

**Терминал для передачи данных
в беспроводных сетях общего назначения**

Оглавление

Оглавление	2
1. Технические характеристики	3
2. Начало работы	4
2.1. Настройка.....	4
2.1.1. Файл IO*.txt.....	5
2.2. Обновление прошивки.....	5
2.2.1. Обновление через файл	5
2.2.2. Обновление через конфигуратор (виртуальный COM-порт).....	6
2.2.3. Аварийный режим обновления.....	6
3. Описание общих регистров.....	8
3.1. Форматы данных	8
3.2. Командный регистр.....	9
4. Модем	11
4.1. Файл SimCard*.txt	11
4.2. Команды	12
4.3. Таблица регистров	16
5. Телеметрия MQTT.....	17
5.1. Параметры связи терминала с Брокером	17
5.2. Файл MQTT*.txt.....	18
5.3. Команды	19
5.4. Таблица регистров	22
6. Контрольно-кассовая техника ККТ	23
6.1. Поток данных при работе с кассой.....	23
6.2. Файл Checks*.txt.....	23
6.3. Файл ККТ*.txt.....	25
6.4. Команды	26
6.5. Таблица регистров	33
7. Информатор	34
7.1. Файл SimCard*.txt.....	34
7.2. Команды	36
7.3. Таблица регистров	41
Примеры.....	42
1. Телеметрия.....	42
1.1. Создание и настройка аккаунта на сервере брокера.....	42
1.2. Настройка MQTT-клиента на сотовом телефоне	44
Приложение А. Таблицы кодирования текстовой информации.....	47
Основная таблица ASCII-символов	47
Расширенная таблица ASCII-символов. Кодировка Windows-1251	48
Приложение В. Формат JSON	49
Приложение С. Протокол Modbus.....	51
Режим RTU.....	51
Расчет контрольной суммы (CRC-16)	52
Передача пакета	53
Прием пакета	53
Приложение D. Распиновка кабеля.....	54
Кабель от Терминала к ККТ	54

1. Технические характеристики

- Питание: постоянное напряжение 24В/0,5А +10%/-20%,
- RS-485: Защита от перенапряжения (от -5В до +10В) и короткого замыкания,
- RS-232: Защита от перенапряжения (до +/-20В) и короткого замыкания,
- Модем: GSM 850/900 МГц, 1800/1900 МГц, GPRS class 12/10, встроенный WatchDog - перезапуск в случае отсутствия связи, автоматическое подключение к сети Интернет при необходимости.
- Антенна: Навесная внутренняя или внешняя. Изолированная от земли.
- Тип SIM-карты: Micro SIM.
- USB: High-speed 2.0. Тип разъема: Mini-USB.
- Температура хранения от -20°C до 50°C.
- Температура эксплуатации от -10°C до 45°C и влажности не более 90% при 20°C без образования конденсата.

 *Важно! Включение Терминала без подключенной антенны или заземление антенны может привести к поломке устройства. Такая поломка не может считаться гарантийным случаем.*

 *Устройство боится электростатических разрядов. Нельзя класть устройство без корпуса на полиэтиленовые пакеты или ткань.*

 *Эксплуатация или хранение устройства в условиях отличных от описанных выше ведет к потере гарантии.*

 *Гарантийные обязательства распространяются только на устройства с заводским браком, при отсутствии следов неквалифицированного ремонта.*

Каналы связи:

- GSM/GPRS
- RS-232
- RS-485
- USB (MassStorage, VirtualCOM-Port)

Протоколы передачи информации:

- Modbus RTU Slave: скорость передачи данных от 1200 до 115200 бод/с, 8 бит, четность: нет/чет/нечет., 1 или 2 стоповых бита. Максимальное количество регистров принимаемое/передаваемое за одну команду: 50. Адрес устройства от 1 до 247. По умолчанию используются параметры: 9600, 8/n/1, адрес 10.
- Протокол передачи информации контрольно-кассовой техники: Штрих-М ККТ 2.0 (PayOnline-01ФА).
- Поддержка алгоритмов AES, режимы: CBC, CFB, CTR, EAX.
- Алгоритм цифровых подписей: OMAC.
- MQTT - стандартный протокол для Интернета Вещей (IoT). Поддержка качества доставки сообщений QOS: 0/1/2.
- SMS-информирование, отправка текстовых сообщений на русском и/или английском языках длиной до 512 символов. Промежуточный буфер с возможностью работы с заранее заданным набором текстовых строк или посимвольной передачей от управляющего устройства Modbus. Строки поддерживают запись с аргументами для вставки числовых переменных.
- Функция дозвона на заданный номер телефона.
- Встроенная телефонная книга.

2. Начало работы

Перед началом работы с устройством необходимо записать на его виртуальный диск файлы с настройками для функций, которые планируется использовать. После чего устройство следует перезагрузить. Файлы с настройками, примеры использования и другая техническая информация всегда доступны на веб-сайте компании по адресу: cyberidea.io

2.1. Настройка

Для конфигурирования устройства следует использовать кабель USB, операционная система определит новые устройства:

- CI Virtual COM Port - виртуальный COM-порт, для полноценной работы нужен фирменный драйвер «VCP Driver»,
- Виртуальный диск с названием «Settings».

Все настройки хранятся на виртуальном диске в виде текстовых файлов, и разделены на функциональные блоки. Если определенный функционал не требуется, то файл с настройками можно не копировать на устройство. Имена файлов состоят из обязательной части, части имени, которую может изменить пользователь и расширения файла. Далее будут приводиться только обязательные части названия, а пользовательская часть будет обозначена символом “*”. Например “IO*.txt” - означает, что обязательная часть это “IO”, символ “*” может быть заменен на символы пользователя, либо опущен, и расширение файла “txt”. Файл с названием “IO_TL_MbSlave.txt” имеет пользовательскую часть “TL_MbSlave” - это помогает лучше понимать назначение и содержимое этого файла. Общая длина названия файла ограничена 50 символами, может содержать символы русского и английского алфавитов, целые числа, символ пробела “ ”, символ дефиса “-” и символ нижнего подчеркивания “_”.

Файлы доступные для конфигурирования:

- FW_Update.txt - файл для перехода в режим обновления прошивки. Для этого нужно скопировать файл на виртуальный диск и перезапустить устройство, файл удалится, а устройство перейдет в режим обновления. Для возврата к работе достаточно обновить прошивку или снова перезапустить устройство,
- IO*.txt - параметры связи с устройством, скорость передачи данных, протокол, адрес в сети ModBus и др.
- SimCard*.txt - параметры для работы модема. Команда запроса баланса, APN, телефонная книга, шаблоны для SMS-сообщений,
- MQTT*.txt - параметры для телеметрии по протоколу MQTT. Имя сервера, пароли, топики, параметры связи,
- KKT*.txt - настройки для ККТ. Параметры для записи в конфигурационные таблицы ККТ. Название объекта, параметры печати чека, и все то, что можно записать через программу для настройки ККТ,
- Checks*.txt - данные для чеков. Скидки/надбавки, названия товаров/услуг, шаблоны готовых чеков.

Файлы имеют текстовый формат и открываются любым текстовым редактором. После внесения изменений в файлы настроек, их следует сохранить на виртуальном диске и перезапустить устройство (сбросить питание). Для доступа к виртуальному диску следует подключить устройство к ПК через USB-кабель. Это не обычная «флешка», на нее нельзя

копировать файлы, кроме тех, что предназначены для устройства. «Безопасное извлечение» после работы не обязательно.

- Если во время записи файлов с настройками произошло отключение питания, то операционная система ПК может перестать корректно работать с виртуальным диском. Для устранения этой проблемы необходимо отформатировать диск: зайти в «Мой компьютер» и нажать правой кнопкой мыши на виртуальном диске, выбрать пункт меню «Форматировать». В окне форматирования диска СНЯТЬ галочку «Быстрое форматирование», и начать процесс. Система сообщит, что не удастся завершить форматирование, следует проигнорировать сообщение и закрыть окно форматирования. Устройство снова готово к работе. Теперь необходимо заново скопировать файлы настроек и перезапустить устройство.

2.1.1. Файл IO*.txt

Ниже представлено описание основных блоков файла (описание формата см.

[Приложение В. Формат JSON](#)):

```
"COM_1": // Настройки для порта COM_1
{
  "BaudRate":9600, // Скорость передачи данных 1200, 2400, 4800, 9600, 19200, 38400,
                  57600, 115200
  "BitsCount":8, // Количество бит данных 7, 8 или 9
  "Parity": "n", // Четность n-нет, e-чет., o-нечет.
  "StopBits":1, // Количество стоповых бит 1 или 2
  "ModBus_Address":10, // Адрес устройства в сети ModBus
},
```

2.2. Обновление прошивки

Новые версии прошивок выпускаются при расширении функционала и внесении исправлений. Некоторые обновления обязательны, некоторые рекомендуются. Описание изменений по версиям прошивок всегда можно посмотреть в файле Errata.txt в архиве с прошивкой. Обновить прошивку возможно несколькими способами, описанными ниже.

- После загрузки в режиме обновления важно чтобы на виртуальный диск не копировались никакие файлы кроме самой прошивки. Если на диск попал любой файл не относящийся к прошивке (системные файлы или файлы пользователя), то процесс обновления будет некорректен. Следует выключить питание устройства и заново начать обновление одним из описанных ниже способов.

2.2.1. Обновление через файл

- Установить связь между устройством и компьютером через порт USB,
- Скопировать на виртуальный диск устройства файл «FW_Update.txt»,
- Выключить питание и снова включить его,
- Устройство перейдет в режим обновления прошивки. Светодиод 2 начнет циклично мигать: 3 коротких вспышки - 3 длинных вспышки - 3 коротких вспышки,
- На компьютере появится новый диск с названием «FW Update»,

- Скопировать файл прошивки с расширением .upd на диск,
- Устройство прошивается во время копирования файла, не прерывайте процесс. Нельзя отключать кабель USB или питание.
- Светодиоды могут перестать мигать на время до 2-х минут, при этом диск для обновления прошивки пропадет из системы, это нормально,
- После окончания процесса обновления, устройство перезагрузится и начнет подмигивать как положенно для данной модели,
- Обновление прошивки завершено, можно продолжить обычное использование устройства. Если в процессе обновления произошел сбой, работа устройства в штатном режиме будет невозможна, тогда следует перейти к режиму аварийного обновления прошивки.

2.2.2. Обновление через конфигуратор (виртуальный COM-порт)

- Подать питание,
- Установить связь между устройством и компьютером через порт USB. Установить драйвера виртуального последовательного порта при необходимости.
- Запустить программу Конфигуратор, выбрать порт, к которому подключено устройство, и нажать кнопку «Connect». В списке справа выбрать появившееся устройство. Отобразится окно настроек,
- Нажать на кнопку «Update Firmware», после чего устройство перейдет в режим обновления прошивки. Светодиод 2 начнет циклично мигать: 3 коротких вспышки - 3 длинных вспышки - 3 коротких вспышки,
- На компьютере появится новый диск с названием «FW Update»,
- Скопировать файл прошивки с расширением .upd на диск,
- Устройство прошивается во время копирования файла, не прерывайте процесс. Нельзя отключать кабель USB или питание.
- Светодиоды могут перестать мигать на время до 2-х минут, при этом диск для обновления прошивки пропадет из системы, это нормально,
- После окончания процесса перепрошивки, устройство перезагрузится и начнет подмигивать как положенно для данной модели,
- Обновление прошивки завершено, можно продолжить обычное использование устройства. Если в процессе обновления произошел сбой, работа устройства в штатном режиме будет невозможна, тогда следует перейти к режиму аварийного обновления прошивки.

2.2.3. Аварийный режим обновления

В следствии неправильной эксплуатации или сбоя в процессе установки прошивки штатный вход в процедуру обновления будет невозможен. Для аварийной прошивки устройства необходимо:

- Снять питание с устройства,
- Нажать кнопку Reset (при наличии), либо разобрать корпус и замкнуть контакты Reset на печатной плате,
- Подать питание на устройство, после чего оно перейдет в режим обновления прошивки. Светодиод 2 начнет циклично мигать: 3 коротких вспышки - 3 длинных вспышки - 3 коротких вспышки,
- Отпустить кнопку или снять перемычку с контактов,

- Установить связь между устройством и компьютером через USB-порт. На компьютере появится новый диск с названием «FW Update»,
- Скопировать файл прошивки с расширением .upd на диск,
- Устройство прошивается во время копирования файла, не прерывайте процесс. Нельзя отключать кабель USB или питание.
- Светодиоды могут перестать мигать на время до 2-х минут, при этом диск для обновления прошивки пропадет из системы, это нормально,
- После окончания процесса обновления, устройство перезагрузится и начнет подмигивать как положенно для данной модели,
- Обновление прошивки завершено, можно продолжить обычное использование устройства. Если в процессе обновления произошел сбой, работа устройства будет невозможна, тогда следует начать процедуру аварийного обновления прошивки сначала.

