

Analfm.exe Software

User Guide

Version 1.3c

Table of Contents

USB and COM Port Communication	3
Connecting the FM analyzer to a PC	3
Windows FM Analyzer Tool (Analfm.exe)	4
Basic features	4
First steps	4
Setup	5
DIP switches	5
Frequency deviation	6
Modulation power	6
MPX	7
RF Carrier	8
Bandscan	10
Radio Data System / Radio Broadcast Data System	11
FTP upload	12
MP3 recorder	12
Script files	13
Zoom option	23
Socket control	24
Task scheduler	25
Tips	25

USB and COM Port Communication

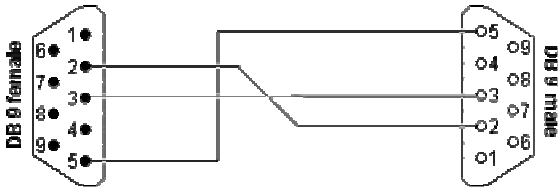
Connecting the FM analyzer to a PC

For configuration and control requirements a PC is connected to the FM analyzer via standard RS-232 interface provided by D-SUB9 female connector (DCE) on the FM analyzer side. On the PC side locate an unused COM port. If the free port exists in the form of a 25-pin connector, use a standard D-SUB9 (male) to D-SUB25 (female) adapter.

It’s preferable to use standard modem serial cable with one male and one female connector. Any USB to RS-232 adapter can be also used.

The P175 version allows direct USB connection. Using appropriate drivers the device will appear as a new COM port in the system so the method of software access is the same for both the RS-232 or USB connections.

FM analyzer	PC
2 (TxD)	2 (RxD)
3 (RxD)	3 (TxD)
5 (GND)	5 (GND)

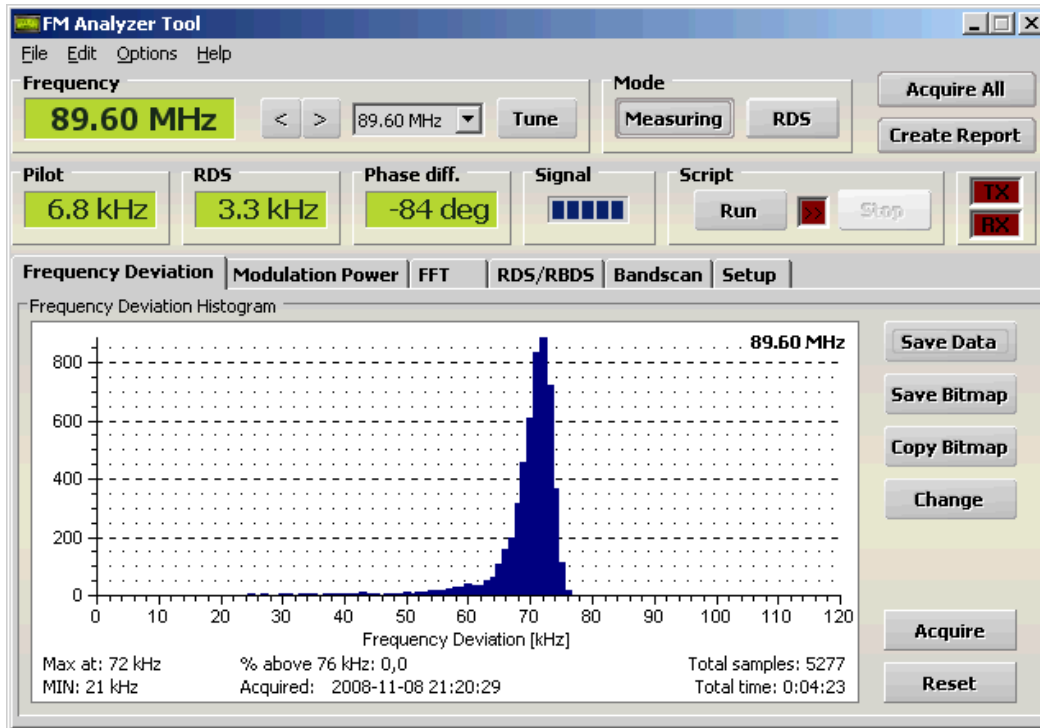


Configure the communication parameters as follows:

Transmission speed	19200 bps
Data bits	8
Parity	None
Stop bits	1
Flow control	None
Parity checking	No
Carrier detection	No

Windows FM Analyzer Tool (Analfm.exe)

The FM Analyzer Tool is a PC application primarily made for the P75/P175 automated control, data visualization and data archiving. The application is provided for free download on the website.



The FM Analyzer Tool main screen.

Basic features

- Graphical, text and Excel data output
- Native script language, fully automated data logging
- Built-in task scheduler
- FTP upload
- TCP/IP remote control
- Free download – no additional costs

First steps

1. In case of USB connection (or USB to RS-232 adapter) install the USB driver first.
Pure RS-232 connection never requires any driver.
2. Connect the device.
3. Run the analfm.exe file
4. Click on Setup card and choose the COM port where the FM analyzer is connected.
If you use USB connection, you may find the COM port number in the adapter's driver setup (usually in Windows Control panels).
5. Make sure the FM analyzer is powered.
6. Now you are ready.

Setup

Connection Type – Select the first option if you use RS-232 or USB connection.

Serial RS232:

RS232 Port – The COM port to which the FM analyzer is connected.

Ethernet TCP/IP:

TCP/IP Host – Host name or IP address of the remote site where the FM analyzer is connected.

TCP/IP Port – Use the same port as on the remote site.

Suitable ethernet-to-RS232 converter is available from Pira CZ at <http://pira.cz/eng/piracom.htm> – run this converter on the remote site.

Connect – Opens the COM port.

Disconnect – Closes the COM port.

Preferred Mode – The mode selected will be used on connect.

Deviation Limit – Frequency deviation limit that is valid in your country (usually 75 kHz).

Online Deviation and Signal – If checked, the deviation and signal values are updated continuously.

Debug mode – If checked, the script execution will stop on errors and error message will be displayed.

Register Script Files – Click if you wish to launch script files from Windows. It stores the information to Windows registry.

The application will remember these settings automatically. The settings are saved on application exit.

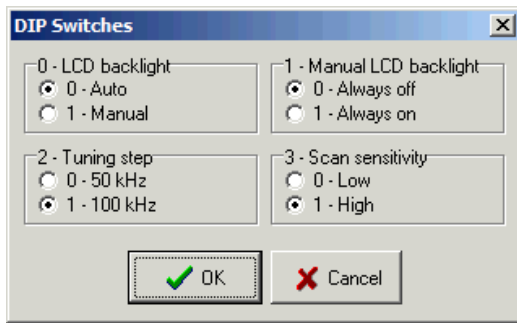
Save Settings – Saves actual FM analyzer settings to its internal EEPROM.

DIP switches

The “DIP switches” nostalgic term represents a way how to configure some special settings of the device. These settings are not directly accessible on the device but can be controlled from the PC.

Select Options/DIP switches from the main menu. To fix the changes do not forget to save them into EEPROM.

Select Options/DIP switches from the main menu. To fix the changes do not forget to save them into EEPROM.



Currently it's possible to configure LCD backlight, tuning step and scan sensitivity.

