BMD100 .NET API

November 22, 2011

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 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 NEUROSKY 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 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 NEUROSKY PRODUCTS OR 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.

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

Contents

November 22, 2011 | © 2009 NeuroSky, Inc. All Rights Reserved.

Introduction

The provided ThinkGear.dll file is a C# library that can be used to retrieve data from NeuroSky devices. This is the documentation for the C# DLL API.

There are two classes that are required under the **NeuroSky.ThinkGear** Namespace, they are **Connector Class** and **TGParser Class**. The **Connector Class** is the class that connects to the serial port and pull out the DataRows to be parsed. This data is then passed into the **TGParser Class** to be parsed into data.

Connector Class

Methods

Name	Description
Close	Performs cleanup of the ThinkGear Connector instance.
Connect(**string**)	Attempts to open a connection with the port name of **string**
ConnectScan	Attempts to open a connection to the first Device seen by the Connector
ConnectScan(**string**)	Same as ConnectScan but uses the first port name of **string**
Disconnect	Closes all open connections
Disconnect(**Connection**)	Closes a specific **Connection**
Disconnect(**Device**)	Closes a specific **Device**
Send(**string**, **byte[]**)	Send a **byte[]** to the port specified by the **string**

November 22, 2011 | © 2009 NeuroSky, Inc. All Rights Reserved.

Events

Name	Description
DeviceFound	Occurs when a ThinkGear device is found. This is where the application chooses to connect to that port or not.
DeviceNotFound	Occurs when a ThinkGear device could not be found. This is usually where the application displays an error that it did not find any device.
DeviceValidating	Occurs right before the connector attempts a serial port. Mainly used to notify the GUI which port it is trying to connect.
DeviceConnected	Occurs when a ThinkGear device is connected. This is where the application links the OnDataReceived for that device.
DeviceConnectFail	Occurs when the Connector fails to connect to that port specified.
DeviceDisconnected	Occurs when the Connector disconnects from a ThinkGear device.

Example Code:

Declare the variable like below.

```
private Connector connector;
```

In the constructor, construct the connector and link all the events with method like below.

```
connector = new Connector();
connector.DeviceConnected += new EventHandler(OnDeviceConnected);
connector.DeviceFound += new EventHandler(OnDeviceFound);
connector.DeviceNotFound += new EventHandler(OnDeviceNotFound);
connector.DeviceConnectFail += new EventHandler(OnDeviceNotFound);
connector.DeviceDisconnected += new EventHandler(OnDeviceDisconnected);
connector.DeviceValidating += new EventHandler(OnDeviceValidating);
```

In the event handler for DeviceConnected, link the DataReceived for that **Device** like the following:

```
void OnDeviceConnected(object sender, EventArgs e){
    Connector.DeviceEventArgs de = (Connector.DeviceEventArgs)e;

    Console.WriteLine("New Headset Created." + de.Device.DevicePortName);

    de.Device.DataReceived += new EventHandler(OnDataReceived);
}
```


November 22, 2011 | © 2009 NeuroSky, Inc. All Rights Reserved.

TGParser Class

Methods

Name	Description
Read(**DataRow**[])	Parses the **DataRow**[] and stores into **ParsedData**.

Example Code:

Following the example above in the event handler of OnDataReceived, use the **TGParser** to parse the **DataRow**[] like below:

```
void OnDataReceived(object sender, EventArgs e){
    Device d = (Device)sender;
    Device.DataEventArgs de = (Device.DataEventArgs)e;
    DataRow[] tempDataRowArray = de.DataRowArray;
    TGParser tgParser = new TGParser();
    tgParser.Read(de.DataRowArray);

    /* Loop through new parsed data */
    for (int i = 0; i < tgParser.ParsedData.Length; i++){
        if (tgParser.ParsedData[i].ContainsKey("Raw")){
            Console.WriteLine("Raw Value:" + tgParser.ParsedData[i]["Raw"]);
        }

        if (tgParser.ParsedData[i].ContainsKey("HeartRate")){
            Console.WriteLine("HR Value:" + tgParser.ParsedData[i]["HeartRate"]);
        }

        if (tgParser.ParsedData[i].ContainsKey("PoorSignal")) {
            Console.WriteLine("PS Value:" + tgParser.ParsedData[i]["PoorSignal"]);
        }

        if (tgParser.ParsedData[i].ContainsKey("BMDConfig")) {
            Console.WriteLine("Config Value:" + tgParser.ParsedData[i]["BMDConfig"]);
        }
    }
}
```


November 22, 2011 | © 2009 NeuroSky, Inc. All Rights Reserved.

Data Types

BMD100 Data Types

Key	Description	Data Type
Raw	EEG	short
HeartRate	Heart rate (BPM)	double
Time	TimeStamps of packet received	double
PoorSignal	Poor Signal	double
BMDConfig	Config Byte	double

Note: As each packet is received in the ThinkGear DLL, the associated timestamp is also recorded

November 22, 2011 | © 2009 NeuroSky, Inc. All Rights Reserved.