

Controller of Radio-Frequency Generators for Quadrupole Analyzers

Firmware Version 1-10

User Manual

Document version A, created on May-04-2015

Contents

- Technical Data 6
 - Characteristics6
 - Human Interface6
 - Digital Interface6
- Brief Description 8
- Quick Setup Guide 10
- Human Interface..... 12
 - Main Program Menu 13
 - Device Control 15
 - Generator Control 15
 - Fan Setup..... 17
 - Temperature Setup 19
 - Interface Control..... 21
 - Device Monitoring 22
 - Housekeeping Data 22
 - Power Monitor 24
 - RFG Monitor..... 26
 - Fan Monitor 27
 - Device Information 28
 - System Information 28
 - Hardware Information 29
 - System Statistics 30
- Software Utilities 31
 - Utility RFGTester 31
 - Tips 32

Utility FlashLoader	35
Utility RemoteControl	37
Driver Installation	39
Installation of the Virtual Port for the USB Interface	39
Installation of the Virtual Port for the LAN Interface	40
Software Interface	42
Functionality of the Software Interface	42
Communication Control	47
Function COM_RFG_SetDataRate.....	47
Function COM_RFG_GetDataRate	47
Function COM_RFG_Open	48
Function COM_RFG_Close	49
Function COM_RFG_Purge.....	49
Function COM_RFG_Buffer_State	49
Function COM_RFG_Device_Purge.....	50
RF-Generator Control	51
Function COM_RFG_Get_Amplitude	51
Function COM_RFG_Set_Amplitude.....	51
Function COM_RFG_Get_Enable	52
Function COM_RFG_Set_Enable.....	52
Monitoring	53
Function COM_RFG_Get_HousekeepingData.....	53
Function COM_RFG_Get_RFGData	53
Function COM_RFG_Get_PowerData	54
Function COM_RFG_Get_InterlockState	54
Error Handling	55
Function COM_RFG_State	55
Function COM_RFG_ErrorMessage.....	55

Function COM_RFG_IO_State	55
Function COM_RFG_IO_ErrorMessage.....	56
Function COM_RFG_OpenDebugFile.....	56
Function COM_RFG_CloseDebugFile.....	56
Various Functions	57
Function COM_RFG_Version	57
Function COM_RFG_GetHardwareCapability.....	57
Function COM_RFG_GetUptime	58
Function COM_RFG_FW_Ver.....	58
Function COM_RFG_FW_Date	58
Function COM_RFG_Prod_No	59
Function COM_RFG_Prod_ID.....	59
Sample Communication Program	60

Figure List

Fig. 1. Program display in device idle state	13
Fig. 2. Main program menu.....	13
Fig. 3. Dialog box "Generator Control".....	15
Fig. 4. Dialog box "Fan Setup".....	17
Fig. 5. Setting the rotational speed of the particular fan in the dialog box "Fan Setup".	17
Fig. 6. Setup of fan stopping in the dialog box "Fan Setup".	18
Fig. 7. Dialog box "Temperature Setup".	19
Fig. 8. Setting the low temperature threshold in the dialog box "Temperature Setup".....	19
Fig. 9. Setting the shutdown temperature in the dialog box "Temperature Setup".	20
Fig. 10. Dialog box "Interface Control".....	21
Fig. 11. Dialog box "Housekeeping Data".....	22
Fig. 12. Dialog box "Power Monitor".....	24
Fig. 13. Dialog box "RFG Monitor".....	26

Fig. 14. Dialog box "Fan Monitor".	27
Fig. 15. Dialog box "System Information".	28
Fig. 16. Dialog box "Hardware Information".....	29
Fig. 17. Dialog box "System Statistics".....	30
Fig. 18. Utility RemoteControl.....	38

Table List

Tab. 1. Command line parameters of the program RFGTester.	33
Tab. 2. Return values of the interface functions	44
Tab. 3. I/O errors.....	46

Technical Data

Characteristics

- control and monitoring of two symmetrical RF outputs
- monitoring of generator supply voltage and heatsink temperature
- monitoring of generator operating voltage and temperature
- serving an interlock loop
- supply voltage supervising
- microcontroller: 16-bit RISC running at 16 MHz
- user-friendly graphical interface
- non-volatile memory (FRAM) to permanently store device setting
- USB and LAN data interface

Human Interface

- monochrome LCD display 128x64 pixel
 - pixel size: 0.5 mm
 - pixel color: yellow, background: blue
 - background illumination: white LED
- keypad: 5 keys: 4x direction + 1x "enter"
- rotary encoder:
 - 24 positions per revolution, integrated press button
 - located beneath the LCD for general control
- optional external shutdown button via the interlock loop

Digital Interface

- interface parameters:
 - data rate: 230.4 kBaud (USB), 115.2 kBaud (LAN)
 - data bits: 8
 - stop bits: 1
 - parity: even
- USB interface according to USB 2.0 standard
 - connector: USB receptacle type B
 - data transfer rate: up to 12 MBit/s (*Full Speed*)
 - effective data transfer rate: >10 kBit/s

- LAN interface according to Ethernet 10/100BaseT standards
connector: LAN receptacle RJ-45
data transfer rate: 10/100 MBit/s (fast Ethernet)
effective data transfer rate: >10 kBit/s

Brief Description

The controller is intended to manage the operation of a radio-frequency (RF) generator. The generator controller is equipped with a USB and/or a LAN data interfaces. They allow to transfer measurement and settings data to or from the device and to control it remotely (see sections "Software Utilities" and "Software Interface").

The device contains a microcontroller that provides the human interface (see section "Human Interface"), controls the USB and LAN data interfaces, and executes several monitoring tasks. It also controls the digital-to-analog converter producing the supply voltage for the generator's end stage. Further, the device contains monitoring circuits of the generator and of the power supplies. They are based on analog-to-digital converters that measure the output amplitude, the generator's operating parameters, and the supply voltages.

To protect the RF generator and its power supply, the controller supervises all operation parameters. According to the measured values of the output amplitude, limiting values of generator's operation parameters are calculated. If the controller discovers that any of the limits is exceeded, the device is deactivated immediately.

The device monitors an interlock loop that disables the operation if it is opened. The interlock loop is typically connected to a vacuum controller, it should disable the device if the vacuum pressure exceeds a certain value above that discharges might occur.

The temperature of the generator and its power supply is measured and supervised. The temperature values are used to control the cooling fans. If the temperature rises above a predefined level, the generator and its power supply are disabled.

The actual settings of the output amplitude and other operation parameters are stored in a non-volatile memory and are automatically restored at the next device startup.

The USB or LAN interface can be used to remotely control the device by simple Windows™ programs. These utilities can also obtain hardcopies, i.e. save the current LCD contents to a bitmap file at the host computer. Alternatively, you may write your own user software based on the supplied software interface (see section "Software Interface"). Upon request, software interface for LabVIEW or other development systems is available. The USB interface can also be used to upgrade

the firmware. For a detailed description, see the section "Software Utilities".

Quick Setup Guide

To prepare the device for an experiment, connect the load, i.e. a quadrupole or another multipole device with suitable cables to the device output. Then check the operation parameters of the device. To do so, follow the next steps:

- Ensure that the proper mains voltage is selected (consult the manual of the main device), connect the line cord and turn the device on. The LCD should light up.
- Press the middle key "enter" and start the interface control by selecting the menu item "Control ► Interface". Check the displayed values and if desired change them. You should enable just the interface that you aim to use. Especially the LAN interface produces a significant amount of heat in the device, thus should be disabled if not used (for details see section "Interface Control"). Close the dialog box when you have finished the setup.
- Close the interlock loop. You may insert a BNC termination plug if the interlock loop should not be used. In a properly designed experiment, however, the interlock loop should be closed by a vacuum control unit when a sufficiently low pressure in the quadrupole chamber is reached.
- Press the middle key "enter" and start the power monitoring by selecting the menu item "Monitor ► Power". Check whether the device recognized that the interlock loop is closed (for details see section "Power Monitor"). Close the dialog box when you have finished checking the interlock loop.
- Press the middle key "enter" and start the generator control by selecting the menu item "Control ► Generator". Adjust the minimum allowable output amplitude by the encoder knob (for the allowed amplitude range see section "Hardware Information"). The measured output amplitude (item "Out") should be almost zero, the minimum indicated value is typically in the range of a few Volts. The shown relative difference does not have any meaning if the generator is not running. The measured output frequency and supply power should be zero.
- Press the encoder knob to enable the device. The output frequency should rise to the nominal value, the measured output amplitude and the supply power should increase. The measured amplitude should reach the set value after a short time. The settling time de-

depends on the output amplitude, the desired value should be reached within several seconds. When this happens, you may adjust the frequency and the symmetry of the output voltage.

