

High-Voltage Amplifier with Programmable Digital Waveform Generator for Driving Piezo Actuators

Version 1.00



User Manual

Document version 1, created on May-21-2010

Contents

Safety Information.....	8
Technical Data.....	9
Characteristics.....	9
Amplifier.....	9
Security.....	9
Monitoring.....	9
Waveform Generator.....	10
Human Interface.....	10
Digital Interface.....	10
Power Supply.....	10
General.....	11
Shipment Contents.....	11
Description.....	12
General.....	12
Control and Indication Elements.....	14
Terminals.....	15
Installation.....	18
Programmable Digital Waveform Generator.....	19
General.....	19
Quick Setup Guide.....	21
Up- and Downloading Waveforms.....	22
Tips.....	24
Human Interface.....	27
Main Program Menu.....	28
Device Management.....	30
Waveform Management.....	30
Measurement Management.....	35
System Preferences.....	41
Measurement Control.....	44

Device Monitoring.....	49
Information	49
Statistics	49
Clock.....	50
Voltage Monitor	51
Temperature Monitor.....	52
Fan Monitor	53
Software Utilities.....	55
Utility FlashLoader.....	55
Utility HardCopy	57
Driver Installation.....	58
Installation of the Virtual Port for the USB Interface	58
Software Interface.....	59
Functionality of the Software Interface.....	59
Communication Control	64
Function COM_DAFG_Open	64
Function COM_DAFG_Close.....	64
Function COM_DAFG_Purge	64
Function COM_DAFG_Buffer_State	65
Function COM_DAFG_Device_Purge	65
Waveform Management.....	66
Function COM_DAFG_DeleteWaveform	66
Function COM_DAFG_CreateWaveform.....	66
Function COM_DAFG_GetWaveform.....	68
Function COM_DAFG_SetWaveformData.....	68
Function COM_DAFG_GetWaveformData	69
Error Handling	71
Function COM_DAFG_State.....	71
Function COM_DAFG_ErrorMessage	71
Function COM_DAFG_IO_State	71
Function COM_DAFG_IO_ErrorMessage.....	71
Function COM_DAFG_INI_ErrorMessage.....	72
Various Functions.....	73
Function COM_DAFG_Version.....	73

Function COM_DAFG_GetIniParams	73
Function COM_DAFG_GetUptime	73
Function COM_DAFG_FW_Ver	74
Function COM_DAFG_FW_Date	74
Function COM_DAFG_Prod_No	74
Function COM_DAFG_Prod_ID	74
Sample Communication Program	76

Figure List

Fig. 1. The output connector.	15
Fig. 2. The front panel.	17
Fig. 3. Program display in device idle state.....	28
Fig. 4. Main program menu.	29
Fig. 5. The dialog box "Waveform".	30
Fig. 6. The menu in the dialog box "Waveform".	30
Fig. 7. The waveform name editor in the dialog box "Waveform".	31
Fig. 8. The dialog box "Waveform Detail".	31
Fig. 9. The dialog box "Waveform Zoom".	32
Fig. 10. The menu in the dialog box "Waveform Zoom".	33
Fig. 11. The dialog box "Measurement".	35
Fig. 12. The menu in the dialog box "Measurement".	35
Fig. 13. The measurement name editor in the dialog box "Measurement".	36
Fig. 14. The menu in the dialog box "Waveform" for selecting waveforms.	36
Fig. 15. Setting the waveform duration in the dialog box "Measurement".	37
Fig. 16. The dialog box "Timing Setup".	37
Fig. 17. Setting the waveform repeating in the dialog box "Measurement".	38
Fig. 18. Setting the number of waveform repetitions in the dialog box "Measurement".	38
Fig. 19. Setting the trigger slope in the dialog box "Measurement".	39
Fig. 20. Setting the amplifier activation in the dialog box "Measurement".	39
Fig. 21. Setting the output signal amplitude in the dialog box "Measurement".	40
Fig. 22. Dialog box "System Preferences".	41

Fig. 23. Setting the waveform duration in the dialog box "System Preferences"	41
Fig. 24. Setting the trigger slope in the dialog box "System Preferences"	42
Fig. 25. Setting the amplifier activation in the dialog box "System Preferences"	42
Fig. 26. Setting the allowable actor temperatures in the dialog box "System Preferences"	43
Fig. 27. The dialog box "Measurement Control"	44
Fig. 28. Start menu of the dialog box "Measurement Control"	45
Fig. 29. The menu in the dialog box "Measurement" for selecting measurements.	45
Fig. 30. The dialog box "Measurement Control" when the device is waiting for the trigger.	45
Fig. 31. The dialog box "Measurement Control" when the device is generating a waveform.	47
Fig. 32. Dialog box "Information"	49
Fig. 33. Dialog box "Statistics"	50
Fig. 34. Dialog box "Clock"	50
Fig. 35. Dialog box "Voltage Monitor"	51
Fig. 36. Dialog box "Temperature Monitor"	52
Fig. 37. Dialog box "Fan Monitor"	53

Table List

Tab. 1. Pin layout of the output connector.	15
Tab. 2. Command line parameters of the program WaveLoader.....	25
Tab. 3. Possible states of the waveform generator.....	48
Tab. 4. The LCD character set.....	54
Tab. 5. Return values of the interface functions.....	61
Tab. 6. I/O errors	63
Tab. 7. Error codes of the function COM_DAFG_CreateWaveform.	67

Safety Information

- The device may be installed and used by authorized and instructed personnel only. Read this manual carefully before installing and using the device. Always follow the safety notes and warnings in this manual.
- The device is designed for indoor dry laboratory use only. Before powering the device on, the device temperature must accommodate to the ambient temperature to avoid moisture condensation. Take this into account when setting up the device after transporting it.
- Do not operate the device if it is damaged or not functioning properly. Never use damaged cables or accessories.
- Do not open the device case, install replacement parts, or perform modifications to the device. There are no user serviceable parts inside.
- To avoid damage, connect the line cord to a properly wired and grounded receptacle only. Be sure that the mains voltage and the fuse rating match the device specification. Never operate the device during electrical storms.
- Never use corrosive or abrasive cleaning agents or polishes, avoid the usage of organic solvents. If necessary, clean the device with a soft moist cloth. Make sure that the device is completely dry and free from contaminants before powering it on.

! Warning: The device produces lethal voltages. Never touch the output terminals even if the device is powered off. The device is equipped with several protection features that eliminate the risk of contact with a high voltage. This protection, however, may fail in case of a fault and unexpected high voltages may be present at the terminals.

! Warning: The amplifier is an electronic device that is sensitive to electrostatic electricity. While manipulating with the amplifier, the ESD (*ElectroStatic Discharge*) protection rules must be kept in mind.

! Warning: If the LED "Amplifier Overheated" lights, avoid touching the heatsink of the device.

! Warning: If the LED "Actor Overheated" lights, avoid touching the actor.

! Warning: The device weighs about 40 kg. Use an appropriate carrier when transporting, and provide a stable support for its operation.

Technical Data

Characteristics

- high-voltage amplifier for heavy capacitive loads
- programmable waveform generator with USB interface
- 19" case

Amplifier

- output connector:
LEMO ERA.4S.737.CLL (panel socket with 2 high-voltage pins)
- output voltage: 0..1100 V
(2 symmetrical outputs with 0..+550 V and 0..-550 V)
- switchable offset: 550 V (± 275 V)
- output current: 1 A continuous, 2 A typ. pulsed
- signal bandwidth: DC..5 kHz
- power bandwidth:
1 kHz without load; with load, limited by the output current
(about 30 Hz with a load of 10 μ F)
- output noise: 30 mV_{pp} without load, 10 mV_{pp} with a load of 10 μ F

Security

- interlock loop:
two pins at the output connector,
short circuit enables the amplifier
- overcurrent protection: output current limiter
- overtemperature protection:
temperature sensors at the heatsink
temperature sensor PT1000 at the actor,
(shutdown and restart temperatures freely programmable)

Monitoring

- monitor outputs:
two 50 Ω BNC sockets
attenuation: 1:100
internal impedance: 10 k Ω parallel with 100 pF
- LEDs:
Power, Amplifier Activated, Interlock, Fan Failure
Amplifier Overheated, Actor Overheated
- supply voltage monitoring

Waveform Generator

- resolution: 16 bit
- data rate: 254 S/s..16.7 MS/s
- total memory size: 512 KW (= 512 KWords = 1 MB)
- memory space for waveform data: 508 KS (= 520192 samples)
- waveform size: variable from 2 S to 64 KS (= 65536 samples)
- maximum waveform number: 100
- maximum measurement number: 100
- trigger: manual/external
- trigger input:
 - 50 Ω BNC socket, level: TTL/CMOS/LV-CMOS,
 - galvanically isolated from the device to eliminate ground loops

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"
- encoder: 24 positions per revolution, integrated press button
- press button for amplifier activation
- optional external shutdown button via the interlock loop

Digital Interface

- USB interface according to USB 2.0 standard
 - connector: USB plug type B
 - data transfer rate: up to 12 MBit/s (*Full Speed*)
 - effective data transfer rate: >100 kBit/s

Power Supply

- rated voltage: 230 V \pm 10%, 50 Hz
- inrush current: limited to 8 A
- power consumption:
 - standby (amplifier deactivated): 30 VA
 - amplifier activated but without signal: 150 VA
 - maximum: 1 kVA
 - peak: 1.6 kVA
- main fuse: T 8 A (slow acting), size \varnothing 5x20 mm

- mains connection:
IEC inlet with EMC filter and integrated fuse holder

General

- 6 U 19" bench-top case:
height: 266 mm, width: 448 mm, depth: 480 mm,
total depth with heatsink, fans, and handles: 650 mm,
total height with case feet: 280 mm,
side panels: black lacquered,
cover and bottom panels: silver gray painted,
front and rear panels: clear anodized
- heatsink:
black anodized extruded aluminum profile at the rear panel
- forced cooling:
four 120 mm fans at the heatsink,
temperature-dependent regulation of the fan speed,
monitored fan operation, warning on fan failures
- weight: about 40 kg
- cleaning: use moist cloth only, avoid use of organic solvents

Shipment Contents

- high-voltage amplifier HV-PA550-2D
- line cord (universal IEC mains lead, length 1.8 m)
- control and diagnostic software
- user manual
- optional:
 - output cable
(1 or 2 LEMO plugs FFB.4S.737, length up to about 10 m)
 - filter unit
(high frequency filter for the output cable)
 - USB cable
(A plug to B plug, length 1, 1.8, 3, or 5 m)
 - USB-2.0 interface card

Description

General

The device is a combination of a high-voltage power amplifier and a programmable digitally controlled waveform generator. It is fit into a metallic 19" case, the front panel is shown in Figure 2.

