**Основные понятия:**

ПУ — пункт управления.

КП — контролируемый пункт (RTU, remote terminal unit).

ОИК — оперативно информационный комплекс.

ЦППС — центральная приемо-передающая станция (CTS, central transceiver station).

ТС — телесигнал.

ТИ — телеизмерение.

ASDU — блок данных, обслуживаемый прикладным уровнем.

APDU — протокольный блок данных прикладного уровня.

COT — причина передачи (cause of transmission).

GSM-устройство — это устройство приёмо-передачи, использующее сети операторов мобильной связи.

SMS — служба коротких сообщений.

Для организации взаимодействия ПУ и КП для целей обмена телемеханической информацией не всегда имеется физическая возможность использовать проводные линии связи, например в связи с территориальной удаленностью от некоего географического центра, или построение проводных линий связи становится экономически нецелесообразным.

Так как зоны покрытия базовых станций операторов сотовой связи постоянно расширяются, то в качестве альтернативы проводным линиям связи для обмена телемеханической информации можно использовать мобильную радиосвязь. В частности, если объем передаваемых данных не велик, то удобной оказывается служба коротких сообщений, описанная в стандарте GSM.

Максимальная длина текстового сообщения в стандарте GSM 140 байт (1120 бит). Таким образом при использовании 7-битной кодировки (латинский алфавит, цифры) возможно отправлять сообщения длиной до 160 символов. Для поддержки других национальных алфавитов (в частности русского) используется 2-байтовая кодировка UCS-2. Таким образом, SMS, написанное кириллицей, не может превышать 70 символов. Текстовый формат сообщений предоставляет наглядную форму для телемеханической информации получаемой от КП, которая может содержать, как телесигнализацию, так и телеизмерения. Для рационального использования доступного объема SMS необходимо определить компактный и практичный формат обмена телемеханической информации.

Далее описывается набор программных средств, который позволяет организовать получение телемеханической информации от КП, поверх существующих сетей сотовой связи посредством SMS, и дальнейшую ее передачу в ЦППС средствами протокол МЭК104.

**Список утилит, для конкретной реализации:**

iecinit, smsget, smsdrop\_verset, smsdrop\_granit, quemngr, asdusend, asduconv, sockwrite,

iecproxy, ieclink, ginterrog, rtupoll, utils

**Список конфигурационных файлов:**

iecenv — набор переменных окружения.

rtu\_list.conf — список КП, где каждая запись имеет вид   
«rtu address» «phone number» «device model».

cts\_list.conf — список ЦППС, где каждая запись имеет вид   
«cts name» «ip-address» «lan port».

iecd\_crontab — набор заданий с расписанием, для опроса определенных КП.

**iecinit**

Iecinit — основной скрипт запуска. Осуществляет следующие действия:

1. Экспортирует из файла iecenv переменные окружения, для дочерних процессов.

2. Для каждой записи из конфигурационного файла ctu\_list.conf запускает отдельный экземпляр программ quemngr и iecproxy в фоне, которые работает с конкретной ЦППС.

3. Инициализирует GSM-устройство и запускает фоновый процесс smsget, который периодически опрашивает данное GSM-устройство на наличие SMS.

4. С помощью демона crond, опрашивает по расписанию определенные КП из файла rtu\_list.conf. Для этого используется программа rtupoll, которая совместно с smsget синхронизирует доступ к GSM-устройству через файл блокировки device.lock и утилиту flock (1).

**smsget**

Smsget — программа, которая получает SMS от GSM-устройства, с помощью определенных AT-команд. GSM-устройство в системе GNU/linux обычно представлено символьным устройством в каталоге /dev, например /dev/ttyUSB0 или /dev/ttyS0.

Smsget используя конфигурационныe файлы rtu\_list.conf и cts\_list.conf, периодически опрашивает GSM-устройства и для каждого считанного сообщения осуществляет следующую последовательность действий:

1. Сравнивает телефонный номер от которого пришло сообщение, номерами из rtu\_list.conf, если номер отсутствует то сообщение удаляется.

2. Если номер оказался валидным, то в зависимости от модели устройства, запускает дочерний процесс smsdrop\_\*, где в качестве аргумента указывает «№ КП», а на стандартный ввод ему передает полученное SMS.

3. Ожидает окончания работы smsdrop\_\* и в зависимости от кода завершения, либо удаляет сообщение из памяти GSM-устройства (код возврата 0 или 2), либо считывает его повторно (код возврата 1).

4. Если smsdrop\_\* завершился успешно, то записанный им временный файл с набором значений телемеханики var/smsdrop/msgsync, smsget копирует в каталог work/drop каждого экземпляра quemngr который был запущен ранее, и переименовывает его в значение inode файла. После этого уведомляет все процессы quemngr о наличии новых сообщений в каталоге work/drop активацией внешнего триггера.

**smsdrop\_\***

Smsdrop\_\* принимает на стандартном вводе SMS, проверяет его на соответствие формату, конвертирует полученное SMS в набор значений для помещения в ASDU и синхронно записывает его на диск во временный файл var/smsdrop/msgsync.