- If a popup window with an error message appears, try to localize the source of the fault. Most probable reasons are an opened interlock loop (see above) or an unmatched load. If the device recognizes an load that is not properly adjusted for the operation frequency, the device shuts down after a short time with an overload error message.

! **Warning:** If the device reports an overload error, check the connected load carefully before trying to enable the device again. Although the device is protected by many security measures, it may be damaged by an operation with an improper load. If you connect the generator to a particular load for the first time, consult the manual to the main device how to adjust the load properly.

If you encounter problems, read carefully the corresponding section in this manual. If you cannot solve the problem, contact the manufacturer of the device. Before doing it, obtain hardcopies of the LCD panel showing all monitoring parameters and send them together with the problem description.

Human Interface

The microcontroller's firmware implements a user-friendly graphical interface. The device is controlled by the help of menus and dialog boxes, they provide an intuitive device control.

The device is equipped with a graphic liquid crystal display (LCD), a set of keys (keypad) and a rotary encoder. The keys are arranged in a circle symbolizing the key function: There are four direction keys ("left", "right", "up", and "down") and a middle key for confirmation or selection ("enter").

The keys are used for menu navigation, for selecting dialog items, or for changing various values. The function of the keypad in every device state is symbolized on the LCD immediately above the keypad. A text describes the function of the middle key, alternatively just the symbol "€" is displayed showing that the middle key can be used. Similarly, arrows show which of the keys has an influence on the operation in the current state. When a menu is active, the vertical direction keys are used to change the current selection. The right direction key as well the middle key opens a submenu, provided it is available. The left direction key closes the submenu or the main menu if there was no opened submenu. The middle key selects the menu item and launches the corresponding action.

The rotary encoder beneath the LCD is used to change numerical values. The function of the encoder in every device state is symbolized on the LCD immediately above the encoder knob. The encoder's push button is used to turn values on or off. If this feature is available, the symbol "☐" is shown on the LCD above the encoder knob. For more details, see next sections.

The encoder uses an enhanced speed control. This enables you to set precisely any desired value or rapidly make large changes, since the value steps are proportional to the rotational speed of the particular encoder. You can change the corresponding value in small steps when you rotate the encoder slowly or make large changes when you spin the encoder knob rapidly.

Main Program Menu

When the device starts, the initialization takes place and the manufacturer logo is shown for couple of seconds. Then, the device enters the idle state (see Fig. 1). The lower right corner shows the uptime, i.e. the time elapsed from the last device start.



Fig. 1. Program display in device idle state.

By pressing the middle key "enter", the main menu can be accessed (see Fig. 2). You can control the device (submenu "Control", see also section "Device Control"), monitor the device function (submenu "Monitor", see also section "Device Monitoring"), obtain information about the device (submenu "Information", see also section "Device Information"), or manage the settings (submenu "Adjustment").

The system settings are the data that can be modified in the submenu "Control". You can save the current settings (menu item "Adjust-

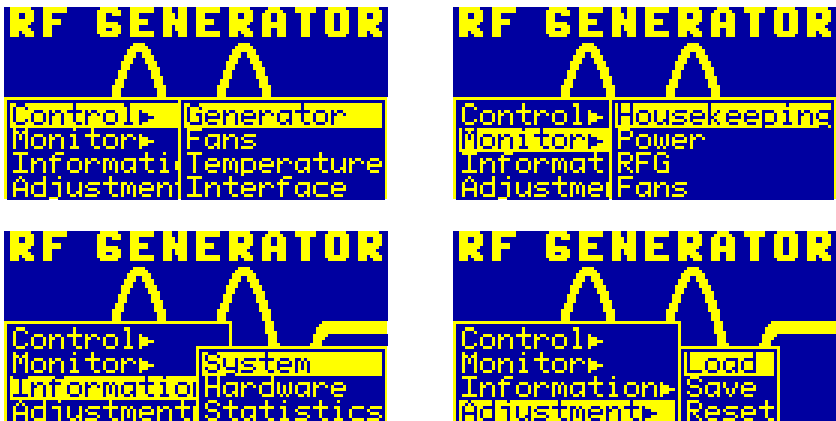


Fig. 2. Main program menu.

The figure shows the different submenus of the main program menu.

ment ► Save"), load the settings back (menu item "Adjustment ► Load"), or reset them to default values (menu item "Adjustment ► Reset"). The recent system settings are saved automatically to a non-volatile memory when the device powers down and are loaded back when the device starts (see also section "Housekeeping Data"). You may save the settings using the main menu prior to the next device shutdown if you have made changes that you wish to keep permanently. When you choose to load the settings back, the settings will be reloaded and the last saved state will be restored, i.e. the settings will be set to the state immediately before the last shutdown or after the last startup, or to the state when you have saved the settings. Further, you may reset the settings to their default values. This is advisable if you wish to restore the original device state.

Device Control

Several dialog boxes are available to control the device and to setup the device behavior.

Generator Control

Using the dialog box "Generator Control", you can set the amplitude of the output voltages and turn the generator on or off (see Fig. 3).

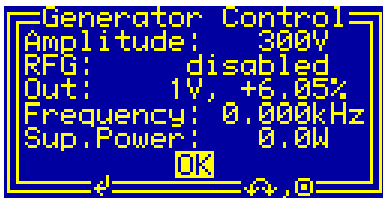


Fig. 3. Dialog box "Generator Control".

To set a particular amplitude, rotate the encoder knob beneath the LCD. The displayed value "Amplitude" is the desired amplitude, the minimum and maximum settable values are defined in the dialog box "Hardware Information" (see Fig. 16 and section "Hardware Information").

The amplitude measured at the generator outputs is indicated as the value "Out". The device shows the mean value measured on both RF outputs (V_{RF1} and V_{RF2}) and the relative difference between them according to the formula $(V_{RF1} - V_{RF2}) / (V_{RF1} + V_{RF2})$ shown in percents. Thus, a relative difference of 0.1% means, for instance, that at an amplitude of 1 kV, the voltages differ by 1 V from the displayed value - i.e. the output amplitudes are 999 V and 1001 V, respectively. For more details about the amplitude measurement, see section "RFG Monitor".

If the device is equipped with a rotary capacitor for symmetry adjustment, you may set the difference to zero to adjust the device to a particular load. This is advisable especially for linear multipoles, the zero difference ensures a zero RF potential on the multipole axis, thus a minimum influence of the RF to the ion transmission.



Note that the relative difference of the amplitudes may be slightly voltage-dependent. When the device is turned off, the measured ampli-

tudes reflect integrated noise only and the indicated values of the amplitude and of the relative difference do not have any real meaning.

The item "Frequency" displays the measured output frequency (see section "RFG Monitor"). The exact value of the output frequency depends on the physical parameters of the output circuitry. The frequency may show a light voltage and temperature dependency. If the device is equipped with a rotary capacitor for frequency adjustment, you may set the output frequency to a desired value to adjust the device to a particular load. An adjustment during the device operation is usually not necessary, the typical frequency variations are less than 1%.

The last item "Sup.Power" shows the supply power of the generator's end stage (see section "Power Monitor"). The most part of the power is due to power losses in the output transformer, in the cables, and in the load. The device supervises the power consumption and turns the device off if the expected values are exceeded. You should, however, observe the value since it reflects the shape of the connected load. In most cases, a higher power consumption is an indication for a coming fault.

When the device has been powered on or when the interlock loop (see also section "Power Monitor") is open, the device is turned off. This is indicated by the state "disabled" in the second row labeled by "RFG". You may turn the device on and off again by pressing the encoder knob.

! **Attention:** Be sure that the proper load is connected to the outputs before you turn the device on. The generator is equipped with several protection measures that prevent overload - they turn the device off if the load does not ensure operation at the desired frequency (for more details, see sections "Power Monitor" and "RFG Monitor").