The amplifier is able to provide an output power of about 1 kVA to loads with a significant capacitance. The amplifier stability has been tested with capacitive loads of several 10 μF . The output terminals produce voltages symmetric with respect to ground. The output voltage at each terminal ranges from 0 to +550 V for the positive terminal and 0 to -550 V for the negative terminal, i.e. the total output voltage is unipolar and may be as high as 1.1 kV.

The typical output load of the device is a piezo actor, the amplifier has been optimized to drive the actors P-035.90P produced by PI Ceramic GmbH. For operation with the actors, the amplifier produces an offset voltage that equals half of the maximum amplitude. The offset voltage corresponds to a zero signal of the waveform generator. The total amplifier's output voltage decays to zero or rises to 1.1 kV if the output of the waveform generator reaches the negative or positive maximum, respectively.

The waveform generator can produce preferably harmonic signals or a combination of them in a frequency region of up to several kHz. The signal output can be synchronized with external instruments via the trigger input. The digital trigger signal at the trigger input starts the signal generation. The device requires a signal with TTL, 5V-CMOS, or 3V-CMOS levels, the trigger slope is selectable. To reduce the risk of ground loops, the trigger input is galvanically isolated from the device.

The device is equipped with an interlock loop that disables and shorts the output if it is opened. The terminals of the interlock loop are located at the output connector. They are typically fed by the output cable to the actor where they are either short connected or wired to a security push button or switch with a breaking contact. The output is disabled if the output cable is disconnected or broken, or if the security push button or switch is activated.

The device is also equipped with a circuitry that scans the temperature measured by an external Pt1000 sensor. This sensor is typically

located at the actor, the amplifier output is disabled if the sensor temperature exceeds a preset value. This ensures a secure operation of the actor and prevents its overloading.

The amplifier and its power supplies are cooled by a large heatsink located at the rear panel of the device. The heatsink is actively cooled by four fans. The fans are controlled by the temperature of the power semiconductor devices. Thus, in the standby state with disabled output, the device produces only minimal acoustic noise, since the rotational speed of the fans is rather low. With heavy output loads, the heatsink temperature increases and the fans are controlled to rotate faster. Under certain conditions, the cooling power of the device is not sufficient and the power semiconductor devices could overheat. This may happen if the output is overloaded or even shorted, if the ambient temperature is high, or if one of the fans fails. To prevent this from happening, the amplifier is deactivated if the temperatures exceed a preset level.

To check the output signal of the amplifier, two monitor outputs are available. They provide a 100-times attenuated voltage of the corresponding output terminal. The internal impedance of the monitor outputs is about 10 k Ω . You can connect it to an oscilloscope with an input impedance of at least 1 M Ω . You should use short cables only, since the cable capacitance influences the frequency response of the attenuator. With a typical 50 Ω cable of a length of 2 m, the capacitance is about 200 pF. Thus, the circuitry behaves like a low pass filter with a corner frequency of $1/(2\pi \cdot 10 \text{ k}\Omega \cdot 200 \text{ pF}) \sim 80 \text{ kHz}$.

The waveform generator is equipped with a USB data interface that allows you to transfer waveform data to or from the device. The USB data interface is galvanically connected to the device case. When making a connection to a host computer, a large ground loop can be created that can influence the device performance. If the host computer should be permanently connected to the device, carefully check the influence of the USB connection on the output signal.

The amplifier and its power supplies are linearly operating circuits. Since the device does not use any power switching circuits such as switching mode power supplies or class D amplifiers, the emission level of electromagnetic radiation is rather low. The power supplies are equipped with toroidal transformers; these produce only a weak stray magnetic field. Due to the low electromagnetic radiation, the de-

vice should not disturb even sensitive instruments located in the proximity of the device.

Control and Indication Elements

All control elements of the device are located on the front panel (see Fig. 2).

Using the rocker switch "Power", you can power the device on or off. When powered on, the LED above the power switch lights green and the display turns on.

The "On/Off" push button activates or deactivates the amplifier. When deactivated, the power supply of the amplifier is turned off and each output terminal is shorted by 500 Ω resistors to ground. When activated, the power supply is turned on, the LED "Amplifier Activated" lights green.

The LED "Interlock" shows the state of the interlock loop. It lights yellow, if the interlock loop is closed. When the loop is opened, the LED "Interlock" is off and the amplifier cannot be activated.

The LED "Fan Failure" lights red if one of the fans fails. The device continues to operate, you should, however, replace the fan as soon as possible. You can locate the defective fan by a dialog box (see Fig. 37 and the corresponding text).

The LED "Amplifier Overheated" lights red if one of the power semiconductor devices of the amplifier or of the power supply units reaches the critical temperature. When the LED lights, the amplifier cannot be activated. The temperature threshold is internally set, you cannot change it. You can monitor the current temperatures by a dialog box (see Fig. 36 and the corresponding text). In case of overheating, you can locate the device part that caused the temperature rise in a dialog box (see Fig. 37 and the corresponding text).

The LED "Actor Overheated" lights red if the temperature sensor at the actor reaches the critical temperature. When the LED lights, the amplifier cannot be activated. The temperature threshold can be modified by a dialog box (see Fig. 26 and the corresponding text). You can monitor the current actor temperature by a dialog box (see Fig. 36 and the corresponding text).

Terminals

The output terminals are located in the connector "Output". The connector is a robust LEMO socket with two high-voltage pins (for the pin layout, see Fig. 1 and Tab. 1). The amplifier output is connected to the two high-voltage pins. The middle pins are connected to the protection ground of the device, they can be used to ground the actor. One pin pair of the output connector is used for the interlock loop, they must be shorted in order to enable the amplifier. The next pin pair is used for the connection of the actor temperature sensor. The connection should be made with a coaxial cable.

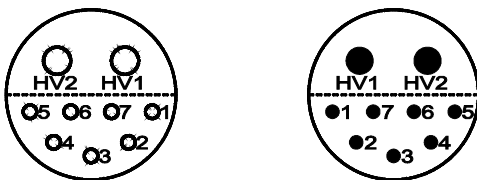


Fig. 1. The output connector.
 Left the panel socket (female), right the cable plug (male).

Tab. 1. Pin layout of the output connector.

Pin	Signal	Description
HV1	OUT-POS	Amplifier Output - Positive Terminal
HV2	OUT-NEG	Amplifier Output - Negative Terminal
1	ACT-TEMP-GND	Actor Temperature Sensor - Ground
2	ACT-TEMP-SIG	Actor Temperature Sensor - Signal
3	PE	Protective Ground
4	INTERLOCK-POS	Interlock Loop - Positive Terminal
5	INTERLOCK-NEG	Interlock Loop - Negative Terminal
6	PE	Protective Ground
7	PE	Protective Ground

The output voltage is attenuated by a passive attenuator and is connected to the BNC sockets "Output Monitor". The connector labeling corresponds to the output terminals.

The digital trigger signal can be connected to the isolated BNC socket "Trigger Input". The trigger input is galvanically isolated from the device.

The socket "USB" is an USB 2.0 compatible data interface. Connect it using a standard USB cable to a host computer.

For the mains connection, an IEC inlet with an EMC filter and integrated fuse holder is located on the rear panel. Use a standard line cord with the universal IEC mains lead to connect the device to the mains. To replace the fuse, remove the line cord and open the fuse holder. Insert the new fuse, close the fuse holder, and connect the line cord back to the IEC inlet.

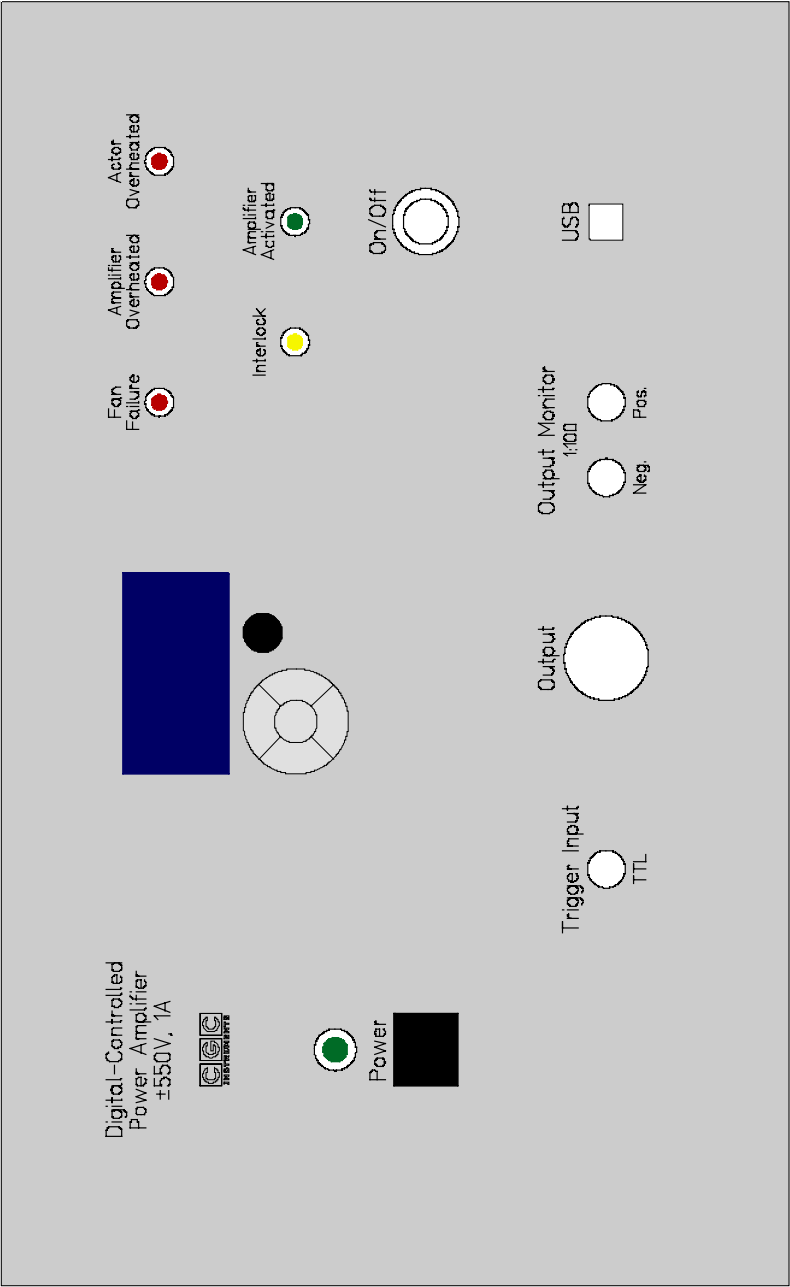


Fig. 2. The front panel.

Installation

To install the device, a space of at least $45 \times 65 \times 30 \text{ cm}^3$ (W \times D \times H) on top of a stable table is required. The device weighs about 40 kg, the supporting surface must guarantee mechanical stability under this load. The installation area must be kept dry and the temperature within the range specified in the section "Technical Data". Avoid exposing the device to direct sun light since this may substantially elevate the device temperature.

