

Oracle® Java ME Embedded

Getting Started Guide for the Reference Platform (STM
32F746GDISCOVERY)

Release 8.3 Developer Preview

E71184-02

July 2016

Beta Draft

ORACLE CONFIDENTIAL.

For authorized use only.

Do not distribute to third parties.

Oracle Java ME Embedded Getting Started Guide for the Reference Platform (STM 32F746GDISCOVERY),
Release 8.3 Developer Preview

E71184-02

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Primary Author:

Contributors:

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

This documentation is in preproduction status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your beta trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Contents

Preface	v
Audience	v
Related Documents.....	v
Shell Prompts.....	v
Conventions.....	v
1 Installing the Java ME Embedded Software on the STM 32F746GDISCOVERY	
Required Hardware and Software Items	1-1
Downloading and Installing the MDK-ARM Development Kit.....	1-1
Setting Up the SD Card.....	1-2
Oracle Java ME Embedded Distribution Bundle	1-3
Downloading and Installing the PuTTY Terminal Emulator Program	1-4
A STM 32F746GDISCOVERY Device I/O Preconfigured List	
GPIO Pins.....	A-1
ADC	A-2
SPI	A-3
I2C	A-4
UART.....	A-4

Glossary

Preface

This guide describes how to install Oracle Java ME Embedded software onto a STM 32F746GDISCOVERY device.

Audience

This guide is for developers who want to run Oracle Java ME Embedded software on a STM 32F746GDISCOVERY device.

Related Documents

For a complete list of documents for the Oracle Java ME Embedded software, see the *Release Notes*.

Shell Prompts

Shell	Prompt
Windows	<i>directory></i>
Linux	\$

Conventions

The following text conventions are used in this guide:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Installing the Java ME Embedded Software on the STM 32F746GDISCOVERY

Learn how to install the Java ME Embedded software 8.3 onto an ST Micro STM 32F746GDISCOVERY board.

Topics

- [Required Hardware and Software Items](#)
- [Downloading and Installing the MDK-ARM Development Kit](#)
- [Setting Up the SD Card](#)

Required Hardware and Software Items

Before you start developing on the STM 32F746GDISCOVERY board, you need to prepare the following hardware and software items.

The following items are required for developing on the STM 32F746GDISCOVERY board:

- An STM 32F746GDISCOVERY board
- The Oracle Java ME Embedded Software Distribution for the STM 32F746GDISCOVERY board, version 8.3
- A desktop computer running Windows 7 or later with at least one USB port
- A 2GB or greater MicroSD card (with an SD adapter, if necessary, for connecting to the desktop computer)
- A USB-A to USB-B cable to flash the embedded board
- A networking LAN cable with RJ-45 interface, if you wish to communicate with the embedded board over a TCP/IP network
- A USB-A to MicroUSB cable, if you wish to communicate with the embedded board over a serial connection
- A terminal emulation program, such as PuTTY

Downloading and Installing the MDK-ARM Development Kit

Installing MDK (Microprocessor development kit) 5.18 is a prerequisite for installing the Java ME Embedded 8.3 on the STM 32F746GDISCOVERY board.

MDK 5.1.8 must be installed in a separate directory. To install MDK 5.18, use the following procedure:

