

Новые GSM/ 3G-терминалы Cinterion

с Java на практике

Сегодня GSM-терминалы служат не только для выхода в Интернет и передачи данных, но и для выполнения интеллектуальных функций, таких как: удаленная перезагрузка; управление цифровым входом/выходом; SMS/веб-управление; SMS/веб-информирование; защищенные каналы связи (TSL/SSL). Новые промышленные GSM-терминалы Cinterion EHS5T, EHS6T (3G) и BGS5T (2G) со встроенной современной высокоэффективной и многопоточной Java-платформой позволяют решать самые разные задачи. Простое ли это решение и насколько, рассмотрим на примере.

Наталья Коротких
Александр Сафонов

Более 10 лет назад Siemens представила свой первый GSM-модуль TC45 со встроенной Java-машиной. Позже были выпущены GSM-модули TC65i и терминалы (модемы) TC65T, широко распространенные в России. Продукция отличается высоким качеством, промышленным исполнением и широким набором интерфейсов. С 2008 г. подразделение Siemens WM (Wireless Modules) получило название Cinterion WM. Производственные линии Cinterion находятся в г. Лейпциг (Германия). Недавно компания представила новые терминалы EHS5T, EHS6T и BGS5T в едином факторе со встроенной Java-машиной.

Терминалы EHSxT и BGS5T

BGS5T можно рассматривать как современный аналог GSM-терминала TC65T. Он является более экономичным по питанию; как и TC65T, выполнен с последовательным интерфейсом RS-232 (разъем DB9), который используется как интерфейс данных и команд. На корпусе имеются 8- и 12-контактные разъемы с вы-

водами GPIO, I²C и аналоговым ADC; разъем USB 2.0 HS (как интерфейс данных и команд, но только в режиме Slave, а также для обновления встроенного ПО).

3G-терминалы EHSxT (рис. 1) разработаны на основе модулей Cinterion EHS5/EHS6, которые относятся к серии Cinterion Evolution и характеризуются высокой скоростью передачи данных через 3G-сети сотовых операторов (HSPA) — до 7,2 Мбит/с (UL) и до 5,76 Мбит/с (DL). Они многофункциональны, надежны, работают в широком температурном диапазоне.

Разработанные на инновационных 3G-модулях, терминалы имеют встроенную платформу Java ME 3.2 и позволяют запускать Java-приложения. Для мобильных устройств (MIDP) такие приложения называются «мидлетами», по аналогии с сервлетами и апплетами. Тем самым терминалы можно «запрограммировать» для решения следующих задач:

- передача сигнала тревоги с датчика вскрытия платежного терминала/банкомата, а в загородном «умном» доме — с датчика открытия дверей;
- трансляция видео с камеры наблюдения на сервер;
- удаленное отключение/включение устройства, например обогревателя, в зависимости от температуры в помещении и т. д.

По завершении тестирования и внедрения терминала в систему разработчик может удаленно обновлять установленные программы.

Java ME 3.2

Java — один из самых распространенных языков программирования в мире. Он является кроссплатформным и объектно-ориентированным. Cinterion предоставляет Java-разработчикам инсталляционный пакет необходимого программного обеспечения, библиотек и документации, в том числе основной инструментарий СМТК, интегрируемый в свободно распространяемые



