Tizen Platform SDK:
The Easy Way to Develop Tizen Platform

Donghyuk Yang, Donghee Yang, Yeongkyoon Lee, Hyungoo Kang
(S-Core Co., Ltd.)
Contents

• Introduction
• Key features (with demo)
• Upcoming features
• Conclusion
• Q&A
Introduction
What is Tizen Platform SDK?

• Development kit to develop Tizen platform module

**Tizen Platform SDK**

- **Platform IDE** (based on Eclipse 3.7)
- **Emulator** (based on QEMU)
- **Dynamic Analyzer** (*available soon)
- **SDB** (Smart Development Bridge)

IDE internally uses GBS to build a platform module

GBS (Git Build System)
Platform IDE

Clone source and create project with project wizard

Build

Debug

Run

Edit sources

Manage resources

Manage emulator and target

Manage Rootstraps
Emulator

• **Features**
  - Providing full system emulation, include CPU, memory and peripheral devices
  - Simulating events with the Emulator Control Panel
  - Accelerating guest operations using host CPU and GPU
Dynamic Analyzer

• Monitoring tool for improve performance and reliability of platform project

• Features
  • Timeline chart
    • CPU load, Heap/Memory, UI event, Disk/Network…
  • Summary
    • Failed API, Leak, Function profiling..
  • UX and other information
    • Call trace, Call stack, Record & replay, Save the trace…
SDB (Smart Development Bridge)

• SDB is device management tool
  • It manages multiple device connections
  • It supplies basic commands for development
    • File transfer
    • Remote shell
    • Port forwarding for a debugger
    • Device log output
    • Connect remote device

commands:
sdb root <on | off> - switch to root or developer account mode
  • 'on' means to root mode, and vice versa
sdb status-window - continuously print device status for a specified device
sdb get-serial - print: serial numbers
sdb get-state - print: offline | locked | device
sdb kill-server - kill the server if it is running
sdb start-server - ensure that there is a server running
sdb version - show version num
sdb help - show this help message
sdb forward <local> <remote> - forward socket connections
  • For example: sdb forward tcp:9999 tcp:9999
  • Uninstall an app from the device
sdb uninstall pkg_id - the pkg_id is an unique 18-digit unique identifier for the application. The
  following command shows an example:
    • Ex.) sdb uninstall com.examplepkg
sdb install pkg_path - push package file and install it
sdb diag [filter spec] - view device log
sdb shell [command] - if argument is null, run remote shell interactively
  • If argument is not null, run command in the remote shell
sdb pull <remote> [<local>] - copy file/dir from device
sdb push <local> <remote> [--with-utf8] - copy file/dir to device
  • --with-utf8 means to create the remote file with utf-8 character encoding
sdb disconnect [-host:<port>]] - disconnect from a TCP/IP device
  • Port 26301 is used by default if no port number is specified
  • Using this command with no additional arguments
    • Will disconnect from all connected TCP/IP devices
sdb connect <host>[::<port>]] - connect to a device via TCP/IP
  • Port 26301 is used by default if no port number is specified
  • List all connected devices
Why use Tizen Platform SDK?

- It makes platform module development easier

<table>
<thead>
<tr>
<th>Development Process</th>
<th>Without SDK</th>
<th>With SDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download source code</td>
<td>GIT command</td>
<td>IDE integrated (Egit, Platform project wizard)</td>
</tr>
<tr>
<td>Edit source code</td>
<td>VI / other editors</td>
<td>IDE integrated (C/C++, Specfile, Cmake, XML editors)</td>
</tr>
<tr>
<td>GBS build locally</td>
<td>GBS command</td>
<td>IDE integrated</td>
</tr>
<tr>
<td>Test</td>
<td>Create image using MIC / depends on developers</td>
<td>IDE supports Run &amp; Debug / Emulator</td>
</tr>
<tr>
<td>Commit changes</td>
<td>GIT command</td>
<td>IDE integrated (Egit)</td>
</tr>
</tbody>
</table>
Key features of Platform IDE

