ThinkGear 开发指南(安卓版)

2013年4月5号



神念科技e NeuroSky®家族产品包括硬件和软件,这些产品 能将生物传感器简单的融合进消费者和工业的终端应用程序 中。所有产品的设计和制造目的都是为了满足消费者对产品质 量、价格和功能的各项需求。神念科技能脱颖而出的关键在于 能够不断构建基块组件解决方案,同时这些方案能发挥与之相 关并技术互补的良好协同作用。

免责声明:神念科技公司系列产品及相关文件资料并不 包含任何明示或暗示的保证,授权或许可。包括有关适 销性,知识产权(包括专利,版权或其他)或特殊目的的 适用性的保证。在任何情况下,神念科技公司或其供货 商均不会就使用或无法使用神念科技公司的产品或资料 而产生的任何损害(包括但不限於利润损失,业务中断, 货物替换成本或信息丢失或损坏)承担责任,无论神念科 技公司是否已被告知可能出现上述损害。由於某些司法 管辖区域明确禁止免除或限制对后果性损害或附带性损 害应承担的责任,上述限制中有些可能对您并不适用。

神念科技公司产品的使用应遵守最终用户许可协议中的 规定。。

"为ipod制造"、"为iphone制造"、"为ipad制造"表示一个电子 配件已经被设计成用来分别连接到ipod、iphone、ipad,并 且已经由开发人员来满足苹果的性能标准认证。苹果公司 对此设备或其遵守的安全操作或者监管标准将不负任何责 任。请注意,使用此配件ipod、iphone、ipad可能会影响到 无线性能。

目录

简介	5
THINKGEAR SDK (安卓版) 内容	
支持的 THINKGEAR 硬件	5
你的第二个工程项目.	-
你的第一个工作项目:	/
开发你自己的启用 THINKGEAR 的安卓应用	8
准备你的安卓丁程	9
	c
的建 IGDEVICE 对象	····· c
按文朴处理数据信息	9
TGDevice 消息类型 (msg.what)	11
TGDevice 状态	14
TGDevice 配置问题	14
使用 TGDEvice 对象	14
THINKGEAR 数据类型	16
通用型	16
POOR_SIGNAL/SENSOR_STATUS	16
RAW_DATA	17
RAW_MULTI	17
EEG	17
ATTENTION 专注度	17
MEDITATION 冥想度	18
ZONE 区域	18
BLINK 眨眼	19
EEG_POWER	19
THINKCAP_RAW	20
POSITIVITY 积极性	20
FAMILIARITY 熟练度	20
ECG/EKG 心电	23
HEART RATE 心率	23
 Smoothed Heart Rate 平滑的心率	23
Heart Rate Acceleration 心跳加速	24
Target Heart Rate for Physical Training 体能训练的目标心率	24
Heart Fitness Level 健康心脏水平	25
RELAXATION 放松度	26

RESPIRATION 呼吸率	26
Heart Risk Awareness 心脏病风险意识	27
HEART_AGE 心脏年龄	29
Personalization 个性化心电识别	30
EKG_RRINT 心电的 RR 间隔	32
恰当的应用设计	
故障排除	
重要声明	35
附录 A: 其他参考文献	
附录 B: UART (非蓝牙) 连接	
附求 C: 后用个性化的使用	
	20
門水 D: ル に ラヘ女早く	



该指南将指导你怎样使用神念科技的ThinkGear SDK(安卓版)来写一个能够使用从神念 科技Thinkgear系列的生物传感器(CardioChip系列产品)获取的生物信号数据的应用程序。这 将使你的安卓应用能够获取和使用生物信号数据,比如从神念科技传感器获取心电图和脑电 图。

本指南仅提供给已经熟练使用Eclipse和Android SDK的程序员使用。如果你不熟悉安卓开发, 请先访问<u>http://developer.android.com</u> 来学习如何建立eclipse和android SDK 开发环境和创建安卓 应用程序。

如果你已经熟悉创建一个安卓应用程序,那接下来的步骤就是确保你已经下载了神念科技的安 卓版ThinkGear SDK。如果你已经在读这篇文档,说明你已经有这个SDK了。如果没有,这个SDK 可以从<u>http://store.neurosky.com/products/developer-tools-3-android</u>下载。

ThinkGear SDK (安卓版) 内容

- ThinkGear SDK (安卓版):开发指南(本文档)
- ThinkGear SDK (安卓版): API 参考
- ThinkGearBase.jar 包
- ThinkGearPackX.jar 包
- •示例 ThinkGear (安卓版)项目

你能发现"API参考"在reference/TG-SDK文件夹下,"thinkGearBase.jar"和"thinkGearPackX.jar" 在lib文件夹下,"HelloEEG","HelloEKG"或者其他示例工程在Sample Projects文件夹下

重要提醒:如果你有测试版本或者其他特殊版本的jar包文件夹,这个文件夹可能替代 TgLabsbase.jar和TgLABsPackX.jar。如果这个jar文件名字中有Pack,那个接下来的字符表示一个 选项包。例如TgLABsPackX.jar或者ThinkGearPackB.jar。

支持的ThinkGear 硬件

ThinkGear SDK (安卓版) 必须和与之兼容的ThinkGear硬件传感设备一起使用。 以下是目前兼容的ThinkGear硬件传感设备:

- MindWave (RF)
- MindWave Mobile



- MindBand
- MindTune
- MindSet
- ThinkCap (*一些更老的型号已不再支持)
- BrainAthlete
- CardioChip Starter Kit Unit

重要提醒: 请确保在使用任何TG-SDK的安卓应用前已经将ThinkGear传感器配对到你的安卓 设备。你可以根据ThinkGear硬件设备中附带的用户手册指导配对。

你的第一个工程项目:

你的SDK可能包含了一个简单工程包括如何设置,连接和处理ThinkGear设备的数据。你可以通过以下步骤添加一个工程到Eclipse IDE。(已测试的Eclipse IDE版本:Mac OS 10.8.2平台的Juno Service Release 1)

- 1. 打开 Eclipse, 选择 File > New > Project...
- 在新建工程的向导中,点开Android选项,并且选择Android Project from Existing Code, 并且单击 Next。
- 3. 点击 Browse... 找到Sample Projects文件下的 Hello 工程并点击Open
- 4. 勾选 Copy projects to workspace 并点击 Finish
- 5. 退出向导
- 6. 用你的安卓设备连接到你的电脑

7. 此时,你应该能够浏览你的代码,做修改,编译和部署任何常规安卓应用程序一样,部署你的应用程序到你的设备或者模拟器上。

注意: 如果出现问题, 可尝试以下的操作:

- 在工程上右击并且选择Properties。单击Android选项确保安装Android2.3.3或以上版本。
- 在尝试连接之前。确保你的安卓设备已通过你的安卓设备的蓝牙设置与ThinkGear硬件 设备配对了。
- 在Eclipse的Package Explorer中,展开你的工程下的libs文件夹,右击ThinkGearBase.jar, 并且选择Build Path > Add to Build Path,并且添加这个文件夹下的ThinkGearPackX.jar 文件。(或者TgLABsBase.jar和TgLABsPackX.jar)
- 为确保应用运行在正确的设备上,检查Eclipse,点击Run> Run Configurations. 在 Target 标签页,勾选"Always prompt to pick device".
- 查看一下Eclipse 控制台,有时会显示需要重启,此时应退出再重新打开Eclipse。
- 如果你看到"The selection cannot be launched, and there are no recent launches."的消息。右击 Package 或 Project Explorer的project,选择"Run As"并勾选"Android Application" 然后尝试 重新运行一遍。