! **Attention:** The generator produces lethal voltages. Never touch the outputs or approach the outputs with conductive objects. The high RF voltage may ignite a discharge that may lead to serious injuries.

Note that the adjustment of the output amplitude has a time constant of several 100 ms. This implies that the device may need a noticeable time to reach the desired output amplitude.

Fan Setup

The dialog box "Fan Setup" (see Fig. 4) shows the current settings of the fan controller and enables you to change them.

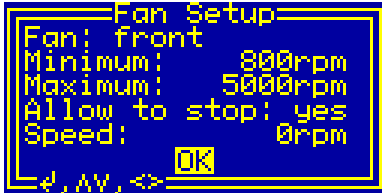


Fig. 4. Dialog box "Fan Setup".

Use the horizontal direction keys to switch between the fans. The device contains three fans (front, internal, and rear), each of them can be set up independently on the other ones.

Use the vertical direction keys to change the currently selected item. The selected item is highlighted by inverting the displayed characters. The settings can be modified by the rotary encoder or by pressing the middle key "enter". Further, using the middle key "enter", the dialog box can be closed (see the selection shown in Fig. 4).

The values "Minimum" and "Maximum" define the minimum and maximum rotational speed of the fans, they can be changed by the rotary encoder (see Fig. 5). Dependent on the device temperature, the rotational speed of the fans varies between the specified values. The minimum rotational speed is set at the lower temperature threshold (see Fig. 8), the maximum rotational speed is applied at the upper temperature threshold (see also section "Temperature Setup").



Fig. 5. Setting the rotational speed of the particular fan in the dialog box "Fan Setup".

If the respective item is selected, the rotational speed of the fans is regulated to the displayed value. This makes it possible to evaluate

the function of the fans directly and choose a speed interval in that the fans run reliably. The real fan speed in revolutions per minute (rpm) is shown at the bottom of the dialog box by the item "Speed".

The value "Allow to stop" enables stopping the particular fan on low temperatures, it can be changed by pressing the middle key "enter" (see Fig. 6). If stopping is enabled and the device temperature is lower than the preset threshold (see Fig. 8 and section "Temperature Setup"), the fan is stopped and does not produce any acoustic noise. If stopping is disabled, the fan is running regardless of the device temperature.

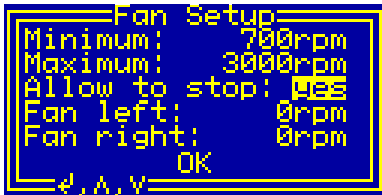


Fig. 6. Setup of fan stopping in the dialog box "Fan Setup".

Note that the selected setting is immediately applied, thus if the device temperature is sufficiently low, the fans can be turned on or off by changing this setting. If the minimum rotational speed of the fan is too low, the fan may not start reliably. Dependent on the fan construction, the fan may start with a high speed that is later decreased by the fan controller. If the minimum rotational speed is too low, the fan controller cannot regulate the fan properly and it stops again during the startup. This leads to an infinite loop. If you observe such difficulties, increase the minimum rotational speed to a value at that the fan starts without any irregularity.

Temperature Setup

The dialog box "Temperature Setup" (see Fig. 7) shows the current settings of the fan controller and enables you to change them.

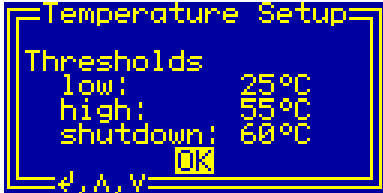


Fig. 7. Dialog box "Temperature Setup".

Use the vertical direction keys to change the currently selected item. The selected item is highlighted by inverting the displayed characters. The settings can be modified by the rotary encoder. Using the middle key "enter", the dialog box can be closed (see the selection shown in Fig. 7).

The values "low" and "high" define the temperature interval in that the fan controller linearly increases the fan speed, they can be changed by the rotary encoder (see Fig. 8). If the device temperature is equal or lower than the lower threshold "low", the fans rotate with the minimum speed (see Fig. 5). Vice versa, if the device temperature is equal or higher than the upper threshold "high", the fans rotate with the maximum speed (see also section "Fan Setup").

The device temperature is the maximum value among the values measured by the temperature sensors. In the most cases, the generator's power supply shows the highest temperature (see Fig. 12 and section "Power Monitor"), thus the device temperature equals to this value.



Fig. 8. Setting the low temperature threshold in the dialog box "Temperature Setup".

The value "shutdown" determines the temperature at that the device shuts down, it can be changed by the rotary encoder (see Fig. 9). If the device temperature exceeds this threshold, it is shut down independently on the fan state. In this case, the generator and its power supply unit are disabled. If this happens, you need to activate the generator again to resume the operation.



Fig. 9. Setting the shutdown temperature in the dialog box "Temperature Setup".

The minimum settable value of all values is 20°C, the maximum is 60°C. The lower threshold must be lower than the higher one and the higher threshold must be lower than the shutdown temperature. The difference between the values must be larger than 1°C, the software prevents you from setting wrong values.

Note that a too high operation temperature of the device may shorten its lifetime and decrease its stability. On the other hand, a too low shutdown temperature may prevent the device from normal operation and too low temperature thresholds of the fan controller may lead to an unnecessary high noise production. You should set up values that ensure the lowest possible operation temperature in the particular environment without affecting the function of the device.

Interface Control

The dialog box "Interface Control" (see Fig. 10) shows the current interface settings and enables you to change them.

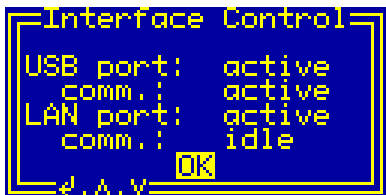


Fig. 10. Dialog box "Interface Control".

Use the vertical direction keys to change the currently selected item. The selected item is highlighted by inverting the displayed characters. By pressing the middle key "enter", the settings can be modified or the dialog box can be closed (see the selection shown in Fig. 10).

The setting of each particular interface can be changed from "active" or "idle" to "off". If the interface state is "off", the interface is disabled and no communication is possible. If the LAN interface is disabled, also its power supply is deactivated to lower heat production. The remaining interface states indicate that the interface is enabled. If the displayed state is "active", the interface is activated and a communication can be launched. If the displayed state is "idle", the interface is activated but cannot communicate. For the LAN interface, the "idle" state is displayed for a short startup time when the interface is being activated and changes to "active" after several 100 ms. For the USB interface, the "idle" state is displayed if there is no connection to a remote host. If a cable connects the device with a functioning remote host, the state changes to "active".

The communication state shown as the items "comm." indicates the state of the handshake lines of the particular interface. The communication state changes between "active" and "idle" depending on the communication activity on the particular interface. The communication state is "idle" if you do not access the device and it changes to "active" if you start a communication software on the remote host.

Device Monitoring

To monitor the function of the device, several dialog boxes are available. You can use them to check the function of the device if you encounter any problem.

Housekeeping Data

The dialog box "Housekeeping Data" (see Fig. 11) shows the supply voltages of the device and the temperature and the load of the microcontroller.

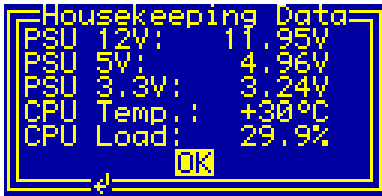


Fig. 11. Dialog box "Housekeeping Data".

The value "PSU 12V" is the main voltage supplying all internal circuits. It is generated by a switching regulator from a rectified output of a secondary winding of the main transformer. The nominal value is 12 V, when it drops down below 11.5 V, the device controller starts to prepare the power shutdown by saving the settings to a non-volatile memory. When the value approaches 11 V, for instance during a mains dropout, the controller is reset and halted until the supply voltage recovers.

The values "PSU 5V" and "PSU 3.3V" are the output voltages of the switching regulators supplying the low-voltage logical circuits. The nominal values are 5.0 V and 3.3 V, respectively.

The value "CPU Temp." shows the temperature of the microcontroller. The temperature is measured by an on-chip sensor, the accuracy is around 1°C.

The value "CPU Load" is the obtained load of the microcontroller. The value gives the approximate percentage of the time in that the microcontroller is active. The load increases especially if the software utility for remote controlling is active (see section "Utility RemoteControl").

