

# Методы и средства разработки мидлетов для GSM-модемов G24-J компании Motorola

**Статья посвящена разработке мидлетов для встраиваемых беспроводных модемов G24-J компании Motorola. Термин «мидлет» означает программу в мобильном устройстве (GSM-модем, сотовый телефон, коммуникатор), написанную на языке Java для виртуальной машины J2ME (Java 2 Micro Edition). Неоспоримым преимуществом использования сотовой связи по сравнению с другими видами беспроводной передачи данных является максимальная дальность передачи — в пределах зоны покрытия сети. GSM-модемы применяются в различных областях: измерительные системы и телеметрия, медицина, транспорт, системы обеспечения безопасности и охраны, M2M (machine-to-machine) автоматизация, игры и досуг и т. д.**

**Владимир Каулио, к. т. н.**  
kv@efo.ru

## Введение

В данной статье речь пойдет о наиболее популярных модемах с поддержкой Java — F6523AAx (G24-J), полностью совместимыми с модемами G24. Основные технические характеристики, функциональные возможности и теоретические сведения, посвященные модемам G24, рассмотрены в [1, 2]. Ядром G24-J является процессор ARM7, позволяющий обращаться к 1,8 Мб ОЗУ для хранения временных данных и 10 Мб ПЗУ для хранения программы и констант. G24-J содержит 15 линий ввода/вывода, 2 интерфейса UART, интерфейс USB, 5-канальный аналого-цифровой преобразователь (АЦП), встроенный TCP/IP-стек, а также поддержку протоколов HTTP (S), защищенного SSL-соединения, возможность обновления системного программного обеспечения. Встроенный АЦП обладает следующими характеристиками: разрешающая способность 10 бит, время срабатывания 10 мс, 3 канала АЦП для измерения внешних аналоговых сигналов, 1 канал для измерения температуры посредством встроенного в G24-J датчика температуры, 1 канал для анализа напряжения питания. Таким образом, возможностей модема в ряде случаев достаточно для создания системы без дополнительного контроллера.

Одной из главных причин использования GSM-модемов со встроенной поддержкой Java является возможность управления системой процессором, который встроен в GSM-модем и берет на себя нагрузку по функционированию беспроводного интерфейса, а также обладает достаточными ресурсами для хранения, защиты и обработки передаваемых данных. Таким образом, мидлет становится не просто полезной утилитой в сотовом телефоне, а основным инструментом разработчика для управления GSM-каналом. Особенность использования Java-мидлетов в качестве средства разработки дает приложению такое преимущество, как кросс-платформенность — возможность запуска приложения на любом устройстве, поддерживающем механизм виртуальной машины J2ME.

В настоящее время разработкой мидлетов занимаются не только фирмы-производители. Мидлеты создают компании и научные группы, которые разрабатывают готовые модемные решения и прикладное программное обеспечение, а также отдельные энтузиасты, увлекающиеся программированием мобильных приложений.

Можно сказать, что с появлением на отечественном рынке мобильных телефонов и беспроводных модемов с поддержкой Java, разработка мидлетов приобретает массовый характер. Причиной этого является стремительный рост информационных технологий, высокая конкуренция среди производителей мобильных решений, а также наличие бесплатных лицензионных средств разработки.

## Программные средства разработки мидлетов

Рассматриваемые в данной статье программные средства для разработки мидлетов можно разделить на две основные группы:

1. Программное обеспечение для редактирования, сборки, симуляции и отладки мидлетов.

2. Утилиты для прошивки мидлетов.

Выбор средств из первой группы программ не зависит от производителя модема и определяется типом операционной системы, а также функциональными возможностями, удобством интерфейса, полнотой справочной системы и т. п. Программы из второй группы предназначены для переноса мидлета из персонального компьютера в модем. Такие утилиты, как правило, предназначены для устройств определенной фирмы и заданного набора моделей телефонов (модемов).

