
MDT for iOS Instruction

October 08, 2016

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 consumer thresholds 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 NEUROSKEY PRODUCT FAMILIES AND RELATED DOCUMENTATION IS PROVIDED "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 NEUROSKEY 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 NEUROSKEY PRODUCTS OR DOCUMENTATION PROVIDED, EVEN IF NEUROSKEY 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.

USAGE OF THE NEUROSKEY PRODUCTS IS SUBJECT TO AN END-USER LICENSE AGREEMENT.

"Made for iPod," "Made for iPhone," and "Made for iPad" mean that an electronic accessory has been designed to connect specifically to iPod, iPhone, or iPad, respectively, and has been certified by the developer to meet Apple performance standards. Apple is not responsible for the operation of this device or its compliance with safety and regulatory standards. Please note that the use of this accessory with iPod, iPhone, or iPad may affect wireless performance.

Contents

[Introduction to MDT](#)

[Introduction to MDT for iOS](#)

[Usage](#)

[References and Bug reports](#)

Introduction to MDT

NeuroSky's **Mind Developer Tools** (hereafter abbreviated **MDT** or **Developer Tools**) are a set of software tools that make it easy to create innovative applications that respond to a user's brainwaves and mental state.

If you already have a NeuroSky headset (such as **MindWave Mobile** and **MindWave Mobile Plus**), you will be able to take full advantage of it with our Developer Tools. If you are trying out the Developer Tools before purchasing a headset, thank you for reviewing the toolset. However, please note the NeuroSky headset is needed while using the developer tools to develop your own app. Our headset is available on Amazon store. To your convenience, here's the direct link to it

http://www.amazon.com/NeuroSky-MindWave-Mobile-BrainWave-Starter/dp/B00B8BF4EM/ref=cm_cr_pr_product_top?ie=UTF8 .

If you have any questions, let us know at support@neurosky.com.

Introduction to MDT for iOS

The **MDT for iOS** includes: **Application Standards**, **MWM Comm SDK for iOS**, **EEG Algorithm SDK for iOS** and **EULA**.

- **Application Standards** Document and Icon Images
- **EULA** End User License Agreement
- **MWM Comm SDK for iOS** is used to help connect your iOS app to a NeuroSky headset via bluetooth, and receive data from headset. It contains the follows file:
 - README.txt
 - neurosky_mwm_comm_sdk_for_ios.pdf
 - lib
 - * libMWMSDK.a
 - * MWMDDelegate.h
 - * MWMDDevice.h
 - * MWMLenum.h
 - MWMSample
 - * Base.lproj
 - * Main.storyboard
 - * Default-568h@2x.png
 - * Default.png
 - * en.lproj
 - * InfoPlist.strings
 - * HelloConnect.entitlements
 - * HelloConnectAppDelegate.h
 - * HelloConnectAppDelegate.m
 - * HelloConnectViewController.h
 - * HelloConnectViewController.m
 - * HelloMWMIOS-Info.plist
 - * HelloMWMIOS-Prefix.pch
 - * Images.xcassets
 - * libn
 - * libMWMSDK.a
 - * MWMDDelegate.h
 - * MWMDDevice.h
 - * MWMLenum.h
 - * LineGraphView.h
 - * LineGraphView.m

- * main.m
- * HelloMWMiOS.xcodeproj

- **EEG Algorithm SDK for iOS** is used to analyze and further interpret EEG data from NeuroSky's headset or TGAM module. It includes Attention, Meditation and Eye Blink Detection. These three algorithms are free to use within your application. It contains the follows file:
 - README.txt
 - eeg_algorithm_sdk_for_ios_development_guide.pdf
 - Algo SDK Sample
 - AlgoSdk.framework

If you want more information about other EEG algorithms, please contact us at support@neurosky.com.

For details, please check within each package.

Usage

Each SDK includes a sample project and documents which teach you how to use them.

In order to make the integration progress for SDKs more smooth, please review the documents of SDKs. For example, review “[neurosky_mwm_comm_sdk_for_ios.pdf](#)” to start with the MWM Comm SDK. For EEG Algorithm SDK for iOS, please review “[eeg_algorithm_sdk_for_ios_development_guide.pdf](#)”.

