Sending Command Bytes to BMD100

Introduction

To internally configure the BMD100, command bytes must be sent to the device after powering.

Usage

When to Send Command Bytes

The command bytes can be sent at any time after powering the BMD100. It can also be sent multiple times (the BMD100 will simply ignore the repeated command bytes). It is recommended to send the command bytes immediately after connecting to the BMD100. Once the command bytes have been sent to the BMD100, the device will remain configured upon subsequent disconnections or connections to the application. However, if the device is power cycled, it will need to be reconfigured once again. For this reason, it is recommended to send the command bytes *every time the application connects to the device.*

Which Bytes to Send

The following 8 bytes must be sent to the BMD100:

```
[0xAA 0xAA 0x04 0x03 0x40 0xF9 0x00 CONFIG_BYTE]
```

Note that these bytes must be sent together, not one at a time. The last byte, CONFIG_BYTE, is outputted from the BMD100 device once per second. For detailed documentation on the CONFIG_BYTE, and how to read it from the BMD100, please see the **BMD100 Communications Protocol** document.

Example Code

The following C# code details how to extract the CONFIG_BYTE from the BMD100, and send the command bytes to the device.

```
namespace NeuroSky
{
    class bmd100_testapp
    {

        /* bytes to send for ASIC configuration */
        private byte[] bytesToSend;

        /* handle the case in which data is recieved */
        void OnDataReceived(object sender, EventArgs e)
        {
            Device d = (Device)sender;
            Device.DataEventArgs de = (Device.DataEventArgs)e;
            /* create a new DataRow to hold data */
            ThinkGear.DataRow[] tempDataRowArray = de.DataRowArray;

            /* create a new TGParser object to parse the data coming in */
            TGParser thinkGearParser = new TGParser();
            thinkGearParser.Read(de.DataRowArray);

            /* Loop through new parsed data */
            for (int i = 0; i < thinkGearParser.ParsedData.Length; i++)
            {
                /* send the configuration bytes to the chip. this happens once every time the
device is connected to */
                if(thinkGearParser.ParsedData[i].ContainsKey("BMDConfig")) {
                    if(bytesToSend == null) {
                        bytesToSend = new byte[8] { 0xAA, 0xAA, 0x04, 0x03, 0x40, 0xF9, 0x00,
(byte)thinkGearParser.ParsedData[i]["BMDConfig"] };
                        connector.Send(device.PortName, bytesToSend);
                    }
                }

                .
                .
                .
```

To see the entire source code, please view the **bmd100_testapp.cs** file under the Software Development Tools\C# Project Code directory.

Corporate Address
NeuroSky, Inc.
125 S. Market St., Ste. 900
San Jose, CA 95113
United States
(408) 600-0129

Questions/Support: http://support.neurosky.com
or email: support@neurosky.com

Community Forum: http://developer.neurosky.com/forum

Information in this document is subject to change without notice.

Reproduction in any manner whatsoever without the written permission of NeuroSky Inc. is strictly forbidden. Trademarks used in this text: eSense™, ThinkGear™, MindKit™, NeuroBoy™and NeuroSky®are trademarks of NeuroSky, Inc.

Disclaimer: The information in this document is provided in connection with NeuroSky products. No license, express or implied, by estoppels or otherwise, to any intellectual property rights is granted by this document or in connection with the sale of NeuroSky products. NeuroSky assumes no liability whatsoever and disclaims any express, implied or statutory warranty relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or non-infringement. In no even shall NeuroSky be liable for any direct, indirect, consequential, punitive, special or incidental damages (including, without limitation, damages for loss of profits, business interruption, or loss of information) arising out of the use of inability to use this document, even if NeuroSky has been advised of the possibility of such damages. NeuroSky makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. NeuroSky does not make any commitment to update the information contained herein. NeuroSky's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

NeuroSky
Brain-Computer Interface Technologies