Рассмотрим программное обеспечение, предоставляемое компанией Motorola. Для получения доступа к лицензионным программам необходимо зарегистрироваться на сайте компании Motorola. Установка USB-драйвера позволяет операционной системе Windows XP распознавать

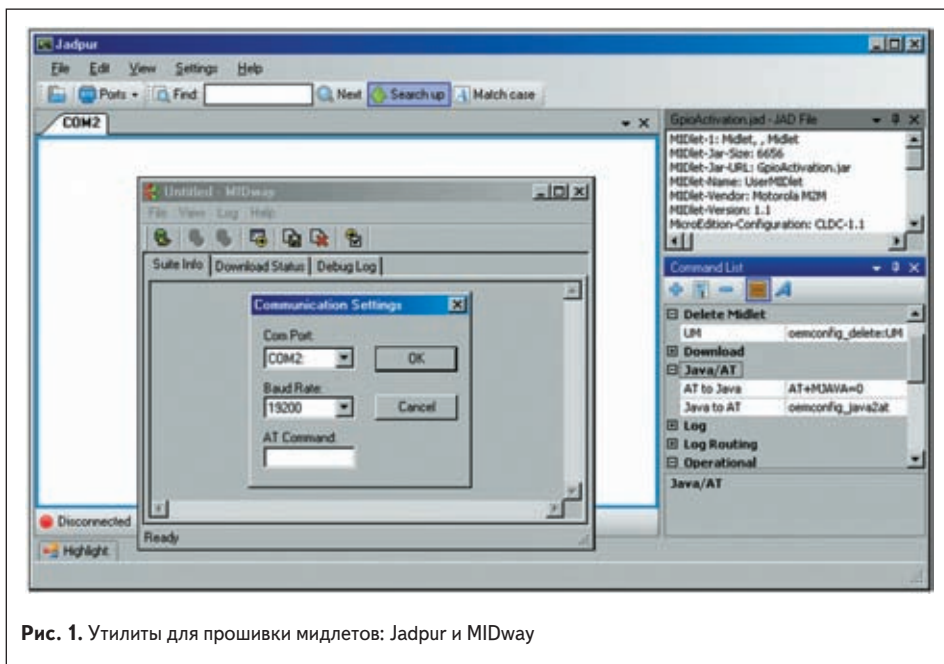


Рис. 1. Утилиты для прошивки мидлетов: Jadrun и MIDWay

модем Motorola и дает возможность посылать AT-команды по USB-интерфейсу. Посылка AT-команд доступна в режиме внешнего управления или смешанном Java-режиме.

Комплект разработчика Motodev SDK for Java ME включает программную платформу модемов G24-J, а также является средством конфигурирования большого числа сотовых телефонов Motorola. Платформа содержит справочник по библиотекам Java, а также библиотеки, требуемые для использования специфических функций модема G24-J, отсутствующие в стандартном наборе Java. Например, функции управления линиями ввода/вывода или получения результатов измерений с интегрированного АЦП. Данный программный продукт требует предварительной установки JRE-среды (Java SE Runtime Environment) не ниже версии 1.5.

Среда исполнения JRE является свободно распространяемой и содержит необходимые для запуска Java-приложений технологии: виртуальную машину, стандартные библиотеки классов, механизм запуска. Последняя версия JRE-среды доступна на сайте компании Sun Microsystems. В частности, JRE-среда входит в состав апрельской версии 2008 г. (v.1.6.0) комплекта разработчика JDK — Java SE Development Kit.

Перенос мидлета из персонального компьютера в GSM-модем или мобильный телефон может быть выполнен с помощью утилиты прошивки мидлета MIDWay v.2.8 [3]. Альтернативой служит утилита G24 MIDlet Downloading Tool Jadrun v1.0, она содержит список конфигурационных команд для удобства их поиска и запуска путем двойного клика, а также может служить в качестве терминала для отправки AT-команд. В качестве примера можно привести команду oemdownload, которая инициализирует процесс прошивки мидлета в GSM-модем. Для запуска файла jadrun.exe требуется установка платформы Microsoft.Net Framework. Особенность обеих программ состоит в использовании последовательного порта для «прошивки» мидлета (рис. 1).

Если на компьютере последовательный порт отсутствует, допустимо использование конвертеров COM-USB с предварительной установкой драйвера виртуального COM-порта.

Для удобства разработки и отладки мидлетов можно использовать полнофункциональную среду разработки Java-мидлетов Motodev Studio for Java ME. Она позволяет выполнять операции редактирования, симулирования, отладки и упаковки мидлета в jar/jad-файлы [1, 4]. В качестве среды разработки могут использоваться и другие средства, например NetBeans IDE, Eclipse, IntelliJ IDEA и др. Но в таком случае требуется отдельно устанавливать пакет Motodev SDK for Java ME и подключать платформу Motorola в используемой среде. Иначе специфические функции доступа к аппаратным возможностям модема G24-J будут недоступны.

