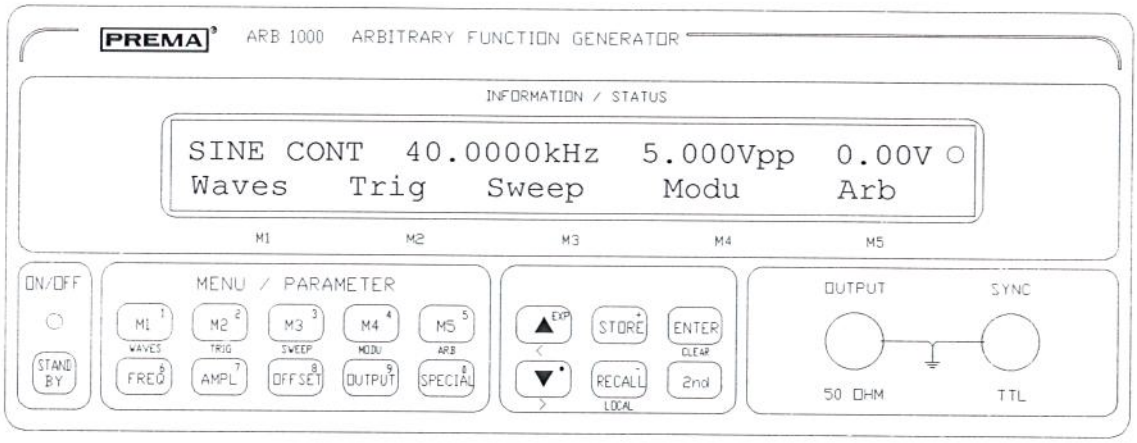
PREMA®

Arbitrary Function Generator ARB 1000

Creative Signal Generation with IEEE-488 and RS232 Interface



ARB 1000

Preliminary User´s Manual

PREMA Semiconductor GmbH
Robert-Bosch-Str. 6
55129 Mainz
Tel: ++49 (0) 6131- 5062- 0
Fax: ++49 (0) 6131- 5062- 220
E-Mail: prema@prema.com
Internet: www.prema.com

PREMA Präzisionselektronik GmbH
Robert-Bosch-Str. 6 • D-55129 Mainz • Germany
Tel. +49-6131-5062-20 • Fax. +49-6131-5062-22
E-Mail: instruments @ prema.com
Internet: http: / / www . prema.com

AS1000-9727
Subject to change without notice

Table of Contents

1 Introduction

PREMA Instruments sets new standards in the arbitrary function generator market, introducing the ARB 1000 arbitrary function generator. This state-of-the-art generator uses the newest techniques like Direct Digital Synthesis and high speed computer programming and incorporates these in a very powerful and low-cost multi purpose waveform generating system.

With this instrument PREMA has created a very powerful generator and which still keeps its famous ease of use.
A lot of standard signal are already preprogrammed and the arbitrary mode gives the user full flexibility when creating its own waveforms. High quality and spectral pure signals are designed in by using high resolution DDS techniques and high quality amplifiers.

We think the ARB 1000 sets the new standard in Arbitrary Function Generation which you where looking for and we are sure you will enjoy working with the PREMA ARB 1000.

1.1 Important Safety Instructions

Reading the User Manual

Proper working procedure with this instrument is possible only after reading all instructions, hints and procedure specifications attentively and understanding them.

Please get in touch with PREMA before commencing operation of the instrument if you do not understand something in the user manual or the instructions, procedural descriptions and safety regulations are unclear.

This user manual has been written to make the instrument understandable for operation in the manner intended. It contains important instructions for safe, correct and efficient operation of the instrument.
Dangers are avoided, repair costs and downtime reduced and the service life of the instrument is extended only when these instructions are observed. The user manual should always be available at the place where the instrument is operated.
Incorrect manual control or failure to observe the instructions given here may endanger persons (also third parties) or cause material damage.

Personnel entrusted with operating this instrument must have read this user manual attentively and must be familiar with all safety instructions.
In addition to the instructions given in this user manual, the local regulations for preventing accidents in force at the operating site apply, as well as the relevant rules for safe and proper working procedure.

Further Safety Instructions

Further safety instructions are contained in the chapter headed "Getting Started". Explanations and instructions are given there for the warning signs and symbols on the instrument for recognising specific sources of danger. It is essential to observe and comply with all safety instructions. The warning symbols must be held complete and in good readable condition.

Predictability of Dangers

The manufacturer cannot anticipate every conceivable danger.
If a task is not carried out in the manner recommended, the operator must make sure that this does not entail any danger for himself and other persons.

He should also make sure that the instrument cannot be damaged or endangered by the chosen manner of operation.
This user operating manual is not an instruction manual for making repairs.
The instrument should be returned to the factory for any necessary repairs.

Proprietary Rights

This user manual is protected by proprietary rights. No part thereof may be copied, reproduced or distributed in any form without prior written permission.

Conformity Declaration

PREMA has issued an EC conformity declaration for this instrument. This declaration certifies that the instrument complies with the relevant requirements of the EC directives.

Proper Utilization as intended

These instruments have been built conforming to the recognized technical safety principles, but nevertheless if it is not used and operated in the manner intended, dangers may arise for body and life of the user or third persons, or damage may be incurred by the instrument and other objects.
The instrument may therefore be operated only in technically perfect condition, in the manner intended and with due awareness of safety considerations and dangers, observing the contents of the user manual and the regulations for the prevention of accidents. It should be used exclusively for the tasks described in this user manual.

All faults on the instruments which impair the safety of the user or third persons must be remedied immediately.
PREMA accepts no liability for damage resulting from utilization of this instrument in any manner other than the intended manner described in the user manual. The user alone carries the risk and responsibility for any deviating utilization of the instrument.

Availability of the User Manual

The user manual must always be available at the place where this instrument is operated. The personnel entrusted with operation of this instrument must be familiar with all task procedures described in the user manual and with all safety instructions.
All warning and safety instructions attached to the instrument must be held complete and in clearly readable condition.
No modifications, attachments or conversions of the instruments are permitted without consent and approval by PREMA, otherwise the conformity becomes void.

2 Getting started

2.1 Delivery

Every PREMA unit is thoroughly and carefully checked before it is shipped, to ensure that it is in flawless condition, and that its technical characteristics are within specifications.
Consequently, upon receipt, the unit should be in perfect condition, mechanically and electrically.

To make sure that the unit has not been damaged during transport, it should be thoroughly checked out immediately after receipt. If damage is detected, a damage claims form should be completed with the shipping carrier.

Please use the following list to assure that delivery is complete:

☞
1. Power Cable
2. User Manual, English
3. Certificate of Quality
4. Product Registration Card, which you should fill out and mail back to PREMA
5. Any optional equipment ordered

Please ensure also, that the unit is set up for the right AC Voltage, with the right type of fuse (see chapter "Connecting the Unit to Main Power").

Important: Do not throw the box and packaging materials away!
If the unit has to be sent back to the factory for recalibration or repair, only the original packaging materials will provide sufficient protection against damage.

2.2 Safety Guidelines

Also refer to the safety guidelines in the "Introduction" chapter, please.

The function generator may only be operated if it is in perfect and safe condition. Accident prevention and environmental protection rules must also be followed. All power-up and power-down procedures described next must be followed. Problems, such as loose connections, damaged or scorched cables, oxidized contacts, and damaged fuses must be immediately removed by a professional.
A safe and ecologically sound disposal of operating and support materials, as well as replacement parts, must be arranged. Only genuine replacement parts shall be used. Otherwise, the manufacturer's warranty and the function generator's conformity will be voided.

Any changes to the function generator, which cause any functional changes, may only be carried out by the manufacturer, or after discussion with and permission by the manufacturer.

This manual contains information and warnings which must be observed by the user to ensure safe operation and retain the apparatus in a safe condition. The instrument has been designed for indoor use within the specified limits of temperature, i.e. 10 to 45 °C. It should not be switched on if there are obvious signs of mechanical damage and it should not be used under wet conditions.

Utilization

The function generator may only be utilized for the functions that are described in the Technical Specifications. It is especially important to adhere to the load limits of the output connectors. PREMA accepts no responsibility for any damage arising from improper operation.

2.3 Safety Symbols

The signs and symbols on the function generator, which provide guidelines for safety and handling, are displayed and described below.

This symbol advises the user of a possible danger area.
Please consult the manual .

The CE mark means, that the manufacturer has issued an EC Declaration of Conformity for this function generator. This declaration certifies, that this function generator conforms to the pertinent requirements of EC directives.

2.4 Accident Prevention

While using this instrument, precautions to prevent an accident should be taken, appropriate to the use of a function generator.

The instrument is safe to operate with covers fitted and these must not be removed under normal usage. The covers protect the user from live parts and they should be removed only by suitable qualified personnel for maintenance and repair purposes.

2.5 Grounding

In order to protect the user, the unit's case is grounded through the grounding lead of the power cable. This is connected to the instrument before the line and neutral supply connections when the supply plug is inserted into the socket on the back of the instrument. If the final connection between this and the supply is made elsewhere, the user must ensure the ground connection is made before line and neutral.
To ensure proper grounding, the power cable should always be connected to a properly grounded power connector.
The unit case is galvanically separated from the output connectors and interface ports.

If any supply cable other than that supplied with the instrument is used, it must carry an adequate protective ground conductor.

Any interruption of the protective ground conductor inside or outside the instrument is likely to make the instrument dangerous. Intentional interruption is prohibited.

Signal connections into the instrument should be connected after and disconnected before the protective ground connection is made, i.e. the supply lead must be connected at all times that signal leads are connected.

The back of the unit is equipped with a grounded screw, identified by the ⏚ symbol, where the user can connect a separate ground line (rack mounting bracket).

2.6 Connecting the Unit to Main Power

This PREMA measurement unit is designed to be connected to AC Main Voltage, at a frequency of 50 Hz or 60 Hz. The rear panel of the unit is equipped with a standard DIN grounded power connector.

Before connecting the unit to power, you should make sure that it is set to the right voltage (indicator and fuse).

The voltage selection switch with integrated fuse is located right under the power connector, where you can also read off the current voltage setting; a setting of "220V" represents an AC voltage from 220V to 240V, "110V" represents a voltage from 100V to 120V.

Switching the AC Voltage is done as follows:

☞
1. Unplug the unit.
2. The clamp for the fuse is located between the plug and the power selector and must be removed. For a setting of "110V" you will need a fuse rated at 2A; for "220V" you'll need a fuse rated at 1 A.
3. Place the necessary fuse in the clamp and push the clamp back in.
4. Turn the cylinder with the voltage indicators once left or once right to the desired setting, so that the voltage that is currently set is indicated by the white arrow on top.
   The indicators are used as follows:

Setting	Voltage Range	Fuse
110 V	90 V_{RMS} to 130 V_{RMS}	2A Slow Blow
220 V	180 V_{RMS} to 265 $_{RMS}$	1A Slow Blow

Table: Main Voltage Ranges and fuses

It will operate at supply frequencies of between 47Hz minimum and 63Hz maximum.

2.7 Ventilation

The instrument relies on convection cooling and must not be operated in a position which restricts air flow through the ventilation slots in the front and the back of the instrument.

2.8 Operating Temperatures

The instrument is designed to be operated in an environment having an ambient temperature of between 10° C and 45°. C. The instrument is guaranteed to operate with full accuracy within a temperature range of 23° C ± 5° C.

2.9 Warranty

PREMA warrants the reliable function of the unit for a period of two years from the date of delivery.
Repairs that need to be carried out during the warranty period are not billed to you.

Damage caused by inappropriate use of the unit, or by surpassing specified limits, does not fall under PREMA's warranty obligations.
Please be aware, also, that PREMA will not be held liable for damages, incidental or coincidental, associated with the use of this Function Generator.

2.10 Certificate

Each Function Generator ARB 1000 is provided with a certificate of quality at the factory, certifying that all mentioned specifications are guaranteed.
Please look for this certificate at delivery time.

2.11 Turning it on

The function generator can be switched on with the STANDBY KEY after connecting the power cable.
Switch off the instrument also with the STANDBY KEY.
The 5017 is then in standby mode. The red LED at the bottom left of the front panel lights up.

Note: The transformer is not disconnected from Mains Power in standby mode.
Please, do not disconnect the power cable before having pressed the STANDBY KEY.