Frequency deviation

Save Data – Opens the save dialog and saves the data in *.csv text format readable by all editors, incl. MS Excel.

Save Bitmap – Opens the save dialog and saves the bitmap picture.

Copy Bitmap – Copies the bitmap picture to Windows clipboard.

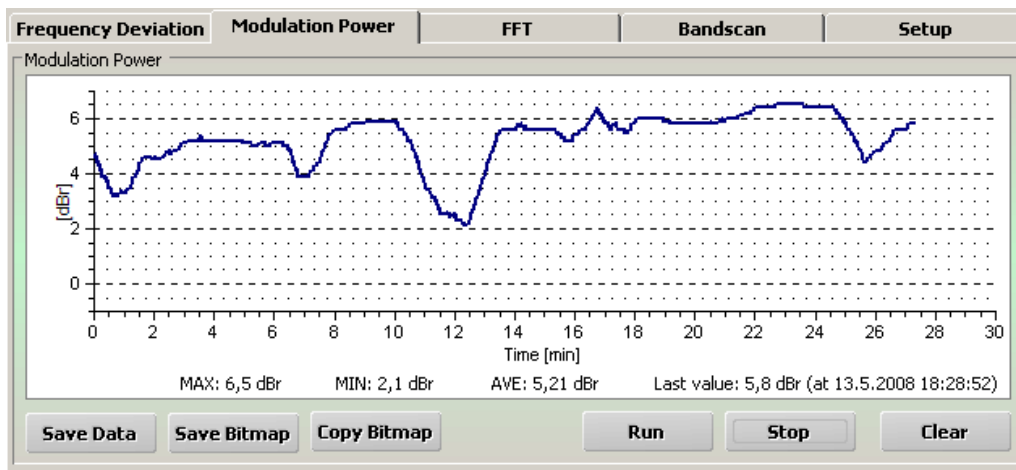
Acquire – Acquires the Frequency deviation data from the FM analyzer.

Change – Switches the display between histogram and accumulated distribution plot.

Reset – Clears all values. Equivalent to the Clear Data option in the FM analyzer's menu.

Modulation power

The application can collect the Modulation power data and show them as a function of time.



Run – Starts the measurement.

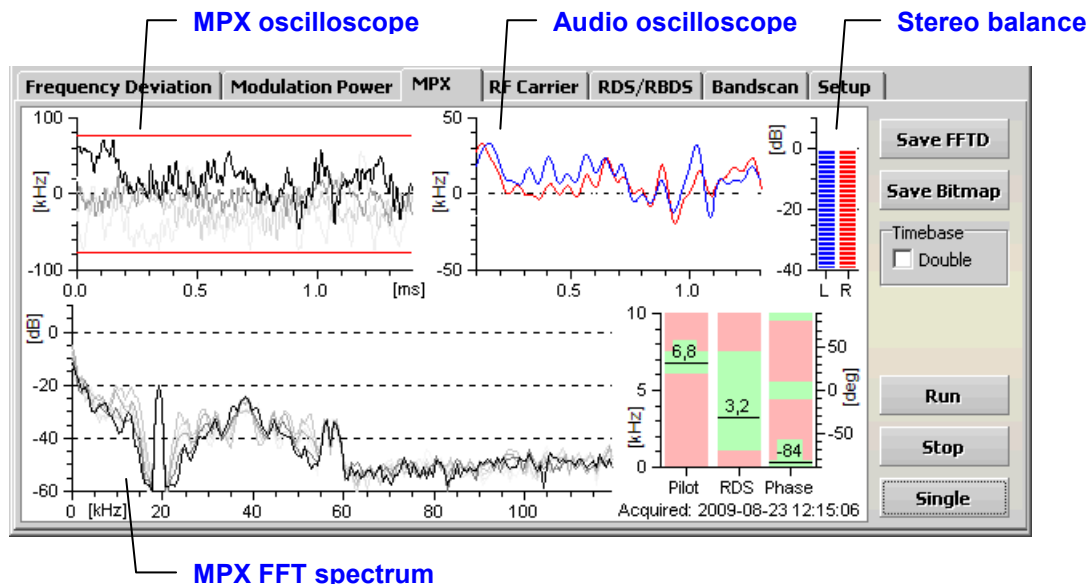
Stop – Stops the measurement.

Clear – Clears the Modulation power data and graphic window.

MPX

The MPX (multiplex) signal is the signal that goes into the modulation input of the FM transmitter. It typically consists of audio L+R, pilot tone 19 kHz, audio L-R modulated on 38 kHz (DSB-SC method) and additional components like RDS.

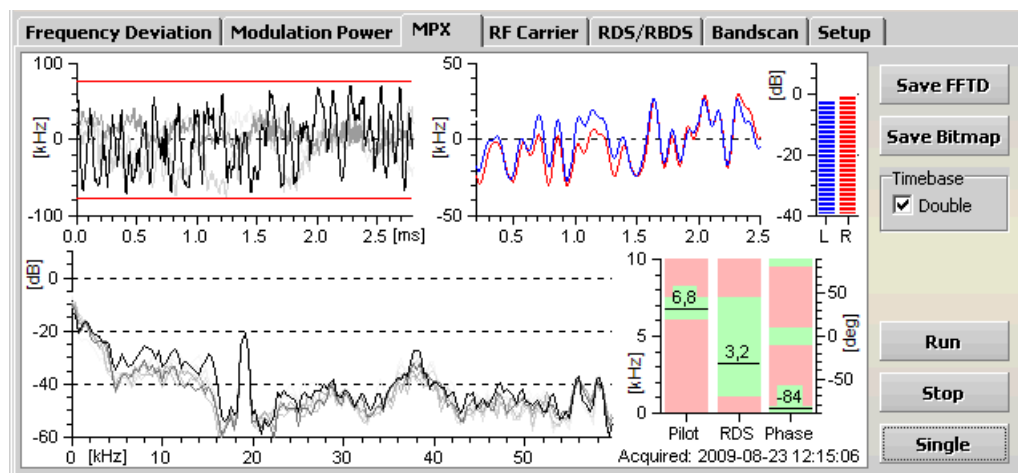
The MPX page includes MPX oscilloscope, audio oscilloscope for both channels, stereo balance meter, MPX spectrum and pilot and RDS levels and their phase relation. Average refresh rate is 3.5 times per second for the scope functions.



Run – Starts the measurement.

Stop – Stops the measurement.

Single – Gets the data in only one process.

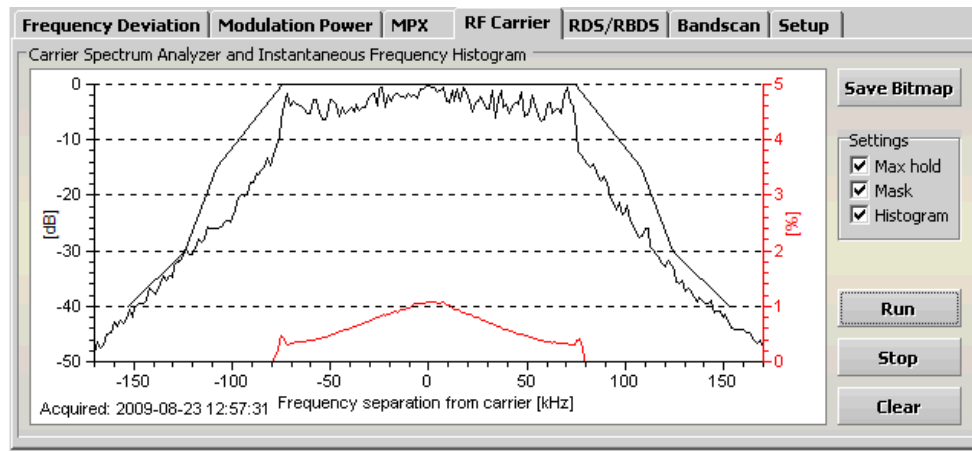


The Double timebase effect on X axes.