3. Описание общих регистров

Это набор регистров в памяти устройства доступный для чтения и для записи, диапазон от 0x0100 до 0x0131, всего 50 регистров. Общие регистры начинаются с командного регистра по адресу 0x0100, назначение остальных регистров может меняться в зависимости от того, какая команда будет запущена на выполнение. Такой массив позволяет легко работать с функциями требующими длительного времени выполнения (например запрос баланса SIM-карты может длиться до 10 секунд, тем временем ПЛК может дать ошибку из-за отсутствия ответа от устройства уже через 100 мс. С командным регистром ошибки по таймауту не произойдет, так как ответ о принятой команде будет отправлен сразу). Также он очень удобен, когда требуется получить выборочный доступ к некоторым регистрам, а затем совершить какое-то действие (например команда “Печать чека” может содержать более 50 регистров и за один раз ПЛК технически не сможет передать их все). Все регистры из массива (0x0100 - 0x0131) доступны как для записи, так и для чтения, с некоторыми ограничениями. После того как задача запущена на выполнение запись в любой из регистров никак не может повлиять на выполнение команды (изменить номер команды или статус невозможно) до тех пор, пока она не завершит свою работу. Чтобы понять на какой стадии выполнения находится команда (0x0101) и узнать номер команды (0x0100), можно свободно читать соответствующие регистры. Как только команда завершит работу, регистр статуса (0x0101) изменит свое состояние на один из вариантов (команда выполнена успешно или ошибка выполнения) после этого, в случае успешного выполнения команды свободные регистры (0x0102 - 0x0131) будут содержать результат выполнения команды, если команда предполагает возврат результата работы.

-  Программист может представить командный регистр как функцию, в которую в качестве аргумента передается буфер с параметрами (регистры 0x0102 - 0x0131). После запуска, она выполняется и возвращает результат обратно в этот же буфер.
-  Чтение результата выполнения команды не является обязательным.

3.1. Форматы данных

Если первым символом является буква U, значит переменная может иметь только положительные значения (без знака):

- Uint8 - беззнаковый 8 бит, значения от 0 до 255 (0xLL),
- Int8 - знаковый 8 бит, значения от -128 до 127 (0xLL),
- Uint16 - беззнаковый 16 бит, значения от 0 до 65535 (0xHHLL),
- Int16 - знаковый 16 бит, значения от -32768 до 32767 (0xHHLL),
- Uint32 - беззнаковый 32 бит, значения от 0 до 4294967295 (0xHHHL LHLL),
- Int32 - знаковый 32 бит, значения от -2147483648 до -2147483647 (0xHHHL LHLL),
- Uint64 - беззнаковый 64 бит, значения от 0 до ($2^{64} - 1$) (0x HHHH HHLL LLHH LLLL),
- Int64 - знаковый 64 бит, значения от $-(2^{63})$ до ($2^{63} - 1$) (0x HHHH HHLL LLHH LLLL),

3.2. Командный регистр

0x0100	Командный регистр	R	Struct.
		W	50

Представляет собой пространство из 50 регистров с адресами от 0x0100 до 0x0131.

Принцип действия:

Регистр по адресу 0x0100 всегда содержит номер выполняемой команды из списка поддерживаемых команд,

Регистр 0x0101 это статус выполнения команды:

- 1 Ошибка выполнения команды,
- 0 Командный регистр еще не запускался,
- 1 Запустить выполнение команды,
- 2 Команда выполняется,
- 3 Команда выполнена успешно. Результат готов для чтения.

Регистры 0x0102 - 0x0131 это свободные регистры (регистры с параметрами).

Назначение и используемое количество регистров зависит от того, какую команду необходимо выполнить. Доступ к регистрам может быть осуществлен в произвольном порядке, например нужно отправить SMS-сообщение, для этого есть команда 4 (детальное описание команд см. ниже).

Можно сразу произвести запись пакета регистров:

- 0x0100 (номер команды) = 4 (отправить SMS-сообщение),
- 0x0101 (статус выполнения) = 1 (запустить команду на выполнение),
- 0x0102 (индекс строки в телефонной книге) = 1 (вторая строка имеет индекс 1)
- 0x0103 (индекс строки с текстом) = 2 (взять третью строку из раздела Strings в файле simcard.txt)

Либо:

1) записать регистры с параметрами команды:

0x0102 (индекс строки в телефонной книге) = 1 (вторая строка имеет индекс 1)

0x0103 (индекс строки с текстом) = 2 (взять третью строку из раздела Strings в файле simcard.txt)

2) записать номер команды:

0x0100 (номер команды) = 4 (отправить SMS-сообщение),

3) запустить команду на выполнение:

0x0101 (статус выполнения) = 1 (запустить команду на выполнение),

Главное условие, чтобы:

-либо все используемые регистры были записаны с адреса 0x0100 одной операцией,

-либо подготовить все регистры, а затем запустить командный регистр на выполнение, записав «1» в регистр 0x0101.

Структура.

Регистр 0x0100: Номер команды,

Регистр 0x0101: Статус выполнения,

Регистр 0x0102: Регистр с параметрами 0,

Регистр 0x0103: Регистр с параметрами 1,

.....

Регистр 0x0131: Регистр с параметрами 47,

4. Модем

При наличии SIM-карты и наличии команд на передачу данных, модем поддерживает соединение с Интернет. Если появляются проблемы со связью, либо необходимо сменить IP-адрес, то модем автоматически выполняет переподключение к сети. Необходимость смены IP-адреса возникает в тех случаях, когда сервер Оператора Фискальных Данных (ОФД для ККТ) или Брокер (для телеметрии) блокируют подключения для текущего IP-адреса. Если смена IP-адреса не дает никакого эффекта, то модем будет пытаться подключиться к серверу пока не будет превышен определенный порог, затем снова переподключится к Интернет для смены IP.

4.1. Файл SimCard*.txt

Ниже представлено описание основных блоков файла (описание формата см.

[Приложение В. Формат JSON](#)):

```
"SIMCARD":
{
    "USSD_GetBalance": "#100#", // Команда на запрос баланса (ответ должен быть на
                                // английском)
    "Call_RetryCount": 5, // Количество попыток дозвона (команда 6)
    "APN": "internet", // Описание APN-точки доступа в интернет
    "APN_USER": "internet",
    "APN_PASS": "",
},

"PhoneBook": // Телефонная книга для дозвона или отправки SMS
[
    "+71112223344",
    "81234445566",
    "+71234445566",
],

"Strings": // Строки для отправки в SMS-сообщениях
[
    "Нет тяги", // Обычная строка с текстом
    "Авария %d для котла №%d", // Строка с динамически подставляемыми данными
                                // (вместо "%d")

    "Строка 4",
    "Строка 5",
],
```

4.2. Команды

0x0100	0	Имя оператора сети GSM	R	Struct.
			W	3 - 33

Запросить имя оператора из сети GSM. Ответ содержит строку в ASCII символах.

- См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,
 Регистр 0x0101: Статус выполнения,
 Регистр 0x0102: ASCII символ первого знака,
 Регистр 0x0103: ASCII символ второго знака,

 Регистр 0x0120: см. описание регистра 0x0102.

0x0100	1	Баланс SIM-карты	R	Struct.
			W	4

Получить баланс SIM-карты, USSD-команда «#100#». Ответ содержит сумму остатка средств на счете в копейках. USSD-команду можно изменить в разделе “USSD_GetBalance” файла simcard.txt

- См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,
 Регистр 0x0101: Статус выполнения,
 Регистр 0x0102 - 0x0103: (Int32) Баланс карты в копейках.

0x0100	2	Качество принимаемого сигнала GSM	R	Struct.
			W	4

Запросить качество принимаемого сигнала GSM. Ответ: значение от -114 дБ до -51 дБ. 99 означает что нет принимаемого сигнала.

- См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,
 Регистр 0x0101: Статус выполнения,
 Регистр 0x0102 - 0x0103: (Int32) Качество принимаемого сигнала.

0x0100	3	Напряжение питание модема	R	Struct.
			W	4

Запросить напряжение питания модема внутри терминала. Ответ: напряжение питания в * 0,01 Вольт.

-  См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,
 Регистр 0x0101: Статус выполнения,
 Регистр 0x0102 - 0x0103: (Uint32) Напряжение питания модема в (x * 0,01) Вольт.

0x0100	10	IMEI-код модема	R	Struct.
			W	18

Запросить IMEI-код модема. Ответ содержит строку в ASCII символах.

-  См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,
 Регистр 0x0101: Статус выполнения,
 Регистр 0x0102: ASCII символ первого знака,
 Регистр 0x0103: ASCII символ второго знака,

 Регистр 0x0111: см. описание регистра 0x0102.

0x0100	11	IP-адрес модема при наличии подключения к Интернет	R W	Struct.
				18

Запросить IP-адрес модема при наличии подключения к Интернет, если модем отключается от сети, то ответом будет строка из 16 нулей. Ответ содержит строку в ASCII символах.

- См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,
 Регистр 0x0101: Статус выполнения,
 Регистр 0x0102: ASCII символ первого знака,
 Регистр 0x0103: ASCII символ второго знака,

 Регистр 0x0111: см. описание регистра 0x0102.

0x0100	15	Принудительный сброс модема.	R W	Struct.
				3

Команда принудительно производит сброс модема и перерегистрацию в сети GSM.

- Начиная с версии прошивки 2.0.2, статус регистрации в сети проверяется каждые 30 секунд. Если модем не может пройти регистрацию в сети в течение 10 минут, Терминал автоматически проведет сброс модема.
- Сброс модема затрагивает только часть отвечающую за SIM-карту, Модем, Информатор и интернет будут временно недоступны. Переменные телеметрии MQTT сохраняться, а работа с ККТ будет продолжаться в штатном режиме.
- См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Структура:

Регистр 0x0100: Номер команды,
 Регистр 0x0101: Статус выполнения,
 Регистр 0x0102: Для сброса модема записать в регистр 0xC0,

0x0200	Флаги состояния Терминала	R	Uint16
			1

Флаги текущего состояния блока для работы с Модемом. Состояние сети обновляется каждые 30 секунд.

Регистр 0x0200: Флаги состояния.

Бит 15: Регистрация в домашней сети (есть возможность звонить/отправлять SMS),

Бит 14: Регистрация в сети, роуминг,

Бит 13: Сеть предоставляет возможность доступа в Интернет,

Биты 12-3: Зарезервированы,

Бит 2: Статус последней команды на отправку SMS-сообщения. Если сообщение отправлено успешно-1, иначе-0,

Бит 1: Статус последней команды голосового вызова на номер абонента. Если вызов прошел и абонент ответил-1, иначе-0,

Бит 0: Наличие подключения к Интернет.