Рис. 1. 3G-терминал EHS5T

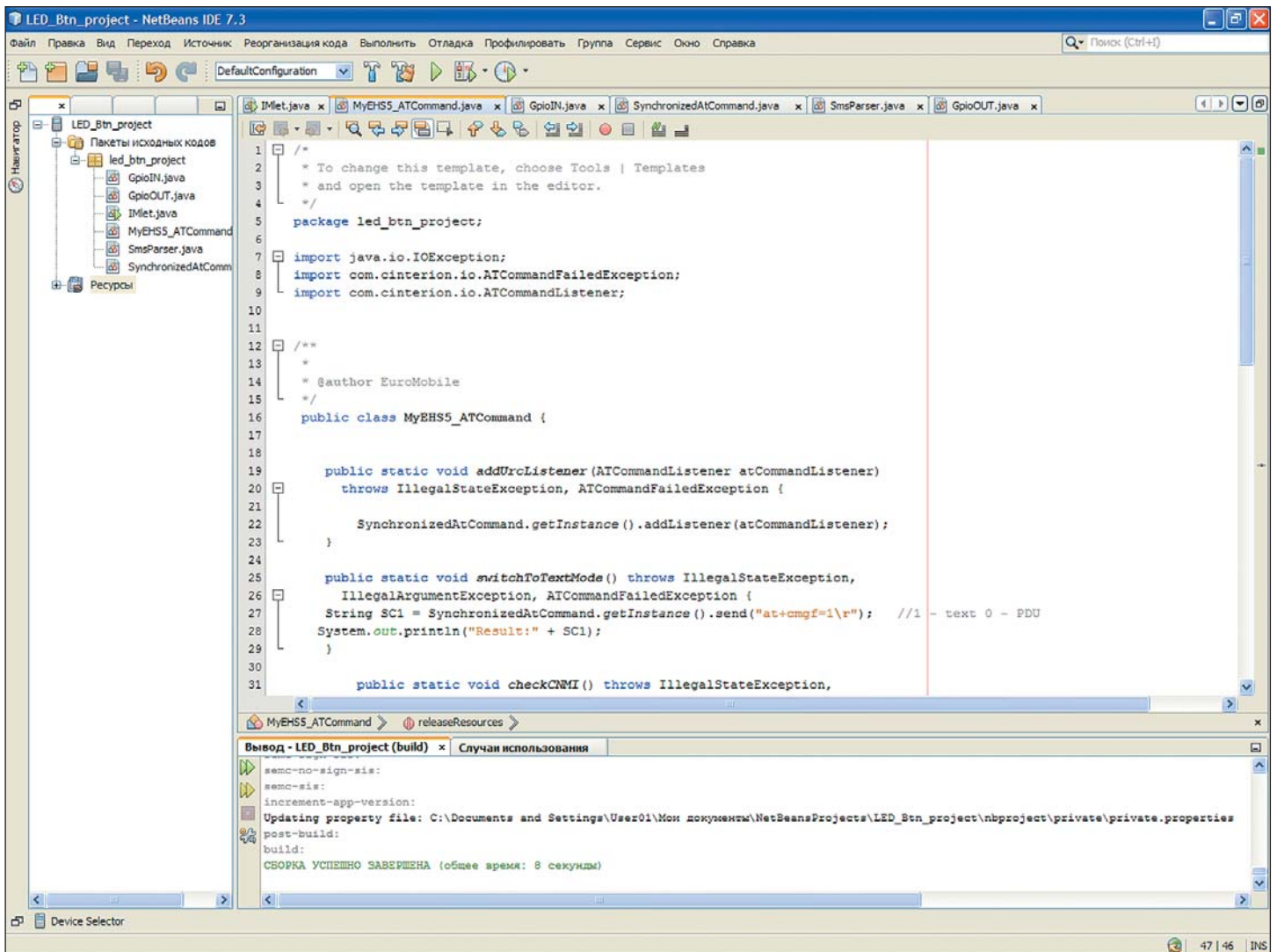


Рис. 2. Проект на языке Java в NetBeans IDE

среды разработки Java IDE NetBeans (Oracle) и Eclipse (IBM). Для разработки собственного Java-мидлета следует установить на ПК среду выполнения Java (JRE, Java Runtime), если она еще не установлена, или обновить имеющуюся а затем инсталляционный пакет Cinterion с поставляемого в комплекте диска. Далее необходимо определиться со средой разработки: NetBeans или Eclipse. После выбора и установки следует последовательно проделать шаги, описанные в [4] для корректного подключения профиля.

Терминал EHSxT может подключаться к компьютеру через USB (кабель USB A — USB B) или последовательный интерфейс и управляться AT-командами из любой терминальной программы. Порт, на который будет транслироваться стандартный вывод Java-мидлета, может быть задан следующей AT-командой: `AT^SCFG="Userware,Stdout","usb3",,,,"on"`, где `usb3` — один из виртуальных COM-портов, появляющихся в результате установки драйвера. При написании программы следует подключить все используемые библиотеки Cinterion: `com.cinterion.ATCommand`, `com.cinterion.io` и т. д. Для этого в ресурсы проекта необходимо добавить библиотеки (архив) Cinterion. Классы Cinterion подробно описаны в прилагаемых файлах

.../CMTK/EHS5/WTK/... В качестве примеров прилагаются готовые проекты, выполненные в Eclipse и NetBeans, демонстрирующие работу с GPIO, AT-командами, SMS-сообщениями, сетевым подключением. Когда программа написана, ее необходимо собрать. Для этого

надо нажать правой кнопкой мыши на проекте в дереве проектов и выбрать пункт «Собрать» или нажать <F6> (рис. 2).