During operation, only the front panel of the device must be accessible. Before installing the device in its final position, you should install the line cord; it is connected to the IEC inlet on the rear panel. The heatsink at the rear panel must not be covered or obstructed during device operation, it must be provided with sufficiently cool air for active cooling of the device. Take necessary precautions to ensure a sufficient supply of cool air when installing the device in closed racks. Inspect and clean the fans at the heatsink regularly. Use a vacuum cleaner to remove dust from the fans and from the heatsink. Large dust layers substantially lower the cooling power of the heatsink, the device may overheat if the heatsink is extremely dirty. Also the lifetime of dirty fans may be significantly shortened.

To power the device, a power mains socket with proper grounding is required. Check the power rating of the mains socket, since the peak power consumption of the device may be as high as 1.6 kVA.

Before powering on the device, read the user manual carefully.

If already connected, disconnect the load and power on the device by toggling the rocker switch "Power". The LED above the power switch will light green and the display will turn on. The LED "Interlock" will indicate the state of the interlock loop. When the device startup completes, the other LEDs should be inactive. If the device is behaving properly, proceed with installing the waveform generator (see section "Quick Setup Guide").

Please consult the manual in case of any discrepancy with this description. Should that not resolve the problem, contact the distributor or the manufacturer.

Programmable Digital Waveform Generator

General

The waveform generator is that part of the high-voltage amplifier that is responsible for generating and managing the waveform signals. The waveform generator directly drives the input of the main amplifier, it also monitors the amplifier's operation state, as well as the state of the power supplies, the actor temperature and the interlock loop.

The waveforms are generated by a 16-bit D/A converter from data stored in a non-volatile memory. The waveform generation is controlled by an FPGA (*Field Programmable Gate Array*). The FPGA also monitors the crucial device parameters and shuts down the amplifier if necessary. The user interface and most of the control logic is realized by a microcontroller connected to the FPGA. The microcontroller serves the human interface (see section "Human Interface"), controls the USB data interface, and executes several monitoring tasks. The microcontroller's firmware realizes a user-friendly graphical interface. The device is controlled by the help of menus and dialog boxes, they provide an intuitive device control.

Using the USB interface, waveforms can be uploaded to the device and stored in its non-volatile memory (see section "Up- and Down-loading Waveforms"). Waveforms are sets of data samples; they do not contain any direct information about the frequency, the amplitude, or other device parameters. You can visualize and manage them using the waveform manager (see section "Waveform Management").

To output a waveform signal, you have to create a measurement. The measurement is a set of settings completely describing the waveform signal. It defines, for instance, which waveform data, which frequency, and which amplitude should be used to create an output signal. The measurement data is stored together with the waveform data in the non-volatile memory. The measurements can be managed by several dialog boxes (see section "Measurement Management").

To control the waveform output, another dialog box is available (see section "Measurement Control"). You can select a particular measurement defining the waveform signal and start, stop, or halt the signal output.

The firmware also provides a set of dialog boxes for monitoring the device parameters (see section "Device Monitoring"). Next, the firmware automatically manages several processes that permanently supervise the device and control the amplifier. For instance, the supply voltages and temperatures of the semiconductor power devices, as well as the actor temperature, are continuously checked and the amplifier is deactivated if necessary.

The waveforms can have sizes of up to 64 KS (65536 samples), the non-volatile memory, with its total size of 512 KW (= 524288 words of 16 bits = 1 MB), provides a space for 508 KS (= 520192 samples) of waveform data. This means that up to 7 waveforms with the maximum size of 64 KS can be stored simultaneously. The remaining 4 KW (8 KB) of the non-volatile memory are used to store the waveform and the measurement parameters as well as the system settings.

The USB interface can be used to upgrade the firmware. Further, a hardcopy utility is available that copies the current display contents via USB to a bitmap file at the host computer. For a detailed description, see the section "Software Utilities").

Quick Setup Guide

The device is delivered with cleared non-volatile memory. This means that there were neither waveforms nor measurements installed and set up. To prepare the device for a measurement, you have to upload waveforms and set up measurements. To do so, follow the next steps:

- Upload the waveforms as described in the section "Up- and Downloading Waveforms".
- Press the middle key "enter" and start the waveform management by selecting the menu item "Manage ► Waveforms" and check the uploaded waveforms. Then close the waveform management. For details see section "Waveform Management".
- Press the middle key "enter" and start the measurement management by selecting the menu item "Manage ► Measurements". Create a new measurement by selecting the menu item "New ► Current". Select the desired waveform and setup the measurement parameters. Save the measurement by selecting the menu item "Save", then close the measurement management. For details see section "Measurement Management".
- Press the middle key "enter" and start the measurement control by selecting the menu item "Measurement". Choose a measurement by selecting the menu item "Select". Launch the measurement by selecting the menu item "Run". Check the output signals with an oscilloscope connected to the monitor outputs. Stop the measurement by selecting the menu item "Stop". For details see section "Measurement Control".

If you encounter problems, read carefully the corresponding section in this manual. If the measurement cannot be launched, check the device parameters by opening the dialog boxes for monitoring (see section "Device Monitoring").

Up- and Downloading Waveforms

The waveform data can be transferred into/from the device via the USB interface. To establish communication with a host computer, you need to install the port driver for USB (see section "Driver Installation"). When the installation is finished, copy the software from the directory "Program" of the enclosed software package to a suitable directory of your computer. Locate the program `WaveLoader.exe` and the port settings file `COM.ini`. Edit the file `COM.ini` as described in the section "Functionality of the Software Interface" to set the virtual port number. Open a command prompt window and launch the program `WaveLoader.exe`; it should start without any error message. In case of any problem, display the port settings by pressing the key 'i' and check whether the port number matches the system settings (see section "Driver Installation"). The remaining parameters should be as follows:

```
BaudRate:      230400
ByteSize:      8
StopBits:      1
Parity:        even
```

If you discover any discrepancies, correct them in the configuration file `COM.ini`. Note that wrong port settings most probably prevent the device from communicating with the host computer. Please also consider that the host computer can communicate with the device only if the device is in the idle state (see Fig. 3). You must close all opened dialog boxes or menus to enable the communication. On the other hand, if you open a menu or a dialog box, the device's communication buffers are erased and the communication stops. Thus, opening and closing a menu or a dialog box may help you, in some cases, to restore a broken communication.

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

```
Product identification: DAFG-1 Rev.1-00
```

If an error occurs, please consult the section "Functionality of the Software Interface". The tables 5 and 6 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 'W' to get the list of all uploaded waveforms. If no waveforms have been uploaded yet, the list should be empty. You may clear the waveforms by pressing the key 'E' or exit the program by pressing 'Esc'.

In practice, you may prefer the command line mode instead of the interactive mode. The former mode allows you, for instance, to save the complete commands in batch files for repeated usage. To list all uploaded waveforms, enter the following command:

```
WaveLoader -W -t
```

Similarly, the command:

```
WaveLoader -E -t
```

erases all waveforms, the command:

```
WaveLoader -e5 -t
```

erases the waveform number 5.

To upload a waveform, enter, for instance, the command:

```
WaveLoader -c0 5f-c.txt 5 "5sine const. ampl." -t
```

This creates a waveform with the number 0 from the file 5f-c.txt. The waveform parameter "Periods" will be set to 5, the name will be "5sine const. ampl." (for more details see section "Waveform Management").

Table 2 summarizes all allowable command line parameters of the program `WaveLoader`. 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.

If you wish to specify a name parameter containing spaces or special characters, use the conventions valid for your operating system. In Windows systems, for instance, enclose the name in quotation marks.

Please note that the program does not request any user confirmation when manipulating the waveforms. Thus, use the list command

```
WaveLoader -W -t
```

prior to any of the waveform manipulations to ensure that you will delete or create waveforms with the proper number.

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:

```
WaveLoader -W -e5 -W -t
```

lists all waveforms, erases the waveform number 5, and then lists all remaining waveforms.

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

```
WaveLoader -q -W -t > WaveList.txt
```

It outputs the waveform listing into the file `WaveList.txt`. Using this file, you can restore the waveform parameters in case of any erroneous operation of the device or an operator failure.

To save all waveform data, you may create a batch file with the following contents:

```
WaveLoader -S0 0.txt -t
WaveLoader -S1 1.txt -t
...
```

By executing this file, the waveform data are saved in hexadecimal format in that they are stored in the non-volatile memory of the device.

To check the uploaded data, download the data to the host computer, for instance, by the command:

```
WaveLoader -s0 0.txt -t
```

The text file `0.txt` will contain the data in floating-point format. The data file may differ from the data originally uploaded, since the waveform samples are stored in the device using 16-bit precision.

Tab. 2. Command line parameters of the program WaveLoader.

Parameter	Explanation
-wN	get the parameters of the waveform number N
-W	get the parameters of all waveforms
-eN	delete the waveform number N
-E	delete all waveforms
-sN FileName	save the data of the waveform number N into a text file with the name FileName, the data are in floating point format ranging from -1 to +1
-SN FileName	save the data of the waveform number N into a text file with the name FileName, the data are in 16-bit hexadecimal format
-cN FileName P Name	create the waveform number N from a text file with the name FileName, the parameter "Periods" will be P, the waveform name Name; the data are in floating point format ranging from -1 to +1
-CN FileName P Name	create the waveform number N from a text file with the name FileName, the parameter "Periods" will be P, the waveform name is Name; the data are in 16-bit hexadecimal format
-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, -U	get the device uptime
-b, -B	get the device buffer status
-z, -Z	purge the communication
-i, -I	display the port parameters read from the configuration file COM.ini

Parameter	Explanation
-t, -T	terminate the program
-q, -Q	quiet mode
-g, -G	debug mode
-?	show the online help

Human Interface

The device is equipped with a graphic liquid crystal display (LCD), a set of keys (keypad) and an 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 encoder is used to change numerical values or select items. The function of the encoder in every device state is symbolized on the LCD immediately above the encoder knob. In most situations, 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 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. In some situations, the encoder's push button is used to confirm the selected value or to switch the encoder function. In such cases, the symbol "⏏" is shown on the LCD above the encoder knob.

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. 3). The lower right corner shows the uptime, i.e. the time elapsed from the last device start.

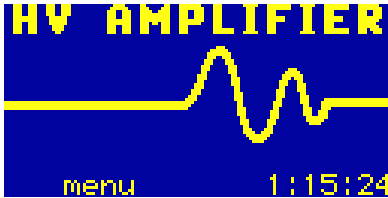


Fig. 3. Program display in device idle state.

By pressing the middle key "enter", the main menu can be accessed (see Fig. 4). You can start a measurement (menu item "Measurement", see also section "Measurement Control"), manage the device configuration (submenu "Manage", see also section "Device Management"), monitor the device function (submenu "Monitor", see also section "Device Monitoring"), or manage the system settings (submenu "Adjustment").