4.3. Таблица регистров

Адрес	CMD	R W	Кол-во регистров	Описание	Значение	Заводское значение
0x0100	0	R W	3-33	Имя оператора сети GSM	Текст ASCII	0
	1		4	Баланс карты в копейках	Int32	0
	2		4	Качество принимаемого сигнала GSM	от -114 дБ до -51, 99 – нет сигнала	0
	3		4	Напряжение питание модема	Напряжение (x * 0,01) Вольт	4200
	10		18	IMEI-код модема	Текст ASCII	0
	11		18	IP-адрес модема при наличии подключения к Интернет	Текст ASCII	0
	15		3	Принудительный сброс модема	0xC0	0
0x0200		R	1	Флаги состояния Терминала	UInt16	-

5. Телеметрия MQTT

Сервис телеметрии доступен в виде приема/передачи 64-битных переменных. В общем случае нужна только оплата за использованный Интернет на SIM-карте, причем оператора можно выбрать свободно и SIM-карта может быть любой, без статического IP-адреса. Телеметрия будет выглядеть для оператора как обычный выход в Интернет с сотового телефона. Брокера можно выбрать свободно на усмотрение пользователя, например бесплатный `hivemq.com` или платные сервисы от IBM, для тестового доступа создан бесплатный аккаунт на сайте `cloudmqtt.com`. Это популярный брокер несколько платных и бесплатный тариф. После создания аккаунта нужно будет только записать настройки в Терминал и можно пользоваться телеметрией. На телефон устанавливается приложение-клиент, их существует огромное множество как платных, так и бесплатных, отличаются качеством работы и «красотой» отображения информации, но все получают эту информацию совершенно одинаково, найти можно в GoogleMarket или AppStore по запросу «MQTT» или «IoT». Простой и рабочий тестовый вариант – приложение «LinerMQTT». Никто из клиентов не знает друг о друге. Важно только чтобы все клиенты интерпретировали информацию из одного топика одинаково. Если первый клиент передает температуру, то второй должен понимать, что он подписался на температуру и третий не должен публиковать свою температуру в тот-же топик, хотя это относится только к примеру. В действительности сколько угодно клиентов могут публиковать сообщения в один топик и сколько угодно сразу могут быть подписаны на него (даже одновременно публиковать и быть подписанным на один топик).

Бесплатный доступ к телеметрии с устройства «Терминал» допускает чтение и публикацию до 10-ти 64-битных переменных. Лицензия на большее количество переменных может быть предоставлена по запросу.

Поток данных может выглядеть следующим образом. ПЛК передает значения переменных в регистры Терминала, Терминал подключается к брокеру, создает топик для каждой переменной и публикует в него данные. Запись в Терминал можно проводить сколько угодно раз, но публикация будет производиться, только если данные изменились (чтобы не дублировать сообщения и не засорять канал), в текущей аппаратной реализации средняя скорость публикации: 2 переменных в секунду. Сообщения публикуются на сервере (брокер), а он отправляет его всем своим подписчикам. Допустим, на нескольких телефонах запущено приложение LinerMQTT, и в них есть подписка на этот топик, тогда они все будут получать отправленные значения переменной.

Обратный поток будет выглядеть с точностью до наоборот. Любой из операторов может опубликовать значение ползунка или состояние кнопки в топик, сообщения попадут на сервер, а Терминалы, подписанные на него получают это значение и сохраняют в памяти для телеметрии. Далее ПЛК может прочитать состояние переменных телеметрии (чтение внутренней памяти Терминала доступно всегда, но после включения и пока Терминал не подключился к брокеру, во всех переменных телеметрии находятся нули).

5.1. Параметры связи терминала с Брокером

Далее перечислены параметры связи по умолчанию. Хранятся они в блоке «MQTT», в файле настроек `mqtt.txt` на виртуальном диске устройства и доступны для изменения пользователем. Если строка с настройкой отсутствует в файле, то будет применено значение по умолчанию, указанное в скобках. Номер устройства можно узнать в программе «Конфигуратор», переведя полученное значение из шестнадцатеричного в десятичный вид.

- `Server` = Адрес Брокера в сети Интернет,

- Port = Номер порта для подключения к Брокеру,
- User = («Terminal») Имя пользователя,
- Password = («Terminal_номер устройстваDec») Пароль,
- TopicTranceive = («Terminal/номер устройстваDec/Sub») Корень топика, куда Терминал будет публиковать свои данные. Это поле должно совпадать с названием шаблона для подписок Sub, созданного на Брокере, исключая «/#»,
- TopicReceive = («Terminal/номер устройстваDec/Pub») Корень топика, на все сообщения которого Терминал будет подписан. Это означает что Терминал будет принимать топика с любыми названиями из указанного корня, но обрабатываться будут только топика с названиями «Корень топика/0», «Корень топика/1» ... «Корень топика/n», где n - максимально доступное количество переменных для данного Терминала,
- KeepAlive = (60) сек. Время в течение которого брокер оставляет соединение открытым для клиента,
- CleanSession = (0). 0 - указывает брокеру, что необходимо хранить данные о сессии и пакеты пока клиент не получит их. 1 - означает, что брокер может удалить данные из пакетов, если сейчас нет подписчиков на топик,
- Qos = (1). 0 - подтверждение об успешно принятом сообщении от брокера не требуется, 1 - брокер обязан подтвердить успешно принятое сообщение, 2 - двукратное подтверждение,
- Retained = (1). 0 - сообщение попадет к подписчику, только если оно будет передано к брокеру, когда подписчики в сети. 1-сохранять последнее опубликованное сообщение и выдавать клиентам сразу при подписке на тему, даже если публикующий клиент уже отключился от сети,
- DelRetainedInRx = (0). 0 - сообщения принимаются «как есть» с любым значением атрибута Retained, устанавливаемого публикующей стороной. 1 - после получения сообщения, терминал будет пытаться снять атрибут Retained с топика (даст команду брокеру на удаление последнего сообщения из топика).

5.2. Файл MQTT*.txt

Ниже представлено описание основных блоков файла (описание формата см.

[Приложение В. Формат JSON](#)):

```
"MQTT":
{
  "Server":"m10.cloudmqtt.com",      // Адрес сервера Брокера
  "Port":18192,                      // Номер порта
  "User":"RooTerminalTest",          // Логин
  "Password":"qwerty",              // Пароль

  "TopicTranceive":"RooTerminal/Test/Sub", // Топик для передачи от Терминала
  "TopicReceive":"RooTerminal/Test/Pub",   // Топик для приема к Терминалу

  "Qos":1,                            // Качество сервиса 0, 1 или 2
  "Retained":1,                        // Сохранять сообщения у Брокера 0-нет, 1-да
  "DelRetainedInRx":0,                // Снимать атрибут Retained, после приема
                                     // Терминалом. 0-нет, 1-да

  "KeepAlive":60,                     // Соединение с Брокером оставлять открытым x сек.
  "CleanSession":0,                   // Брокер должен хранить данные о сессии 0-да, 1-нет
},
```

5.3. Команды

0x0100	1	Баланс карты	R	Struct.
			W	4

Команда запрашивает баланс карты, USSD-команда “#100#”. Ответ содержит сумму остатка средств на счете в копейках. USSD-команду можно изменить в разделе “USSD_GetBalance” файла simcard.txt

 См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,
 Регистр 0x0101: Статус выполнения,
 Регистр 0x0102 - 0x0103: (Int32) Баланс карты в копейках.

0x0200	Флаги состояния Терминала	R	Uint16
			1

Флаги текущего состояния блока для работы с Информатором. Состояние сети обновляется каждые 30 секунд.

Регистр 0x0200: Флаги состояния.

Бит 15: Регистрация в домашней сети (есть возможность звонить/отправлять SMS),

Бит 14: Регистрация в сети, роуминг,

Бит 13: Сеть предоставляет возможность доступа в Интернет,

Биты 12-3: Зарезервированы,

Бит 2: Статус последней команды на отправку SMS-сообщения. Если сообщение отправлено успешно-1, иначе-0,

Бит 1: Статус последней команды голосового вызова на номер абонента. Если вызов прошел и абонент ответил-1, иначе-0,

Бит 0: Наличие подключения к Интернет.

0x1000	Флаги состояния Телеметрии	R	Uint16
			1

Флаги текущего состояния блока для работы с Телеметрией.

Регистр 0x1000: Флаги состояния.

Биты 15-1: Зарезервированы,

Бит 0: Наличие подключения к Брокеру.

0x1100	Доступ к буферу телеметрии по 16 бит	R W	Int16
			1 - 100

Обращение к одной переменной или диапазону переменных телеметрии, так, если бы каждая переменная имела длину 16 бит.

Порядок байт в регистрах:

0x HHLL,

.....

0x HHLL.

Регистр 0x1100: 16 бит переменной 0,

Регистр 0x1101: 16 бит переменной 1,

.....

Регистр 0x1163: 16 бит переменной 99.

0x1300	Доступ к буферу телеметрии по 32 бита	R W	Int32
			1 - 200

Обращение к одной переменной или диапазону переменных телеметрии, так, если бы каждая переменная имела длину 32 бита.

Порядок байт в регистрах:

0x HNNL LHLL,

.....

0x HNNL LHLL.

Регистр 0x1300: Младшие 16 бит переменной 0,

Регистр 0x1301: Старшие 16 бит переменной 0,

Регистр 0x1302: Младшие 16 бит переменной 1,

Регистр 0x1303: Старшие 16 бит переменной 1,

.....

Регистр 0x13C6: Младшие 16 бит переменной 99,

Регистр 0x13C7: Старшие 16 бит переменной 99.

0x1700	Доступ к буферу телеметрии по 64 бита	R W	Int64
			1 - 400

Обращение к одной переменной или диапазону переменных телеметрии, так, если бы каждая переменная имела длину 64 бита.

Порядок байт в регистрах:

0x НННН ННЛЛ ЛЛНН ЛЛЛЛ,

.....

0x НННН ННЛЛ ЛЛНН ЛЛЛЛ.

Регистр 0x1700: Младшие младшие 16 бит переменной 0,

Регистр 0x1701: Младшие старшие 16 бит переменной 0,

Регистр 0x1702: Старшие младшие 16 бит переменной 0,

Регистр 0x1703: Старшие старшие 16 бит переменной 0,

Регистр 0x1704: Младшие младшие 16 бит переменной 1,

Регистр 0x1705: Младшие старшие 16 бит переменной 1,

Регистр 0x1706: Старшие младшие 16 бит переменной 1,

Регистр 0x1707: Старшие старшие 16 бит переменной 1,

.....

Регистр 0x188C: Младшие младшие 16 бит переменной 99,

Регистр 0x188D: Младшие старшие 16 бит переменной 99,

Регистр 0x188E: Старшие младшие 16 бит переменной 99,

Регистр 0x188F: Старшие старшие 16 бит переменной 99.

5.4. Таблица регистров

Адрес	CMD	R W	Кол-во регистров	Описание	Значение	Заводское значение
0x0100	1	R W	4	Баланс карты в копейках	Int32	0
0x0200		R	1	Флаги состояния Терминала	UInt16	-
0x1000		R	1	Флаги состояния Телеметрии	UInt16	-
0x1100 ... 0x1163		R W	1 - 100	Доступ к буферу телеметрии по 16 бит	Int16	0
0x1300 ... 0x13C7		R W	1 - 200	Доступ к буферу телеметрии по 32 бита	Int32	0
0x1700 ... 0x188F		R W	1 - 400	Доступ к буферу телеметрии по 64 бита	Int64	0

6. Контрольно-кассовая техника ККТ

Согласно закону РФ 54-ФЗ, практически все предпринимателям и компаниям, а также вендинговым автоматам (Кофе, шоколадки и др.), мойкам самообслуживания, всем кто производит денежные операции, необходимо выдавать кассовый чек покупателю по установленным стандартам. Информацию о проведенной денежной операции необходимо предоставить в Налоговые органы (ФНС) и сохранить на специальном носителе информации, который регистрируется в ФНС, как и сам кассовый аппарат. Носитель называется ФН - фискальный накопитель. Электронным посредником для передачи данных от кассового аппарата является ОФД - оператор фискальных данных. Для соблюдения закона необходимо приобрести сам ККТ, принтер чеков и ФН, заключить договор с ОФД и зарегистрировать все это в налоговой. Обычно на месте оператора находится компьютер, к которому подключен Интернет, и касса. На нем запущена специальная торговая программа, которая управляет кассовым аппаратом и передает данные в ОФД. Но что делать, если это автомат, который стоит на улице и не предполагает присутствие оператора? Необходимо установить в него компьютер, модем, сторож для сброса модема, кассу, и как-то завязать всю автоматику, монето- и купюро- приемники с компьютером. Далеко не каждый автомат можно так «модернизировать» и тем более затраты будут очень велики, кроме того, компьютер это потенциально ненадежная система с жестким диском, которые боятся низких и высоких температур, а программное обеспечение и операционная система дополнительно снижают надежность всей системы, что приведет к необходимости более частого, квалифицированного обслуживания.

Теперь для работы с кассой вам не нужен компьютер, модем и другое дополнительное оборудование, не нужен оператор, достаточно подключить кабель от кассы к Терминалу и вставить в него SIM-карту. Ваша система (торговый автомат) принимает деньги от клиента и дает команду Терминалу «напечатать чек», остальное он сделает сам.

6.1. Поток данных при работе с кассой.

