



Annales UMCS Informatica AI X, 1 (2010) 7-14

DOI: 10.2478/v10065-010-0031-y

Annales UMCS

Informatica

Lublin-Polonia

Sectio AI

<http://www.annales.umcs.lublin.pl/>

Developing brain electric activity acquisition software for Linux

Sławomir Kotyra, Grzegorz M. Wójcik*

*Institute of Computer Science, Maria Curie-Skłodowska University,
Plac Marii Curie-Skłodowskiej 5, 20-031 Lublin, Poland.*

Abstract – This article discusses the first successful phase of our work on the construction of a complete modular research station for EEG signal acquisition and analysis performed in real-time mode and in a way that meets our needs. Our intentions were presented in [1].

1 Introduction

The first attempts to analyse electric signals corresponding to brain activity were made at the end of 19th century, when Richard Caton attached a sensitive mirror galvanometer directly to the surface of exposed animal brains [2].

The term "electroencephalography" (EEG) was introduced in 1924 by Hans Berger, who carried out the first non-invasive registration and analysis of human EEG signals. He discovered regular waves of two frequencies which depend on brain activity level. Berger called those "alpha" and "beta" waves [3].

In 1940s William Grey Walter built an automatic low frequency analyser which separated the EEG spectrum into distinct frequency bands [4, 5]. His further work led to the development of the toposcope - the first multichannel EEG system. As a consequence, the EEG topography issue appeared [6].

The general idea of electroencephalography (EEG) is based on recording and analysing electric brain activity which is generated by the joint action of neurons in a chosen part of the cortex [7]. Nowadays, more and more sensitive equipment is used

*gmwojcik@gmail.com

for this purpose. Likewise, signal analysis is carried out by means of more and more advanced and dedicated software [8].

2 Hardware

MindSet MS 1000, manufactured by Nolan Computer Systems, is used as the hardware source of an EEG signal (Fig. 1). It is a real-time device containing a 16-channel analogue differential amplifier which receives signals from cap electrodes and amplifies them 32000 times. An amplified analogue signal is converted into a digital one as it goes through sample-and-hold circuits and an analogue-to-digital converter. Inside the device, there is a 32kB RAM buffer enabling it to withstand arrhythmical cooperation with a PC [9].



Fig. 1. Mindset MS-1000 EEG System.

Cooperation with the controlling computer is achieved by means of SCSI interface. There is a suitable set of commands implemented by the MS-1000 EEG device developer. The Inquiry command may be mentioned as an example here [10].

In order to control the MS-1000 device, a computer must have an external SCSI port. In our case it is the PCI AVA 2906 card (Fig. 2) manufactured by Adaptec [11]. The device is connected to a computer with a special multi-wire cable.

3 Solution

Individual programs were developed as small independent modules. A well-known piping mechanism was chosen for the purpose of interprocess communication. In this way it was easy to configure the set of modules according to current needs. It is an important feature as far as research and development perspectives are concerned.



Fig. 2. Adaptec AVA 2906 controller.

For practical reasons (good quality development tools and documentation without restrictive license limits), our work is carried out in the GNU/Linux operation system environment with the application of a gcc compiler.

4 Driver

The Linux OS kernel has a compiled-in `aic7xxx` module which is responsible for servicing SCSI devices [12]. It is possible to develop SCSI devices software after installing a `sg3_utils` tool packet [13]. This packet contains, among others, the `rescan-scsi-bus.sh` script which detects SCSI bus changes (detects and registers newly-connected devices in the system) and the `sg_linux_inc.h` header file that correctly attaches `sg.h` and `scsi.h` header files to compiled programs. A simplified example of the code performing the Inquiry command with the help of `sg3_utils` is presented in Listing 1.

Listing 1. Example of the code performing the Inquiry command.

```
1: #include "sg_linux_inc.h"
2:
3: #define CMD_LEN 0x6 //command length
```

```
4: #define CMD 0x12 //inquiry command code
5: #define REPLEN 0x24 //expected response length
6:
7: char cmd[ CMD_LEN ] = { CMD, 0, 0, 0, REPLEN, 0 }; //command
    - 6 bytes
8: unsigned char rbf[ REPLEN + 1 ]; //response buffer
9: //transitional buffer
10: sg_io_hdr_t *io_hdr = malloc( sizeof( sg_io_hdr_t ) );
11:
12: //buffer preparing
13: io_hdr->interface_id = 'S'; //'S' is obligatory for SCSI
14: io_hdr->dxfer_direction = SG_DXFER_FROM_DEV; //data flow
    direction
15: io_hdr->cmdp = cmd;
16: io_hdr->cmd_len = CMD_LEN;
17: io_hdr->dxferp = rbf;
18: io_hdr->dxfer_len = REPLEN;
19: io_hdr->sbp = NULL; //don't care
20: io_hdr->mx_sb_len = 0;
21: io_hdr->timeout = 1000; //milliseconds
22:
23: ioctl( iDH, SG_IO, io_hdr ); //right call
```

After the function call (line 23) the device's response will be placed in the memory area reserved in line 8. The address of this memory unit was passed in line 17.