Note to FFT module: 0 dB ref. to 75 kHz frequency deviation for sine modulation signal.

RF Carrier

The RF Carrier page provides an access to the RF carrier spectrum and histogram of its instantaneous frequency.



Run – Starts the measurement.

Stop – Stops the measurement.

Clear – Clears the data and graphic window.

Mask – Shows the spectral mask as defined by REC 54-01 E.

Note: 0 dB ref. to full level of unmodulated RF carrier.

Note: To show the carrier spectrum, the Double timebase from the MPX page must be disabled (unchecked).

Spectrum mask based method and its limitations

This method was used in past as a verification to indicate whether the frequency deviation of an FM broadcasting station exceeds the limits. The method has mainly historical reason and was designed for analogue spectrum analyzers. It's a simple "go – no go" test based on a spectrum mask which **cannot** replace precise measurements of the peak frequency deviation that was described on previous pages.

The essence of the method is to determine whether the spectrum plot (max hold mode) is within the limits of the mask. Record the transmitter signal over a period of minimally 10 minutes (the more the better). The user must ensure that no measurement results are evaluated which have been distorted by impulse interference. For the same reason the measurement should be repeated twice.

The main disadvantage of the mask based method and its FFT implementation is that it may give positive result although the peak deviation is above 75 kHz. In other words, the transmission device may pass the mask based method test but this does not mean that the peak deviation is really within the limit.

The reason is that spectrum is always associated with energy averaged over some time. This time is considerably longer than modulation peaks that may occur. Single strong peak in the signal may still carry less energy than a series of weak peaks. This basic fact determines that spectrum based method cannot be reliably used for finding peak deviation. It is an approximation method that is not portable to other devices. Stations with a small number of peaks above 75 kHz can pass the mask based method. Applied to reality, stations that do not use strong audio compression and clipping can pass the mask based method although the peak frequency deviation is above the limit. It must be also noted that real peak deviation measurement covers 100 % of the time and suppresses interference impact while the spectrum based method doesn't.

Conclusions:

- The benefit of the mask based method is that compares real spectrum occupation caused by the station.
- If the spectrum plot is outside the mask on any frequency, the transmission device may be considered as overmodulating. If the spectrum plot is within the mask, additional peak frequency deviation measurement must be always made to verify if the station exceeds the peak deviation limit or not.

Advanced use of the RF carrier spectrum analysis

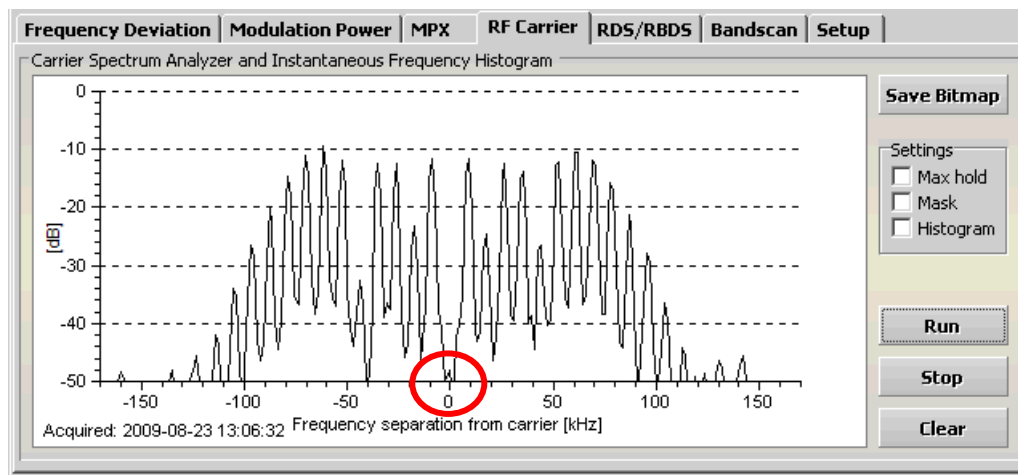
The RF carrier spectrum analysis is a very useful tool for very accurate measuring of FM deviation and modulation index and for making fast and accurate adjustments of FM transmitters on selected single modulation frequency (sine tone). The transmitter is adjusted to a precise frequency deviation with the aid of the spectrum analysis using the effect called "carrier zero" and selecting the appropriate modulating frequency f_m and modulation index β . This method requires a sine signal generator connected to the FM transmitter modulation input.

Modulation index is expressed as:

$$\beta = \Delta F / f_m$$

Following table gives the lowest order modulation indexes that result in carrier zero in the spectrum:

Order of carrier zero	1	2	3	4	5	6	n (n>6)
Modulation index β	2.40	5.52	8.65	11.79	14.93	18.07	$18.07 + \pi(n-6)$



In the figure above, a modulation frequency of 8.67 kHz and a modulation index of 8.65 (third carrier null) necessitate a carrier peak frequency deviation of 75 kHz.

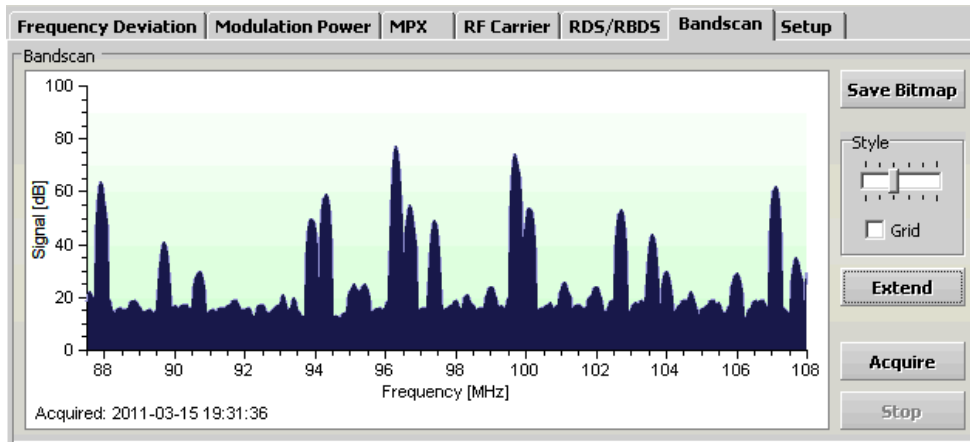
Since we can accurately set the modulation frequency on a signal generator and since the modulation index is also known accurately, the frequency deviation thus transmitted will be equally accurate (typically better than ± 0.3 kHz).

