

About EPR Commander 4.0

EPR – electron paramagnetic resonance.

This program is intended for:

- spectrum registration from EPR-spectrometer;
- spectrum registration through [socket connection](#) (TCP/IP);
- spectrum output on graphic display, printer;
- spectrum export to graphic and text data files;
- mathematical processing of spectrum (smoothing, integration, differentiation, base line correction);
- measurement of spectrum line parameters (width, intensity, concentration).

The program is used together with the EPR-spectrometer, which is connected through a serial port or [socket connection](#). EPR Commander was designed for the Microsoft Windows 98 and Windows 2000 operation systems. The computer must have at least 32 Mbytes of memory for the program works properly.

Copyright © 2001, ADANI Co.

<http://www.adani.by>

Program using

The program has the multidocument interface, which permits to open any number of windows in one program (each spectrum has its own window). Each window has horizontal and vertical rules with automatic binding to a scrolled spectrum. A user can scroll spectrum by mouse or keyboard (pressing "**Home**", "**End**", "**Page Up**", "**Page Down**", "**Right arrow**", "**Left arrow**").

Mouse mode.

There are three modes for viewing a spectrum: determining position, scaling and dragging. You can select this mode from [View menu command](#) or from popup menu, which is available by clicking right mouse button.

When the mouse mode is set to "*position*", you can see a cursor of a current position (black vertical band) and the mouse cursor looks like a cross. User can change the cursor by pressing and holding the left mouse button or control it from keyboard: press "**Ctrl**"+"**Right arrow**" ("**Ctrl**"+"**Left arrow**") for small increment (decrement) or press "**Shift**"+"**Right arrow**" ("**Shift**"+"**Left arrow**") for large increment (decrement). You can see the data corresponding to the cursor (channel, magnetic field, amplitude) in the status bar.

When the mouse mode is set to "*scale*", the mouse cursor looks like the two arrows. To scale an image you should press and hold left mouse button, move the mouse pointer and release the mouse button. An image will scale and automatically scroll to the interesting site. If you select a part of image that you want to be scaled and still hold the mouse button, operation can be canceled by pressing the "**Esc**".








When the mouse mode is set to "*drag*", the mouse cursor looks like a hand. You can simply drag an image by pressing and holding left mouse button.

Base line correction mode.

This mode is available from [Transform menu command](#). When you run base line correction mode, the line through extreme points is drawn. You can add points through which base line passes doing the following: set mouse mode to "*position*", press "**Ctrl**" or "**Shift**" and click left mouse button or press "**Enter**". It is very easy to add points from keyboard. Having added enough points apply base line correction function or cancel the mode.




Menu "File"

Commands:

-  **"New"** – create a new file with default registration parameters and empty spectrum.
-  **"Open"** – load a spectrum from disk.
-  **"Save"** – save current file on disk.
-  **"Close"** – close an opened file.
-  **"Print"** – print current spectrum.
-  **"Export"** – convert spectrum file format to graphic file (*.pcx, *.bmp, *.emf) or to text data file (*.dat). Text data file represents the two columns of numbers, converted to a string. If you export spectrum in a graphic format, you can edit a size of a future graphic image. The size may be fixed (for example 700x500) or actual (a size of a window).
-  **"Send"** – send current spectrum by mail.
- **"Comment"** – set file comment. A comment is a string, which consist of 75 characters. It is used when a spectrum is printing.
- **"Exit"** – quit from program.




Menu "View"

Commands:

- **"Vertical stretch"** – stretch a spectrum vertically in a window or draw it in an actual size.
- **"Fit spectrum"** – fit a scaled spectrum horizontally in a window.
-  **"Spectrum line params"** – calculation of a line characteristics in a current fragment:
 - **"Width"** – is calculated as a difference between magnetic field value at minimum and maximum spectrum values;
 - **"H0"** – a middle value;
 - **"Intensity"** – maximum difference of amplitudes on a fragment;
 - **"Concentration"** – calculation of a double integral of a fragment.
-  **"Marks window"** – show marks window. When marks window is visible you can calculate the differences between two positions. Press **'Fix mark'** button to save the current position. The differences will be calculated relatively fixed mark when the current position changes.
-  **"Dash-board panels"** – show or hide panels with edit controls, indicators, buttons and progress bar.
- **"Mouse mode"** – change mouse mode: [position, scale and drag](#).