The values are updated about 2-times per second, the CPU load once per second.

Power Monitor

The dialog box "Power Monitor" (see Fig. 12) shows the output parameters of the power supply unit supplying the generator's end stage, its heatsink temperature, and the state of the interlock input.



Fig. 12. Dialog box "Power Monitor".

The values "Sup. volt." and "Sup. curr." are, respectively, the values of the voltage and current supplying the generator's end stage. The power supply unit is a switching regulator that produces the supply voltage from a rectified output of a secondary winding of the main transformer. The regulator can produce output power higher than 200 W. The maximum output voltage is limited to a value defined in the dialog box "Hardware Information" (see Fig. 16 and section "Hardware Information").

The microcontroller protects the RF generator by supervising the output parameters of the voltage regulator. For the currently measured output amplitude, it obtains the maximum allowable voltage and current supplying the generator's end stage and turns the generator and its supply off if either value is exceeded.

The value "Temperature" is the temperature of the regulator's heatsink. The measurement uses a semiconductor sensor and has an accuracy of about 2°C. The temperature of the regulator's heatsink is usually the highest one in the device, thus the fan controller assumes that the device temperature equals to this value (see also section "Temperature Setup").

The value "Interlock" shows the interlock loop status. The possible values are "closed" or "open". If the interlock state is "open", the device cannot be activated.

The microcontroller checks the values continuously and turns the device off if the temperature rises above the allowed value (see also section "Temperature Setup") or if the interlock loop is opened.

The values are updated about 2-times per second, the interlock state once per millisecond.

RFG Monitor

The dialog box "RFG Monitor" (see Fig. 13) shows the measured operation parameters of the RF generator.

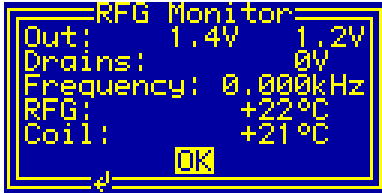


Fig. 13. Dialog box "RFG Monitor".

The values "Out" are obtained by fast precise rectifiers that measure the output voltages. The rectifiers measure the mean absolute value of each output voltage, the measured values are scaled to show the signal amplitudes. The rectifiers are adjusted such that they measure exactly the same voltage at the operation frequency, thus the measured values can be used to accurately adjust the symmetry of the output voltages.

The voltage "Drains" is the peak voltage at the drains of the output MOSFETs in the generator's end stage. The microcontroller protects the generator by supervising this value. It deactivates the generator and its supply if the voltage limit for a particular output amplitude is exceeded.

The item "Frequency" displays the measured output frequency of the generator. The value is obtained by a digital counter from attenuated output signals. If the device supplies a high-resolution mass filter, the measured output frequency can be used to determine the proper amplitude value to adjust the desired mass range.

The items "RFG" and "Coil" are the temperatures of the generator's heatsink and of the cooling air of the output transformer, respectively. The measurement uses semiconductor sensors with an accuracy of about 2°C. The microcontroller checks the values continuously and turns the device off if any temperature rises above the allowed value (see also section "Temperature Setup").

The values are updated about 2-times per second.

Fan Monitor

The dialog box "Fan Monitor" (see Fig. 14) shows the state of the fans located in the device.



Fig. 14. Dialog box "Fan Monitor".

The shown values indicate the rotational speed in revolutions per minute (rpm) of each particular fan. For the details of the fan control, see also sections "Fan Setup" and "Temperature Setup".

Device Information

To obtain information about various parts of the device, several dialog boxes are available. You can use them to check the firmware version and other parameters that may be required when you encounter any problem and like to contact the manufacturer.

System Information

The dialog box "System Information" (see Fig. 15) summarizes general information about the device.



Fig. 15. Dialog box "System Information".

The text "ID" shows the product identification, "Product No." is the unique product serial number. The items "HW version" and "FW version" designate the hardware and firmware versions, respectively. The text "FW date" is the compilation date of the firmware.

In case of any trouble with the device, please use these values when contacting the manufacturer.

Hardware Information

The dialog box "Hardware Information" (see Fig. 16) summarizes general information about the device hardware.

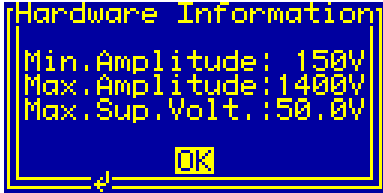


Fig. 16. Dialog box "Hardware Information".

The items "Min.Amplitude" and "Max.Amplitude" define the minimum and maximum settable values of the output amplitude (see section "Generator Control").

The item "Max.Sup.Volt." define the maximum allowable supply voltage of the generator's end stage (see also section "Generator Control").

These values were programmed by the manufacture to determine the safe operation range of the device and/or the connected load.

The values from the dialog box "Hardware Information" can be remotely accessed by software (see the function `COM_RFG_GetHardwareCapability`), thus can be used to modify the software function according to the installed hardware features.

System Statistics

The dialog box "System Statistics" (see Fig. 17) shows information about the operation time of the device.

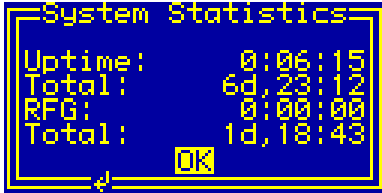


Fig. 17. Dialog box "System Statistics".

The "Uptime" shows the time elapsed since powering on the device or since the last firmware restart. The next line "Total" shows the total operation time of the device.

The time "RFG" shows the active time of the RF generator elapsed since powering on the device or since the last firmware restart. The next line "Total" shows the total operation time of the RF generator.

Software Utilities

The software utilities can be found in the directory "Program" of the enclosed software package. Before using them, the virtual USB and/or LAN port drivers must be installed (see section "Driver Installation"). The utilities do not require any additional installation, you need only to copy them to a suitable directory on your computer.

Utility RFGTester

The RFGTester is a simple Windows™ program that runs in text mode and is supposed to be launched from a command line. It enables you to control the device remotely and check the operation parameters. Launching the utility `RFGTester.exe` without any parameters displays a simple help:

```
RFGTester
```

To start the program, you need to specify at least the number of the virtual COM port to which the device is attached:

```
RFGTester COMNumber
```

If the device is, for instance, attached to the port COM3, start the utility by the following command:

```
RFGTester 3
```

The program should start without any error message. In case of any problem, check whether the port number matches the system settings (see sections "Driver Installation").

To check the communication, press the key 'p' to obtain the product identification text. The device should respond as follows:

```
Product identification: RFG-300W 1-00
```

If an error occurs, please consult the section "Functionality of the Software Interface". The tables 2 and 3 explain the possible error messages; they should help you to localize the reason for the software failure.

If the device responds properly, you may try other program commands. Press '?' to obtain the help listing of all available commands. Finally, press 'Esc' to exit the program.

In practice, you may prefer to use the command line mode instead of the interactive mode. The command line mode allows you, for instance, to save the complete commands in batch files for repeated usage. For instance, to view the set amplitude value of a device attached to COM3, enter the following command:

```
RFGTester 3 -a -t
```

Similarly, the command:

```
RFGTester 3 -e -t
```

displays the enable state of the device, the command:

```
RFGTester 3 -f -t
```

shows the measured values of the generator data.

To set the output amplitude to 500 V, please enter

```
RFGTester 3 -A 500 -t
```

to enable the device, enter the command:

```
RFGTester 3 -E Y -t
```

Table 1 summarizes all allowable command line parameters of the program RFGTester. The parameters are processed from left to right. If an error in the command line is encountered, the program stops with an error text showing the allowed values of the parameters.

If the parameter `-t` is found, the program stops without processing any following parameter. If you do not specify the parameter `-t` at all, the program does not stop and enters the interactive mode after having processed the complete command line.

The quiet program mode reduces the program text output, contrary to that, the debug mode provides a detailed output for error analysis.

Tips

You can combine several command line parameters. For instance, the following command:

```
RFGTester 3 -A 500 -E Y -t
```

set the output amplitude to 500 V and enables the device.

Tab. 1. Command line parameters of the program RFGTester.