По окончании сборки программы необходимо скопировать `jar` и `jad`-файлы из папки `Project/dist` в память терминала (диск A, рис. 3).

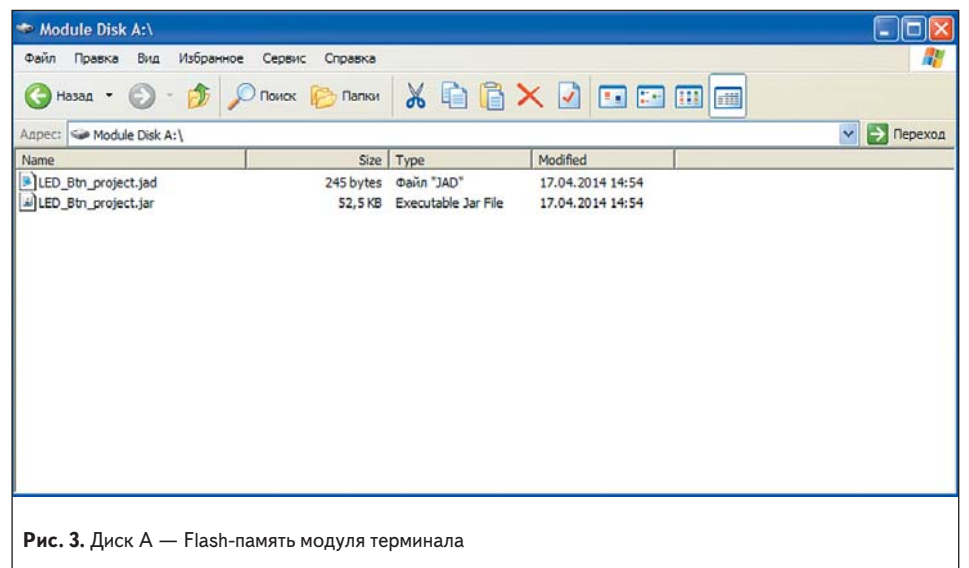


Рис. 3. Диск A — Flash-память модуля терминала

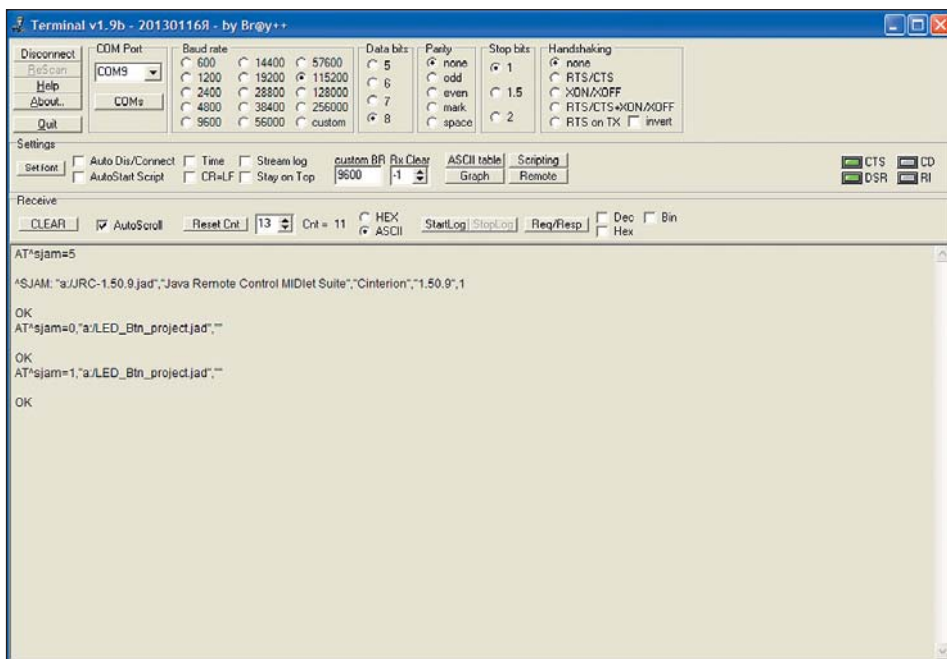


Рис. 4. Установка и запуск мидлета в терминале EHS5T

Для доступа к памяти терминала используется утилита Mobile Exchange Suite (MES), которая поставляется вместе с терминалом. После того как файлы скопированы во Flash-память модуля, можно приступить к запуску мидлета.