2.12 Rack Mounting

A rack adapter with two height units is offered for the ARB 1000. The unit has a width of one-half 19-inch, so it can be combined with another half-19-inch unit. More information about rack mounting can be found in the Chapter "Accessories."

When installing the unit into a 19-inch rack, you should take into account, that the ventilation openings in the back are not covered up. In addition, it should be possible to cut power to the unit in an emergency, through the use of an EMERGENCY OFF switch somewhere nearby.

3 General Information

This section of the manual is aimed at the first time user of the ARB 1000. It describes the basic principles of the ARB 1000, the front-panel lay-out and the connectors of the instrument and how to use them.
The best way to learn how to use the instrument is by reading this part of the manual and after that pushing all the buttons and viewing the result on an oscilloscope.

3.1 Basic Principle of arbitrary Waveform Generation

Arbitrary waveform generation is based on a quite simple principle. Digital data, representing the waveform, is stored in a Random Access Memory (RAM). Converting each data point sequentially to an analog waveform by using an analog to digital converter, the basic waveform is generated. This waveform is amplified and/or filtered by several analog amplifiers. The control logic takes care of the more difficult tasks like calculating the waveform data and programming it in memory, triggering, gating or generating bursts of waveforms and the access to the user interface.

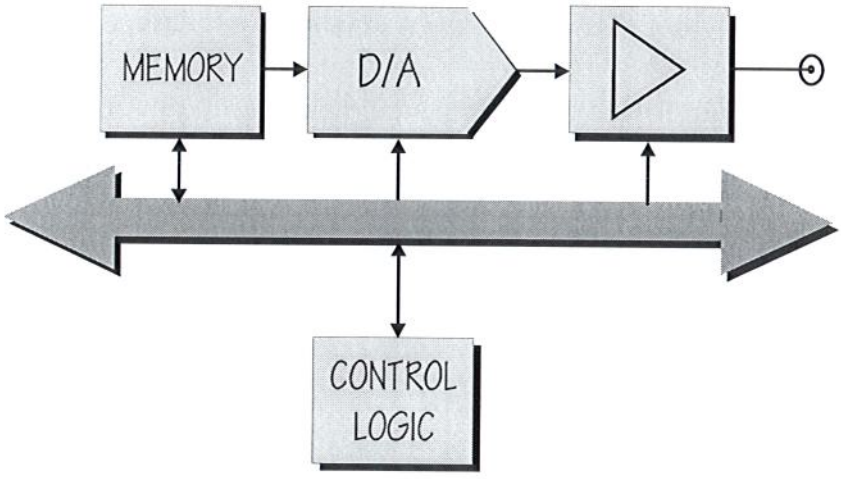


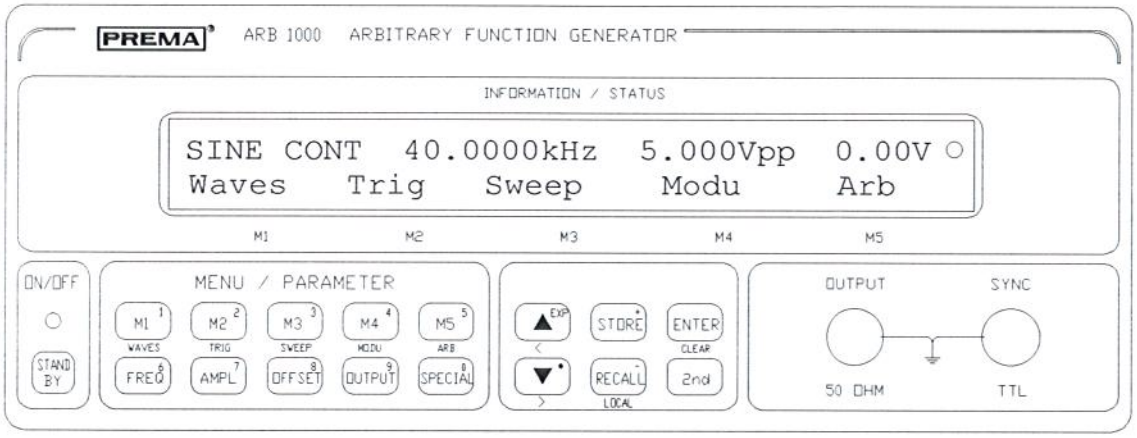
Fig: Basic principle of an arbitrary function generator.

As you can see, the arbitrary waveform generating part (Memory, D/A-conversion and amplification) is a quite direct approach and defines the physical specifications of an ARB. However, the control logic defines the ease of use, the practical application e.g. the real possibilities of the instrument.

3.2 Mains power

Before the instrument can be operated, the mains power connection at the rear of the instrument has to be connected to a mains power socket. Now, the instrument is supplied with mains power and is in stand-by mode. The ON/OFF switch at the front of the instrument can be used to switch the instrument on and off.

3.3 The display



The 2-line display informs you concerning menu's, settings and status of the instrument.
Following formats and rules apply to displayed text:

- **Information, status and current settings are displayed in the upper line of the display called 'information line'.**
  Pressing Δ or ∇ allows you to scroll through several information lines.
  The following information can be selected to be displayed in this line:
  - The name of the currently selected menu.
  - Currently set waveform and parameter settings in the following format:

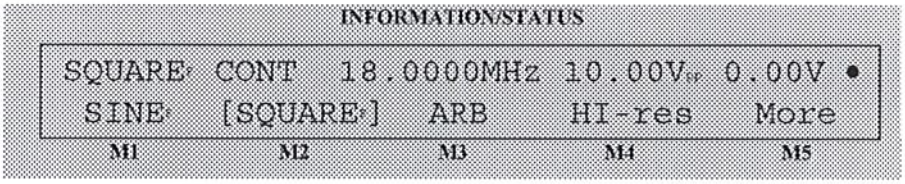
    WAVEFORM | OPERATING MODE | FREQUENCY | AMPLITUDE | OFFSET

  - Optional parameters and parameter settings for particular waveforms.
  - An arrow down (↓) as most right character indicates there is more information available by pressing Δ or ∇ .

- A filled bullet (●) as most right character indicates the output is switched on, an open bullet (○) indicates the output is switched off.

• **Up to 5 selectable menu-items are displayed in the bottom line of the display.**
- These items can be selected by pressing the M1 to M5 keys. When optional parameter information is required for a selection, queries for this information are displayed in the bottom line of the display.
- **Square brackets [ ]** around a menu-item indicate that this item is currently selected.
- **Triangular brackets < >** around a menu-item indicate that this item is currently selected and that particular parameters for this selection can be set by selecting this item again.
- Items in lowercase characters indicate a sub-menu.
- Items in uppercase characters indicate selectable items in this menu.

Example:
Following is displayed:



At the moment a continuous square-wave (fast frequency mode) of 10Vpp with a frequency of 18 MHz is generated. SINE, SQUARE and ARB can be selected directly. Selecting HI-Res and More will display a sub-menu.

3.4 The Keyboard

The keyboard allows the user to access the instrument and uses a very convenient keyboard handling system which allows you to select and use all possibilities of the instrument with just seventeen different keys.

The ON/OFF key.

The ON/OFF key allows the user to switch the instrument on and off while it is in 'Stand-by' mode. Before unplugging the instrument by its powercable, first switch the instrument to stand-by (off) by using the ON/OFF key, so the internal processor can take care of saving all the instruments settings and waveforms.

The instrument control keys.

The control keys of the instrument consist of ten menu/parameter keys and six multi-purpose keys.
The menu/parameter keys perform several functions. First, they can be used to select menus. Secondly, they can be used to select menu-items and last but not least, they can be used to enter parameter values.
The six multi-purpose keys at the right can be used to store and recall settings, increase and decrease values, select value-digits, scroll through menu's and to select the menu-options of the menu/parameter keys.

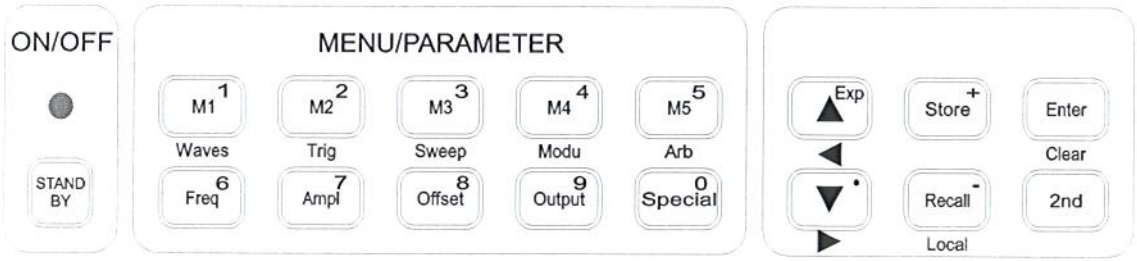


Fig. The keyboard

The layout of the keyboard is chosen in such way that a minimum of key operations is needed to perform a selection or operation.

Selecting menus and parameters

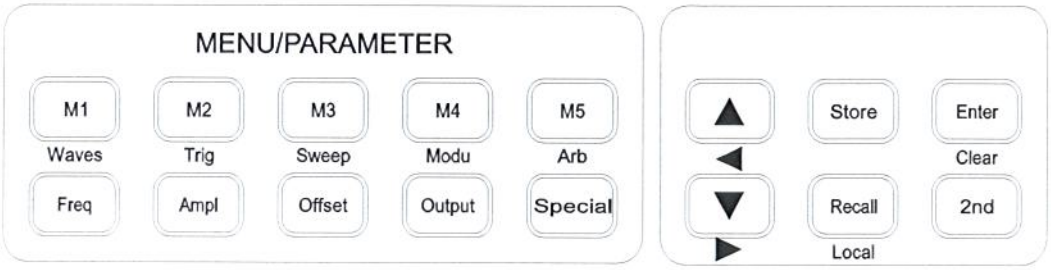


Fig. The Keyboard

The following key function are available after power-up. This is also the most common key status of the instrument.

Selecting menu items: M1 ... M5.

- Use these keys to select a menu option displayed in the bottom display line.
- When a "More"-item is displayed at the right of the bottom display line, more menu-items are available and can be selected by this item.
- When a menu-item is displayed in upper-case characters (like SINE, SQUARE and ARB), it can be selected directly. When a menu-item is displayed in lower-case characters (like HI-Res and Pulses) a new menu will be displayed when the item is selected.

Selecting parameters:

Freq (Frequency input).

- When this key is pressed you will be prompted to enter the frequency parameter. See entering parameter values.

Ampl (Amplitude input).

- When this key is pressed you will be prompted to enter the amplitude parameter. See entering parameter values.

Offset (Offset input).

- When this key is pressed you will be prompted to enter the offset parameter. See entering parameter values.

Selecting menus:

2nd Waves (Waveform selection)

- Use this combination of keys to select the waveform selection menu.

2nd Trig (Trigger mode selection)

- Use this combination of keys to select the trigger mode selection menu.

2nd Sweep (Sweep selection)

- Use this combination of keys to select the sweep selection menu.

2nd Modu (Modulation selection)

- Use this combination of keys to select the modulation selection menu.

2nd Arb (Arbitrary options selection)

- Use this combination of keys to select the arbitrary options selection menu.

Output (Output options selection)

- Use this key to select the output options menu.

Special (Special options selection)

- Use this key to select the special options menu.

Enter (Returning to previous menu level)

- During menu-item selection, the enter-key can be used to go back to the previous menu-level when you are more than one level deep.
  When the **2nd** key is selected by mistake, you can return to the previous input level by again using the **2nd** key.

Entering Parameter Values

The following key functions are available when you are prompted to enter a value.

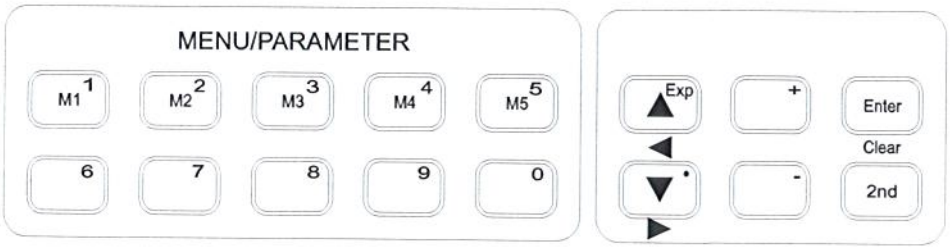


Fig. Available key funtions entering a value

Entering values:

0 ... 9 Exp . + -

Use these keys to enter values. Values can be entered using standard decimal notation or exponential notation. When entering a value, this value must start with a number or + or -.