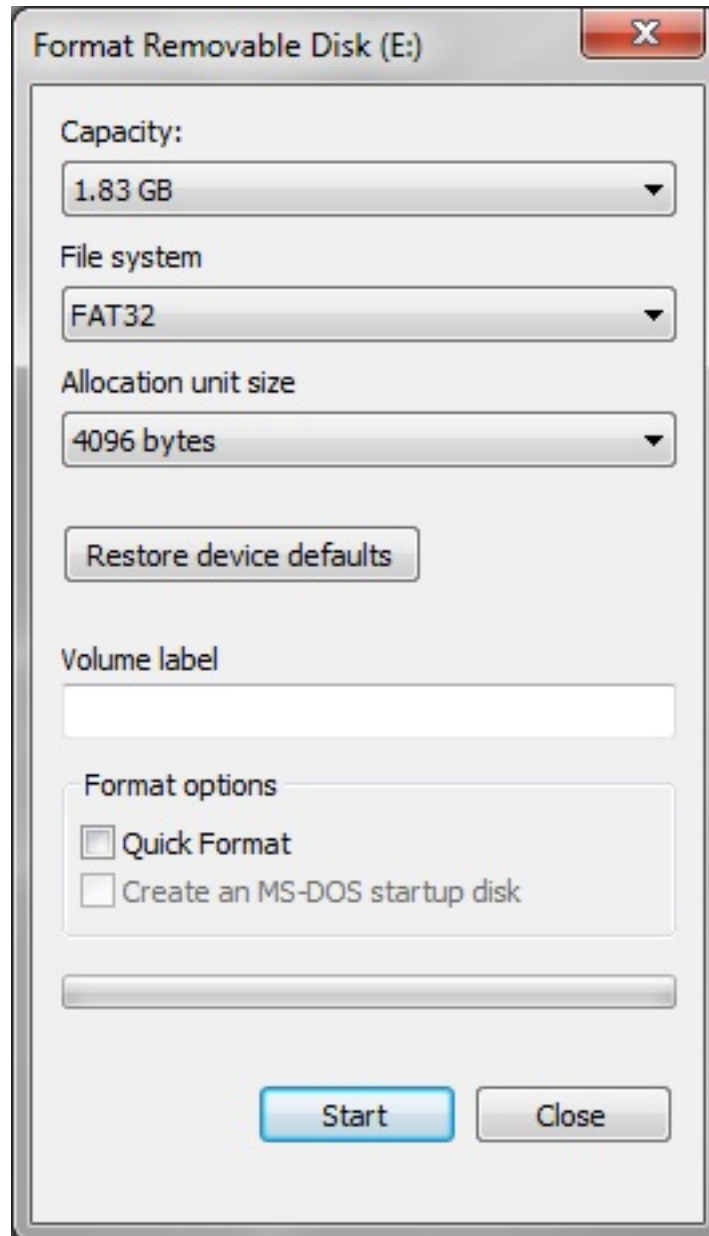
1. Download MDK 5.18 from the following site:<http://www2.keil.com/mdk5/install>.
2. Install the MDK tool by double-clicking on its executable.

Setting Up the SD Card

The SD card must be formatted in FAT/FAT32. The SD card contains configuration files used by the Java ME Embedded native distribution, including the initialization properties and security policy files.

Follow these steps to prepare the card:

1. Insert the card on the Windows desktop computer, select the card in the My Computer window, and right-click and select **Format**.
2. Select File System as FAT32, Allocation Unit Size as the default allocation unit size, and ensure that Quick Format is **not** selected, as shown in [Figure 1-1](#). The Volume label is optional. Press the **Start** button. Once the formatting is completed, move to the next step.

Figure 1-1 Windows Formatter Settings

3. Copy the `java/` folder inside the `sd_card/` directory of the Oracle Java ME Embedded distribution to the root directory of the SD card. Do not eject the SD card yet.

Oracle Java ME Embedded Distribution Bundle

The Oracle Java ME Embedded software for the STM 32F746GDISCOVERY board is delivered as a ZIP archive.

Download and unzip the Oracle Java ME Embedded Distribution for the STM 32F746GDISCOVERY board. The distribution bundle contains four directories that consist of the following important files:

/java/ deploy .bat

The `deploy.bat` file is a Windows script file that invokes the uVision MDK-ARM tool to flash the board with the contents of the Oracle Java ME Embedded binary file. You must edit this file to point to the installation directory of the Keil MDK-ARM tools directory (typically `C:\Keil_v5`) before flashing the embedded board.

/java/ java .uvoptx, /java/ java .uvprojx

These are uVision project files that are used by the Keil uVision MDK-ARM tool. You should not modify these files.

/java/ jmee .axf

This is the Oracle Java ME Embedded binary file that will be flashed on the ST Micro board, stored in an ARM executable format. Do not modify this file.

/java/ jmee .bin

/lib/ classes .zip

This ZIP file contains the Java class structures for all publicly reachable classes used by the Java ME executable.

/sd_card/ java/ jwc_properties .ini

This is the main properties file for the Java ME binary executable. Modify this file before copying the `/sd_card` directory to the SD card that is inserted into the embedded board to control various runtime elements of the Oracle Java ME Embedded binary.