Клиент (пользователь) вносит деньги и выбирает товар (или услугу), система обрабатывает запрос, и дает команду в кассу. Касса (ККТ) делает запись в Фискальный Накопитель (ФН) и передает Оператору Фискальных Данных (ОФД). ОФД передает данные в налоговую и формирует ответ, что информация о чеке верная. Далее ККТ получает ответ от ОФД, записывает в ФН подтверждение и выдает чек покупателю. ККТ сообщает системе «чек выдан», значит все в порядке. Услуга или товар выдаются клиенту. Довольно сложный список действий, но это все «кухня» системы и вникать в нее более нет необходимости.

Терминал не требует специальных личных кабинетов и дополнительных абонентских плат. Нужна только SIM-карта с доступом в интернет. Строки с наименованием товаров/услуг можно задать непосредственно в команде на печать чека. Либо один раз сформировать файл с настройками и текстами для чеков, и в команде просто указать номер строки с нужным текстом.

6.2. Файл Checks*.txt

Ниже представлено описание основных блоков файла (описание формата см. [Приложение В. Формат JSON](#)):

```
"Defaults": // Значения принимаемые по умолчанию
{
```

```
"NalogGroup1":0, // Налоговая ставка. Индекс строки из таблицы 6 в ККТ
"DepartmentNumber":1, // Наименование отдела. Индекс строки из таблицы 5 в ККТ

"EndStringNumber":0, // Индекс последней строки в чеке (из блока "EndStrings")
},

"Templates": // Шаблоны с параметрами для чеков
{
  "Т_0": // Шаблон 0
  {
    "DiscountSurcharge_Percent":"0", // Размер скидки/надбавки на товар в 0.01%
  },
},

"LineStrings": // Строки с названиями товаров/услуг
[
  "Жетон",
  "Депозит на карту",
  "Услуга 1",
  "Услуга 2",
],

"EndStrings": // Строки для последней строки в чеке
[
  "Спасибо",
  "Спасибо за покупку!",
  "Спасибо, ждем вас снова!",
],

"Checks": // Шаблоны заранее приготовленных чеков (для команды 50)
{
  "Check_0": // Чек 0
  [
    {
      "LineStringNumber":0, // Индекс строки из блока "LineStrings"
      "Quantity":"1", // Количество товара/услуги, может быть дробным
                        "0.001" шт.
      "Price":"10", // Цена в рублях, может быть дробной "0.01" руб.
    },
    {
      "EndStringNumber":2, // Индекс строки из блока "EndStrings"
    },
  ],
  "Check_1": // Чек 1
  [
    { // строка с товаром 0
      "LineString":"Перчатки (пара)", // название товара/услуги
      "Quantity":"1", // Количество
      "Price":"77.50", // Цена
    },
    {"LineString":"Шапка", "Quantity":"3", "Price":"110.00", }, // строка с товаром 1
    {"LineStringNumber":2 "Quantity":"1", "Price":"1", }, // строка с товаром 2
    {"EndStringNumber":1, }, // последняя строка в чеке
  ],
},
},
```

6.3. Файл ККТ*.txt

Ниже представлено описание основных блоков файла (описание формата см. [Приложение В. Формат JSON](#)):

```
"TablesCount":30,      // Количество таблиц настроек для ККТ
"Tables":             // Таблицы, которые будут записаны при включении Терминала
{
    "Т17R1F10": "1",   // Т-номер таблицы, R-Номер ряда, F-Номер поля. Значение для
                       // записи - "1"
},
```

6.4. Команды

0x0000	Флаги состояния	R	Uint16
			1

Флаги текущего состояния блока для работы с ККТ.

Регистр 0x0000: Флаги состояния.

Биты 15-1: Зарезервированы,

Бит 0: Наличие связи с ККТ.

0x0001	Количество ошибок связи с ККТ	R	Uint16
			1

Счетчик ошибок связи. При достижении значения 40, терминал сбрасывает счетчик и пытается восстановить связь с ККТ.

Регистр 0x0001: Счетчик ошибок связи.

0x0002	Запросить код ошибки последней неудачно выполненной команды ККТ	R	Int32
			2

Команда возвращает последний код ошибки, если одна из команд ККТ вернула его, по умолчанию значение = 0. Код ошибки в процессе работы не сбрасывается и служит как информация для диагностики проблем с ККТ. Коды ошибок зависят от производителя и протокола используемого ККТ. По этому точное описание кода ошибки следует искать в описании протокола конкретного производителя (Описание кодов ошибок находится в документации к ККТ, файл Protocol_KKT_FN_1-16_Paykiosk.pdf, стр.61).

Регистр 0x0002 - 0x0003: (Int32) Код ошибки.

0x0100	1	Баланс SIM-карты	R W	Struct.
				4

Получить баланс SIM-карты, USSD-команда «#100#». Ответ содержит сумму остатка средств на счете в копейках. USSD-команду можно изменить в разделе “USSD_GetBalance” файла simcard.txt

- См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,
 Регистр 0x0101: Статус выполнения,
 Регистр 0x0102 - 0x0103: (Int32) Баланс карты в копейках.

0x0100	40	Статус информационного обмена (команда ККТ 0xFF39)	R	Struct.
			W	9

Запросить из ККТ статус информационного обмена.

- См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,
 Регистр 0x0101: Статус выполнения,
 Регистр 0x0102: Статус информационного обмена,
 Регистр 0x0103: Состояние чтения сообщения,
 Регистр 0x0104: Количество сообщений ожидающих передачу в ОФД,
 Регистр 0x0105 - 0x0106: (UInt32) Номер первого в очереди документа ожидающего передачу в ОФД,
 Регистр 0x0107-0x0108: (UInt32) Дата и время первого в очереди документа ожидающего передачу в ОФД.

0x0100	41	Запросить пароль администратора из памяти ККТ	R	Struct.
			W	4

Запросить пароль администратора из памяти ККТ (Заводская установка = 30).

- См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,
 Регистр 0x0101: Статус выполнения,
 Регистр 0x0102 - 0x0103: (UInt32) Пароль администратора.

0x0100	42	Запросить пароль налогового инспектора из памяти ККТ	R	Struct.
			W	4

Запросить пароль налогового инспектора из памяти ККТ.

-  См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,

Регистр 0x0101: Статус выполнения,

Регистр 0x0102 - 0x0103: (Uint32) Пароль налогового инспектора.

0x0100	43	Параметры модели ККТ (команда ККТ 0xF7)	R	Struct.
			W	5-20

Команда получения настроек ККТ. Позволяет получить: поддержка датчиков (А НЕ ИХ СОСТОЯНИЕ), пароли системного администратора и налогового инспектора, и т.д.. Команда и ответ ККТ будут обработаны автоматически, остается только указать смещение и количество байт для доступа к данным ответного сообщения.

Например, чтобы получить 1 байт ответного сообщения, начиная с байта 5, следует записать в регистр смещения 5, а в регистр количества байт 1. В каждом регистре будет находиться по 2 байта ответа, сначала старший, затем младший. 0xHLLL.

-  См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Некоторые байты ответа для ККТ Paykosk:

- Байт 0 - Параметры модели (Uint64),
 - бит 1 - Весовой датчик чековой ленты поддерживается,
 - бит 3 - Оптический датчик чековой ленты поддерживается,
 - бит 4 - Датчик крышки поддерживается,
 - бит 6 - Датчик рычага термоголовки чековой ленты поддерживается,
- ...
- Байт 15 - Пароль налогового инспектора (Uint32),
- Байт 19 - Пароль системного администратора (Uint32),
- ...

-  Полное описание байт ответа на эту команду находится в документации к ККТ (файл Protocol_KKT_FN_1-16_Paykiosk.pdf, стр.37, команда 0xF7).

Описание регистров:

Регистр 0x0100: Номер команды,

Регистр 0x0101: Статус выполнения,

Регистр 0x0102: Смещение в байтах. Это номер байта ответа, с которого начать вывод в регистры (0x0104 ..),

Регистр 0x0103: Запрашиваемое количество байт ответа от ККТ,

Регистр 0x0104: Байты 1 и 0 ответа от ККТ, учитывая смещение заданное в регистре (0x0102),

...

Данные в остальных регистрах будут зависеть от количества запрошенных байт ответа (в регистре 0x0103).

0x0100	44	Короткий запрос состояния ККТ (команда ККТ 0x10)	R	Struct.
			W	5-12

Позволяет получить: наличие бумаги в принтере и состояния других датчиков, режимы работы ККТ, и т.д.. Команда и ответ ККТ будут обработаны автоматически, остается только указать смещение и количество байт для доступа к данным ответного сообщения.

Например, чтобы получить 1 байт ответного сообщения, начиная с байта 5, следует записать в регистр смещения 5, а в регистр количества байт 1. В каждом регистре будет находиться по 2 байта ответа, сначала старший, затем младший. 0xHHLL.

 См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Некоторые байты ответа для ККТ Paykosk:

- Байт 0 - Порядковый номер оператора (UInt8),
- Байт 1 - Флаги ККТ (UInt16),
 - бит 1 - Рулон чековой ленты (0 - нет, 1 - есть),
 - бит 7 - Оптический датчик чековой ленты (0 - бумаги нет, 1 - есть),
 - бит 9 - Рычаг термоголовки чековой ленты (0 - поднят, 1 - опущен),
 - бит 10 - Крышка корпуса ККТ (0 - опущена, 1 - поднята),
- Байт 3 - Режим ККТ (UInt8),
- Байт 4 - Подрежим ККТ (UInt8),
- ...
- Байт 12 - Результат последней печати (UInt8),

 Полное описание байт ответа на эту команду находится в документации к ККТ (файл Protocol_KKT_FN_1-16_Paykiosk.pdf, стр.16 команда 0x10).

Описание регистров:

Регистр 0x0100: Номер команды,

Регистр 0x0101: Статус выполнения,

Регистр 0x0102: Смещение в байтах. Это номер байта ответа, с которого начать вывод в регистры (0x0104 ..),

Регистр 0x0103: Запрашиваемое количество байт ответа от ККТ,

Регистр 0x0104: Байты 1 и 0 ответа от ККТ, учитывая смещение заданное в регистре (0x0102),

...

Данные в остальных регистрах будут зависеть от количества запрошенных байт ответа (в регистре 0x0103).

0x0100	45	Длинный запрос состояния ККТ (команда ККТ 0x11)	R	Struct.
			W	5-23

Позволяет получить: наличие бумаги в принтере и состояния других датчиков, режимы работы ККТ, версию ПО, дату/время, и т.д.. Команда и ответ ККТ будут обработаны автоматически, остается только указать смещение и количество байт для доступа к данным ответного сообщения.

Например, чтобы получить 1 байт ответного сообщения, начиная с байта 5, следует записать в регистр смещения 5, а в регистр количества байт 1. В каждом регистре будет находиться по 2 байта ответа, сначала старший, затем младший. 0xHHLL.

-  См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Некоторые байты ответа для ККТ Paykosk:

- Байт 0 - Порядковый номер оператора (Uint8),
- Байт 1 - Версия ПО (Uint16),
- Байт 3 - Сборка ПО (Uint16),
- ...
- Байт 9 - Сквозной номер текущего документа (Uint16),
- Байт 11 - Флаги ККТ (Uint16),
 - бит 1 - Рулон чековой ленты (0 - нет, 1 - есть),
 - бит 7 - Оптический датчик чековой ленты (0 - бумаги нет, 1 - есть),
 - бит 9 - Рычаг термоголовки чековой ленты (0 - поднят, 1 - опущен),
 - бит 10 - Крышка корпуса ККТ (0 - опущена, 1 - поднята),
- Байт 13 - Режим ККТ (Uint8),
- Байт 14 - Подрежим ККТ (Uint8),
- ...
- Байт 26 - Номер последней закрытой смены (Uint16),
- ...

-  Полное описание байт ответа на эту команду находится в документации к ККТ (файл *Protocol_KKT_FN_1-16_Paykiosk.pdf*, стр.17 команда 0x11).

Описание регистров:

Регистр 0x0100: Номер команды,

Регистр 0x0101: Статус выполнения,

Регистр 0x0102: Смещение в байтах. Это номер байта ответа, с которого начать вывод в регистры (0x0104 ..),

Регистр 0x0103: Запрашиваемое количество байт ответа от ККТ,

Регистр 0x0104: Байты 1 и 0 ответа от ККТ, учитывая смещение заданное в регистре (0x0102),

...