You can save the current system settings (menu item "Adjustment ► Save"), load the settings back (menu item "Adjustment ► Load"), or reset them to default values (menu item "Adjustment ► Reset"). The system settings are saved automatically to a non-volatile memory when the device powers down and are loaded back when the device starts. 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 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. Note that the settings reset influences neither the waveform data nor the measurement settings. If you wish to also clear this data, start the waveform or the measurement management (see sections "Waveform Management" and "Measurement Management") and clear the data in the corresponding dialog box.

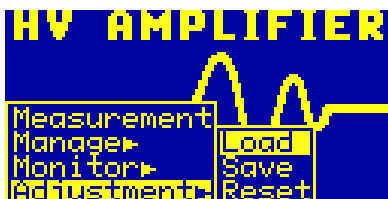
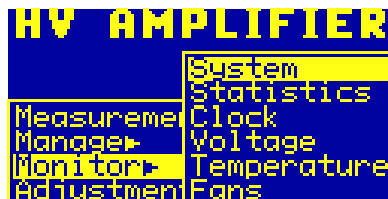
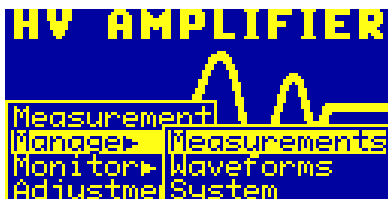


Fig. 4. Main program menu.
The figure shows the different submenus of the main program menu.

Device Management

Several dialog boxes are available to control the device settings and to setup waveforms and measurements.

Waveform Management

Using the dialog box "Waveform", you can visualize and manage the waveforms (see Fig. 5).



Fig. 5. The dialog box "Waveform".

To select a particular waveform, use the encoder. The current waveform number is shown in the title bar of the dialog box. The next row displays the waveform name. In the main part of the dialog box, the waveform is drawn. For longer waveforms, the drawing may take several seconds. However, the drawing runs as a background process, it does not block the operation of the device.

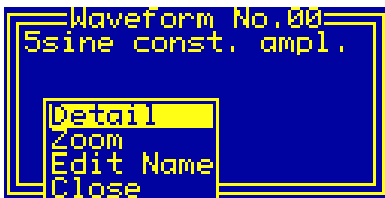


Fig. 6. The menu in the dialog box "Waveform".

By pressing the middle key "enter", the menu can be accessed (see Fig. 6). You can select one of the waveform functions or close the dialog box. The functions "Detail" and "Zoom" open new dialog boxes, the menu item "Edit Name" invokes a simple editor in the current dialog box (see Fig. 7).

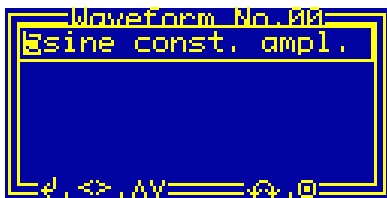


Fig. 7. The waveform name editor in the dialog box "Waveform".

The waveform name editor is controlled by the direction keys and by the encoder. The horizontal direction keys select the current character, it is displayed inversely. The vertical direction keys or the encoder change the current character. The change occurs according to the ASCII character code, it may range from 32 (space) to 127 (copyright sign); for other codes, see the standard ASCII character table or check the LCD character set (see Tab. 4). By pressing the middle key "enter" or the encoder button, the editor will be closed and the changes saved.

The dialog box "Waveform Detail" displays the detailed information about the current waveform (see Fig. 8).

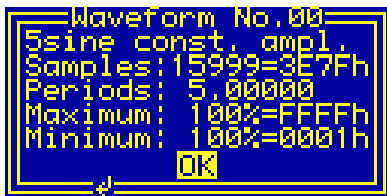


Fig. 8. The dialog box "Waveform Detail".

The title and the next row are identical to those of the main "Waveform" dialog box (see Fig. 5). The value "Samples" shows the number of waveform samples decreased by one. Thus, the waveform shown in Fig. 8 consists of 16,000 samples. The value is shown in decimal (the first value) and in hexadecimal format (the second value terminated by 'h').

The value "Periods" is the number of signal periods within the waveform. This auxiliary value should help you to set up the waveform timing, however, it does not influence the waveform output in any way. You should set the value to a number that best characterizes your

waveform. The value can be changed during waveform creation only, you cannot edit it.

The values "Maximum" and "Minimum" show the extremes of the waveform data. The values are shown as percent values (the first values terminated by '%') and in hexadecimal format (the second values terminated by 'h'). The hexadecimal values correspond to the real 16-bit waveform data, the absolute minimum equals 0000h, the absolute maximum is FFFFh. The percent values are related to these extremes. For longer waveforms, the calculation of the extreme values may take up to about one second. However, the calculation runs as a background process, it does not block the operation of the device.

Using the dialog box "Waveform Zoom", you can visually inspect the selected waveform in detail (see Fig. 9).



Fig. 9. The dialog box "Waveform Zoom".

Similarly to the main "Waveform" dialog box (see Fig. 5), the title bar shows the current waveform number. The next row displays the currently used waveform scale. The first two numbers are the horizontal limits of the drawing, the second two numbers the vertical limits. The values are shown in hexadecimal format.

Using the horizontal direction keys, you can change the horizontal resolution, the vertical direction keys change the vertical resolution: The keys "left" and "up" increase the resolution, the keys "right" and "down" decrease it. When the resolution is being changed, the new display limits are obtained such that the display center does not move. This is, however, not possible when the resolution is being decreased and one of the new display limits would exceed the allowable values, i.e. the waveform size. In this case, the drawing limits are shifted accordingly while maintaining the desired scale.

Using the encoder, the current view can be shifted. By pressing the encoder button, you can change the shift direction between horizontal and vertical. The default direction is horizontal as shown in Fig. 9.

The menu of the dialog box is accessible by pressing the middle key "enter" (see Fig. 10). Using the menu, you can modify the drawing scale or close the dialog box and return to the main dialog box "Waveform" (see Fig. 5).

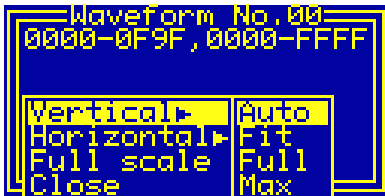


Fig. 10. The menu in the dialog box "Waveform Zoom".

Selecting the menu item "Full scale", the drawing resets to the initial state using the full scales in both directions. You can set the full scale for each direction separately by selecting the menu items "Vertical ► Full" or "Horizontal ► Full", respectively. By contrast, the menu items "Vertical ► Max" or "Horizontal ► Max", select the maximum possible resolution in the particular direction. If the maximum resolution is chosen, one display pixel step corresponds exactly to one data step, i.e. no waveform details can be lost by the drawing. Due to the limited display resolution, however, only a rather small part of the waveform can be drawn using the maximum resolution. Thus, in practice, lower resolutions are more suitable.

By selecting the menu item "Vertical ► Fit", an optimum vertical scale is calculated and the waveform is redisplayed using the new scale. This procedure requires about twice the time to stabilize the drawing compared to a normal operation. Be sure that you select this menu item after the waveform drawing with the currently selected horizontal resolution has completed, otherwise the proper scaling data are not available and the function fails.

If you choose the menu item "Vertical ► Auto", the automatic optimization of the vertical scale is applied permanently. This takes more time but provides the best results when inspecting a waveform in detail. When the horizontal resolution or the horizontal position change, the waveform extremes are obtained while drawing the waveform.

When the drawing finishes, the optimum scale is obtained and the waveform is redrawn. These processes run in background, thus do not block the device in any way. You can shift the waveform or change the display scale without having to wait for the drawing to finish.

Please note that it does not make sense to move the waveform vertically or to change the vertical resolution if the automatic optimization of the vertical scale is active. After the device has drawn the waveform using the new display limits, the waveform will be redrawn again with the original optimum scale so that the final drawing does not change at all.

The recommended procedure for waveform inspection is as follows: Increase the horizontal resolution such that you can see the details that you are searching for. Turn on the automatic optimization of the vertical scale and use the encoder to move through the waveform. If you shifted the waveform by about one half of the display width, wait a moment until the drawing stabilizes. Repeat this procedure for the whole waveform or for the part that you are inspecting.

Measurement Management

Using the dialog box "Measurement", you can setup and manage the measurements (see Fig. 11).

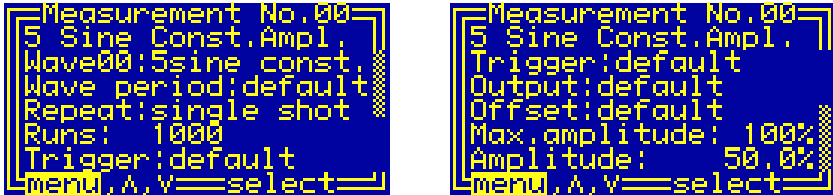


Fig. 11. The dialog box "Measurement".

Two different views on the dialog box that correspond to different positions of the scrollbar located at the right side.

To select a particular measurement, use the encoder. The current measurement number is shown in the title bar of the dialog box. The next row displays the measurement name.

Use the vertical direction keys to change the currently selected item in the dialog box. The selected item is highlighted by inverting the displayed characters. By pressing the middle key "enter", the settings can be modified or the dialog menu can be activated (see the selection shown in Fig. 11). The menu allows you to manage the measurements, save the current measurement, or close the dialog box (see Fig. 12). Please note that the changes made to a measurement will be lost if you progress to another measurement or close the dialog box without having selected the menu item "Save".

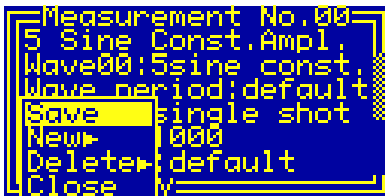


Fig. 12. The menu in the dialog box "Measurement".

By selecting the menu items "New ► Current" or "New ► Last", a new measurement will be created. The menu item "New ► Current" inserts the measurement at the currently selected position; all meas-

urements with equal or higher numbers are renumbered. The menu item "New ► Last" appends a new measurement at the end of the measurement set.

By selecting the menu items "Delete ► Current" or "Delete ► All", measurements can be erased. The menu item "Delete ► Current" deletes the currently selected measurement; all measurements with higher numbers are renumbered. The menu item "Delete ► All" deletes all available measurements. Each deletion must be confirmed in a separate message box. When deleting all measurements, two confirmations must be positively answered to allow the erasing. This measure prevents accidental deleting of the measurement data.

By pressing the middle key "enter" when the measurement name is selected, a simple editor in the current dialog box is invoked (see Fig. 13). The function of the editor equals the waveform name editor (see Fig. 7 and the corresponding text).