Recommended procedure for setting up a known deviation:

1. Select the required deviation ΔF .
2. Select a modulation index β that gives a modulating frequency commensurate with the normal modulation bandwidth of the transmitter to be tested.
3. Set the modulating frequency to $\Delta F / \beta$, and monitor the output spectrum of the FM transmitter on the RF Carrier page. Step up the amplitude of the modulating signal from zero level and stop when the carrier is at the desired order of zero.

Bandscan

This is an additional feature which shows quick graphic representation of the band occupation and estimated relative signal strength. By clicking on a station peak you may directly tune the station. This feature is not intended for coverage analysis nor signal strength measurement.



Notes (P75): 0 dB ref. to approx. 0.5 μ V on the antenna input. Max. dynamic range is <80 dB. The signal value on the Y-axis is estimated from the Signal and Noise Level values from LCD page 5. Precision is not specified for this function.

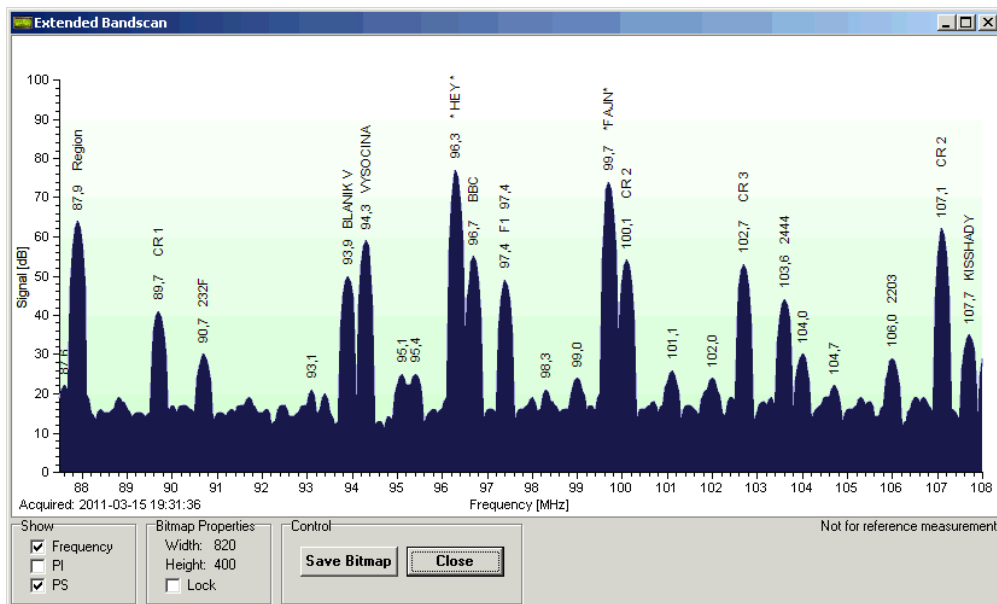
Note: The bandscan supports 100 kHz tuning step only.

Acquire – Acquires the Bandscan. The scanning is indicated in the application bottom line.

Stop – Aborts the operation.

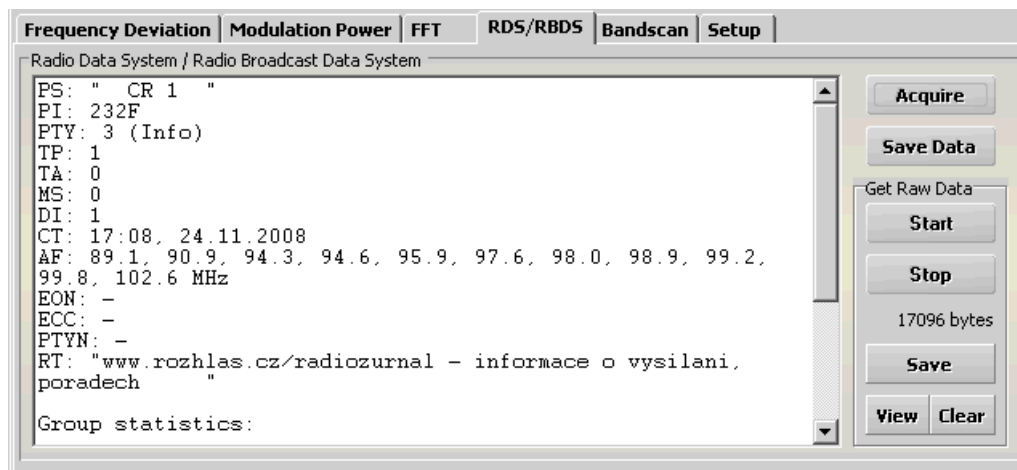
Style – Selects from various displaying styles.

Extend – Opens the bandscan in a new window allowing resizing and showing more details:



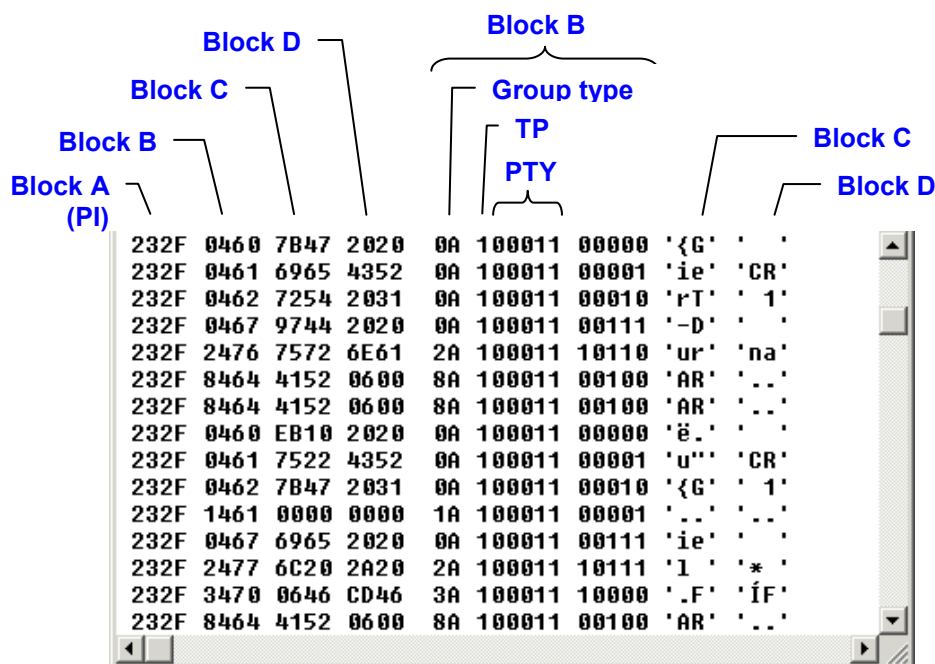
If RDS data is enabled for bandscan in Setup, the extended bandscan shows PI and PS for each station. However it may take a few minutes before the bandscan finishes.

Radio Data System / Radio Broadcast Data System



Acquire – Acquires the RDS data. To see actual data, the Mode must be set to RDS. In addition, error or warning messages are included in the report if any problem is found in the RDS settings.

