FM Scope

Windows control software for FM analyzers

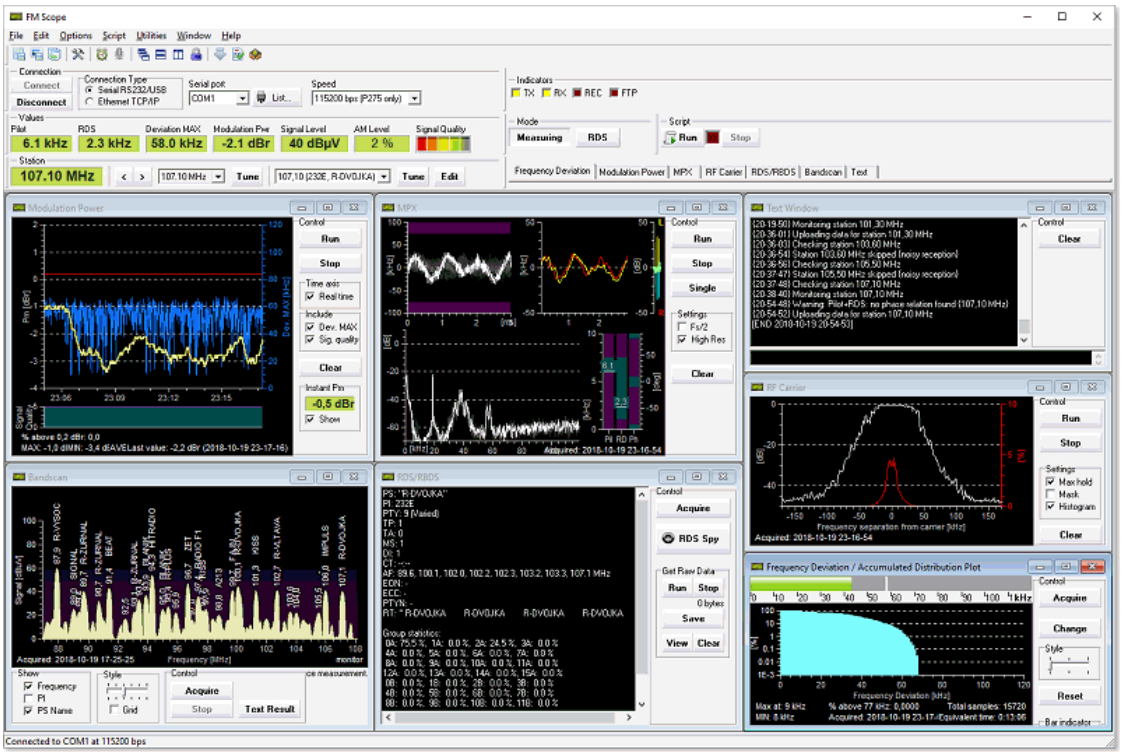
Suported devices:
P75, P175, P275, P275E, P275R

User Guide

Table of Contents

FM Scope for Windows

The FM Scope is a PC application primarily made for the P75/P175/P275 automated control, data visualization and data archiving. The application is provided for free download on the website.



The FM Scope main screen.

Basic features

- Graphical, text, spreadsheet and HTML data output
- Proprietary script language allows fully automated monitoring and data logging
- Built-in task scheduler, FTP upload, MySQL and SMTP email client
- TCP/IP remote control, built-in simple web server
- Free download – no additional costs

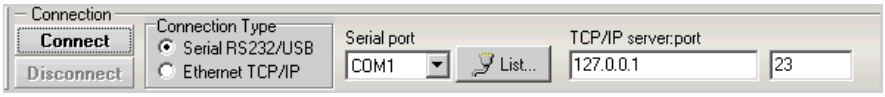
System requirements

- FM Analyzer or FM Monitor (https://pira.cz)
- Windows operating system

First steps

1. In case of USB connection (or USB to RS-232 adapter) install the USB driver first.
   Get latest USB drivers at https://ftdichip.com/drivers/vcp-drivers/
   Pure RS-232 connection never requires any extra driver.
2. Install the FM Scope.
3. Connect the equipment and run the FM Scope.
4. Select the COM port where the FM analyzer is connected and click Connect.
   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. Acquire a bandscan to get a list of stations or select a frequency and click Tune. The frequency tuned must be indicated in the Station bar.
7. Now you are ready.

Connection to the device

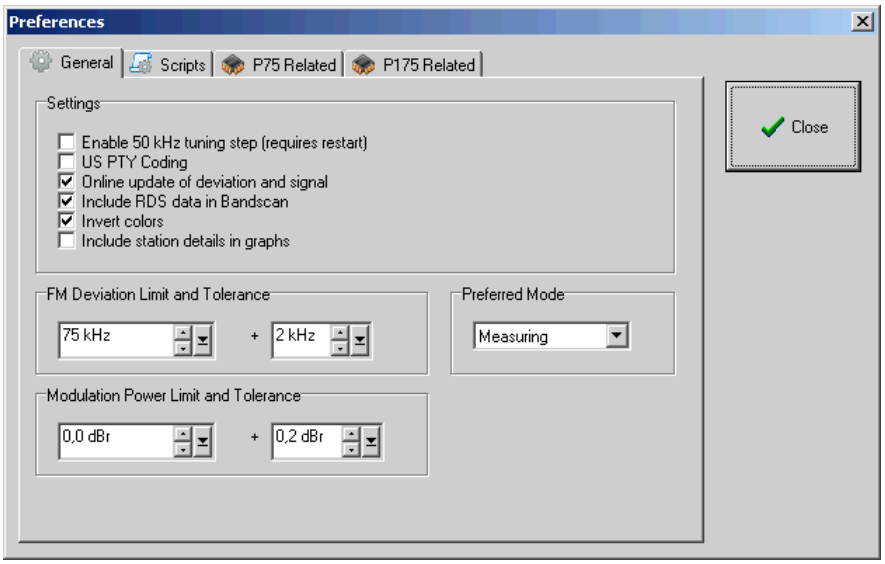


Connection Type – Select the first option if you use RS-232 or USB connection.

Serial RS232/USB:
Serial Port – The COM port to which the FM analyzer is connected.
Speed – Select higher speed for faster response. **Works only with P275, firmware version 2.2 or later.**
Ethernet TCP/IP:
TCP/IP Server – 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.
Any commercially available Ethernet-to-RS232 adapter can be used on the remote site.
Software based Ethernet-to-RS232 converters exist as well.

Connect – Opens the connection.
Disconnect – Closes the connection.

Preferences



Enable 50 kHz tuning step – Check **only** if there are stations in your area broadcasting in 0.05 MHz raster.
US PTY Code – Check if your location belongs to US broadcasting area.
Online Update of Deviation and Signal – If checked, the deviation and signal values are updated continuously. Typically, this option must be enabled in order to see actual values!
Invert Colors – Enables "night mode", inverts all colors in the graphs and text windows.
Allow multiple instances – Allows launching more application instances at one time. Each application must be installed separately in a different sub-folder.
Common settings for all Windows users – Switches between common application settings for all Windows users or keeping separate settings for each user.

Include RDS data in Bandscan – Enables PI and PS showing in bandscan.
Always check for actual PS and PI – Enables reading actual RDS data (PS and PI) for the station although it is already known or was entered manually. Has no effect if previous option is disabled.

FM Deviation / Modulation Power Limit and Tolerance – Broadcast limits and additional measuring tolerances that are given by authority in your country. *The tolerance settings have no effect on the measurement precision.*

Preferred Mode – The mode selected will be used on connect. Does not apply to the P275.

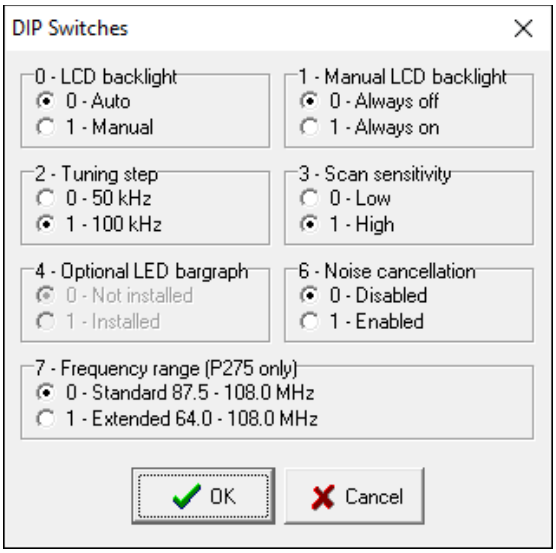
Debug mode – If checked, the script execution will stop on errors and error message will be displayed.

The application will remember these settings automatically. The settings are saved on application exit or can be saved manually using File – Save Settings Now.

DIP switches and Analogue I/O

The "DIP switches" nostalgic term represents a way how to configure some special settings of the device. Additional options are available for analogue setup, if you use P275, firmware version 2.2b r6 or later.