Установка мидлета в EHS5T из программы NurerTerminal или аналогичной (рис. 4) осуществляется по команде `AT^SJM=0,\"a:/Midlet.jad\"`. Запуск — `AT^SJM=1,\"a:/Midlet.jad\"` (также можно настроить автозапуск приложения). Остановка — `AT^SJM=2,\"a:/Midlet.jad\"`.

Для настройки автозапуска, то есть автоматического выполнения сразу после включения EHS5T или спустя некоторое время, необходимо подать AT-команду `AT^SCFG: «Userware/Autostart/Delay»,»,»,1»`, где *Delay* — время после включения устройства, но перед запуском мидлета. AT-команда `AT^SJM=5` позволяет узнать,

какие Java-приложения запущены в терминале в данный момент.

Стенд

Для решения простой тестовой задачи управления цифровыми выходами был выбран 3G-терминал EHS5T и собран стенд (рис. 5). Три GPIO терминала EHS5T предназначены для подключения различных цифровых устройств: датчиков, светодиодов, кнопок и др. Чтобы управлять светодиодами с помощью кнопки, следует задать GPIO терминала, к которым подключены светодиоды как выходы, а GPIO кнопки — как вход. Очень важно правильно запитать подключенные устройства, чтобы значение 1 соответствовало 1 (высокому уровню) на выходе, без инверсий. По событию нажатия кнопки включается/выключается синий светодиод и гаснет красный, который изначально при включении загорается. Ниже приведен листинг 1 кода по работе с GPIO.

Листинг 1

```
//класс для работы с GPIO входами
public class GpioIN {
    InPort inport;
    //конструктор класса
    protected GpioIN(String num_GPIO) throws IOException {
        Vector pins, values;
        pins = new Vector(1);
        pins.addElement(num_GPIO);
        inport = new InPort(pins);
    }
    //слушатель события изменения входа GPIO
    public void addListener(InPortListener listener) {
        inport.addListener(listener);
    }
    //освобождение GPIO от работы с Java
    public void release()throws IOException{
        if (inport!= null){
            inport.release();
        }
    }
}
//класс для работы с GPIO-выходами
public class GpioOUT {
    public OutPort outport;
    //конструктор класса
    protected GpioOUT(String num_GPIO) throws
    IOException {
        Vector pins, pinsOut, values;
        values = new Vector();
        pinsOut = new Vector();
        values.addElement(Integer.valueOf(«1»));
        pinsOut.addElement(num_GPIO);
        outport = new OutPort(pinsOut, values);
    }
    //изменение состояния GPIO
    public synchronized void Change(boolean bOnlyOFF){
        try{
            if (bOnlyOFF){
                if (outport.getValue() == 0){
                    outport.setValue(1);
                }
            }
            else if (outport.getValue() == 1){
                outport.setValue(0);
            }
        }
        catch (Exception ex) {
            System.out.println(«Exception» + ex.toString());
        }
    }
}
//освобождение GPIO от работы с Java
public void release()throws IOException{
    if (outport!= null) {
        outport.release();
    }
}
}
```

Запуск основного потока мидлета начинается с выполнения стандартной функции `startApp()` (листинг 2).

Листинг 2

```
//class IMlet extends MIDlet
public void startApp() {
    try{
        //создаем 3 gpio - LED_blue, LED_red, Button
        LED_blue = new GpioOUT(«GPIO7»);
        LED_red = new GpioOUT(«GPIO6»);
        //инициализация кнопки
        Button = new GpioIN(«GPIO8»);
    }
}
```

