Всеволод Нестеров (КОМПЭЛ)

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ C-GPS OPEN AT-ПРИЛОЖЕНИЯ

*Итак, бортовой мобильный **навигационный контроллер** на базе беспроводного **процессора Wavecom** и **C-GPS-чипсета eRide** собран. Время запускать программные приложения, разработанные Wavecom на базе открытого программного комплекса **Open AT.***

В предыдущей статье были рассмотрены некоторые аппаратные аспекты интеграции C-GPS-плагина компании Wavecom в бортовое навигационное оборудование систем слежения за подвижными объектами. Теперь мы поговорим о программных средствах.

Как уже говорилось, компания Wavecom предоставляет отладочные комплекты для реализации C-GPS. Они состоят из дочерней платы C-GPS и набора разработчика для беспроводных процессоров Q2686/87 или WMP50/100/150.

Помимо аппаратных средств, для начала разработки приложения разработчику потребуется программный комплекс Open AT нужной версии, который бесплатно предоставляется компанией Wavecom. Его можно скачать с FTP-сайта компании КОМПЭЛ. Чтобы получить логин и пароль для скачивания, достаточно прислать запрос. Каждой версии внутренней операционной системы беспроводного процессора соответствует своя версия Open AT, но поддерживается и обратная совместимость со старыми версиями. Другими словами, приложения, скомпилированные в ранних версиях, будут прекрасно работать на новых операционных системах беспроводных процессоров, работающих с новыми Open AT. Приложения в Open AT пишутся на языке C. В состав Open AT входят: GCC-компилятор, среда разработки Eclipse, Java-машина для работы с Eclipse; набор Development Toolkit, в состав которого входят программы Terminal Emulator, Serial Link Manager и Target Monitoring Tool; документация; бинарные файлы с операционной системой, соответствующей этой Open AT; примеры программ, библиотеки с плагинами.

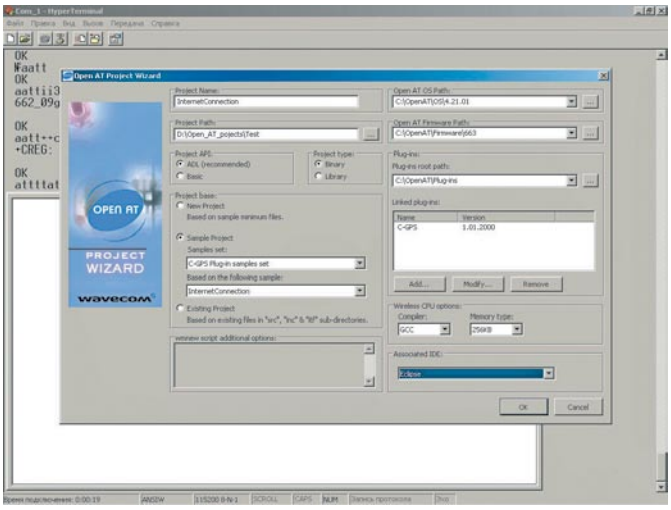

Рис. 1. **Программа Project Wizard**

Во время установки Open AT необходимо убедиться, что выставлены галочки, соответствующие нужным плагинам (в нашем случае C-GPS).

После установки Open AT можно запустить программу Project Wizard, которая формирует проект: новый или на базе имеющихся примеров. Для загрузки примера C-GPS надо в правом окне подключить библиотеку C-GPS, а в левом – сам пример (рис. 1).

Через некоторое время после того, как вы выберете в поле Associated IDE Eclipse и нажмете OK, загрузится среда Eclipse с вашим проектом (рис. 2), после чего с ним можно начинать работать.

Компания Wavecom предоставляет **четыре приложения** для работы с C-GPS: QueryApplication, IntervalUpdate, TCPInternetConnection и RTCTimeAiding.

Приложение **QueryApplication** использует дочернюю плату C-GPS и принимает GPS-данные от нее. Пользователь может послать конфигурационное SMS-сообщение на беспроводной процессор. Приложение определит координаты, время и скорость и вышлет эти данные пользователю через SMS-канал. У пользователя есть возможность установить пароль для получения этих данных. Также можно задать номер телефона для их отсылки. Приложение может выдавать координаты в формате протокола NMEA на порт UART.