NetBeans IDE - Integrated Development Environment — свободно-распространяемая интегрированная среда разработки для создания профессиональных персональных, промышленных, сетевых и мобильных приложений с использованием языков программирования Java, C/C++ и др. на платформах Windows, Linux, Mac OS X, Solaris и т. п. При разработке и тестировании мидлетов автором использовалась NetBeans IDE версии 6.1. В отличие от предыдущей версии, данный продукт не требует установки отдельного модуля мобильных приложений. Он содержит встроенные средства редактирования Java-кода, имеет графический симулятор панели сотового телефона, обладает удобной системой контекстной помощи и дружелюбный пользовательский интерфейс.

### Создание мидлетов

Рассмотрим процесс создания простейшего мидлета для управления линиями ввода/вывода GSM-модема G24-J. Первый этап — создание нового проекта в среде NetBeans IDE. Для этого выбирается пункт меню “File→New Project...”, в появившемся диалоговом окне (рис. 2) осуществляется выбор проекта с указанием категории проекта — “Mobility” и вида проекта — MIDP Application (Mobile Information Device Profile — профиль для мобильного устройства с информационными функциями). Модем G24-J поддерживает спецификацию MIDP 2.0, которая определяет функции жизненного цикла, управления, интерфейса и безопасности мидлета. Следующим шагом данного этапа является задание имени и размещения проекта. В заключение указывается платформа Java ME, используемая по умолчанию, а также указываются версии ее составляющих: конфигурация CLDC 1.1 (Connected Limited Device Configuration — определяет вычислительные возможности с ограничениями подключаемого к сети мобильного устройства) и профиль MIDP 2.0. Если на момент создания проекта платформа Motorola недоступна, ее необо-

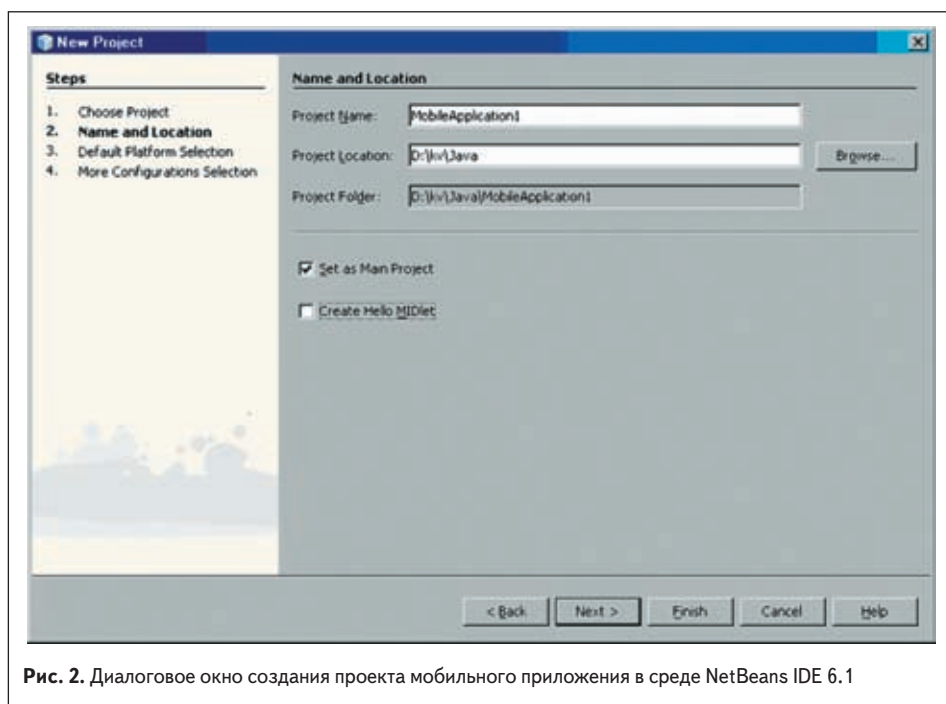


Рис. 2. Диалоговое окно создания проекта мобильного приложения в среде NetBeans IDE 6.1

димо добавить отдельно в свойствах проекта после создания.