/sd_card/ java/ appdb/ _main .ks

This the Java security keystore file that is used by the Java ME binary executable on the board. Do not attempt to modify this file directly. Instead, use the CLI keystore commands to access keys and certificates in this file.

/sd_card/ java/ appdb/ _policy .txt

This is the security policy file that defines policy groups for the Java ME binary executable on the board. Modify this file as needed to create your own security policies.

/sd_card/ java/ appdb/ daapi_config .json

The root filesystem is a user-accessible directory that can be used to store and retrieve files and data through programs running on the board. Note that data above this directory is inaccessible by user programs.

/toolkit-lib

/util/ proxy .jar

This is an executable JAR file that can be used to connect to the CLI proxy to issue commands to the board.

Downloading and Installing the PuTTY Terminal Emulator Program

The PuTTY terminal emulator is used to connect to the AMS command-line interface (CLI) that sends commands to the board.

Download the PuTTY terminal emulator program (`putty.exe`) from the following site:

<http://www.putty.org/>

Note:

Using the PuTTY terminal emulator program is highly recommended. You can use any terminal program to connect to the CLI, however, Oracle cannot guarantee that other terminal programs work with the CLI in the same manner as PuTTY.

STM 32F746GDISCOVERYDevice I/O Preconfigured List

This appendix describes the proper ID and names for the various peripheral ports and buses for the STM 32F746GDISCOVERY board, which are accessible using the Device I/O APIs.

To access any device from the preconfigured peripheral list, the following permission is required:

```
jdk.dio.DeviceMgmtPermission(%Name%:%ID%);
```

You can find the names and IDs for specific devices in the tables that follow in this appendix. You must also specify an action. An empty string means open.

The tables use the following legend:

- **Device ID:** an integer identifier that can be used to open the device with the methods of the `DeviceManager` class.
- **Device Name:** the string name of a device that can be used to open it by name with the methods of the `DeviceManager` class.
- **Mapped:** all hardware-related information regarding a peripheral, such as physical location, mapping, or port. This information enables the user to determine the peripheral's location on a target board.
- **Configuration:** properties that are passed to the specific `DeviceConfig` constructor to open the peripheral by ID or name. The configuration can be used to open the peripheral using the `DeviceManager` with the appropriate configuration.

GPIO Pins

The following GPIO pins are preconfigured.

Device ID	Device Name	Mapped	Configuration
1	USER_BUTTON GPIO8.11 GPIO0	A blue user button marked as B1 on the board	<code>controllerNumber = 8</code> <code>pinNumber = 11</code> <code>direction =</code> <code>GPIOPinConfig.DIR_INPUT_ONLY</code> <code>driveMode =</code> <code>GPIOPinConfig.MODE_INPUT_PULL_D</code> <code>OWN</code>

Device ID	Device Name	Mapped	Configuration
2	LED_1 GPIO8.1 GPIO1	A green user LED marked as LD1 on the board	controllerNumber = 8 pinNumber = 1 direction = GPIOPinConfig.DIR_OUTPUT_ONLY driveMode = GPIOPinConfig.MODE_OUTPUT_PUSH_PULL initValue = false

Consider the following implementation notes concerning GPIO on the STM32F746GDISCOVERYboard.

Port	Pin	Arduino Connector Pin
0	8	ARD_D5
0	15	ARD_D3
1	4	ARD_D3
6	6	ARD_D2
6	7	ARD_D4
7	6	ARD_D6
8	1	ARD_D13 or USER GREEN LED
8	2	ARD_D8
8	3	ARD_D7
8	11	USER BUTTON

- Device shared mode is not supported.
- Unassigned controllerNumber is treated as 6.
- Unassigned pinNumber is treated as 6.
- Unassigned driveMode is PULL_DOWN for input pins and OPEN_DRAIN for output pins.

ADC

Device ID	Device Name	Mapped	Configuration
1100	WDG WATCHDOG0	WatchdogTimer	

Please note the following items about ADC on the STM 32F746GDISCOVERY.

