ActiveD

The primary objective of this project was to give the investigator some user-friendly and configurable environment for work with ***Active Two*** system. By 'user-friendly' I mean, first of all, the possibility to display **all** the information investigator needs – and **only** the information she/he needs. It includes multiple views (with different time scales, filter settings, channels and reference schemas), color-mapping of electrode offsets, combination of monopolar and bipolar charts etc.

As the program was designed for ERP-oriented research first of all, it contains some special features for non-continuous, event-oriented recording, using Active2 digital trigger signals. But it is still possible to record data continuously.

Program was designed and tested using *original (**not Mk2**)* Active2 system with 40 channels (32 pin-channels and 8 touch-proofs), but without AIB box and additional sensors. Starting from the version *alpha 0.03* (November 15, 2005) it will (hopefully) recognize ***Mk2*** system and work with it – but **not** in **'daisy chained'** or **AIB** modes.

1 Many views.

1.1 Scalabitily

Here I mean the simplest thing – the size of the program window, which is resizable. It will allow program to work even on portable notebooks (with resolution 1024x768 – and even 800x600) – and on any large screens as well.

Program 'view' may consist of many windows (in this version they are of one type only – so-called 'oscilloscope view', I use also 'snapshots' of the last recorded trial, in plans is list of trials). You can re-arrange this windows within the main window – and even put any of them outside. The latter can be achieved by right-clicking on the child window title bar, or left-clicking the button in the View system menu. This feature can be very useful for multi-monitor systems, which are of common use now.

In different views you can choose different channels to view, different reference schema etc.

So, it is possible to run program on a portable notebook, then connect external monitor to it – and to put on it a large window with large number of channels.

Additional views can be added and deleted 'on the fly' – without stopping the

acquisition.

1.2 **Selectability of useful information**
Any view can be tuned up to reflect information you need. You can select any number of monopolar channels, change reference schema, turn trigger channel on or off. You can add to the view any number of bipolar channels (for example, it may be useful to look at the EEG and EOG simultaneously).

1.3 **Bipolars**
You can add to the view any number of bipolar channels. Clicking on <Add new> string in Bipolar listbox (right side of the View tab), in which you can select correspondent channels and set the name for the resulting bipolar. Clicking on the string with the existing Bipolar name evokes the same dialog, you can change the channels and/or name – or remove selected bipolar.

1.4 **Any kind of bipolars**
Bipolar channels may be build from any electrodes – not 'touchproofs' only.

1.5 **Striping-chart view**
It is possible to switch any view to striping-chart ('Scrolling') mode, to which a number of researchers is more accustomed. This kind of view may be flickering, especially on LCD screens. A flicker may be reduced by changing a view's 'Refresh timer' (in the view options -  button on the view tab)

1.6 **Every point is drawn**
Every point on the plot that can be shown will be shown – you'll not loose the shortest peak even in the lowest time resolution

1.7 **On-the-fly changes**
All the view parameters: electrodes (including references and bipolars), horizontal scale (window length), vertical scale, High- and low-pass filters can be changed on the fly, without stopping the acquisition

1.8 **Remembering the setup**
All views save their parameters (including window size and position) – and then restore to the saved setup. This behavior can be changed in the View Options dialog ( button on the view tab)

1.9 **Selectable colors**
It is possible to select colors for view background and for different channels – in the 'Colors' tab of main settings menu (Tools->Settings of  button on main toolbar)

1.10 **Viewing trigger**
Trigger channel can be shown as a color stripe, ling in ActiView. You can turn trigger channel on or off. On the Trigger View tab of main settings menu you can select colors for individual bits. You can assign 'Active state' (Low or High) to every individual bit. That will not change the behavior of the program – only the way the bit state will be shown on the screen.

2 Choosing channels

2.1 Selecting a number of modules

You may select a proper number of installed modules, that will reduce the risk of recording from non-existing electrodes. The number of modules can't be changed while the program is in 'Run' mode.

2.2 Electrode maps

Selecting of electrodes (channels) is performed by means of 'Electrore maps', which you can see at the bottom-left corner of the main program window (Main channel map), on the View tabs and in the Bipolar selection dialog. To switch on/off the electrode simply click on the correspondent cell. Clicking on the 'Group cell' (leftmost cell with group name in it) will switch on/off the whole group.

Right-click anywhere on the map to evoke a 'Map Dialog' – a larger (and resisable) version of the same map

When the mouse pointer is moving over any map, the number (or name) of the electrode under pointer is shown in tooltip window.

If the map is large enough, electrode numbers (or names) will be shown in corresponding cells.