Menu "EPR"

Commands:

-  "Connect" – connect (disconnect) to a spectrometer or to a socket.
-  "Manual control" – open the dialog for manually controlling the spectrometer motors.
-  "Settings" – open the [settings dialog](#) to change the current setting or to load/save from/to another file.

Menu "Batch"

Commands:

-  "Run" – open and run batch file.
-  "Editor" – open batch file for [editing](#).

Menu "Transform"

Commands:

- "Smooth" – smoothing of a spectrum.
- "Integrate" – integration of a spectrum (means arithmetic spectrum value is subtracted previously).
- "Derive" – differentiation of a spectrum.
- "Flip" – flip spectrum horizontally or vertically.
- "Base line correction" – run [base line correction mode](#).
- "Apply" – apply base line correction function.
- "Cancel" – cancel [base line correction mode](#).
- "Restore" – restore the modified spectrum.

Registration

When you choose the "**Connect**" menu command the program checks the presence of a spectrometer, if a soft emulator option is not checked, or try to connect with the server. If there is no an error, the selected serial port becomes busy for other programs until you choose the "**Disconnect**" menu command.

Pressing "**Start registration**" button starts the spectrum registration and opens control panel with three buttons:

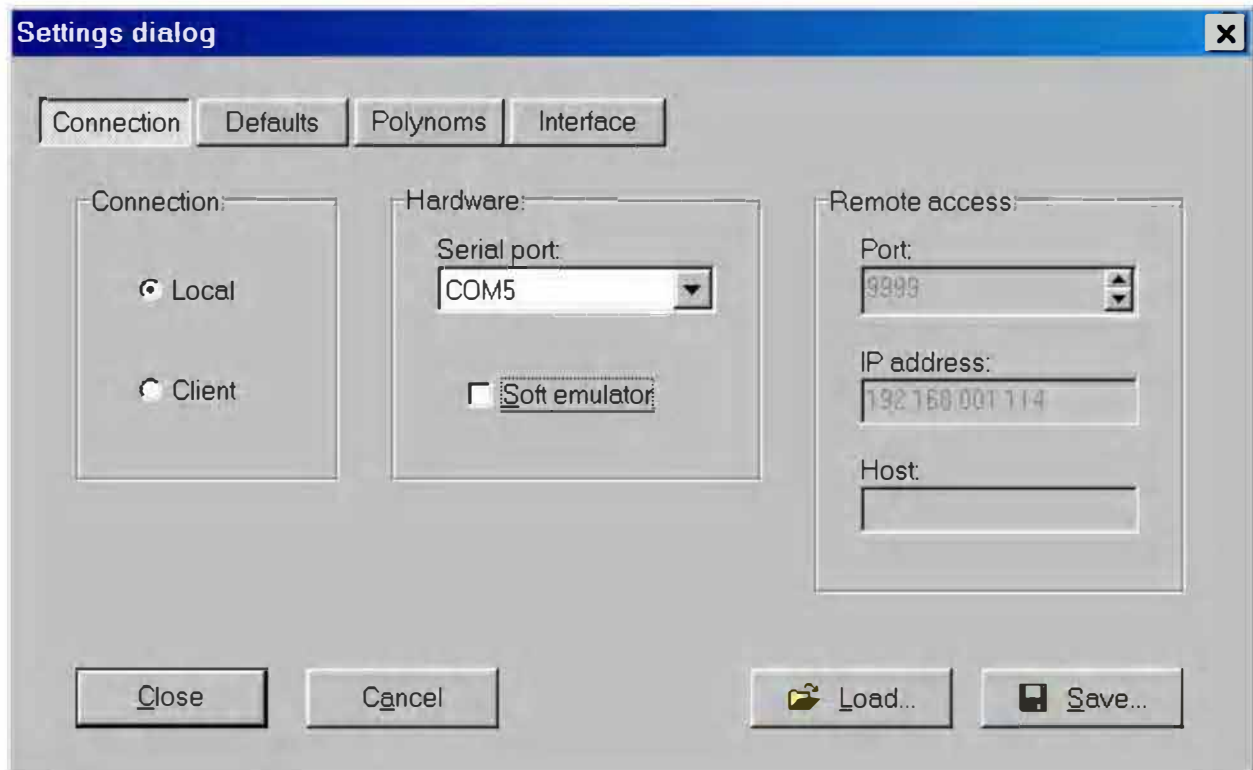
- "**Stop**" – stop registration.
- "**Pause**" – stop forming impulses without registration stoppage.
- "**Compensation**" – run compensation of a constant component of a signal.