Каждое значение описывается следующей совокупностью полей:

1. Общий адрес ASDU (rtu address).

2. Общий адрес объекта информации.

3. Идентификатор типа.

4. Значение объекта информации.

5. Описатель качества (если таковой используется).

6. Метка времени.

Пример

130 6 30 0 not\_used 27600:40:14:29:4:3:18

130 1 36 32 not\_used 27600:40:14:29:4:3:18

…

В поле «Описатель качества» вместо not\_used можно задействовать следующий набор набор битов OV — переполнение (Overflow), SB — подстановки вручную (Substituted), NT — не актуальное значение (Not topical).

Также smsdrop\_\* сохраняет полученный набор значений в виде отдельного файла в базу архивных сообщений на диске, для возможности отслеживания хронологии событий, если это потребуется. Помимо этого каждое полученное значение, но уже без меток времени, сохраняется на диске в виде отдельного файла в базе активных значений для возможной процедуры общего опроса.

В зависимости от результата обработки, smsdrop\_\* будет иметь следующий код завершения: 0 — обработка прошла успешно, 1 — синхронная запись сообщения на диск не удалась, 2 — неверный формат сообщения.

**Quemngr**

Quemngr — менеджер очередей, управляет очередью отправки полученных сообщений в ЦППС, а также сканирует каталог входящих сообщений при активации триггера ожидания (файл fifo).

Структура рабочего каталога:

work/drop — очередь входящих сообщений. Сканированные сообщения из входящей очереди попадают во внутреннюю, только если их имя совпадает с собственным inode.

work/in — внутренняя очередь сообщений.

work/act — рабочая очередь на отправку. В данной очереди находятся сообщения, отправка которых осуществляется в данный момент.

work/df — очередь отложенных сообщений. Сюда помещаются сообщения, отправка которых не удалась, но через определенный тайм аут возможно повторная. Очереди с определенным тайм аутом представляют собой именованный определенным образом каталог, например 0h0m50s, 0h20m0s, 1h30m25s.

work/fail — каталог сообщений, отправка которых не удалась после определенного числа попыток.

При запуске quemngr, сканирует каталоги очередей и восстанавливает связанные с ними структуры данных, при наличии канала до ЦППС начинает немедленную отправку сообщений.

Все новые сообщения из входящей очереди помещаются во внутреннюю, которая уже повторно не сканируется. Quemngr поддерживает структуры данных, при перемещении сообщений из одной очереди в другую. Сообщения передаются по возможности одновременно, но с ограничением на количество (конкуренцию) процессов передачи. Таким образом можно управлять нагрузкой на ЦППС и предотвратить шквал передачи в случае приёма большого количества SMS. Сам quemngr передаёт данные опосредованно. Его основная задача: управление очередью и выдача команд на передачу сообщений.

Во время инициализации quemngr запускает сопроцесс asdusend, с которым он связан неименованными каналами. Если во внутренней очереди имеются сообщения, quemngr переносит их в рабочую очередь и оповещает asdusend, передавая ему на стандартный ввод идентификатор сообщения. На стандартный вывод asdusend асинхронно записывает номер сообщения и статус. Успешно переданные сообщения quemngr (статус 0) удалит из рабочей очереди, а с теми сообщениями передача, которых не удалась (статус 2 и более), поступит в соответствии с политикой повторной передачи: либо перенесет в отложенную очередь с отправкой через определенный тайм аут, либо сбросит в каталог work/fail для последующего анализа и обработки вручную.

**asdusend**

Asdusend — сопроцесс quemngr, который на каждое передаваемое сообщение из рабочей очереди инициирует процедуру его передачи в ЦППС. Получив на стандартном вводе идентификатор сообщения, asdusend делает следующее:

1. Записывает на стандартный вывод номер сообщения со статусом 1, уведомляя quemngr о возможности передачи следующего сообщения из внутренней очереди.

2. Запускает два дочерних процесса asduconv и sockwrite, которые связаны между собой неименованным каналом, и осуществляющие передачу сообщения в iecproxy.

Статус завершения sockwrite вместе с индетификатором сообщения записывается на стандартный вывод.

**Asduconv**

Asduconv — программа для преобразовния телемеханической информации в текст ASCII HEX (список строк длиной до 492 символов). Asduconv имеет один ключ -c с помощью, которого задается причина передачи (3 - спорадическая передача, 20 — общий опрос). На стандартный ввод или в виде файла ей передается набор значений, формат которых был описан выше. Данный набор сортируется по общему адресу, затем по типу и адресу объекта. Если адреса объектов информации будут преимущественно по порядку, то при формировании ASDU указывается адрес и значение первого объекта информации, а далее перечисляется только последовательность значений оставшихся объектов информации, в классификаторе переменной структуры бит SQ = 1. Иначе в каждом объекте информации указывается собственный адрес, а бит SQ = 0. После того как список ASDU сформирован, он передается на стандартный вывод, разделителем ASDU служит перенос строки (LF).

