

# ST25 SDK introduction (ST25 Software Development Kit)

January 2017





## ST25 SDK

Software library to be used in Java applications

What do I get from ST25 SDK?

Why shall I use it?

## ST25SDK in a nutshell

### Multiplatform

Can be run by any platform supporting JVM  
(Windows, Android, Linux, Mac)

Some components can be re-used for iOS

### Flexibility

Abstraction for NFC Forum & ISO Commands  
Abstraction for Tags, and Readers

RF commands isolated from UI interface  
Code can be reused in other Java applications

Datasheet is optional. Tag features documented in API (Javadoc )

**To accelerate and ease your development**



# ST25 SDK in a nutshell

What form does the ST25SDK take?

How do I get the latest ST25 SDK package?

What if I wish more documentation?

A public archive file (zip) containing:

- st25sdk.jar Java library file
- API documentation (HTML Javadoc files)
- Native libraries for reader board (CR95HF, ST25R3911B-DISCO)
- Basic applications source code (Windows PC, Android) + executable)
- Helper classes

<http://st.com/st25sdk>

Data brief + User Manual on st.com





# ST25 SDK in a nutshell

How is the ST25SDK licensed?

Through Mix\_MyLiberty (SLA 0052 v1)  
Included at the beginning of each source file

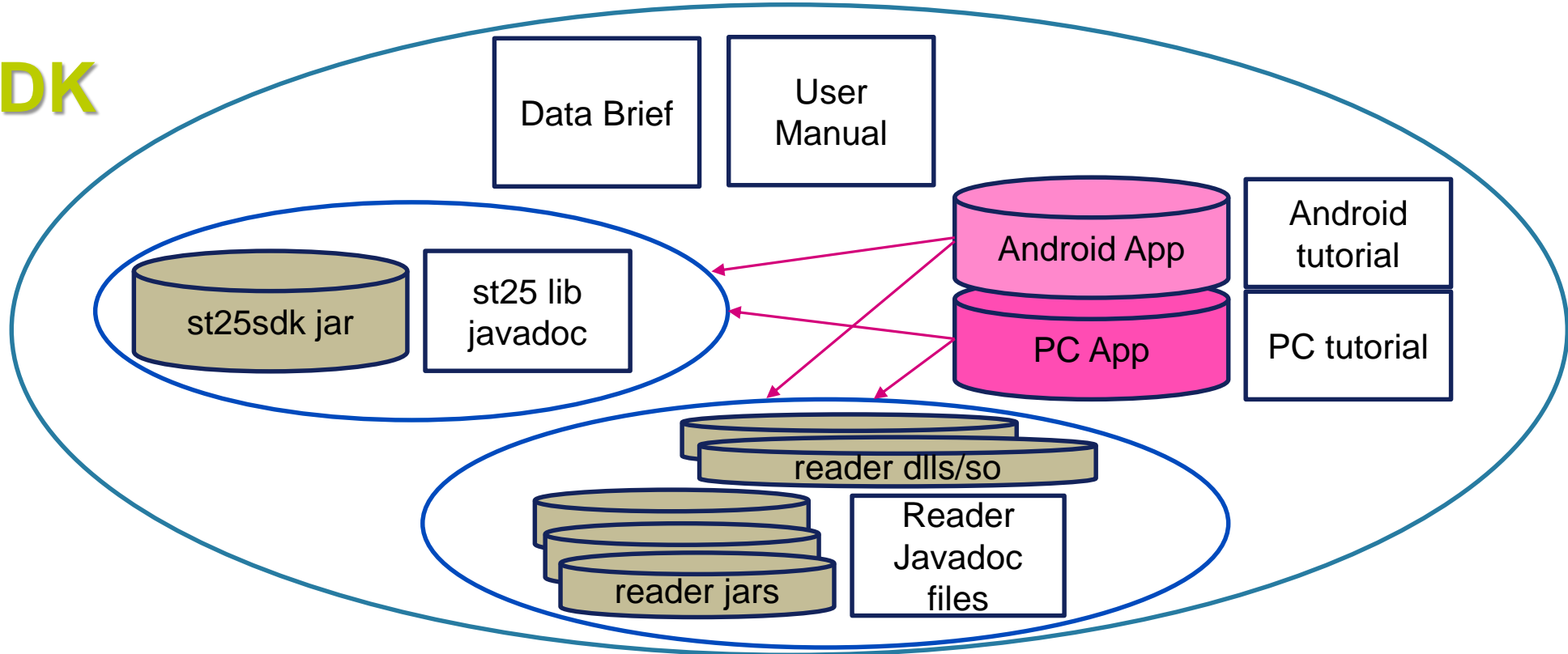
If I have any further questions?

