NEESit

TN-2007-01

Macintosh-Based Accelerometer/Camera Driver

Paul Hubbard [1]

[1] NEES Cyberinfrastructure Center (NEESit),
San Diego Supercomputer Center,
University of California, San Diego

Last Modified: 2007-03-16 Version: 1.0

Acknowledgment: This work was supported by the George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES) Program of the National Science Foundation under Award Number CMS-0402490. Visit http://it.nees.org/ for more information.

1 Abstract

All Macintosh laptop computers today are equipped with a 3-axis accelerometer. This built-in accelerometer helps to freeze the hard drive and save its contents if the accelerometer/processor detect a potential impending free-fall of the laptop. Through the Macintosh software framework, the accelerometer and the built-in camera can be accessed for display and recording of the dynamic accelerometer response and the camera image. Employing the NEESit software RDV, response of the accelerometer may be displayed along with the synchronized video images. The response in the form and images and accelerations may be recorded for future display and review. The data can be later archived for use in various other applications.

2 Introduction

The earthquake community often employs an external accelerometer connected to a PC data acquisition board to record and display dynamic response. Applications span the broad scope of k-12, undergraduate, and graduate education and research. Adding the capability of recording motion through a video camera is a direction that is currently receiving some attention as well.

All new Macintosh laptops currently include a built-in 3-dimensional (3D) accelerometer and a video camera. Utilizing these capabilities would allow the laptop to be employed as a recording station, in a most convenient and straightforward fashion. Recognizing the presence of the built-in accelerometer researchers from the NSF IRIS project have developed a valuable portable demo (http://www.suitable.com/tools/seismac.html) for deployment and use on any recent Macintosh laptop (SeisMac, Figure 1).

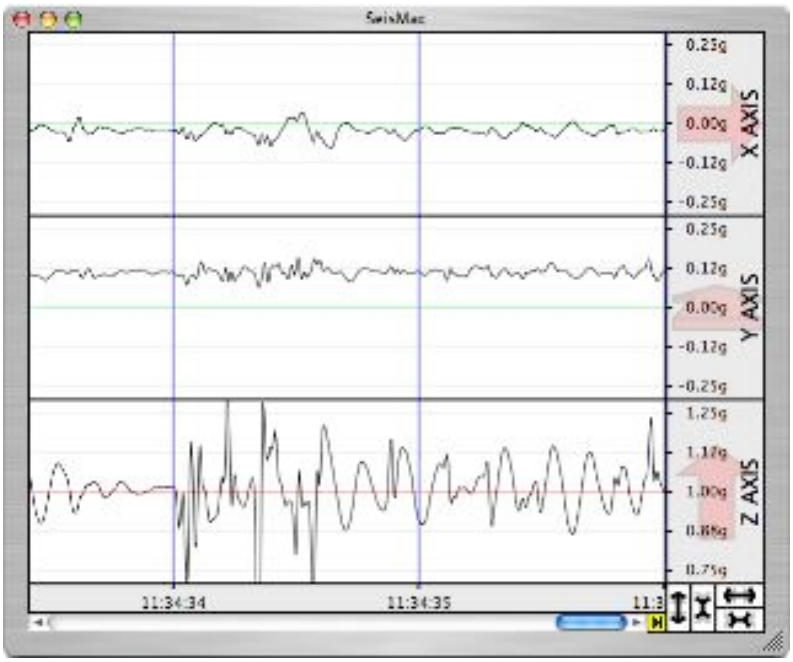

Figure 1: Screenshot from the IRIS SeisMac Display

SeisMac is an NSF-funded application, and is free to download. However, the source code is currently unavailable, and no capability is available for saving numeric data (for further analysis). As mentioned

earlier, SeisMac is based on the fact that all recent Apple laptops have a built-in accelerometer http://docs.info.apple.com/article.html?artnum=300781 known as the Sudden Motion Sensor, or SMS. Apple uses this accelerometer to detect a drop, so that the hard drive can park and prevent data loss. However, Apple does expose the SMS to programmers and SeisMac is one demonstration of that (as a side note, several newer IBM ThinkPads have the same chip, but is not available to programmers.)

Based on our intended application, a way was needed to read the SMS from Java, so that a simple program can be written that read the chip and sent the readings to the NEESit DataTurbine environment. Through the DataTurbine, this data would then become available to the full suite of telepresence tools developed by NEESit. To achieve this goal, resort was made to http://members.optusnet.com.au/lbramsay/programs/unimotion.html by Lincoln Ramsay, which puts a simple API in C on the SMS. The SMS Java Library by Daniel Shiffman's further simplified the process http://www.shiffman.net/p5/sms. On this basis, what remained to achieve a working code was to write the invocation and the DataTurbine code.

3 About the Code

Currently, the program is contained in a single Java file, sms_rbnb.java. The project in Subversion also contains the SMS libraries required to build and run it, as well as an Ant build.xml for compiling and running the code. There is also an Xcode project for working on the code. The structure of the code is fairly straightforward:

1. Initialize RBNB connection
   i) Setup append connection, with cache and archive
   ii) Register channels and metadata for same (units, MIME type)
2. Initialize SMS library
3. Loop until interrupted
   i) Read SMS values
   ii) Create timestamp from system clock
   iii) Convert data from counts into acceleration
   iv) Push data into DataTurbine
   v) Sleep 50 milliseconds (20Hz update rate)
4. Detach from the DataTurbine, so that data is saved after the test is complete

4 Caveats and Limitations

The code was written as a cognitive exercise, and will be further improved with time. It is important to note, most of the values (cache, archive, hostname, channel names, etc, etc) are hardwired, and there is no command line parser. The sample rate is also hardwired, and there is some odd behavior with RDV that has yet to be identified. The current code should be considered as proof of concept and/or demo code. It is not production code yet, but it is believed that a production version can be achieved without much difficulty. Among the potential enhancements:

1. Move it into a package, probably: org.nees.rbnb.sms
2. Parse command line for
   i) Cache and archive size
   ii) Sample rate
   iii) Calibration values (Multiplier to convert counts into acceleration)

   iv) Source name
   v) RBNB host and port
3. Fix interrupt handling
4. Test and verify

5 Discussion, Ideas and Extensions

This code offers a self-contained demo that it neat, and one can simply run the program.

Figure 2: Screenshot of RDV Displaying Live Data

Once this is running, the next thing to integrate is the Apple iSight camera built-in all recent Apple laptops (http://en.wikipedia.org/wiki/ISight). There are at least two ways to do this:

1. Use Tomcat, WebDAV and "isightcapture: to capture single frames periodically. Slow but easy to get running (http://www.intergalactic.de/hacks.html) .
2. Use Terry Weymouth's ISightSource from the dev section of NEESit telepresence (http://svn.nees.org/svn/telepresence/dataturbine-dev/turbine-dev/). This is fast, but uses a lot of CPU for unknown reasons right now.

Both approaches were tested, but NEESit developer Wei Deng took the code and rolled it up into a clean scripted demo. The combination is pretty nifty: video plus accelerometer data, all on your laptop with no wires required!

6 Access to the Source Code

The current version of the code is available at NEESforge (http://neesforge.nees.org/). If used or updated, it would be appreciated if NEESit and the original author Paul Hubbard were acknowledged.