5 Daemon

The main goal of this phase of our work was to control the MindSet MS-1000 EEG device. The computer program developed for this purpose is called "mstd" (MindSet Thousand Daemon). Running the program without any arguments will display a short usage (Listing 2.)

Listing 2. A short usage displayed after running the program without any arguments.

```
./mstd: Program expects one or two arguments:
    -dDevice - file name for device representing Mindset MS-1000,
    -rRate - EEG signal sampling rate (64, 128, 256, 512, 1024)
(samples amount per second).
Omitting sample rate confines program execution to device
    diagnostic
only.
Example:
```

```
mstd -d/dev/sg2 -r256
```

Running the program with one argument will perform a diagnostics of the pointed device looks like Listing 3.

Listing 3. A diagnostics of the pointed device.

```
./mstd: recognised arguments: /dev/sg2.
devOpen(): device /dev/sg2 opened with descriptor 3.
testSCSI() completed successfully , host: 0

//pending operations will be performed only for SCSI device
getAdapter() completed successfully:
Adaptec AIC7XXX EISA/VLB/PCI SCSI HBA DRIVER, Rev 7.0
<Adaptec 2902/04/10/15/20C/30C SCSI adapter>
aic7850: Single Channel A, SCSI Id=7, 3/253 SCBs

requestSense() completed successfully ,
sense buffer:
$70 p; $00; $00; $00; $00; $00; $00; $0A;
$00; $00; $00; $00; $00; $00; $00; $00; $00; $00;

scsiInquiry() completed successfully ,
recognised device: AquaLabsMS-1000 21

// pending operations will be performed only for AquaLabsMS-1000 21
getMSstatus( 1 ) completed successfully:
0. sample rate: no sampling
1. buffer head: 0x2200 (8704)
3. buffer tail: 0x2200 (8704)
5. block size: 512
6. buffer overflow: NO
7. processor type: 68B08
8. serial number: MS1000-USA000001-R1.00

//changing block size to 128B
setBlockSize( 2 ) completed successfully.
getMSstatus( 0 ) completed successfully:
0. sample rate: no sampling
1. buffer head: 0x2200 (8704)
3. buffer tail: 0x2200 (8704)
5. block size: 128
6. buffer overflow: NO

//returning to 512B block size. lunching sampling with 256sps rate
```

```
setBlockSize( 8 ) completed successfully.
setSampleRate( 3 ) completed successfully.
getMSstatus( 0 ) completed successfully:
0. sample rate: 256
1. buffer head: 0x2200 (8704)
3. buffer tail: 0x2200 (8704)
5. block size: 512
6. buffer overflow: NO

//buffer tail change after 2 seconds of sampling
getMSstatus( 0 ) completed successfully:
0. sample rate: 256
1. buffer head: 0x2200 (8704)
3. buffer tail: 0x6200 (25088)
5. block size: 512
6. buffer overflow: NO

//buffer head change after retrieving sampled data
retrieveSampledData() completed successfully.
getMSstatus( 0 ) completed successfully:
0. sample rate: 256
1. buffer head: 0x2400 (9216)
3. buffer tail: 0x6200 (25088)
5. block size: 512
6. buffer overflow: NO

//stop sampling
setSampleRate( 0 ) completed successfully.
getMSstatus( 0 ) completed successfully:
0. sample rate: no sampling
1. buffer head: 0x2200 (8704)
3. buffer tail: 0x2200 (8704)
5. block size: 512
6. buffer overflow: NO
devOpen(): device /dev/sg2 closed with code 0.
```

Running the program with two proper arguments will cause the printing of sampling results to the standard output in the text format (Table 1.).

This solution makes it easy to catch data with a well-known piping mechanism.