Get Raw Data – Captures raw RDS data in real-time and allows saving them to a text file in following format:



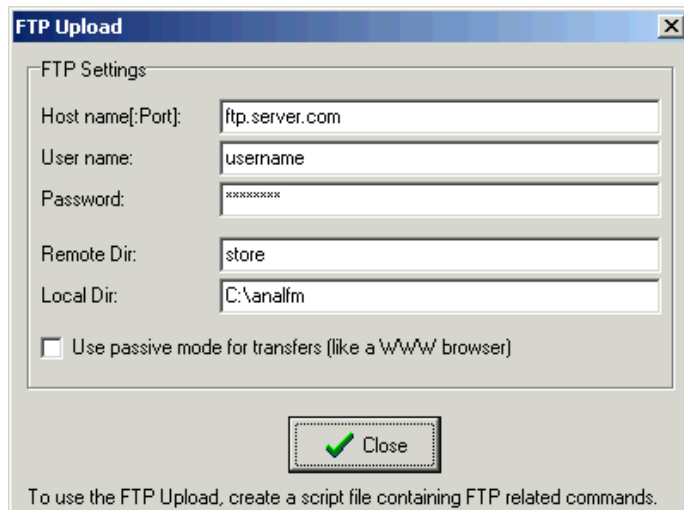
Maximum capture buffer length is 128 kB. Each RDS group consists of 8 bytes (4 blocks of 2 bytes each). The file is readable in any text viewer and provides full possibility of RDS stream analysis. All content is showed in hexadecimal representation and where applicable, additional decimal, binary or ASCII representation is used. The file starts with older RDS groups and ends with last RDS group.

Note: For complex real-time RDS analysis, please download the RDS Spy software: <http://rdsspy.com>

FTP upload

File Transfer Protocol (FTP) is a network protocol used to transfer data from one computer to another through a network such as the Internet. This operation can be fully automated.

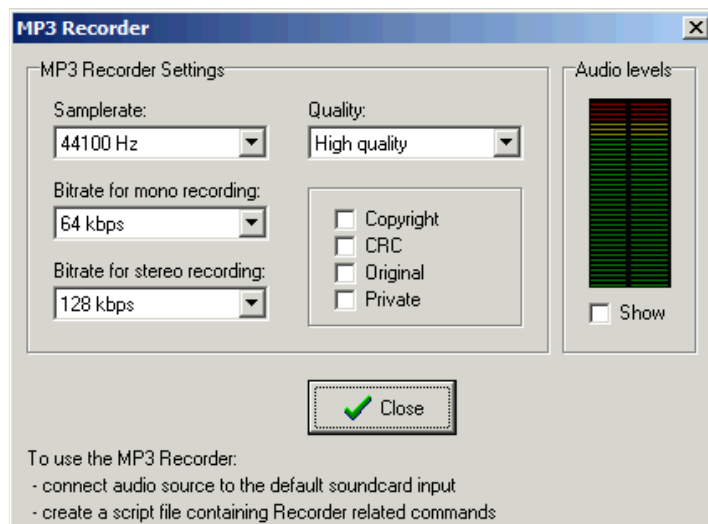
The FTP Upload feature needs to be set before first use. In main menu, choose Options/FTP Upload.



To use the FTP upload, create a script file containing FTP related commands (see the list of commands and the examples). This feature can be also combined with Task scheduler to create fully automated online system of monitoring with www based output.

MP3 recorder

The MP3 Recorder feature needs to be set before first use. In main menu, choose Options/MP3 Recorder.



To use the MP3 recorder, connect the audio source to the default soundcard input using any metallic audio cable. Create a script file containing MP3 recorder related commands (see the list of commands and the examples). This feature can be also combined with Task scheduler to create fully automated system of recording.

Make sure the file lame_enc.dll is placed in the application folder. This file can be downloaded from the net.

Important! Recording audio content may violate copyright laws!

Script files

The application allows running script files. You can fully automate the measurements and data storage. Presence of this feature and its precision and possibilities may be compared with top systems from this branch.
The script files extension is .fms.

To run the script, choose corresponding option in the application or use command line option, for example:

```
analfm.exe measure.fms
```

The application also allows registering the .fms extension so you can run the script files directly or using any task scheduler application for this purpose.

Each line in the script file may contain one command. The command interpreter is not case-sensitive. Max. line length is 1024 characters, max. file length is 1024 lines.

List of commands

Command	Meaning	Example of use
---------	---------	----------------

General purpose and application control:

connect	Open a COM port for communication or connect to a remote site. Useful mainly for switching between more P75/P175 units driven from a PC.	connect(1) connect(localhost:23) connect(127.0.0.1:23)
send	Send any string to serial port.	send(?F) send(*5) send(*B) send(?C)
disconnect	Close the connection.	disconnect
exit	Exit the Anal-FM application.	exit
page.show	Switches the Anal-FM display to the page specified as parameter (0 to 6).	page.show(2)
showmessage	Show a dialogue window with user defined message. Useful for debug purposes.	set(variable1) showmessage(Variable value: %variable1)
settrayiconhint	Set the application icon hint in the system tray.	settrayiconhint(Monitoring %frequency)
statusbartext	Set the application status bar text.	statusbartext(Monitoring %frequency)

Measurements:

tune	Tune to a frequency specified in MHz.	tune(89.6)
setmode	Set the operating mode to Measuring (0) or RDS (1).	setmode(0)
getpilot	Get pilot deviation value.	getpilot

getrds	Get RDS deviation value.	getrds
getquality	Get signal quality value.	getquality
getphase	Get pilot-to-RDS phase difference value.	getphase
frequencydeviation.acquire	Acquire actual frequency deviation data (incl. histogram).	frequencydeviation.acquire
frequencydeviation.histogram	Switch the display between histogram (1) and accumulated distribution plot (0).	frequencydeviation.histogram(1)
modulationpower.run	Run the modulation power measurement.	modulationpower.run
modulationpower.stop	Stop the modulation power measurement.	modulationpower.stop
modulationpower.clear	Clear the modulation power data.	modulationpower.clear
mpx.single	Get actual MPX and RF data.	mpx.single
mpx.run	Run the MPX and RF Carrier related measurement.	mpx.run
mpx.stop	Stop the MPX and RF Carrier related measurement.	mpx.stop
mpx.clear	Clear the MPX data.	mpx.clear
mpx.doubletimebase	Set normal (0) or double (1) timebase.	mpx.doubletimebase(0)
rf.run	Equal to mpx.run	rf.run
rf.stop	Equal to mpx.stop	rf.stop
rf.clear	Clear the RF data.	rf.clear
rf.maxhold	RF spectrum max hold mode.	rf.maxhold(1)
rf.mask	Spectral mask display.	rf.mask(1)
rf.histogram	Instantaneous RF carrier histogram display.	rf.histogram(1)
rds.getdata	Get RDS data.	getrdsdata
rds.getstatistics	Get RDS group statistics.	getrdsstatistics
rawdata.start	Start to capture raw RDS data.	rawdata.start
rawdata.stop	Stop capturing RDS data.	rawdata.stop
rawdata.clear	Clear the raw RDS data buffer.	rawdata.clear
bandscan.acquire	Acquire actual bandscan.	bandscan.acquire
bandscan.style	Select the bandscan style. Value 0 (zero) corresponds with the first option.	bandscan.style(3)
imf.setasnormal	Set actual 2nd IF frequency as a normal. Allows to measure carrier offsets and to show this value in Report.	imf.setasnormal
imf.get	Get actual 2nd IF frequency.	imf.get