Например, на ввод передаем набор значений с причиной передачи 3  
130 1 30 0 not\_used 27600:40:14:29:4:3:18

130 2 30 1 not\_used 27600:40:14:29:4:3:18

130 1 36 32 not\_used 27600:40:14:29:4:3:18  
на выходе имеем две сформированные ASDU в виде ASCII HEX  
1E020300820001000000D06B680E9D031202000001D06B680E9D0312

2401030082000100000000004200D06B680E9D0312.

**Sockwrite**

Sockwrite — простая программа, принимающая на стандартном вводе текст, который записывает в указанный через ключ -s unix сокет. Sockwrite получив от asduconv сформированный список ASDU и отправив его iecproxy, закрывает свой конец unix-сокета через системный вызов shutdown(fd, SHUT\_WR) и ждет окончания передачи. Iecproxy сообщает о статусе передачи в ЦППС короткими служебными сообщениями:

«<» передача прошла успешно (sockwrite завершается с кодом 0).

«-» передача не удалась (sockwrite завершается c кодом 2)

**Iecproxy**

Iecproxy — программа посредник между quemngr и ЦППС, в качестве дочернего процесса запускает ieclink, с которым связана неименованными каналами.

Проксирует запросы на передачу ASDU в ЦППС. Принимает через свой unix-сокет список ASDU в виде ASCII HEX текста, который записывает на стандартный ввод ieclink. Отслеживает статус переданых ASDU.

Уведомляет об отключении или востановлении канала до ЦППС процесс с pid, который она получает через перемунную окружения PID\_NOTIFY (в нашем случае это quemngr). Для этого использует пользовательские скрипты connect\_hook и disconnect\_hook.

Управляет прикладной процедурой общего опроса.

**Ieclink**

Ieclink — программа, реализующая канальный уровень протокола IEC104. В качестве аргументов принимает ip-адрес и порт сервера ЦППС. Устанавливает и поддерживает соединение (c помощью процедуры тестирования) с ЦППС.

Ieclink читает и пишет ASDU на стандартном вводе-выводе в виде текста ASCII HEX (разделителем ASDU служит символ переноса строки). Получив на стандартном вводе список ASDU, который завершается служебным символом «>», ieclink начинает отправку в ЦППС, дожидается передачи всех кадров на канальном уровне с подтверждением и записывает на стандартный вывод статус передачи.

Для уведомления iecproxy о событиях посылает на стандартный вывод короткие служебные сообщения «+» - соединение с ЦППС установлено, «-» - соединение с ЦППС потеряно (передача списка ASDU не удалась), «<» - передача списка ASDU прошла успешно.

**Ginterrog**

Ginterrog — программа, которая запускается iecproxy в качестве дочернего процесса в ответ на активацию прикладной процедуры общего опроса. Ginterrog формирует из базы активных сигналов, расположенных на диске в каталоге db\_active, список ASDU с причиной передачи 20 в виде текста ASCII HEX и записывает его на стандартный вывод.

**Rtupoll**

Rtupoll — скрипт, запускаемый по расписанию, опрашивает определенные КП из конфигурационного файла rtu\_list.conf. Механизм опроса зависит от возможностей устройств КП: это либо звонок с прерыванием через определенный тайм аут, либо посылка SMS определенного формата. Разделяет GSM-устройство установленное в системе вместе smsget, через файл блокировки device.lock.

Основные идентификаторы типов передаваемых сообщений, используемые в текущей реализации M\_SP\_NA\_1 (1), M\_ME\_NC\_1 (13), M\_SP\_TB\_1 (30), M\_ME\_TF\_1 (36).

**Основной сценарий работы для прикладной процедуры «Общий опрос»:**

1. ЦППС посылает команду общего опроса C\_IC\_NA\_1 с COT = 6 «активация». После подключения ЦППС периодически, через определенный тайм аут выполняет процедуру общего опроса или же по запросу оператора.

2. Iecproxy подтверждает прием C\_IC\_NA\_1 на прикладном уровне, дублируя его в направлении ЦППС с COT = 7 «подтверждение активации».

3. Если это первая процедура общего опроса после подключения, то iecproxy дожидается от quemngr отложенных сообщений, которые могли прийти во время отсутствия канала с ЦППС или же служебного сообщения об их отсутствии. Даные ASDU передаются в ЦППС с COT = 3 «спорадически». Статус передачи каждого отложенного сообщения передается quemngr.

4. Далее iecproxy запускает программу ginterrog, которая формирует список ASDU из базы текущих значений с COT = 20 «общий опрос», и начинает его передачу в ЦППС.

5. Закончив передачу последнего ASDU, iecproxy посылает команду C\_IC\_NA\_1 с COT = 10 «завершение активации».

6. Всю работу с канальным уровнем обеспечивает программа ieclink.

**Основной сценарий работы при установленном соединении:**

1. Все вновь пришедшие сообщения, находящиеся в рабочей очереди quemngr передаются в ЦППС с COT = 3 «спорадически».

2. В зависимости от статуса передачи они либо удаляются из рабочей очереди, либо переносятся в отложенную очередь с повторной отправкой через определенный тайм аут.