MindSet Development Tools (MDT) Development Guide

June 28, 2009

NeuroSky®
Brain-Computer Interface Technologies

The NeuroSky product families consist of hardware and software components for simple integration of this biosensor technology into consumer and industrial end-applications. All products are designed and manufactured to meet exacting consumer specifications for quality, pricing, and feature sets. NeuroSky sets itself apart by providing building-block component solutions that offer friendly synergies with related and complementary technological solutions.

NO WARRANTIES: THE DOCUMENTATION PROVIDED IS "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, INCLUDING PATENTS, COPYRIGHTS OR OTHERWISE, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT SHALL NEUROSKY OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, COST OF REPLACEMENT GOODS OR LOSS OF OR DAMAGE TO INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE DOCUMENTATION PROVIDED, EVEN IF NEUROSKY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. , SOME OF THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES.

Contents

Chapter 1

Introduction

This development guide will direct you to the files and code samples you'll need to develop BCI-enabled applications using the NeuroSky MindSet™ for any platform, from PCs to microprocessors. The MDT offers 4 levels of interfaces for communicating with MindSet headsets. From highest level to lowest level, they are:

1. the ThinkGear Connector (TGC) (Windows and Mac OS X executables)
2. the ThinkGear Communications Driver (TGCD) (Windows, Windows Mobile, Mac OS X, and J2ME (Symbian) libraries)
3. the ThinkGear Stream Parser (source code for any C platform)
4. the MindSet Communications Protocol (specs for any platform with Bluetooth serial I/O)

The highest level interfaces supply executables and binary libraries for some of the most common platforms, like Windows and Mac OS X, while the lower level interfaces provide source code and low level communication stream specs that allow MindSet development on virtually any platform that can receive a Bluetooth serial data stream.

Please jump directly ahead to the appropriate chapter for the interface and platform you are interested in developing for, or read on to learn about all the interfaces.

Chapter 2

ThinkGear

ThinkGear is the technology inside every NeuroSky product (including the MindSet), or partner product, that enables a device to interface with the wearer's brainwaves. It includes the sensor that touches the forehead, the contact and reference points located on the ear pad and the onboard chip that processes all of the data. Both the raw brainwaves and the eSense Meters (Attention and Meditation) are calculated on the ThinkGear chip.

It is with the ThinkGear technology within the MindSet that the development tools, drivers, and APIs provided in the MDT interact with.

ThinkGear Connector (TGC)

The ThinkGear Connector (TGC) is an executable that provides a daemon-like service that manages communications with ThinkGear devices, such as the MindSet, that are connected to the computer. The TGC runs continuously in the background, and keeps an open socket on the local user's computer, allowing applications to connect to it and receive information from the connected ThinkGear devices. This means that any application in any language that can open and read from sockets (such as Flash's ActionScript3, and other scripting languages) can connect to and receive data from MindSet headsets.

The TGC is provided as an executable for the following platforms:

- Windows
- Mac OS X

For more information on the TGC, including how to run it, and how to write applications to communicate with it, please browse to the MDT's `tgc/` directory, and read the ThinkGear Connector Development Guide.

ThinkGear Communications Driver (TGCD)

The ThinkGear Communications Driver (TGCD) is a device driver with a simple API that allows communication between an Application on a computer (or mobile device) and a ThinkGear chip/-module/headset. It is available as a .dll (for x86 or ARMV4I platforms), as a .bundle (for Mac OS X platforms), or as a .java library (for J2ME/Symbian platforms).

Windows PC Development

The easiest way to begin Windows PC development is by using the provided ThinkGear Communications Driver (TGCD), for Win32 or Windows Mobile, and looking over the example program.

Files

Here are the development file(s) you'll need to add to your development environment/project. You only need to add file(s) for the language you intend to use:

Language	**Project File(s) (under `tgcd/win32/`)**
C/C++	`thinkgear.h` and `thinkgear.lib`
C#	`ThinkGear.cs`
Java (via JNI)	`ThinkGear.java`

API Documentation

Look for API documentation in the directories and files below:

Reference	**Language**	**File (under `tgcd/docs/`)**
TGCD API (doxygen)	C/C++	`html/index.html`
TGCD API (javadoc)	Java (via JNI)	`java/index.html`

The primary documentation for the TGCD API's for the C# and Java(JNI) languages in the `.cs` and `.java` wrapper files respectively, and are almost identical to the C/C++ API from which they are derived, with only possibly some cosmetic naming changes. It is recommended that you refer directly to the contents of the `.cs` or `.java` files themselves for API documentation for those languages. Otherwise, you may refer to the doxygen-generated C/C++ html documentation mentioned in the table above.