Данные в остальных регистрах будут зависеть от количества запрошенных байт ответа (в регистре 0x0103).

0x0100	50	Печать чека “Продажа” по шаблону из файла	R	Struct.
			W	3

Команда на печать чека “Продажа” по шаблону. Состав чека заранее сохранен в файле Checks*.txt в блок "Checks", и представляет собой массив из блоков “Check_x”, где x - номер чека. Блок “Check_x” это массив с товарами/услугами (Текстовое название или индекс строки с текстом из блока “LineStrings”, Количество в 0.001 штук и Цена в 0.01 руб. (количество и цена задаются дробным числом, разделитель дробной части - точка) и Текст завершающей строки или индекс строки с текстом из блока “EndStrings”).

 См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,

Регистр 0x0101: Статус выполнения,

Регистр 0x0102: Номер чека из блока "Checks".

0x0100	51	Печать чека “Продажа” расширенная	R	Struct.
			W	11

Команда на печать чека “Продажа”, позволяющая задать его параметры, остальное хранится в файле Checks*.txt. Тип налогообложения, скидки/надбавки и номера отделов заданы в блоке “Templates”. Названия товаров/услуг записаны в блоке “LineStrings”. Строки с текстом завершающей строки заданы в блоке “EndStrings”.

 См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,

Регистр 0x0101: Статус выполнения,

Регистр 0x0102 - 0x0103: (Uint32) Реально полученная сумма денег в копейках. Если задана как 0, то будет посчитана автоматически и равна итогу чека,
 Регистр 0x0104: (Hi) - Номер шаблона из блока “Templates” (Налоговая группа и размер скидки/надбавки в 0.01 %. Если 0, то будут взяты значения из блока “Defaults”). (Lo) - Индекс завершающей строки из блока “EndStrings”,
 Регистр 0x0105: Количество позиции в чеке (количество товаров/услуг) от 1.
 Регистр 0x0106: (Hi) - Номер шаблона из блока “Templates” (Налоговая группа и номер отдела для товара/услуги. Если 0, то будут взяты значения из блока “Defaults”). (Lo) - Индекс строки с названием товара/услуги из блока “LineStrings”,
 Регистр 0x0107 - 0x0108: (Uint32) Количество единиц товара в 0.001 шт.
 Регистр 0x0109 - 0x010A: (Uint32) Цена товара в копейках за 1 целую единицу.
 (Регистр 0x010B): для товара 2, см. описание регистра 0x0106,
 (Регистр 0x010C - 0x010D): для товара 2, см. описание регистров 0x0107 - 0x0108,
 (Регистр 0x010E - 0x010F): для товара 2, см. описание регистров 0x0109 - 0x010A,

...

Остальные регистры задают параметры для следующих позиций в чеке и соответствуют регистрам 0x0106 - 0x010A. Количество позиций задается в регистре в регистре 0x0105.

0x0200	Флаги состояния Терминала	R	Uint16
			1

Флаги текущего состояния Терминала. Состояние сети обновляется каждые 30 секунд.

Регистр 0x0200: Флаги состояния.

Бит 15: Регистрация в домашней сети (есть возможность звонить/отправлять SMS),

Бит 14: Регистрация в сети, роуминг,

Бит 13: Сеть предоставляет возможность доступа в Интернет,

Биты 12-3: Зарезервированы,

Бит 2: Статус последней команды на отправку SMS-сообщения. Если сообщение отправлено успешно-1, иначе-0,

Бит 1: Статус последней команды голосового вызова на номер абонента. Если вызов прошел и абонент ответил-1, иначе-0,

Бит 0: Наличие подключения к Интернет.

6.5. Таблица регистров

Адрес	CMD	R W	Кол-во регистров	Описание	Значение	Заводское значение
0x0000		R	1	Флаги состояния	UInt16	-
0x0001		R	1	Количество ошибок связи с ККТ	0-40	-
0x0002		R	2	Запросить код ошибки последней неудачно выполненной команды ККТ	Int32	0
0x0100	1	R W	4	Баланс карты в копейках	Int32	0
	40		9	Запросить статус информационного обмена	Структура данных	-
	41		4	Запросить пароль администратора из памяти ККТ	UInt32	30
	42		4	Запросить пароль налогового инспектора из памяти ККТ	UInt32	-
	43		5-20	Параметры модели ККТ (команда ККТ 0xF7)	Структура данных	-
	44		5-12	Короткий запрос состояния ККТ (команда ККТ 0x10)	Структура данных	-
	45		5-23	Длинный запрос состояния ККТ (команда ККТ 0x11)	Структура данных	-
	50		3	Печать чека “Продажа” по шаблону из файла	UInt16	-
	51		11	Печать чека “Продажа” расширенная	Структура данных	-
0x0200		R	1	Флаги состояния Терминала	UInt16	-

7. Информатор

В некоторых случаях есть необходимость незамедлительно оповестить пользователя, о каком либо событии, причем доступ к Интернет может быть ограничен, тогда становится важной возможность отправки SMS-сообщений или звонков на заданный номер телефона.

Для отправки сообщения нужно дать команду на запись текста во временный буфер, а затем команду с номером телефона получателя. Команд отправки может быть неограниченное количество, на каждую команду будет отправлено сообщение с текстом из буфера. Текст может иметь длину до 512 символов русских и/или английских букв. Также доступна короткая команда отправки сообщения с заранее подготовленной строкой записанной в файле simcard.txt на номер из телефонной книги.



Следует помнить, что максимальная длина одного SMS-сообщения состоящего только из латинских букв равна 160 символов, если в тексте встречается хотя бы один символ из кириллицы, то максимальная длина сокращается до 80 символов. Если в буфере записана строка длиннее лимита для одного сообщения, то она автоматически разобьется на несколько сообщений (особенности передачи SMS-сообщений в GSM сетях). При этом оператором сотовой связи может взиматься соответствующая плата!

Строки могут быть записаны с аргументами, например когда необходимо в текст добавить изменяющуюся числовую информацию. Строка с аргументами имеет вид «Авария номер %d температура %d°С» или «Вы оплатили товар на сумму %d руб.», где символы «%d» это места в строке, которые будут подменены на соответствующие цифровые значения в регистрах. Каждая строка поддерживает до 10 аргументов, считаются они всегда слева направо от 0 до 9. Тогда если аргумент 0 будет равен 1234, аргумент 1 равен 18, то мы получим готовые строки: «Авария номер 1234 температура 18°С» и «Вы оплатили товар на сумму 1234 руб.». Если в строке задано меньше аргументов чем передано в регистрах, то лишние аргументы будут отброшены. Если же в строке задано больше аргументов чем передано, то в строку будут подставлены нули, либо значения оставшиеся в регистрах от прошлых команд. Каждый аргумент имеет размер Signed Integer (целый, 32 бита со знаком).



Буфер для итоговой строки SMS-сообщения после подмены аргументов может вместить до 612 символов.

Чтобы совершить звонок, достаточно дать команду с номером телефона. Время вызова равно 30 секунд, если абонент не поднимает трубку или сбрасывает входящий звонок, то Терминал автоматически повторит его заданное количество раз. Количество попыток дозвона находится в файле simcard.txt, параметр «Call_RetryCount» = n (3). Если все попытки оказались неудачными, то в статусе выполнения команды (регистр 0x0101) будет ошибка. Если абонент ответил на вызов, то будет поднят соответствующий флаг во флагах состояния Терминала (по адресу 0x0200), а команда завершит свое выполнение со статусом «Команда выполнена успешно». Также доступна короткая команда вызова по номеру из телефонной книги.

7.1. Файл SimCard*.txt

Ниже представлено описание основных блоков файла (описание формата см.

[Приложение В. Формат JSON](#)):

```
"SIMCARD":
{
    "USSD_GetBalance": "#100#", // Команда на запрос баланса (ответ должен быть на
                                // английском)
    "Call_RetryCount": 5,      // Количество попыток дозвона (команда 6)

    "APN": "internet",        // Описание APN-точки доступа в интернет
    "APN_USER": "internet",
    "APN_PASS": "",
},

"PhoneBook": // Телефонная книга для дозвона или отправки SMS
[
    "+71112223344",
    "81234445566",
    "+71234445566",
],

"Strings": // Строки для отправки в SMS-сообщениях
[
    "Нет тяги", // Обычная строка с текстом
    "Авария %d для котла №%d", // Строка с динамически подставляемыми данными
                                // (вместо "%d")

    "Строка 4",
    "Строка 5",
],
```

7.2. Команды

0x0100	1	Баланс карты	R	Struct.
			W	4

Команда запрашивает баланс карты, USSD-команда «#100#». Ответ содержит сумму остатка средств на счете в копейках. USSD-команду можно изменить в разделе “USSD_GetBalance” файла simcard.txt

-  См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,
 Регистр 0x0101: Статус выполнения,
 Регистр 0x0102 - 0x0103: (Int32) Баланс карты в копейках.

0x0100	4	Отправить SMS-сообщение (короткая команда)	R	Struct.
			W	4-14

Команда заполняет буфер для SMS-сообщений текстом одной из строк заданных в файле simcard.txt в разделе “**Strings**”. А затем отправляет содержимое буфера на номер из телефонной книги в разделе “**PhoneBook**”. Индексы строк начинаются с нуля. Если индекс строки с текстом указан неверно, сообщение будет иметь вид «Нет строки с индексом N», где N-заданный индекс. Если индекс строки с номером телефона указан неверно, то команда вернет ошибку выполнения.

-  Если индекс строки в телефонной книге задан равным (-1), то сообщения будут переданы на все номера телефонной книги. Если индекс равен (-2), то сообщения будут переданы на все номера телефона до первой успешной передачи сообщения.

Команда поддерживает строки с аргументами, когда необходимо в текст добавить изменяющуюся числовую информацию. Например в строке «Температура воды 35°C» можно менять значение показаний, при этом переменные будут вставлены в текст автоматически. Серым цветом выделены регистры предназначенные для передачи аргументов строки, передавать нужно только те, которые будут использованы, либо не передавать вовсе, если строка не содержит аргументов. Подробную информацию смотри в описании этого раздела.

-  См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.

Регистр 0x0100: Номер команды,
 Регистр 0x0101: Статус выполнения,

Регистр 0x0102: Индекс строки в телефонной книге. Раздел “**PhoneBook**” файл simcard.txt

Регистр 0x0103: Индекс строки с текстом. Раздел “**Strings**” файл simcard.txt

Регистр 0x0104 - 0x0105: (32 бита) аргумент 0,

Регистр 0x0106 - 0x0107: (32 бита) аргумент 1,

....

Регистр 0x0116 - 0x0117: (32 бита) аргумент 9,

0x0100 5

Отправить SMS-сообщение с текстом из буфера на указанный номер телефона.
Каждый регистр содержит 1 символ

R
W

Struct.

4 - 25

Команда отправляет SMS-сообщение с текстом из буфера на указанный номер телефона. Формат номера может быть как “+7xxxxxxxxxx”, так и “8xxxxxxxxxx”. Строка с номером передается в виде ASCII символов, каждый регистр содержит по 1 букве.

Например если номер имеет вид +79281112233, то в ASCII символах он примет вид 0x2B 0x37 0x39 0x32 0x38 0x31 0x31 0x31 0x32 0x32 0x33 0x33. Тогда регистр 0x0102, будет содержать значение 0x002B, регистр 0x0103 = 0x0037, .. регистр 0x010D = 0x0033, регистр 0x010E = 0x0000. Завершающим символом строки служит число 0.

- ☺ *Чтобы использовать номер из телефонной книги (файл simcard.txt, раздел “**PhoneBook**”), необходимо в **регистр 0x0102** записать значение 0x1B, а в **регистр 0x0103** записать индекс строки (начиная с 0). Если индекс указан неверно, команда вернет ошибку выполнения.*
- ☺ *Если индекс строки в телефонной книге задать равным (-1), то сообщения будут переданы на все номера телефонной книги. Если индекс равен (-2), то сообщения будут переданы на все номера телефона до первой успешной передачи сообщения.*

Команда поддерживает строки с аргументами, когда необходимо в текст добавить изменяющуюся числовую информацию. Например в строке «Температура воды 35°С» можно менять значение показаний, при этом переменные будут вставлены в текст автоматически. Серым цветом выделены регистры предназначенные для передачи аргументов строки, передавать нужно только те, которые будут использованы, либо не передавать вовсе, если строка не содержит аргументов. Подробную информацию смотри в описании этого раздела.