You can send parameters without starting the registration by pressing "**Send parameters only**" button. "**Adjust UHF**" button runs UHF adjusting.

Indicators

There are three indicators on dashboard panel: AFC (automatic frequency control), detector current and field error. Their values are updated ten times per second. In order the field error indicator works properly the sweep width is assigned to zero every time the value is updated. So, when you send parameters to spectrometer you can see the deflection of an arrow on field indicator only for a short time.

Settings



Connection

"Local" – spectrum registration through a serial port.

"Client" – spectrum registration through a [socket connection](#) (TCP/IP).

Hardware

Port combo box displays a list of available local serial ports. Select the COM port to which the spectrometer is connected. Soft emulator option was created to debug the program. If there is no the spectrometer you can select this option and work with the program as the device is in presence.

Remote access

"Port" – the ID of the desired server connection.

"IP address" – the IP address of the server to which the socket should connect.

"Host" – a string containing the domain name and service of a particular system, such as
`http://www.adani.by`

Defaults

Default parameters are copied to the new spectrum.

Polynomial

Polynomial give an opportunity to set the nonlinear dependence of the parameters.

Load/save settings

At the beginning the program loads the settings from "eprcmd40.stg" file, which must be situated in the program's folder. You can modify and save settings in the other file. If you want to use saved setting simply load them. But remember that when the main window of the program is closed the current settings are saved in "eprcmd40.stg" file.

Socket connection

The program can register spectrum through a socket connection. Run the EPR Server 4.0 program on a remote machine.

The screenshot displays the EPR Server 4.0 software interface. The window title is "EPR Server 4.0".

Hardware section:

- Soft emulator
- Serial port: COM5
- Port: 3999
- Disconnect button (with a warning icon)
- Readiness:

Indicators section:

- AFC level: 0.0 V
- Detector current: 0.0 V
- Field error: 0.0 V

Parameters section:

- Center field, [G]: 3360.0
- Mod. amplitude, [mG]: 1000
- Gain value: 5
- Sweep time, [s]: 20
- Sweep width, [G]: 120.0
- Power attenuation, [dB]: 20
- Gain order: 2
- Stop button (with a red square icon)

Spectrum Plot: A blue line graph showing a derivative spectrum with a prominent peak and a smaller dip.

Channel/Block/Pass section:

- Channel: 1814/4095
- Block: 29/64
- Pass: 1/1
- Progress bar: 44%

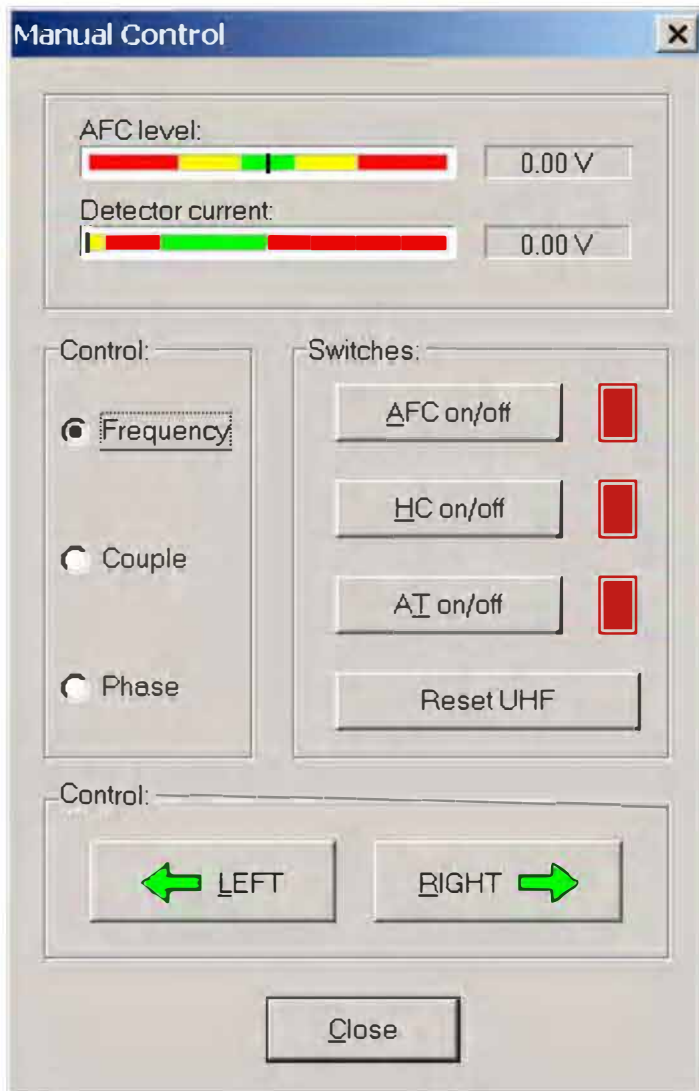
Status bar:

- Connected: SHURICK (192.168.1.114)
- Scanning...
- Sending... (493)

The serial port combo box displays a list of available local serial ports. Select the COM port to which the spectrometer is connected. Soft emulator option was created to debug the program. If there is no the spectrometer you can select this option and work with the program as the device is in presence. The port text control allows inputting the ID of the desired server connection. Press the "**Connect**" button to connect with the spectrometer and to start a listening server connection. Now the EPR Server 4.0 waits for a client. In the [settings dialog](#) of the EPR Commander 4.0 choose "Client" connection, set the IP address or host of the server to which the socket should connect, set the port value and close the dialog. Now use the program, as there is a local connection.

While registering a spectrum, the server sends every point of the spectrum to a socket. A client can control the registration process until the registration terminates. If the whole spectrum was registered, but only a part of it was send, the server sends the remains of the spectrum in one large block.

Manual control



Through this dialog you can adjust the microwave bridge.

Using the «**Frequency control**» set the «**Detector current**» indicator in the yellow or left red zone by pressing the «**Left**» and «**Right**» buttons.

Turn the «**AFC**» switch to the **ON** position. Upon doing this the «**AFC level**» indicator must be in the green zone.

Set the «**Detector current**» indicator in the 0V position by the «**Couple control**». If the detector current has only the minimum value, but does not reach 0V, set the flat cell more accurately and try again (in case of water contained samples).

When detector current reaches 0V, slide the «**HC**» switch to the **ON** position.

Set the «**Detector current**» indicator in the minimum position in the green zone by the «**Phase control**».