Windows clipboard:

frequencydeviation.copybitmap	Copy the frequency deviation bitmap to clipboard.	frequencydeviation.copybitmap
modulationpower.copybitmap	Copy the modulation power bitmap to clipboard.	modulationpower.copybitmap
mpx.copybitmap	Copy the MPX bitmap to clipboard.	mpx.copybitmap
rf.copybitmap	Copy the RF bitmap.	rf.copybitmap
bandscan.copybitmap	Copy the bandscan bitmap to clipboard.	bandscan.copybitmap

Working with files:

savetext	Save a text to a file (overwrite).	savetext(%date.txt,OK)
appendtext	Save a text to a file (append).	appendtext(%date.log,%crPilot: %pilot)
frequencydeviation.savedata	Save the frequency deviation data.	frequencydeviation.savedata(histogram.txt)
frequencydeviation.savebitmap	Save the frequency deviation bitmap.	frequencydeviation.savebitmap(%date.bmp)
modulationpower.savebitmap	Save the modulation power bitmap.	modulationpower.savebitmap(pm.bmp)
modulationpower.savedata	Save the modulation power data.	modulationpower.savedata(pm.txt)
mpx.savedata	Save the FFT data.	mpx.savedata(fft.txt)
mpx.savebitmap	Save the MPX bitmap.	mpx.savebitmap(mpx.jpg)
rf.savebitmap	Save the RF bitmap.	rf.savebitmap(rf.jpg)
bandscan.savebitmap	Save the bandscan bitmap.	bandscan.savebitmap(%date.bmp)
extendedbandscan.savebitmap	Save the extended bandscan bitmap.	extendedbandscan.savebitmap(band.jpg)
rds.savedata	Save the RDS data (incl. group statistics)	rds.savedata(rds data %rdsps.txt)
rawdata.save	Save the raw RDS data.	rawdata.save(rds raw data.txt)
createreport	Create and save the report.	createreport(report %freq.txt)
recorder.run	Run the MP3 recording to the file and from the channel(s) specified. The channel(s) may be left, right, mono or stereo.	recorder.run(c:\rec.mp3,mono)
recorder.stop	Stop the MP3 recorder.	recorder.stop
checkfilename	Remove invalid filename characters from second argument, store result to the variable in first argument.	checkfilename(fn,%freq %rdsps) savetext(C:\%fn.txt,%rdsps%cr%rdsps)

Script flow control:

execute	Execute any application or open any document. Optional parameter(s) may be specified. Does not wait until the application terminates.	execute(C:\batch.bat) execute(notepad.exe,text.txt) execute(command,/c md c:\data)
sleep	Stop executing next commands for a time specified in seconds.	sleep(30)
goto	Go to another line of the script. The line is identified by a unique label followed by pit-pair. In the goto argument the pit-pair is omitted.	repeat: ... goto(repeat)
call	Call subroutine. The first line of the subroutine is identified by a unique label followed by pit-pair. In the call argument the pit-pair is omitted. Nesting of multiple subroutines calls is possible.	call(myproc) ... stop myproc: ... return
return	Return from subroutine. Continue on next command after last call.	
if	Allows for conditional execution of script fragments. Only one condition is allowed per one if command. Only these operators are allowed: >, <, =, >=, <=, !=. Operators like <i>and</i> , <i>or</i> , <i>not</i> are not permitted, however nesting of multiple if commands is possible.	getquality if (%quality>3) call(myproc) endif
stop	Don't execute next commands in the script file. Stop processing the script.	stop

Working with variables:

set	Initiate a variable (general type) and assigns a value to the variable. If no value is specified, a value of 0 is assigned. Any variable that is not set first has permanently a value of -1. If using more variables at a time, the user must ensure that no variable name starts with another variable name. When need the variable value, a prefix % must be added. The variables remain in the application memory until its exit.	<pre> set(variable1) set(msg,Counting...) repeat: showmessage(%msg %variable1) inc(variable1) if (%variable1<5) goto(repeat) endif </pre>
inc	If there's a number stored in the variable, Increase the variable by specified value. This value may be negative or floating point as well. If no value is specified, the variable is increased by 1. Primary purpose of the set and inc commands is to create cycles and enumerate measurement results.	
input	Permit the user to enter a text (or number). If any text is entered, a variable inputvalid is set to 1 and variable inputtext contains the text.	<pre> input(Enter your location) if (%inputvalid=1) checkfilename(fn,%inputtext) bandscan.savebitmap(%fn %time.jpg) endif </pre>

Working with FTP:

ftpconnect	Connect to the FTP server specified in FTP Upload settings.	ftpconnect
ftpput	Put the file on the server. Optionally the filename on the server may differ from the local filename.	<pre> ftpput(%date.log) ftpput(%freq.jpg,histogram.jpg) </pre>
ftpdconnect	Disconnect from the FTP server.	ftpdconnect

Socket control:

get	Redirects the string to the socket control client. If terminal echo is enabled, the string is followed by CR+LF.	<pre> get(L: %lmaxdb, R: %rmaxdb) get(%pilot) </pre>
+++	Breaks any running script. The socket control related command. Must be also validated by <Enter>.	+++

Following tags may be used to create dynamic content or insert special characters:

Tag	Meaning
%time	Actual time
%date	Actual date
%frequency	Frequency in long format (87.5 MHz)
%freq	Frequency in compact format (08750)
%pilot	Pilot deviation
%rds	RDS deviation
%phase	Pilot-to-RDS phase difference
%quality	Signal quality (if previously get by getquality)
%abovelimit	% above dev. limit (histogram function)
%maxat	MAX at (histogram function)
%totalsamples	Total number of samples (histogram function)
%min	MIN
%maxhold	MAX (the value updated every second)
%tsmaxhold	10 sec. MAX (if previously get by ?X)
%pmlast	Modulation power – last value
%pmmax	Modulation power – MAX value
%pmmin	Modulation power – MIN value
%pmave	Modulation power – AVE value
%lmaxdb	Max. dB peak in left channel (card MPX)
%rmaxdb	Max. dB peak in right channel (card MPX)
%balance	Linear channel balance R/L (if previously get by ?C)
%cr	CR+LF (end of line)
%tab	Tabulator
%rdspi	RDS PI
%rdsps	RDS PS
%rdsrt	RDS RT
%rdspty	RDS PTY
%rdsms	RDS M/S
%rdstp	RDS TP
%rdsta	RDS TA
%rdsdi	RDS DI
%rdsaf	RDS AF
%rdseon	RDS EON
%rdsecc	RDS ECC
%rdsptyn	RDS PTYN
%rdsstat1	RDS Group statistics 0A to 7A
%rdsstat2	RDS Group statistics 8A to 15A
%rdsstat3	RDS Group statistics 0B to 7B
%rdsstat4	RDS Group statistics 8B to 15B
%rdserr	RDS Settings Errors and Warnings

%stationcount	Total number of stations detected in bandscan
%stationindex	Points to station of interest from all stations detected in bandscan. Can be 1 to %stationcount.
%station.frequency	Frequency of the station pointed by the %stationindex
%station.level	Reception level of the station pointed by the %stationindex
%station.noise	Reception noise of the station pointed by the %stationindex
%station.rds_pi	RDS PI of the station pointed by the %stationindex
%station.rds_ps	RDS PS of the station pointed by the %stationindex
%inputtext	The text entered using input command query
%inputvalid	Contains 1 if any text was entered using the input command query, otherwise contains 0.