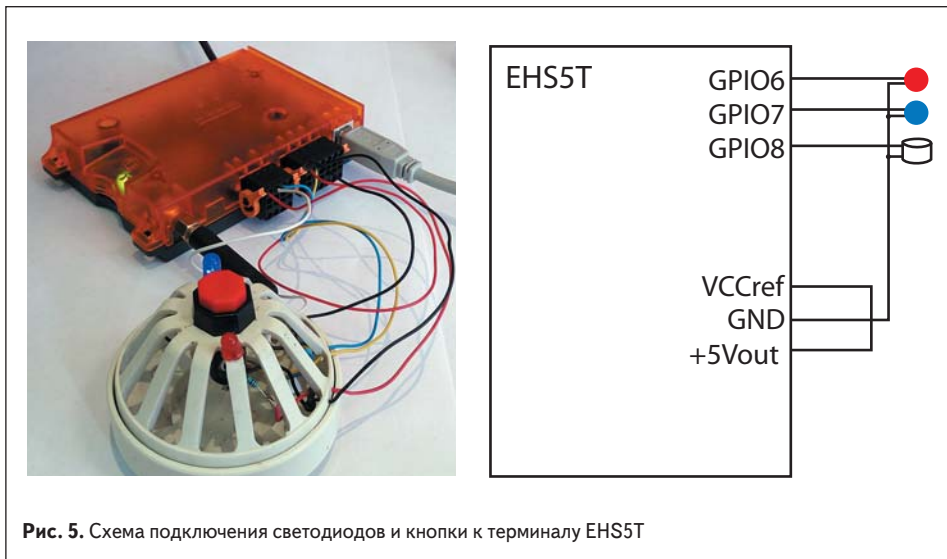


Рис. 5. Схема подключения светодиодов и кнопки к терминалу EHS5T

```

cnt_press = 0;
//мониторинг события нажатия кнопки
Button.addListener(new InPortListener(){
    boolean bfirst = true;
    public void portValueChanged(int portValue){
        try{
            if (portValue == 1){
                if (bfirst){
                    bfirst = false;
                } else{ LED_red.Change(false);}
                cnt_press++;
                if (cnt_press == 2){
                    //гасим синий LED
                    LED_blue.outport.setValue(0);
                    cnt_press = 0;
                }
                else if (cnt_press == 1)
                {
                    //зажигаем синий LED
                    LED_blue.outport.setValue(1);
                }
            }
        }
        catch (Exception ex) {
            System.out.println(«Exception» + ex.toString());
        }
    }
});
//отправка AT-команд
MyEHS5_ATCommand.enableSmsUrucs();
MyEHS5_ATCommand.switchToTextMode();
Thread.sleep(500);
//отслеживаем новое SMS
startSmsMonitor();
}
catch (Exception e) {
    System.out.println(«Exception» + e.toString());
    destroyApp(true);
}
}
//мониторинг нового SMS
public void startSmsMonitor() throws
IllegalStateException, ATCommandFailedException {
    //создаем слушатель

```



Рис. 6. Отладочный стенд 3G-модуля EHS6

```

MyEHS5_ATCommand.addUrcListener(new
ATCommandListener() {
    public void RINGChanged(boolean arg0) {
        // TODO Auto-generated method stub
    }
    public void DSRChanged(boolean arg0) {
        // TODO Auto-generated method stub
    }
    public void DCDCChanged(boolean arg0) {
        // TODO Auto-generated method stub
    }
    public void CONNChanged(boolean arg0) {
        // TODO Auto-generated method stub
    }
    public void ATEvent(String Event) {
        if (Event.indexOf(«+CMTI») > 0) {
            //анализ текста SMS
            SmsParser parser = new SmsParser(Event,
            LED_red);
            parser.start();
        }
    }
});

```

Красный светодиод загорается по приходу SMS со словом «START». Для мониторинга события прихода нового SMS-сообщения в EHS5T включаем индикатор URC: *enableSmsUrucs()*, для этого отправляем следующую AT-команду:

«AT+CNMI=1,1,r». При поступлении нового SMS-сообщения разбираем его (*parser.java*) и, в зависимости от содержимого, гасим или зажигаем красный светодиод. SMS тут же удаляем, чтобы не засорять память.

Данный мидлет будет работать и в отдельно взятом GSM-модуле EHS6, который может быть установлен на специально разработанную отладочную плату (рис. 6). Скачать его (полностью проект NetBeans) можно по ссылке [6].

Заключение

Разработанный мидлет отвечает за управление цифровыми выходами и может использоваться для удаленного открытия и закрытия ворот, дверей, отключения/включения обогревателей и других устройств, что значительно расширяет стандартный функционал 3G-терминалов и выделяет их из оборудования данной ценовой категории. Например, можно использовать данное решение как SMS-выключатель питания в платежном терминале, то есть получить удаленно управляемый Watchdog и 3G-терминал в одном устройстве. При этом базовый функционал терминала позволяет по беспроводным каналам передавать на сервер данные с подключенных устройств и выходить в Интернет. Для получения опытных образцов новых терминалов обращайтесь в компанию «Евромобайл», которая является официальным дистрибьютором Gemalto M2M (Cinterion) в России и странах СНГ. ■

Литература

1. <http://cinterion-m2m.ru/index.html>
2. <http://m2m.gemalto.com/>
3. EHSxT Hardware Description — Gemalto, Германия.
4. Java User Guide — Gemalto, Германия.
5. Cinterion EHS5E AT Command Set v02.000 — Gemalto, Германия.
6. www.euromobile.ru/wp-content/uploads/LED Btn_project.rar