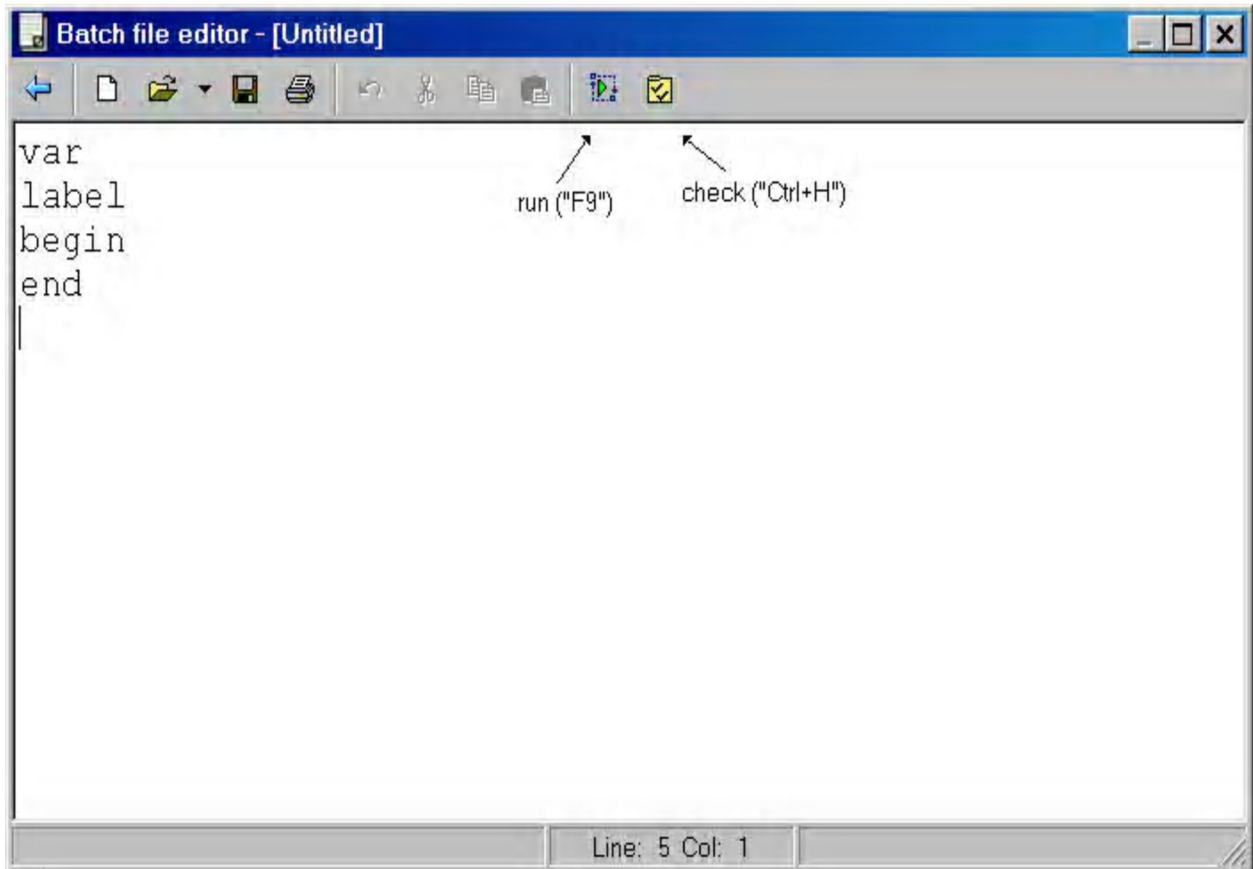
The voltage levels of the «**AFC level**» indicator must be equal whether the «**HC**» turn off or on.

«**AT**» switch turns on/off the automatic adjusting.

«**Reset UHF**» switch turns off the automatic control of the spectrometer.

Press «**Close**» button after adjusting the microwave bridge to close the dialog.

Batch editor



Batch files are very useful in the work. If you don't want to repeat the same operations manually you may write a batch file, run it and go for a cup of coffee.

The program has an integrated text editor called batch editor. After the code has been written you can check the batch file ("**Ctrl**"+"**H**") or run it ("**F9**").

[Batch language](#)

[Functions list](#)

Language

Every batch file must have a structure:

```
var
label
begin
end
```

Variables

After reserved word **var** the declaration of variables must follow. Every variable must be declared in a new string. There are three variable types: integer, real and string. Declaration of a variable includes the name, symbol ":", the type and a concluding symbol ";". The name of a variable must consist of the letters, numerals and a character "_". The first character of a name must be a letter.

Labels

After reserved word **label** the declaration of labels must follow. Every label must be declared in a new string. Declaration of a label includes the name and a concluding symbol ";".