These tags are reserved. This means that no variable name may contain any of the reserved tags. The same rule applies for all commands and labels.

To avoid showing of unit, place _ behind the tag, for example %maxhold will show 75.0 kHz but %maxhold_ will show 75.0 only.

Note: For automated monitoring without need of creating any scripts, please download the FM Guard software: http://pira.cz/fm_broadcast_analyzer/

Script file example 1

(Measure a station, save report, exit the FM Analyzer Tool application.)

```
setmode(0)
tune(100.5)
sleep(600)
frequencydeviation.acquire
getpilot
getrds
getphase
createreport(c:\reports\%date %time – %frequency.txt)
exit
```

Script file example 2

(Measure a station, save some values to a log file every minute in an infinite loop.)

```
setmode(0)
tune(91.9)
savetext(log.txt,%frequency log file starting at %date %time%cr)
repeat:
sleep(60)
getpilot
getrds
appendtext(log.txt,<%date %time> Pilot deviation: %pilot, RDS deviation: %rds%cr)
goto(repeat)
```

Script file example 3

(Periodically tune to stations of interest, append some RDS information to text files, each station one file)

```
setmode(1)
repeat:
tune(91.9)
call(savedata)
tune(95.8)
call(savedata)
tune(104.2)
call(savedata)
goto(repeat)

savedata:
sleep(180)
rds.getdata
appendtext(%freq.txt,%date %time%cr)
appendtext(%freq.txt,PS: %rdsps, PTY: %rdspty, TA: %rdsta,%crRT: %rdsrt%cr%cr)
return
```

Script file example 4

(Tune to a station, get RDS data and save them.)

```
setmode(1)
tune(91.9)
sleep(60)
rds.getdata
rds.getstatistics
rds.savedata(rds %freq – %rdsps.txt)
```

Script file example 5

(Get the carrier offset in the Report.)

```
tune(87.9)           ;this will be the reference station
sleep(3)
imf.setasnormal

tune(107.1)
sleep(3)
imf.get

createreport(C:\report – 10710.txt)

tune(104.3)
sleep(3)
imf.get

createreport(C:\report – 10430.txt)
```

Script file example 6

(Acquire a bandscan, measure stations of interest and put all results to ftp server. The FTP Upload must be set before, incl. local folder c:\analfm\data\ . The folders must be created before.)

```
bandscan.acquire
bandscan.savebitmap(c:\analfm\data\bandscan.jpg)

ftpconnect
ftpput(bandscan.jpg)
ftpdconnect

tune(89.1)
call(processdata)
tune(91.4)
call(processdata)
tune(92.2)
call(processdata)
tune(97.5)
call(processdata)
tune(104.1)
call(processdata)
stop

processdata:
setmode(0)
sleep(60)
modulationpower.run
sleep(900)
modulationpower.stop
modulationpower.savebitmap(c:\analfm\data\pm%freq.jpg)
frequencydeviation.acquire
frequencydeviation.histogram(1)
frequencydeviation.savebitmap(c:\analfm\data\hi%freq.jpg)
frequencydeviation.histogram(0)
frequencydeviation.savebitmap(c:\analfm\data\dp%freq.jpg)
mpx.run
sleep(10)
mpx.stop
mpx.savebitmap(c:\analfm\data\mx%freq.jpg)
setmode(1)
sleep(60)
rds.getdata
rds.getstatistics
rds.savedata(c:\analfm\data\rd%freq.txt)
createreport(c:\analfm\data\rp%freq.txt)

ftpconnect
ftpput(hi%freq.jpg)
ftpput(dp%freq.jpg)
ftpput(mx%freq.jpg)
ftpput(pm%freq.jpg)
ftpput(rd%freq.txt)
ftpput(rp%freq.txt)
ftpdconnect

return
```

Script file example 7

(Permanently monitor the station, send email if there's no signal or no audio for more than one minute.)

```
tune(106.2)
setmode(0)
set(signal_counter)
set(audio_counter)

repeat:
getquality
send(?X)

if (%quality<3)
    inc(signal_counter)
    goto(stage2)
endif
set(signal_counter)

stage2:
if (%tsmaxhold<25)
    inc(audio_counter)
    goto(stage3)
endif
set(audio_counter)

stage3:
if (%signal_counter=5)
    call(send_email)
endif
if (%audio_counter=5)
    call(send_email)
endif

sleep(10)
goto(repeat)

send_email:
execute(blatt.exe,-u a -pw b -server c -f d -to f -subject g -body "h")
return
```

Notes:

- *This script requires Blat command line emailer placed in the anal_fm folder. The Blat can be downloaded from <http://www.blat.net/>. Replace the letters a to h by the parameters required. For the Blat setting and usage follow the documentation of this utility. This is a 3rd party utility, we do not provide any support for it.*

Script file example 8

(Save stations from the bandscan to an Excel file.)

```
bandscan.acquire
set(stationindex,1)
set(filename,Bandscan %date %time.csv)
savetext(%filename,"Frequency";"Level [dBuV]";"RDS PI";"RDS PS"%cr)
rpt:
if (%stationindex>%stationcount)
    stop
endif
appendtext(%filename,"%station.frequency";"%station.level";"%station.rdspi";"%station.rdsps"%cr)
inc(stationindex)
goto(rpt)
```

Script file example 9

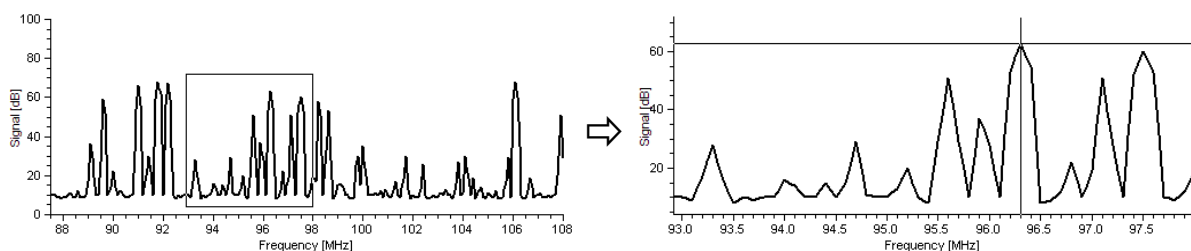
(Measure stations from the bandscan, save histograms if there's enough of samples.)

```
bandscan.acquire
set(stationindex,1)
setmode(0)

rpt:
if (%stationindex>%stationcount)
    statusbartext(Done.)
    stop
endif
statusbartext(Measuring %station.frequency %station.rdsps ...)
tune(%station.frequency)
sleep(120)
frequencydeviation.acquire
if (%totalsamples<20)
    goto(skip)
endif
checkfilename(filename,%freq %station.rdsps)
frequencydeviation.savebitmap(%filename.jpg)
skip:
inc(stationindex)
goto(rpt)
```

Zoom option

With the zoom option you can zoom in on your graph, or in other words enlarge a certain part. The zoom option uses mouse and can be applied to all graphs in the application.



Press the CTRL key and keep it pressed, click the mouse in the graph area and pull a rectangle. To unzoom the graph select another card or simply press the CTRL key and click with the mouse on the graph without moving it.