你可以使用相同的步骤部署其他的示例工程。

开发你自己的启用ThinkGear的安卓应用

准备你的安卓工程

- 首先确保你工程的"Android build target"至少是Android 2.3.3.版本。
 - 1. 右击你的工程并选择Properties。
 - 2. 点击Andorid 标签确保Target Name至少为Android 2.3.3。
- 接着你需要添加 ThinkGearBase.jar和ThinkGearPackX.jar到你的工程中去。
 - 1. 使用Eclipse的Package Explorer在你的安卓项目的根目录中创建一个lib文件夹。
 - 2. 使用Windows Explorer (Mac的Finder)把ThinkGear jar从TG-SDK for Android的lib文件夹拷贝到你Eclipse工程的lib文件夹中。
 - 3. 打开Eclipse Package Explorer, 右击ThinkGear-Base.jar (Mac用户使用Ctrl-click),

选择Build Path » Add to build path. 同时添加ThinkGearPackX.jar

• Assuming your app will be connecting to the inkGear hardware device via bluetooth, you will need to enable the BLUETOOTH permission in your app's manifest ële:

● 假设你的app会通过蓝牙连接ThinkGear硬件设备,你需要在app的manifest文件中启用 BLUETOOTH权限。

<manifest ... >

<uses-permission android:name="android.permission.BLUETOOTH" />

</manifest>

创建TGDevice对象

TGDevice对象用来管理单个ThinkGear硬件设备的连接问题的。本书仅涉及了最普遍的连接场景,即你的安卓app会通过TGDevice(BluetoothAdapter, Handler)的构造的标准安卓蓝牙来连接 ThinkGear硬件设备。(替换构造函数(TGDevice(InputStream, outputStream, Handler)只能 在特殊不常见的情况下使用)。

•在Activity中导入以下的包:

import com.neurosky.thinkgear.*;

import com.android.bluetooth.*; import com.android.util.Log; •在你的Actity中声明TGDevice对象和 BluetoothAdapter 对象:

```
public class HelloEEGActivity extends Activity {
   11
   TGDevice
                   tgDevice = null;
   BluetoothAdapter btAdapter = null;
   //...
  • 在onCreate() 方法中初始化 btAdapter 和 tgDevice :
   public void onCreate(Bundle savedInstanceState) {
       //...
       btAdapter = BluetoothAdapter.getDefaultAdapter();
       if( btAdapter != null ) {
           tgDevice = new TGDevice( btAdapter, handler );
       3
       //...
   }
注意:我们将会在下节中讨论handler对象。
```

接受和处理数据信息

TGDevice会同通过消息发送到Handler对象来与你的app做交流。在你的app中创建一个 Handler对象来处理传入的数据。

```
private final Handler handler = new Handler() {
    @Override
    public void handleMessage( Message msg ) {
        switch( msg.what ) {
            case TGDevice.MSG MODEL IDENTIFIED:
                // connection has been established and //
                the device identified.
                // now configure any special options
                if (setBlinkDetectionEnabled(true)) {
                     // return true, means success
                     Log.v( "HelloEEG", "BlinkDetection is Enabled");
                }
                else {
                     // return false, meaning not supported because:
                     // + connected hardware doesn't support
                     //
                         + conflict with another option already set
                     // + not support by this version of the SDK
                     Log.v( "HelloEEG", "BlinkDetection can not be Enabled");
                }
                break;
            case TGDevice.MSG_STATE_CHANGE:
                switch( msg.arg1 ) {
                    case TGDevice.STATE_IDLE:
```

break; case TGDevice.STATE_ERR_BT_OFF: break; case TGDevice.STATE_CONNECTING: break; case TGDevice.STATE_ERR_NO_DEVICE: break; case TGDevice.STATE_NOT_FOUND: break; case TGDevice.STATE_CONNECTED: tgDevice.start(); break; case TGDevice.STATE DISCONNECTED: break; default: break; } /* end switch on state change type */ break; case TGDevice.MSG_POOR_SIGNAL: Log.v("HelloEEG", "PoorSignal: " + msg.arg1); break; case TGDevice.MSG_ATTENTION: Log.v("HelloEEG", "Attention: " + msg.arg1); break; case TGDevice.MSG_RAW_DATA: int rawValue = msg.arg1; break; case TGDevice.MSG EEG POWER: TGEegPower ep = (TGEegPower)msg.obj; Log.v("HelloEEG", "Delta: " + ep.delta); break; case TGDevice.MSG_ERR_CFG_OVERRIDE: Log.v("HelloEEG", "Feature Not Allowed: " + msg.arg1); break; case TGDevice.MSG_ERR_NOT_PROVISIONED: Log.v("HelloEEG", "Feature Not Included: " + msg.arg1); break; default: break; } /* end switch on message type */ } /* end handleMessage() */

}; /* end Handler */

每个消息的类型由 msg.what的值来决定, 而实际的数据值由msg.arg1 (简单的数值) 或 msg.obj (复杂的数值)来决定。

TGDevice 消息类型 (msg.what)

Msg.what 措	j述 数	据
MSG_MODEL_IDENTIFIED	TGDevice 已经连接	连接已经被建立,数据 包正在被接收,硬件的 类型已经被识别。产品 已经被激活。
MSG_ERR_CFG_OVERRIDE	用户配置已经被重写	与己连接的硬件和软件 不兼容的用户配置已经 被重写,命令在消息的 arg1中(见下面表格: TGDevice配置问题)
MSG_ERR_NOT_PROVISIONED	尝试配置不在释放 (release)中的特征。	尝试配置不在释放 (release)之中的特 征。命令在消息的arg1 中(见下面表格的 TGDevice配置问题)
MSG_STATE_CHANGE	TGDevice 的状态发生改 变	状态改变的信息被存储 在消息的arg1中(见下 面表格的TGDevice配置 问题)
MSG_POOR_SIGNAL	生物信号强度/状态	生物信号的强度或状态 存储在消息的arg1中
MSG_RAW_DATA	Raw样本数值	单通道raw的样本数值 (大多数的ThinkGear 设备)以int形式被存储 在消息的arg1中
MSG_RAW_MULTI	多通道raw的数值	多通道raw的样品数值 (只适用某个 ThinkGear设备)以 TGRawMulti对象形式存 储在消息的obj中
MSG_RAW_MULTI_NEW	多通道raw的数值	多通道raw的样品数值 (只适用某个 ThinkGear设备)以 TGRawMulti对象形式被 存储在消息的obj中

Msg.what	描述	数据
MSG_ATTENTION	专注度等级	用户的专注度等级存储 在消息的arg1中
MSG_MEDITATION	冥想度等级	用户的冥想度等级存储 在消息的arg1中
MSG_ZONE	区域等级	用户的区域等级被存储 在消息的arg1中
MSG_BLINK	被检测到的眨眼强度	用户的眨眼强度被存储 在消息的 arg1 中
MSG_EEG_POWER	脑电波	用 户 的 脑 电 波 以 TGEegPower 对象形式被 存储在消息的 obj 中
MSG_THINKCAP_RAW	多通道 raw 的数值	 多通道 raw 的样品数值 (只 适 用 某 个 ThinkGear 设备)以 TGRawMulti对象形式被 存储在消息的 obj 中
MSG_POSITIVITY	-100.0 to 100.0	用户的情感或者情绪 值。正数越多表示用户 处在或者靠近积极的情 绪中。负数越多表示用 户处在或者靠近消极的 情绪中。
MSG_FAMILIARITY	浮点	能被用来比较用户最新 学习到的技能熟悉程 度。一分钟收集到的数 据可以做一次试验,并 且产生一个熟悉度索引 值。这个被区分开来的 熟悉度索引值可以做试 验间的比较。
MSG_DIFFICULTY	浮点	能被用来比较用户最新 学习到的技能难度级 别。一分钟收集到的数 据可以做一次试验,并 且产生一个难度索引 值。这个被区分开来的 难度索引值可以做试验 间的比较。