- The board supports sampling intervals (in μ s):
 - 1.1
 - 2.0
 - 3.0
 - 5.0
 - 7.1
 - 9.2
 - 11.6
 - 36.4

SPI

SPI is supported on the STM 32F746GDISCOVERY. No SPI devices are preconfigured because there are no on-board SPI devices.

Consider the following implementation notes concerning SPI on the STM 32F746GDISCOVERY board.

- Only one SPI bus is supported with `controllerNumber = 2` or `DeviceConfig.UNASSIGNED`.
- SPI pins are mapped as follows.

SPI Line	Hardware Pins
MOSI	ARD_D11 or PB15
MISO	ARD_D12 or PB14
SCK	ARD_D13 or PI1
SS, CS, or NSS	ARD_D5

- Only address 0 is supported.
- `clockFrequency.UNASSIGNED` is treated as 10MHz.
- Supported `wordLength` are only 8bit and 16bit. `wordLength.UNASSIGNED` is treated as 8.
- Only `BIG_ENDIAN` and `LITTLE_ENDIAN` ordering are supported. `UNASSIGNED` bit ordering is treated as `BIG_ENDIAN`.

I2C

Device ID	Device Name	Mapped	Configuration
200	ARD_UNO_I2C I2C1.29 I2C0	I2C pins on Arduino UNO connector or D14 and D15 pins on STM32F746G- DISCOSTM 32F746GDISC OVERY board	controllerNumber=1 address=29 addressSize=7 clockFrequency=100000

Please note the following items about I2C on the STM 32F746GDISCOVERY.

- Only one I2C bus is supported. It is SDA and SCL pins that are present in Arduino UNO connector on a board.
- The `controllerNumber` can be set to 1. If `DeviceConfig.UNASSIGNED` is set to `controllerNumber`, it is treated as 1.
- Both address sizes `I2CDeviceConfig.ADDR_SIZE_7` and `I2CDeviceConfig.ADDR_SIZE_10` are supported. `DeviceConfig.UNASSIGNED` is treated as `I2CDeviceConfig.ADDR_SIZE_7`.
- The value of `DeviceConfig.DEFAULT` when applied to the `addressSize` is 7.
- The `clockFrequency` can be set to 100 000 and 400 000. `DeviceConfig.UNASSIGNED` is treated as 100 000.

UART

The following UART devices are preconfigured:

Device ID	Device Name	Mapped	Configuration
100	USART6 UART0	USART pins mapped to ARD_D0 and ARD_D1 pins	controllerName = USART6 baudRate = 115200 dataBits = DATABITS_8 parity = PARITY_NONE stopBits = STOPBITS_1 flowcontrol = FLOWCONTROL_NONE

Please note the following items about UART on the STM 32F746GDISCOVERY.

- Only two UART controllers are supported on the STM 32F746GDISCOVERY device.
- UART with controller name USART6 is mapped to ARD_D0 and ARD_D1 pins on the Arduino connector.

- UART with controller name USART1 is connected to ST-LINK embedded dongle and visible on host computer after connecting the board and installing ST-LINKv2 device drivers. This port can also be used for the device logging and can not be opened by application in such case.
- `PARITY_MARK` and `PARITY_SPACE` are not supported.
- Software and hardware flow control types are not supported.

Glossary

access point

A network-connectivity configuration that is predefined on a device. An access point can represent different network profiles for the same bearer type, or for different bearer types that may be available on a device, such as WiFi or Bluetooth.

ADC

analog-to-digital converter. A hardware device that converts analog signals (time and amplitude) into a stream of binary numbers that can be processed by a digital device.

AMS

Application Management System. The system functionality that completes tasks such as installing applications, updating applications, and managing applications between foreground and background.

APDU

Application Protocol Data Unit. A communication mechanism used by SIM cards and smart cards to communicate with card reader software or a card reader device.

API

application programming interface. A set of classes used by programmers to write applications that provide standard methods and interfaces and eliminate the need for programmers to reinvent commonly used code.

ARM

Advanced RISC Machine. A family of computer processors using reduced instruction set (RISC) CPU technology, developed by ARM Holdings. ARM is a licensable instruction set architecture (ISA) used in the majority of embedded platforms.

AT commands

A set of commands developed to facilitate modem communications, such as dialing, hanging up, and changing the parameters of a connection. Also known as the Hayes command set. AT means *attention*.