Parameter	Explanation
-A f	set the RF amplitude to the value f
-a	get the RF amplitude
-E b	set the RF state to the value b (Y = enable, N = disable)
-e	get the RF state
-i, -I	get the interlock state
-h, -H	get the housekeeping data
-s, -S	get the power supply data
-f, -F	get the RF-generator data
-c, -C	get the hardware capabilities
-d, -D	get the device firmware date
-v, -V	get the device firmware version
-p, -P	get the product identification text
-n, -N	get the product number
-u	get the device uptime
-U	show the device uptime continuously
-k, -K	simulate the keypad, this starts an interactive procedure that must be finished by pressing 'Esc'
-l	get display data in 8 steps
-L	get whole display data at once
w f g p, W f g p	cycle the RF amplitude between the voltages f and g, the cycle period is p seconds, the cycling must be finished interactively by pressing SPACE
x s f g d p, X s f g d p	create test protocol into the file s, step the RF amplitude between the voltages f and g with the step d, the step duration is p seconds, X introduces a pause after each voltage step

Parameter	Explanation
-b, -B	get the device buffer status
-z, -Z	purge the communication
-t, -T	terminate the program
-1	higher data rate (USB interface)
-2	lower data rate (LAN interface)
-q, -Q	quiet mode
-g	debug mode
-G	debug mode with output into Debug.txt
-?	show the online help and terminate

To save the current device configuration, you may enter the command:

```
RFGTester 3 -q -a -t > RFGConfig.txt
```

It outputs the set amplitude value into the file RFGConfig.txt. Similarly, the command:

```
RFGTester 3 -q -f -s -h -t > RFGParams.txt
```

outputs the set measured generator, power supply, and housekeeping data into the file RFGParams.txt. Using this file, you can protocol the measurement.

If not explicitly specified, the program detects the data rate for the communication automatically. This is necessary since the USB and LAN interfaces use different data rates. In the automatic mode, the program starts with the lower data rate for the LAN interface, if an error-free communication cannot be established, the program switches to the higher data rate for the USB interface. The disadvantage is a longer startup time for the USB communication.

To speed up the program startup, the data rate can be set by two switches ("1" and "2", see Tab. 1). The following command

```
RFGTester 3 -1
```

sets the higher data rate used by the USB interface. Vice versa, the following command

```
RFGTester 3 -2
```

sets the lower data rate used by the LAN interface.

! **Note:** The switches "1" and "2" for setting the data rate must be specified prior to other switches requiring any communication with the device. The program processes the commands from left to right, thus any parameter requiring data transfer will open the communication channel with the data rate specified so far, thus use the automatic rate detection in no rate specification was given.

Utility FlashLoader

The FlashLoader is a simple 32-bit Windows™ program running in text mode and is supposed to be launched from a command line. It enables you to upgrade the firmware of the device. You should perform the upgrade if you have received or downloaded a new firmware file from the device manufacturer. Launching the utility `FlashLoader.exe` without any parameters displays a simple help text with the expected syntax of the command line.

Before upgrading the firmware, you should first test the device and the communication by verifying the current firmware version. To do so, connect the device via USB to the host computer and start the following command:

```
FlashLoader COMNumber Firmware.txt -v
```

where `COMNumber` is the number of the virtual COM port to which the device is attached and `Firmware.txt` is the file containing the current firmware. If the device is, for instance, attached to the port COM3, start the utility by the following command:

```
FlashLoader 3 Firmware.txt -v
```

If the proper parameters are specified, the verification starts and produces the following output:

```
Code file Firmware.txt from 01/02/2015, 12:00:00
Flash Loader 1.12
Verifying code file Firmware.txt
Verifying finished at Fri, 01/02/2015, 13:00:00
50258 (C452h) bytes processed, 49920 (C300h) bytes
verified
Resetting the target
Program finished ok
```

During the verify procedure, a message box is displayed at the device LCD informing the user that the flash loader has been activated. When the verify finishes, the device is restarted.

! **Attention:** Never start the utility if the device is active. The utility stops all processes running at the microcontroller, thus the monitoring is not available and the device would not be deactivated if a critical situation arises. To be completely sure that the device cannot be activated before the FlashLoader is started, open the interlock loop.

! Note that the firmware upgrade can be done using the USB interface only, the utility fails if you try to start it using the LAN interface.

If any error occurs, do not proceed with the firmware upgrade. To locate the reason for the error, follow the instructions described in the section "Utility RFGTester".

If the verify has succeeded, you may start the firmware upgrade by entering the command:

```
FlashLoader COMNumber Firmware.txt
```

where `COMNumber` is the number of the virtual COM port and `Firmware.txt` is the file with the new firmware. The program should produce the following output:

```
Code file Firmware.txt from 01/02/2015, 12:00:00
Flash Loader 1.12
Programming code file Firmware.txt
Programming finished at Fri, 01/02/2015, 13:00:00
94634 (171AAh) bytes processed, 94976 (17300h)
bytes programmed
Resetting the target
Program finished ok
```

During the programming procedure, a message box is displayed at the device display. When the programming finishes, the device is restarted with the new firmware.

If an error occurs, the flash loader utility at the microcontroller may remain active. This is the case if the message box at the device display is still present and the device did not restart. In this case, you may retry the action with the command line parameter '-i':

```
FlashLoader COMNumber Firmware.txt -i
```

This will prevent the utility at the host computer from initializing the flash loader utility at the microcontroller again and it will just try to reprogram the file `Firmware.txt`. If the error persists, contact the manufacturer.

! **Attention:** You must not power down the device if the firmware upgrade did not succeed. Otherwise, the device will not operate properly or might even not restart at all. Should this happen, it would be necessary to reprogram the device in the factory.

If the current firmware is damaged so that the device is inoperable, you may try to start the flash loader utility at the microcontroller manually. Press, simultaneously, the horizontal direction keys and hold them while powering on the device. If this small part of the firmware is still working, the flash loader will start. Then, try to launch the Flash-Loader with the command line parameter '-s':

```
FlashLoader COMNumber Firmware.txt -s
```

This does not start the flash loader utility at the microcontroller but will only try to reprogram the file `Firmware.txt`. If an error occurs that you cannot solve, contact the manufacturer.

Utility RemoteControl

The RemoteControl is a 32-bit Windows™ GUI program that enables you to control the device remotely and obtain a hardcopy of the current LCD content.

To launch the utility, start the following command:

```
RemoteControl COMNumber
```

The number `COMNumber` is the number of the virtual COM port to which the device is attached. When started with a proper parameter, the program opens a dialog box that shows the current content of the LCD at the device (see Fig. 18). The direction keys and the encoder can be simulated by clicking the corresponding buttons or using the keyboard shortcuts. For the detailed description of them, see the on-line help.

You may obtain a hardcopy of the LCD at any time by selecting the menu item File ► Save Hardcopy or by pressing Ctrl+S. The program

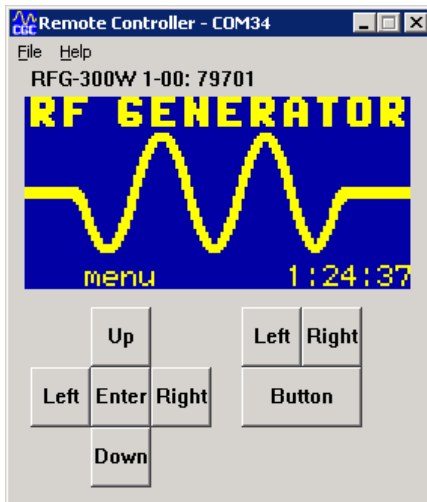


Fig. 18. Utility RemoteControl.

opens a dialog box to select the destination bitmap file. The hardcopy is performed immediately without waiting for any confirmation.

Note that the utility RemoteControl continuously transfers display data over the USB or LAN interface. Compared to the idle state, this leads to a higher load of the microcontroller controlling the device. Further, the utility prevents any other program from accessing the device via the interface.

As the utility RFGTester, the program RemoteControl detects the data rate for the communication automatically. You may override it and speed up the program start by specifying the switches "1" or "2" (see Utility RFGTester). The following command

```
RemoteControl COMNumber -1
```

sets the higher data rate used by the USB interface. Vice versa, the following command

```
RemoteControl COMNumber -2
```

sets the lower data rate used by the LAN interface.