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

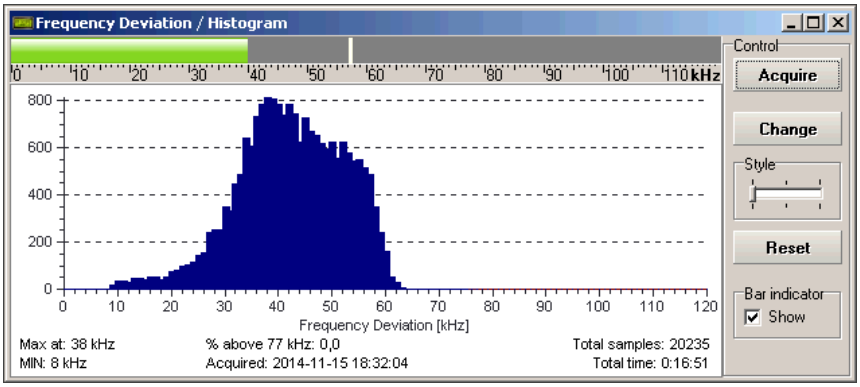


Currently it's possible to configure LCD backlight, tuning step, scan sensitivity, noise cancellation between stations (squelch and scan mute). For the P275 it's also possible to select operating frequency range and MPX filter bandwidth for FM deviation measurements.

Tool bar

- Save data from current window, usually in *.csv text format readable by MS Excel or Calc.
- Save graph image from current window
- Copy bitmap image from current window to Windows clipboard
- Show Preferences dialogue box.
- Show Task Scheduler
- Station lists
- Cascade windows
- Tile windows horizontally
- Tile windows vertically
- Lock all windows – protect from unwanted resizing or movement
- Acquire all static data from the device
- Save HTML report incl. inline SVG graphs. Template is placed in the application *lib* folder.
- Store current device settings to its internal EEPROM memory
- Open audio stream player

Frequency deviation, histogram and accumulated distribution plot



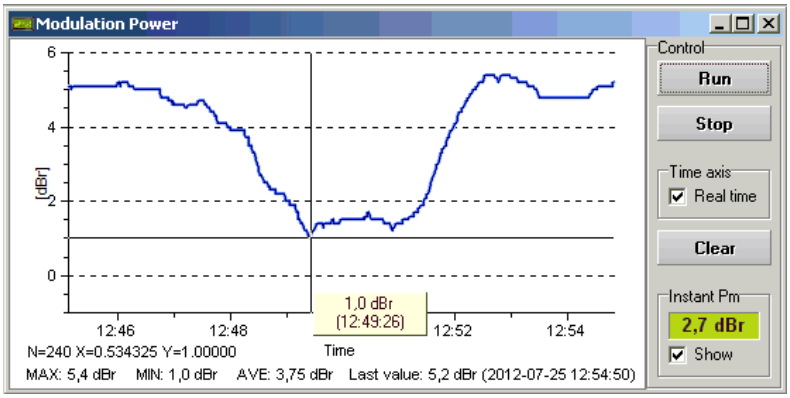
Acquire – Acquires the Frequency deviation data from the FM analyzer.
Change – Switches the display between histogram and accumulated distribution plot.
Reset – Clears all values. Equivalent to the Clear Data option in the FM analyzer's menu.
Show Bar Indicator – Shows fast peak deviation bar indicator with 1 sec. max hold function. The update rate is approx. once per 50 ms. Clicking on the scale, the unit can be switched between kHz and %. The 'Online update' option must be enabled in the Preferences. Requires firmware version 1.5 or later!

The frequency deviation histogram reflects the station's modulation characteristics captured over a time. It does not and it cannot provide instantaneous values, however using the histogram data the modulation is best described.

The frequency deviation histogram typically needs the station to be tuned at least a few minutes before clicking the Acquire button to provide relevant data.

Modulation power

The application can collect the Modulation power (also known as MPX power, P_m) data and show them as a function of time. Clicking in the graph the user may track the data.

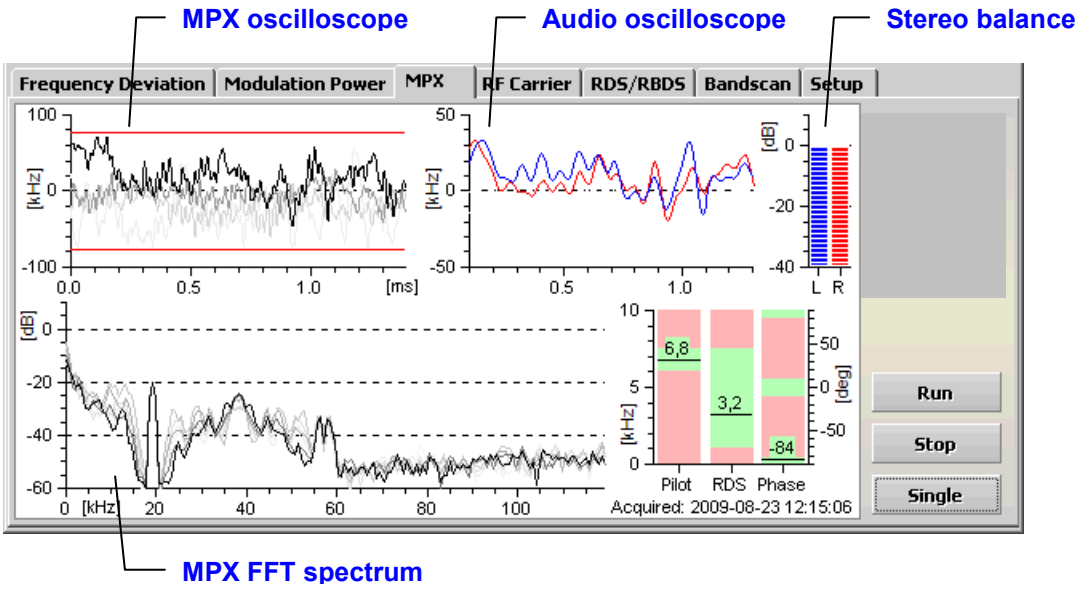


Run – Starts the measurement. Since Modulation Power is averaged over 60 seconds, first value will be available 1 minute after tuning the station.
Stop – Stops the measurement.
Time axis – Assigns real time or 0:00 to the first value in the data.
Include Deviation MAX – Adds deviation MAX curve.
Include Signal Quality – Adds signal quality (0 to 5) curve. Due to better readability the value is multiplied by two.
Clear – Clears the Modulation power data and graphic window.
Instant Modulation Power (P_m) – Shows the last one-second component of the modulation power, i.e. the modulation power averaged over one second. This value is useful especially for checking and adjusting modulation processor's algorithms and settings. It's available almost immediately after the station is tuned. Note that this value can't be compared with modulation power limit due to short averaging period.

MPX

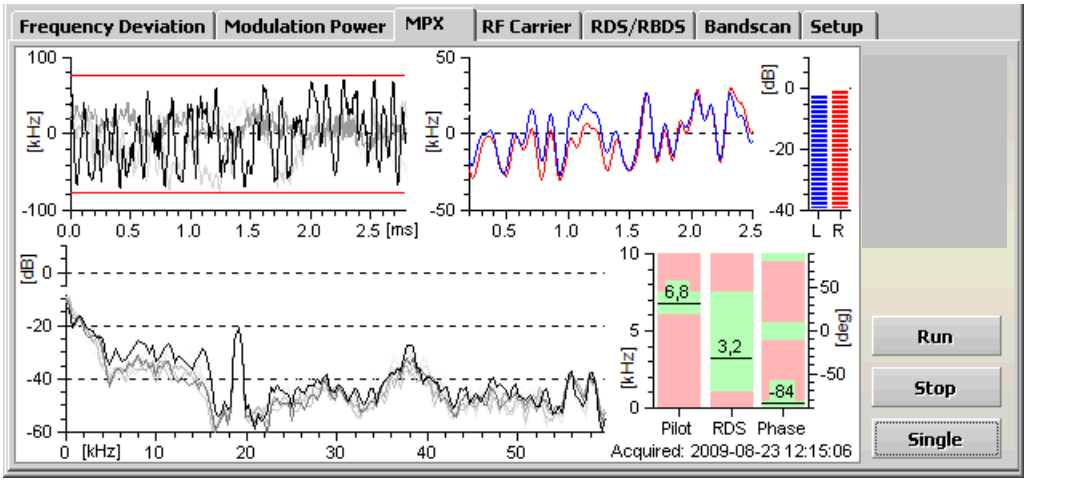
The MPX (multiplex) signal is the signal that goes into the modulation input of the FM transmitter. It typically consists of audio L+R, pilot tone 19 kHz, audio L-R modulated on 38 kHz (DSB-SC method) and additional components like RDS.

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.



Run – Starts the measurement.
Stop – Stops the measurement.
Single – Gets the data in only one process.
Clear– Clears the data and graphic window.