Following values are valid to enter:

0 . 0 0 1 1 Exp - 3 - 2 . 5 Exp 1

Press the Enter key to accept the value and return to the menu-item selection. When a mistake has occurred entering a value, the combination 2nd < can be used to delete the last entered digit (Backspace) and the combination 2nd CLR can be used to abort the whole entry and display the previous value.

Increasing/Decreasing:

Δ ∇ > <

The arrow-keys can be used to increase and decrease digits when entering values.

To increase a digits value press the Δ key when you are entering a value.

To decrease a digits value press the ∇ key when you are entering a value.

To select a digit use the < and > keys.

When using this method the value of a parameter will change immediately without using the Enter key.

Selecting optional menu-items:

During parameter value input, optional menu-items are available. These items can be selected by using the 2nd key in combination with one of the five menu-item selection keys M1 ... M5.

Specific Functions

The following key function are specific and always available (except during value entry).

Store and Recall

Store

When this key is used, you will enter the store settings menu.
You can select either to store your current instrument settings into the POWER-ON settings memory or into one of nine USER settings.

Recall

When this key is used, you will enter the recall settings menu.
You can recall the information from a USER settings memory in which you previously stored an instrument setting, the POWERON settings memory in which the instrument settings at power on are stored, a totall RESET settings memory in which standard settings are stored or FACTORY settings.

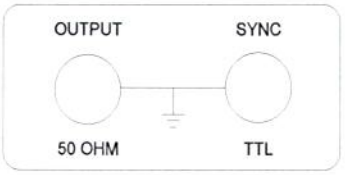
Back to local

2nd + Local

When the instrument is remote operated and needs to be switched back to local manually, the Local key needs to be used. Remote operations will be aborted and operation by keyboard will be possible again.

3.5 Front panel connectors

The following two output connectors are situated on the frontpanel and have the following functions:



OUTPUT

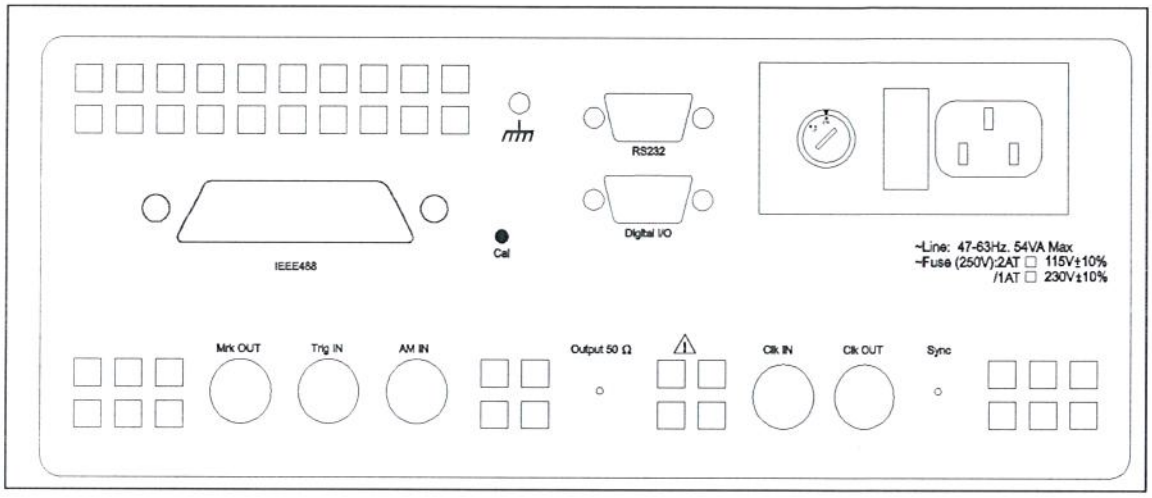
Matched signal output.
The instrument has been designed and calibrated with a high quality coaxial cable to operate as a closely matched system when terminated into a 50Ω load. For equipment with an input impedance much greater than 50Ω an accurate and adequate termination should be connected between the coaxial cable and the equipment.
When the instrument is operating into an impedance higher than the output impedance, up to twice the maximum terminated output amplitude can be obtained. Under open circuit conditions, the actual output amplitude will be two times the amplitude indicated by the front panel indication.
When operating, the instrument always considers the total load impedance and its effect on the output amplitude.

SYNC

TTL synchronisation output.
The SYNC-output can drive up to 10 TTL standard loads.
When using the instrument in an arbitrary mode the SYNC-output will be synchronised to the first address of the arbitrary wave.

3.6 Rear Panel

At the back of the instrument you can find the following connectors and switches:



Input Voltage

Power cord connector for AC-mains voltage
Connect the AC-mains voltage to this connector. Before connecting the power cord, be sure to check if the mains voltage is adequate for the AC-voltage of the instrument (see printing on the back of the instrument). If not, contact your supplier.
Upon changing fuses, please check the proper fuse configuration (see Chapter 'Getting Started') .

IEEE-488

IEEE-488 interface connector
The IEEE-488 interface connector can be used for connections via the IEEE-488 interface bus to external devices like computers and digital storage oscilloscopes. Complete control over the instrument is possible via this bus. Data transfer via the IEEE-488 interface is fast and reliable.

RS232

RS232 interface connector
The RS-232 interface connector is a standard sub-D connection to the RS-232 interface bus. Complete control over the instrument is possible via this bus. Speed and frame-settings can be installed using the RS-232 installation menu.

Digital I/O

Not used

Cal

Cal switch
Operating this switch will cause an automatic internal calibration of the instrument.

Mrk-OUT

TTL Marker output
The MRK-output will give a TTL pulse when, during a frequency sweep, the output frequency matches one of the frequencies set with the marker option in the sweep-markers-menu .

Trig IN

TTL Trigger input
By connecting a TTL pulse to the trigger input the generator can be synchronised with this pulse when in triggered, burst or gated mode.

AM-IN

Amplitude modulation input
Connecting a voltage in a specified range the input results in an amplitude modulation of 0 % up to 100 % . (See also chapter 'Technical Specifications').

Clk-IN

Arbitrary clock input
This TTL- input can be used to connect an external arbitrary sample-clock to the instrument. To use this input the instrument needs to be set to the external clock mode. To do this you must switch the clock to EXTernal in the ARBitrary Clock menu.

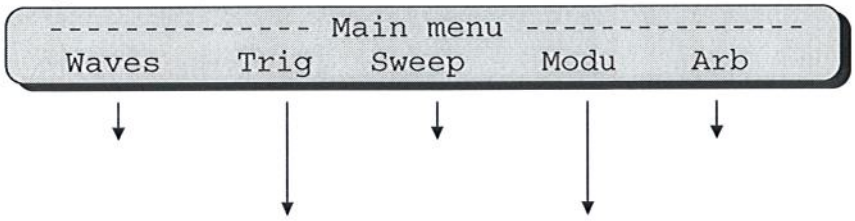
Clk-OUT

Clock output
BNC-connector for TTL-output of the arbitrary sample clock signal.

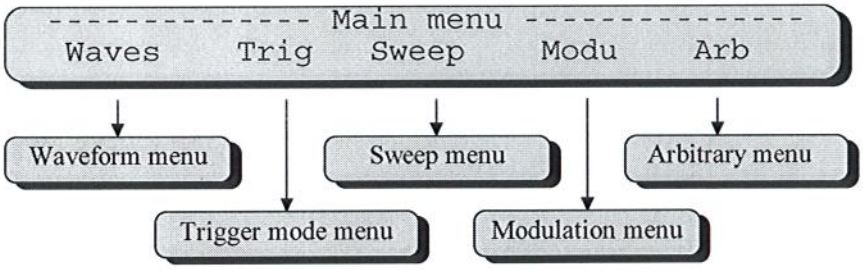
4 Manual Operation

Menu's in this manual are displayed as a diagram. Following conventions apply to these menu-diagrams:

- Menu displays are presented as a multiple line display.
- More than two lines in the presented display indicate multiple line-display possibilities.
- A "down-arrow"(↓) in the upper menu-line indicates that you can scroll through multiple menu information lines by pressing corresponding up/down keys.



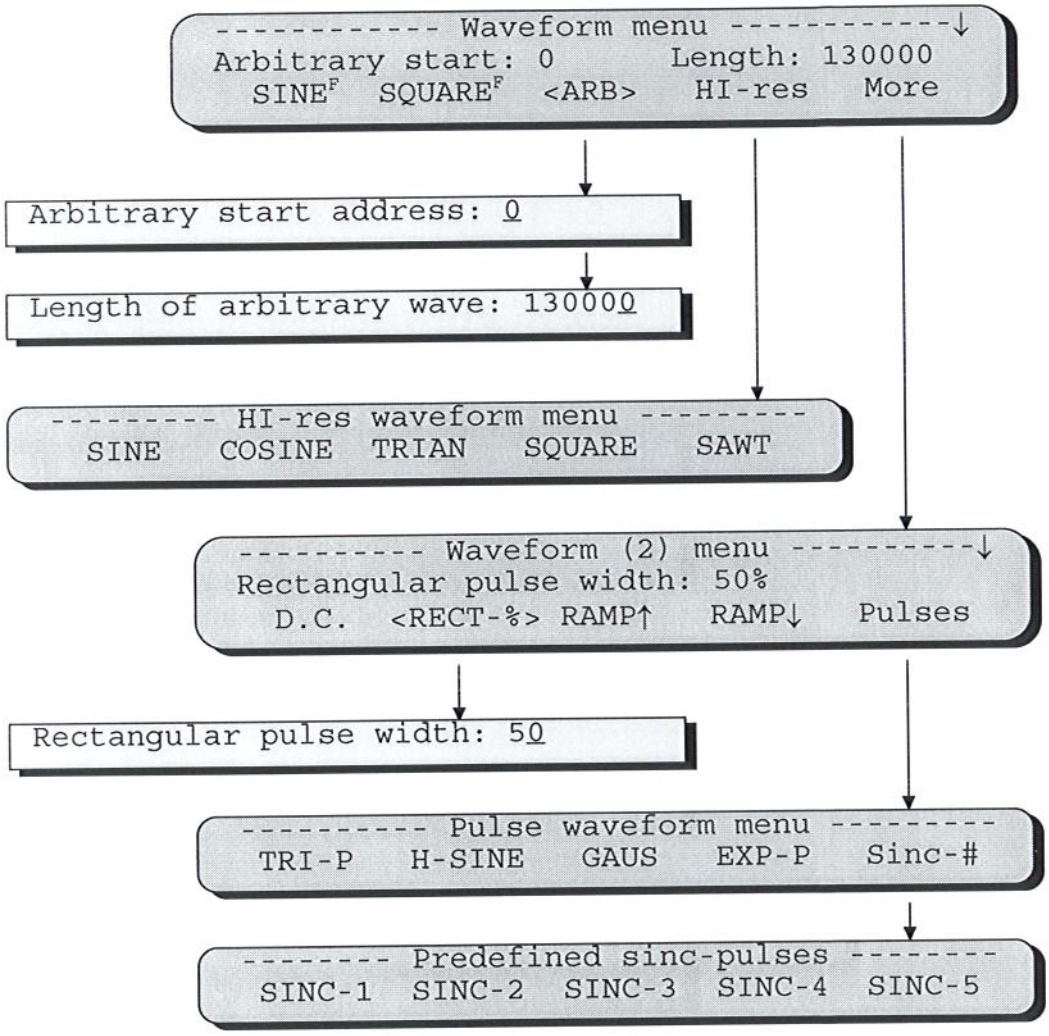
4.1 Main Menu



The **Main** menu is the basis to go to the whole menu structure. From here, the five function menu's waveform, trigger-mode, sweep, modulation and arbitrary are accessed. Pressing one of the menu-keys M1 to M5 gives access to the function menu's.

4.2 Waveform Menu

The waveform menu can be entered by pressing **2nd** and **WAVES** key. The bottom line of the display gives you the menu-options which you can select . Following diagram displays the entire **waveform** menu structure.