Второй этап — добавление мидлета с исходным кодом на Java (по аналогии с добавлением CPP-файла в C++). Для этого в среде NetBeans IDE выбирается пункт меню “File→New File”, в появившемся диалоговом окне выбирается категория (MIDP) и тип файла (MIDlet). Затем указывается имя файла, автоматически будет сгенерировано имя класса. Файл с исходным текстом мидлета имеет расширение java. Новый мидлет содержит строки для подключения базовых библиотек и класс с основными функциями: startApp (выполнение мидлета), pauseApp (состояние в режиме паузы), destroyApp (завершение мидлета) [5–7].

Независимо от того, какое название дано проекту, классам и исходным файлам, требуется установить значение параметра MIDlet-Name — UserMIDlet. Все параметры, начальные значения параметров и константы (например, номер телефона для отправки SMS-сообщений) хранятся в файле с расширением jad, который создается в процессе подготовки мидлета к прошивке командой Deploy. В результате данной операции также создается файл архива с расширением jar, который содержит упакованный код, готовый к записи в модем для выполнения на виртуальной машине Java2ME. Операция переименования является обязательным требованием для загрузки мидлета в модем G24-J и может быть выполнена путем редактирования jad-файла в текстовом режиме, либо в среде NetBeans IDE: “Properties→Application descriptor→Attributes→MIDletname→Edit...”. Размещение параметров в jad-файле позволяет редактировать свойства программы, не затрагивая основной код программы и не меняя характеристик архива. Данное свойство удобно использовать в проектах, где необходима большая гибкость без доработки мидлета.

Написанный на Java мидлет не требует компиляции, среда автоматически проверяет код на наличие синтаксических ошибок. Убедиться в этом можно, добавив следующий код подключения библиотек Motorola для возможности использования функций чтения/записи состояний линий ввода/вывода модема G24-J (рис. 3). Если платформа Motorola не была предварительно установлена, то функции, отсутствующие в стандартном пакете Java, будут недоступны. Среда проектирования NetBeans IDE в режиме реального времени выполняет автоматическую проверку и сигнализирует об ошибках. Например, на рис. 3 индикаторы красного цвета с восклицательными знаками означают недоступность пакетов функций Motorola.

Чтобы сделать функции Motorola доступными для использования, необходимо подключить платформу Motorola: “Properties→Platform→Manage Emulators→Add platform→Java ME MIDP Platform Emulator→M2M\_WTK\_12” и выбрать “MOTO2MOTO Wireless Toolkit 1.2 Powered by Sun Java (TM) Technology” в качестве платформы.

Далее добавляется код для обнаружения состояния входа GPIO2 и вывода результата на выход GPIO1 (рис. 4), выполняется сборка (Build) проекта и подготовка мидлета к переносу на внешнее устройство (модем). Восклицатель-

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

import com.motorola.oem.hapi.GPIOException;
import com.motorola.oem.hapi.GPIOInput;
import com.motorola.oem.hapi.GPIOOutput;
```

Рис. 3. Библиотеки с функциями ввода/вывода

ный знак желтого цвета предупреждает, что импорт пакета lcdui не используется:

```
import javax.microedition.lcdui.*;
```

Размещение основного кода программы в функции startApp означает, что код будет выполнен однократно при старте мидлета. В данном примере выполняется постоянный опрос состояния линии ввода/вывода в бесконечном цикле. На практике бесконечный цикл в функции startApp не используется, поскольку такой подход исключает возможность выполнения других функций.

В данном примере бесконечный цикл приведен в качестве демонстрации по аналогии с программами для микроконтроллеров. Несмотря на то, что модем G24-J содержит одно ядро, и в процессе вычислений возможно только последовательное выполнение инструкций процессора, существует способ организовать псевдопараллельное выполнение задач, используя механизм потоков [8].

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

import com.motorola.oem.hapi.GPIOException;
import com.motorola.oem.hapi.GPIOInput;
import com.motorola.oem.hapi.GPIOOutput;

/**
 * @author kv
 */
public class UserMIDlet extends MIDlet
{
    private GPIOOutput gpioOutput;
    private GPIOInput gpioInput;

    public void startApp() throws MIDletStateChangeException
    {
        try
        {
            gpioOutput = new GPIOOutput(1);
            gpioInput = new GPIOInput(2);

            while(true)
            {
                if(gpioInput.read() == true)
                {
                    gpioOutput.write(true);
                }
                else
                {
                    gpioOutput.write(false);
                }
            }
        }
        catch (GPIOException e)
        {
            if(gpioInput != null)
                gpioInput.clear();
            if(gpioOutput != null)
                gpioOutput.clear();
            e.printStackTrace();
        }
    }
}
```

Рис. 4. Пример кода для работы с линиями ввода/вывода модема G24-J

## Управление режимами работы модема и прошивка мидлетов

Рассмотрим команды перехода между режимами работы модема, поскольку множество вопросов, связанных с программированием модема G24-J, возникает именно на этапе его включения.

