | Установка оружия, поле weapon -  int WEAPON\_FOIL = 1;  int WEAPON\_EPEE = 2;  int WEAPON\_SABRE = 3; |
| RC | SETTIMER | 0x08 | [time] - 4 байта  [mode] - 1 байт | Установка таймера, поле timer - время в миллисекундах,  поле mode -  int *TIMER\_MODE\_MAIN* = 0;  int *TIMER\_MODE\_PAUSE* = 1;  int *TIMER\_MODE\_MEDICINE* = 2; |
| RC | STARTTIMER | 0x09 | [state] - 1 байт | Старт/стоп таймера, поле state - 0 - остановлен, 1 - запущен |
| RC | SWAP | 0x0A | - | Поменять местами бойцов |
| SM02 | BROADCAST | 0x0B | [weapon] - 1 байт  [flag left] - 1 байт  [flag right] - 1 байт  [timer] - 4 байта  [timer state] - 1 байт | Оповещение о состоянии SM01:  weapon - установленное оружие int WEAPON\_FOIL = 1;  int WEAPON\_EPEE = 2;  int WEAPON\_SABRE = 3;  flag left/flag right - состояние флага слева/справа - 0-нет, 1-белый, 2-цвет  timer - время в миллисекундах (integer)  timer state - состояние таймера, 0-остановлен, 1-запущен |
| RC | VISIBILITY | 0x0C | [is video counters]- 1 байт  [is photo]- 1 байт  [is passive]- 1 байт  [is country]- 1 байт | Отображение соответствующих полей на экране SM02, 0-элемент невидим, 1 - видим |
| RC | VIDEO COUNTERS | 0x0D | [left]- 1 байт  [right]- 1 байт | Установка соответствующих счетчиков видеоповторов, integer (0-2) |
| RC | HELLO | 0x0E | - | Ответ на команду TCP PING. |
| RC | DISCONNECT | 0x0F | - | Сброс данных SM02 об RC - он снова отображает Qr код и ждет подключения нового RC |
| RC | PASSIVE | 0x10 | [is shown] - 1 байт  [locked] - 1 байт  [default] - 4 байта | Установка состояния таймера пассива - is shown - видим/невидим, locked - заблокирован/нет, флаги имеют значение 0 или 1  default - значение по умолчанию в миллисекундах, integer |
| SM02 | PASSIVE MAX | 0x11 | - | Оповещение о достижении максимума таймера пассива |
| SM02 | PAUSE FIN | 0x12 | - | Оповещение о завершении паузы/медицинской паузы |
| RC | SET DEF TIME | 0x13 | [time] - 4 байта | Установка начального времени периода в миллисекундах Integer |
| RC | COMPETITION | 0x14 | [name length] - 1 байт  [name] - строка в UTF-8 | Установка названия соревнований |
| RC | VIDEO ROUTES | 0x15 | [cam number] - 1 байт  [cam 1 name length]- 1 байт  [cam 1 name] - строка в UTF-8  [cam 1 target length]- 1 байт  [cam 1 target name] - строка в UTF-8  …..  cam N…... | Установка соответствия между камерами и адресатами, куда камерам отправлять видео. В списке может быть меньше камер, чем есть в системе, тогда остальные не выгружают видео никуда. В поле cam N target length может быть указан 0, тогда в поле cam N target name ожидается тип устройств (SM02 - 1, повторители - 4) |
| RC | LOAD FILE | 0x16 | [name length] - 1 байт  [name] - строка в UTF-8 | Команда выгрузки файлов с камер адресатам в соответствии с предыдущей командой, указывается имя необходимого файла |
| RC | PLAYER | 0x17 | [speed]- 1 байт  [mode]- 1 байт  [timestamp]- 4 байта | Управление воспроизведением видео - speed скорость воспроизведения от 0 до 10,  mode - 0-stop, 1- play, 2-pause,  timestamp - перемотка на нужную отметку времени в миллисекундах (перемотка игнорируется при mode - play) |
| RC | RECORD | 0x18 | [mode]- 1 байт | Управление записью видео - mode: 0-запись выключена, 1 - пуск записи, 2 - пауза - останавливает текущую запись, обрезает видео и начинает новую запись, по окончании обрезки и по готовности видео SM02 оповещает командой VIDEO READY(0x1B) |
| RC | DEV REQUEST | 0x19 | - | Запрос списка устройств в системе |
| SM02 | DEVICE LIST | 0x1A | [dev count]- 1 байт  [dev 1 type]- 1 байт  [dev 1 name length]- 1 байт  [dev 1 name] - строка в UTF-8  ….  dev N... | Список устройств в системе -  количество устройств, для каждого - тип (SM02 - 1, RC - 2, камера - 3, повторитель - 4, резервный судья - 6), длина имени и имя |
| SM02 | VIDEO READY | 0x1B | [name length] - 1 байт  [name] - строка в UTF-8 | Уведомление о готовности видеоролика к отправке на просмотр, содержит длину имени файла и имя файла. |
| SM02 | VIDEO RCVD | 0x1C | - | Уведомление о готовности проигрывать ролик |
| RC | RESET | 0x1D | - | Сброс данных боя на SM02 |
| RC | ETH NEXT PREV | 0x1E | [mode]- 1 байт | Запрос следующего/предыдущего боя (mode: 0/1) |
| RC | ETH APPLY | 0x1F | - | Команда принятия боя, пришедшего от сервера соревнований |
| RC | ETH FINISH ASK | 0x20 | - | Запрос завершения боя у сервера соревнований |
| SM02 | ETH DISP | 0x21 | [score left] - 1 байт  [score right] - 1 байт  [period] - 1 байт  [card left] - 1 байт  [card right] - 1 байт  [left name length] - 1 байт  [left name] - строка в UTF-8  [right name length] - 1 байт  [right name]  - строка в UTF-8  [time] - 4 байта | Оповещение о получении данных боя от сервера соревнований с данными боя, где score - счет слева/справа, period - номер периода, card left/right - byte *CARD\_STATUS\_NONE* = (byte) 0x01;  byte *CARD\_STATUS\_YELLOW* = (byte) 0x02;  byte *CARD\_STATUS\_RED* = (byte) 0x03;  и имена бойцов |
| SM02 | ACK NAK | 0x22 | [mode] - 1 байт | Уведомление о решении сервера соревнований завершить или не завершить бой (0 - NAK, 1 - ACK) |
| RC | AUTH | 0x23 | [device type] - 1 байт  [code length] - 1 байт  [code] - строка в UTF-8  [name length] - 1 байт  [name] - строка в UTF-8 | Отправляется после установки соединения по TCP.  Поле device type для RC - 2  code - строка кода, полученная после сканирования QR кода SM02 или сохраненная с предыдущего запуска.  name - имя устройства  После отправки ожидается команда AUTH ANSW от SM02 |
| SM02 | AUTH ANSW | 0x24 | [answer] - 1 байт | Команда отправляется после получения команды AUTH от RC.  Поле answer -  1 - OK  2 - неправильный код  3 - в системе уже есть подключенный RC.  RC при получении ответа OK продолжает работу с SM02.  При получении другого ответа уведомляет пользователя о типе ошибки и закрывает соединение по TCP. |