2.3 Main Channel map

Main channel map is (bottom-left corner of the program window) allows you to choose only channels you need, thus reducing the system load and resulting file size. Selection will be stored along with other program parameters if this saving is turned on (Settings->General and Windows settings).

Main channel map can't be changed wile the program is in 'Run' mode.

2.4 View channel maps

In View channel maps (display and reference) and in Bipolars dialog you can switch on any electrode that is selected in the Main map.

2.5 Electrode names

can be loaded from the Biosemi *.cfg file. The command is available in map dialog evoked from Main channel map (Load Channel Names button) or from the Main menu (Tools->Load electrode names). If names are loaded, they will be displayed instead of electrode numbers in all maps and views

Loaded electrode names will be stored along with other program parameters if this saving is turned on (Settings->General and Windows settings)

2.6 Electrode offset mapping

To allow the permanent monitoring of electrodes state, absolute values of electrodes offsets are reflected through colors in all electrode maps. Five different

colors and corresponding critical values can be selected on 'Offset to colors mapping' tab in the main settings menu (Tools-Settings)

3 Decimation

3.1 Mode-independent selection

Instead of using the decimation ratio, program allows you to select desired final frequency directly. The list includes
2048, 1024, 512, 256 and 128 Hz.

3.2 Pre-decimation filtering

Before decimation signals are filtered with FIR filter. Cut-off frequency is selected as 0.75 of future (after-decimation) F_{Ny} (Nyquist frequency). Suppression at the F_{Ny} is 110dB

3.3 Decimation the trigger

Trigger channel decimation collects and reflects all the changes in the decimation interval.

To have the accurate timing information, you can switch trigger decimation off. In that case the trigger channel will have length different from others.

4 File Format

The main difference between file format of my program (so-called BDF+D) and original BDF file format is just the same, that the difference between EDF and EDF+ file formats: the existence of additional ‘Annotation channel’ with the label “EDF Annotations”. To distinguish between two file formats, I’ve also changed signature from ‘24BIT’ to ‘24BIT BDF+D’. Now I’m using annotation channel of fixed length: 192 bytes (64 three-byte ‘words’, or 96 ‘normal words’) per record, independently of currently selected sample rate and record length. Additionally, I’ve changed the ‘Labels’ fields: according to EDF+, they will read like ‘EEG Fp1’ etc. To preserve the original channels numbers, I’ve added them to the ‘Transducer type’ field, so the later looks now like ‘Active electrode #A1#’ etc.

It is possible to switch off the ‘PLUS’ extensions by clearing the checkbox “Plus file format” on the “General and Windows settings” tab of the Options window. In such a case and if record length (Trigger and Acquisition tab) is set to 1000 ms (1 s), the resulting file will be compatible with ‘original BDF’ (ActiView) files.

4.1 File storage.

I usually prefer to store files in a separate directory (and on separate drive). This behavior is regulated by ‘Store data files in’ checkbox “General and Windows settings” tab of the Options window. The directory itself may be selected with ‘Browse’ button on the same window – or when creating a first data file. It is also possible to create a small ‘directory tree’ with separate branches for different types of experiment (eq. ERP, MMN etc.) and different subjects (patients) - if the correspondent checkboxes are checked.

When you’ll select “New file” command (from menu or toolbar), the dialog will appear in which you can define standard subfields of “Local Patient description” and “Local recording description” file header fields. Once entered, the values of these fields will be stored in the configuration file (for patient, his personal data together with patient code). “Investigation code” is obligatory only if you selected to use separate subfolders for different types of experiment. “Patient code” field is obligatory, all the other fields you may leave empty.

If you have selected separate data folder (and separate subfolders), and the corresponding directory does not exist, you will be prompted to create it. The default file name for the new file will be created according to the CODE_YYMMDD_HHMM_.bdf scheme, where CODE is ‘Patient code’ field, YY – two last digits of the current year, MM – month, DD – date, HH – hour and MM – minutes. Of course, you are free to select any other directory (and enter any other name).

As now I’m using NTFS-5 file system only, I do not have options for checking file size (and breaking up files larger than 2Gb)

5 Triggering and Acquisition

Now program supports two acquisition modes – Gated and Triggered.

In Gated mode data will be recorded continuously (with records of selected length, only duration of 1000 ms (1 s) is compatible with original BDF files) – while the selected conditions are satisfied. Individual bits in ‘condition’ boxes are combined by the bitwise ‘OR’ along the raw – and by the bitwise ‘AND’ by the column. You can start and stop recording manually by the toolbar ‘Record’ and ‘Pause’ buttons (in Gate mode they work as a Radio buttons). Changes in hardware triggers override those buttons – but

only on the edge of the change. You can also temporary disable writing of the fresh data to the file.