Приложение **TCPInternetConnection** также использует дочернюю плату C-GPS и принимает GPS-данные от нее. Пользователь может послать конфигурационное SMS-сообщение на беспроводной процессор. Приложение может быть сконфигурировано таким образом, что начнет периодически отсылать координаты через установленный промежуток времени, используя TCP IP-сокет. Как и в предыдущем случае, есть возможность задать пароль. Приложение может выдавать координаты в формате протокола NMEA на порт UART.

Функции приложения **IntervalUpdate** те же, что и у приложений QueryApplication и TCPInternetConnection, но передавать координаты оно может двумя способами: или через SMS-канал, или через TCP IP-сокет. Координаты также выдаются в виде NMEA-сообщения на UART.

Приложение **RTCTimeAiding** похоже на QueryApplication, но оно способно получать данные с часов реального времени, которые есть в беспроводном процессоре, и передавать их дочерней плате C-GPS. Это сокращает время TTFF (Time To First Fix) с 40 секунд до 5…10 секунд при использовании внешней выносной GPS-антенны. Необходимым условием работы этого приложения является наличие источника питания для часов реального времени.

Приложение Open AT может быть скомпилировано в двух режимах: в режиме **Target** и в режиме **RTE.**

Режим **Target** – это режим построения конечного бинарного файла. При этом генерируется файл с расширением .dwl, который с помощью стандартной программы Hyperterminal можно «залить» в беспроводной процессор, используя протокол X-Modem. Процедура следующая: если в результате проверки, не запущено ли в данный момент Open AT-приложение командой AT+WOPEN, получен ответ 1, то приложение запущено и его надо остановить командой AT+WOPEN=0, если ответ 0, то приложение не запущено. Команда AT+WDWL начинает процесс «заливки» приложения: беспроводной процессор выдает значки §§§§§§, говорящие о том, что он готов к приему файла; после этого надо выбрать файл и отправить его в процессор, используя протокол X-modem. По окончании загрузки необходимо осуществить сброс командой AT+CFUN=1, а затем запустить приложение AT+WOPEN=1. С этого момента оно будет работать даже после перезапуска питания.

В режиме **RTE** приложение генерирует исполняемую программу, которая может быть запущена на компьютере. Эта программа эмулирует работу ядра беспроводного процессора, управляя процессором с помощью собственного протокола через последовательный порт (рис. 3).

В окне этой программы можно установить ряд параметров запуска, а также выставить галочки уровней трасс. С помощью этих уровней можно отладить программу, вставив в нужные места кода определенный макрос, например, вида TRACE ((1, “Embedded: Hello World”));. Сообщения этих макросов будут видны в окне специальной программы