Msg.what	描述	数据
MSG_HEART_RATE	心率	心率值以 int 形式存储
		在消息的 arg1 中
MSG_EKG_RRINT	两个波峰之间的间隔	在一毫秒之间的两个波
		峰之间的间隔以 int 形
		式存储在消息的 arg1
		中。
MSG_RELAXATION	放松度等级	用户的放松度等级以
		int 形式存储在消息的
		arg1 中
MSG_RESPIRATION	呼吸率	用户的呼吸率以 Float
		对象存储在消息的 obj
		中
MSG_HEART_AGE	心脏年龄	用户的心脏年龄以 int
		形式存储在消息的 arg1
		中
MSG_HEART_AGE_5MIN	心脏年龄	用户的心脏年龄以 int
		形式存储在消息的 arg1
		中
MSG_EKG_TRAIN_STEP	训练步骤	用户的训练步骤以 int
		形式存储在消息的 arg1
		中
MSG_EKG_TRAINED	训练结束	不返回任何值
MSG_EKG_IDENTIFIED	被识别结果	识别结果以 string 形式
		存储在消息的 arg1 中

注意:根据你TGDevice连接到的ThinkGear硬件设备的类型,可能由于数据类型不适用于 ThinkGear硬件设备造成上述列表里的一些数据将不会发送到你的应用程序中的Handler中。查 看每个ThinkGear硬件设备的开发说明书详细可以得知你的应用程序是否适用。

重要:特别说一下MSG_STATE_CHANGE 这个消息(表示TGDevice当前正在做的事), MSG_POOR_SIGNAL 这个消息(表示ThinkGear硬件设备当前的状态)。例如,如果 MSG_POOR_SIGNAL表示ThinkGear硬件设备在此刻没有被用户使用(ThinkGear脑电设备为 200,或者在心电上为0)。那么当你应用的Handler传入的任何一个raw数据或者脑电强度时, 应该适当的忽略这些值。

更多有关专注度, 冥想度或放松度等其他数据类型的详细信息, 请参考下面的 ThinkGear数据类型章节。

一些特定设备可以额外参考其ThinkGear硬件设备的参考规范和开发注释信息。

TGDevice 状态

状态	描述
STATE_IDLE	TGDevice 最初的状态。在 stop()方法之后没有连
	接任何 ThinkGear 硬件设备
STATE_ERR_BT_OFF	在 TGDevice() 构造函数或者 connect() 方法时,
	安卓设备的蓝牙适配器没有打开或者不可用
STATE_CONNECTING	在运行 connect () 方法时,努力去连接一个
	ThinkGear 硬件设备
STATE_ERR_NO_DEVICE	没有蓝牙设备配对到这个安卓设备
STATE_NOT_FOUND	没有任何一个 ThinkGear 硬件设备配对到这个安卓
	设备
STATE_CONNECTED	数据流已经被打开
STATE_DISCONNECTED	在运行 close()方法之后,连接模块丢失

TGDevice配置问题

 状态 描	述
ERR_MSG_BLINK_DETECT	不支持任何软硬件,不允许心电设备
ERR_MSG_TASKFAMILIARITY	不被任何软硬件支持,不支持心电设备或者
	Positivity 是 enabled 的时候
ERR_MSG_TASKDIFFICULTY	不被任何软硬件支持,不支持心电设备或者
	Positivity 是 enabled 的时候
ERR_MSG_POSITIVITY	不被任何软硬件支持,不支持心电设备或者
	Familiarity 或者 Difficulty 是 enabled 的时候
ERR_MSG_RESPIRATIONRATE	不被任何软硬件支持,不支持脑电设备

使用TGDevice对象

当Handler 对象准备接受数据并且TGDevice对象已创建时,你的app便可以命令TGDevice对象去尝试连接你的ThinkGear硬件设备,同时开始处理接受到的数据了。

● 调用tgDevice的connect()方法启动连接过程。tgDevice会搜索安卓设备中所有配对 的蓝牙设备列表,并且去尝试连接第一台可兼容的ThinkGear设备(查看支持的设备):

tgDevice.connect(true);

注意: 设置connect()方法的rawEnabled属性为true,允许raw sample数据被发送到你的应用程序。设置false将会阻止raw数据被解析,也将不再把数据发送到你的应用程序,这样 能避免Handler解析并保存太多可能用不到的raw sample,从而减少CPU的占用率。

 成功通过蓝牙发现并连接到ThinkGear硬件设备后,tgDevice会发送一个 STATE_CONNECTED的消息给Handler。要根据这条消息开始接受数据,需要调用 tgDevice的start()方法:

tgDevice.start();

● 当你的应用程序不再需要从TGDevice接受数据时,可以通过close()放大关闭连接:

tgDevice.close();

这时你的应用程序现在应该能够连接到ThinkGear硬件设别并已接受数据了!要了解更多关于TGDevice方面的细节,你可以随时打开Thinkgear SDK for Android的reference文件夹参考Thinkgear SDK for Android: API docs的内容。

或者阅读下一章节的内容,你可以得知你的应用程序是否确实在接受一些Thinkgear的数据。

ThinkGear数据类型

ThinkGear的数据类型可以大致分成三个部分:只适用于EEG的传感器设备的数据类型,只 适用于ECG/EKG(CardioChip)的传感器设备的数据类型,适用于所有启用ThinkGear的设 备的数据类型,包括EEG和ECG/EKG。

通用型

这些数据类型可以适用于大部分的ThinkGear硬件设备。

POOR_SIGNAL/SENSOR_STATUS

这些整型值是用来反应传感器芯片上生物信号强弱的。该值通常由ThinkGear硬件设备每秒 输出一次。

这个值非常重要,任何应用程序都需要它来读懂,理解,处理ThinkGear传感器的数据。想 据你的应用和用户的使用情况,你的应用程序是否需要进行调整取决于当前的POOR_SIGNAL / SIGNAL_STATUS值。比如,假设这个值表明生物传感器还没接触到物体,那么这段时间接收到 的RAW_DATA或EEG_POWER应该被算作浮点噪音,而不是从人体获取的数据,基于应用的 需求这些数据应被舍弃。这个也可以用来让用户调节传感器或第一次使用设备时的基准。

由于不同类型传感器的解释不同 ,为了更好诠释这个值你必须首先确定好你所使用的神念电子 传感器的类型。

对于EEC的传感器设备:值为0表示生物传感器接受到的信号很好没有任何明显问题。而1~199 中得到越高的值表示生物传感器检测到信号问题就越多。值200表示传感器可能甚至没有接触 到导电体(例如,EEG耳机还没有被穿戴好的时候)。

对于ECG/EKG (CardioChip) 传感器设备:值200表示目前已良好接触到导电体(如用户的皮肤) 并接收数据,而值0则正好相反:它表示目前还没能接触导电体并接收数据。

Poor signal 会由很多因素引起. 按照其影响由强到弱排序, 分别是:

•传感器、地面或参考电极不在人体的头上或身体上(即当用户没有穿戴ThinkGear设备)。

•传感器、地面或参考电极与人的皮肤接触不良。(如头发的遮挡,耳机设计与大脑不能贴合, 耳机没有正确佩戴等)。

- •佩戴者过多的行为 (如过多的摇晃大脑、身体,碰撞耳机或传感器).
- 过度的环境静电噪音 (一些环境有很强的电信号或是人穿戴传感器时有静电积聚).
- 过度的生物噪声(如不需要的EMG, EKG/ECG, EOG, EEG等)

对于EEG来说, ThinkGear传感器的正常使用下有一些噪音干扰是不可避免的,为此神念科技 设计了过滤技术和算法来检测,纠正,补偿,解释,承担各种类型的信号噪声。大多数用户 都是只对eSence的值有兴趣,比如专注度和冥想度的值。不用太担心POOR_SIGNAL的影响, 只要注意POOR_SIGNAL的值为0时就能取到专注度和冥想度的值,并且没带耳机的 POOR_SIGNAL是200。POOR_SIGNAL的值针对一些对噪声更敏感的应用(比如一些医学或研 究的应用程序)或者需要很快检测出轻微噪音的应用更有帮助。