• Visual & Easy Editing
• Build based on Rootstrap
• Easy Testing (Run & Debug)
• Quick Launch (RDS)
• Platform Crash Analysis (Core dump debug)
• Easy Package Management (Package Manager)
Key features
Visual & Easy Editing

• Editors
  • C/C++ Editor
  • Spec File Editor
  • CMake Editor
  • XML Editor

• Features
  • Highlighting, Content assist
  • Open declaration, Hover
  • Outline view
Build based on Rootstrap

- **What is Rootstrap?**
  - All required tools, libraries and include files for building a platform project
  - User creates multiple Rootstraps based on platform version and snapshot and can try to build platform module based on it.
Build based on Rootstrap

- **Rootstrap View**

  Open debug source to debug other platform module
  1. Open debug source file which installed in Rootstrap
  2. Set breakpoint

  Select a Rootstrap for building and debugging a project

  Create a new Rootstrap

  Manage packages
Build based on Rootstrap

• How to build a platform project
  1. Select a project
  2. Select a Rootstrap on Rootstrap View
  3. Click build icon
Demo (Create a Rootstrap & Build)
Easy Testing (Run & Debug)

- IDE provides a simple way to test platform module

Run

- Click
  - Transfer and install packages to device
  - Launch wizard (It’s skipped from second time)
  - Execute run command

Debug

- Click
  - Install debug packages to rootstrap
  - Execute gdb and gdbserver
  - Connect gdb to gdbserver
Easy Testing

- Launch wizard
  - Check package installation log
  - Install additional packages
  - Set additional environment variables
  - Set run command and arguments
    - In case of application, it is set automatically
    - In case of library, user set command manually
Easy Testing

• Debug process

1. Install debug packages
2. Execute `gdb`
3. Transfer and install packages
4. Execute `gdbserver`
5. Connect `gdb` to `gdbserver`
Demo (Run & Debug)
Quick Launch (RDS)

- RDS lets you develop a platform module rapidly by saving deployment time.
- After the first launching, rpm transfer and install process is skipped and only modified files are installed in the target.
Quick Launch

• **Performance**
  • It depends on number of modified files.
  • In case of Debugging, *launch time is reduced by more 74.27%*
    • It has a good effect because package installation time for Rootstrap greatly reduce.
  • In case of Running, *launch time is reduced by more 12.4%*
    • It has an issue that launch time is increased if modified files are many.
    • RDS is not operated if modified files are more than 20.
Platform Crash Analysis (Core dump debug)

- IDE provides feature to debug dump file
  - To debug core dump
    - Debug As > Tizen Platform Project – Coredump
- The needs for debugging core dump
  - Snapshot information of target image
  - Core dump files (/opt/usr/share/crash/)
    - *.tar.gz
    - *.core / *.cs
Platform Crash Analysis

- Launch wizard

  - Set coredump files
  - Set Rootstrap. Packages for debugging are installed to Rootstrap automatically
  - Check installed packages and install additional packages if it is needed
Demo (Core dump debug)
Easy Package Management (Package Manager)

- **Package Manager**
  - It provides features to manage package of Rootstrap and target.
    - Add / Remove / Modify repository
    - Install / Uninstall / Upgrade package
  - To run Package Manager
    1. Select a Rootstrap on Rootstrap View
    2. Click Manage Packages from the context menu
Easy Package Management

- Add repository
- Remove repository
- Modify repository
- Set priority of repository
- Refresh package list
- Upgrade packages
- Install remote package
- Install local package
- Uninstall package

Rootstrap & Device tab
Repository list
Package list
Command log view
Other features of Platform IDE

- Multi device management (Connection Explorer)
- Build configuration (Debug/Release)
- Tool command management (GBS/rpm/zypper/pkgcmd)
- Profiling (Oprofile/Valgrind)
- Remote development environment on Windows
Upcoming features

• Supporting Mac OS and other Linux distribution
• Build speed improvement
  • Divide into build and packaging step
  • Supporting Predictive build
• Supporting Attach debugging
Conclusion

• Tizen Platform SDK allows you to develop platform module easily and quickly

• Effective cases:
  • If you develop platform application or library
  • If you need to debug core dump
  • If you are new platform developer