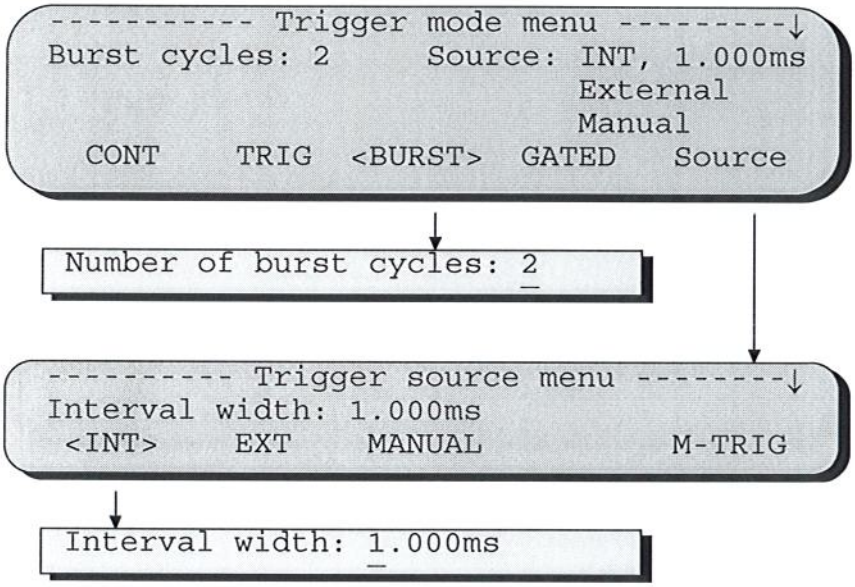
The sine and square waveforms can be generated either in fast mode (ᶠ) or in HI-res mode. Fast mode can be used in the frequency range of 100mHz to 20MHz and with a resolution of 100mHz.
HI-res mode has a higher resolution (100 µHz) but is only applicable for lower frequencies (max. 40kHz). Frequency ranges are available in the specifications chapter of this manual.

The **HI-res-** option gives access to following submenu, displaying the waveforms which can be generated in hi-resolution mode.
These waveforms are programmed automatically in the arbitrary waveform memory when they are selected.
When e.g. a rectangular pulse **(RECT-%)** waveform is selected the generator will generate the waveform with the previously set parameter for the pulse-width. The selected item, in this case, will then be displayed in triangular **(<RECT-%>)** brackets, indicating that a parameter can be set when this menu-key is pressed again. Pressing this menu-key will cause the generator to query you for the rectangular pulse width. The pulse-width can now be entered and will be generated after pressing the **ENTER**-key.

When an **ARB-** wave is selected the generator will query for the start address and the wavelength of your arbitrary waveform in the arbitrary waveform memory. The start-address can be selected between 0 and 129999, the wavelength between 2 and 130000 points. After the wavelength is entered the generator will switch to arbitrary waveform generation. The frequency displayed in the frequency field will now be changed to the sample rate.

4.3 Trigger Menu

In the trigger-menu several trigger modes of the generator can be selected .



Continuous

The continuous mode can be selected by pressing the **CONT**- option. When continuous is selected the output will give a continuous signal at the currently programmed frequency, amplitude and offset.

Triggered

The triggered mode can be selected by pressing the **TRIG**- option. When the triggered mode is selected the output will be quiescent until the generator is triggered, then it will generate one cycle of the programmed waveform at the currently programmed frequency, amplitude and offset.
Triggering can be done externally, manually or internally (see Trigger source).

Burst

The burst mode can be selected by pressing the **BURST**- option. When the burst mode is selected the generator will ask to enter the number of cycles (N) which have to be generated. The number of cycles must be between 1 and 65535. After entering the number of cycles the generator will generate N cycles of the currently selected waveform when a trigger pulse occurs.
Triggering can be done externally, manually or internally (see Trigger source).

Gated

The gated mode can be selected by pressing the **GATED**- option. When the gated mode is selected the selected waveform will be generated at the currently selected frequency for the duration of the trigger signal (during trigger input = high). The last cycle started is completed when the generated signal (sine, triangle or square) of the waveform menu is selected. When a digitally generated signal (pulse, ramp or arbitrary) is selected the signal will be sampled at the currently selected frequency during trigger signal = high.
Triggering can be done externally, manually or internally (see Trigger source).

Trigger Source

By pressing the **Source** option the trigger source selection menu can be entered. In this menu the trigger source can be selected.

INT

When the **INT**- option is selected the generator queries you to enter the required trigger rate (selectable between 1.000 us. and 1000 s. with a step rate of 1.000 us.). After entering the trigger rate the internal trigger generator will trigger the generator at the selected trigger rate.

EXT

When the **EXT**- option is selected the generator can be triggered via an external signal source (TrigIn-input, trigger at positive edge).

MANUAL

When the **MANUAL**- option is selected the generator can be triggered by operating the M5 (M-TRIG) key.

M-TRIG

This key is used to activate a manual trigger.

4.4 Sweep Menu

The sweep menu makes it possible to enter all essential parameters to turn your generator into a powerful sweep generator. The generator can sweep linear, logarithmic and even on an arbitrary wave. You can also set up to 4 marker frequencies to be indicated by a TTL-pulse at the marker output.

A sweep is generated by the internal processor of the generator and is build up out of 2 to 1.000.000 steps. The time during which a step (frequency-point) is generated can be set. Using more steps allows a high resolution while using less steps for a sweep enables higher sweep speed.
Following settings and selections are applicable to the **SWEEP**-menu:

Range

The sweep **range** menu allows you to set the sweep **START** frequency, the sweep **STOP** frequency, the step- **TIME** and the number of **STEPS**. The type of sweep (linear, logarithmic or arbitrary) can be selected by pressing the **Spacing** option (see **Spacing**).

Spacing

The **Spacing** menu allows you to set the type of sweep which is used. Following type can be selected:

LINear:

A linear sweep is calculated as a straight linear frequency lapse starting at the set start-frequency and stopping at the set stop-frequency. The time for the sweep is the set step-time times the set number of step.

LOGarithmic

A linear sweep is calculated as a logarithmic frequency lapse starting at the set start-frequency and stopping at the set stop-frequency. The time for the sweep is the set step-time times the set number of step.

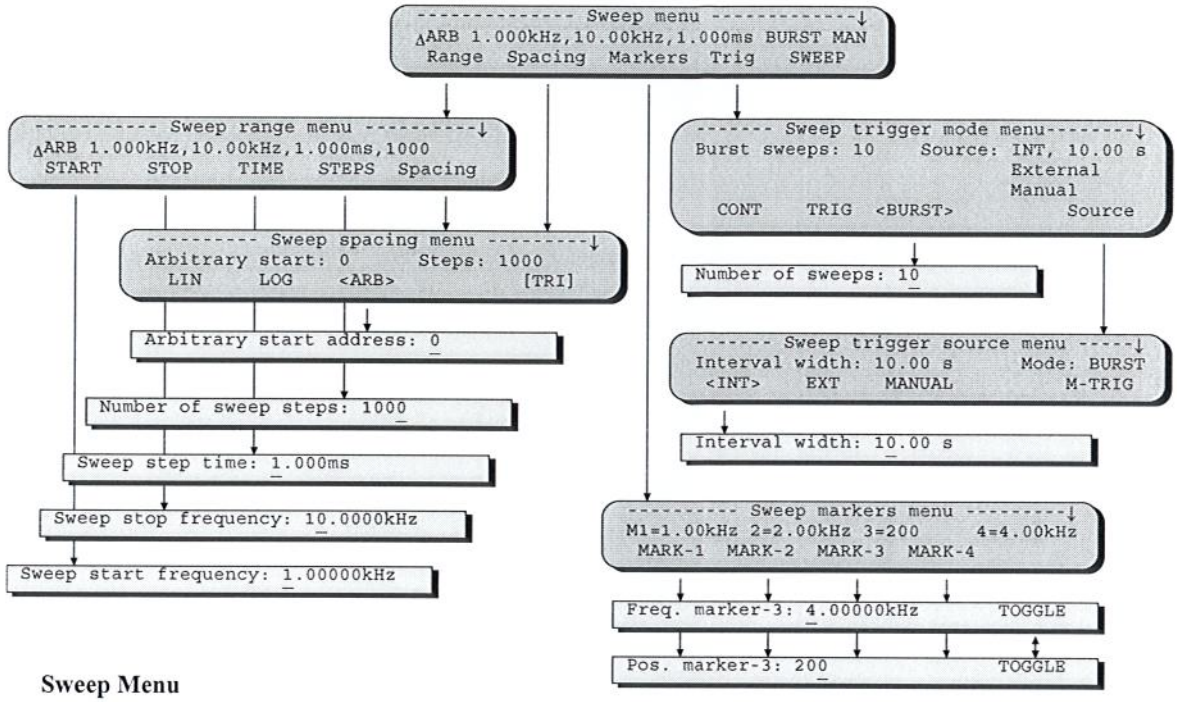
ARBitrary

An arbitrary sweep is calculated as an arbitrary frequency lapse which is defined by the data in the arbitrary data memory. The set start-frequency equals an arbitrary data point with the value of -2048 while the set stop-frequency equals an arbitrary point of +2047. The time for the sweep is the set step-time times the set number of step. The form of the arbitrary sweep is given by the currently selected data in the arbitrary data memory. The start-address for the data in the arbitrary memory can be selected by pressing the **ARB**-option while selected. The number of points used from the arbitrary memory is defined by the number of steps currently set in the sweep-settings.

TRIangular

All three sweeping modes mentioned above can be set in triangular sweep mode. In this mode the sweep will be started at the start-frequency going to the stop-frequency, then continuing to the start frequency again, and so on.
This means that each second sweep, start- and stop-frequency are exchanged.

However, the start- and stop-frequency points will only be generated once at the switching points.



Sweep Menu

Markers

You can set up to four sweep markers with the **Markers** menu. A set marker will cause the Mrk-output at the back of the instrument to go high (TTL) during the sweep-step(s) which the marker value is valid. Each of the four markers **MARK-1** to **MARK-4** can be set in two ways. To toggle between the two options **2nd M5** must be pressed.

Frequency marker

The marker is valid each time the sweep reaches the frequency set by this marker.

Position marker

The marker is valid at the step position set relative to the start of the sweep.

Trig

The **Trig** option enables the sweep trigger menu. This is a part of the trigger menu explained in the chapter THE TRIGGER MENU. The sweep does not have the **Gated** trigger mode. All settings here are only applicable to the currently set sweep.

SWEEP

Select **SWEEP** to easily switch the sweep mode on and off, using the current sweep settings.

4.5 Modulation Menu

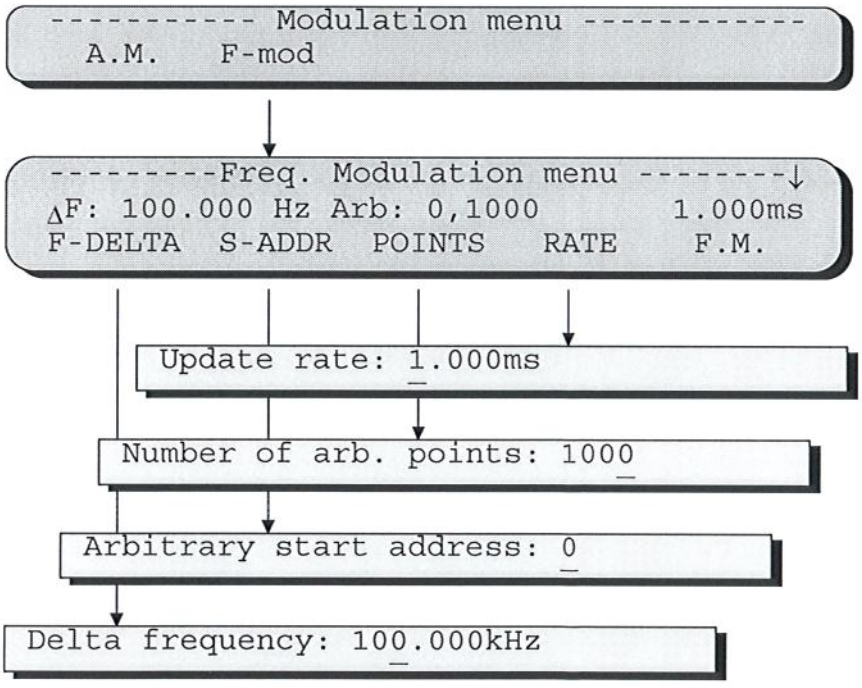
In the modulation menu you can choose A.M. and F.M. modulation.