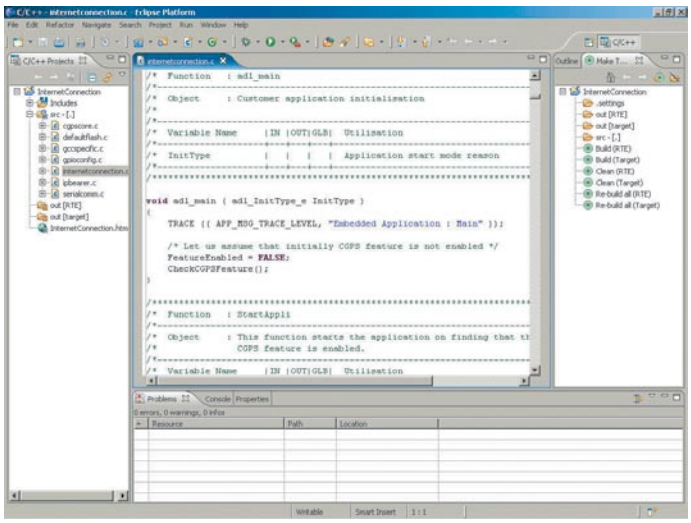

Рис. 2. **Среда разработки Eclipse**

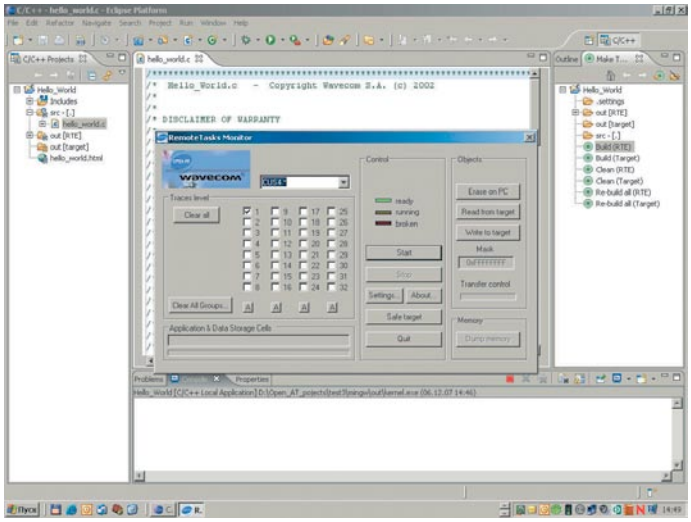

Рис. 3. **Режим работы RTE**

Target Monitoring Tool (TMT), которая идет в комплекте с Open AT. Например, приведенный макрос выводит в окне TMT сообщение “Embedded: Hello World” на первом уровне трасс. Чтобы настроить TMT, необходимо выставить нужные уровни трасс, нажать кнопку Auto Detect, чтобы TMT увидел беспроводной процессор, и инициализировать этот процессор, нажав Commands, Init Target. Вид окна TMT при этом показан на рисунке 4.

Уровни трасс можно увидеть и при работе в режиме Target. В этом режиме беспроводной процессор выдает трассы на последовательный порт, и они могут быть просмотрены программой TMT. В этом случае процессор также надо инициализировать, нажав Commands, Init Target.

Некоторые замечания по работе с приложением C-GPS в режиме RTE:

- режим RTE доступен с некоторыми ограничениями,
- скорость соединения с компьютером должна быть 460800,
- в файл проекта gpioconfig.h нужно добавить код. Если используется процессор Q2686, код будет следующий:

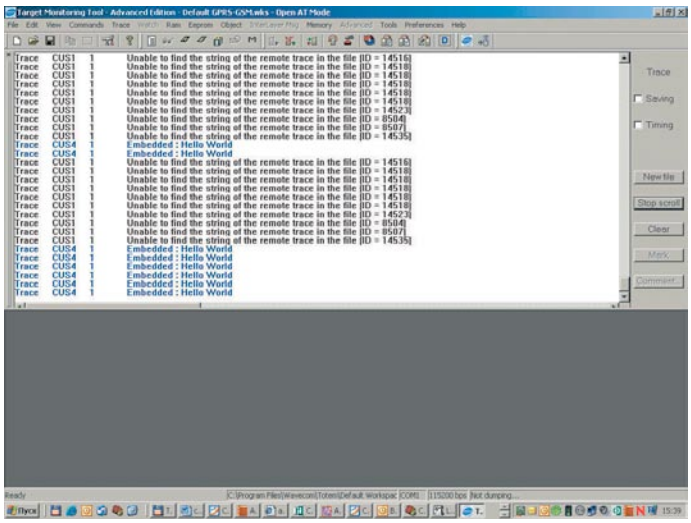

Рис. 4. **Вид окна программы TMT**

```
#if defined (_REMOTETASKS_)
/*Define the product type only for RTE mode*/
/*ADL_IO_PRODUCT_TYPE_Q2686 or ADL_IO_PRODUCT_
TYPE_Q2687*/
#define RTE_PRODUCT_TYPE ADL_IO_PRODUCT_TYPE_
Q2686
#endif
```

Если используется процессор Q2687, то код:

```
#if defined (_REMOTETASKS_)
/* Define the product type only for RTE mode
*/
/*  ADL_IO_PRODUCT_TYPE_Q2686  or  ADL_IO_
PRODUCT_TYPE_Q2687*/
#define RTE_PRODUCT_TYPE ADL_IO_PRODUCT_TYPE_
Q2687
#endif
```

Есть еще одна программа, которую следует рассмотреть – Terminal Emulator (TE). С помощью этой терминальной программы можно во время отладки подавать AT-команды, а также просматривать получаемые и передаваемые данные. Во время запуска программ TMT или TE, автоматически запускается еще одна полезная программа, Serial Link Manager (Selima), которая распределяет поток данных с процессора: трэйсы передаются в TMT, а AT-команды и данные – в TE.