Fs/2 – Magnifies the lower frequency region by reducing the sampling rate to a half or original.
High Resolution – Increases the FFT resolution. **Works only with P275, firmware version 2.2 or later.**

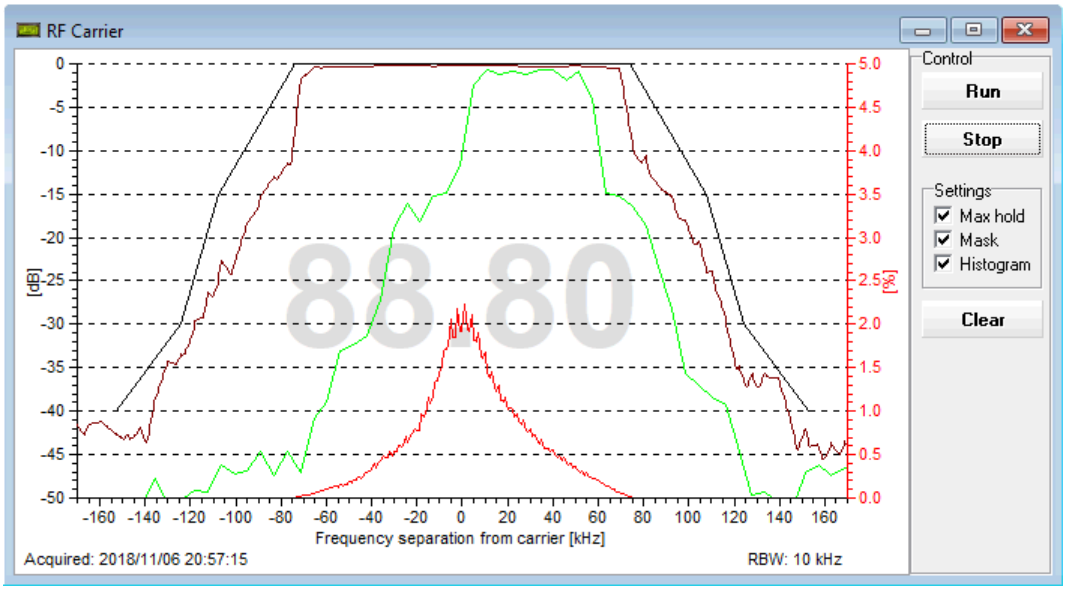


Effect of the Fs/2 option on X axes.

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/Stop – Starts/Stops the measurement.
Clear – Clears the data and graphic window.
Mask – Shows the spectral mask as defined by REC 54-01 E, and sets RBW to 10 kHz

Note: 0 dB ref. to full level of unmodulated RF carrier.
Note: To show the carrier spectrum, the Fs/2 option from the MPX page must be disabled (unchecked).

Spectrum mask based method and its limitations

The application implements the spectrum mask based method as defined in REC 54-01 E. This method was used especially 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. Implementation in the FM Scope is based on Fast Fourier Transform (FFT) rather than sweeping over the band, however the resolution bandwidth (RBW) is adapted to achieve maximum comparability. Unlike the spectrum analyzers, the FM Scope implementation is insensitive to signal level fluctuations because the RF carrier is sampled after an amplitude limiter.

The mask method is 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 (in max hold mode) is within the limits of the mask.

Recommended minimum period of the signal recording is:
- 5 minutes if connection speed is 115200 bps
- 15 minutes if connection speed is 19200 bps

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 amount of interference can be determined from the level of spectrum components in the maximum distance from the carrier (around ±160 kHz).

Advanced use of the RF carrier spectrum analysis

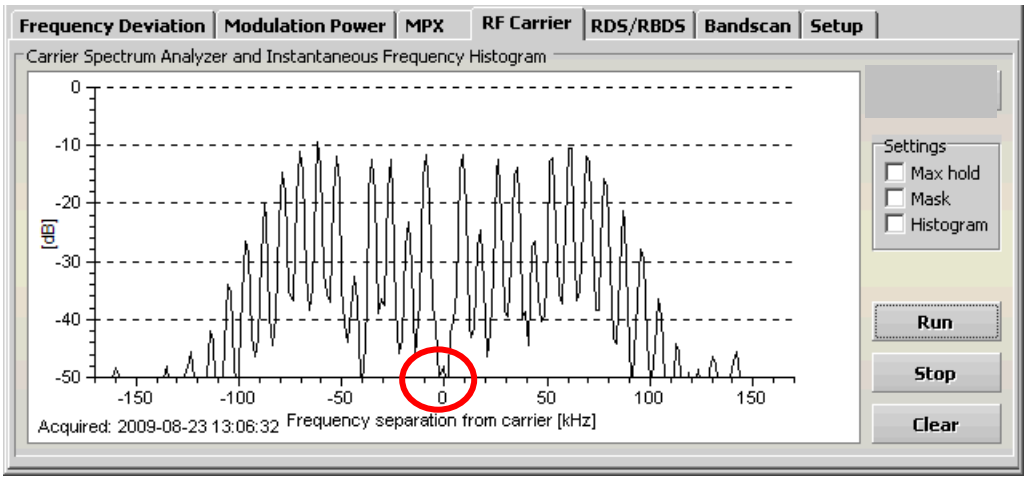
The RF carrier spectrum analysis is a very useful tool for very accurate measuring of FM deviation and modulation index and for making fast and accurate adjustments of FM transmitters on selected single modulation frequency (sine tone). The transmitter is adjusted to a precise frequency deviation with 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+π(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:

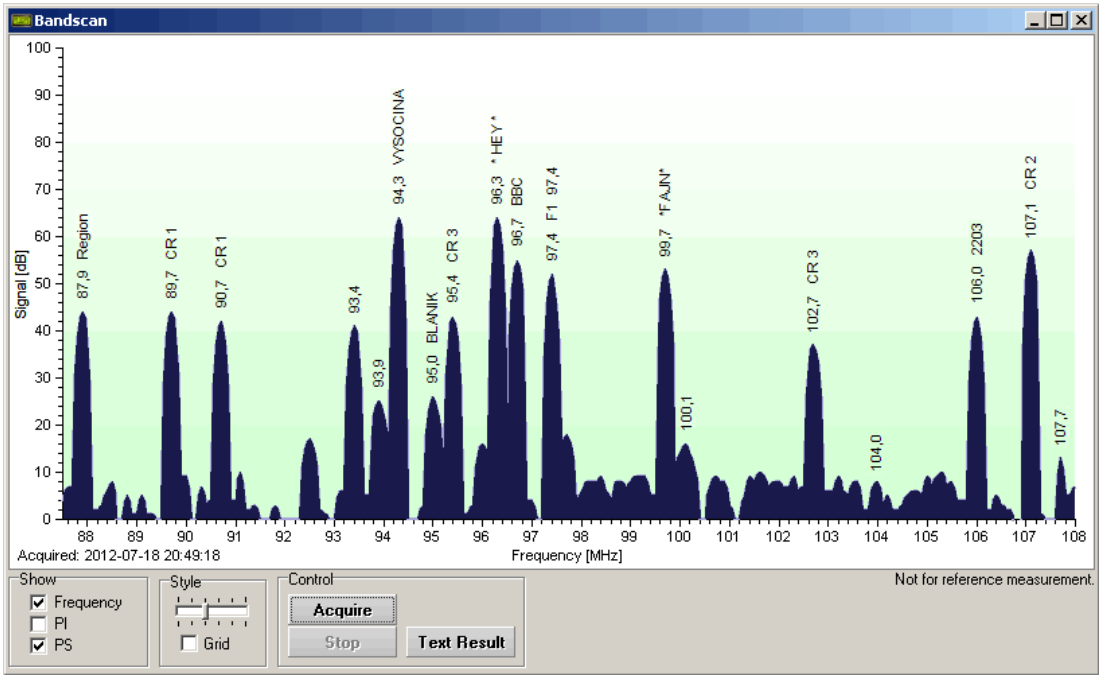
1. Select the required deviation ΔF.
2. Select a modulation index β that gives a modulating frequency commensurate with the normal modulation bandwidth of the transmitter to be tested.
3. Set the modulating frequency to ΔF / β, and monitor the output spectrum of the FM transmitter on the RF Carrier page.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

Note: Pilot tone and RDS must be switched off on the transmission equipment before making this measurement.

Bandscan

This is an additional feature which shows quick graphic representation of the band occupation and estimated signal strength. By clicking on a station peak you may directly tune the station. The strongest stations are copied into current station list and showed in the Station bar, allowing quick tuning.

Due to physical characteristics of the device, this feature is not intended for coverage analysis nor reference signal strength measurement! The bandscan effectively shows a visualized list of stations detected rather than showing a simple print of amplitude spectrum. The station's detection ability is among others affected by the RF signal bandwidth which is fixed at 280 kHz, as required for proper measurement of modulation characteristics.



Acquire – Acquires the Bandscan. The scanning is indicated in the application bottom line.
Stop – Aborts the operation.
Text Result – Copies the station list in the Text window.

Style – Selects from various displaying styles.

If RDS data is enabled for bandscan in Preferences, the bandscan may show PI and PS for each station. However it may take a few minutes before the bandscan finishes due to additional time required for getting RDS data.

The bandscan frequency range can be changed in Station list editor. Please refer to the device manual for supported frequency range.

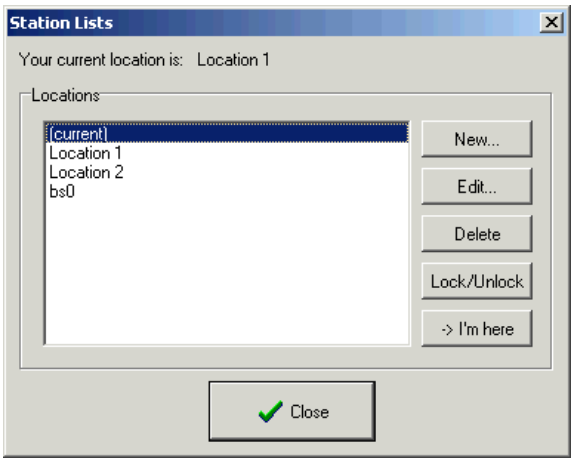
Note (P75 only): 0 dB ref. to approx. 0.5 µV on the antenna input. Max. dynamic range is <80 dB. The signal value on the Y-axis is estimated from the Signal and Noise Level values from LCD page 5. Precision is not specified for this function.

Note: The bandscan currently supports 100 kHz tuning step only.