A.M.

When amplitude modulation is selected the amplitude of the signal can be controlled by a signal on the A.M.-input. The A.M. modulation signals must be connected to the B.N.C. input at the back of the generator. A.M. can only be switched on and off. When A.M. is selected an input voltage at the A.M.-input of -0,5V to +0,5V is equal to a modulation-depth of 0% to 100%. The currently selected waveform and ist according settings are used as the carrier-wave.

F.M.

The F.M. modulation is generated internally according to specified settings. Only the sine-wave can be used as an F.M.-carrier wave. The modulating wave is defined by data in the arbitrary memory. F.M. can be switched on/off by the **F.M.** option.



Following F.M. parameters can be set:

F-DELTA

This is the max. frequency change of the F.M.-modulated wave in respect to the set carrier wave frequency. The frequency range of the F.M. signal will be Fcarrier ± (Fdelta/2).

S-ADDR

This is the start address in the arbitrary memory of the arbitrary waveform defining the modulating signal. An arbitrary value of -2048 will give a frequency of Fcarrier -(Fdelta/2). An arbitrary value of +2047 will give a frequency of Fcarrier+(Fdelta/2).

POINT

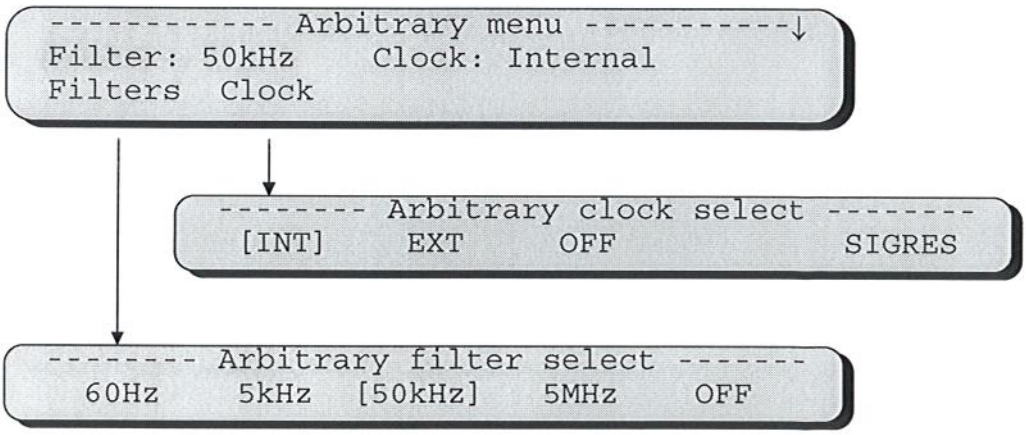
This is the number of points of arbitrary memory defining the modulating signal.

RATE

This is the duration time of one memory point in arbitrary memory specifying the F.M. frequency when F.M. is switched on. The F.M. signal will be modulated periodically with a period time of (POINTS*RATE).

4.6 Arbitrary Menu

The arbitrary menu allows you to access the arbitrary filter settings and the arbitrary clock settings.



Following settings can be changed:

Filters

There are four selectable arbitrary hardware filters at different cut-off frequencies. All four filters are 2e order Bessel type filters. Only one filter at a time can be selected. Filtering can be switched of by selecting the OFF option.

Clock

The clock signal for the arbitrary waveform generator can be switched off, generated internally (normal use) or applied externally. Only one of these options at a time can be selected.
The option SIGRES (**Sig**nal **Res**et) allows you to reset the arbitrary signal when the clock is set to external. This is useful when multiple generators are linked.

Multi generator linking and synchronising

Several generators can be linked together and synchronised when multiple channels are required in a synchronous system.

ATTENTION:
Multi generator linking and synchronising is only possible when arbitrary waveforms are used and only makes sense when the wavelengths of these arbitrary waveforms are all the same. When the wavelengths are not the same the waveforms of the channels will phase-shift automatically. Of course it is possible to do so, because there are always people who want to make something 'special'.

To create a synchronised multi generator system, the generators have to be coupled using the Clk-IN connector on the rear panel and the Clk-OUT connector on the rear panel of the instruments.

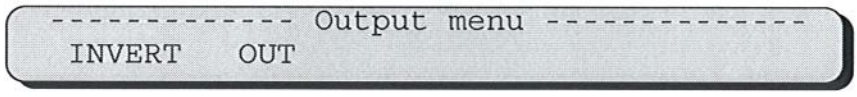
To link and synchronise two or more generators together next steps have to be followed:

- Set **Arbitrary Clock** of generator 1 (master) **off**.
- Select the waveforms in arbitrary memory you want to generate.
- Switch the **Arbitrary Clock** of generator 2 (slave) to **EXT**. Press **SIGRES** to reset the signal to its start-address.

Press **SIGRES** at generator 1 to reset the signal to its start-address. Switch the **Arbitrary Clock** of generator 1 to **INT**.

4.7 Output Menu

The **output** menu allows you to change the settings of the output signal.



Following settings can be modified:

INVERT

Selecting **INVERT** will invert the polarity of the current output signal.

OUT

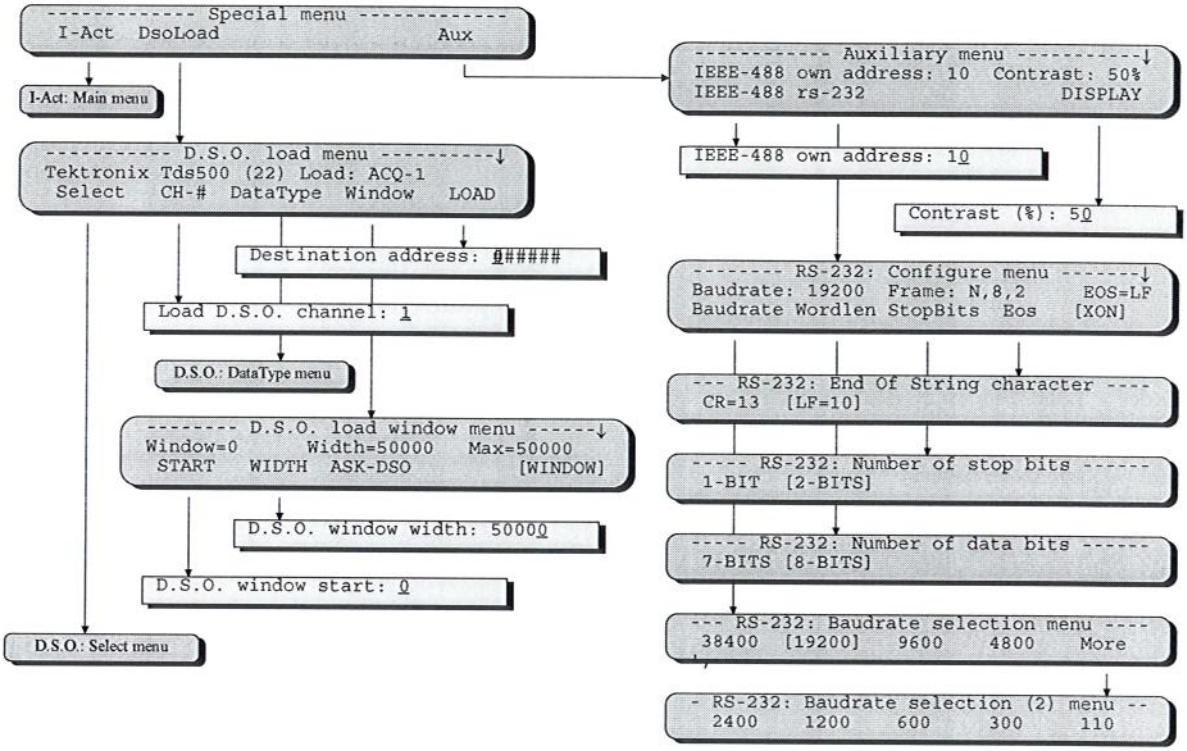
Toggles between output switched on and off.

4.8 Special Menu

The special menu allows you to access the Inter-Active arbitrary programming mode the DSO Load programming mode and the I/O modes and settings. By pressing one of the menu- keys the appropriate sub- menu will be displayed.

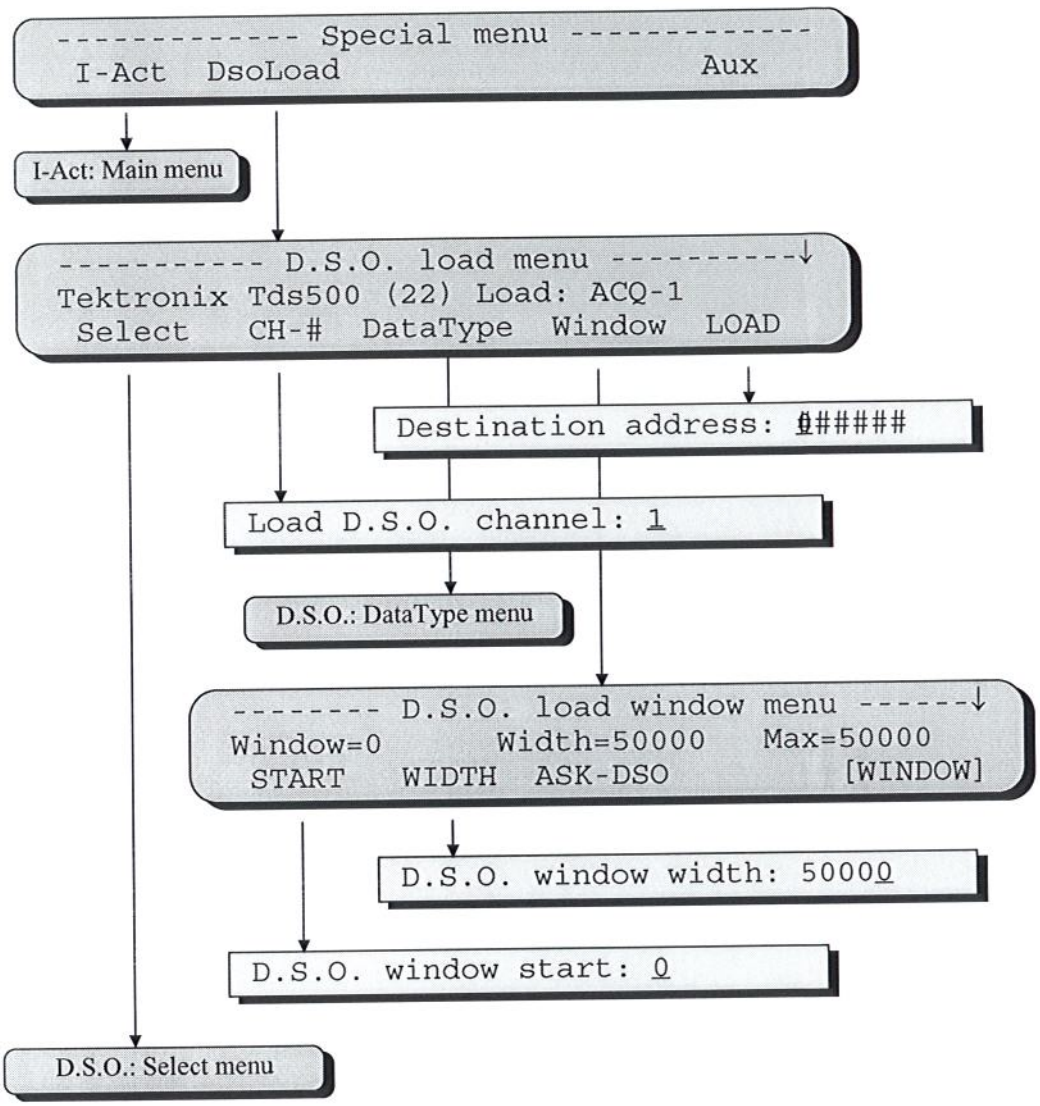
The **I-Act** option (Inter-Active arbitrary programming mode) and the **DSO Load** option (direct loading of DSO-waveforms) allow fast and easy programming of the arbitrary waveform memory.
Description of the I-Act option can be found in the next chapter 'Inter-active Main Menu'.