- ☺ *См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.*

Регистр 0x0100: Номер команды,

Регистр 0x0101: Статус выполнения,

Регистр 0x0102: Первый ASCII-символ номера телефона,

Регистр 0x0103: Второй ASCII-символ номера телефона,

.....

Регистр 0x010E: Последний ASCII-символ номера телефона.

Регистр 0x010F - 0x0110: (32 бита) аргумент 0,

Регистр 0x0111 - 0x0112: (32 бита) аргумент 1,

....

Регистр 0x0120 - 0x0121: (32 бита) аргумент 9,

0x0100 6

**Звонок на указанный номер телефона.
Каждый регистр содержит 1 символ**

R	Struct.
W	4 - 15

Команда совершает звонок на указанный номер телефона. Время вызова равно 30 секунд, если абонент не поднимает трубку или сбрасывает входящий звонок, то Терминал автоматически повторит его заданное количество раз. Количество попыток дозвона находится в файле simcard.txt, параметр "Call_RetryCount" = n (3). Если все попытки оказались неудачными, то в статусе выполнения команды (регистр 0x0101) будет ошибка. Если абонент ответил на вызов, то будет поднят соответствующий флаг во флагах состояния Терминала (по адресу 0x0200), а команда завершит свое выполнение со статусом «Команда выполнена успешно».

Формат номера может быть как "+7xxxxxxxxx", так и "8xxxxxxxxx". Строка с номером передается в виде ASCII символов, каждый регистр содержит по 1 букве.

Например если номер имеет вид +79281112233, то в ASCII символах он примет вид 0x2B 0x37 0x39 0x32 0x38 0x31 0x31 0x31 0x32 0x32 0x33 0x33. Тогда регистр 0x0102, будет содержать значение 0x002B, регистр 0x0103 = 0x0037, .. регистр 0x010D = 0x0033, регистр 0x010E = 0x0000. Завершающим символом строки служит число 0.

- ☛ *Чтобы использовать номер из телефонной книги (файл simcard.txt, раздел "PhoneBook"), необходимо в **регистр 0x0102** записать значение 0x1B, а в **регистр 0x0103** записать индекс строки (начиная с 0). Если индекс указан неверно, команда вернет ошибку выполнения.*
- ☛ *Если индекс строки в телефонной книге задан равным (-1), то дозвон будет произведен поочередно на все номера телефонной книги. Если индекс равен (-2), то дозвон будет произведен поочередно на все номера до первой поднятой трубки.*
- ☛ *См. описание в разделе 3. «Описание общих регистров». Команда должна быть записана в регистр 0x0100. Запуск команды на выполнение производится записью числа «1» в регистр 0x0101.*

Регистр 0x0100: Номер команды,

Регистр 0x0101: Статус выполнения,

Регистр 0x0102: ASCII символ первого знака,

Регистр 0x0103: ASCII символ второго знака,

.....

Регистр 0x010E: см. описание регистра 0x0102.

0x0200	Флаги состояния Терминала	R	Uint16
			1

Флаги текущего состояния блока для работы с Информатором. Состояние сети обновляется каждые 30 секунд.

Регистр 0x0200: Флаги состояния.

Бит 15: Регистрация в домашней сети (есть возможность звонить/отправлять SMS),

Бит 14: Регистрация в сети, роуминг,

Бит 13: Сеть предоставляет возможность доступа в Интернет,

Биты 12-3: Зарезервированы,

Бит 2: Статус последней команды на отправку SMS-сообщения. Если сообщение отправлено успешно-1, иначе-0,

Бит 1: Статус последней команды голосового вызова на номер абонента. Если вызов прошел и абонент ответил-1, иначе-0,

Бит 0: Наличие подключения к Интернет.

0x0300	Буфер для SMS-сообщений. Доступ по 8 бит. В каждом регистре 1 буква	R W	Uint8
			1 - 512

При обращении к буферу по адресам 0x0300 - 0x04FF, в каждый регистр будет укладываться по 1 элементу буфера (по 1 букве). Доступ может быть получен к одному из регистров или диапазону регистров.

Буфер для текста SMS-сообщений представляет из себя массив из 512 элементов по 8 бит в каждом. Символ из таблицы ASCII всегда имеет длину 8 бит, таким образом в буфере может храниться до 512 символов текста. Завершающим значением в строке всегда должен быть символ со значением 0x00 (кроме случая, когда строка занимает весь буфер, тогда завершающий символ будет подставлен внутри терминала автоматически).

Поддерживаются строки с аргументами. Подробную информацию смотри в описании этого раздела.

Регистр 0x0300: ASCII символ первой буквы,

Регистр 0x0301: ASCII символ второй буквы,

.....

Регистр 0x04FF: см. описание регистра 0x0300.

0x0500	Буфер для SMS-сообщений. Доступ по 16 бит. В каждом регистре по 2 буквы	R	Uint8
		W	1 - 256

При обращении к буферу по адресам 0x1000 - 0x10FF, в каждый регистр будет укладываться по 2 элемента буфера (по 2 буквы). Доступ может быть к одному из регистров или диапазону регистров.

Буфер для текста SMS-сообщений представляет из себя массив из 512 элементов по 8 бит в каждом. Символ из таблицы ASCII всегда имеет длину 8 бит, таким образом в буфере может храниться до 512 символов текста. Завершающим значением в строке всегда должен быть символ со значением 0x00 (кроме случая, когда строка занимает весь буфер, тогда завершающий символ будет подставлен внутри терминала автоматически).

Поддерживаются строки с аргументами. Подробную информацию смотри в описании этого раздела.

Структура.

Регистр 0x0500: Младшая часть регистра содержит первый ASCII символ текста, старшая часть регистра содержит второй ASCII символ текста,

Регистр 0x0501: Младшая часть регистра содержит третий ASCII символ текста, старшая часть регистра содержит четвертый ASCII символ текста,

.....

Регистр 0x05FF: см. описание регистра 0x0500.

7.3. Таблица регистров

Адрес	CMD	R W	Кол-во регистров	Описание	Значение	Заводское значение
0x0100	1	R W	4	Баланс карты в копейках	Int32	0
	4	W	4 - 14	Короткая команда. Отправить SMS-сообщение с готовой строкой на номер из телефонной книги.	Структура данных	-
	5	W	4 - 25	Отправить SMS-сообщение на указанный номер телефона. Каждый регистр содержит 1 символ	Структура данных	-
	6	W	4 - 15	Звонок на указанный номер телефона. Каждый регистр содержит 1 символ	Структура данных	-
0x0200		R	1	Флаги состояния Терминала	UInt16	-
0x0300 ... 0x04FF		R W	1-512	Буфер для SMS-сообщений. Доступ по 8 бит	Текст ASCII (UInt8)	0
0x0500 ... 0x05FF		R W	1-256	Буфер для SMS-сообщений. Доступ по 16 бит	Текст ASCII Структура (UInt8UInt8)	0

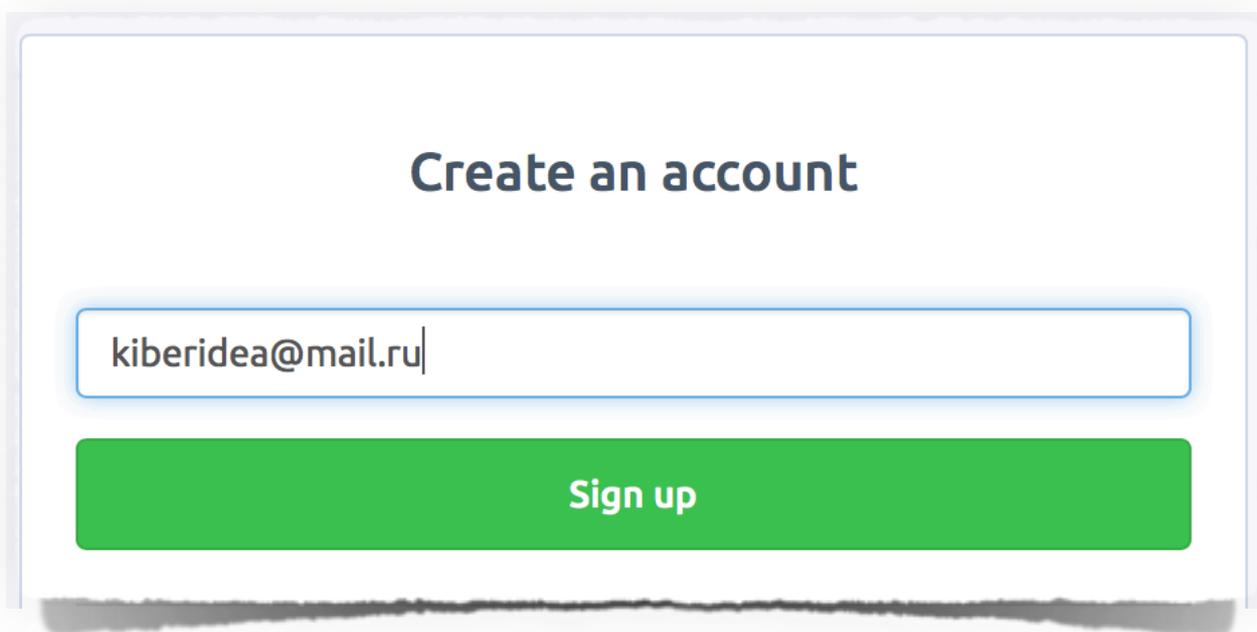
Примеры

1. Телеметрия

В данном примере мы подготовим все, что необходимо для начала работы с телеметрией.

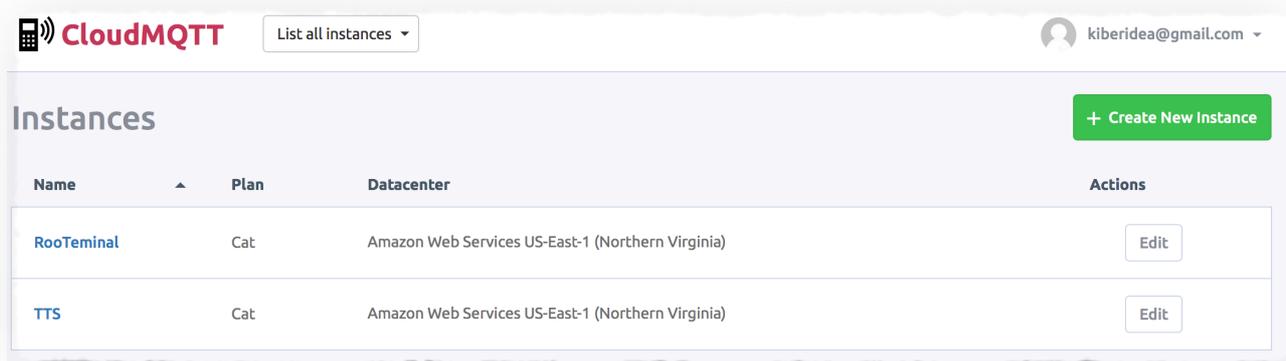
1.1. Создание и настройка аккаунта на сервере брокера

Создадим аккаунт с бесплатным тарифом на сайте cloudmqtt.com. Для этого на главной странице вводим адрес электронной почты. После нажатия на кнопку «Sign up»,



по указанному адресу будет выслано письмо с ссылкой для подтверждения регистрации. Переходим по ссылке, после чего нужно придумать пароль для доступа к новому аккаунту. Этот пароль нужен только для конфигурирования и не имеет ничего общего с паролем для телеметрии. Владелец аккаунта, как провайдер, может контролировать все свои облака и решать кому предоставить доступ к сети.

Нажатием на кнопку «Create New Instance» создаем новое облако, к которым будут подключаться устройства и клиенты.



В поле Name задаем название облака, в поле Plan выбираем бесплатный тарифный план Cute Cat. Поля Data center и Tags не трогаем, и снова нажимаем кнопку «Create New Instance». В списке появится только что созданное облако.

- 🔒 Владелец аккаунта может создавать неограниченное количество облаков, с различными тарифами для разных целей.
- 🔒 Каждое облако представляет собой закрытую сеть для совместной работы группы устройств. В общем случае отдельные облака не имеют доступа к информации друг друга. Но при необходимости можно настроить так называемый «мост» между облаками, рассмотрение которого выходит за рамки примера.

