**PJSIP-Android**

**Daily/Weekly Status Report**

**For week ending 16/01/2011**

**Project Manager:**Chris Moore, Voxygen

|  |  |  |  |
| --- | --- | --- | --- |
| Milestones | **Planned Dates** | **Actual Dates** | **Comments** |
| PJSIP 1.8.10 compilation using Android NDK R5 on OSX | 3-4 days (18/01/2011) | Delayed | In progress |
|  |  |  |  |
|  |  |  |  |

## Accomplished this week

|  |  |
| --- | --- |
| **(Task)** 1. (13/01/2011) Compiled csipsimple library on Ubuntu linux 10.1 | |
| **(Task)** 2. (13/01/2011) Compiled csipsimple library on MAC OS X using Android NDK R4B with crystax patch | |
| **(Task)** 3. (13/01/2011) Compared pjsip in csipsimple and PJSIP 1.8.10 | |
| **(Task)** 4. (13/01/2011) Attempted to patch Android NDK R5 (in progress) | |
| Notes for 13/01/2011  **(Notes) 1. The csipsimple project contains latest pjsip 1.8.10 along with some minor android specific modifications to pjsip and some extra third party codecs (silk, g729), jni and openssl. Patch created using ‘diff’ showed the same. The original PJSIP downloaded from pjsip.org can be patched to match csipsimple code and a shell script written to copy the jni, codecs and openssl directories from csipsimple project to original PJSIP if need be.**  **(Notes) 2. Patch available online to patch Android NDK R4b. A patched version of Android NDK-R4b was used to compile pjsip on OSX. PJSIP was downloaded from the csipsimple website.**  **(Notes) 3. OSX and Linux compilations of the library were smooth. Requires a case sensitive disk on OSX.**  **(Notes) 4. Trying to use the same patch on Android NDK R5. This is work is progress.** | |
| **(Task)** 5. (14/01/2011) Completed to patch Android NDK R5 | |
|  | |
| **(Task)** 6. (14/01/2011) Compiling “earm-eabi toolchain” with patched Android NDK R5 | |
| Notes for 14/01/2011  **Read on forums that the crystax patch makes csipsimple more responsive. Otherwise PJSIP should compile with standard R5. May be because C++ support with crystax patch is better. Will try that tomorrow.** | |
| **(Task)** 7. (15/01/2011) Patched Android NDK R5 tool chain has compile errors | |
|  | |
| (Task) 8. (15/01/2011) Toolchain sources downloaded and freshly compiled also showed errors. Aborted compilation for this reason for the time being. |
| (Task) 9. (15/01/2011) Compiled pjsip\_android from crsytax site using standard NDK R5. |
| (Task) 10. (15/01/2011) Installed and tested Csipsimple on Android Development Phone and was able to make calls from the android device to a sip client using a wifi connection. Details are given in the Installation instructions. |
| Notes for 15/01/2011  **The errors seen in building the patched as well as standard version of R5 are the same. They are probably related a missing binary/library. Possible to compile PJSIP using standard R5. But found that there are two missing C++ libraries in the R5 arm tool chain on OSX which are present in crystax – std c++: libstdc++ and and c++ support: libsupc++.**  **libstdc++ is present in R5 under platforms directory. I decided to take both these files from the crystax R4b build and use them with the rest of the standard R5. Details are given in the instruction document. I think it is safe to stop here for this project since the CSipSimple test went smoothly. I don’t think compiling a patched version will make a big difference at this point though we can attempt to do so if need be.** |
|  |

**For week ending 23/01/2011**

|  |
| --- |
| **(Task)** 18/01/2011   1. Waited for review comments yesterday. Today, compiled the pjsip\_android using dummy android source. Created a patch. Started patching standard PJSIP. 2. Received Unify and UI code from Voxygen |
| **(Task)** 19/01/2011   1. Completed patching standard PJSIP. 2. Created a configure-android script which makes all the required changed to PJSIP for android. 3. Compiled standard PJSIP after using configure-android. 4. Started with VOXYGEN code read thru for integration. |
| **(Task)** 20/01/2011 |
| 1. Completed VOXYGEN study. 2. Integrated patched PJSIP with Voxygen Unify code 3. Compilation with Android NDK R5 showed some linker errors. |
| **(Task)** 21/01/2011   1. Fixed linker errors and completed compilation of PJSIP-Unify with Android NDK R5. 2. The compiled PJSIP library as part of Android app was installed and tested on Android Dev Phone 2 and android emulator.   Notes  Testing on Android device showed log messages indicating a PJSIP account was being added as expected. However no message were going out and the app also did not respond to incoming calls as was expected to. Debugging is in progress.  ***GitHub Voxygen repository***  ***A private GitHub repository has been created for all Voxygen. UI and Unify code has been checked into it. I need a GitHub username to grant access.*** |
|  |
|  |
|  |
|  |

**For week ending 30/01/2011**

|  |
| --- |
| **(Task)** 24/01/2011   1. Identified problems – REGISTRATION not working, android device address not being taken. 2. Worked on getting REGISTER messages to work |
| **(Task)** 25/01/2011   1. Hardcoded IP address 2. Worked on fixing a crash seen while sending REGISTER messages |
|  |
| **(Task)** 26/01/2011   1. Fixed crash. 2. REGISTERATION still not working. |
| **(Task)** 27/01/2011   1. Fixed the via header in SIP messages 2. Changed applied patch and pj\_sockaddr\_in struct 3. REGISTRATION succeeds after that but incoming calls get disconnected immediately after connection establishment. MEDIA not seen. |
| **(Task)** 28/01/2011   1. Fixed call being disconnected immediately after connection establishment. 2. Issues remaining – 3. getaddrinfo fails, gethostbyname returns the loopback address 4. Media missing. No sound is seen at either end though RTP packets are seen 1-way |
|  |
|  |
|  |

**For week ending 06/02/2011**

|  |
| --- |
| **(Task)** 31/01/2011   1. Worked on fixing the IP address problem |
| **(Task)** 01/02/2011  Stepped out. Did not work. |
|  |
| **(Task)** 02/02/2011   1. IP address not being read was fixed. Was related to a compiler issue. A macro was being read incorrectly, 2. Started work on Media issue. |
| **(Task)** 03/02/2011  Media issue found related to lack of Alsa compatible layer which is required for Android devices. The CSipSimple app uses Alsa compatible code in its jni layer which is called when audio is initialized. Since this code is not present in our PJSIP, the media does not work. We either have to port CSipSimple jni layer to c file which contains android specific settings or we can try to simply use their code for now with some changes. Awaiting feedback. |
| **(Task)** 04/02/2011   1. Tried to port CSipSimple audio layer to Unify layer 2. Tried to port ALSA to android so it can be used by Unify later (work in progress). 3. Tried to use alsa\_dev.c along with Android’s sound library 4. Tested whether sound works using all the above 5. Checked in the same on GitHub.   (NOTES)  Audio still does not work. Currently code checked into GitHub works till call establishment but no media. Full logs have been left on.  Andoird audio is compatible with ALSA layer. Spent lot of time porting CSipSimple JNI audio layer to be used with unify. Then tried to use alsa\_dev.c and tried to port ALSA library separately for Android. This is work in progress. All required ALSA for Android code has been checked into github. |
|  |
|  |

## Issues / Risks Identified

|  |
| --- |
| **(Description)** |
| **(Description)** |
|  |