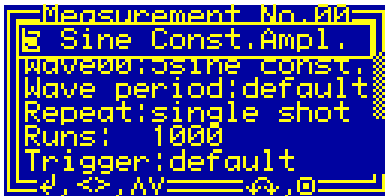


Fig. 13. The measurement name editor in the dialog box "Measurement".

When the waveform name is selected, press the middle key "enter" to change the waveform assigned to the measurement. A new dialog box will be opened that is similar to the dialog box "Waveform" (see Fig. 5). The only difference to the dialog in Fig. 5 is the presence of the last menu item "Select" (see Fig. 14) that assigns the selected waveform to the current measurement.

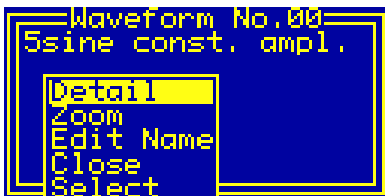


Fig. 14. The menu in the dialog box "Waveform" for selecting waveforms.

The value "Wave period" shows the currently set waveform duration, it can be changed by a menu using the direction keys (see Fig. 15). You can select the default value (see Fig. 23 and the corresponding text) or setup any other waveform duration. When the menu item "setup" is selected, a new dialog box opens (see Fig. 16).

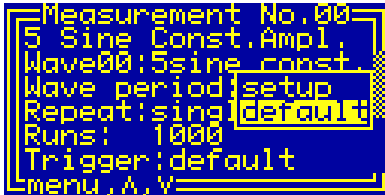


Fig. 15. Setting the waveform duration in the dialog box "Measurement".

The dialog box "Timing Setup" (see Fig. 16) allows you to set the waveform duration; it is displayed as the value "Wave period". Use the encoder to change the value or press the middle key "enter" to close the dialog and save the settings.

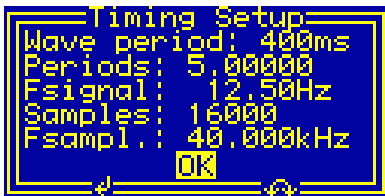


Fig. 16. The dialog box "Timing Setup".

The values "Fsignal" and "Fsampl." show the signal and sampling frequencies, respectively. The signal frequency is calculated from the waveform duration and from the value "Periods" that is a part of the waveform definition:

$$F_{\text{signal}} = \text{Periods} / \text{Wave period.} \quad (1)$$

Similarly, the sampling frequency is calculated from the waveform duration and from the waveform size "Samples":

$$F_{\text{sampling}} = \text{Samples} / \text{Wave period.} \quad (2)$$

Note that you cannot exit the dialog box "Timing Setup" without saving the data. You can, however, discard the changes of the settings in the main dialog box "Measurement".

The value "Repeat" in the dialog box "Measurement" displays the operation mode of the waveform generator, it can be changed by a menu using the direction keys (see Fig. 17). If "single shot" is selected, the generator produces one waveform per trigger and waits for the next trigger. If "continuous" is selected, the generator produces the waveform repeatedly, once it has been triggered.

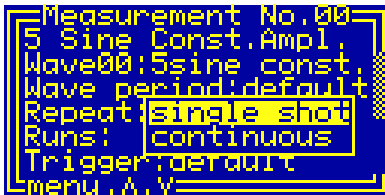


Fig. 17. Setting the waveform repeating in the dialog box "Measurement".

The value "Runs" shows how many waveforms should be output until the generator stops the waveform generation. The value can be changed by the encoder (see Fig. 18). To set large numbers, press the direction key "left" - the value will change one hundred times faster. The minimum value is 1 run, the maximum is 999,999 runs. Thus, when generating a sine waveform with the frequency of 1 kHz, the total signal duration is limited to 1000 seconds, i.e. about 16.5 minutes. When this period has elapsed, the measurement must be restarted to continue the waveform generation.

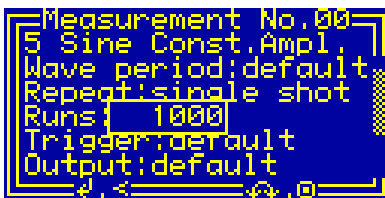


Fig. 18. Setting the number of waveform repetitions in the dialog box "Measurement".

The value "Trigger" shows the trigger slope, it can be changed by a menu using the direction keys (see Fig. 19). You can select between rising and falling slopes, use the device default value (see Fig. 24 and the corresponding text), or set the manual triggering. When the manual triggering is selected, the waveform generation will be started by a menu item. In other cases, an external signal with the appropriate polarity at the trigger input launches the waveform generation.

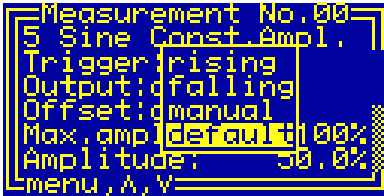


Fig. 19. Setting the trigger slope in the dialog box "Measurement".

The values "Output" and "Offset" show the amplifier settings, they can be changed by a menu using the direction keys (see Fig. 20). You can select between "permanent on" and "automatic on", or select the default value (see Fig. 25 and the corresponding text). If the "Output" is effectively set to "permanent on", the amplifier needs to be activated by the push button on the front panel and it remains active until it is turned off. If the "Output" is effectively set to "automatic on", the device automatically activates the amplifier at the beginning of the measurement and deactivates it when the measurement finishes. This mode reduces the power consumption and the heat production of the device. If the "Offset" is effectively set to "permanent on", the amplifier generates the offset voltage immediately when it is activated. If the "Offset" is effectively set to "automatic on", the amplifier starts to generate the offset voltage at the beginning of the measurement and turns the offset voltage off when the measurement finishes. This option can be applied only if the "Output" is set to "permanent on" or to "default", otherwise the item is not selectable and the value is set to "automatic on".

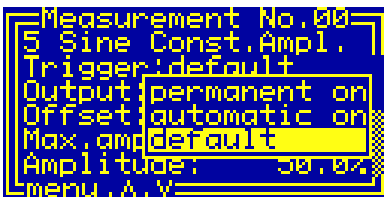


Fig. 20. Setting the amplifier activation in the dialog box "Measurement".

The values "Max. amplitude" and "Amplitude" show the settings of the output signal amplitude, they can be changed by the encoder (see Fig. 21). The value "Max. amplitude" defines the maximum allowable signal amplitude, the output signal is in any case limited to this value.

The value "Amplitude" is used as the starting signal amplitude when the dialog box "Measurement Control" is opened (see Fig. 27 and section "Measurement Control"). The signal amplitude can be changed during the waveform generation, it can, however, not exceed the value "Max. amplitude". The value "Max. amplitude" may not be smaller than the value "Amplitude". The device corrects the settings automatically to match this requirement.

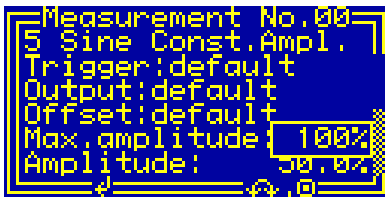


Fig. 21. Setting the output signal amplitude in the dialog box "Measurement".

System Preferences

The dialog box "System Preferences" (see Fig. 22) shows the current system settings and enables you to change them. Some of the settings can be used as default values for setting up a measurement. You can modify all measurements using these defaults in a single step by changing the system settings.

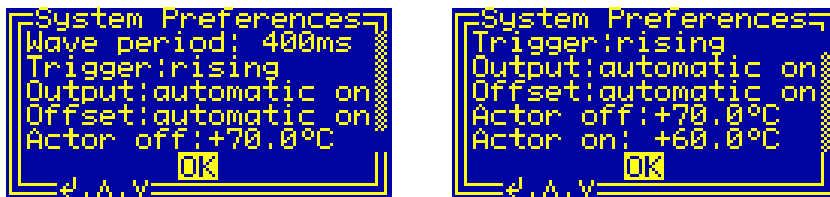


Fig. 22. Dialog box "System Preferences".

Two different views of the dialog box that correspond to different positions of the scrollbar located at the right side.

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. 22).

The value "Wave period" shows the default waveform duration, it can be changed using the encoder (see Fig. 23). The minimum value is 1 ms, the maximum 999 s. The value is used as device default for the waveform duration. You can set the waveform duration to "default" in the dialog box "Measurement" (see Fig. 15), the device will use the device default value set in this dialog.

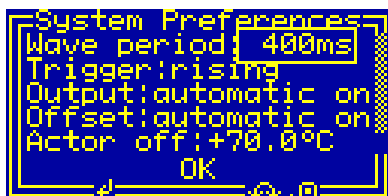


Fig. 23. Setting the waveform duration in the dialog box "System Preferences".

The value "Trigger" shows the default trigger slope, it can be changed by a menu using the direction keys (see Fig. 24). You can select be-

tween rising and falling slopes. The value is used as device default for the trigger slope. You can set the trigger slope for a measurement to "default" in the dialog box "Measurement" (see Fig. 19), the device will use the device default value set in this dialog.

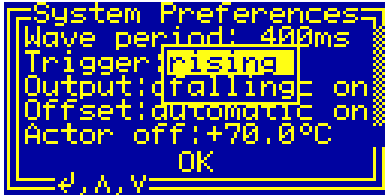


Fig. 24. Setting the trigger slope in the dialog box "System Preferences".

The values "Output" and "Offset" show the default amplifier settings, they can be changed by a menu using the direction keys (see Fig. 25). You can select between "permanent on" and "automatic on". If the "Output" is set to "permanent on", the amplifier needs to be activated by the push button on the front panel and it remains active until it is turned off. If the "Output" is set to "automatic on", the device automatically activates the amplifier at the beginning of the measurement and deactivates it when the measurement finishes. If the "Offset" is set to "permanent on", the amplifier generates the offset voltage immediately when it is activated. If the "Offset" is set to "automatic on", the amplifier starts to generate the offset voltage at the beginning of the measurement and turns the offset voltage off when the measurement finishes. This option can be applied only if the "Output" is set to "permanent on", otherwise the item is not selectable and the value is set to "automatic on". The values from the dialog box are used as device defaults for the amplifier settings. You can set the "Output" and the "Offset" for a measurement to "default" in the dialog box "Measurement" (see Fig. 20), the device will use the device default values set in this dialog.

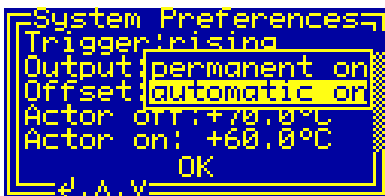


Fig. 25. Setting the amplifier activation in the dialog box "System Preferences".

The values "Actor off" and "Actor on" show the thresholds of the actor temperature. If the actor becomes warmer than the value "Actor off", the amplifier will be deactivated and its output will be shorted so that the actor will not be loaded any more, and thus should cool down. The amplifier can be activated again, if the actor temperature has decreased below the value "Actor on". The values can be changed using the encoder (see Fig. 26). The minimum settable value is 30°C, the maximum is 99.9°C. The difference between the values must be larger than 1°C, the values are automatically corrected to meet these requirements. You should set the values according to the data sheet of the actor. Note that a too high operation temperature of the actor may shorten its lifetime or even lead to its permanent destruction.