Executing code

Executing code must be between two reserved words **begin** and **end**. There you can assign values to the variables, do operations, call procedures and functions and create cycles. Every line of code must have a concluding symbol ";". Every procedure, function, operation, assignment, goto statement must be situated in a separate string.

Assignment example:

```
MyVar1 = 10;                (MyVar1 – integer type)
Middle = 1006.35;          (Middle – real type)
FileName = "d:\spectr.spe"; (FileName – string type)
```

Operations addition (+), subtracting (-), multiplication (*), division (/) can use only two parameters, for example:

```
MyVar1 = MyVar1 + 46;
Middle = 45.78 * 24.8;
```

The following strings cause an error:

```
Range=10 + MyVar1 + 5;
Range=10-5*(7-x);
```

Compiler doesn't understand complicated expressions.

Procedure or function call must consist of procedure or function name and a list of parameters enclosed in brackets. For example:

```
A = Power(5, 2);           (function)
InputVar(FileName);       (procedure with var parameter)
OutputVar(Middle);        (procedure)
StartRegistration();       (procedure)
```

Function has a result value, procedure – not. Procedure with var parameter must have only a variables in parameters list, but not numeral constants.

If and goto statements allow to make cycles. These statements must be declared into a separate string.

Goto example:

```
goto-> label1;            (the "->" is necessary)
```

If statement example:

```
If (Count, label1, label2, label3);
If Count < 0, the control will be sent to the label1,
If Count = 0, the control will be sent to the label2,
If Count > 0, the control will be sent to the label3.
```

You can use REM statement. For example:

```
rem SaveFile(FileName);
```

Function list

Function list

Variant type means you can use any type (real or integer or string).

Functions:

1. Sin(x: Real): Real;
2. Cos(x: Real): Real;
3. Tan(x: Real): Real;
4. Exp(x: Real): Real;
5. Log(x: Real): Real;
6. Ln(x: Real): Real;
7. Sqrt(x: Real): Real;
8. Power(x, y: Real): Real;
9. Abs(x: Real): Real;
10. Int(x: Real): Real;
11. Round(x: Real): Integer;
12. IntToStr(Value: Integer): String;
13. FloatToStr(Value: Real): String;
14. StrToInt(S: String): Integer;
15. StrToFloat(S: String): Real;
16. Concat(S1: String; S2: String; S3: String): String;

Procedures

1. InputVar(var Value: Variant);
2. OutputVar(Value: Variant);
3. ShowVar(Value: Variant; Milliseconds: Integer);
4. Sleep(Milliseconds: Integer);
5. WinExec(CommandLine: String);
6. Reset(FileName: String);
7. Rewrite(FileName: String);
8. Close();
9. Read(var S: String);
10. Readln(var S: String);
11. Write(S: String);
12. Writeln(S: String);
13. NewFile();
14. OpenFile(FileName: String);
15. SaveFile(FileName: String);
16. CloseFile();
17. ExportAsGraphic(FileName: String; ExportType: Integer);
 If ExportType = 1 then spectrum will be exported as PCX file,
 If ExportType = 2 then spectrum will be exported as BMP file,
 If ExportType = 3 then spectrum will be exported as EMF file,
18. ExportAsText(FileName: String);
19. PrintFile();
20. SetFileComment(Comment: String);
21. InitPort();
22. ClosePort();
23. SendParams();
24. AdjustUHF();
25. StartRegistration();
26. SetParams(CenterField, SweepWidth, ModAmpl, PowerAtten, GainValue, GainOrder, SweepTime, PassNumber: Real);
27. GetParams(var Centerfield, SweepWidth, ModAmpl, PowerAtten, GainValue, GainOrder, SweepTime, PassNumber: Real);
28. Measure(var MinField, Maxfield, Width, HO, Concentr, MinIntensity, MaxIntensity, Intensity: Real);
29. SmoothV(alue: Real; Dimension: Integer);
 If Dimension = 0 then Value must be in Gauss, [G], If
 Dimension = 1 then Value must be in seconds, [s].
30. Integrate();
31. Derive();