По умолчанию модем G24-J находится в Java-режиме, в котором AT-команды [9] не воспринимаются. В Java-режиме модем может находиться в двух состояниях [4]: MIDlet Mode (управление со стороны мидлета) и JTOOL Mode (управление конфигурационными командами по последовательному порту UART). Переключение между состояниями осуществляется соответственно подачей сигнала логической единицы или нуля на вход GPIO8, контакт 42 [10].

Стандартное управление модемом с помощью AT-команд возможно для модемов G24-J, как и для всех других модемов Motorola серии G24. Данный способ управления возможен либо в режиме внешнего управления, либо в смешанном Java-режиме. Переход из Java-режима в режим внешнего управления осуществляется конфигурационной командой oemconfig\_java2at. Команда посылается по последовательному порту UART2. Возврат в Java-режим осуществляется AT-командой AT+MJAVA=0. Управление AT-командами из ограниченного набора возможно и в Java-режиме. Включение реакции на AT-команды в Java-режиме называется сме-



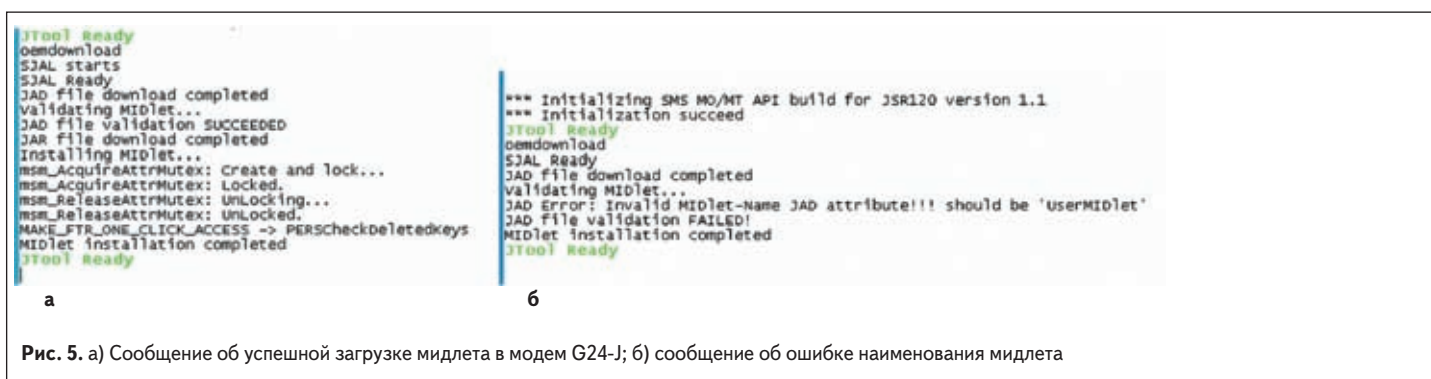


Рис. 5. а) Сообщение об успешной загрузке мидлета в модем G24-J; б) сообщение об ошибке наименования мидлета

шанным Java-режимом. Переход в смешанный режим возможен только из мидлета с помощью функции `OSC.setProperty` [9]. Данная функция может быть использована и для перехода между основными режимами работы GSM-модема. «Прошивка» мидлета осуществляется в Java-режиме при низком уровне сигнала на линии GPIO8. Рассмотрим процесс «прошивки» программой `Jadpur`. Сначала выбирается COM-порт, к которому подключен модем. Список возможных подключений находится на панели инструментов. Подтверждением успешного подключения служит индикатор `Connected` зеленого цвета в левой нижней части главного окна программы. Если модем находится в Java-режиме, на экран будет выведено сообщение о готовности — `JTool Ready`. Затем выбирается JAD-файл: «File → Open JAD...». «Прошивка» мидлета выполняется нажатием клавиши `Download`. Результат загрузки будет выведен на экран (рис. 5). Стирание мидлета выполняется по команде `oemconfig_delete: UM`. Выполнение мидлета начинается после переключения линии GPIO8 в состояние логической единицы.