Далее нажимаем на имя облака и в открывшемся окне выбираем первую вкладку «Details»

Для подключения устройств и клиентов к облаку нам будут нужны поля Server и Port.

Details Statistics

Server	m13.cloudmqtt.com
User	asdalfjj Restart
Password	uwflj2PTKHKR Rotate
Port	19371
SSL Port	29371
Websockets Port (TLS only)	39371
Connection limit	10

Active Plan



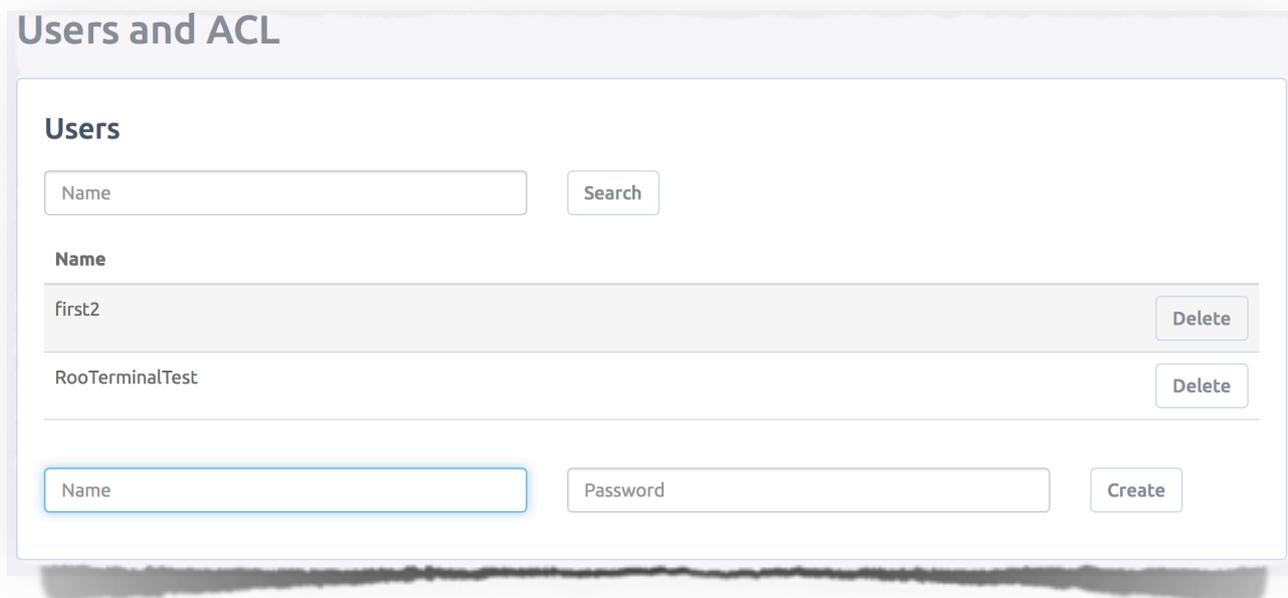
Cute Cat

Upgrade Instance

Reset DB

This will erase all stored messages and sessions. The instance will be restarted. Reset DB

Переходим на вкладку «Users».



В этой вкладке **необходимо создать пользователя** для каждого устройства/клиента, полученные User name и Password будут нужны для подключения к облаку.

И наконец нужно создать топики. Для примера создадим шаблон Terminal/Test/Pub/# , клиенты будут публиковать в них сообщения, а терминалы подписываться на них. И шаблон Terminal/Test/Sub/# , клиенты будут подписываться на сообщения из топиков соответствующих этому шаблону, а терминалы будут публиковать в них сообщения со значениями своих переменных.

Если представить, что сообщения от клиентов это «файлы» с данными, то топики это «папки», в которые записывают данные при публикации сообщения и читают при подписке. При этом топики могут хранить только по одному сообщению и каждое новое сообщение затирает предыдущее. Топики необходимо создать перед использованием за ранее по одному, каждый топик будет привязан к конкретному пользователю, этот вариант дает полный контроль над топиками. Либо можно создать шаблон, описывающий группу топиков, тогда при публикации сообщения топик будет создан автоматически, если он соответствует шаблону.

 *На вкладке «WEB Socket UI» можно производить мониторинг новых публикаций и делать тестовые публикации в ручную.*

Брокер готов к работе, далее необходимо записать в терминалы полученные настройки для телеметрии и сконфигурировать приложение-клиент на сотовом телефоне.

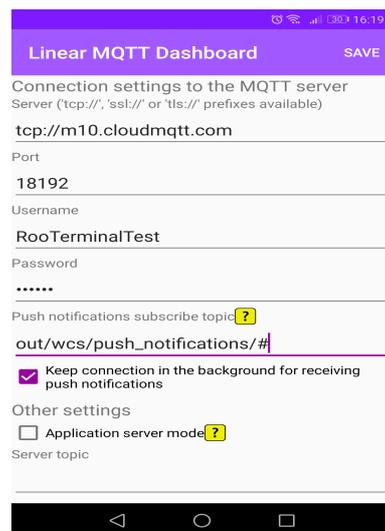
1.2. Настройка MQTT-клиента на сотовом телефоне

В качестве MQTT-клиента доступно огромное количество приложений для Интернета Вещей (Internet of Things), работающих по протоколу MQTT. Для демонстрации возможностей телеметрии установим приложение LinearMqtt, доступное бесплатно через GooglePlay и ApplePlayMarket.

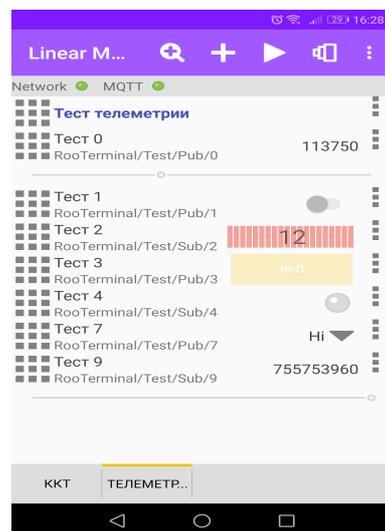
Интерфейс программы достаточно прост и не перегружен. При первом запуске после установки первая страница будет заполнена различными элементами для демонстрации

возможностей приложения. Топики нам не подойдут и их нужно будет исправить или создать новые элементы самостоятельно.

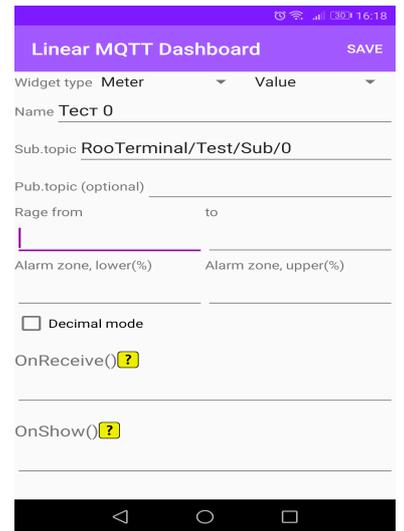
Останавливаем телеметрию. Заходим в настройки «App settings» и заполняем поля Server, Port, User name и Password. Сохраняем настройки.



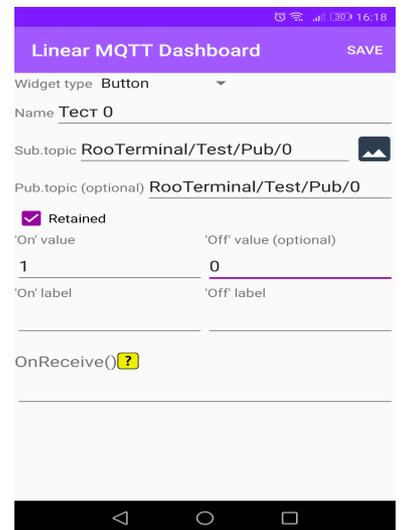
Создаем элементы для работы с переменными телеметрии. Путь к топикам для публикации соответствует созданному ранее шаблону Terminal/Test/Pub/, далее идет номер переменной телеметрии начиная с нуля. Для подписки на топик применяется тот же принцип адресации переменных.



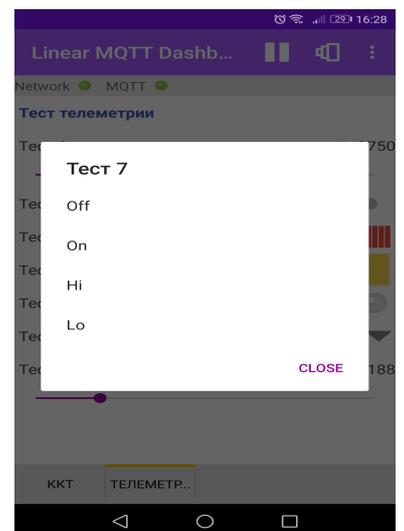
Пример элемента **Meter**, который только отображает прочитанные значения переменных, публикуемые терминалом.



Пример элемента **Кнопка**, которая публикует свое состояние (и подписана на него-же) для передачи в терминал.



Пример элемента **ComboBox**, который позволяет публиковать или читать сразу несколько значений, причем каждому цифровому значению переменной может соответствовать свой текст.



Приложение А. Таблицы кодирования текстовой информации

Для кодирования текстовой информации используются специальные таблицы символов. Основные символы цифр, знаков препинания, и английских букв кодируются стандартной таблицей символов ASCII. Для кодирования русских букв и некоторых дополнительных символов применяется кодировка Windows-1251.

 Некоторые специальные символы могут не отображаться.

Основная таблица ASCII-символов

DEC	DEC	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
0	0x00	[NULL]	32	0x20	[SPACE]	64	0x40	@	96	0x60	`
1	0x01	[START OF HEADING]	33	0x21	!	65	0x41	A	97	0x61	a
2	0x02	[START OF TEXT]	34	0x22	"	66	0x42	B	98	0x62	b
3	0x03	[END OF TEXT]	35	0x23	#	67	0x43	C	99	0x63	c
4	0x04	[END OF TRANS.]	36	0x24	\$	68	0x44	D	100	0x64	d
5	0x05	[ENQUIRY]	37	0x25	%	69	0x45	E	101	0x65	e
6	0x06	[ACKNOWLEDGE]	38	0x26	&	70	0x46	F	102	0x66	f
7	0x07	[BELL]	39	0x27	'	71	0x47	G	103	0x67	g
8	0x08	[BACKSPACE]	40	0x28	(72	0x48	H	104	0x68	h
9	0x09	[HORIZONTAL TAB]	41	0x29)	73	0x49	I	105	0x69	i
10	0x0A	[LINE FEED]	42	0x2A	*	74	0x4A	J	106	0x6A	j
11	0x0B	[VERTICAL TAB]	43	0x2B	+	75	0x4B	K	107	0x6B	k
12	0x0C	[FROM FEED]	44	0x2C	,	76	0x4C	L	108	0x6C	l
13	0x0D	[CARRIAGE RETURN]	45	0x2D	-	77	0x4D	M	109	0x6D	m
14	0x0E	[SHIFT OUT]	46	0x2E	.	78	0x4E	N	110	0x6E	n
15	0x0F	[SHIFT IN]	47	0x2F	/	79	0x4F	O	111	0x6F	o
16	0x10	[DATA LINK ESCAPE]	48	0x30	0	80	0x50	P	112	0x70	p
17	0x11	[DEVICE CONTROL 1]	49	0x31	1	81	0x51	Q	113	0x71	q
18	0x12	[DEVICE CONTROL 2]	50	0x32	2	82	0x52	R	114	0x72	r
19	0x13	[DEVICE CONTROL 3]	51	0x33	3	83	0x53	S	115	0x73	s
20	0x14	[DEVICE CONTROL 4]	52	0x34	4	84	0x54	T	116	0x74	t
21	0x15	[NEGATIVE ACK.]	53	0x35	5	85	0x55	U	117	0x75	u
22	0x16	[SYNCHRONOUS IDLE]	54	0x36	6	86	0x56	V	118	0x76	v
23	0x17	[ENG OF TRANS. BLOCK]	55	0x37	7	87	0x57	W	119	0x77	w
24	0x18	[CANCEL]	56	0x38	8	88	0x58	X	120	0x78	x
25	0x19	[END OF MEDIUM]	57	0x39	9	89	0x59	Y	121	0x79	y
26	0x1A	[SUBSTITUTE]	58	0x3A	:	90	0x5A	Z	122	0x7A	z
27	0x1B	[ESCAPE]	59	0x3B	;	91	0x5B	[123	0x7B	{
28	0x1C	[FILE SEPARATOR]	60	0x3C	<	92	0x5C	\	124	0x7C	
29	0x1D	[GROUP SEPARATOR]	61	0x3D	=	93	0x5D]	125	0x7D	}
30	0x1E	[RECORD SEPARATOR]	62	0x3E	>	94	0x5E	^	126	0x7E	~
31	0x1F	[UNIT SEPARATOR]	63	0x3F	?	95	0x5F	_	127	0x7F	[DEL]