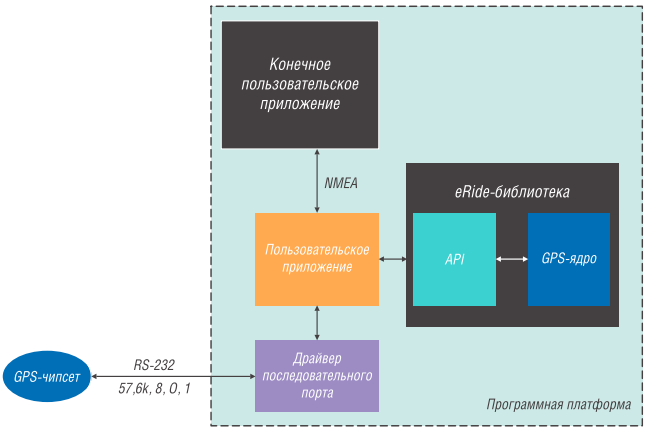

Рис. 5. **Структурная схема программной платформы C-GPS**

Рассмотрим основные моменты программной реализации C-GPS. Структурная схема программной платформы C-GPS-решения представлена на рисунке 5.

Данные GPS в программной платформе C-GPS передаются в виде структур. Все эти структуры описаны в документации:

- *erCallbackStruct* – обратные аппаратные вызовы, которые приложение передает ядру GPS-библиотеки.
- *erNvDataStruct* – RAM-копия навигационных данных. Приложение выделяет под них память.
- *erTcoConstantsStruct, erTcoTableStruct* – используется для передачи GPS-ядру информации о термокомпенсированном генераторе.
- *erTimeInputStruct* – показание часов реального времени, которое передается GPS-ядру.

Информационные GPS структуры:

- *erPvtStruct* – главная структура с навигационными данными.
- *erSvStatusStruct* – содержит информацию о спутниках: отношение сигнал/шум, азимут, возвышение.
- *erFixSetStruct* – спутники, используемые для текущего определения навигационных данных.
- *erTimeStruct* – время, полученное от спутников.
- *erUtcStruct* – спутниковые константы UTC.
- *erGpsMeasStruct* – служебная информация, используемая в тестах при производстве.

В программной платформе C-GPS используются API-функции четырех основных типов:

- Функции управления GPS-ядром, используемые для запуска, работы и останова ядра;
- Функции обратных вызовов, используемые для регистрации аппаратных средств на ядре;
- Функции инициализации GPS-ядра, используемые для конфигурации его работы;
- Функции получения навигационных данных, статусов.

Ядро должно запускаться и выполняться из пользовательского приложения.

Функции управления ядром:

- *void erGpsStart(void)* – запускается после окончания инициализации. Инициализирует ядро.
- *void erGpsStop(void)* – используется для останова ядра и перевода C-GPS-чипсета в состояние OFF.
- *void erGpsStandby(void)* – используется для останова ядра и перевода C-GPS-чипсета в состояние STANDBY.
- *int erGetGpsState(void)* – возвращает GPS-ядро в текущее состояние. Возможны три состояния: ER_GPS_STATE_OFF, ER_GPS_STATE_ON и ER_GPS_STATE_STANDBY.

Главная функция управления ядром

Главной функцией управления ядром является *int erGpsCoreTask(unsigned char *buf, unsigned short length)* – она передает данные с последовательного порта GPS в буфер.