Station lists

The idea of Station lists is to store and switch between different bandscans and station presets. This is especially useful if one makes monitoring in different locations as it can quickly tune the stations of interest. A separate station list can also be a source for some automation scripts without affecting station presets dedicated for manual control.

To access the Station lists, click on the button  .



New… – Create new location (station list). Up to 15 locations can be created.
Edit… – Edit the location. Allows to edit, add and delete stations from the list.
Delete – Delete the location selected.
Lock/Unlock – Affects how the Bandscan feature processes current list. Stations from locked list are merged with stations from new bandscan. Stations from unlocked list may disappear if they are not recognized in new bandscan. However stations that were entered or edited manually will never disappear from the list, regardless of the lock/unlock state.
-> I'm here – Load the location selected.

To fill the list with station presets, take a bandscan or click on the Edit button.

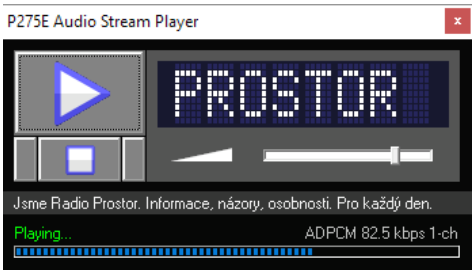
To tune the station, select it in the Station toolbar and click on Tune.

To show complete station list data, click on the Text output button in Bandscan window.

Note: Please refer to the device manual for supported frequency range.

Audio stream player

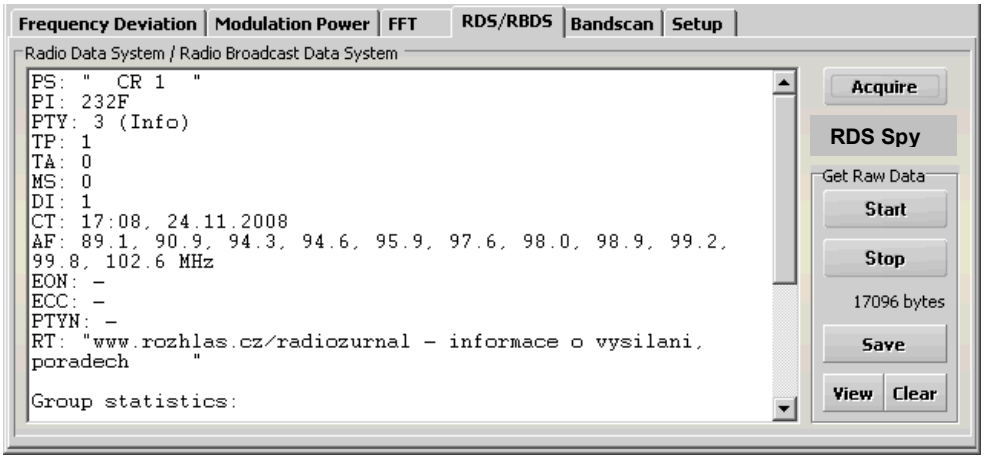
The audio stream player provides remote audio listening of currently tuned station, incl. RDS functions PS, RT and TA. It supports all devices connected via TCP/IP or serial line 115200 bps (P275, P275E, P275R).



The audio streaming uses lossy audio compression. It is thus not intended for audio quality evaluating.

Minimum version required: FM Scope 1.7, P275 firmware 2.2d. Following FM Scope function are not available while the audio stream player is active: MPX, RF Carrier, Bandscan.

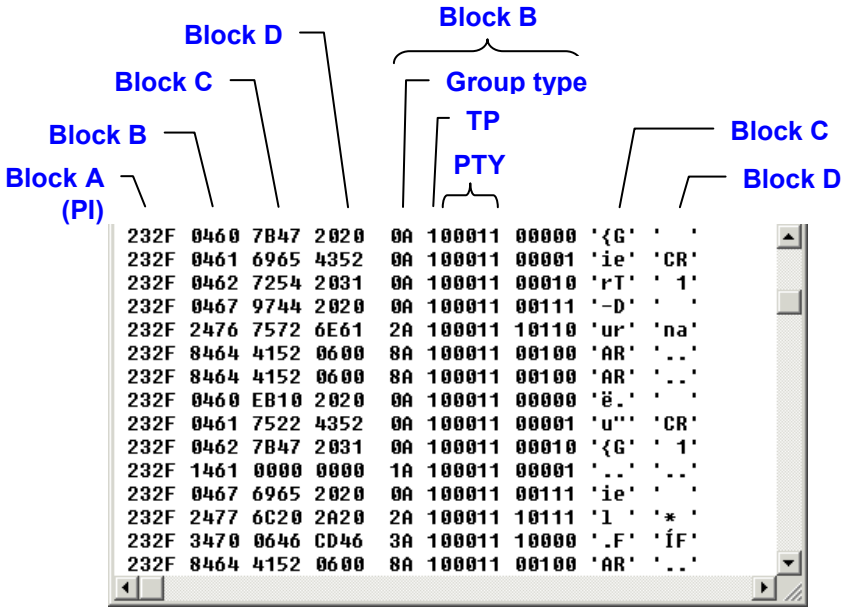
Radio Data System / Radio Broadcast Data System



The FM Scope application provides basic functions for working with RDS data. For complex real-time RDS analysis, click on the **RDS Spy** button.

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.

Text window

The Text window primarily provides one of the text output possibilities for automation scripts.

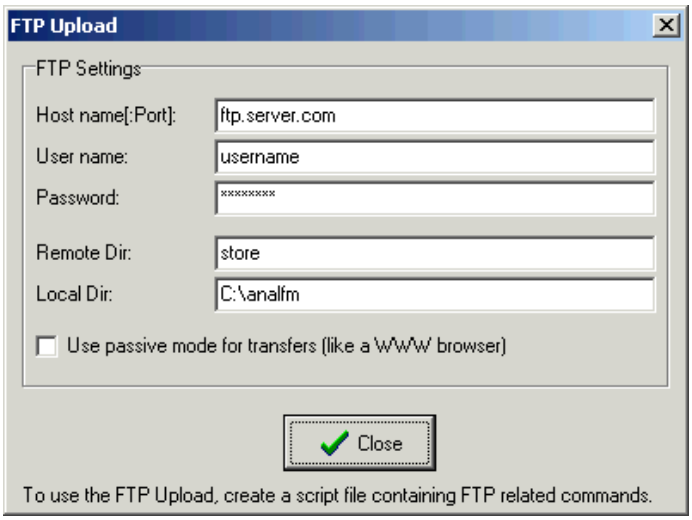
The Text window also serves as a console, i.e. it allows entering script commands in the bottom line. For example if you type exit confirmed by <Enter>, the application terminates. If you type %time, the window shows content of the variable named time – current time. To use the console, there must be no other script running.

The script language is described later in this manual.

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 through the script files.

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

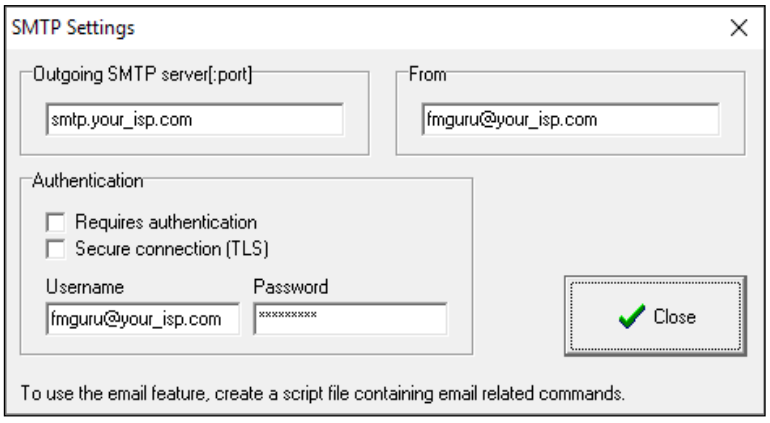


SSL/TLS – Enable for secure FTP connection (FTPS). Supported mode of operation is TLS1.0 explicit 'PROT C'.

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.

SMTP Email client

The application includes simple SMTP client that allows sending emails to one or more recipients with optional file(s) attached. The email client is controlled via script. Setting up the SMTP parameters is necessary before use. In main menu, choose Options/SMTP Settings.



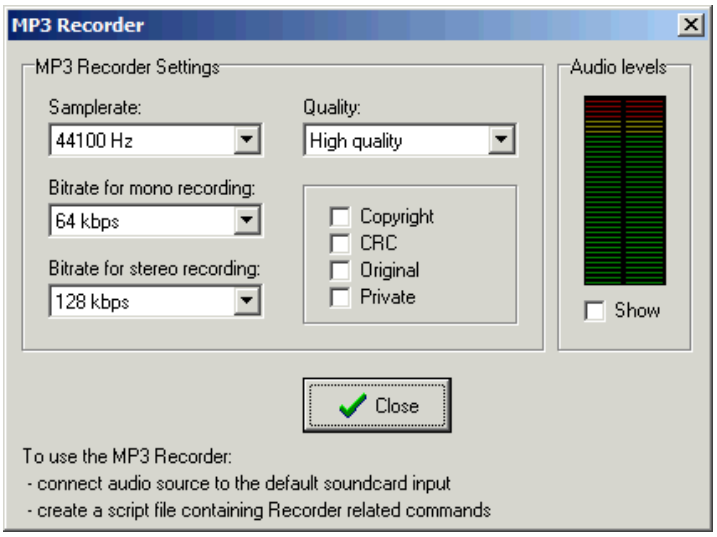
Information about how to set up the SMTP parameters is available from your mail account support or internet connection provider (ISP). The 'From' field must also be specified, it is usually the email address associated with your email account.