AXF

ARM Executable Format. An ARM executable image generated by ARM tools.

BIP

Bearer Independent Protocol. Allows an application on a SIM card to establish a data channel with a terminal, and through the terminal, to a remote server on the network.

CDMA

Code Division Multiple Access. A mobile telephone network standard used primarily in the United States and Canada as an alternative to GSM.

CLDC

Connected Limited Device Configuration. A Java ME platform configuration for devices with limited memory and network connectivity. It uses a low-footprint Java Virtual Machine such as the CLDC HotSpot Implementation, and several minimalist Java platform APIs for application services.

configuration

Defines the minimum Java runtime environment (for example, the combination of a Java Virtual Machine and a core set of Java platform APIs) for a family of Java ME platform devices.

DAC

digital-to-analog converter. A hardware device that converts a stream of binary numbers into an analog signal (time and amplitude), such as audio playback.

ETSI

European Telecommunications Standards Institute. An independent, non-profit group responsible for the standardization of information and communication technologies within Europe. Although based in Europe, it carries worldwide influence in the telecommunications industry.

GCF

Generic Connection Framework. A Java ME API consisting of a hierarchy of interfaces and classes to create connections (such as HTTP, datagram, or streams) and perform I/O.

GPIO

general purpose I/O. Unassigned pins on an embedded platform that can be assigned or configured as needed by a developer.

GPIO port

A group of GPIO pins (typically 8 pins) arranged in a group and treated as a single port.

GSM

Global System for Mobile Communications. A 3G mobile telephone network standard used widely in Europe, Asia, and other parts of the world.

HTTP

HyperText Transfer Protocol. The most commonly used Internet protocol, based on TCP/IP that is used to fetch documents and other hypertext objects from remote hosts.

HTTPS

Secure HyperText Transfer Protocol. A protocol for transferring encrypted hypertext data using Secure Sockets Layer (SSL) technology.

I2C

Inter-Integrated Circuit. A multimaster, serial computer bus used to attach low-speed peripherals to an embedded platform

ICCID

Integrated Circuit Card Identification. The unique serial number assigned to an individual SIM card.

IMP-NG

Information Module Profile Next Generation. A profile for embedded "headless" devices, the IMP-NG specification (JSR 228) is a subset of MIDP 2.0 that leverages many of the APIs of MIDP 2.0, including the latest security and networking+, but does not include graphics and user interface APIs.

IMEI

International Mobile Equipment Identifier. A number unique to every mobile phone. It is used by a GSM or UMTS network to identify valid devices and can be used to stop a stolen or blocked phone from accessing the network. It is usually printed inside the battery compartment of the phone.

IMlet

An application written for IMP-NG. An IMlet does not differ from MIDP 2.0 MIDlet, except by the fact that an IMlet cannot refer to MIDP classes that are not part of IMP-NG. An IMlet can only use the APIs defined by the IMP-NG and CLDC specifications.

IMlet Suite

A way of packaging one or more IMlets for easy distribution and use. Similar to a MIDlet suite, but for smaller applications running in an embedded environment.

IMSI

International Mobile Subscriber Identity. A unique number associated with all GSM and UMTS network mobile phone users. It is stored on the SIM card inside a phone and is used to identify itself to the network.

ISA

Instruction Set Architecture. The part of a computer's architecture related to programming, including data type, addressing modes, interrupt and exception handling, I/O, and memory architecture, and native commands. Reduced instruction set computing (RISC) is one kind of instruction set architecture.

JAD file

Java Application Descriptor file. A file provided in a MIDlet or IMlet suite that contains attributes used by application management software (AMS) to manage the MIDlet or IMlet life cycle, and other application-specific attributes used by the MIDlet or IMlet suite itself.

JAR file

Java ARchive file. A platform-independent file format that aggregates many files into one. Multiple applications written in the Java programming language and their required components (class files, images, sounds, and other resource files) can be bundled in a JAR file and provided as part of a MIDlet or IMlet suite.