RAW_DATA

这个数据类型提供了生物传感器的原始样本值。采样率(同时产生的输出率),大概的范围值 和这些值的解释(从原始单位转换为伏特单位)依赖于ThinkGear硬件设备执行抽样的硬件特 点。你必须参考ThinkGear硬件的各种类型的技术文档了解细节来支持你的应用程序。

例如,大多数ThinkGear设备样本频率值设为512Hz,具体范围可以从-32768到32767。

再比如,要把基于TGAT 的EEG传感器的值(如TGAT, TGAM, MindWave Mobile, MindWave, MindSet)转换为伏特值,要这样做:

(rawValue * (1.8/4096) / 2000

注意:CardioChip或者基于BMD10X的设备的 ECG/EKG元数据必须使用不同的转换方式。

RAW MULTI

这个数据类型不被目前任何正在支持的ThinkGear产品所使用。将其保存 这里的目的是兼容一些已淘汰的产品,也能看作是未来产品可能需要的一个占位符。

EEG

这些数据类型只适用于EEG传感器硬件设备,例如 MindWave Mobile, MindSet, MindBand和 TGAM 芯片、模块。s

ATTENTION专注度

"eSense专注度指数"表明了使用者精神"集中度"水平或"注意度"水平的强烈程度,例如,当你能 够进入高度专注状态并且可以稳定地控制你的心理活动,该指数的值就会很高。该指数值的 范围是0到100。心烦意乱、精神恍惚、注意力不集中以及焦虑等精神状态都将降低专注度指数 的数值。请参见eSense指数 — 概述章节内容了解eSense指数的详细说明。 默认情况下会启用该数值的 输出,通常每秒输出一次。

MEDITATION冥想度

"eSense冥想度指数"表明了使用者精神"平静度"水平或者"放松度"水平。该指数值 的范围是0到100。需要注意的是,放松度指数反映的是使用者的精神状态,而不是其身 体状态,所以,简单地进行全身肌肉放松并不能快速地提高放松度水平。然而,对大多 数人来说,在正常的环境下,进行身体放松通常有助于精神状态的放松。放松度水平的 提高与大脑活动的减少有明显的关联。长期观察结果表明:闭上眼睛可以使得大脑无需 处理通过眼睛看到的景象从而降低大脑精神活动水平。所以,闭上眼睛通常是提高放松 度值的有效方法。心烦意乱、精神恍惚、焦虑、激动不安等精神状态以及感官刺激等都 将降低放松度指数的数值。请参见"eSense指数 — 概述"章节内容了解eSense指数的详细 说明。

默认情况下会启用该数值的 输出,通常每秒输出一次。

eSense™ 指数

对所有不同类型的eSense(如专注度和冥想度)其指数以1到100之间的具体数值来指示。数值在40 和60之间表示此刻该项指数的值处于一般范围,这一数值范围类似于常规脑电波测量技术中确 定的"基线"。(但是ThinkGear的基线测定方法是自有的专利技术,与常规脑电波的基线测定办 法不同)。数值在60—80之间表示此刻该项指数的值处于"较高值区",也就是说略高于正常水 平(即当前情况下你的专注度或者是放松度比正常情况下要高)。数值在80—100之间表示处 于"高值区"。它表示你的专注度或放松度达到了非常高的水平,即处于非常专注的状态或者是 非常放松的状态。

同理,如果数值在20—40之间则表示此时的eSense指数水平处于"较低值区",数值在1—20则意 味着处于"低值区"。与前述其它区值所代表的人的精神状态相反,eSense指数处于这2个区域则 表示被试者的精神状态表现为不同程度的心烦意乱、焦躁不安、行为反常等。

ZONE 区域

这个值记录了当前目标的表现区域,取值范围从0到3,而且这个值只有当目标从一个区域过 渡到另一个区域时才会传送。

该算法使用专注度和冥想度指导目标的最佳性能。

要进入精英Zone (3),目标必须保持他们的注意力水平至少值82,同时持有冥想水平稳定或 增加。

要进入中等 Zone (2), 目标必须保持他们的注意力水平至少值67,同时持有冥想水平稳定或增加。

要进入初级Zone(1), 目标必须保持他们的注意力水平至少值53,同时持有冥想水平稳定或增加。

还未就绪状态 Zone (0) 是指所有注意力水平低于53并且目标的冥想的初级Zone水平也在下 降。.

2013年4月5日 | © 2012 NeuroSky, Inc. 版权所有.

如果传感器没能和人体有效接触,所有Zone的计算会暂停,其数值也会重置。

注意:相比其他NeuroSky的产品,这是不同的性能实现Zone。 参考: Golf Putting Training Algorithm v 2.0, 2012年9月, Dr. KooHyoung Lee.

BLINK眨眼

这个整型值记录了用户最近眨眼的强度。取值范围从1到255,每次检测到眨眼变会记录数据。 这个值表示相对眨眼强度没有单位。

必须启用眨眼检测。

```
if (setBlinkDetectionEnabled(true)) {
     // return true, means success
     Log.v( "HelloEEG", "BlinkDetection is Enabled");
}
else {
     // return false, meaning not supported because:
     // + connected hardware doesn't support
     //
        + conflict with another option already set
     //
        + not support by this version of the SDK
     Log.v( "HelloEEG", "BlinkDetection can not be Enabled");
}
当前配置可以恢复。
if (getBlinkDetectionEnabled()) {
     // return true, means it is enabled
     Log.v( "HelloEEG", "BlinkDetection is configured");
}
else {
     // return false, meaning not currently configured
     Log.v( "HelloEEG", "BlinkDetection is NOT configured");
}
```

注意:如果这些方法在MSG_MODEL_IDENTIFIED恢复之前被调用,则它被认为是一个鉴定 连接设备时需要处理的请求。很有可能这个特性被启用后突然发现它不再启用了。一旦连接 设备被鉴定,如果请求和软硬件不兼容,他就会被重写,并且发送MSG_ERR_CFG_OVERRIDE 作为提醒。

EEG_POWER

这个数值代表当前普遍认可的8个量级的EEG频率带。

这8个EEG频率段是: delta (0.5 - 2.75Hz), theta (3.5 - 6.75Hz), low-alpha (7.5 - 9.25Hz), high-alpha (10 - 11.75Hz), low-beta (13 - 16.75Hz), high-beta (18 - 29.75Hz), low-gamma (31 -

第四章 - ThinkGear数据类型

39.75Hz), 和 mid-gamma (41 - 49.75Hz). 这些值没有单位,只是用来和其他频率段的样品值作对比。

默认情况下,启用此数据值的输出,输出大约每秒一次。

THINKCAP_RAW

这个数据类型不被目前任何正在支持的ThinkGear产品所使用。将其保存 这里的目的是兼容一些已淘汰的产品,也能看作是未来产品可能需要的一个占位符。

POSITIVITY积极性

取值范围从-100到+100,表明Values -100 to +100,表明目标对某事的关注度,负值越多说明目 标关注度越低,正值越高说明越关注。

注意:目前这个功能还不可用。

FAMILIARITY 熟练度

该算法试图表达目标对某种运动技能的熟练程度,它可以和一些复杂算法结合,研究已掌握的 运动技能的不同方面。但它也可以单独使用。

它可以用来比较一个测试对象学习一个新技能的熟练度。一分钟的收集的数据构成一次试验, 并且会产生一个针对此人的熟练度索引值。运动技能单独试验的熟练度索引可以用来作为同 一个人的比较。

熟练度索引值是浮点数,没有单位,只是用来对比来自同一个人的其他数据。有利于展示出 从基线 (或过去的试验)到当前试验的变化百分比。注意不同值可以表示正面也可以表示负面 的。观察HelloEEG示例应用程序是如何应用这些信息的。

