1. Общая информация о протоколах обмена системы с сервером.

Система «общается» с сервером используя протокол HTTP версии 1.1. Это позволяет строить серверное ПО на базе любого распространенного хостинга, а не только на базе специализированных выделенных серверов.

Запросы могут быть GET или POST.

Пример запроса:

```
GET /addlog?imei=123456789012345&text=Питание%20системы%20включено. HTTP/1.1<cr><lf>
Host: gps-maps.appspot.com<cr><lf>
Connection: keep-alive<cr><lf>
<cr><lf>
```

где:

значение после '**imei=**' содержит идентификатор системы IMEI; любой запрос системы будет содержать этот параметр. Если система с таким идентификатором не зарегистрирована на сервере, то должна быть автоматически добавлена в базу. Если в запросе присутствует параметр '**phone=xxxxx**' то для данной системы должен быть установлен указанный телефонный номер.

значение после '**Host:**' содержит адрес сервера;

перед последним <cr><lf> возможны и другие строки из протокола HTTP. Очередность может быть произвольной. Очередность параметров в запросе (разделенных символом '&' также может быть произвольной).

Для краткости, далее по тектсу, где это возможно, запрос будем записывать опуская поля HTTP/1.1, Host, Connection и т.п:

```
GET /addlog?imei=123456789012345&text=Питание%20системы%20включено.
```

Ответ сервера имеет следующий вид:

```
HTTP/1.1 200 OK<cr><lf>
Content-Type: text/plain<cr><lf>
<cr><lf>
ADDLOG:<space>OK<cr><lf>
```

Ответ сервера анализируется следующим образом: игнорируются все строки до двойного <cr><lf>, а строки после воспринимаются как данные для системы. Обычно это одна или несколько текстовых строк.

Внимание! Не поддерживается chunk-кодирование ответа сервера. Если нет возможности отключать HTTP Chunking, то можно выставлять Content-Type: application/octet-stream, т.е. текстовый ответ передавать в бинарном виде.

```
HTTP/1.1 200 OK<cr><lf>
Content-Type: application/octet-stream<cr><lf>
Content-Length: 12<cr><lf>
<cr><lf>
ADDLOG:<space>OK<cr><lf>
```

Для простоты, далее по тексту документа, ответ сервера будем записывать без заголовка:

```
ADDLOG: OK
```

2. Отправка информационных сообщений.

Данные информационные сообщения являются отладочными, и для применения конечными пользователями не предполагаются.

Отправка информационного сообщения производится методом GET, пример:

```
GET /addlog?imei=123456789012345&text=Питание%20системы%20включено.
```

где строка после '**text=**' и до пробела содержит передаваемое сообщение.

На данный запрос сервер должен ответить:

```
ADDLOG: OK
```

3. Отправка бинарных GPS-данных.

Система отправляет GPS-данные на сервер используя запросы в формате HTTP.

Отправка данных производится методом POST, пример пакета:

```
POST /bingps?imei=123456789012345 HTTP/1.1
Content-type: application/octet-stream
Content-Length: 29

HILDDDDDDDDHILDDDDD...HILDDDDDCC
```

Где: **HILDDDDDDDDHILDDDDDHILDDDDDCC** — передаваемые данные (заголовок-идентификатор-длина-данные-заголовок-идентификатор-длина-данные-...-заголовок-идентификатор-длина-данные-контрольная_сумма).

Сервер проверяет контрольную сумму и если проверка прошла успешно - отвечает:

```
BINGPS: OK
```

Если контрольная сумма совпала (получен ответ BINGPS:<space>OK), то система удаляет отправленные данные из памяти. Достоверность данных проверяется только(!) контрольной суммой, а не анализом каждого пакета в запросе.

В одном запросе может быть передано несколько накопившихся к текущему моменту пакетов. Максимальных размет одного запроса 16384 байт. Если к моменту отправки в памяти накопилось больше данных, то они будут переданы в двух или более запросах. Модуль делит посылки таким образом чтобы пакеты между запросами не были разделены. Желательно чтобы сервер мог воспринимать пакеты и большего размера, в идеале — произвольного.

Последние 2 байта (CC) запроса содержат циклическую контрольную сумму CRC16-CCITT: Poly=0x1021, Init=0, Revert=false, XorOut=0x0000, Check: CRC("123456789")=0x31C3.