Java ME platform

Java Platform, Micro Edition. A group of specifications and technologies that pertain to running the Java platform on small devices, such as cell phones, pagers, set-top boxes, and embedded devices. More specifically, the Java ME platform consists of a configuration (such as CLDC) and a profile (such as MIDP or IMP-NG) tailored to a specific class of device.

JCP

Java Community Process. The global standards body guiding the development of the Java programming language.

JSR

Java Specification Request. A proposal for developing new Java platform technology, which is reviewed, developed, and finalized into a formal specification by the JCP program.

JVM

Java Virtual Machine. A software “execution engine” that safely and compatibly executes the byte codes in Java class files on a microprocessor.

KVM

A Java Virtual Machine designed to run in a small, limited-memory device. The CLDC configuration was initially designed to run in a KVM.

LCDUI

Liquid Crystal Display User Interface. A user interface toolkit for interacting with liquid crystal display (LCD) screens in small devices. More generally, a shorthand way of referring to the MIDP user interface APIs.

MIDlet

An application written for MIDP.

MIDlet suite

A way of packaging one or more MIDlets for easy distribution and use. Each MIDlet suite contains a Java Application Descriptor file (.jad), which lists the class names and files names for each MIDlet, and a Java ARchive file (.jar), which contains the class files and resource files for each MIDlet.

MIDP

Mobile Information Device Profile. A specification for a Java ME platform profile, running on top of a CLDC configuration that provides APIs for application life cycle, user interface, networking, and persistent storage in small devices.

MSISDN

Mobile Station Integrated Services Digital Network. A number uniquely identifying a subscription in a GSM or UMTS mobile network. It is the telephone number to the SIM card in a mobile phone and used for voice, FAX, SMS, and data services.

MVM

Multiple Virtual Machines. A software mode that can run more than one MIDlet or IMlet at a time.

obfuscation

A technique used to complicate code by making it harder to understand when it is decompiled. Obfuscation makes it harder to reverse-engineer applications and therefore, steal them.

optional package

A set of Java ME platform APIs that provides additional functionality by extending the runtime capabilities of an existing configuration and profile.

preverification

Due to limited memory and processing power on small devices, the process of verifying Java technology classes is split into two parts. The first part is preverification which is done off-device using the preverify tool. The second part, which is verification, occurs on the device at runtime.

Profile

A set of APIs added to a configuration to support specific uses of an embedded or mobile device. Along with its underlying configuration, a profile defines a complete and self-contained application environment.

Provisioning

A mechanism for providing services, data, or both to an embedded or mobile device over a network.

Pulse Counter

A hardware or software component that counts electronic pulses, or events, on a digital input line, for example, a GPIO pin.

Push Registry

The list of inbound connections, across which entities can push data. Each item in the list contains the URL (protocol, host, and port) for the connection, the entity permitted to push data through the connection, and the application that receives the connection.

RISC

reduced instruction set computing. A CPU design based on simplified instruction sets that provide higher performance and faster execution of individual instructions. The ARM architecture is based on RISC design principles.

RL-ARM

Real-Time Library. A group of tightly coupled libraries designed to solve the real-time and communication challenges of embedded systems based on ARM processor-based microcontroller devices.

RMI

Remote Method Invocation. A feature of Java SE technology that enables Java technology objects running in one virtual machine to seamlessly invoke objects running in another virtual machine.

RMS

Record Management System. A simple record-oriented database that enables an IMlet or MIDlet to persistently store information and retrieve it later. MIDlets can also use the RMS to share data.

RTOS

Real-Time Operating System. An operating system designed to serve real-time application requests. It uses multi-tasking, an advanced scheduling algorithm, and minimal latency to prioritize and process data.

RTSP

Real Time Streaming Protocol. A network control protocol designed to control streaming media servers and media sessions.

SCWS

Smart Card Web Server. A web server embedded in a smart card (such as a SIM card) that allows HTTP transactions with the card.

SD card

Secure Digital cards. A nonvolatile memory card format for use in portable devices, such as mobile phones and digital cameras, and embedded systems. SD cards come in three different sizes, with several storage capacities and speeds.

SIM

Subscriber Identity Module. An integrated circuit embedded into a removable SIM card that securely stores the International Mobile Subscriber Identity (IMSI) and the related key used to identify and authenticate subscribers on mobile and embedded devices.