Расширенная таблица ASCII-символов. Кодировка Windows-1251

DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
128	0x80	Ђ	160	0xA0	[пробел]	192	0xC0	А	224	0xE0	а
129	0x81	Ѓ	161	0xA1	Ў	193	0xC1	Б	225	0xE1	б
130	0x82	„	162	0xA2	ў	194	0xC2	В	226	0xE2	в
131	0x83	ѓ	163	0xA3	Ј	195	0xC3	Г	227	0xE3	г
132	0x84	„	164	0xA4	Ѡ	196	0xC4	Д	228	0xE4	д
133	0x85	...	165	0xA5	Ѕ	197	0xC5	Е	229	0xE5	е
134	0x86	†	166	0xA6	Ї	198	0xC6	Ж	230	0xE6	ж
135	0x87	‡	167	0xA7	§	199	0xC7	З	231	0xE7	з
136	0x88	€	168	0xA8	Ё	200	0xC8	И	232	0xE8	и
137	0x89	‰	169	0xA9	©	201	0xC9	Й	233	0xE9	й
138	0x8A	Љ	170	0xAA	€	202	0xCA	К	234	0xEA	к
139	0x8B	<	171	0xAB	<<	203	0xCB	Л	235	0xEB	л
140	0x8C	Њ	172	0xAC	¬	204	0xCC	М	236	0xEC	м
141	0x8D	Ќ	173	0xAD	-	205	0xCD	Н	237	0xED	н
142	0x8E	Ћ	174	0xAE	®	206	0xCE	О	238	0xEE	о
143	0x8F	Ќ	175	0xAF	İ	207	0xCF	П	239	0xEF	п
144	0x90	ђ	176	0xB0	°	208	0xD0	Р	240	0xF0	р
145	0x91	‘	177	0xB1	±	209	0xD1	С	241	0xF1	с
146	0x92	’	178	0xB2	І	210	0xD2	Т	242	0xF2	т
147	0x93	“	179	0xB3	ı	211	0xD3	У	243	0xF3	у
148	0x94	”	180	0xB4	ѓ	212	0xD4	Ф	244	0xF4	ф
149	0x95	•	181	0xB5	µ	213	0xD5	Х	245	0xF5	х
150	0x96	–	182	0xB6	¶	214	0xD6	Ц	246	0xF6	ц
151	0x97	—	183	0xB7	·	215	0xD7	Ч	247	0xF7	ч
152	0x98	[NULL]	184	0xB8	ë	216	0xD8	Ш	248	0xF8	ш
153	0x99	™	185	0xB9	№	217	0xD9	Щ	249	0xF9	щ
154	0x9A	љ	186	0xBA	є	218	0xDA	Ъ	250	0xFA	ъ
155	0x9B	›	187	0xBB	»	219	0xDB	Ы	251	0xFB	ы
156	0x9C	њ	188	0xBC	j	220	0xDC	Ь	252	0xFC	ь
157	0x9D	ќ	189	0xBD	S	221	0xDD	Э	253	0xFD	э
158	0x9E	ћ	190	0xBE	s	222	0xDE	Ю	254	0xFE	ю
159	0x9F	џ	191	0xBF	ï	223	0xDF	Я	255	0xFF	я

Приложение В. Формат JSON

Формат JSON используется для представления объектов в виде текстовой строки. Это один из наиболее удобных форматов хорошо воспринимаемых человеком.

Формат предполагает записи вида:

- { ... } - объект. Все объекты начинаются с фигурной скобки и завершаются фигурной скобкой с запятой,
- [...] - массив, элементами могут быть другие объекты или значения, размерность массивов не задается,
- Значения одного из типов:
 - строки в двойных кавычках,
 - целое число может быть записано в двойных кавычках или без них. Число с плавающей запятой пишется только в двойных кавычках, разделителем дробной части может быть только точка “.”,
 - логическое значение true/false, двойные кавычки не обязательны.

Имя параметра всегда пишется в двойных кавычках, значение параметра может быть записано в двойных кавычках или без них в зависимости от его типа (строки всегда пишутся в двойных кавычках). Имя параметра отделяется от его значения символом двоеточия “:”, пробелы запрещены. Параметры, объекты, массивы и элементы массивов разделяются запятой “,”. В данной реализации комментарии запрещены, весь текст является чувствительным к регистру, например записи “Count”:10 и “count”:10 не равнозначны. Каждое значение объекта может быть получено через обращение к нему по имени. Если “объект_1” вложен в “объект_0”, то значение объекта 1 будет получено через путь: “объект_0.объект_1”. Элементы массивов начинаются с нуля и неразрывно получают индекс +1 к каждому следующему элементу. Если “объект_1” находится в массиве “массив_0” и имеет индекс 1, то значение объекта 1 будет получено через путь: “массив_0[1].объект_1”.

Пример:

```
{
  "Параметр_0": "Значение параметра 0",
  "Параметр_1": "Значение параметра 1",

  "Блок_0":
  {
    "Блок_0": { "Параметр_0": 0, "Параметр_1": 1, },
    "Блок_1": { "Параметр_0": 0, "Параметр_1": 1, },
  },

  "Массив_0":
  [
    "Элемент 0",
    "Элемент 1",
    ...
    "Элемент n",
  ],

  "Блок_1":
  {
    "Массив_0":
    [
      { "Параметр_0": 0, "Параметр_1": "1", "Параметр_2": "777.50", },
      { "Параметр_0": 1, },
    ],
  },
}
```

```
    "Массив_1":  
    [  
        { "Параметр_0": "Строка с текстом", },  
        { "Параметр_0": "Строка с текстом", "Параметр_1": true, },  
    ],  
},
```

Приложение С. Протокол Modbus

Протокол Modbus RTU является частью стандартного протокола Modicon-Modbus, с официальным стандартом можно ознакомиться по адресу:

http://modbus.org/docs/PI_MBUS_300.pdf.

Далее будут описаны основные принципы работы протокола, достаточные для полного использования возможностей Терминала.

Устройства в сети Modbus работают по принципу Мастер -> Подчиненный (один ко многим). Это означает, что только Мастер сети может начать обмен (сделать Запрос), а Подчиненные всегда слушают сеть и выполняют команды или отвечают на запросы от Мастера.

Запрос мастера содержит адрес подчиненного, код функции, указывающий на необходимое действие, данные для отправки и контрольную сумму для проверки сообщения. Ответное сообщение от подчиненного содержит поля подтверждения выполнения команды, данные для возврата и контрольную сумму для проверки сообщения. Если найдена ошибка в принятом сообщении или подчиненный не может выполнить запрошенное действие, то подчиненный возвращает сообщение об ошибке.

Режим RTU

Система кодирования:

- каждый символ состоит из 8-бит, с шестнадцатеричным значением 0x00 - 0xFF,
- значения 16-бит и более передаются сначала старший байт, затем младший (Hi Lo Hi Lo) (кроме контрольной суммы CRC),

Каждый байт содержит:

- 1 стартовый бит,
- 8 информационных бит, младший значащий бит передается первым (LSB first),
- 1 бит проверки четности чет./нечет.; без бита, если проверка не требуется,
- 1 стоповый бит, если есть проверка четности; 2 бита, если нет проверки четности,

Контрольная сумма пакета:

- CRC16, передается сначала младший байт, затем старший (Lo Hi),
- стартовое значение 0xFFFF,
- полином 0xA001.

В этом режиме каждое сообщение должно начинаться с тишины интервалом не менее 3,5 символов (T1-T2-T3-T4, как показано ниже), время передачи одного символа T равно скорости передачи в бодах/с. При реализации алгоритма удобное всего использовать таймаут на отсутствие принимаемых символов (равный времени передачи 4x символов), запускаемый при приеме первого символа сообщения и сбрасываемый при каждом новом полученном символе. Когда таймаут сработал, считается, что пакет принят полностью и готов к обработке. Первым полем сообщения является адрес устройства.

Все устройства Modbus постоянно слушают сеть, и каждое декодирует сообщение, чтобы понять адресовано оно или нет. *Сообщение должно оканчиваться интервалом*

тишины и если он меньше 3,5 символов, то следующее сообщение будет принято как продолжение текущего, тогда последнее поле с контрольной суммой CRC не совпадет.

НАЧАЛО	АДРЕС	ФУНКЦИЯ	ДАнные	CRC	КОНЕЦ
T1-T2-T3-T4	8 БИТ	8 БИТ	n x 8 БИТ	16 БИТ	T1-T2-T3-T4

Пример кадра с сообщением ModBus RTU.

Поле «Адрес» может содержать значения от 1 до 247 (от 248 до 255 зарезервированы) (от 1 до 32 в прошивках до версии 1.5.0 включительно), это адрес подчиненного устройства, для которого предназначено сообщение. Адрес 0 используется для широковещательных команд не требующих ответа, сразу для всех устройств (в функционале Терминала не поддерживается)

Поле «Функция» содержит номер функции, которую должен выполнить подчиненный. Например чтение или запись значения регистра по заданному адресу. В зависимости от функции, подчиненный должен дать ответ о успешно выполненной операции или об ошибке.

Например мастер дает команду на чтение регистра:

0000 0011 (0x03)

Если команда выполнена успешно, то ответом будет сообщение с содержимым регистра. Если произошла ошибка, то ответное сообщение модифицируется, установкой старшего бита в байте команды:

1000 0011 (0x83)

К коду команды для сообщения с ошибкой добавляется код ошибки, указывающий на тип возникшей проблемы. При этом мастер сети ModBus должен указать оператору на ошибки связи с устройством и/или попытаться отправить сообщение снова.

Поле «Данные» содержит полезные данные и в зависимости от номера функции может иметь разный формат.

Поле «Сумма CRC» содержит контрольную сумму для проверки целостности пакета.

Расчет контрольной суммы (CRC-16)

Контрольная сумма пакета данных занимает 2 байта и располагается в конце пакета. Порядок байт: младший байт, затем старший (Lo Hi). Ниже представлена функция для расчета CRC, в параметрах передается указатель на буфер с пакетом и его длина.

```
uint16_t GetCRC16(char *buf, int len)
{
    uint16_t crc = 0xFFFF;

    for (int pos = 0; pos < len; pos++) {
        crc ^= (uint16_t)buf[pos];

        for (int i = 8; i != 0; i--) {
```

```
        if ((crc & 0x0001) != 0) {
            crc >>= 1;
            crc ^= 0xA001;
        }
        else
            crc >>= 1;
    }
}

return crc;
}
```

Передача пакета

При формировании пакета на передачу во временный буфер записывается сам пакет + 2 байта резервируются для контрольной суммы. CRC должно быть вычислено без учета зарезервированных байт. Результат записывается в зарезервированные байты и пакет передается в сеть.

Прием пакета

После приема, адрес указанный в пакете сравнивается с адресом устройства, если они совпадают, то пакет предназначен для этого устройства. Далее пакет проверяется на целостность, для этого вызывается функция подсчета CRC но уже для всего пакета, включая байты CRC. Если пакет не поврежден, то функция вернет 0 и устройство начинает выполнение команды.



Если адрес или контрольная сумма не совпали, то устройство ничего не отвечает,

После успешного выполнения команды, устройство формирует ответ соответствующий номеру функции. Мастер принимает ответ и команда считается выполненной.

Приложение D. Распиновка кабеля

Кабель от Терминала к ККТ

Длина кабеля не более 2-х метров. Тип провода - многожильный, желательно экранированный. Проводом соединяющим корпуса разъемов может служить экран кабеля.

Терминал (DB-9 Male)		Направление	ККТ (DB9 Male)	
№	Название		№	Название
2	RX	←	3	TX
3	TX	→	2	RX
5	GND	↔	5	GND
Корпус	Case	↔	Корпус	Case

Соединение устройств проводить только при отключенном питании, горячее подключение НЕ ПОДДЕРЖИВАЕТСЯ!