Records will be displayed in a separate ‘Snapshot’ view. You can create more then one snapshot view (with different channels, filters, references etc). Snapshot views can be in a ‘locked’ and ‘unlocked’ state. All views in a ‘locked’ state (yellow ‘closed’ lock button on a corresponding dialog tab) display one and the same data record (usually – the latest one). Unlocked view will display any selected record that you can chose either by ‘record number’ field in the dialog – or from the list tab in the leftmost program panel.

In case of ‘Plus’ format enabled in the list tab (and on the status lines of snapshot views) additional information will be displayed:

- record number
- time (from the beginning of the file)
- record type (‘New’ for first record after interrupted recording or ‘cnt’ – for continuous rrecording)
- for continuous recording – time from the start of the current continuous portion
- value of the trigger word at the start of the current record

If you’ll switch off ‘Plus’ format, you’ll have only record numbers…

5.1 Gated input.

Gate conditions.

Gate state (true or false) is a logical (Boolean) result of trigger-channel data analysis against the conditions set by user on the 'Trigger and Acquisition' tab of the 'Options' window. Let me describe these conditions with the help of a simple example.

Let us suppose we have 3 buttons (for example, one experimentalist button and two subject response buttons), connected to trigger inputs 0, 1 and 2 respectively; pressed button caused **low** bit state.

1. To perform recording when any of three buttons is pressed:

Bit	0	1	2	…
High				
Low	▼	▼	▼	

2. To perform recording only when all three buttons is pressed:

Bit	0	1	2	…
High				
Low	▼			

AND

Bit	0	1	2	…
High				
Low		▼		

AND

Bit	0	1	2	…
High				
Low			▼	

3. To perform recording only when experimentalist button (0) is pressed, but both response buttons are not:

Bit	0	1	2	…
High				
Low	▼			

AND

Bit	0	1	2	…
High		▼		
Low				

AND

Bit	0	1	2	…
High			▼	
Low				

4. To perform recording only when experimentalist button (0) is pressed, and at least one of response buttons is released (or, in other word, disable recording when both response buttons are pressed)

Bit	0	1	2	…
High				
Low	▼			

AND

Bit	0	1	2	…
High		▼	▼	
Low				

Recording process.

File is recorded in portions of fixed and equal length (records). Only record length of 1000 ms (1 s) is compatible with ActiView BDF files.

New record starts after:

- Gate conditions switch from **false** to **true**. In this case the beginning of the new record will be synchronized with the change of trigger conditions.

OR

- User press 'Record' ( ) toolbar button

After writing each record to the disk program closes and reopens the output file (without changing 'Number of records' field in the header). If there were no changes in gate conditions (and user didn't press 'Pause' ( ) toolbar button, new record will be started immediately, providing continuous flow of data.

When using 'Plus' file format, ***Annotation*** channel of first record in a sequence will have following fields:

- time stamp (in seconds, from the beginning of the first record in a file), eq. **+10.324**
- record type – **NEW**
- the first value of the trigger channel in the record, in hexadecimal format with leading '0x' signature eq. **0xFFFE**
- bits of trigger channel that were set comparing to the previous value, in hexadecimal format with leading signature 'set', eq set0800
- bits of trigger channel that were cleared comparing to the previous value, in hexadecimal format with leading signature 'clr', eq clr0001

Annotation channel of **continued** recording will have fields:

- time stamp (in seconds, from the beginning of the first record in a file), eq. **+11.324**
- record type – **cnt**
- time of continued recording (offset from the beginning of previous NEW record to the beginning of the current record), eq **2.000**
- the first value of the trigger channel in the record, in hexadecimal format with leading '0x' signature eq. **0xFFFE**

Recording will be continued till:

- Gate condition switch from **true** to **false**

OR

- User press 'Pause' ( ) toolbar button.

In these cases recording will be stopped, but current record will be finished. If user will stop the acquisition process (by pressing  toolbar button), the acquisition will be stopped only after saving current record to the disk.

User can also select to disable writing of new records to the file by pressing 'Write disable' () toolbar button. In that case program will continue to analyze gate conditions and plot current records in snapshot views, but no real recording of new data to the file will be performed. As it will violate the recorded data continuity, the first record after depressing 'Write disable' () toolbar button will be marked as **NEW**.

5.2 Triggered input.

Triggered-input mode was designed to synchronize recording with external events, that is essential for ERP and EP research.

Trigger parameters.

Usually external trigger signal coincides in time with stimulus, but you may want to record some portion of data preceding it. For that purpose the length of triggered record can be divided to **pre-trigger** and **post-trigger** time. Limitations on these parameters are that together they will sum up to record length and each of them must me greater than (or equal to) zero (actually, one post-trigger point will be present in the data).