Общий формат единичного пакета данных:

Номер байта	Имя	Назначение
0	HEADER	Заголовок = 0xFF
1	ID	Идентификатор пакета
2	LENGTH	Длина пакета в байтах, включая HEADER, ID и LENGTH
3..(LENGTH-1)	D3..Dn	Данные пакета

На данный момент поддерживаются следующие пакеты:

ID = 0xF1 - Минимальные GPS-данные: дата-время, долгота, широта, скорость, направление, кол-во спутников

Позиция	Назначение	Описание	Примечание
D3	день	День месяца = 1..31	
D4	месяц | ((год-2010) << 4)	Месяц = 1..12 год = 0..14 → 2010..2024	годы до 2010 и после 2024 не поддерживаются (метка времени с неправильным годом не будет создана)
D5	Часы	Часы = 0..23	
D6	Минуты	Минуты = 0..59	
D7	Секунды	Секунды = 0..59	
D8	Широта (LL)	Градусы широты = 0..89	Latitude - "LLll.mmnn"

Позиция	Назначение	Описание	Примечание
D9	Широта (ll)	Минуты целая часть = 0..59	
D10	Широта (mm)	Минуты дробная часть1 = 0..99	Первые 2 цифры дробной части
D11	Широта (nn)	Минуты дробная часть2 = 0..99	Вторые 2 цифры дробной части
D12	Долгота (LLL)	Градусы долготы = 0..179	Longitude — "LLLll.mmnn"
D13	Долгота (ll)	Минуты целая часть = 0..59	
D14	Долгота (mm)	Минуты дробная часть1 = 0..99	Первые 2 цифры дробной части
D15	Долгота (nn)	Минуты дробная часть2 = 0..99	Вторые 2 цифры дробной части
D16	D16.0 = NS D16.1 = EW D16.2 = (Course & 1)	D16.0=0 для N D16.0=1 для S D16.1=0 для E D16.1=1 для W D16.2=0 для четных Course D16.2=1 для нечетных Course	Полушарие и младший разряд направления
D17	Спутники	Кол-во спутников 3..12	Если спутников менее 3-х, то данные не валидны.
D18	Скорость	Скорость в узлах 0..239	скорости более 239 узлов не поддерживаются (239 узлов = 442 км/ч)
D19	Скорость дробная часть	Дробная часть скорости 0..99	
D20	Направление	Направление/2 = 0..179	Направление передается деленное на 2, младший разряд направления передается совместно NS/EW, см. D16
D21	Направление дробная часть	Дробная часть направления 0..99	

ID = 0xF2 - Минимальные GPS-данные + информация датчиков: дата-время, долгота, широта, скорость, направление, кол-во спутников, напряжение внешнего питания, напряжение внутреннего питания, состояние входов и т.д.

Позиция	Назначение	Описание	Примечание
D0.. D21	Аналогично ID=1		
D22	Напряжение внешнего питания	Напряжение/10 = 0..200	0 — соответствует 0В 10 — соответствует 1В 120 — соответствует 12В Допустимые значение 0..20.0 В
D23	Напряжение внутреннего аккумулятора	Напряжение/50 = 0..239	0 – соответствует 0В 50 — соответствует 1В 210 — соответствует 4.2В Допустимые значения 0..4.62В
D24	Вход 1. Напряжение	Напряжение/10 = 0..239	
D25	Вход 2. Напряжение	Напряжение/10 = 0..239	
D26	Логические входы	Биты 0..6 содержат состояние входов 1..7	До 7-ми входов.
D27	Зарезервировано	=0	

D28	Зарезервировано	=0	
D29	Зарезервировано	=0	
D30	Зарезервировано	=0	
D31	Зарезервировано	=0	

ID = 0xF3 – События.

Позиция	Назначение	Описание	Примечание
D3	день	День месяца = 1..31	
D4	месяц | ((год-2010) << 4)	Месяц = 1..12 год = 0..14 → 2010..2024	годы до 2010 и после 2024 не поддерживаются (метка времени с неправильным годом не будет создана)
D5	Часы	Часы = 0..23	
D6	Минуты	Минуты = 0..59	
D7	Секунды	Секунды = 0..59	
D8	Идентификатор события	MID	См. таблицу ниже
D9	Параметр1 события	MP0	
D10	Параметр2 события	MP1	

Примечание: В моменты когда сигнал GPS отсутствует или модуль GPS отключен, точность даты-времени может быть низкой (± 2%).

MID	Параметры	Описание
0x01		Изменение состояния питания
	MP0 = 1	Отключение внешнего питания
	MP0 = 2	Возобновление внешнего питания
0x02	MP0 = 1	Пропадание сигнала GPS
	MP1 = 2	Появление сигнала GPS
	MP1 = 3	Пропадание GSM/GPRS
	MP1 = 4	Возобновление GSM/GPRS
0x03		Нажатие тревожной кнопки SOS.
0x11		Нарушение целостности системы
	MP0 = 1	Вскрытие корпуса — контактый выключатель
	MP0 = 2	Вскрытие корпуса — датчик освещенности
TBD	TBD	TBD
0x??		
0x??		Слив топлива.
0x??		Термо-контроль. Выход за установленную зону.
	MP0 = 1	
	MP1	Значение температуры

Примеры бинарных пакетов можно загрузить тут: http://gps-maps.appspot.com/binbackup

4. Сохранение конфигурации.