熟练度任务的计算必须已经被启用。

if (setTaskFamiliarityEnable(true)) {
 // 返回 true, 意味成功
 Log.v("HelloEEG", "TaskFamiliarity is Enabled");
}
else {
 // return false, meaning not supported because:
 // + connected hardware doesn't support
 // + conflict with another option already set
 // + not support by this version of the SDK
 Log.v("HelloEEG", "TaskFamiliarity can not be Enabled");
}
当前配置可以被恢复。

if (getTaskFamiliarityEnable()) {

// 返回 true, 说明已启用

Log.v("HelloEEG", "TaskFamiliarity is configured");

```
}
else {
    // 返回 false, 说明当前未配置。
    Log.v( "HelloEEG", "TaskFamiliarity is NOT configured");
}
可以配置熟练度任务连续运行。配置这个功能不会改变基本熟练度任务的配置。
if (setTaskFamiliarityRunContinuous(true)) {
    // return true, means success
    Log.v( "HelloEEG", "TaskFamiliarity Continuous operation");
}
else {
    // return false, meaning not supported because:
    // + connected hardware doesn't support
    // + conflict with another option already set
    // + not support by this version of the SDK
    Log.v( "HelloEEG", "TaskFamiliarity normal operation ");
}
当前配置可以被恢复。
if (getTaskFamiliarityRunContinuous()) {
    // return true, means it is enabled
```

Log.v("HelloEEG", "TaskFamiliarity Continuous operation");

} else {

// return false, meaning not currently configured

Log.v("HelloEEG", "TaskFamiliarity normal operation");

}

注意: 如果这些方法在MSG_MODEL_IDENTIFIED恢复之前被调用,则它被认为是一个鉴定 连接设备时需要处理的请求。很有可能这个特性被启用后突然发现它不再启用了。一旦确了 连接设备,若请求和软硬件不兼容,它会被重写,再发送提醒消息MSG_ERR_CFG_OVERRIDE。

注意:该算法是资源和计算密集型的。如果你需要运行Android调试器,请注意这个计算可能需 要好几分钟(估计5分钟)才能完成。然后它会完成计算并给出结果。不用调试器,这个计算将在 几秒钟完成。

DIFFICULTY 困难度

该算法试图表达目标的脑力劳动值,他可以和熟练度算法一起使用来检查学习技能的不同方面。但它也可以单独使用。

它可以用来比较一个测试对象掌握一个新技能的困难程度。一分钟的收集的数据构成一次试验,并且会产生一个针对此人的困难度索引值。运动技能单独试验的困难度索引值可以用来作 为同一个人的比较。

困难度索引值是浮点数,没有单位,只是用来对比来自同一个人的其他数据。有利于展示出 从基线 (或过去的试验)到当前试验的变化百分比。注意不同值可以表示正面也可以表示负面 的。观察HelloEEG示例应用程序是如何应用这些信息的。

困难度任务的计算必须已经被启用。

```
if (setTaskDifficultyEnable(true)) {
    // return true, means success
```

Log.v("HelloEEG", "TaskDifficulty is Enabled");

} else {

// return false, meaning not supported because:

// + connected hardware doesn't support

// + conflict with another option already set

// + not support by this version of the SDK

```
Log.v( "HelloEEG", "TaskDifficulty can not be Enabled");
```

}

当前配置可以被恢复。

```
if (getTaskDifficultyEnable()) {
    // return true, means it is enabled
    Log.v( "HelloEEG", "TaskDifficulty is configured");
}
else {
    // return false, meaning not currently configured
    Log.v( "HelloEEG", "TaskDifficulty is NOT configured");
}
可以配置困难度任务连续运行。配置这个功能不会改变基本困难度任务的配置。
```

if (setTaskDifficultyRunContinuous(true)) {
 // return true, means success

```
Log.v( "HelloEEG", "TaskDifficulty Continuous operation");
```

} else {

// return false, meaning not supported because:

- // + connected hardware doesn't support
- // + conflict with another option already set
- // + not support by this version of the SDK

Log.v("HelloEEG", "TaskDifficulty normal operation ");

}

当前配置可以被恢复。

第四章 - ThinkGear 数据类型

if (getTaskDifficultyRunContinuous()) {
 // return true, means it is enabled

Log.v("HelloEEG", "TaskDifficulty Continuous operation");

} else {

// return false, meaning not currently configured

Log.v("HelloEEG", "TaskDifficulty normal operation");

}

注意: 如果这些方法在MSG_MODEL_IDENTIFIED恢复之前被调用,则它被认为是一个鉴定 连接设备时需要处理的请求。很有可能这个特性被启用后突然发现它不再启用了。一旦确定 了 连 接 设 备 , 若 请 求 和 软 硬 件 不 兼 容 , 它 会 被 重 写 , 再 发 送 提 醒 消 息 MSG_ERR_CFG_OVERRIDE。

注意:该算法是资源和计算密集型的。如果你需要运行Android调试器,请注意这个计算可能需 要好几分钟(估计5分钟)才能完成。然后它会完成计算并给出结果。不用调试器,这个计算将在 几秒钟完成。

ECG/EKG 心电

这些数据类型只适用于ECG/EKG传感器(CardioChip)硬件设备,例如CardioChip Starter Kit Unit和 BMD10X 芯片、模块。

HEART_RATE 心率

这个整型值记录了当前的用户的心率,以每分钟的跳动为单位(BPM)。与许多其他设备常见的 心率报告不同,该值是根据每个用户实际心跳的R-peak(波峰)间隔时间精确实时计算的。 这将得到一个非常精确并连续记录的随每个用户的实际的心跳而变化的心率值。

为了方便得到类似其他ECG/EKG设备一样普遍的"平滑,平均"心率值,使用下面描述的平滑的心率的值作为输入。

Smoothed Heart Rate 平滑的心率

通常,在多数ECG/EKG设备上查看心率值时,显示"平滑的"值可以表明基于目标的自然HRV 心率中不存在有节奏的波动。差不多相同 "平滑的"效果可以通过精确的HEART_RATE值获得, 利用SDK提供的HeartRateAcceleration 类的getAcceleration ()方法即可。

查看心跳加速章节中如何计算平滑心率的描述,接着可到API**参考**的HeartRateAcceleration类获 取全部细节。

Heart Rate Acceleration 心跳加速

一个潜在的有用的心率指标就是加速率。正加速度值表示用户在一个特定的时间段内(如超 过10秒)由一定数量的BPM得知心脏速率加快,而负加速度值表示用户的心脏速率在给定时 间内一定数量的BPM减慢了。当开始运动,或在运动过后,这种加速度量可以作为一个人分别在 参加运动时心跳的加快和运动后心跳多快恢复正常的指标。

为了计算心跳加速率(或者平缓心跳率),首先要初始化应用中的HeartRateAcceleration()对象:

HeartRateAcceleration heartRateAcceleration = new HeartRateAcceleration();

这将使用10秒的时间段初始化计算。(您也可以根据你应用的需要选择重载的构造函数用更长 或更短的时间来初始化计算。)

接着,当新的心率值可用时,可以像这样获取平缓心跳率和加速率:

```
int[] result = heartRateAcceleration.getAcceleration( heartRate, poorSignal );
if( result[0] != -1 ) {
    int smoothedHeartRate = result[0];
    int heartRateAcceleration = result[1];
}
```

利用API参考的HeartRateAcceleration类获取全部细节。

Target Heart Rate for Physical Training 体能训练的目标心率

给定用户的年龄和性别信息,可以确定一个他们的心率范围是否可以达到特定的体能训练 "区域"。加上传感器的<u>HEART_RATE</u>信息,应用程序可以告诉用户他们的当前心率是否 在目标训练区域中(例如运动后正常)。