If you'll use for triggering some hardware with unstable contact (any simple button, for example), you may observe not a single-moment switching from one state to another, but a series of transitions that may result in a number of closely spaced (in time) triggers. To avoid such a situation, I've used not an advanced anti-flickering filtering, but a simple parameter – **trigger idle time**: when the program recognizes the trigger, the next one will be recognized only after that period of time. However, **trigger idle time** may be shorter **than record length** that will allow you to have an overlapped acquisition – but I have limited a number of overlapping windows to **four**. That means that if you have record length of 2000 ms and a sequence of trigger signals with 300 ms

interval, first four triggers will be recognized (and four records will be started) – but fifth trigger will be skipped.

Trigger conditions itself (**START Trigger field**) are defined not in terms of current bit state, but in terms of transition direction (**low-to-high** or **high-to-low**). You can also define additional Gate conditions: if the result of current trigger-channel data analysis against these conditions will be false, it will prevent trigger recognition. Returning to the example with buttons: I usually tell testee that he/she may press two response buttons simultaneously to pause the experiment:

Bit	0	1	2	…
Low -Hi	▼			
Hi- Low				

AND

Bit	0	1	2	…
High		▼	▼	
Low				

In special cases the program can also send custom messages to external (stimulation) programs to pause stimulation as well.

You can also specify additional bits in **Free input gate** field to enable continuous (non-triggered, gated) recording.

Recording process

In triggered recording 'Record' () and 'Pause' () toolbar buttons work independently of each other, and have different behavior.

'Pause' () toolbar button disables (when pressed) trigger recognition. The state of the button changes with the change of Trigger gate conditions.

'Record' () allows to record continuous (non-triggered) portions of data (for example, in the beginning, end or middle of experiment). The state of the button changes with the change of **Free input gate** conditions.

Thus, when both 'Record' () and 'Pause' () toolbar buttons are pressed, you'll have continuous gated recording**. If trigger recognition is enabled, first**

recognized trigger will switch continuous recording off. For example, it can be used to have a portion of continuous data at the beginning of the file, before the first stimulus.

When using 'Plus' file format, ***Annotation*** channel of triggered record will have following fields:

- time stamp of the record beginning (in seconds, from the beginning of the first record in a file), eq. **+10.324**
- record type – **Trg**
- time stamp of the trigger signal (in seconds, from the beginning of the first record in a file), eq. **+10.624**
- the value of the trigger channel at the moment of trigger recognition, in hexadecimal format with leading '0x' signature eq. **0x1403**
- bits of trigger channel that were set comparing to the previous value, in hexadecimal format with leading signature 'set', eq **set1400**
- bits of trigger channel that were cleared comparing to the previous value, in hexadecimal format with leading signature 'clr', eq **clr0020**

Two time stamps (start of record and trigger) will enable you to define exact stimulus position in the record and pre-trigger time.

The program analyses the trigger-channel values and extracts relevant trigger signals **before** the possible **downsampling** (decimation). So, the situation is possible when the program will recognize very short trigger signal – but you'll not see it on the screen and in the data. To avoid such a situation, you may uncheck **'Decimate trigger'** checkbox on the **Hardware** panel. In such a case, channels in resulting file will have different length. It does not violate the standards, but still may increase the problems with compatibility.

Finishing the recording.

After the end of experiment you must close the file (with File->Close menu command or  toolbar button. At that moment correct information about number of records will be written to the file header.

6 File viewing.

The program can be used as BDF and BDF+ file viewer.

After opening the file you'll see (in the leftmost panel of the main window) **File Info** tab with information from the file header and (in the same panel) **List** tab with the list of current file records. When working with 'original BDF' files, this list will contain only record numbers and time (not real, just the result of multiplication of record number to the record length). For BDF+ files, list will contain also:

- record type (**New**, **cnt**, or **Trg**)
- record timestamp
- values of trigger word, cleared and set bits

The data itself will be displayed in a Snapshot view window with corresponding dialog. You can open any number of additional snapshot windows (Window->New snapshot menu command or  toolbar button) with different parameters (displayed channels, referents, bipolars, filters, scales), place additional windows outside from the main program window (with  button in the title bar of snapshot window). Colors of 'Display' and 'Reference' maps in snapshot dialog will reflect **mean offset** of the electrodes for the current record.