Driver Installation

Installation of the Virtual Port for the USB Interface

The virtual port driver is required for the operation of the device with a USB interface. The installation files are located in the directory "USB" of the enclosed software package. If you use the operating system MS Windows™, please note the following:

- As an alternative to the drivers from the enclosed software package, you can use the update function of the operating system at the host computer or download the most recent driver from the homepage of the manufacturer of the USB adapter. The drivers are located at the following address: <http://www.ftdichip.com/Drivers/VCP.htm>. Please choose the correct driver version according to your operating system.
- To install the driver, administrative rights are required.
- The installation is described in detail in one of the pdf documents located in the abovementioned directory. Please read this description carefully before starting the installation.
- After the installation, the number of the virtual port can be set. You can change the settings in the device manager by opening the settings of the device *USB Serial Port*. To modify the settings, administrative rights are required. The settings are applied immediately, you do not need to reboot the PC to activate them.

The software can also be used at computers running other operating systems than MS Windows™, for instance, Linux. You can launch the software utilities using the Windows™ API translator wine (see <http://www.winehq.org/>).

Starting with Linux Kernel 3.0.0-19, all FTDI devices are already supported without the necessity of compiling additional kernel modules. For more details, see the directory "USB" of the enclosed software package or consult the homepage of the manufacturer of the USB adapter: <http://www.ftdichip.com/Drivers/VCP.htm>.

The system has to be configured in the following way:

- Use, for instance, the program 'dmesg' to find out to which USB port the device is attached: Look for a line similar to "FTDI USB Serial Device converter now attached to ttyUSB0"

- Link the Linux device to the virtual COM port of wine:

```
ln -s /dev/ttyUSB0 ~ /.wine/dosdevices/com3
```

This assumes that the device is attached to ttyUSB0 and will be linked to COM3

Installation of the Virtual Port for the LAN Interface

The virtual port driver is required for the operation of the device with a LAN interface. The installation files are located in the directory "LAN" of the enclosed software package. If you use the operating system MS Windows™, please note the following:

- Locate the installation program in the enclosed software package or download an actual version of the software from the following address: <http://tibbo.com/downloads/soi/tdst.html>. Please choose the correct driver version according to your operating system.
- To install the driver, administrative rights are required.
- Start the installation program and choose the custom installation. You need to select just the core files and the start-menu shortcuts, other features are not required for the communication.
- Open the installed utility "Tibbo Connection Wizard" and proceed with installing the virtual port. Choose the desired COM number for the "Virtual Serial Port" and select the "Device Server" from the list. Let other settings unchanged and finish the installation. Note that the device that you wish to connect to must be accessible during installing the virtual port. The port driver is not digitally signed, you have to allow its installation despite any warnings.
- The installation is complete, you can connect the device via the selected COM port.
- Note that the COM number of a previously installed virtual port cannot be changed. You can, however, select the attached device to any time. To do so, open the "Tibbo Connection Wizard" and repeat the abovementioned setup procedure. The settings are applied immediately, you do not need to reboot the PC to activate them. You do not need administrative rights to change the attached device, the administrative rights are required just to create new virtual ports.

Note that you may need to install several virtual ports if you like to connect several devices simultaneously. If you install just one virtual port but wish to communicate with more devices, you will need to se-

lect the proper device server by executing the "Tibbo Connection Wizard" before starting the communication.

Note that the port assignment done by the "Tibbo Connection Wizard" is user-specific. The device that you select as computer administrator is valid for all accounts until you change it when logged on as another user. It implies also that changes done for a particular user account are not visible for other users.

Software Interface

The software interface for the device consists of a 32-bit dynamic link library `COM-RFG.dll`. It is located in the directory "Program" of the enclosed software package. The software interface is a stand-alone software package, it does not require any additional library or driver, except the virtual port drivers (see section "Driver Installation").

The user functions in the dynamic link library `COM-RFG.dll` can be called from any popular programming language. For the details, please consult the user manual of your compiler. The definition of the interface functions is located in the C-header file `COM-RFG.h`. If your compiler cannot create an import library from the dynamic link library `COM-RFG.dll`, please link the library `COM-RFG.lib` instead of the dynamic link library to your project.

To simplify the integration in Pascal software packages, all functions of the software interface use the Pascal calling convention.

The sample program at the end of this section is written for the Borland C++ compiler. However, you should also be able to compile it with other C++ compilers without needing to modify it.

Functionality of the Software Interface

The software interface controls up to eight communication channels for the data transfer from/to different devices. Each communication channel must be opened before starting the communication. The open procedure assigns the selected communication channel to a virtual serial port. Next, the virtual serial port is configured and the port buffers are cleared. The used communication channel should be closed at the program end. If this fails to happen, the software interface does it automatically for you when the dynamic link library `COM-RFG.dll` is unloaded from the system memory.

Most of the interface functions require as a first parameter the variable `PortNumber` that is the number of the communication channel, which should be used for the particular operation. This number is an unsigned integer that must be lower than `COM_RFG_MAX_PORT`, i.e. must be in the range from 0 to 7.

The return value of most interface functions is a signed 32-bit number (`int`) describing the success of the particular operation. The last return value can be reloaded by the function `COM_RFG_State`. Table 2 summarizes the possible return values together with the error messages, which can also be obtained by the function `COM_RFG_ErrorMessage`. The error codes in Tab. 2 correspond to the codes `COM_RFG_ERR_xxx` in the C-header file `COM-RFG.h`. If a failure in data transfer has occurred, you can discover the reason by calling the functions `COM_RFG_IO_State` and `COM_RFG_IO_ErrorMessage`. The first one returns the last I/O error, the second one the corresponding error message (see Tab. 3.).

If you have troubles understanding the errors or establishing the communication, please contact the manufacturer of the device.

Tab. 2. Return values of the interface functions

Return value	Error message	Description
0	No error	The data transfer finished successfully.
-2	Error opening the port	The port could not be opened. For the possible reasons, see Tab. 3.
-3	Error closing the port	The port could not be closed. For the possible reasons, see Tab. 3.
-4	Error purging the port	The port buffers could not be cleared.
-5	Error setting the port control lines	The port control lines could not be set.
-6	Error reading the port status lines	The port status lines could not be read.
-7	Error sending command	The data transfer to the device failed. For the possible reasons, see Tab. 3.
-8	Error sending data	
-9	Error sending termination character	
-10	Error receiving command	The data transfer from the device failed. For the possible reasons, see Tab. 3. Check also the used data rate.
-11	Error receiving data	
-12	Error receiving termination character	
-13	Wrong command received	The device sent an unexpected response.
-14	Wrong argument received	
-15	Wrong argument passed to the function	One of the arguments passed to the function was out of the allowable range.

Return value	Error message	Description
-100	Device not connected	The port status lines indicate that the device is not connected.
-101	Device not ready	The port status lines indicate that the device is not ready. This may be caused by a running back-ground process. Repeat the operation to resolve the problem.
-102	Device state could not be set to not ready	The device did not react properly. Try to reset the communication or restart the device by powering it off and on.
-200	PortNumber out of range	The <code>PortNumber</code> must be lower than <code>COM_RFG_MAX_PORT</code> .
-300	Memory for the port data could not be allocated	The port initialization failed and the requested amount of memory could not be allocated.
-400	Error opening the file for debugging output	The text file for debugging output could not be opened.
-401	Error closing the file for debugging output	The file for debugging output could not be closed.

Tab. 3. I/O errors

Return value	Error message	Description
0	No error	The data transfer finished successfully.
1	Port has not been opened yet	You attempted to use the communication channel before having opened it.
2	Cannot open the port	The specified port could not be opened. Either the port does not exist or it is being currently used by another program.
3	Cannot get the state of the port	The system could not get the state of the port.
4	Cannot set the state of the port	The system could not set the state of the port.
5	Cannot set the timeouts for the port	The system could not set the timeouts for the port.
6	Cannot clear the port	The system could not clear the port buffers.
7	Error reading data from the port	The system could not read data from the port. Most probably, no data is available because the device is either disconnected or does not respond.
8	Error writing data to the port	The system could not write data to the port.
9	Wrong data amount written to the port	The system could not write the proper data amount to the port.
10	Error setting the control lines of the port	The system could not set the state of the port control lines.
11	Error reading the status lines of the port	The system could not get the state of the port status lines.
12	Device is busy	The device is not ready to communicate.