为了确定人的目标心率范围,首先要创建 TargetHeartRate 对象:

TargetHeartRate targetHeartRate = new TargetHeartRate();

接着 ,要让用户在任何时候可以确定目标心率范围(最小到最大值)是否达到特定的体能训 练区域,使用getTargetHeartRate()方法:

int age = 25; String gender = "Male"; String zone = "Aerobic"; int[] range = targetHeartRate.getTargetHeartRate(age, gender, zone); int lowerBound = range[0];

int upperBound = range[1];

下限值和上限值可以用来比对用户的HEART_RATE 或 平滑心跳率来确定用户的目标范围是 否适用于目标训练区域

性别必须是"男"或者"女"。 区域必须是下列之一:

- "Light Exercise"
- "Weight Loss"
- "Aerobic"
- "Conditioning"
- "Athletic"

如果任何参数有误,那么该方法会返回int[] of -1, -1 。

重要:测量心率不应该在用户从事体力活动的时候,用户需要暂时停止活动,然后再测量他们 的心率。

(参考文献)

1. http://www.heart.org/HEARTORG/GettingHealthy/PhysicalActivity/Target-Heart-Rates_UCM_434341_

- 2. http://www.cdc.gov/physicalactivity/everyone/measuring/heartrate.html
- 3. http://www.heart.com/heart-rate-chart.html
- 4. http://www.thewalkingsite.com/thr.html

Heart Fitness Level 健康心脏水平

.给定一个人的年龄,性别,和当前的静息心率,可以大致得到该人目前的心脏健康值 , 并能标记为 "欠佳","低于平均水平","一般","高于平均水平","好","优秀"或"运动员"。

要确定一个人的心脏的健康水平,首先创建一个HeartFitnessLevel对象:

HeartFitnessLevel heartFitnessLevel = new HeartFitnessLevel();

然后,一旦获取了一个人的年龄,性别,和当前心率,既使用getHeartFitnessLevel()方法:

int age = 25; String gender = "Male"; // "Male" or "Female" int restingHR = 60; String heartFitnessLevel = heartFitnessLevel.getHeartFitnessLevel(age, gender, restingHR);

性别必须是"男"或者"女",否则该方法会返回空字符串("")。

heartFitnessLevel的值会返回为下列之一: "欠佳","低于平均水平","一般","高于平均水 平","好","优秀"或"运动员"

(参考文献)

1. http://www.topendsports.com/testing/heart-rate-resting-chart.htm

RELAXATION 放松度

放松度数值可以基于用户的心率可变性(HRV)特征 ,提示用户其心跳是否属于放松,或者兴奋 , 压力,疲劳。取值范围从1到100。高放松值趋向于表示一种放松的状态,而低放松值则趋向于 表示兴奋,压力,或疲劳。

要通过MSG_RELAXATION接收值给Handler,只需将TGDevice连接到ThinkGear ECG/EKG传 感器(CardioChip),即用户能至少持续一分钟获取良好,干净的ThinkGear ECG/EKG传感器 (CardioChip)的信号(SENSOR_STATUS = = 200每分钟)。如果信号中断,SENSOR_STATUS超过 200,那么这个计算将重置并重新开始,再接收一分钟纯数据返回MSG_RELAXATION值。

为达到最佳效果,用户在数据收集时应平静端坐。

(参考文献)

- 1. Neurosci Biobehav Rev. 2009 Feb; 33(2): 71-80. Epub 2008 Jul 30. Heart rate variability explored in the frequency domain: a tool to investigate the link between heart and behavior. Montano N, Porta A, Cogliati C, Costantino G, Tobaldini E, Casali KR, Iellamo F.
- Int J Cardiol. 2002 Jul; 84(1): 1-14. Functional assessment of heart rate variability: physiological basis and practical applications. Pumprla J, Howorka K, Groves D, Chester M, Nolan J.
- 3. International Conference on Computer and Automation Engineering. A Review of Measurement and Analysis of Heart Rate Variability. Dipali Bansal, Munna Khan, A. K. Salhan.
- 4. Neurosci Biobehav Rev. 2009 Feb; 33(2): 81-8. Epub 2008 Aug 13. Claude Bernard and the heart-brain connection: further elaboration of a model of neurovisceral integration. ayer JF, Lane RD.

RESPIRATION 呼吸率

呼吸率数据值记录了用户大概每分钟的呼吸率。它是根据用户的ECG/EKG和心率变异性 (HRV)特征计算的。

要通过MSG_RESPIRATION接收数值给应用的Handler,只需将TGDevice连接到ThinkGear ECG/EKG传感器(CardioChip),即用户能至少持续64秒在良好,干净的ThinkGear ECG/EKG传 感器(CardioChip)获取信号(SENSOR_STATUS>= 200每64秒)。一旦收到第一个 MSG_RESPIRATION消息,更新的值将每10秒通过随后的MSG_RESPIRATION发送。如果信号 中断,SENSOR_STATUS会少于200,然后这个计算将重置并重新开始,再接收64秒的纯数据返回 MSG_RESPIRATION值。

为达到最佳效果,用户在数据收集时应平静端坐。

(参考资料)

1. Rosenthal, Talma, Ariela Alter, Edna Peleg, and Benjamin Gavish. "Device-guided breathing exercises reduce blood pressure: ambulatory and home measurements." American Journal of Hypertension. 14. (2001): 74-76.

e Calculation of Respiration Rate must be enabled. And once enabled it will run continuously until disabled.

第四章 - ThinkGear数据类型

```
if (setRespirationRateEnable(true)) {
     // return true, means success
     Log.v( "HelloEKG", "RespirationRate is Enabled");
}
else {
     // return false, meaning not supported because:
     // + connected hardware doesn't support
     // + conflict with another option already set
     // + not support by this version of the SDK
     Log.v( "HelloEKG", "RespirationRate can not be Enabled");
}
当前配置可以被恢复。
if (getRespirationRateEnable()) {
     // return true, means it is enabled
     Log.v( "HelloEKG", "RespirationRate is configured");
}
else {
     // return false, meaning not currently configured
     Log.v( "HelloEKG", "RespirationRate is NOT configured");
```

```
}
```

注意:如果这些方法在MSG_MODEL_IDENTIFIED接收之前被调用,则它被认为是一个鉴定连接设备时需要处理的请求。很有可能这个特性被启用后突然发现它不再启用了。一旦确定了连接设备,若请求和软硬件不兼容,它会被重写,再发送提醒消息MSG_ERR_CFG_OVERRIDE。

注意:该算法是资源和计算密集型的。如果你需要运行Android调试器,请注意这个计算可能需 要好几分钟(估计5分钟)才能完成。然后它会完成计算并给出结果。不用调试器,这个计算将在 几秒钟完成。

Heart Risk Awareness 心脏病风险意识

心脏风险意识数值的目的是为了提高用户的健康意识,比如发现HRV非常低时。低HRV已被 证实会增加心脏病发作的风险和死亡率。

要确定一个人的心脏风险意识,先创建NeuroSkyHeartMeters对象:

NeuroSkyHeartMeters neuroSkyHeartMeters = new NeuroSkyHeartMeters();

然后用下面两个方法之一来计算:

● 使用R-R间隔集

当应用Handler接收到MSG_EKG_RRINT消息,保存R-R间隔值到一个缓冲区。一旦缓冲区有 至少60个R-R间隔,使用NeuroSkyHeartMeters 类的calculateHear-tRiskAware(Integer[] rrIntervalInMS)方法:

```
private final Handler handler = new Handler() {
        ArrayList<Integer> temp_rrintBuffer = new ArrayList<Integer>();
        Integer[] rrinterBuffer = new Integer[60];
        @Override
        public void handleMessage( Message msg ) {
            switch( msg.what ) {
              //...
              case MSG_EKG_RRINT:
                  temp_rrintBuffer.add( msg.arg1 );
                  if( temp_rrintBuffer.size()==60 ) {
                      for( int i = 0; i<60; i++ ) {
                          rrintBuffer[i] = temp_rrintBuffer.get(i);
                      }
                      temp_rrintBuffer.clear();
                      int heartRiskAwareness = neuroSkyHeartMeters.calculateHeartRiskAware(
rrintBuffer );
                  }
                  break;
                //...
            } /* end switch on message type */
        } /* end handleMessage() */
    }; /* end Handler */
```