Important: In Windows, COM port names should have a `\\.\` prepended to them. It is a required prefix for addressing serial ports above `COM9`, but is optional otherwise. Read this MSDN document for more details.

Examples

Here is an example C program that uses the most important parts of the TGCD API, along with instructions for setting up and compiling the program in Visual Studio:

Example/tutorial/guide	File (under tgcd/win32/)
Sample C/C++ App	thinkgear_testapp.c
Visual Studio 2005 Guide	README.txt

The function calls relating to the TGCD API in that example are almost identical to the function calls that would need to be made in C# or Java/JNI, so please use the C example as a guide towards building your first MindSet application in those languages.

Execution

For all languages, once you have built an executable, you'll need to place the TGCD DLL (called thinkgear.dll) in the same directory as the executable (or in an appropriate place on your system PATH) in order to run the executable:

Driver	File (under tgcd/win32/)
TGCD DLL	thinkgear.dll

Once you are comfortable with the API and the example program, you'll find additional low level details, full descriptions of ThinkGear Data Types, Command Byte tables, and other advanced information in the MindSet Communications Protocol document (found in the MDT's tg/ directory).

Windows Mobile Development

Using the TGCD to interface with a MindSet on devices running Windows Mobile (WinMo) is virtually identical to Windows PC Development, except that you'll need to use a different set of files.

Files

Here are the development file(s) you'll need to add to your development environment/project. You only need to add file(s) for the language you intend to use:

Language	Project File(s) (under tgcd/winmobile/)
C/C++	thinkgear.h and thinkgear_ARMV4I.lib
C#	ThinkGear_ARMV4I.cs
Java (via JNI)	*Not supported*

API Documentation

The API for WinMo is the same as the one described in Windows PC Development. C#-specific API is documented within the .cs wrapper file itself, and is almost identical to the C/C++ API, with only possibly some cosmetic naming changes. Please refer directly to the `ThinkGear_ARMV4I.cs` file for C# API documentation.

Important: Serial COM ports on WinMo devices have a colon appended to their names, e.g. `COM1:` , `COM2:` , etc..

Also, please refer to Windows PC Development for code samples and starter projects that can be adapted to WinMo, as they are similar for WinMo (since the APIs are the same). WinMo-specific examples and tutorials are not available at this time.

Execution

Once your program is compiled and deployed, you'll also need to use this DLL, compiled for mobile ARMV4I processors. Place it in the same directory as your deployed executable, or wherever else DLLs can be found by the WinMo system `PATH`:

Driver	File (under `tgcd/winmobile/`)
TGCD DLL	`thinkgear_ARMV4I.dll`

Mac OS X Development

Here are the development file(s) you'll need to add to your development environment/project. You only need to add file(s) for the language you intend to use:

Language	Project File(s) (under `tgcd/macosx/`)
C/C++	`ThinkGear.bundle`
C#	*Not supported*
Java (via JNI)	*Not supported*

API Documentation

The API for Mac OS X is the same as the one described in Windows PC Development. In addition, refer to the How To Use The ThinkGear API In Xcode (Mac OS X) document for additional information about using the API on Mac OS X (found in `tgcd/macosx/`).

An example Cocoa/Carbon application using the TGCD API is described in the document ThinkGear API MacOSX Example (also found in `tgcd/macosx/`).

J2ME (Symbian) Mobile Development

Symbian devices supporting J2ME can use the Java libraries in the MDT's `tgcd/j2me/` directory to develop applications:

- See `tgcd/j2me/javadoc/index.html` for documentation.

ThinkGear Stream Parser

Generic Development (All Other Platforms, Including Microprocessors)

For all other platforms not covered by the TGC nor TGCD, it is up to the application to open the serial I/O communication channel (COM port or direct serial UART). Please refer to the platform's documentation for "serial I/O" or "UART" on how to open and read from such a channel, as serial I/O APIs tend to be platform-specific.

Once the communication channel is open and bytes can be read, we provide a Packet parsing library for parsing and decoding the incoming data bytes. The library is in the form of ANSI C source code (for compatibility even with most microprocessors), and can easily be ported to other languages as necessary:

(You'll find these files in the MDT's `tg/` directory)

- `ThinkGearStreamParser.h`
- `ThinkGearStreamParser.c`

Refer to the MindSet Communications Protocol (also found in the MDT's `tg/` directory) for low-level specs, instructions, and sample code.