Communication Control

Function COM_RFG_SetDataRate

```
void pascal COM_RFG_SetDataRate  
(WORD PortNumber, int DataRate);
```

Sets the data rate for opening the communication channel.

The parameter `DataRate` should be one of the constants `COM_RFG_RATE_xxx`. The constant `COM_RFG_RATE_AUTO` equals to the default setting and ensures automatic detection of the data rate. The constants `COM_RFG_RATE_USB` and `COM_RFG_RATE_LAN` switch the data rate to the value required to communicate via the USB or the LAN interface, respectively.

You must call the function `COM_RFG_SetDataRate` prior to opening the communication channel by the function `COM_RFG_Open`, otherwise the function call does not have any effect to the already established communication.

In the practice, you do not need to call the function `COM_RFG_SetDataRate` at all, the software interface will use the default settings `COM_RFG_RATE_AUTO` and detect the data rate automatically. However, if you specify the data rate by calling the function `COM_RFG_SetDataRate` explicitly, the opening procedure will speed up.

Note that the software interface changes the default settings during the opening procedure to the finally selected value that can be acquired by the function `COM_RFG_GetDataRate`. If you like to reopen the communication channel and attach it to another port with a different data rate, you have to reset the settings to the default value `COM_RFG_RATE_AUTO`, otherwise the opening procedure will fail.

Function COM_RFG_GetDataRate

```
void pascal COM_RFG_GetDataRate  
(WORD PortNumber, int & DataRate);
```

Gets the data rate used to open the communication channel.

The return value in the variable `DataRate` is one of the constants `COM_RFG_RATE_xxx`.

You can call this function prior to opening the port by the function `COM_RFG_Open` to obtain the settings that will be used during the next opening procedure. When you call this function after the opening procedure has finished, the return value reflects the data rate selected for the communication. In this case, the values can be the constants `COM_RFG_RATE_USB` or `COM_RFG_RATE_LAN` only, corresponding to the communication via the USB or the LAN interface, respectively.

Function `COM_RFG_Open`

```
int pascal COM_RFG_Open (WORD PortNumber,  
BYTE COMNumber);
```

Opens the specified communication channel and returns an error code according to Tab. 2.

The function opens the channel with the number `PortNumber` and attaches it to the serial port with the number `COMNumber`. The variable `PortNumber` must be lower than the number of implemented channels `COM_RFG_MAX_PORT`. The serial port number `COMNumber` must point to a valid serial port to that the controller is attached. Note that this number is the number of the COM port, i.e. `COMNumber = 1` points to the port COM1. The function accepts all numbers that are supported by the operating system, thus any port between COM1 and COM255 can be used for the communication.

You must call this function prior to any other communication function. If the function returns an error, no data communication is possible.

You may select the data rate for the communication by calling the function `COM_RFG_SetDataRate` before opening the communication channel. The data rate selected by the function `COM_RFG_Open` can be obtained by calling the function `COM_RFG_GetDataRate`.

Note that an incorrectly selected data rate leads to communication errors like `COM_RFG_ERR_COMMAND_RECEIVE` (see Tab. 2). If the function `COM_RFG_Open` returns such errors, check the data rate selected prior to calling it.

Function COM_RFG_Close

```
int pascal COM_RFG_Close (WORD PortNumber);
```

Closes the specified communication channel and returns an error code according to Tab. 2.

You can use the function to free the port for another application.

If an application that has exclusively used the software interface COM-RFG.dll finishes, the opened communication channel closes automatically. Thus, the programmer does not need to call the function COM_RFG_Close explicitly.

Function COM_RFG_Purge

```
int pascal COM_RFG_Purge (WORD PortNumber);
```

Clears the port data buffers and returns an error code according to Tab. 2.

The function can be used to repair a disturbed communication. In case of a crash of a user program, this function should be called to erase data incorrectly received from the device.

The function COM_RFG_Purge is automatically called by the function COM_RFG_Open.

Function COM_RFG_Buffer_State

```
int pascal COM_RFG_Buffer_State  
(WORD PortNumber, BOOL & Empty);
```

Gets the state of the device's input data buffer in the variable `Empty`. The return value is an error code according to Tab. 2.

When a large data amount should be transferred to the device, this function can be used to ensure that the input data buffer contains enough free space. If the return value of the variable `Empty` is false, the input buffer is not empty and there is no guarantee that the device will be able to receive the data. This situation can occur if the device has just received a large amount of data and its processing has not yet finished. In such a case, the call to the function

`COM_RFG_Buffer_State` should be repeated after several milliseconds until the return value becomes `true`.

Function `COM_RFG_Device_Purge`

```
int pascal COM_RFG_Device_Purge  
(WORD PortNumber, BOOL & Empty);
```

Clears the device's output data buffer and gets the state of the device's input data buffer in the variable `Empty` like the function `COM_RFG_Buffer_State`. The return value is an error code according to Tab. 2.

The function can be used to repair a disturbed communication. If the device does not respond properly, the function `COM_RFG_Device_Purge` should be called repeatedly until it returns the value `true` in the variable `Empty`.

RF-Generator Control

Function COM_RFG_Get_Amplitude

```
int pascal COM_RFG_Get_Amplitude  
(WORD PortNumber, double & Amplitude);
```

Gets the output amplitude in the variable `Amplitude`, and returns an error code according to Tab. 2.

The return value in the variable `Amplitude` is the set output amplitude in Volts.

For more details, see section "Generator Control".

Function COM_RFG_Set_Amplitude

```
int pascal COM_RFG_Set_Amplitude  
(WORD PortNumber, double & Amplitude);
```

Sets the output amplitude according to the variable `Amplitude`, gets the set value back in the variable `Amplitude`, and returns an error code according to Tab. 2.

The return value in the variable `Amplitude` is the set output amplitude in Volts.

After calling the function `COM_RFG_Set_Amplitude`, you should check the return value in the variable `Amplitude` to obtain the real value set by the device. If you, for instance, try to set an output amplitude outside the specified range, the device truncates the value and the variable `Amplitude` returns the modified value.

To prevent the truncation, obtain the allowed amplitude range first by calling the function `COM_RFG_GetHardwareCapability`.

For more details, see sections "Generator Control" and "Hardware Information".

Function COM_RFG_Get_Enable

```
int pascal COM_RFG_Get_Enable  
    (WORD PortNumber, BOOL & Enabled);
```

Gets the state of the device in the variable `Enabled` and returns an error code according to Tab. 2.

For more details, see section "Generator Control".

Function COM_RFG_Set_Enable

```
int pascal COM_RFG_Set_Enable  
    (WORD PortNumber, BOOL & Enabled);
```

Sets the state of the device to the variable `Enabled`, gets the set value back in the variable `Enabled`, and returns an error code according to Tab. 2.

After calling the function `COM_RFG_Set_Enable`, you should check the return value in the variable `Enabled` to obtain the real state of the device. If you, for instance, try to enable a device with opened interlock loop (see the function `COM_RFG_Get_InterlockState`), the enable procedure fails and the variable `Enabled` returns `FALSE`.

For more details, see section "Generator Control".

Monitoring

Function COM_RFG_Get_HousekeepingData

```
int pascal COM_RFG_Get_HousekeepingData
    (WORD PortNumber, double & Voltage12V,
     double & Voltage5V, double & Voltage3V,
     double & Temperature);
```

Gets the housekeeping data and returns an error code according to Tab. 2.

The return values in the variables `Voltage12V`, `Voltage5V`, and `Voltage3V` are supply voltages in Volts. The return value in the variable `Temperature` is the microcontroller temperature in °C.

For more details, see section "Housekeeping Data".

Function COM_RFG_Get_RFGData

```
int pascal COM_RFG_Get_RFGData
    (WORD PortNumber,
     double & Amplitude1, double & Amplitude2,
     double & VoltageDrain, long & Frequency,
     double & TemperatureHeatsink,
     double & TemperatureCoil);
```

Gets the operating parameters of the RF generator and returns an error code according to Tab. 2.

The return values in the variables `Amplitude1`, `Amplitude2` and `VoltageDrain` are the output amplitudes and the drain voltage of the generator's end stage in Volts. The return value in the variable `Frequency` is the output frequency in Hertz. The return value in the variables `TemperatureHeatsink` and `TemperatureCoil` are the heatsink and air temperatures in °C.