● 使用储存数据

只需使用NeuroSkyHeartMeters类的calculateHeartRiskAware(String fileName)方法:

int heartRiskAwarness = neuroSkyHeartMeters.calculateHeartRiskAware("john");

注意:方法里的"fileName"参数是储存了已计算好的心脏年龄的文件名。

Results 结果

返回值将是一个"心脏病风险意识"指数,其值为"0","1","2"或"3"。可以根据用户的"心脏风险意 识"由应用程序将以下信息反馈给用户:

HeartRiskAwareness = 0

你的HRV此时并不低。低HRV已被证实会增加心脏病发作的风险和死亡率。 现在你的HRV 值表明你目前承担的风险有限或没有风险。

HeartRiskAwareness = 1

你的HRV相对较低。低HRV已被证明会增加心脏病发作的风险和死亡率。 建议你多运动,注 意饮食。你可以吃些可预防心脏病的食物,如坚果、鱼类、粗粮、蔬菜,多喝绿茶。 HeartRiskAwareness = 2

你的HRV很低。低HRV已被证实会增加心脏病发作的风险和死亡率。会影响心脏的健康的坏 习惯有:总吃高脂高糖的食物,抽烟,酗酒,缺乏锻炼,心理压力高,长时间的睡眠不足。建议你改 变这些坏习惯,适量饮酒,健康饮食,适当锻炼,控制体重,养成良好的睡觉习惯,保持和平的心态。

HeartRiskAwareness = 3

你的HRV非常低。低HRV已被证明会增加心脏病发作的风险和死亡率。建议你改变你的一些 习惯。你要考虑戒烟,停止饮酒,还要确保一定的运动量,控制你的体重,养成良好的睡眠习惯, 饮食多吃纤维少盐,保持和平的心态。心脏病的症状包括胸痛、肩痛、呼吸困难,消化不良,严重 疲劳。如果出现这些症状,请去医院求诊。

(参考内容,详见 心脏年龄)

HEART_AGE 心脏年龄

心脏年龄数值表示了目标心脏的相对年龄,该值是根据他们的心率变异性(HRV)与普遍人群特 征对比得出的。低HRV会增加死亡风险,也代表可能高于该用户的生理年龄的心脏年龄(如一 个35岁用户的HRV特征表明其心脏是45岁)。该计算将考虑用户的生理年龄。要用这个数据建 议目标对象生理年龄至少10岁。

要通过MSG_HEART_AGE接收数据给应用的Handler,首先通过TGDevice对象设定用户的生理 年龄tgDevice.inputAge = 25(理所应当用25代替用户的实际年龄)。接着,只需将TGDevice连 接到ThinkGear ECG/EKG传感器(CardioChip),即用户能至少持续60次心跳在良好,干净的 ThinkGear ECG/EKG传感器(CardioChip)获取信号(SENSOR_STATUS>= 200每60次心跳)。如果 信号中断,SENSOR_STATUS会少于200,然后这个计算将重置并重新开始,再接收60次心跳的纯 数据返回MSG_HEART_AGE值。

为达到最佳效果,用户在数据收集时应平静端坐。

举一个你的应用和用户可能会用到这个信息的例子:你的应用可以根据心脏年龄值反馈下述 信息给客户:

青少年心脏: < 25岁

你的心脏年龄是xx岁,比你的实际年龄大/少xx岁。你年轻的心脏年龄让你精力充沛,积极思考, 它有助于你处理高要求的工作和运动。年轻的心脏同时也需要照顾。建议您避免熬夜深夜,做 适量的运动,保持一个和平积极的心态。你还应多吃新鲜水果和蔬菜,减少高脂肪食物的摄入来 保持你的状态。

年轻人心脏: 26 - 39 岁

你的心脏年龄是xx岁,比你的实际年龄大/少xx岁。你有一颗成熟的心脏。在紧张的工作环境里, 请不要忘了保证足够的睡眠和锻炼,吃好,照顾好自己。

中年人心脏: 40 - 55 岁

你的心脏年龄是xx岁,比你的实际年龄大/少xx岁。请密切关注您的心脏健康并合理安排你的工 作和生活以此来减轻心脏的负担。建议你多吃对心脏有益的食物,如鱼类、全麦、豆类、坚果、 蔬菜、红酒和绿茶。你也应当保证一定的运动量来强化你的心脏。

有活力的老年人心脏: 56 - 70 岁

你的心脏年龄是xx岁,比你的实际年龄大/少xx岁。你的心脏功能正在向老年前进。建议您讲 究生活纪律,避免变得过于兴奋或紧张。你也应该定期体检和进食更多对你的心脏有益的食 物,如鱼类、粗粮、豆类、坚果、蔬菜、红酒和绿茶。同样重要的是你也应当保证适当的运 动量让你的心脏有效工作。

老年人心脏: >70 岁

你的心里年龄是xx岁,比你的实际年龄大/少xx岁。建议您定期去医院体检查,仔细按照医生的 指示来预防或治疗心脏病。你也需要适量的运动,保持和平的心态。讲究生活纪律和健康饮食 可以改善心血管系统的功能,防止心脏病。

(参考文献)

- 1. Res Sports Med. 2010 Oct; 18(4):263-9. Age and heart rate variability after soccer games. Yu S, Katoh T, Makino H, Mimuno S, Sato S.
- 2. J Am Coll Cardiol. 1998 Mar 1; 31(3): 593-601. Twenty four hour time domain heart rate variability and heart rate: relations to age and gender over nine decades. Umetani K, Singer DH, McCraty R, Atkinson M.
- 3. Am J Cardiol. 2010 Apr 15; 105(8): 1181-5. Epub 2010. Relation of high heart rate variability to healthy longevity. Zulëqar U, Jurivich DA, Gao W, Singer DH.
- 4. Cardiovasc Electrophysiol. 2003 Aug; 14(8): 791-9. Circadian proële of cardiac autonomic nervous modulation in healthy subjects: differing effects of aging and gender on heart rate variability. Bonnemeier H, Richardt G, Potratz J, Wiegand UK, Brandes A, Kluge N, Katus HA.
- 5. Pacing Clin Electrophysiol. 1996 Nov; 19(11 Pt 2): 1863-6. Changes in heart rate variability with age. Reardon M, Malik M.