При включении система передает на сервер список параметров конфигурации и собственный номер SIM-карты.

```
POST /config?imei=123456789012345&cmd=save&phone=%2B380679332332 HTTP/1.1
Content-type: text/plain
Content-Length: <N>

gps.T1.1    INT     600     600
gps.T1.0    INT     30      30
gps.B1.3    INT     512     512
gps.B1.2    INT     512     512
…
…
…
gps.T0.2    INT     10      10
gps.T0.3    INT     10      10
akkum.U.1   INT     907     907
akkum.U.0   INT     862     862
akkum.U.3   INT     984     984
akkum.U.2   INT     911     911
akkum.U.4   INT     911     911
gps.V0.2    INT     10      10
gps.V0.3    INT     20      20
gps.V0.0    INT     5       5
gps.V0.1    INT     20      20
gps.T4.0    INT     720     720
gps.T4.1    INT     240     240
gps.T4.2    INT     720     720
END
```

где:

каждая строка содержит 4 поля, разделенных пробелами или знаками табуляции:

первое поле — имя параметра

второе поле — тип значения параметра (в данной реализации только INT=-32768...32767)

третье поле — текущее значение параметра

четвертое поле — заводское значение параметра

Ответ сервера:

```
CONFIG: OK
```

5. Обновление конфигурации.

Если на какой-либо запрос системы сервер, перед обычным ответом, отвечает строкой:

```
CONFIGUP
```

то система запрашивает обновление параметров конфигурации:

```
GET /params?imei=123456789012345&cmd=params
```

Формат ответа сервера:

```
PARAM<space>gps.V0.1<space>15
PARAM<space>akkum.U.1<space>560
FINISH
```

где:

строки вида 'PARAM name value' содержат список параметров (name), значения (value) которых, необходимо изменить.

Завершает список изменяемых параметров строка FINISH.

Если не требуется менять параметры (система может запрашивать изменения при получении определенных SMS-команд или при входящем звонке), то сервер отвечает:

```
NODATA
```

После обработки система подтверждает обновление конфигурации запросом:

```
GET /params?imei=123456789012345&cmd=confirm
```

Ответ сервера:

```
CONFIRM: OK
```

6. Обновление внутреннего программного обеспечения.

Поддерживается дистанционное обновление программного обеспечения.

Если на какой-либо запрос системы сервер, перед обычным ответом, отвечает строкой:

```
FWUPDATE
```

или если на систему отправить SMS-команду: **fwupdate**, то система начинает процедуру обновления внутреннего программного обеспечения.

Система запрашивает версию последней доступной прошивки:

```
GET /firmware?imei=123456789012345&cmd=check&hwid=3031
```

Ответ сервера:

```
SWID:<space>3060
```

Где значение после SWID: - это версия (в 16-тиричном виде) последей доступной прошивки для запрошенного hwid.

Если прошивки для запрошенного hwid недоступны, то сервер отвечает:

```
NOT FOUND
```

Система, если видит необходимость обновления (версия SWID больше чем та что в нем установлена), то система запрашивает прошивку:

```
GET /firmware?imei=123456789012345&cmd=get&hwid=3031
```

Сервер отправляет образ в следующем виде:

```
SWID:3060
LENGTH:xxxx
LINE0000:000102030405060708090A0B0C0D0E0F10111213141516171819 1A1B1C1D1E1F
LINE0001:000102030405060708090A0B0C0D0E0F10111213141516171819 1A1B1C1D1E1F
LINE0002:000102030405060708090A0B0C0D0E0F10111213141516171819 1A1B1C1D1E1F
…
LINExxxx:000102030405060708090A0B0C0D0E0F10111213141516
CRC:xxxx
ENDDATA
```

где:

SWID:xxxx - версия в 16-тиричном виде

LENGTH: - размер образа в 16-тиричном виде

строки LINExxxx:, где xxxx – порядковый номер строки в 16-тиричном виде. В каждой строке LINE передается 32 байта образа представленных в 16-тиричной системе. (последняя строка может содержать менее 32-х байт образа).

CRC:xxxx — значение циклической контрольной суммы CRC16-CCITT (в 16-тиричном виде)

Завершается ответ строкой ENDDATA