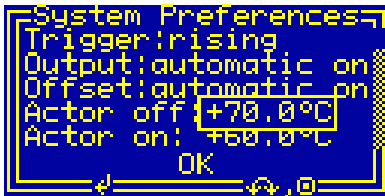


Fig. 26. Setting the allowable actor temperatures in the dialog box "System Preferences".

Measurement Control

The waveform generation is controlled by the dialog box "Measurement Control" (see Fig. 27).

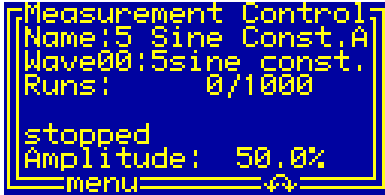


Fig. 27. The dialog box "Measurement Control".

The text "Name" is the name of the selected measurement (see also Fig. 13), "WaveNN" is the name of the corresponding waveform, the two numerals NN in the waveform label ("00" in Fig. 27) stand for the waveform number (see Fig. 5). The values "Runs" show the number of waveforms that have been generated and the total number of waveforms that should be generated. The latter value is given by the measurement settings (see Fig. 18). The waveform generation stops if the left number reaches the value of the second number. The text "stopped" shows the device status, it changes according to the measurement phase or to a possible failure (see Tab. 3). The value "Amplitude" is the currently set output signal amplitude. Its initial and maximum values are determined by the corresponding measurement settings (see Fig. 21). The amplitude value can be set by the encoder between 0% and the maximum allowed amplitude.

By pressing the middle key "enter", the menu can be accessed. The menu structure changes according to the device state. After opening the dialog box "Measurement Control", i.e. when the waveform generator is stopped, the menu allows you to select a measurement, to launch it, or to close the dialog box (see Fig. 28). With a submenu, the device monitoring can be started (for details see section "Device Monitoring"). The function of the submenu is identical to the submenu "Monitor" of the main menu (see Fig. 4). The device monitoring is accessible from all menus independent of the device state.



Fig. 28. Start menu of the dialog box "Measurement Control".

The menu item "Select" opens a new dialog box that allows you to choose a measurement. This dialog box is similar to the dialog box "Measurement" (see Fig. 11). The only difference to the dialog in Fig. 11 is the presence of the last menu item "Select" (see Fig. 29) that selects the current measurement.

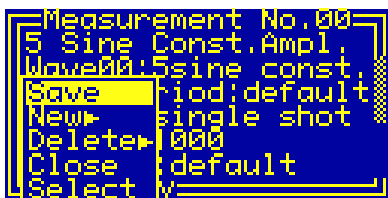


Fig. 29. The menu in the dialog box "Measurement" for selecting measurements.

The menu item "Run" launches the selected measurement. If required, the amplifier and its output offset voltage will be activated (see Fig. 20 and the corresponding text). Next, the device waits for the trigger (see Fig. 30). If manual triggering is selected (see Fig. 19), the status changes to "waiting for start". The waveform output can be launched by the menu item "Start" that can be found in the dialog menu now.

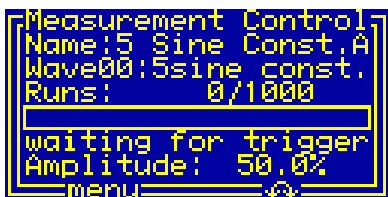


Fig. 30. The dialog box "Measurement Control" when the device is waiting for the trigger.

If the device has been triggered, the waveform output starts (see Fig. 31). The progress bar above the status text graphically indicates the time elapsed from the beginning of every particular waveform. Please note that due to the display delay, the real picture may appear blurred if the progress bar changes rapidly.

Depending on the settings, one waveform or a series of waveforms is generated until the device stops. The waveform generation may also be interrupted by a fault condition. This may be an opened interlock loop, amplifier or actor overheating, or a supply voltage fault. The waveform output may be also halted by the menu item "Halt" or stopped by the menu item "Stop". If halted, the waveform output can be restored again by the menu item "Continue". If manual triggering is selected, the menu item "Continue" also triggers the device so that the waveform output is launched immediately. However, if external triggering is selected, the device waits for the next trigger before the waveform output is restarted.

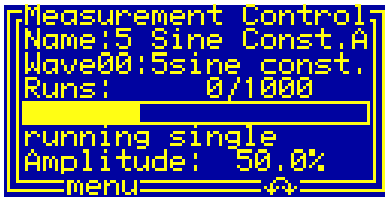


Fig. 31. The dialog box "Measurement Control" when the device is generating a waveform.

If the waveform generation stops, the amplifier and the output offset are deactivated if necessary. The deactivation does not happen, if the amplifier and the output offset are turned permanently on (see Fig. 20 and the corresponding text). When this shutdown has finished, the device enters the idle state, i.e. the same state as before the measurement was started (see Fig. 27).

The possible states of the waveform generator are summarized in Tab. 3. The table lists the texts that can be displayed in the status line of the dialog box "Measurement Control" and their explanations.

Tab. 3. Possible states of the waveform generator.

Text	Explanation
stopped	the waveform generator is idle
waiting for power	the device is waiting until the power supply voltages reach their correct values
waiting for offset	the device is waiting until the output offset voltage stabilizes
waiting for trigger	the device is waiting for an external trigger
waiting for start	the device is waiting for a manual trigger, select the menu item "Start" to launch the waveform generation
waiting for shutoff	waveform generation has finished, the device is shutting off the power supply and is waiting until the power supply voltages reach safe values
running single	single waveform is being generated
running continuous	waveforms are being generated continuously
halted	waveform generation is halted, select the menu item "Continue" to restart the waveform generation
!!supply failure!!	one of the supply voltages was out of the allowable range since the power supply was turned off, overloaded, overheated, or is defective; the amplifier was deactivated and the waveform generation stopped
activation failure!	the device could not be activated, the interlock loop may be open, the amplifier or the actor may be overheated
frequency failure!	the required sampling frequency is out of the allowable range, check and correct the measurement settings

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.

Information

The dialog box "Information" (see Fig. 32) summarizes general information about the device. The text "ID" shows the product identification, "Product No." is the unique product serial number. The entries "HW version" and "FW version" designate the hardware and the firmware versions, "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.

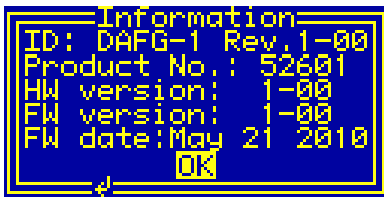


Fig. 32. Dialog box "Information".

Statistics

The dialog box "Statistics" (see Fig. 33) shows information about the operation time of the device. 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 "Amplifier" shows the operation time of the amplifier elapsed since powering on the device or since the last firmware restart. The next line "Total" shows the total operation time of the amplifier.



Fig. 33. Dialog box "Statistics".

Clock

The dialog box "Clock" (see Fig. 34) shows the current frequencies of various clock signals used internally to control the device. The value "CPU" is the frequency of the main microcontroller clock that controls the device. Since the microcontroller clock is dynamically changed according to the current load, this value may vary according to the number of active processes.



Fig. 34. Dialog box "Clock".

The value "Periph." is the frequency used to control the peripheral devices of the microcontroller. It is regulated by the microcontroller to a value of about 15.67 MHz.

The value "System" is the frequency of the clock controlling the FPGA (*Field Programmable Gate Array*). The FPGA is responsible for the waveform generation and for activating the amplifier, its clock is gained from the main quartz oscillator.

The value "DAC" is the frequency of the clock controlling the DAC (*Digital Analog Converter*) generating the waveforms. The clock is exactly three-times higher than the waveform sampling rate, it is gained by a PLL (*Phase Locking Loop*) circuitry. The PLL multiplies and divides the frequency of the main quartz oscillator to get the desired DAC clock.

The value "DMA" is the frequency of the clock controlling the DMA (*Direct Memory Access*) circuit used to move waveform data when a waveform is being inserted or deleted. The clock is gained by a PLL circuitry, the clock frequency is set to 80 MHz.

The frequencies are measured as ratios between the particular frequency and the frequency of an auxiliary quartz oscillator running at 32.768 kHz. If this oscillator fails, all measured values are subjected to systematic deviations.

Voltage Monitor

The dialog box "Voltage Monitor" (see Fig. 35) shows the current values of the supply voltages of the device.

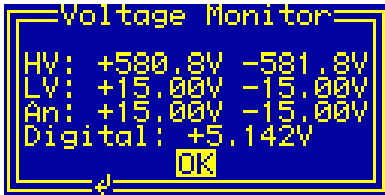


Fig. 35. Dialog box "Voltage Monitor".

The values "HV" are the positive and negative high voltages supplying the amplifier. If the amplifier is inactive, the voltages are near zero. They rise to about 580 V if the amplifier is activated.

The values "LV" are the positive and negative low voltages supplying the amplifier. If the amplifier is inactive, the voltages are near zero. They rise to about 15 V if the amplifier is activated.

The values "An" are the positive and negative voltages supplying the analog circuits of the device. The nominal values are 15 V. The voltages are present independent of the amplifier state.

The value "Digital" is the voltage supplying the digital circuits of the device. The nominal value is 5 V. The voltage is present independent of the amplifier state.

The values are measured about 2.5-times per second. The microcontroller checks them continuously and deactivates the amplifier if the voltages differ from their nominal values. This may happen when the amplifier is overloaded or when one of the power supplies overheats.

If you aim to manipulate the output cables, the connectors, or the actor, use the dialog box "Voltage Monitor" to check the supply voltages. Be sure that you open the connectors or touch the actor only if the supply voltages have decreased to values that are not lethal.

Temperature Monitor

The dialog box "Temperature Monitor" (see Fig. 36) shows the measured temperatures of the device and of the connected actor.

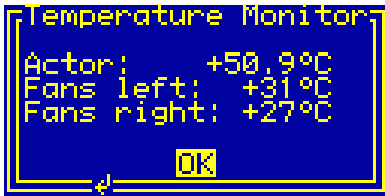


Fig. 36. Dialog box "Temperature Monitor".

The value "Actor" is the temperature measured using a Pt1000 sensor at the actor. The sensor is connected via the output cable to the device. The microcontroller calculates the temperature from the sensor resistance using the simple linear formula:

$$R = 1 \text{ k}\Omega \cdot (1 + 0.00385 \cdot t / ^\circ\text{C}), \quad (3)$$

where R stands for the sensor resistance and t the sensor temperature.