Таблица 1. **Значения, возвращаемые функцией erGpsCoreTask**

Тип	Возвращаемое значение	Функция	Скорость, Гц
ER_PVT_AVAIL	erPvtStruct	erGetPvt	1
ER_TIME_AVAIL	erTimeStruct	erGetTime	1
ER_SVSTAT_AVAIL	erSvStatusStruct	erGetSvStatus	1
ER_FIXSET_AVAIL	erFixSetStruct	erGetFixSet	1
ER_TCO_AVAIL	unsigned int int	erGetTcoFrequency erGetTemperature	1
ER_LCTIME_AVAIL	erTimeStruct	erGetTime	1
ER_NVDATA_AVAIL	erNvDataStruct	App-specific	
ER_TESTDATA_AVAIL	erGpsMeasStruct	erGetTestData	1

Значения, возвращаемые этой функции, представлены в таблице 1.

Состояние ядра в момент выполнения этой функции должно быть ER_GPS_STATE_ON. Возвращаемое значение – бит, который указывает, что новые данные доступны. Данные принимаются со скоростью около 800 байт в секунду. Ядро должно вызываться каждые 100 мс.

Функции обратных вызовов

Функции обратных вызовов обеспечивают интерфейс между «железом» и ядром. Не все из этих функций востребованы. Они должны вызываться перед erGpsStart.

Функция *void erRegisterCallbacks(erCallbackStruct *cb)* регистрирует функции обратных вызовов, необходимые приложению.

Функции инициализации ядра

Эти функции конфигурируют работу GPS-ядра и вызываются перед erGpsStart:

- *void erSetNvData(erNvDataStruct *nvdata)* – запись и сохранение данных в энергонезависимой памяти;
- *void erSetMxMode(int mode)* – разрешение режима отладки;
- *void erSetTcoData (erTcoConstantsStruct *tc, erTcoTableStruct *tt)* – отправка GPS-ядру информации о термокомпенсированном генераторе;
- *void erGpsSetTime(erTimeInputStruct *time)* – получение информации о системном времени;
- *void erGpsSetChipsetMode(int mode)* – обеспечение совместимости с предыдущими дизайнами;
- *erEnablePositionOutageProgatation(int sec)* – установление максимального количество секунд, в течении которых происходит передача навигационных данных после потери сигнала от спутников;
- *erEnableLatencyPositionPropagation(int msecs)* – установление системной задержки.

Функции получения навигационных данных и статусов

Эти функции базируются на возвращаемых параметрах функции erGpsCoreTask (кроме функций const char *erGetVersion(void) – получение версии GPS-ядра и erUtcStruct *erGetUtc(void) – получение спутниковых UTC-параметров).

Параметр ER_PVT_AVAIL:

- *erPvtStruct *erGetPvt(void)* – главный интерфейс GPS-ядра, возвращает указатель на структуру, содержащую навигационную информацию: местоположение, скорость, PDOP;
- *int erPvtIsFix(erPvtStruct *pvt)* – использует структуру PvtStruct для индикации надежности последней полученной навигационной информации;
- *int erPvtIsFixPropagated(erPvtStruct *pvt)* – использует структуру PvtStruct для индикации ненадежности последней полученной навигационной информации.

Параметр ER_TIME_AVAIL:

- *erTimeStruct *erGetTime(void)* – возвращает указатель на структуру, содержащую текущее GPS- и UTC-время;

Параметр ER_SVSTAT_AVAIL:

- *erSvStatusStruct *erGetSvStatus(void)* – возвращает указатель на структуру, содержащую статус найденных спутников, отношение сигнал/шум, возвышение и азимут;

Параметр ER_FIXSET_AVAIL:

- *erFixSetStruct *erGetFixSet(void)* – возвращает указатель на структуру, содержащую список спутников, участвующих в текущем решении;

Параметр ER_TCO_AVAIL:

- *int erGetTemperature(void)* – возвращает текущую температуру в градусах Цельсия;
- *unsigned long erGetTcoFrequency(void)* – возвращает частоту термокомпенсированного генератора в Гц;

Параметр ER_NVDATA_AVAIL:

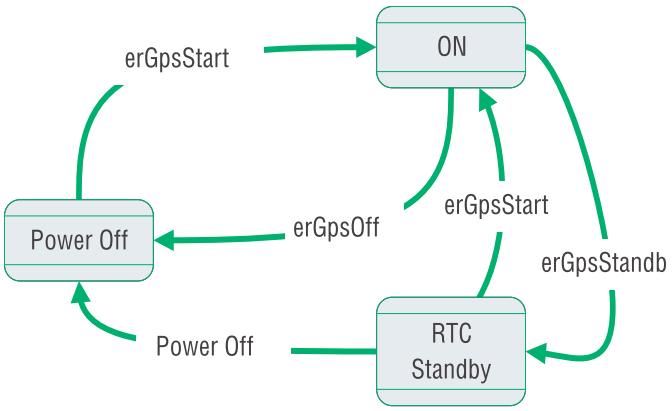

Рис. 6. **Структурная схема управления GPS-ядром**

• Флаг указывает, что данные были изменены ядром;

Параметр ER_TESTDATA_AVAIL:

• *erGpsMeasStruct *erGetTestData(void)* – используется для производственных тестов.

Минимально необходимый набор функций для работы с C-GPS:

• Запуск GPS ядра

Необходимо несколько раз вызвать функцию erGpsCoreTask для обработки входящих данных. Получение навигационных данных производится через функцию erGetPvt. Опционально можно сохранять данные в энергонезависимой памяти.

• Останов системы

erGpsOff, erGpsStandby.

Рассмотренные библиотеки функций eRide в полном объеме позволят добавить GPS-функциональность беспроводному процессору на базе Q2686/87, WMP50/100/150.

Ответственный за направление в КОМПЭЛе – Олег Пушкарев

Получение технической информации, заказ образцов, поставка – e-mail: wireless.vesti@compel.ru

Инновация в сфере сотовой связи M2M

wavecom

SIM-карта (модуль идентификации абонента), разработанная для рынка мобильной связи, не идеальна для применения в M2M-устройствах, где возможны большие вибрации или резкие перепады температур. Также следует учитывать, что SIM-карты, используемые в мобильной связи, изначально рассчитаны на срок службы 18-24 месяца, в то время как карты, устанавливаемые в M2M устройства должны служить гораздо дольше – 5-10 лет.

Для производителей, которые хотели бы подключать свои M2M устройства к сотовым сетям, существует еще ряд трудностей. Такие устройства как, например, терминалы для производства платежей в месте совершения покупки, системы охраны и измерительные приборы, зачастую используются в разных странах и должны быть подключены к разным сетям. Каждому такому устройству требуется своя SIM-карта – различная в зависимости от сети, и обычно подключаемая на месте установки.

Компания **Wavecom** – производитель сотовых модулей создала новый тип SIM-карты, который устанавливается непосредственно в сотовый модуль. Такая карта получила название **inSIM,** она по-прежнему надежно хранит данные мобильного оператора, но жестко встроена в модуль в процессе производства. Это решает проблемы, связанные с кратковременным соединением и действием окружающей среды. Таким образом, решается и вопрос увеличения срока службы карты.

Инновация хорошо сочетается с другим нововведением, представленным провайдером беспроводной M2M связи **Jasper Wireless** (Калифорния), который на данный момент предлагает единую GSM-, SIM-карту для покрытия множества рынков разных стран. Jasper поставляет устройство для установки SIM-карты в оборудование непосредственно на месте производства и для его последующей автоматической активации при запуске, независимо от того, в какой стране производится инсталляция. Компания заключила соглашение с компанией Wavecom о сочетании этих двух передовых M2M разработок в продукт под названием – **Wavecom's inSIM.** Это объединенное решение способно радикально снизить затраты и рационализировать процесс подключения большого количества аппаратуры по всему миру в единую сотовую сеть.

Еще одна, недавно заявленная, разработка компании Wavecom's называется **StarService.** Эта новая бизнес-модель для промышленных M2M-рынков концентрируется на послепродажном обслуживании. Сервис компании Wavecom в области интеллектуальных устройств позволяет пользователю осуществлять надежное удаленное управление беспроводным оборудованием «по воздуху». Целью данной технологии является усовершенствование ПО, конфигурации устройств и их удаленная диагностика – все это ценные элементы удаленного сервиса. Платформа **StarService** создана таким образом, чтобы легко интегрироваться с существующими серверами и функционировать в системе.

(По материалам компании Harbor)