If your ISP blocks default SMTP port 25, use port 587 instead, for example smtp.email.com:587

Note for Gmail users: Google no longer supports 3rd party applications such as the FM Scope.

MP3 recorder

The MP3 Recorder is a bonus feature which works independently of the FM analyzer device but may record its headphones audio output as well. The recorder needs to be configured before first use. In main menu, choose Options/MP3 Recorder.



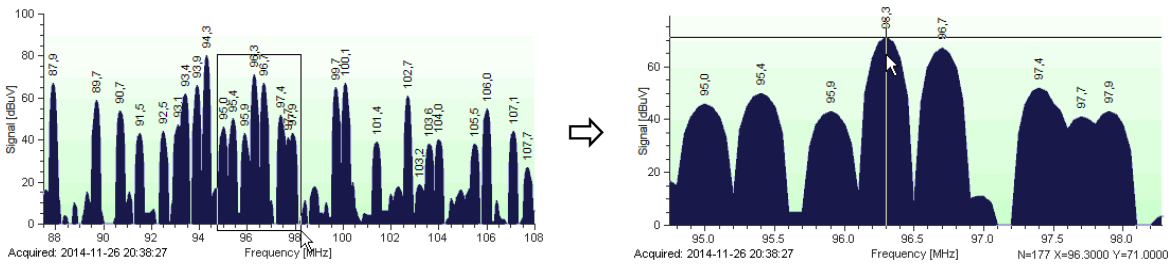
To use the MP3 recorder, connect the audio source to the default soundcard input using any metalic audio cable. Create a script file containing MP3 recorder related commands (see the list of commands in section Script Files / List of commands / Working with files). This feature can be also combined with Task scheduler to create fully automated system of recording.

Make sure the file lame_enc.dll is placed in the application folder. This file is a part of the installation package or may be downloaded from the net.

Important! Recording audio content may violate copyright laws!

Zoom option

Any graph window can be maximized over the entire workspace. Above that you can zoom in on your graph with the zoom function, or in other words enlarge a certain part of the graph. The zoom option uses mouse and can be applied to all graphs in the application if no graph is being refreshed by new data.



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

Script files

The application supports proprietary script language which massively extends the area of use and number of features. User can fully automate the measurements and data storage. The script files can contain only a few steps as well as large blocks of data acquisition and processing procedures. Presence of that feature and its 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's menu or use command line option, for example:

```
fmscope.exe measure.fms
```

Each line in the script file may contain one command. The command interpreter is not case-sensitive. Maximum line length is 1024 characters, maximum script total length is 2048 lines. The script codepage is expected to be ANSI.

Quick access

User can quickly run the script by selecting its name in the Script submenu. That script file must be saved in the application's subfolder named 'usr'.

List of commands

Command	Meaning	Example of use
General purpose and application control:		
connect	Open a COM port (given by number and optional speed) or TCP connection (given by host:port) for communicating with the device. If TCP connection fails for any reason, automatic reconnection attempts follow with 20 sec. period. If no argument is specified, last connection parameters are used. If a connection is already established, nothing happens.	connect(1) connect(1,19200) connect(localhost:23) connect(127.0.0.1:23) connect
disconnect	Close the connection. If no connection is opened, nothing happens.	disconnect
send	Send any string to serial port.	send(?F) send(*B) send(?C)
setproperty	Set an application property to a new value. Compete list of properties is given in Annexes.	readproperty(TempValue,InvertColors) setproperty(InvertColors,0) ... your script here ... setproperty(InvertColor,%TempValue)
readproperty	Read an application property and place its value to a variable.	
page.show	Switch the FM Scope display to the page (window) 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)

Measurements:

location.load	Select and load the location (station list) by its name. If no location of that name is found, it is created automatically.	location.load(MyLocation1)
location.lock	Lock current location (merge mode for bandscans).	location.lock
location.unlock	Unlock current location.	location.unlock
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
getamod	Get current AM value in % (P175/P275 only).	getamod
getlevel	Get current signal level (P175/P275 only).	getlevel textwindow.addline(Level is %level)
getnoise	Get current noise	getnoise
getphase	Get pilot-to-RDS phase difference value.	getphase
frequencydeviation.acquire	Acquire current frequency deviation data (incl. histogram).	frequencydeviation.acquire
frequencydeviation.histogram	Switch the display between histogram (1) and accumulated distribution plot (0).	frequencydeviation.histogram(1)
frequencydeviation.showbar	Enable (1) or disable (0) showing of the instantaneous frequency deviation bar.	frequencydeviation.showbar(0)
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
modulationpower.realtime	Enable (1) or disable (0) showing of real time / deviation MAX / signal quality in the modulation power graph.	modulationpower.realtime(1)
modulationpower.includedeviation		modulationpower.includedeviation(1)
modulationpower.includesignal		modulationpower.includesignal(1)
mpx.single	Get current 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 Fs/2 (1) mode for MPX.	mpx.doubletimebase(0)
mpx.highresolution	Set normal (0) or high resolution (1) mode for MPX.	mpx.highresolution(1)
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.	rds.getdata
rds.getstatistics	Get RDS group statistics.	rds.getstatistics
rds.clear	Clear the RDS data held by the app.	rds.clear
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 current bandscan.	bandscan.acquire
bandscan.style	Select the bandscan style. Value 0 (zero) corresponds with the first option.	bandscan.style(3)
bandscan.grid	Enable (1) or disable (0) the bandscan grid.	bandscan.grid(0)
bandscan.showfrequency	Enable (1) or disable (0) the frequency indication above the station's peak.	bandscan.showfrequency(1)
bandscan.showpi	Enable (1) or disable (0) the RDS PI indication.	bandscan.showpi(0)
bandscan.showps	Enable (1) or disable (0) the RDS PS indication.	bandscan.showps(1)
imf.setasnormal	Set current 2nd IF frequency as a normal. Allows to measure carrier offsets and to show this value in Report.	imf.setasnormal
imf.get	Get current 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)
loadtext	Load a text file into specified variable.	loadtext(control.txt,var1) loadtext(logo.svg,inlinesvg)
frequencydeviation.savedata	Save the frequency deviation data.	frequencydeviation.savedata(histogram.txt)
frequencydeviation.savegraph	Save the frequency deviation graph. *	frequencydeviation.savegraph(%date.svg)
modulationpower.savegraph	Save the modulation power graph. *	modulationpower.savegraph(pm.jpg)
modulationpower.savedata	Save the modulation power data.	modulationpower.savedata(pm.txt)
mpx.savedata	Save the FFT data.	mpx.savedata(fft.txt)
mpx.savegraph	Save the MPX graph. *	mpx.savegraph(mpx.jpg) mpx.savegraph(inlinesvg)
rf.savegraph	Save the RF graph. *	rf.savegraph(rf.jpg)
bandscan.savegraph	Save the bandscan graph. *	bandscan.savegraph(%date.bmp) bandscan.savegraph(bs.jpg,640x480)
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)
textwindow.savedata	Save current content of the text window to a file.	textwindow.savedata(c:\text.txt)
createreport	Create and save basic 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,%rdspi%cr%rdsps)
json.escape	Returns the string JSON-escaped. Gets the string from second argument, store result to the variable in first argument.	rds.getdata json.escape(jsonps,%rdsps) savetext(C:\ps.json,{ "ps": "%jsonps"})
savesettings	Save the application settings now.	savesettings

** Following graphic formats (file extensions) are supported: bmp, jpg, wmf, svg. Optional image resolution may be specified in format(width)x(height). See the example of use. A special filename inlinesvg loads the image into the %inlinesvg variable.*

Sending email:

email.to	Recipient's address(es).	email.to(info@radio.com, luis@radio.com)
email.subject	Message subject	email.subject(FM Scope Message)
email.body	Message body	email.body(Silence detected on %frequency!)
email.attachfile	Attach the file specified. To attach more files, use this command repeatedly. After the email is sent, the attachment's queue is cleared.	email.attachfile(C:\monitor\histogram.jpg)
email.send	Send the email. Don't forget to configure the SMTP settings in Options submenu prior to send any emails!	email.send

Script flow control:

execute	Execute any application, open any document or run another script. Optional command line parameter(s) may be specified. If no path is specified, default path is %appfolder\. Does **not** wait until the application terminates!	execute(C:\batch.bat) execute(notepad.exe,text.txt) execute(command,/c md c:\data) sleep(2) execute(usr\another_script.fms)
sleep	Stop executing next commands for a time specified in seconds.	sleep(30)
include	Include another script file (replacing the include command) into this script file **before** processing the script. If relative file path is entered, the application first searches in the script folder, next in the application *lib* folder.	include(savedata.fms) include(report_svg.fms)
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 allowed.	call(myproc) ... stop myproc: ... return
return	Return from subroutine. Continue on next command after last call.	
exit	Exit the FM Scope.	exit

if	Allows for conditional execution of script fragments.
	Only one condition is allowed per one if command.
	
	Only these operators are allowed:
	> greater than,
	< lower than,
	>= greater than or equal to,
	<= lower than or equal to,
	= equal to (numeric comparison),
	!= not equal to (numeric cmp.),
	$= equal to (string cmp.),
	@ included in (string cmp.),
	& bitwise and.
	