<https://community.st.com/community/st25-nfcrcfid-tags-and-readers>





## ST25 SDK

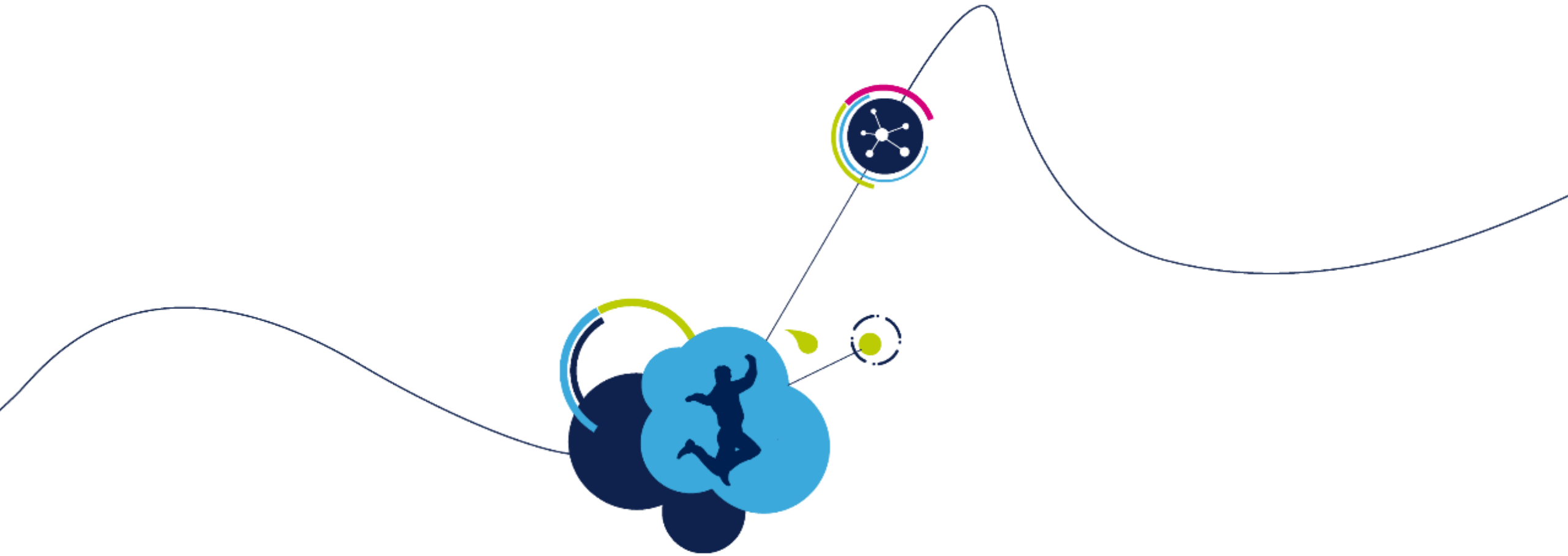


### Application examples:

Detect tag

Extract size information from **getSystemInfo** with **tag.getMemSizeInBytes()** call

Write NDEF message containing a URI record



# st25sdk Library Presentation



# The st25sdk Java library

7

- The st25sdk.jar library is central to the SDK
  - Include it inside an application to benefit from a lot of tag-related features
- st25sdk library architecture is designed to ease NFC application development

- Object oriented structure

- Tags have common properties
  - Specific fields and methods for tags that have extra features or different from base tags
  - We do not reinvent the wheel every time, just add or override fields and methods
- Tags inherit features based on tag hierarchy

- Cross platforms/readers support

- Can be run on any platform supporting Java Runtime Environment
  - Windows, Android, Linux, macOS
  - iOS: some components can be re-used (ndef)

- Cache tag information

- useful when tag is out of RF field



# How to benefit from st25sdk: a few examples

- Explicit API allows the application to read or write **NDEF formatted message** or **raw binary data** into each memory area of the tag.

```
st25DVTag.writeNdefMessage(MultiAreaInterface.AREA2, ndefMsgToWrite);
ndefMsgFromTag = st25DVTag.readNdefMessage(MultiAreaInterface.AREA1);

st25DVTag.writeBytes(offset, dataToWrite);
byte[] dataFromTag = st25DVTag.readBytes(offset, length);
```

- The read or write actions work no matter what the tag type is (Type5, Type4A...)

User doesn't have to change the command. It is identical for all tag types

- In case of error, an explicit **STException** is raised

The user application doesn't have to interpret the code returned by the command..

- With Type5 tags, extended commands should be used above the first 8kbits.

The application doesn't have to bother with specific read/write commands. The st25sdk manages it