“[ApplicationStandards.pdf](#)” is very useful. It tells you how to use the icons to mark the status of connection in your project.

Here is a code snippet which shows how to use these SDK together:

```
// Init MWMDevice and TGBleManager shared Instance
NSArray *arr = [[NSArray alloc] initWithObjects:@"MindWave Mobile", nil];
[[TGBleManager sharedTGBleManager:DeviceType_MWM] candidateScan:arr];
[[MWMDevice sharedInstance] setDelegate:self];
[[TGBleManager sharedTGBleManager:DeviceType_MWM] setDelegate:self];

// Scan MWM Device
[[MWMDevice sharedInstance] scanDevice];

// MWM Comm SDK return the found Headsets
-(void)deviceFound:(NSString *)devName MfgID:(NSString *)mfgID DeviceID:(NSString *)deviceID
{
    //mfgID is null or @"", NULL
    if ([mfgID isEqualToString:@""] || nil == mfgID || NULL == mfgID) {
        return;
    }

    dispatch_async(dispatch_get_main_queue(), ^{ // do all alerts on the main thread

        if (![devicesArray containsObject:deviceID])
        {
            [tempDevicesArray addObject:deviceID];
            devicesArray = tempDevicesArray;
            [devNameArray addObject:devName];
            [mfgIDArray addObject:mfgID];
            //store
            [deviceTypeArray addObject:@0];
            [devicePicker reloadData];
        }

        if (devicesArray.count > 0) {
            devicePicker.userInteractionEnabled = YES;
        }
        else{
```

```

        devicePicker.userInteractionEnabled = NO;
    }
    });
}

// Connect Headset by deviceId
NSString *deviceId = [devicesArray objectAtIndex:row_number];
[[MWMDevice sharedInstance] connectDevice:deviceId];

// Disconnect Headset
[[MWMDevice sharedInstance] disconnectDevice];

// Configure Headset Notch Filter
[[MWMDevice sharedInstance] writeConfig:TGMWMConfigCMD_ChangeNotchTo_50];
[[MWMDevice sharedInstance] readConfig];

// Get data from NeuroSky Headset
-(void)eegSample:(int)sample
{
    [ekgLineView addValue:sample];
};

-(void)eegPowerLowBeta:(int)lowBeta HighBeta:(int)highBeta LowGamma:(int)lowGamma
MidGamma:(int)midGamma
{
    NSLog(@"%s >>>>>>-----eegPower: lowBeta:%d highBeta:%d lowGamma:%d midGamma:%d",
    __func__, lowBeta, highBeta, lowGamma, midGamma);
}

-(void)eegPowerDelta:(int)delta Theta:(int)theta LowAlpha:(int)lowAlpha HighAlpha:(int)highAlpha
{
    NSLog(@"%s >>>>>>-----eegPower: delta:%d theta:%d lowAlpha:%d hightAlpha:%d", __func__, delta,
    theta, lowAlpha, highAlpha);
}

-(void)eSense:(int)poorSignal Attention:(int)attention Meditation:(int)meditation
{
    NSLog(@"%s >>>>>>-----eSense:%d Attention:%d Meditation:%d", __func__, poorSignal, attention,
    meditation);
}

-(void)mwmBaudRate:(int)baudRate NotchFilter:(int)notchFilter
{
    NSLog(@"%s >>>>>>-----mwmBaudRate:%d NotchFilter:%d ", __func__, baudRate, notchFilter);
}

```

[MWMDevice sharedInstance] is used to get information returned by MWM Comm SDK, and eegSample, eegPowerLowBeta, eegPowerDelta, eSense, mwmBaudRate delegate functions are used to receive data. You can pass the data to EEG SDK here.

References and Bug reports

You can get the latest developer information from here:

<http://developer.neurosky.com/>

Learn about NeuroSky's EEG Data Types here:

http://developer.neurosky.com/docs/doku.php?id=thinkgear_communications_protocol

You may find some additional useful information in the Knowledge Base:

<http://support.neurosky.com/kb>

If you find any bugs, please contact us at:

support@neurosky.com