Slave mode

Describes the relationship between a master and one or more devices in a Serial Peripheral Interface (SPI) bus arrangement. Data transmission in an SPI bus is initiated by the master device and received by one or more slave devices, which cannot initiate data transmissions on their own.

smart card

A card that stores and processes information through the electronic circuits embedded in silicon in the substrate of its body. Smart cards carry both processing power and information. A SIM card is a special kind of smart card for use in a mobile device.

SMS

Short Message Service. A protocol allowing transmission of short text-based messages over a wireless network. SMS messaging is the most widely-used data application in the world.

SMSC

Short Message Service Center. Routes messages and regulates traffic. When an SMS message is sent, it goes to an SMS center first, and then gets forwarded to the destination. If the destination is unavailable (for example, the recipient embedded board is powered down), the message is stored in the SMSC until the recipient becomes available.

SOAP

Simple Object Access Protocol. An XML-based protocol that enables objects of any type to communicate in a distributed environment. It is most commonly used to develop web services.

SPI

Serial Peripheral Interface. A synchronous bus commonly used in embedded systems that allows full-duplex communication between a master device and one or more slave devices.

SSL

Secure Sockets Layer. A protocol for transmitting data over the Internet using encryption and authentication, including the use of digital certificates and both public and private keys.

SVM

Single Virtual Machine. A software mode that can run only one MIDlet or IMlet at a time.

task

At the platform level, each separate application that runs within a single Java Virtual Machine is called a task. The API used to instantiate each task is a stripped-down version of the Isolate API defined in JSR 121.

TCP/IP

Transmission Control Protocol/Internet Protocol. A fundamental Internet protocol that provides for reliable delivery of streams of data from one host to another.

Terminal Profile

Device characteristics of a terminal (mobile or embedded device) passed to the SIM card along with the IMEI at SIM card initialization. The terminal profile tells the SIM card what values are supported by the device.

UART

Universal Asynchronous Receiver/Transmitter. A piece of computer hardware that translates data between serial and parallel formats. It is used to facilitate

communication between different kinds of peripheral devices, input/output streams, and embedded systems, to ensure universal communication between devices.

UICC

Universal Integrated Circuit Card. The smart card used in mobile terminals in GSM and UMTS networks. The UICC ensures the integrity and security of personal data on the card.

UMTS

Universal Mobile Telecommunications System. A third-generation (3G) mobile communications technology. It utilizes the radio spectrum in a fundamentally different way than GSM.

URI

Uniform Resource Identifier. A compact string of characters used to identify or name an abstract or physical resource. A URI can be further classified as a uniform resource locator (URL), a uniform resource name (URN), or both.

USAT

Universal SIM Application Toolkit. A software development kit intended for 3G networks. It enables USIM to initiate actions that can be used for various value-added services, such as those required for banking and other privacy-related applications.

USB

Universal Serial Bus. An industry standard that defines the cables, connectors, and protocols used in a bus for connection, communication, and power supply between computers and electronic devices, such as embedded platforms and mobile phones.

USIM

Universal Subscriber Identity Module. An updated version of a SIM designed for use over 3G networks. USIM is able to process small applications securely using better cryptographic authentication and stronger keys. Larger memory on USIM enables the addition of thousands of details including subscriber information, contact details, and other custom settings.

WAE

Wireless Application Environment. An application framework for small devices, which leverages other technologies, such as Wireless Application Protocol (WAP).

WAP

Wireless Application Protocol. A protocol for transmitting data between a server and a client (such as a cell phone or embedded device) over a wireless network. WAP in the wireless world is analogous to HTTP in the World Wide Web.

watchdog timer

A dedicated piece of hardware or software that "watches" an embedded system for a fault condition by continually polling for a response. If the system goes offline and no response is received, then the watchdog timer initiates a reboot procedure or takes other steps to return the system to a running state.

WCDMA

Wideband Code Division Multiple Access. A detailed protocol that defines how a mobile phone communicates with the tower, how its signals are modulated, how datagrams are structured, and how system interfaces are specified.

WMA

Wireless Messaging API. A set of classes for sending and receiving Short Message Service (SMS) messages.

XML Schema

A set of rules to which an XML document must conform to be considered valid.