Socket control

The socket control feature is a simple method to control the application from another via TCP/IP sockets. Before attempting to use this function its parameters must be set first in Options/Socket Control:

Enabled – Enables internal Socket control server.

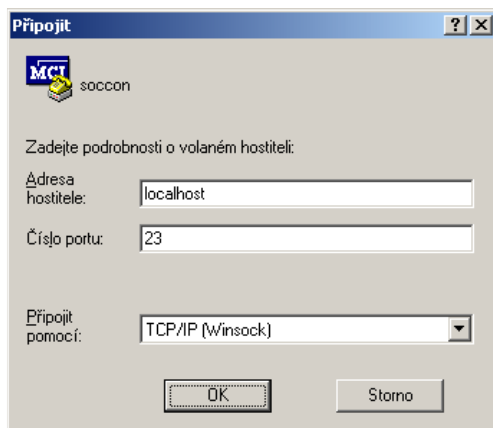
Terminal Echo – If enabled, all incoming characters are sent back.

TCP/IP Port – The port number on which the server listens for a client.

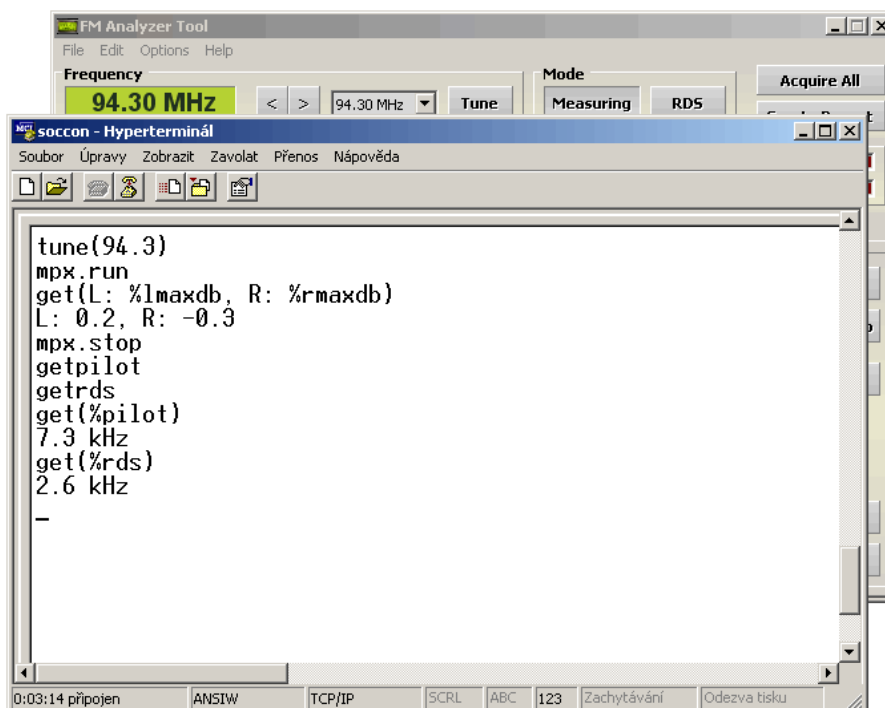
The Socket Control accepts all script commands and moreover it provides also backward channel with the aid of command `get` (see the list of commands). Each command entered is considered as a separate script. If a script file is being processed, all commands entered by the Socket Control are placed in queue and processed afterwards. The exception from this rule is command `+++` which breaks any running script.

How to begin?

One of the best illustrations of use is to control the application from Windows Hyperterminal. Run this application, select TCP/IP connection type and fill the server address and port.



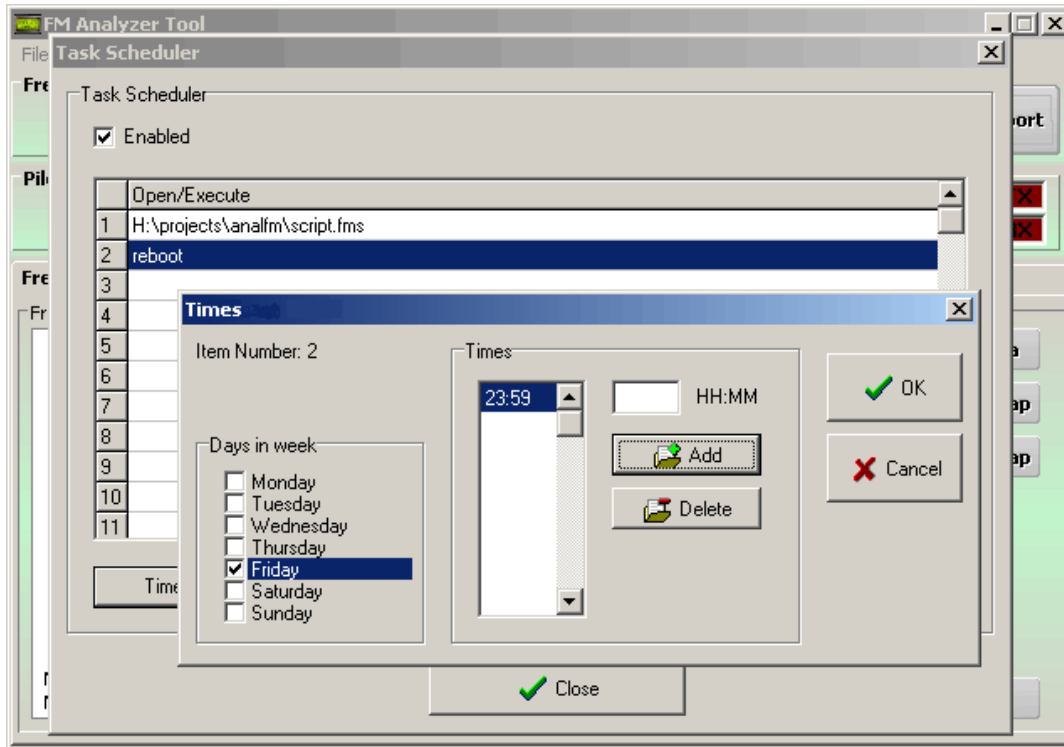
Click OK to connect. Make sure the Socket control feature is enabled and the application is running.



Task scheduler

Built-in task scheduler allows to schedule script files (*.fms) execution and any other application execution or document opening. You may also use it to restart/shutdown the PC, to send any string to COM port or to exit the application.

To open the task scheduler, select Options/Task Scheduler in the main menu.



Enabled – Enables the Task Scheduler.

Times – Allows you to specify days in week and times in the day when the task will be executed.

Delete – Removes the selected task.

Find File – Easy way how to find a file/application to be executed. This may be the script file or any other file.

These keywords are also accepted in the task line:

exit – Exits the Anal-FM application

reboot – Reboots the PC

shutdown – Shutdowns the PC

send: – Sends any RDS command, for example SEND:096200*F

Tips

- Place the application into a folder where write access is enabled.
- Place a link to the analfm.exe in Start/Programs/Start-up folder.
- Set read-only attribute for analfm.ini to avoid unwanted configuration changes.
- To run more instances of the application place each exe file to another folder and create a file named “multi.set” in each folder. This file may be empty or any content.