File navigation may be performed by:

- entering desired record number into 'Record' field of snapshot dialog
- using 'First' ([<<]), 'Previous' ([<]), 'Next' ([>]) and 'Last' ([>>]) buttons near that field
- using keyboard:
    - <Up> and <Left> arrows – previous record
    - <Down> and <Right> arrows – next record
    - <PageUp> – 8 records to the beginning of the file
    - <PageDown> - 8 records to the end of the file
    - <Home> - first record
    - <End> - last record
- using list panel

Any views (but the first) may be in 'Locked' or 'Unlocked' state. When you'll change record number for the locked ( ) view, the same record will be displayed in all locked views, thus allowing you to examine the result of different referent schemas, filters etc. If you'll unlock the view (press button – and it will became), it will be possible to select different record in this view and thus compare different parts of the recording.

Though the program is single-document (but multiple-view) application, you can run multiple instances of the program to view and compare multiple files.

Now 'View' mode allows you only to inspect files and export data to Matlab. Future use supposes:

- channel removing
- rejection of bad records
- filtering
- conversion to EDF+ format
- may be, some elementary data analysis

7 Matlab Export

As my 'BDF+' file format is new and still developing, I've preferred not to write multiple import functions for multiple matlab packets, but to export my data directly to Matlab data files. I also preferred to use not new (structured and rather complex) file format of Matlab 5 (6, 7 etc) – but simple format of Matlab 4, that may be compatible with a number of 'Matlab-like' software.

Export to the matlab file may be perfomed by Tools->Matlab export menu command, or by pressing toolbar button. After that you'll be asked for a file name (default it the original file name with *.mat extension) and place (the program will remember your choice of Matlab data directory). Then the conversion will be performed (that may be a lengthy process), after which you'll see the dialog box, confirming that data was written to the selected file.

This dialog box also contains the reminder "Type **eval(CommandString)** to build 3-d array (channels*time*records)."

That reminder reflects the fact that Matlab 4 file format does not support structures and 3-d arrays, so the real data will be written as a set of different variables (one variable per record). Consequently, additional operation is required after loading such a file into matlab workspace. As this operation is rather simple, I'm writing it in the text variable **ComandString**, and **eval(CommandString)** command just performs this operation. Alternatively, you can store the next few lines in the Matlab *.m script file:

```
data = zeros(NumberOfChannels, length(times), NumberOfRecords);
for k = 1:NumberOfRecords,
    data(:,:,k) = eval(RecNames(k,:));
    clear(RecNames(k,:));
end;
clear RecNames;
```

The result of this operation is 3-d array data, that can be easily imported into any Matlab packet (for example, to the EEGLAB I use) – or can be analyzed with the help of your own scripts.

So, variables you'll see in workspace are:

data – 3-d array of EEG data, not including trigger

Trigger – 2-d array of trigger channel data (I've put it separately, but in case it has the same sample rate, it can be easily united with data array)

FileTitle – title of the original BDF file

LocalPatient and **LocalRecording** – text variables with the information from the file header

NumberOfChannels – it is ☺

NumberOfRecords – it is ☺

ChanNames – char array with channel names (electrode labels)

srate – sample rate, in Hz

times – record time axe

Additional variables for BDF+ files:

Annotation – char array of record annotations

Latency – double array of record latencies

Type – char array of record types

TWORD, **TSET**, **TCLR** – char array of trigger words, set and cleared bit values explained above

For further use char array must be converted to cell array (eq. **Type = cellstr(Type);** command), after that rather simple operations will produce arrays that may be used either for import (in EEGLAB or other software) – or for the use with custom-written scripts. For example:

```
Type = cellstr(Type);
isTrig = strncmp(Type,'Tr',2);

TW = cellstr(TWORD);
TW(~isTrig) = {' '};
iL = strncmp(TW,'8',1);
iR = strncmp(TW,'0',1);
i720 = strncmp(TW,'08',2) | strncmp(TW,'88',2);
i540 = strncmp(TW,'04',2) | strncmp(TW,'84',2);

speed = cell(NumberOfRecords,1);
speed(~isTrig) = {'0'};
speed(i720) = {'720'};
speed(i540) = {'540'};

dir = cell(NumberOfRecords,1);
dir(~isTrig) = {'0'};
dir(iL) = {'L'};
dir(iR) = {'R'};

ev = cell(NumberOfRecords,3);
ev(:,1) = num2cell(Latency);
ev(:,2) = speed;
ev(:,3) = dir;
```

In this little script I build events according to different stimuli types (moving with angle speeds of 720 and 540 degrees per second to the left or to the right) that were encoded in trigger channel. Indexes (**iL**, **i720** etc) I use in my own scripts (to extract interesting records from the data array), resulting **ev** array can be directly imported into the EEGLAB by **'Import Epoch Info'** command.