The values "Fans left" and "Fans right" are the temperatures measured using 8 semiconductor sensors in the amplifier. The sensors are located directly at the 8 power semiconductor devices of the amplifier. The sensors are grouped into two groups with 4 sensors each. The measured value is the maximum temperature determined over the 4 sensors of each group. The sensors are also used to control the fans at the heatsink, each group controls one fan pair.

The values are measured about 2.5-times per second. The microcontroller checks the values continuously and deactivates the amplifier if the temperature rises above the allowed values. The microcontroller is also able to discover failures such as a disconnected or shorted sensor; any failure causes an immediate deactivation of the amplifier.

Fan Monitor

The dialog box "Fan Monitor" (see Fig. 37) shows the state of the fans located at the heatsink of the device.

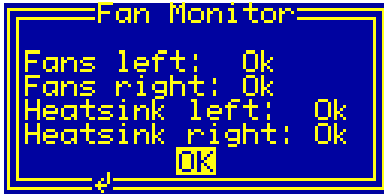


Fig. 37. Dialog box "Fan Monitor".

The values "Fans left" and "Fans right" indicate the state of the fans. The values shown may be either "Ok" if the fans are running properly or "failure" if one of the fans has failed.

The values "Heatsink left" and "Heatsink right" show the temperatures of the sensors controlling the fans (for more details, see the section "Temperature Monitor"). The values shown may be either "Ok" or "hot" if the temperature exceeds the allowed limit.

Tab. 4. The LCD character set.

value	char	value	char	value	char	value	char
20h	space	38h	8	50h	P	68h	h
21h	!	39h	9	51h	Q	69h	i
22h	"	3Ah	:	52h	R	6Ah	j
23h	#	3Bh	;	53h	S	6Bh	k
24h	\$	3Ch	<	54h	T	6Ch	l
25h	%	3Dh	=	55h	U	6Dh	m
26h	&	3Eh	>	56h	V	6Eh	n
27h	'	3Fh	?	57h	W	6Fh	o
28h	(40h	@	58h	X	70h	p
29h)	41h	A	59h	Y	71h	q
2Ah	*	42h	B	5Ah	Z	72h	r
2Bh	+	43h	C	5Bh	[73h	s
2Ch	,	44h	D	5Ch	\	74h	t
2Dh	-	45h	E	5Dh]	75h	u
2Eh	.	46h	F	5Eh	^	76h	v
2Fh	/	47h	G	5Fh	_	77h	w
30h	0	48h	H	60h	`	78h	x
31h	1	49h	I	61h	a	79h	y
32h	2	4Ah	J	62h	b	7Ah	z
33h	3	4Bh	K	63h	c	7Bh	{
34h	4	4Ch	L	64h	d	7Ch	
35h	5	4Dh	M	65h	e	7Dh	}
36h	6	4Eh	N	66h	f	7Eh	~
37h	7	4Fh	O	67h	g	7Fh	©

Software Utilities

The software utilities can be found in the directory "Program" of the enclosed software package. Before using them, the virtual USB port driver 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. If necessary, edit the file `COM.ini` to set the virtual port number, as described in the section "Functionality of the Software Interface".

Utility FlashLoader

The FlashLoader is a simple Windows program running in text mode. It enables you to upgrade the firmware of the waveform generator. 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, start the following command:

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

where `Firmware.txt` is the file containing the current firmware. The program should produce the following output:

```
FL 1.00
Verifying code file Firmware.txt
Verifying finished
Resetting the target
Program finished ok
```

During the verify procedure, a message box is displayed at the device display 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 amplifier is active. The utility stops all processes running at the microcontroller, thus the monitoring is not available and the amplifier would not be deactivated if a critical situation arises. To be completely sure that the amplifier cannot be activated when the FlashLoader is active, open the interlock loop, for instance, by disconnecting the output cable.

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 "Up- and Downloading Waveforms".

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

```
FlashLoader Firmware.txt
```

where `Firmware.txt` is the file with the new firmware. The program should produce the following output:

```
FL 1.00
Programming code file Firmware.txt
Programming finished
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 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 re-program 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 power on the device. If this small part of the firmware is still working, the flash loader will start. Then, try to launch the FlashLoader with the command line parameter '-s':

```
FlashLoader 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 HardCopy

The HardCopy is a simple Windows program running in text mode. It enables you to receive a copy of the display contents and to store it in a bitmap file. This may be useful for documentation purposes or for creating a bug report for the manufacturer. Launching the utility `HardCopy.exe` without any parameters displays a simple help text with the expected syntax of the command line.

To make a hardcopy of the current display contents, start the following command

```
FlashLoader Image.bmp -v
```

where `Image.bmp` is the bitmap file that will be created by the utility. The utility starts and waits for the data from the device. You can interrupt it by pressing 'Esc' at the host computer keyboard. To activate the hardcopy, press, simultaneously, both vertical direction keys at the device's keypad. The device sends the hardcopy data, the utility receives it, stores, and terminates itself. If successfully finished, it should produce the following output:

```
Waiting for response, press Esc to break
Program finished successfully, bitmap file
Image.bmp created
```

If you encounter any error, follow the instructions described in the section "Up- and Downloading Waveforms".

Please note that the device may interpret the subsequently pressed direction keys and perform the corresponding action. Since the device does not use any combination of simultaneously pressed keys to control its operation, you may hold the last pressed key that opened the dialog box or menu that you wish to hardcopy. Then press one of the vertical direction keys without releasing the first key. Release the first key and press the second vertical direction key. This starts the hardcopy without invoking any other operation.

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. Please note the following:

- As an alternative to the drivers from the enclosed software package, you can 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. Please note that the software interface can communicate only with ports COM1 to COM9. 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.

Software Interface

The software interface for the device consists of a dynamic link library `COM-DAFG.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 driver (see section "Driver Installation").

The user functions in the dynamic link library `COM-DAFG.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-DAFG.h`. If your compiler cannot create an import library from the dynamic link library `COM-DAFG.dll`, please link the library `COM-DAFG.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 Pascal calling conventions.

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

Functionality of the Software Interface

The software interface controls one communication channel for the data transfer from/to the device. The communication channel must be opened before starting the communication. The open procedure uses the initialization file `COM.ini` to determine the virtual port parameters. It configures the used virtual serial port according to these parameters and clears the port buffers. You may need to change the virtual port number in the file `COM.ini` according your system settings. To do this, open the file with any ASCII editor, change the settings, and save the file.

The 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-DAFG.dll` is unloaded from the system memory.

The return value of most interface functions contains a number describing the success of the particular operation. The return value is a signed 32-bit number (`int`). The last return value can be reloaded by

the function `COM_DAFG_State`. Table 5 summarizes the possible return values together with the error messages, which can also be obtained by the function `COM_DAFG_ErrorMessage`. If a data transfer failure has occurred, you can discover the reason by calling the functions `COM_DAFG_IO_State` and `COM_DAFG_IO_ErrorMessage`. The first one returns the last I/O error, the second one the corresponding error message (see Tab. 6.).

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

Tab. 5. Return values of the interface functions

Return value	Error message	Description
0	No error	The data transfer finished successfully.
-1	Error processing the COM.ini file	The initialization file either could not be opened or contains wrong entries. To obtain more details about the error, call the function <code>COM_DAFG_INI_ErrorMessage</code> .
-2	Error opening the port	The port could not be opened. For the possible reasons, see Tab. 6.
-3	Error closing the port	The port could not be closed. For the possible reasons, see Tab. 6.
-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. 6.
-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. 6.
-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. The communication with the device is possible only if it does not execute any process. Terminate all dialog boxes and menus at the device and retry the operation.
-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.

Tab. 6. 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	Port has already been opened	You attempted to open the communication channel that you have already opened.
3	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.
4	Cannot get the state of the port	The system could not get the state of the port.
5	Cannot set the state of the port	The system could not set the state of the port.
6	Cannot set the event mask for the port	The system could not set the event mask for the port.
7	Cannot set the timeouts for the port	The system could not set the timeouts for the port.
8	Cannot clear the port	The system could not clear the port buffers.
9	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.
10	Error writing data to the port	The system could not write data to the port.
11	Wrong data amount written to the port	The system could not write the proper data amount to the port.
12	Error setting the control lines of the port	The system could not set the state of the port control lines.
13	Error reading the status lines of the port	The system could not get the state of the port status lines.

Communication Control

Function COM_DAFG_Open

```
int pascal COM_DAFG_Open();
```

Opens the communication channel and returns an error code according to Tab. 5.

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

Function COM_DAFG_Close

```
int pascal COM_DAFG_Close();
```

Closes the communication channel and returns an error code according to Tab. 5.

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

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

Function COM_DAFG_Purge

```
int pascal COM_DAFG_Purge();
```

Clears the port data buffers and returns an error code according to Tab. 5.

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

The function COM_DAFG_Purge is automatically called by the function COM_DAFG_Open.

Function COM_DAFG_Buffer_State

```
int pascal COM_DAFG_Buffer_State  
(BOOL & empty);
```

Returns the state of the device's input data buffer in the variable `empty`. The return value is an error code according to Tab. 5.

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

Function COM_DAFG_Device_Purge

```
int pascal COM_DAFG_Device_Purge  
(BOOL & empty);
```

Clears the device's output data buffer and returns the state of the device's input data buffer in the variable `empty` like the function `COM_DAFG_Buffer_State`. The return value is an error code according to Tab. 5.

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

Waveform Management

Function COM_DAFG_DeleteWaveform

```
int pascal COM_DAFG_DeleteWaveform  
(WORD WaveformNumber);
```

Deletes the specified waveform and returns an error code according to Tab. 5.

The variable `WaveformNumber` specifies the waveform, its value must be between 0 and `WAVEFORM_NUMBER-1` (= 99). If the waveform does not exist, the function does not return any error.

Before calling this function, it is advisable to obtain the waveform parameters by calling the function `COM_DAFG_GetWaveform`. The success of the function can be also checked by the function `COM_DAFG_GetWaveform`.

Please note that the waveforms with numbers higher than `WaveformNumber` will be renumbered by this function - the numbers will be decremented to ensure a continuous waveform numbering. Further, the measurements using the specified waveform will become invalid since the waveform will not exist anymore.

Function COM_DAFG_CreateWaveform

```
int pascal COM_DAFG_CreateWaveform  
(WORD & WaveformNumber, WORD & Origin,  
WORD Size, double Periods,  
const char Name [WAVEFORM_NAME_SIZE+1]);
```

Creates a waveform with the specified number and returns an error code according to Tab. 5.

The `WaveformNumber` specifies the waveform number that should be created. The return value in this variable is the number of the waveform that has been created. For the allowed range of values for `WaveformNumber`, see the function `COM_DAFG_DeleteWaveform`. If a waveform with the specified number already exists, all waveforms with numbers higher or equal to this number will be renumbered to gain the space necessary for the new waveform. The new waveform will be inserted such that it becomes the specified number. Thus, the returned value equals to the value passed to the function. If the

waveform number points to the space, i.e. if the waveform with the specified number does not exist, the new waveform will be appended behind the last waveform. In this case, the number of the new waveform, i.e. the return value may differ from the specified one, that is from the value passed to the function. The easiest way to append a new waveform is to insert a waveform with the number `WAVEFORM_NUMBER-1` (= 99).