Personalization 个性化心电识别

该算法允许TGDevice尝试去识别已连接到ECG/EKG数据的用户。要使用它,用户需要"培训"他们到TGDevice的ECG/EKG数据。然后,当TGDevice从用户读取心电图数据时,它可以试图确定具体读取的哪个培训用户(如果有的话)的数据。

参见 附录 C and 附录 D 的步骤可以启用这个功能。

使用个性化心电识别算法有两个步骤:

培训

记录用户ECG/EKG数据的第一部分是使用TGDevice类的EKGstartLongTraining(String userName)方法。如果用户能和ECG/EKG传感器设备保持良好干净的接触,消息 MSG_EKG_TRAIN_STEP将会发送到你应用的Handler,告诉你目前在哪一步。完成这两个 步骤后,MSG_EKG_TRAINED会发到应用的Handler表明记录结束,接着Handler会使用 EKGstopTraining()方法来停止。

检测

这部分是根据第一部分保存的数据来识别用户的。要识别用户,调用EKGstartDetection()方法。 接着,如果用户能与ECG/EKG传感器硬件保持良好干净的连接,TGDevice会将消息 MSG_EKG_IDENTIFIED发送到应用程序的Handler。返回值将是一个已注册过的用户名,或"供 名"。

```
private final Handler handler = new Handler() {
    @Override
    public void handleMessage( Message msg ) {
        switch( msq.what ) {
            //...
            case MSG EKG TRAIN STEP:
                int trainStep = msg.arg1;
                break;
            case MSG_EKG_TRAINED:
                tgDevice.EKGstopTraining();
                break;
            case MSG EKG IDENTIFIED:
                String result = msg.obj;
                // result is the text of the userName for the matched profile //
                or "Unknown" if there isn't a good match
                tgDevice.EKGstopDetection();
                break;
            //...
       } /* end switch on message type */
   } /* end handleMessage() */
}; /* end Handler */
//...
tgDevice.EKGstartLongTraining( "John" );
```

//...

tgDevice.EKGstartDetection();

EKG_RRINT 心电的RR间隔

每当沿着用户的心电轴检测到R-peak(波峰)时,消息MSG_EKG_RRINT会被发送到Handler, 用毫秒表示距离上次R-peak的R-R间隔。

恰当的应用设计

重要:要在应用程序正式发布之前,确保你的应用已考虑或处理了以下几点:

如果你的应用Handler收到了MSG_STATE_CHANGE消息或者除了 STATE_CONNECTING和STATE_CONNECTED以外的任意消息,就应该仔细把每一 个可能发生的错误的处理信息通过UI发送给客户。不在UI处理好这些错误,会导致用 户体验很差。下面是一些例子说明:

——如果收到STATE_ERR_BT_OFF消息,应该提示用户打开蓝牙适配器,然后再重新尝试连接。

——如果收到STATE_ERR_NO_DEVICE消息, 应该提醒用户先根据他们ThinkGear硬件设备收 到的指令,把ThinkGear硬件设备和Android设备的蓝牙进行配对。

——如果收到STATE_NOT_FOUND消息,应该提醒用户检查ThinkGear硬件设备是否和Android 设备已成功配对 (和STATE_ERR_NO_DEVICE的情况相同),比如查看他们ThinkGear硬件设备 是否打开,是否在距离范围内,电池电量是否充足。

查看 TGDevice 状态 获取更多信息。

- 确保你的应用程序总是在处理POOR_SIGNAL/SENSOR_STATUS数据类型。它是由 ThinkGear设备输出,并且提供了传感器是否正常与用户接触的重要信息。如果它正在提 示可能出了一些问题(即EEG设备不等于0或ECG/EKG设备不等于200),那么应用程序 应 该 通 知 用 户 应 该 正 确 穿 戴 好 ThinkGear 硬 件 设 备 , 或 者 当 POOR_SIGNAL/SENSOR_STATUS持续显示问题期间忽视其输出的数据值,可根据 实际需求做出最符合你应用程序的判断。
- 为了做出让用户体验保持一致性、熟悉性、容易掌握和使用的跨平台设备,您的应用程 序设计应该遵循NeuroSky的App Standards的准则和惯例。

故障排除

注意:目前还没有已知问题,如果你遇到任何bug或问题,请访问http://support.neurosky.com,或联系邮箱 support@neurosky.com.

- 如果你需要进一步的帮助,可以访问链接http://developer.neurosky.com查看是否有最新信息。
- 要联系 NeuroSky 技术支持,请访问 http://support.neurosky.com,或者发送邮件给 support@neurosky.com 。
- 关于开发者的论坛支持,请访问我们的社区论坛:

http://www.linkedin.com/groups/NeuroSky-Brain-Computer-Interface-Technology-3572341

重要声明

SDK里的算法是专门用来提高个人健康养生意识的,并不能代替医疗护理。 这些算法不用于 诊断,治疗,治愈或预防任何疾病,开出任何药物,亦或是代替医疗设备的治疗。在某些情况下,该 算法可能也会产生错误或不准确的结果。SDK文档中的算法和显示的数据只是针对特定用户 使用的。NeuroSky算法公开可用,不承担其最终使用和显示内部代码的责任。

如果用户拥有心脏起搏器,该算法功能可能不好也可能显示准确的数据。为了收集最佳效果的EEG数据,需要用户平静端坐,规律呼吸,尽可能小幅动作。

警告和免责声明:

这些算法禁止任意形式的非法使用, 禁止应用于生命保障,安全设备或系统的组件,禁止应 用于军事或核程序,禁止任意程序中的算法故障会导致人员伤亡。你使用软件开发工具包,算 法和其他NEUROSKY的产品或服务是基于现状的,神念科技公司不做任何明示或 暗示的保 证,授权或许可。包括有关适销性,知识产权(包括专利,版权或其他) 或特殊目的的适用性 的保证。

神念电子科技概不负责任何意外,偶然,间接的伤害。包括但不限于利润或收入的损失,无 论神念科技公司是否已被告知,此类损害均有可能发生。

附录 A: 其他参考文献

• http://developer.android.com/guide/topics/wireless/bluetooth.html

附录 B: UART (非蓝牙) 连接

一些安卓设备,比如手机或平板,被设计成通过UART(特别是CardioChip)连接到ThinkGear硬件设备,安卓的TG-SDK提供了另一个构造函数和连接方法:

public TGDevice(InputStream s, OutputStream o, Handler h); public synchronized void connectStream(boolean rawEnabled);

取代使用构造函数中的BluetoothAdapter和蓝牙通信的是,它利用了InputStream和OutputStream进行 沟通。这个构造函数的connectStream()方法类似于基于蓝牙构造函数的connect()方法。

要使用这个SDK的API,必须首先准备如下的Android / Linux系统:

1. 手动把ThinkGear硬件设备的UART插脚连接到安卓设备的UART插脚中。

2. 确保 Linux OS 层在 Android OS 层 下,使UART 端口作为一个通行串口 ,比如"/mnt/uart1"

接着由于Android不提供API来直接访问串行端口,我们必须在Android层和Linux串口层使用JNI:

1. 编译uart.cpp 文件(由NeuroSkt单独提供的一个生成文件)到本地libJRDBCM.lib的库文件,JNI 便能被调用。该库能让你的应用启用串行端口。

2. 使用程序的 JNI 调用 SerialJNI_open() 方法。就会 打开 FileDescriptor,然后你便可以获取 FileInputStream 和FileOutputStream 应用于TGDevice 构造函数。

```
public class Native {
    static {
        System.loadLibrary( "JRDBCM" );
    }
}
String dev = "/mnt/uart1";
FileDescriptor serialPort = Native.SerialJNI_open( dev );
InputStream iStream = new FileInputStream( serialPort );
OutputStream oStream = new FileOutputStream( serialPort );
TGDevice tgDevice = new TGDevice( iStream, oStream, handler );
tgDevice.connectStream( true );
```

从这里开始,TG API便和本文档其他描述蓝牙的部分一样了。记住当你完成TGDevice 的操作,关闭 TGDevice 之后,使用Native.SerialNJI.close ()关闭FileDescriptor。

附录 C: 启用个性化的使用

你的代码要初始化 btAdapter 和 tgDevice 必须要先启用个性化设置。 同时也必须允许写入 安卓文件系统。参见 附录 D.

```
public void onCreate(Bundle savedInstanceState) {
    //...
    btAdapter = BluetoothAdapter.getDefaultAdapter();
    if( btAdapter != null ) {
        TGDevice.ekgPersonalizationEnabled = true;
        tgDevice = new TGDevice( btAdapter, handler );
    }
    //...
}
```

附录 D: 允许写入安卓文件系统

一个Android应用程序要在本地文件系统创建文件,必须要在你工程AndroidManifest.xml加入语句:

<uses-permission android:name="android.permission.BLUETOOTH" /> <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>