В процессе разработки и тестирования мидлетов следует помнить о системе защиты, которая блокирует запуск мидлета в случае несоответствия параметров электронной подписи. Например, подпись необходима для мидлетов, содержащих функции отправки SMS. Существует два способа применения электронной подписи для модемов G24-J:

1. Генеральная подпись — используется, когда разработка мидлета полностью завершена и не требует изменений. После подписи мидлет будет функционировать на любом экземпляре G24-J, но редактирование кода приведет к не-

обходимости новой подписи. Для получения генеральной подписи мидлет посылается в технический отдел компании Motorola.

2. Модульная подпись — используется на этапе разработки мидлета, позволяет многократно редактировать мидлет, но разрешает функционирование подписанного мидлета на выборочном наборе модемов от 1 до 10 на одну подпись. Модульная подпись вырабатывается на основе параметров модема: IMEI (International Mobile Equipment Identity — Международный Идентификатор Мобильного Оборудования) и UID (Universal Identifier — универсальный идентификатор). Полный список функций, на которые накладываются ограничения по неавторизованному выполнению, а также подробное описание извлечения параметров модема изложено в [11].

## Заключение

В статье выполнен обзор программного обеспечения, используемого при разработке мидлетов для GSM-модемов G24-J компании Motorola. Рассмотрены этапы создания проекта в среде NetBeans IDE 6.1 и процесс «прошивки» мидлета посредством программы `Jadpur`. Особое внимание уделено командам перехода между режимами работы модема G24-J. Приведен пример мидлета для управления линиями ввода/вывода. ■

## Литература

1. Федоров В. В. Встраиваемые GSM/GPRS/EDGE-модули g24 компании Motorola // Электронные компоненты. 2007. № 8.

2. Модуль G24. Техническое описание и ссылки на документацию по GSM-модемам Motorola, <http://www.efo.ru/doc/Motorola/Motorola.pl?2475>.
3. Pinsonneault M. White Paper. Installing J2ME MIDlets Using MIDway. January 30, 2003.
4. Motorola G24 KJAVA. User's Guide, Technical information. May 15, 2007. [ftp://ftp.efo.ru/pub/motorola/gsm\\_modem/G24\\_KJAVA\\_User\\_Guide.pdf](ftp://ftp.efo.ru/pub/motorola/gsm_modem/G24_KJAVA_User_Guide.pdf)
5. Пирумян В. Платформа программирования J2ME для портативных устройств / Пер. с англ. М.: КУДИЦ-ОБРАЗ, 2002.
6. Горнаков С. Г. Программирование мобильных телефонов на Java 2 Micro Edition. М.: ДМК-Пресс, 2004.
7. Буткевич Е. Л. Пишем программы и игры для сотовых телефонов. СПб.: Питер, 2006.
8. Threading in Java ME. Technical article. [https://developer.motorola.com/docstools/technicalarticles/Threading\\_in\\_Java\\_ME.pdf](https://developer.motorola.com/docstools/technicalarticles/Threading_in_Java_ME.pdf)
9. Motorola G24 Developer's Guide. AT Commands Reference Manual, Technical information. December 31, 2007. [ftp://ftp.efo.ru/pub/motorola/gsm\\_modem/G24\\_AT\\_Commands\\_Reference\\_Manual\\_ver\\_E\\_310107.pdf](ftp://ftp.efo.ru/pub/motorola/gsm_modem/G24_AT_Commands_Reference_Manual_ver_E_310107.pdf)
10. Motorola G24 Developer's Guide. Module Hardware Description, Technical information. December 31, 2007. [https://developer.motorola.com/docstools/developerguides/G24\\_HardwareDescription\\_Developer\\_Guide.pdf](https://developer.motorola.com/docstools/developerguides/G24_HardwareDescription_Developer_Guide.pdf)
11. MIDlet Signing. Technical article. [https://developer.motorola.com/docstools/technicalarticles/MIDlet\\_Signing.pdf](https://developer.motorola.com/docstools/technicalarticles/MIDlet_Signing.pdf)