Selecting the **Aux** option accesses the menus for setting IEEE-488, RS-232 and DISPLAY parameters.



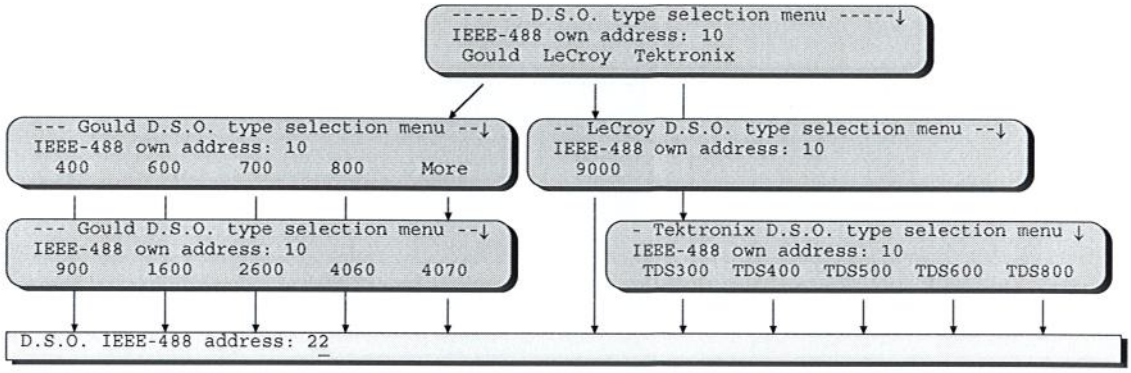
DsoLoad Menu

Selecting **DsoLoad** displays menu options which can be selected for downloading a waveform, from a DSO which is connected to the generator via the IEEE-488 connector, to the generators arbitrary waveform memory.



Before transferring a signal from an external device the generator must be connected to one or more of these devices by a standard IEEE-488 interface cable. Before the actual transfer of the signals can be started, the attached device from which the data is required must first be selected in the **D.S.O.:Select** menu.

When this option is selected the generator will display a new menu with several manufacturers names listed. Selecting the manufacturer will display the device type names. Selecting one of these will initialise the generator to transfer data from this device using the appropriate build-in driver.
The generator will also ask for the IEEE-488 bus address of the attached device. By default the addresses normally installed by the manufacturer will be displayed. After the selection of the IEEE bus address the generator will display the **D.S.O. load menu**.



D.S.O.: Select Menu

In the D.S.O. load menu following selections can be made:

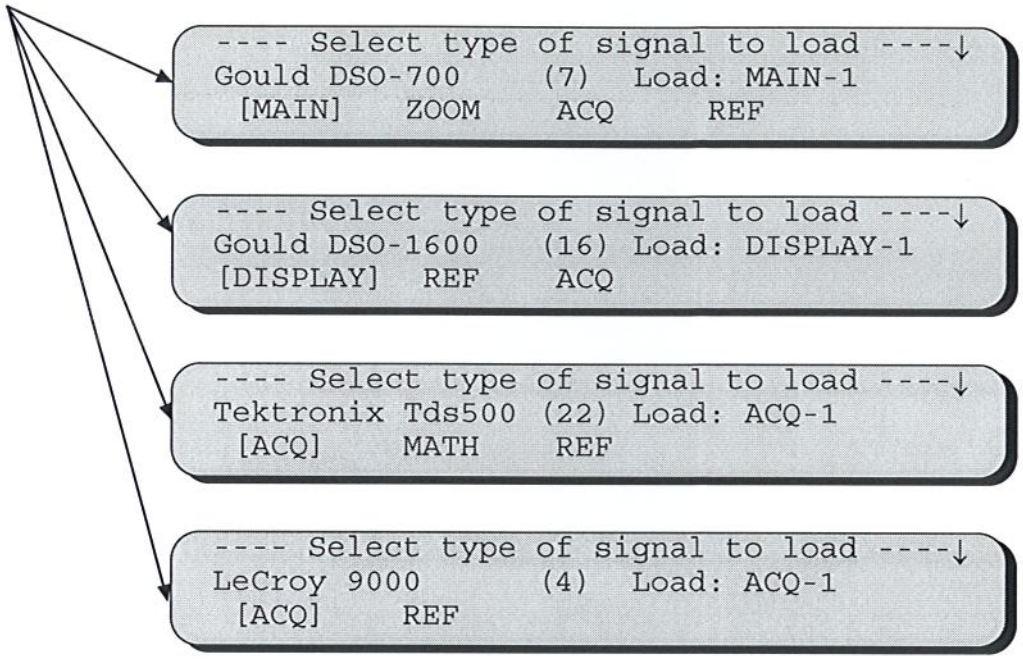
CH-#

This option is used for selecting the channel of the D.S.O. from which the waveform data has to be retrieved.

DataType

This option, which is only displayed when applicable for the selected D.S.O., selects the **DataType** menu. In this menu the type of memory from which the data has to be retrieved must be selected. For information concerning possible selections we refer to the user's manual of the specified D.S.O.

D.S.O.: DataType Menues



Window

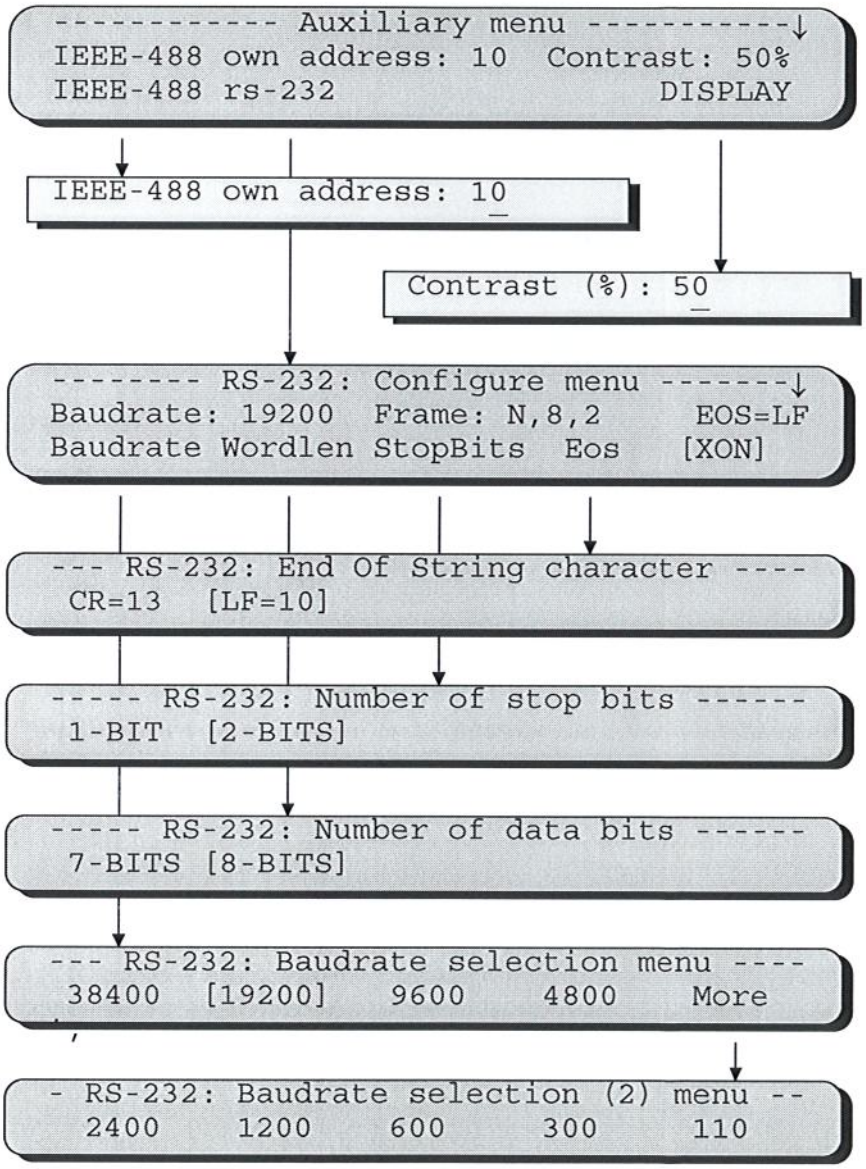
This menu option enables you to select a window in the channel data of the D.S.O., which has to be transferred. You can define the start and width of the window or 'ask' the D.S.O. for the currently used display window. For more information concerning windows we refer to the user's manual of the D.S.O.

LOAD

Selecting **LOAD** initiates the data transfer with the specified settings. You will be prompted for the start-address in the arbitrary memory to which the data must be programmed.

Aux Menu

Selecting **Aux** will display the auxiliary settings menu. In this menu you can set the IEEE-488 address of the generator, set the RS-232 transfer settings and change the contrast of the display.



Following items can be selected:

IEEE-488

You will be prompted to enter the 'ownaddress' of the generator as used during IEEE-488 control and data transfer. The value can be set from 0 to 30.

RS-232

Several setting for RS-232 communications can be selected to be according to the format used by your computer (program).

DISPLAY

Use this item to set the contrast of the LCD from 0% to 100%.

4.9 Inter-Active Main Menu

The Inter-Active Arbitrary Programming Menu

Selecting **I-Act** will display the Inter-Active arbitrary programming mode. This powerful mode enables you to program complex waveforms into the arbitrary memory of the generator.
The generator prompts you to enter waveform parameters and mathematical functions in a combination of menus and data values. If data is entered, the processor will calculate the waveform in arbitrary memory and the generator will ask again for data. In this way you can program very easily complex waveforms with noise, spikes, phase-shifts or even combine waves which are already loaded in the arbitrary memory from a digital storage oscilloscope with some of the mentioned possibilities. Inter active editing is allowed for waveforms with a maximum length of the whole waveform memory. However, be careful, this will overwrite all previous data in the arbitrary memory.
The 'I-ACT'- structure is built up out of one main menu and two sub-menus (Create and Add).

In the I-Act main menu the part of the arbitrary waveform memory on which we are going to operate (SEGMENT) is defined. After defining it, several operations can be performed; Creating a waveform from scratch in the SEGMENT by using the **Create** option (submenu), adding a waveform (glitch or whatever) to an already existing waveform in the selected SEGMENT by using the **Add** option or performing some mathematical processes on the SEGMENT by using the **Process** option.

SEGMENT

A SEGMENT is a part of the arbitrary memory in which we operate during Inter-Active programming. Select the **SEGMENT** option in the I-Act main menu to define a SEGMENT. A SEGMENT is defined by its start-address in the arbitrary waveform memory and its length (number of arbitrary memory points).

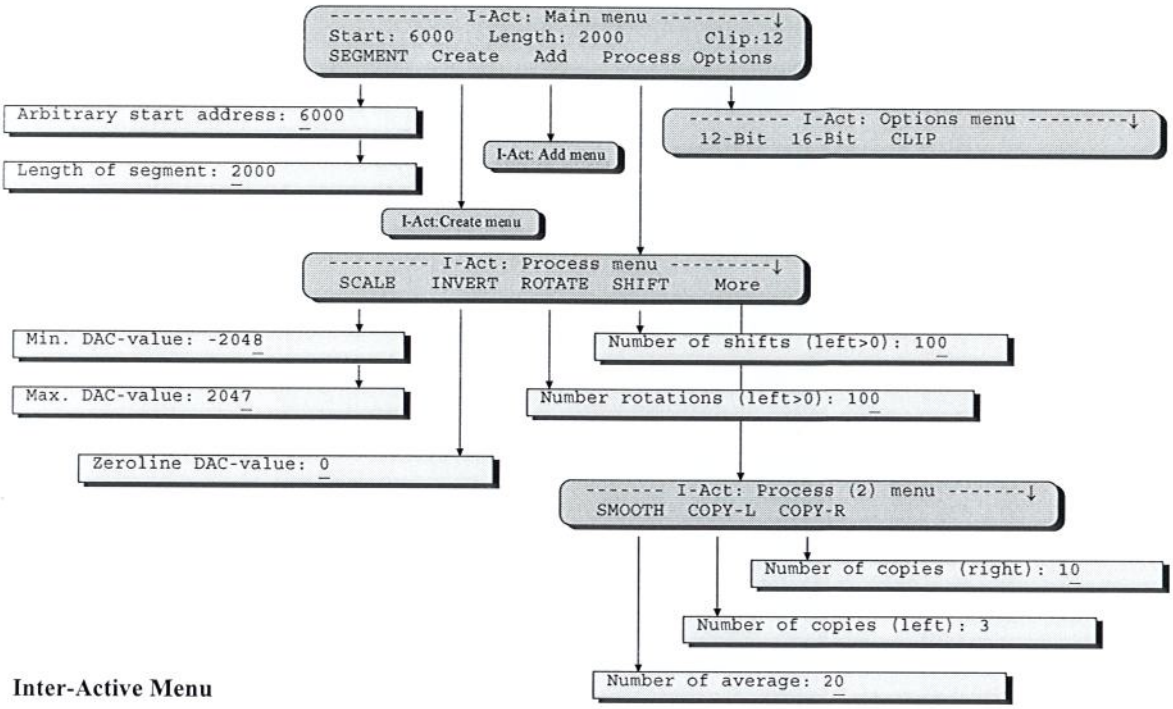
Options

The **Options** submenu allows you to set some optional parameters when using Inter-Active programming:

12-Bit
Sets the I-act mode to operate/calculate with 12-bit values. This is the normal operating mode because the generator uses a 12-bit DA-converter.