	Logical operators like *and*, *or*, *not* are not permitted, however nesting of multiple if commands is possible.
	
	If the variable does not contain valid numeric value, its numeric value is considered as -1 for the purpose of numeric comparisons.
		If (%connected=0)
		  showmessage(Not connected!)
		  stop
		  endif
		
		getquality
		if (%quality>3)
		  call(myproc)
		  endif
		
		getfrequency
		if (%freq@09630 10600 10710)
		  return
		  endif

stop	Don't execute next commands in the script file. Stop processing the script.	stop

Text window output:

textwindow.addline	Add a new line to the bottom of the text window.	textwindow.addline(This is a new line)
textwindow.rewriteline	Rewrite the last line in the text window.	textwindow.rewriteline(New text at %time)
textwindow.addtext	Add a text to the end of the last line.	textwindow.addtext(Line now ends here.)
textwindow.clear	Clear entire content of the text window.	textwindow.clear

Working with FTP:

ftp.connect	Connect to the FTP server specified in FTP Upload settings.	ftp.connect
ftp.put	Put the file on the server. Optionally the filename on the server may differ from the local filename.	ftp.put(%date.log) ftp.put(%freq.jpg,histogram.jpg) ftp.put(%date.log,results/log.txt)
ftp.makedir	Create a new directory on the FTP server.	ftp.makedir(logs) ftp.makedir(logs/%date)
ftp.disconnect	Disconnect from the FTP server.	ftp.disconnect

Working with HTTP:

httpget	Access the URL specified, optionally pass parameter(s) using GET method. The accessed site's content is not stored anywhere.	httpget(http://site.com/log.php?s=%freq)

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.
	When need the variable value, a prefix % must be added.
	The variables remain in the application memory until its exit or applying reset command.
		set(variable1)
		set(msg,Counting...)
		repeat:
		showmessage(%msg %variable1)
		inc(variable1)
		if (%variable1<5)
		goto(repeat)
		endif

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.	set(starttime,%date %time) ... showmessage(Running since %starttime)

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.
		input(Enter your location name)
		if (%inputvalid=1)
		checkfilename(fn,%inputtext)
		bandscan.savegraph(%fn %time.jpg)
		endif

savedialog	Show standard save file dialogue box. Optional filter may be specified.
	If a file name is confirmed, a variable inputvalid is set to 1 and variable inputtext contains the file name (incl. path).
		savedialog(*.jpg)
		if (%inputvalid=1)
		bandscan.savegraph(%inputtext)
		endif

opendialog	Show standard open file dialogue box. Optional filter may be specified.
	If a file name is confirmed, a variable inputvalid is set to 1 and variable inputtext contains the file name (incl. path).
		opendialog
		if (%inputvalid=1)
		execute(%inputtext)
		endif

save	Save content of specified variable to disk to vars.ini file. Variable saved this way is available to all scripts using command load. The file can be changed by assigning a full path to the VariableStorage.	save(counter) set(VariableStorage,D:\Config\vars1.ini) save(counter)
load	Load content of specified variable from disk from vars.ini file. If no variable of that name has been saved before, a default value behind the comma is assigned. The file can be changed by assigning a full path to the VariableStorage.	load(counter,0) load(counter,%counter)
loadexternal	Like the load command but loads the content of variable from external file specified. The external file format is as follows: *[Global]* *variable1=value1* *variable2=value2* etc.	loadexternal(d:\control.txt,counter,0)

reset	Remove all variables from RAM. It is preferred to call this command on the beginning of your script whenever you don't need to keep variables created by previous scripts.	reset

Socket control:

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

MySQL client:

mysql.connect	Connect to the MySQL server specified in MySQL Client settings.	mysql.connect
mysql.query	Sends a database query using the SQL syntax. *Does not allow to retrieve data from the database.*	mysql.query(CREATE TABLE table1 (column1 FLOAT)) mysql.query(INSERT INTO table1 VALUES (%maxhold_))
mysql.disconnect	Disconnect from the MySQL server.	mysql.disconnect

Following tags may be used to create dynamic content, to get system values or to insert special characters:

Tag	Meaning
%time	Current time in format HH-NN-SS
%date	Current date in format YYYY-MM-DD
%timestamp	Number of seconds that have passed since 1.1.2000
%frequency	Frequency in long format (87.5 MHz)
%freq	Frequency in compact format (08750)
%offset	Carrier offset (if previously got by imf.get)
%offsetnoref	Carrier offset (if previously got by imf.get) without mention the reference frequency
%pilot	Pilot deviation
%rds	RDS deviation
%phase	Pilot-to-RDS phase difference
%quality	Signal quality (if previously got by getquality)
%amod	AM modulation in % (if previously got by getamod, P175/P275 only)
%level	Signal level in dBµV (if previously got by getlevel, P175/P275 only)
%noise	Noise as showed on LCD (if previously got by getnoise)
%abovelimit	% above dev. limit (histogram function)
%devtotallimit	Deviation limit incl. tolerance (as set in Preferences)
%maxat	MAX at (histogram function)
%totalsamples	Total number of samples (histogram function)
%min	Deviation MIN Hold
%devave	Deviation AVE
%maxhold	Deviation MAX (the value updated every second)

%tsmaxhold	10 sec. MAX (if previously got by send(?X))
%pmlast	Modulation power – last value
%pmmax	Modulation power – MAX value
%pmmin	Modulation power – MIN value
%pmave	Modulation power – AVE value
%pmabovelimit	% of time the modulation power is above limit
%pmtotallimit	Modulation power limit incl. tolerance (as set in Preferences)
%lmaxdb	Max. dB peak in left channel (MPX window)
%rmaxdb	Max. dB peak in right channel (MPX window)
%balance	Linear channel balance R/L (if previously got by *B followed 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
%connected	Returns 1 if communication port is opened or TCP connection is alive, otherwise returns 0.
%percent	Inserts % sign without processing subsequent expression as variable.
%location	Current location name
%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 - long format (87.5 MHz)
%station.freq	Frequency of the station pointed by the %stationindex - compact format (08750) suitable for use in file names
%station.level	Reception level of the station pointed by the %stationindex
%station.noise	Reception noise of the station pointed by the %stationindex
%station.rdspi	RDS PI of the station pointed by the %stationindex
%station.rdsps	RDS PS or description of the station pointed by the %stationindex
%station.userdata	Optional user data of the station pointed by the %stationindex
%inputtext	The text entered using input query or dialogue box
%inputvalid	Contains 1 if any text was entered using the input query or dialogue box, otherwise contains 0.
%tempfolder	A Windows folder where temporary files may be placed
%appfolder	FM Scope application folder, without trailing delimiter (\)

%lasterror	Contains a text of the last script execution error. User writable.
%tcpconfailed	Returns 1 if a TCP connection issue or manual disconnect has occurred during processing of the script. User writable – if used in the script, it must be set to 0 before communicating with the device. After the communication ends, the %tcpconfailed variable can be tested by if command. Measurement results should be discarded if the value of 1 is detected.
%inlinesvg	Holds SVG image in format required for inline insertion into HTML, i.e. removes all content outside the <svg>…</svg> pair. See also savegraph and loadtext.
%rfnoise	A function of the RF Carrier graph. Holds maximum noise level relative to the overall level.
%rdsber	RDS Block Error Rate (if previously got by rds.getdata). Requires firmware version 2.2c r2 or later.
%rdsrtp1c %rdsrtp1t %rdsrtp2c %rdsrtp2t	RDS RT+ class name and text content of tag 1 and tag 2 (if previously got by rds.getdata). Requires firmware version 2.2c r2 or later.
%rdsrtpbits	RDS RT+ bits Item Running and Item Toggle (if previously got by rds.getdata). Requires firmware version 2.2c r2 or later.

Important!
The tags above are reserved and are processed preferentially. This means that no variable name may start with any of these reserved tags, otherwise it may yield unexpected results.

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:
The script language is strictly optimized for use with the FM analyzer equipment. The script engine maintains simplicity, for example it does not support mathematical operations on the values measured.

Script file example 1

(Measure a station, save basic report.)

