EVA Release Note

For TC3182/RT65168 Chipset

(GDI Version: VIKING1)

Version: 1.0.0

Release date: 2011-09-28

Table of Contents

1 Revision History

Revision history:

Revision	Author	Date	Description
1.0.0	Serena	2011/09/28	Initial version

2 Release Note

1. This EVA release is only for TC3182/RT65168 family chipsets. To use EVA and DSP features for VoIP development, users must select the correct chipset and use the right BSP and toolchain. Please follow the release note and user guide in the chipset BSP to configure correctly for VoIP support.
2. In EVA release, it provides Makefile(s) for user's reference. Users may setup the tool-chain at the same location as in Makefile, or modify the Makefile if users would like to make a change.
3. This EVA release is based on the TC3182 VoIP EVB hardware design which uses Legerity's Le89116/Le89316 SLIC/DAA for 2x FXS + 1x FXO interfaces support. If users need only 1x FXS, he/she can use Le89116, or Le89316 only for 1x FXS + 1x FXO and follow the same hardware design and this EVA should work for backward compatible. For other hardware design change, please contact MediaTek to consult for technical support or software patch.

3 Release Contents

** EVA, Enhance VoIP Architecture, is a VoIP development framework provided by MediaTek. For more detail about EVA, please read EVA_User_Guide and other documents.

The EVA release structure is showed below. It contains GDI, ADAM and EVCOM.

Folder, File(s)	**Descriptions**
adam	Contains ADAM source code and its Makefile
bin	EVA build outputs. Contains EVAOM program, ADAM and GDI library
common	Contains exported public header files.
doc	Contains release note, user guide, and programmer reference manual.
evcom	Contains EVCOM program source code and its Makefile
gdi	Contains the GDI public header files and its library
module	Contains DSP kernel modules
Makefile	EVA Top-level makefile

4 Build EVA Components

4.1 Build all EVA components

Get into the eva source tree director, the following command build all eva components. You can change INFCNUM=1 or 2 by your hardware design.

```
$ make PLATFORM=tc3182 INFCNUM=3
```

You can get “evcom”, “libadam.a” and “libgdi_viking.a” in eva/bin folder.

*NOTE 1: TC3182 or RT65168 uses the same PLATFORM definition to build EVA components.
*NOTE 2: User should give correct interface number (INFCNUM) definition to build EVA components which reflects the real hardware platform. Giving incorrect INFCNUM may result unexpected error!

4.2 Clean all EVA components

Get into the eva source tree directory. Use the *make clean* command to clean eva components.

```
$ make clean
```

4.3 Build only GDI library

Get into the eva source tree directory, the following command build only GDI library. You can change INFCNUM=1 or 2 by your hardware design.

```
$ make PLATFORM=tc3182 INFCNUM=3 gdi
```

4.4 Build only ADAM library

Get into the eva source tree directory, the following command build only ADAM library. But make sure there is already the “libgdi_viking.a” in eva/bin folder. You can change INFCNUM=1 or 2 by your hardware design.

```
$ make PLATFORM=tc3182 INFCNUM=3 adam
```

4.5 Build only EVCOM program

Get into the eva source tree directory, the following command build only EVCOM program. But make sure there is already the “libgdi_viking.a” and “libadam.a” in eva/bin/ folder. You can change INFCNUM=1 or 2 by your hardware design.

```
$ make PLATFORM=tc3182 INFCNUM=3 evcom
```

5 Loading and Run EVCOM program

5.1 Download DSP kernel modules and EVCOM program

5.2 Put the DSP kernel modules and EVCOM program on the TFTP server.

```
# cd /tmp
# tftp –gr extern_osal_kernel_gpl.ko <tftp_server_ipAddress>
# tftp –gr osal_kernel.ko <tftp_server_ipAddress>
# tftp –gr ve_vtsp_hw.ko <tftp_server_ipAddress>
# tftp –gr ve_vtsp_rt.ko <tftp_server_ipAddress>
# tftp –gr evcom <tftp_server_ipAddress>
# chmod +x evcom
```

5.3 Insert DSP kernel modules and execute EVCOM program

Put the DSP kernel modules and evcom on the TFTP server.

```
#mkdir –p /var/tmp/osal
# insmod extern_osal_kernel_gpl.ko
# insmod osal_kernel.ko
# insmod ve_vtsp_hw.ko
# insmod ve_vtsp_rt.ko
```

After Kernel Modules are loaded, execute EVCOM program.

```
# ./evcom
```