16-Bit
Sets the I-act mode to operate/calculate with 16-bit values, thus using the total width of the internal memory. This is only useful when using internal sweep or F.M. because this will increase the resolution (accuracy) of the modulating signal.

CLIP
Setting CLIP (normally on) causes the calculated signals to clip to their maximum value and not giving an overflow (going out of the range).



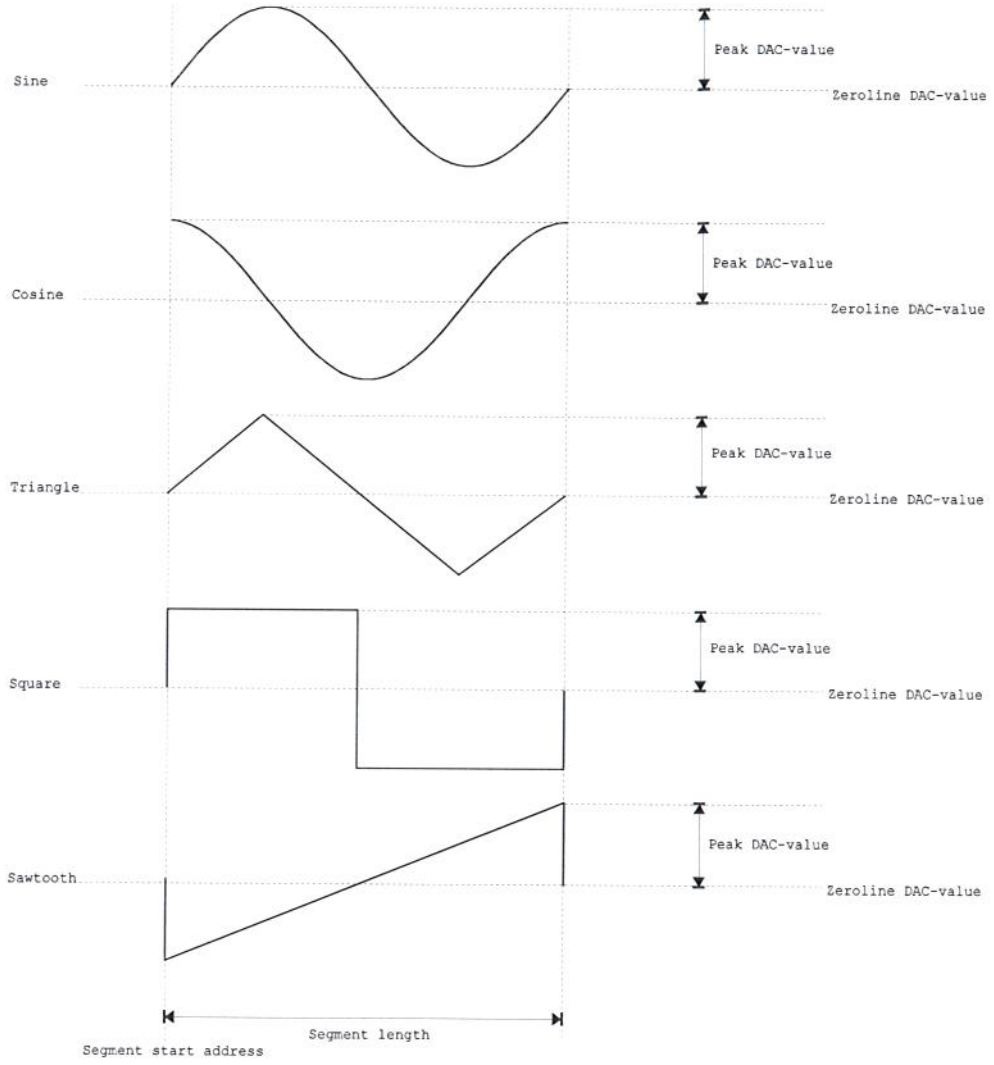
Inter-Active Menu

I-Act Create Menu

In the I-Act **Create** menu, waveforms can be programmed in the part of arbitrary memory defined as SEGMENT. The programming of the waveform will overwrite the previous data. Several types of waveforms are grouped in menus. Following waveforms can be created by entering the requested parameters:

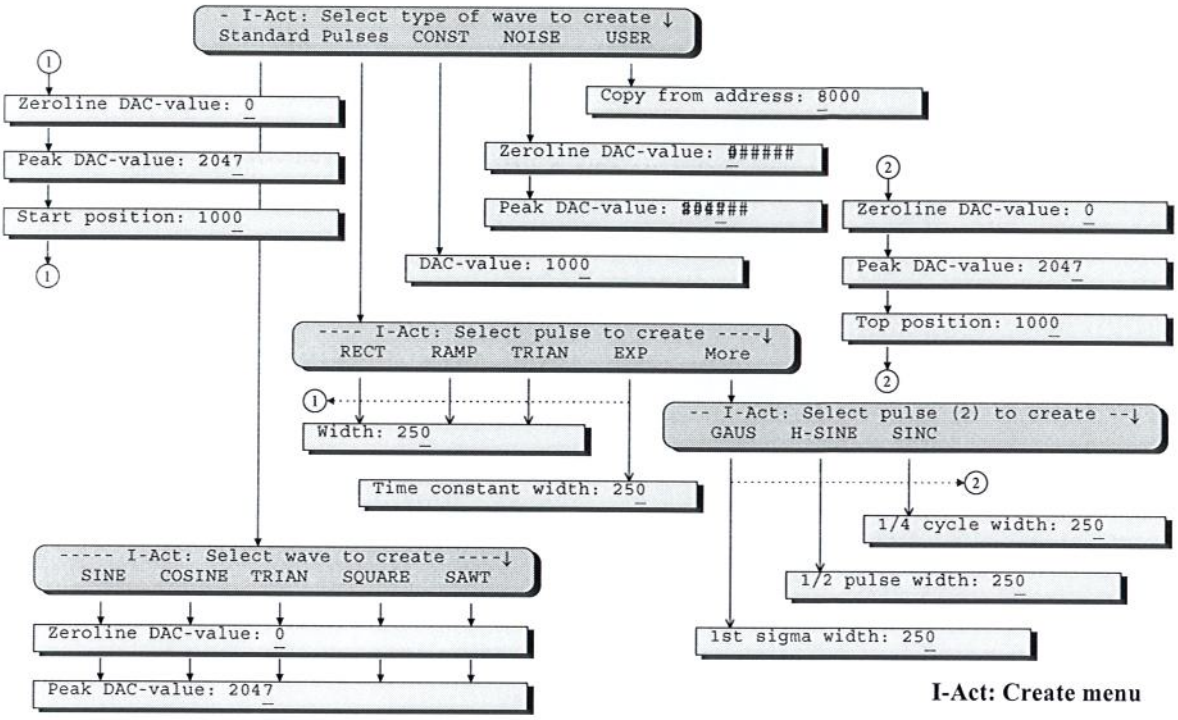
In the Standard waveforms menu

All these waveforms are bi-directional which means that they are programmed in an equal amount on both sides of the zeroline. The zeroline can be shifted and the amplitude which has to be entered is the peak amplitude in DAC- values. If a negative value for the peak amplitude is entered the waveform will start to go to the negative side (180° phase- shift).



If you select one of these waveforms the following data must be entered

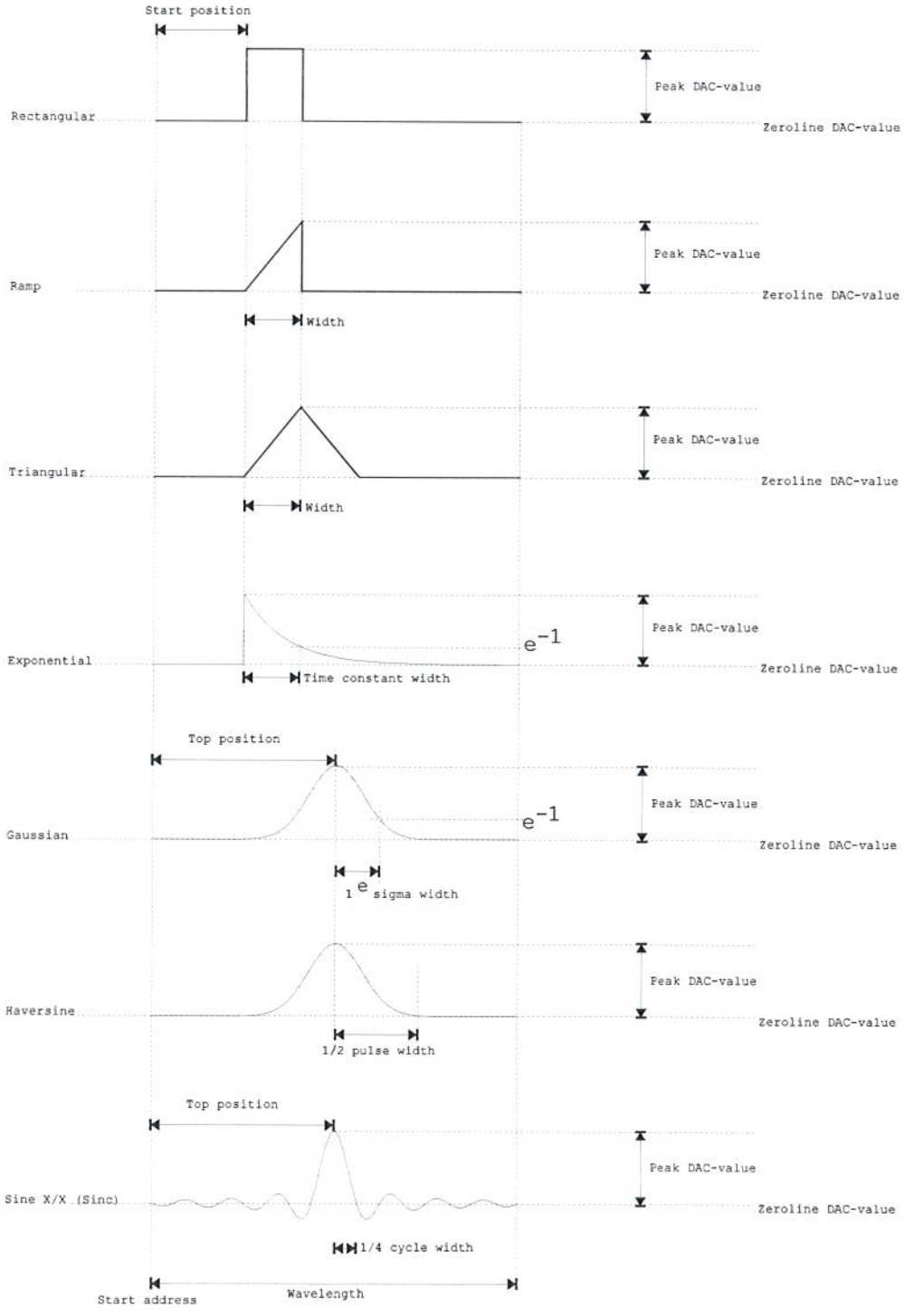
Data to enter	Value range (12-bits mode)	Value range (16bits-mode)
Zeroline DAC-value	-2048 ... +2047	-32768...+32767
Peak DAC-value	(-2047-Zero) ... (+2047-Zero)	(-32767+Zero)...(+32767-Zero)



I-Act: Create menu

In the Pulses waveforms menu

All these waveforms are uni- directional which means that they are programmed to one side of the zeroline. The zeroline can be shifted and the amplitude (positive or negative) which has to be entered is the peak- amplitude in DAC-values.