The variable `Size` specifies the size of the new waveform in samples. The value is equal to the real size decremented by one and it should not be zero. The waveform with the maximum allowable size of 64 KS (`MAX_WAVEFORM_SIZE` = 0x10000 = 65536) will be created with `Size` = 0xFFFF (65535).

The variable `Periods` specifies the number of signal periods within the waveform. This number is used when setting the device output timing to display the signal frequency. The value `Periods` does not have any direct influence on the waveform output; it is just an auxiliary number used for a user friendly frequency output. If the waveform consists of a combination of signals with several frequencies, the variable `Periods` should be set to the value that best characterizes the waveform.

The variable `Name` is a pointer to a null-terminated character string that specifies the waveform name. The maximum length of the waveform name is defined by the constant `WAVEFORM_NAME_SIZE` and equals 24 characters. The waveform name can be edited later in the device by a simple editor, however, to set it by calling the function `COM_DAFG_CreateWaveform` directly at the host computer is usually easier.

The variable `Origin` returns the address of the waveform data in the device memory. In most cases, you can ignore this value; it is in-

Tab. 7. Error codes of the function `COM_DAFG_CreateWaveform`.

Error Code	Value	Explanation
<code>COM_DAFG_CRT_ERR_WAVE_NUM</code>	0xFFFFE	no more free waveforms are available
<code>COM_DAFG_CRT_ERR_MEM_FULL</code>	0xFFFF	not enough memory space is available

tended for diagnostic purposes only. If the creation of the waveform fails, the variable `Origin` returns one of the error codes described in Tab. 7.

The function `COM_DAFG_CreateWaveform` creates the waveform only and does not setup its data. To set the waveform data, use the function `COM_DAFG_SetWaveformData`. Note: When appending a new waveform, ensure that you use the proper waveform number when calling the function `COM_DAFG_SetWaveformData`. You should use the return value in the variable `WaveformNumber` instead of the value originally passed in this variable to the function `COM_DAFG_CreateWaveform` (see above). In any case, it is advisable to repeatedly call the function `COM_DAFG_GetWaveform` in order to obtain the parameters of all valid waveforms before calling the abovementioned functions that manipulate the waveforms.

Function COM_DAFG_GetWaveform

```
int pascal COM_DAFG_GetWaveform
  (WORD WaveformNumber, WORD & Origin,
   WORD & Size, double & Periods,
   char Name [WAVEFORM_NAME_SIZE+1]);
```

Obtains the waveform parameters and returns an error code according to Tab. 5.

For the description of the parameters, see the function `COM_DAFG_CreateWaveform`. If the waveform does not exist, the variable `Size` returns zero. Since the waveform numbers are continuous, the first waveform number returning `Size=0` determines the origin of the free waveform space, i.e. there are no valid waveforms with higher numbers.

Function COM_DAFG_SetWaveformData

```
int pascal COM_DAFG_SetWaveformData
  (WORD WaveformNumber, WORD DataBase,
   WORD DataSize,
   WORD Data [MAX_WAVEFORM_BLOCK_SIZE]);
```

Sets the waveform data and returns an error code according to Tab. 5.

The `WaveformNumber` specifies the waveform number. For the allowed value range, see the function `COM_DAFG_DeleteWaveform`. The waveform with the specified number must already exist, otherwise an error occurs.

The variable `DataSize` specifies the number of samples that will be written. The maximum allowable value is given by the constant `MAX_WAVEFORM_BLOCK_SIZE` and equals 0x100 (256).

The variable `DataBase` specifies the offset of the data block with respect to the waveform origin. The offset is given in units of `MAX_WAVEFORM_BLOCK_SIZE` (0x100=256) samples.

The variable `Data` is a pointer to the buffer with the waveform data. The waveform data are unsigned 16-bit values (WORD), the data is coded as follows: The value `WAVEFORM_DATA_ZERO` (0x8000=32768) is the value corresponding to the zero output. The value `WAVEFORM_DATA_SCALE` (0x7FFF=32767) is the waveform data scale corresponding to the unity signal. Thus, a signal value x normalized to unity can be converted into 16-bit values using the expression $\text{Data} = \text{WAVEFORM_DATA_ZERO} + x * \text{WAVEFORM_DATA_SCALE}$.

Note: To set the data of the whole waveform larger than `MAX_WAVEFORM_BLOCK_SIZE` (0x100=256) samples the function `COM_DAFG_SetWaveformData` must be called repeatedly. To set the data of a waveform with 64 KS, 256 calls to the function `COM_DAFG_SetWaveformData` are necessary.

Function COM_DAFG_GetWaveformData

```
int pascal COM_DAFG_GetWaveformData
  (WORD WaveformNumber, WORD DataBase,
   WORD & DataSize,
   WORD Data [MAX_WAVEFORM_BLOCK_SIZE]);
```

Gets the waveform data and returns an error code according to Tab. 5.

For the description of the parameters, see the function `COM_DAFG_SetWaveformData`. The value returned in the variable `DataSize` equals the number of data samples that have been read. It equals 0 if the waveform does not exist or if the `DataBase` points behind the waveform data. The normal return value equals

MAX_WAVEFORM_BLOCK_SIZE (0x100=256), only for the last data segment, the value may be lower if the waveform size is not a multiple of MAX_WAVEFORM_BLOCK_SIZE (0x100=256).

Error Handling

Function COM_DAFG_State

```
int pascal COM_RFID_State();
```

Returns the state of the software interface according to Tab. 5.

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_DAFG_ErrorMessage

```
const char * pascal COM_DAFG_ErrorMessage();
```

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. 5.

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

Function COM_DAFG_IO_State

```
int pascal COM_DAFG_IO_State();
```

Returns the interface state of the serial port according to Tab. 6.

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_DAFG_IO_ErrorMessage

```
const char * pascal  
COM_DAFG_IO_ErrorMessage();
```

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. 6.

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

Function COM_DAFG_INI_ErrorMessage

```
const char * pascal  
COM_DAFG_INI_ErrorMessage();
```

Returns the error message corresponding to the COM.ini file error. The return value is a pointer to a null-terminated character string.

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

Various Functions

Function COM_DAFG_Version

```
WORD pascal COM_RFID_Version();
```

Returns the version of the software interface (the dynamic link library COM-DAFG.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-DAFG.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_DAFG_GetIniParams

```
void pascal COM_DAFG_GetIniParams  
  (char * Buffer);
```

Returns the parameters read from the COM.ini file. The parameter is a pointer to a buffer. The buffer must be large enough to store the null-terminated character string with the parameters.

Function COM_DAFG_GetUptime

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

Returns the device uptime in the variables `Seconds` and `Milliseconds`. The function return value is an error code according to Tab. 5.

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_DAFG_FW_Ver

```
int pascal COM_DAFG_FW_Ver (WORD & Version);
```

Returns the device's firmware version in the variable `Version`. The function return value is an error code according to Tab. 5.

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_DAFG_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_DAFG_FW_Date

```
int pascal COM_DAFG_FW_Date  
(char * DateString);
```

Returns the device's firmware date in the variable `DateString`. The function return value is an error code according to Tab. 5.

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_DAFG_Prod_No

```
int pascal COM_DAFG_Prod_No (DWORD & Number);
```

Returns the device's product number in the variable `Number`. The function return value is an error code according to Tab. 5.

Function COM_DAFG_Prod_ID

```
int pascal COM_DAFG_Prod_ID  
(char * Identification);
```

Returns the device's product identification in the variable `Identification`. The function return value is an error code according to Tab. 5.

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-DAFG.h"

int main()
{
    // check the DLL version:
    if ((COM_DAFG_Version() & 0xFF00) != 0x0100)
    {
        printf ("Wrong DLL Version: expected 1.x, found %d.%02d\n",
            COM_DAFG_Version() / 0x100, COM_DAFG_Version() & 0xFF);
        return -1;
    }

    // open the port:
    if (COM_DAFG_Open())
    {
        // handle a possible error:
        printf ("%s\n", COM_DAFG_ErrorMessage());
        return 1;
    }

    // list waveforms and save them:
    for (WORD WaveNum = 0; WaveNum < WAVEFORM_NUMBER; WaveNum++)
    {
        // get the waveform parameters:
        WORD Origin, Size; double Periods;
        char Name [WAVEFORM_NAME_SIZE+1];
        if (COM_DAFG_GetWaveform (WaveNum, Origin, Size,
            Periods, Name))
        {
            // handle a possible error:
            printf ("%s\n", COM_DAFG_ErrorMessage());
            return 1;
        }

        // is the waveform empty?
        if (!Size)
        {
            if (WaveNum == 0)
                printf ("No waveforms found\n");
            break;
        }

        // print the waveform parameters:
        printf ("Waveform number %2hd: Size: %5d, "
            "Periods: %8.5lf, Name: %s\n", WaveNum, 1+Size,
            Periods, Name);

        // generate the file name:
        char FileName [0x10];
        sprintf (FileName, "%02hd.txt", WaveNum);
    }
}

```

```

// open the file:
FILE * const DataFile = fopen (FileName, "wt");
if (DataFile == NULL)
{
    // handle a possible error:
    printf ("Error opening the data file %s\n", FileName);
    return -1;
}
printf ("Saving waveform into data file %s\n",
        WaveNum, FileName);
// loop for all data segments:
for (WORD Base = 0;
     Base <= Size/MAX_WAVEFORM_BLOCK_SIZE;
     Base++)
{
    WORD DataSize; WORD Data [MAX_WAVEFORM_BLOCK_SIZE];
    // get a data segment:
    if (COM_DAFG_GetWaveformData (WaveNum, Base,
                                   DataSize, Data))
        // on error, retry the operation:
        if (COM_DAFG_GetWaveformData (WaveNum, Base,
                                       DataSize, Data))
        {
            // handle a possible error:
            printf ("%s\n", COM_DAFG_ErrorMessage());
            fclose (DataFile);
            return -1;
        }
    // loop for all data in the segment:
    for (int n = 0; n < DataSize; n++)
        // output the data:
        if (fprintf (DataFile, "%.6lf\n",
                    double (Data[n]-WAVEFORM_DATA_ZERO) /
                    WAVEFORM_DATA_SCALE) == EOF)
        {
            // handle a possible error:
            printf ("Error writing the data file %s\n",
                    FileName);
            return -1;
        }
    }
// close the file:
if (fclose (DataFile) == EOF)
{
    printf ("Error closing the data file %s\n",
            FileName);
}
}

//finish the program:
return 0;
}

```