To check if the device and the program cooperate correctly and the retrieved data can be interpreted correctly, the calibrator of the MS-1000 device was used. The calibrator is a stable 16 Hz sinusoidal wave oscillator with a low distortion factor [14]. On the basis of the retrieved data the following chart (Fig. 3) was drawn.

Table 1

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	...
21	-18	-10	-8	-8	-8	-9	-8	-4	-9	-11	-4	...
-22	2	-1	0	4	3	1	1	4	0	0	1	...
...
7	23	11	9	15	4	15	6	8	9	17	4	...

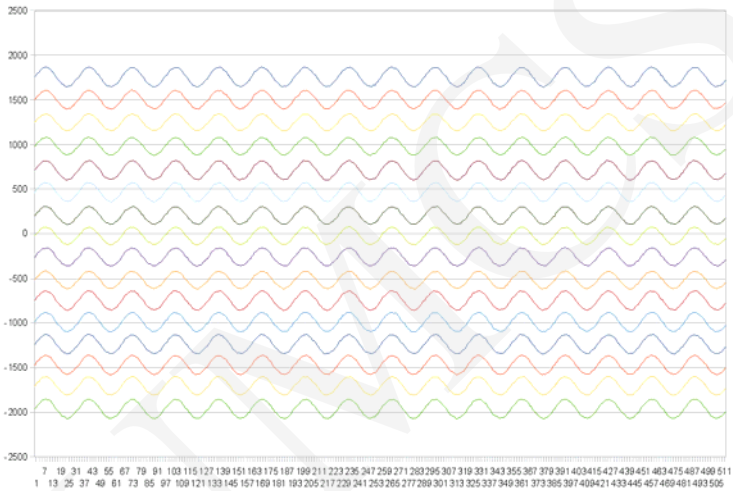


Fig. 3. One second of 16-channel, 16 Hz, 50 μV calibrator signal and 512sps sampling.

The obtained signal as a text data stream is easy to modify dynamically with general purpose tools such as "awk".

At the present stage it is possible to control the device, retrieve correct EEG signals and perform real-time modifications. The above proves that the assumptions stated in [1] are correct.

6 Summary

We presented the method of data collection of brain electrical activity from the EEG system. The created software is easy to use, works under control of Linux OS, is free and gives the possibility of modification. The data acquired into text files can be analysed, interpreted in a broad range of ways. Having the Linux driver we can continue with our EEG experiments practically without limits set up by commercial software.

7 Acknowledgements

The necessary information about the device and computer cooperation protocol through the SCSI interface was kindly provided by Mr Wayne Nolan from Nolan Computer Systems, Colorado. The authors would like to thank Mr Nolan for his invaluable help.

We would also like to thank Ms. Joanna Rola for the support in the editorial work for this paper.

References

- [1] Kotyra S., Wojcik G. M., The system of electric brain activity acquisition from EEG equipment for Linux OS, *Annales Informatica UMCS* (2008): 151–155.
- [2] Caton R., The electric currents of the brain, *British Medical Journal* 2(1) (1975): 278.
- [3] Berger H., Über das elektroenkephalogramm des menschen, *Archiv für Psychiatrie und Nervenkrankheiten* 87 (1929): 527.
- [4] Walter W. G., An automatic low frequency analyser, *Electronic Engineering* 16 (1943): 9–13.
- [5] Walter W. G., An improved low frequency analyser, *Electronic Engineering* 16 (1943): 236–238.
- [6] Walter W. G., H. Shipton, A new toposcopic display system, *J EEG Clin Neurophysiol* 3 (1951): 281–292
- [7] Niedermeyer E., Silva, F., *Electroencephalography* (4th edition) (Williams & Wilkins, Baltimore, 1999): 1258.
- [8] Makeig S., Gramann K., Jung T. et al., Linking brain, mind and behavior, *International Journal of Psychophysiology* 73 (2009): 95–100.
- [9] *Mindset Software Users Manual* (Nolan Computer Systems, 2001).
- [10] Wikipedia contributors, SCSI (Wikipedia, 10 Jan 2010).
- [11] Adaptec - Adaptec SCSI Card 2906, <http://www.adaptec.com/en-US/support/scsi/2900/AVA-2906/>.
- [12] Linux Kernel Documentation, <http://www.mjmwired.net/kernel/Documentation/scsi/aic79xx.txt>.
- [13] Linux sg3_utils package, http://sg.danny.cz/sg/sg3_utils.html, http://freshmeat.net/projects/sg3_utils.
- [14] *MindSet MS-1000 Calibrator* (Nolan Computer Systems, 2001).