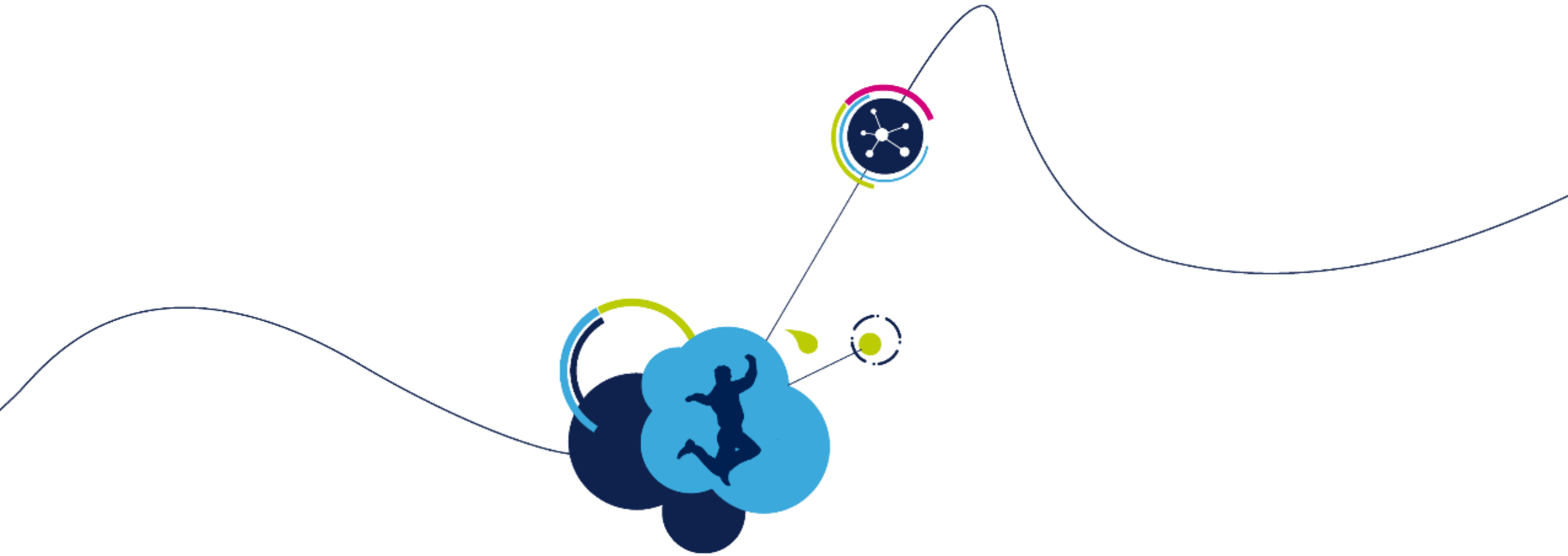




# Coding with and without SDK

- Imagine that we have a ST25DV64K tag configured with 4 memory areas
  - we want to write a NDEF message containing the URI [www.st.com](http://www.st.com) into Area4.
- Let's compare the (pseudo)code with and without SDK:

Without st25sdk library	With st25sdk library
<pre>Build a NDEF message:<ul style="list-style-type: none"><li>• Either with your own code</li><li>• Or with some other code like Android's NDEF message (<a href="https://developer.android.com/reference/android/nfc/NdefMessage.html">https://developer.android.com/reference/android/nfc/NdefMessage.html</a>)</li></ul>// Serialize the data: byte[] ndefData = ndefMsg.serialize();  // Read 'EndA3' register to know the « End Of Area 3 » and thus the beginning of Area4. Send readConfig command: 22 a0 02 9e 11 e3 03 00 27 02 e0 09 00 bf  // Compute the Area4 offset from EndA3 register value  // Write the data to the relevant memory addresses and with the relevant command:  Send extendedWriteSingleBlock command: 22 31 9e 11 e3 03 00 27 02 e0 02 06 03 00 d1 01  Send extendedWriteMultipleBlock command: 22 34 9e 11 e3 03 00 27 02 e0 03 06 02 00 07 55 01 73 74 2e 63 6f 6d fe ff ff  Send extendedWriteSingleBlock command: 22 31 9e 11 e3 03 00 27 02 e0 02 06 03 0b d1 01</pre>	<pre>NDEFMsg ndefMsg = new NDEFMsg();  UriRecord uriRecord = new UriRecord(NDEF_RTD_URI_ID_HTTP_WWW, "st.com"); ndefMsg.addRecord(uriRecord);  st25DVTag.writeNdefMessage(MultiAreaInterface.AREA4, ndefMsg);</pre> <div data-bbox="1302 953 2005 1196" style="border: 2px solid #90EE90; padding: 10px; margin: 10px 0;"><p><b>The exact same code works also for other tags with multiple Areas like M24SR or ST25TV!</b></p><p><b>It runs on Android and PC.</b></p></div>



Thank You !

**CONFIDENTIALITY OBLIGATIONS:**

THIS DOCUMENT CONTAINS SENSITIVE INFORMATION.

IT IS CLASSIFIED "MICROCONTROLLERS, MEMORIES & SECURE MCUs (MMS) RESTRICTED AND ITS DISTRIBUTION IS SUBMITTED TO ST/MMS AUTHORIZATION

AT ALL TIMES YOU SHOULD COMPLY WITH THE FOLLOWING SECURITY RULES:

- DO NOT COPY OR REPRODUCE ALL OR PART OF THIS DOCUMENT
- KEEP THIS DOCUMENT LOCKED AWAY
- FURTHER COPIES CAN BE PROVIDED ON A "NEED TO KNOW BASIS", PLEASE CONTACT YOUR LOCAL ST SALES OFFICE OR DOCUMENT WRITTER.

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is registered trademark of STMicroelectronics

All other names are the property of their respective owners

© 2010 STMicroelectronics - All Rights Reserved

STMicroelectronics group of companies Australia - Brazil - Canada - China - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States.

[www.st.com](http://www.st.com)