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

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:\fmscope\data\. The folders must be created before.)

```
bandscan.acquire
bandscan.savegraph(c:\fmscope\data\bandscan.jpg)

ftp.connect
ftp.put(bandscan.jpg)
ftp.disconnect

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

ftp.connect
ftp.put(hi%freq.jpg)
ftp.put(dp%freq.jpg)
ftp.put(mx%freq.jpg)
ftp.put(pm%freq.jpg)
ftp.put(rd%freq.txt)
ftp.put(rp%freq.txt)
ftp.disconnect

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:
email.to(info@radio.com, luis@radio.com)
email.subject(FM Scope Message)
email.body(Failure detected on %frequency!)
email.send
return
```

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.savegraph(%filename.jpg)
skip:
inc(stationindex)
goto(rpt)
```

Script file example 10 (Advanced users)

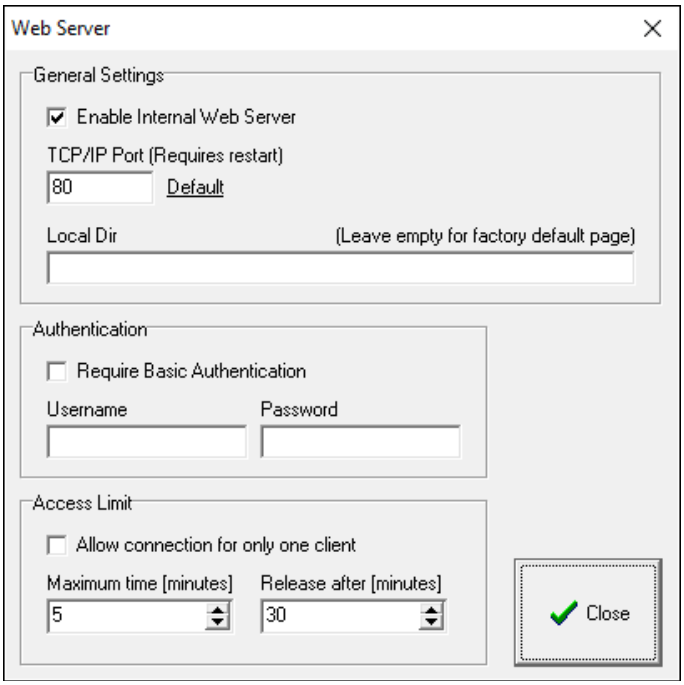
(Working with array – a set of variables whose names are created dynamically using another variable)

```
set(index,0)

rpt:
if (%index>9)
  stop
  endif

set(item%index,Item %index content)
textwindow.addline(item%index value is %item%index)

inc(index)
goto(rpt)
```

Web server

The software includes internal web server, the content is available to the operator through a standard web browser or even mobile device. The web server supports showing of special dynamic content as well as HTML forms (GET, POST) for setting variable content and controlling a script. Basic control is implemented without need of any script running. The web server is not intended for public presentation of the results and does not support PHP or ASP.

Before attempting to use the web server, it must be configured first in Options/Web Server:

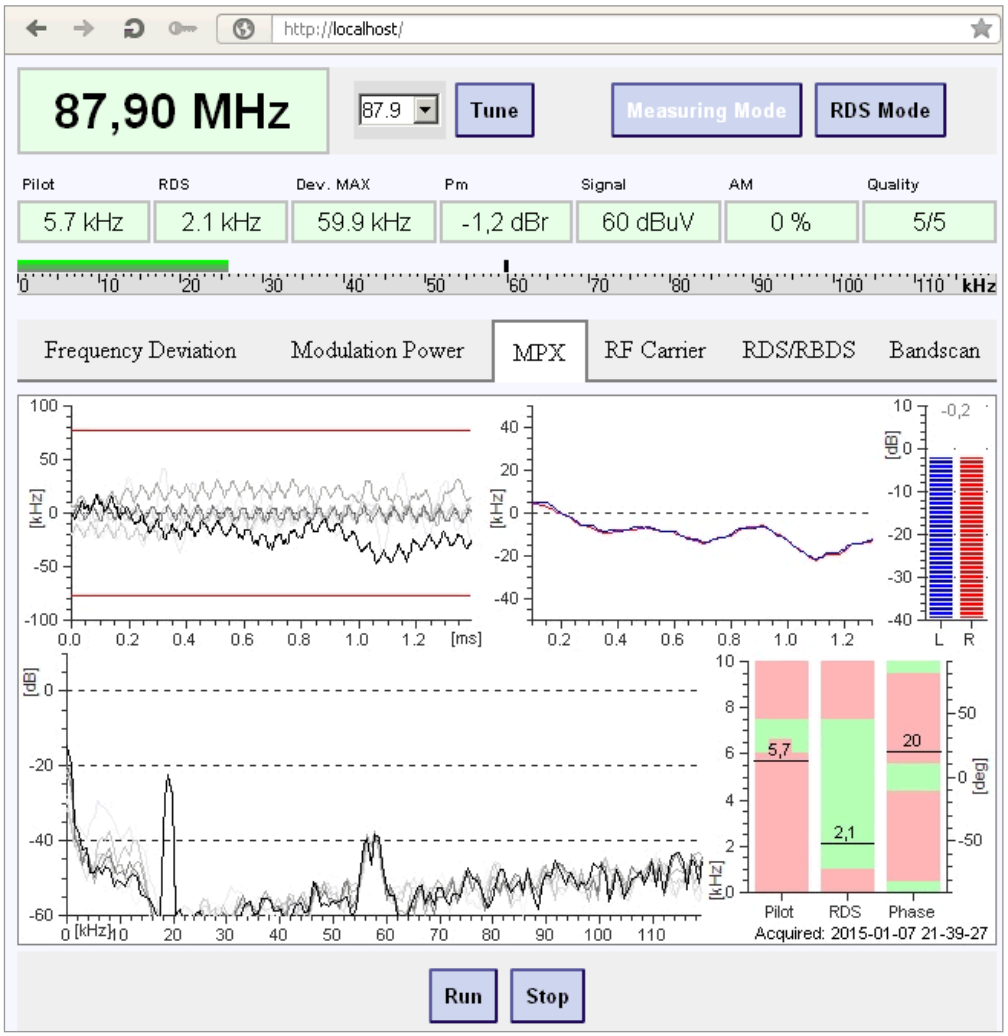


Enable Internal Web Server – Enables the Web server.
TCP/IP Port – The port number on which the server provides the content. Default is 80.
Local Dir – If not empty, specifies the user web root directory on your local hard disk. Usually, this is the folder which contains the main file index.htm.

Authentication – The authentication should be enabled if the web server can be accessed from public internet.

Access Limit – If enabled, the web server allows access for only one client at a time. The access is denied after the Maximum time elapses. The same client can connect again after reaching the Release time from the latest active operation. This allows another client to connect. Each client is identified by IP address and http agent.

Once configured, open the web browser and type the address http://localhost/ (for default port 80) or http://localhost:<port>/ (for other ports):



If the values do not update automatically, make sure that Online Update of Deviation and Signal is enabled in Options / Preferences / General and your browser supports JavaScript.

Hint: Following paragraphs are intended for those who know writing dynamic web pages.

Dynamic content

Real-time parameters can be referred to in the website through the use of dynamic tags. These tags are equivalent to the script dynamic tags and variables. Also the syntax is the same. When dynamic tags are placed in an HTML web page, corresponding actual values are sent by the web server when the page is served out. Syntactically, the dynamic tags are variable names prefixed by single % character.

The following files can include dynamic tags: .HTM, .HTML, .JS, .INC, .XML, .CSS, .XHTML

The user website is composed of standard HTML files, which may contain links to internal or external websites, Java Scripts, graphic files and more. Using the dynamic tags, the page can also be used to dynamically display and update measured values, script variables etc. For example, to display the current pilot level value, enter %pilot anywhere on the page, as in the following example:

```
<HTML>
<HEAD>
<TITLE>SAMPLE PAGE</TITLE>
</HEAD>
<BODY>
<h1>%pilot</h1>
</BODY>
</HTML>
```

When serving this home page, the web server replaces the %pilot string in the served page with the current value of that parameter.

Note: The web server cannot show any measured value which has not been previously received from the device. When serving the content, the web server typically does not update the values from the device but uses the latest values known.

Reserved addresses

Following addresses provide special content:

Address	Description	Example
/getvar/<variablename>	Returns the content of the specified variable (dynamic tag) in plain text format.	http://localhost/getvar/time
/getstr/<string>	Returns the string in plain text format, with all dynamic tags replaced by actual content. The string must be URL-encoded.	http://localhost/getstr/ The+time+is+%25time +and+date+is+%25date
/getimg/<imagename>	Returns the graph image specified in svg format. The image name can be one of the following: frequencydeviation, modulationpower, mpx, rf, bandscan.	<img src=”/getimg/mpx”>
/getimg/<imagename>/ <resolution>	The same as previous but you specify also the image resolution in pixels (width x height, without spaces).	<img src=”/getimg/mpx/ 640x480”>

Reserved variables

There's a reserved variable named 'webaction' which is used to perform basic control of the device. Once the website returns this variable via either GET or POST method, following operation will be performed, depending on the variable content:

'Webaction' variable content	Action equal to
H	Frequency Deviation / Acquire
P	Modulation Power / Run
p	Modulation Power / Stop
O	MPX / Run
o	MPX / Stop
D	RDS/RBDS / Acquire
A	Bandscan / Acquire
G	switching to Measuring mode
g	switching to RDS mode
S	stop current script
s	restart current script
frequency for example 91.9	entering the frequency and clicking on Tune button

For some operations, internal variable named 'webwaitingfor' indicates whether the operation has completed or not:

%webwaitingfor	Following operation has not completed yet
H	Frequency Deviation / Acquire
D	RDS/RBDS / Acquire
A	Bandscan / Acquire
F	tuning to a new frequency

Basic state of the FM Scope application is indicated to the website via variable named 'updating':

%updating	Meaning
bit 0	Modulation Power running (1) or not (0)
bit 1	MPX running (1) or not (0)
bit 2	Fast peak deviation bar indicator active (1) or not (0)
bit 3	Measuring mode (0) or other mode (1)

Interaction with running script

The web server provides not only the value viewing and basic control but can take control over a FM Scope script as well. The prerequisite is that the script processes variables passed from the web site. Typically, the web site provides a form for user input. Once the user (or JavaScript) submits the form data by either GET or POST method, the script running in the FM Scope will detect the variable content change and make corresponding operations. All parameters passed from the form are visible for the script under the same name of variable, i.e. the parameter content can be accessed by *%parameter*.

Summary: The web site cannot execute script commands directly but if a script is already running, the web site can control its flow by means of the variable content given by GET/POST methods.

Socket control

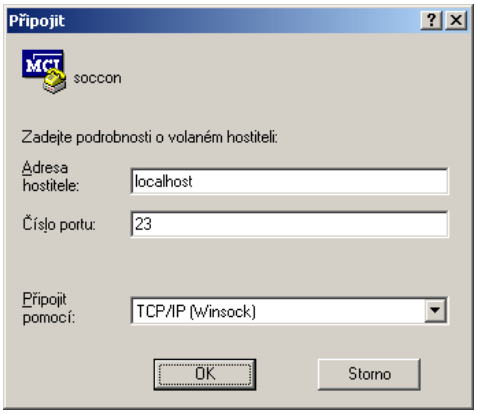
The socket control feature is intended only for advanced users. It's a simple method how to control the application from another via TCP/IP sockets. This function is however primarily not intended for remote control of the equipment (see section 'Connection' instead).
Before attempting to use this function 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.

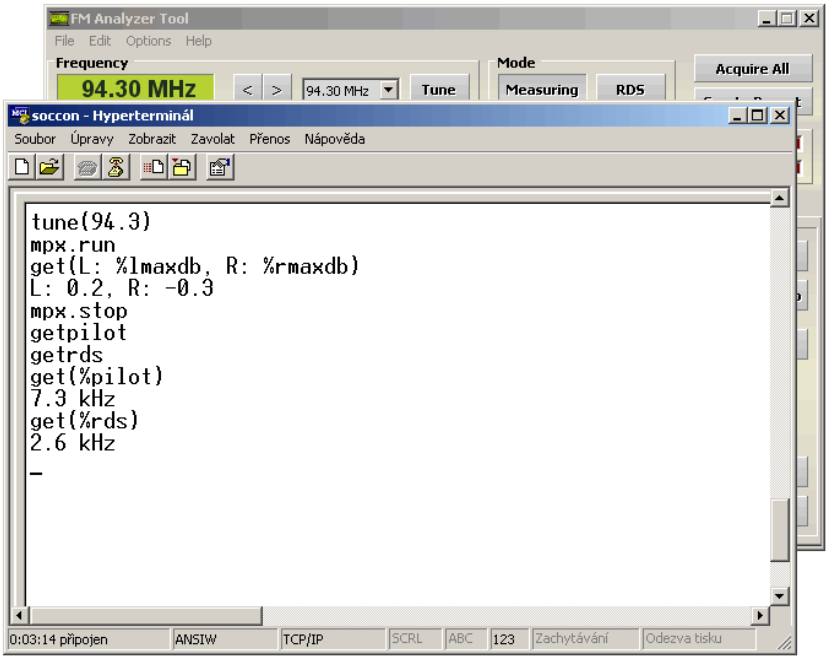
The Socket Control accepts all script commands and moreover it provides also backward channel with the aid of the command get (see the list of commands). Each command entered is considered as a separate script. If a script file is being processed, all commands entered by the Socket Control are placed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

How to start?

One of the best illustrations of use is to control the application from Windows Hyperterminal (or other terminal, like PuTTY). Run the terminal, select TCP/IP connection type and fill the server address and port.



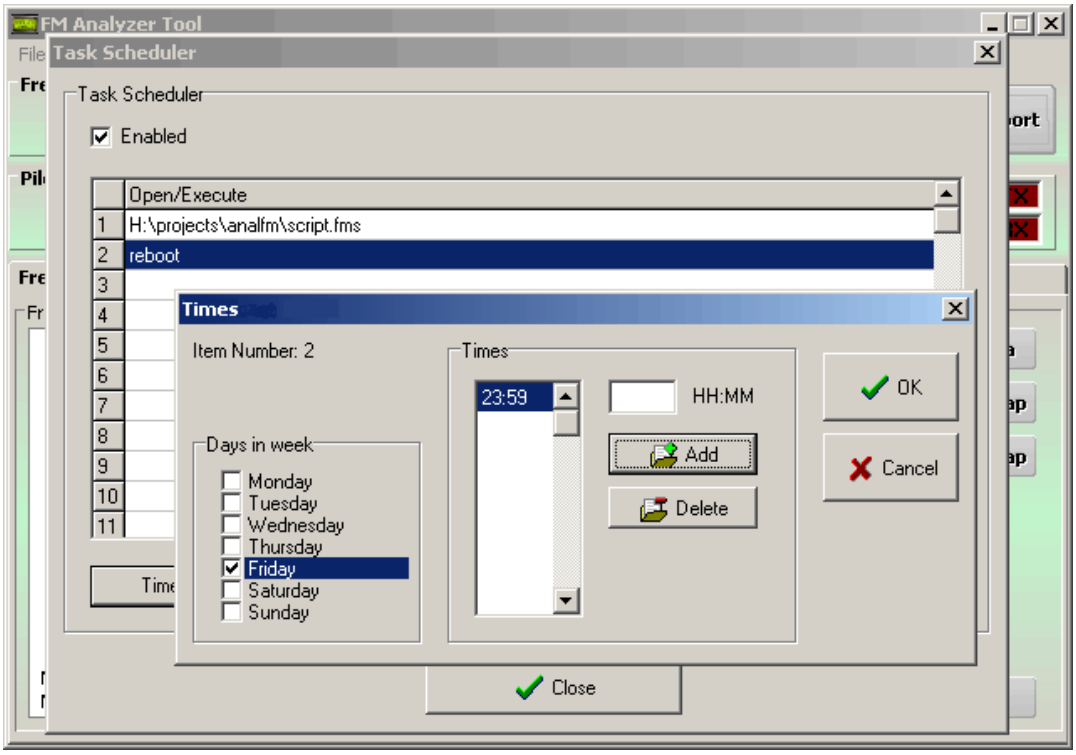
Click OK to connect. Make sure the Socket control feature is enabled and the application is running.



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 the connection 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 a script file or any other file.

These keywords are also accepted in the task line:
exit – Exits the FM Scope application
reboot – Reboots the PC
shutdown – Shutdowns the PC
send: – Sends any RDS command, for example SEND:096200*F

Tips

- Set read-only attribute for fmscope.ini to avoid unwanted configuration changes.
  The fmscope.ini file as well as the script variable content file is usually placed in local settings folder, for example:
  C:\Documents and Settings\[user]\Local Settings\Application Data\FM Scope\
- Any reports are welcome! Please send your suggestions or bug reports. It will help to create new improved version of this application.
- Visit the Technical forum on our website.

SetProperty and ReadProperty options

Property	Value expected or returned
USPTY	Integer value, 0 (disabled) or 1 (enabled)
OnlineUpdate	Integer value, 0 (disabled) or 1 (enabled)
ScanRDS	Integer value, 0 (disabled) or 1 (enabled)
AlwaysCheckRDS	Integer value, 0 (disabled) or 1 (enabled)
InvertColors	Integer value, 0 (disabled) or 1 (enabled)
HighContrast	Integer value, 0 (low) or 1 (high)
DeviationLimit	Integer value, in kHz
DeviationTolerance	Integer value, in kHz
PmLimit	Floating point value, in dBr
PmTolerance	Floating point value, in dBr
ExecutionSpeed	Integer value, 0 (lowest speed, 1 line / sec.) ... 4 (highest speed, 100 lines / sec.)
DebugMode	Integer value, 0 (disabled) or 1 (enabled)
LogErrors	Integer value, 0 (disabled) or 1 (enabled)
NoBreak	Integer value, 0 (disabled) or 1 (enabled)
HistogramBitmapWidth	Integer value, in pixels
HistogramBitmapHeight	Integer value, in pixels
PmBitmapWidth	Integer value, in pixels
PmBitmapHeight	Integer value, in pixels
MPXBitmapWidth	Integer value, in pixels
MPXBitmapHeight	Integer value, in pixels
RFBitmapWidth	Integer value, in pixels
RFBitmapHeight	Integer value, in pixels
BandscanBitmapWidth	Integer value, in pixels
BandscanBitmapHeight	Integer value, in pixels
OthersBitmapWidth	Integer value, in pixels
OthersBitmapHeight	Integer value, in pixels
DetailsInGraphs	Integer value, 0 (disabled) or 1 (enabled)
PmBoundaryMin	Integer value, -12 ... 11
PmBoundaryMax	Integer value, -11 ... 12
PmXAxisLength	Integer value, 1 ... 120, in minutes
WatermarkEnabled	Integer value, 0 (disabled) or 1 (enabled)
WatermarkText	Text string
WatermarkSize	Integer value, 0 (first option) ... 8 (last option)
WatermarkPosition	Integer value, 0 (first option) ... 2 (last option)
TrayIconHint	Text string (can't be read by readproperty)
StatusBarText	Text string
FTPHost	Text string
FTPUserName	Text string
FTPPassword	Text string
FTPRemoteDir	Text string
FTPLocalDir	Text string
FTPPassive	Integer value, 0 (disabled) or 1 (enabled)
SMTPServer	Text string
SMTPFrom	Text string
SMTPRequiresAuthentication	Integer value, 0 (disabled) or 1 (enabled)
SMTPSecureConnection	Integer value, 0 (disabled) or 1 (enabled)
SMTPUserName	Text string
SMTPPassword	Text string
MySQLServer	Text string
MySQLDb	Text string
MySQLUserName	Text string
MySQLPassword	Text string
Encoding	Text string, either UTF-8 or ANSI. Affects text save functions in scripts.