For more details, see section "RFG Monitor".

Function COM_RFG_Get_PowerData

```
int pascal COM_RFG_Get_PowerData  
(WORD PortNumber, double & Voltage,  
double & Current, double & Temperature);
```

Gets the power supply data and returns an error code according to Tab. 2.

The return values in the variables `Voltage` and `Current` are the supply voltage and current of the generator's end stage in Volts and Amperes, respectively. The return value in the variable `Temperature` is the heatsink temperature in °C.

For more details, see section "Power Monitor".

Function COM_RFG_Get_InterlockState

```
int pascal COM_RFG_Get_InterlockState  
(WORD PortNumber, BOOL & Closed);
```

Gets the state of the interlock loop in the variable `Closed` and returns an error code according to Tab. 2.

Note that the device can be enabled only if the interlock loop is closed, i.e. if the variable `Closed` returns `TRUE` (see also the function `COM_RFG_Set_Enable`).

Error Handling

Function COM_RFG_State

```
int pascal COM_RFG_State (WORD PortNumber);
```

Returns the state of the software interface according to Tab. 2.

This function can be used to obtain the last return value of an interface function.

The function does not have any influence on the communication and can be called at any time.

Function COM_RFG_ErrorMessage

```
const char * pascal COM_RFG_ErrorMessage  
(WORD PortNumber);
```

Returns the error message corresponding to the state of the software interface. The return value is a pointer to a null-terminated character string according to Tab. 2.

The function does not have any influence on the communication and can be called at any time.

Function COM_RFG_IO_State

```
int pascal COM_RFG_IO_State  
(WORD PortNumber);
```

Returns the interface state of the serial port according to Tab. 3.

The function can be used to obtain the result of the last I/O operation at the port.

The function does not have any influence on the communication and can be called at any time.

Function COM_RFG_IO_ErrorMessage

```
const char * pascal COM_RFG_IO_ErrorMessage  
    (WORD PortNumber);
```

Returns the error message corresponding to the interface state of the serial port. The return value is a pointer to a null-terminated character string according to Tab. 3.

The function does not have any influence on the communication and can be called at any time.

Function COM_RFG_OpenDebugFile

```
int pascal COM_RFG_OpenDebugFile  
    (WORD PortNumber, const char * FileName);
```

Opens a text file for debugging output and returns an error code according to Tab. 2.

You can use the debugging output in case of any communication problems. If a communication error occurs, a detailed error analysis is saved in the specified text file.

The function does not have any direct influence on the communication and can be called at any time.

Function COM_RFG_CloseDebugFile

```
int pascal COM_RFG_CloseDebugFile  
    (WORD PortNumber);
```

Closes the file for debugging output and returns an error code according to Tab. 2.

The function does not have any direct influence on the communication and can be called at any time.

Various Functions

Function COM_RFG_Version

```
WORD pascal COM_RFG_Version();
```

Returns the version of the software interface (the dynamic link library COM-RFG.dll).

The function should be used to check whether a software interface with the correct version is being used. The function should be called prior to any other function of the software interface.

The return value is an unsigned 16-bit integer (WORD). The higher byte contains the main version number, the lower byte the subversion, i.e. the version order within the main version. The identical main version number of two different libraries COM-RFG.dll means that they implement the same functions. The difference in the subversion denotes that various corrections have been made to the interface functions. If the main version number is different, there is no guarantee that the library can be used. Mostly, the user software has to be re-compiled or modified to match the new definition of the software interface.

Function COM_RFG_GetHardwareCapability

```
int pascal COM_RFG_GetHardwareCapability  
  (WORD PortNumber, double & MinAmplitude,  
   double & MaxAmplitude,  
   double & MaxPSUVoltage);
```

Gets the device hardware capabilities and returns an error code according to Tab. 2. The return values are the parameters that can be viewed by the dialog box "Hardware Information" (see section "Hardware Information").

The return values in the variables `MinAmplitude` and `MaxAmplitude` are the output amplitude limits in Volts. If you try to set the output amplitude to a value outside these limits, the controller will adjust it (see also function `COM_RFG_Set_Amplitude`).

The return value in the variable `MaxPSUVoltage` is the maximum allowable supply voltage of the generator's end stage in Volts.

Function COM_RFG_GetUptime

```
int pascal COM_RFG_GetUptime  
    (WORD PortNumber, DWORD & Seconds,  
    WORD & Milliseconds);
```

Gets the device uptime in the variables `Seconds` and `Milliseconds`. The function return value is an error code according to Tab. 2.

The device uptime is the time elapsed from the last (re)start of the device. The function can be used, for instance, to see whether the device restarted unexpectedly.

Function COM_RFG_FW_Ver

```
int pascal COM_RFG_FW_Ver (WORD PortNumber,  
    WORD & Version);
```

Gets the device's firmware version in the variable `Version`. The function return value is an error code according to Tab. 2.

The return value in the variable `Version` should be used to check whether the software interface is in an appropriate version.

The return value is similar to the return value of the function `COM_RFG_Version`. It is an unsigned 16-bit integer (WORD) containing the main version and the subversion numbers. The software interface is compatible to the firmware if the main version numbers are identical.

Function COM_RFG_FW_Date

```
int pascal COM_RFG_FW_Date (WORD PortNumber,  
    char * DateString);
```

Gets the device's firmware date in the variable `DateString`. The function return value is an error code according to Tab. 2.

The return value in the variable `DateString` is a null-terminated character string with the firmware compilation date. The buffer passed to the function must be created before the function call, it must be at least 16 bytes long.

Function COM_RFG_Prod_No

```
int pascal COM_RFG_Prod_No (WORD PortNumber,  
    DWORD & Number);
```

Gets the device's product number in the variable `Number`. The function return value is an error code according to Tab. 2.

Function COM_RFG_Prod_ID

```
int pascal COM_RFG_Prod_ID (WORD PortNumber,  
    char * Identification);
```

Gets the device's product identification in the variable `Identification`. The function return value is an error code according to Tab. 2.

The return value in the variable `Identification` is a null-terminated character string with the product identification. The buffer passed to the function must be created before the function call; it should be 81 bytes long.

Sample Communication Program

```

#include <stdio.h>
#include <windows.h>
#include "COM-RFG.h"

// handle a possible error:
static int Error (int code)
{
    printf ("%s\n", COM_RFG_ErrorMessage (0));
    return code;
}

int main (int argc, char * argv[])
{
    if (argc != 2)
    {
        printf ("The number of the COM port must be specified!\n");
        return -1;
    }
    // check the DLL version:
    WORD DLL_Version = COM_RFG_Version();
    if ((DLL_Version & 0xFF00) != 0x0100)
    {
        printf ("Wrong DLL Version: expected 1.x, found %d.%02d\n",
            DLL_Version / 0x100, DLL_Version & 0xFF);
        return 1;
    }

    // open the port 0:
    WORD COMNumber;
    sscanf (argv[1], "%hu", &COMNumber);
    if (COM_RFG_Open (0, COMNumber))
        return Error (2); // handle a possible error

    // Get interlock state:
    BOOL Closed;
    if (COM_RFG_Get_InterlockState (0, Closed))
        return Error (3); // handle a possible error
    printf ("Interlock: %s\n", Closed ? "closed" : "open");

    // Get RFG data:
    double Amplitude1, Amplitude2, VoltageDrain;
    long Frequency;
    double TemperatureHeatsink, TemperatureCoil;
    if (COM_RFG_Get_RFGData (0, Amplitude1, Amplitude2,
        VoltageDrain, Frequency,
        TemperatureHeatsink, TemperatureCoil))
        return Error (4); // handle a possible error
    printf ("Measured RF output values:\n"
        "\tAmplitudes: %.11fV, %.11fV\n"
        "\tFrequency: %.4fMHz\n",
        Amplitude1, Amplitude2, 1E-6*Frequency);

```

```
// Get power supply data:
double Voltage, Current, Temperature;
if (COM_RFG_Get_PowerData (0, Voltage, Current, Temperature))
    return Error (5); // handle a possible error
printf ("Measured supply values:\n"
        "\tVoltage: %5.2lfV\n"
        "\tCurrent: %5.2lfA\n"
        "\tPower:   %5.1lfW\n",
        Voltage, Current, Voltage*Current);

//finish the program:
return 0;
}
```