If you select one of these waveforms the following data must be entered

Rectangular pulse
Ramp pulse
Rectangular pulse

Data to enter	Value range (12-bits mode)	Value range (16bits-mode)
Zeroline DAC-value	-2048 ... +2047	-32768...+32767
Peak DAC-value	(-2047-Zero) ... (+2047-Zero)	(-32767+Zero)...(+32767-Zero)
Start Position	0 ... (Segmentlength-1)	0 ... (Segmentlength-1)
Width	1 ... Segmentlength	1 ... Segmentlength

Exponential pulse

Data to enter	Value range (12-bits mode)	Value range (16bits-mode)
Zeroline DAC-value	-2048 ... +2047	-32768...+32767
Peak DAC-value	(-2047-Zero) ... (+2047-Zero)	(-32767+Zero)...(+32767-Zero)
Start Position	0 ... (Segmentlength-1)	0 ... (Segmentlength-1)
Time Constant Width	1 .. (Segmentlength-Start position)	1 ... (Segmentlength-Start position)

Gaussian pulse

Data to enter	Value range (12-bits mode)	Value range (16bits-mode)
Zeroline DAC-value	-2048 ... +2047	-32768...+32767
Peak DAC-value	(-2047-Zero) ... (+2047-Zero)	(-32767+Zero)...(+32767-Zero)
Start Position	0 ... (Segmentlength-1)	0 ... (Segmentlength-1)
1e Sigma Width	1 .. (Segmentlength-Start position)	1 ... (Segmentlength-Start position)

Haversine pulse

Data to enter	Value range (12-bits mode)	Value range (16bits-mode)
Zeroline DAC-value	-2048 ... +2047	-32768...+32767
Peak DAC-value	(-2047-Zero) ... (+2047-Zero)	(-32767+Zero)...(+32767-Zero)
Start Position	0 ... (Segmentlength-1)	0 ... (Segmentlength-1)
½ Pulse Width	1 .. (Segmentlength-Start position)	1 ... (Segmentlength-Start position)

Sine x/x (Sinc) pulse

Data to enter	Value range (12-bits mode)	Value range (16bits-mode)
Zeroline DAC-value	-2048 ... +2047	-32768...+32767
Peak DAC-value	(-2047-Zero) ... (+2047-Zero)	(-32767+Zero)...(+32767-Zero)
Start Position	0 ... (Segmentlength-1)	0 ... (Segmentlength-1)
¼ Cycle Width	1 .. (Segmentlength-Start position)	1 ... (Segmentlength-Start position)

Three more types of waveforms can be selected in the **Create** waveforms menu:

CONSTant value:

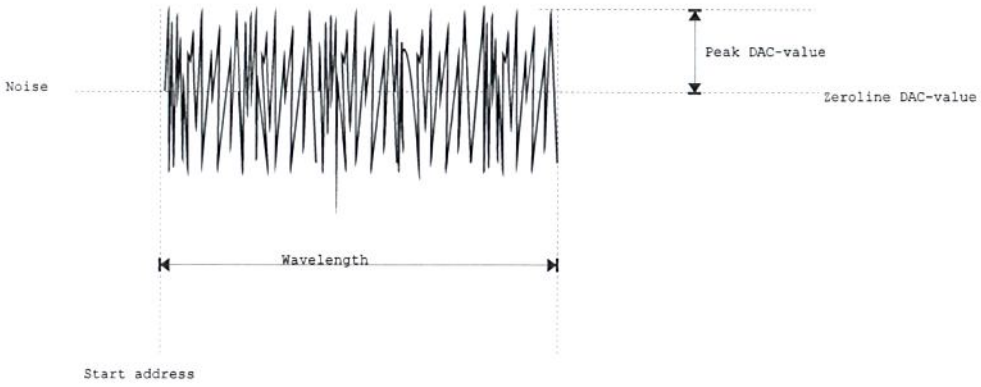
The **CONST**-option allows you to define a value to a range of points. The generator prompts you to enter the DAC-value to which all memory in the SEGMENT has to be programmed.

The following ranges are valid:

Data to enter	Value range (12-bits mode)	Value range (16bits-mode)
DAC-value	-2048 ... +2047	-32768...+32767

NOISE:

NOISE programs random values in the range defined by SEGMENT.



Following parameters can be set:

Noise

Data to enter	Value range (12-bits mode)	Value range (16bits-mode)
Zeroline DAC-value	-2048 ... +2047	-32768...+32767
Peak DAC-value	(-2047-Zero) ... (+2047-Zero)	(-32767+Zero)...(+32767-Zero)

USER defined waves:

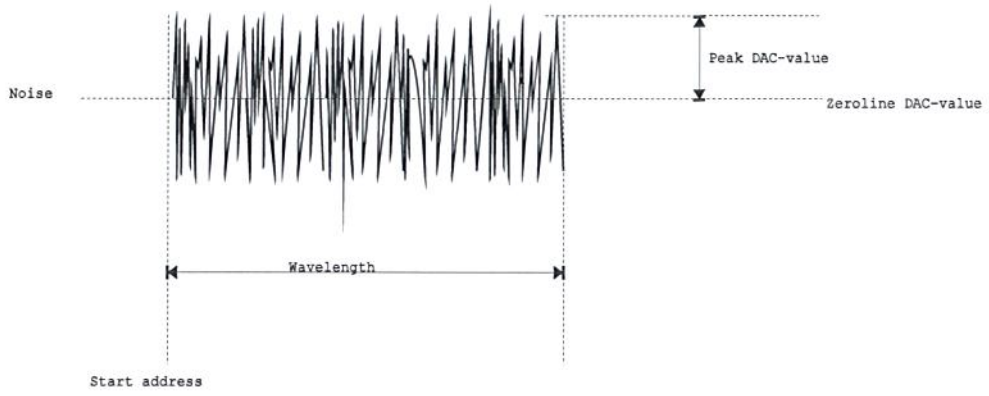
The **USER**-option allows you to define an existing waveform in your arbitrary waveform memory as user-defined waveform and copy this to your segment. The generator will prompt you to enter the following information:

Copy from address: Enter the address where you waveform must be *copied from.*

A waveform with the same length as SEGMENT and starting at the selected address will be copied to your SEGMENT.

NOISE:

NOISE programs random values in the range defined by SEGMENT.



Following parameters can be set:

Noise

Data to enter	Value range (12-bits mode)	Value range (16bits-mode)
Zeroline DAC-value	-2048 ... +2047	-32768...+32767
Peak DAC-value	(-2047-Zero) ... (+2047-Zero)	(-32767+Zero)...(+32767-Zero)

USER defined waves:

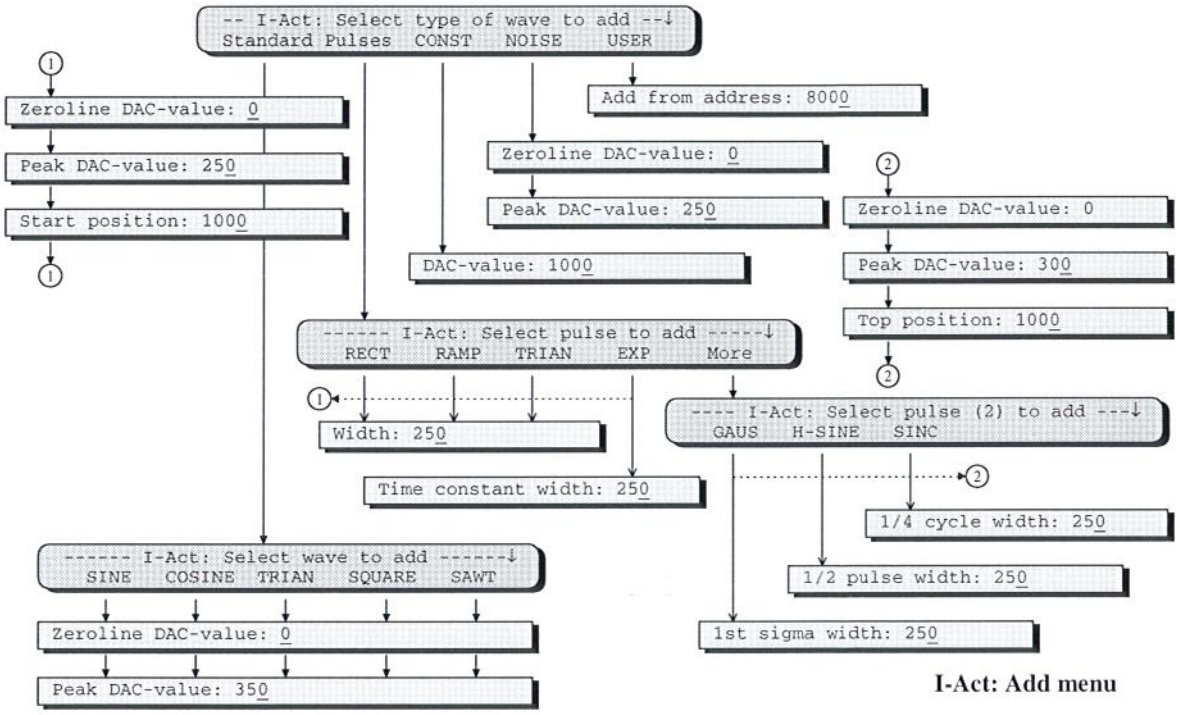
The **USER**-option allows you to define an existing waveform in your arbitrary waveform memory as user-defined waveform and copy this to your segment. The generator will prompt you to enter the following information:

Copy from address: Enter the address where you waveform must be *copied from*.

A waveform with the same length as SEGMENT and starting at the selected address will be copied to your SEGMENT.

I-Act Add Menu

In the I-Act **Add** menu, waveforms can be added to the existing waveform in your SEGMENT. The structure and waveforms used are the same as in the **Create** menu. If you want to add a waveform to a part of your SEGMENT, you must redefine your SEGMENT.
For an overview of the menu-structure we refer to following page.
For an overview and description of the waveforms, we refer to the previous part of this menu (The I-Act create menu)



I-Act: Add menu

ROTATE

Rotate lets you rotate the values of a SEGMENT while specifying the amount of memory points. You will be prompted to enter the "Number rotations (left>0)". Entering a negative value will rotate to the right, a positive value to the left.

Data to enter	Value range (12-bits mode)	Value range (16bits-mode)
Number rotations (left>0)	1 ... Segmentlength	1 ... Segmentlength

SHIFT

Shift lets you shift the values of a SEGMENT while specifying the amount of memory points. You will be prompted to enter the "Number of shifts (left>0)". Entering a negative value will shift to the right where the number of points shifted will be added to the left with the value of the memory point most left in your SEGMENT, A positive value will shift to the left where the number of points shifted will be added to the right with the value of the memory point most right in your SEGMENT.

Data to enter	Value range (12-bits mode)	Value range (16bits-mode)
Number of shifts (left>0)	1 ... Segmentlength	1 ... Segmentlength

SMOOTH

Smooth calculates a moving average over the number of points specified, thus filtering (smoothing) you signal in the SEGMENT.

Data to enter	Value range (12-bits mode)	Value range (16bits-mode)
Number of average (left>0)	1 ... Segmentlength	1 ... Segmentlength

COPY-L and **COPY-R:**

Copy allows you to copy your SEGMENT to the left or to the right in your arbitrary memory. You will be prompted for the number of copies you which to create.

Data to enter	Value range (12-bits mode)	Value range (16bits-mode)
Number of copies	Max. possible in memory	Max. possible in memory

I-Act example

To give an idea of the possibilities of the 'I-ACT'- programming mode we will give an example.

Example:
We want to program a sine wave starting at address 0 with wavelength of 1000 points and an amplitude of 2000 dac- values peak. This wave must be programmed around the zero- volt line when the offset parameter is zero. On this sine wave 10% noise must be added (200 dac- values).

To view what we are doing we can connect the generator to an oscilloscope.

To program this waveform we must select:

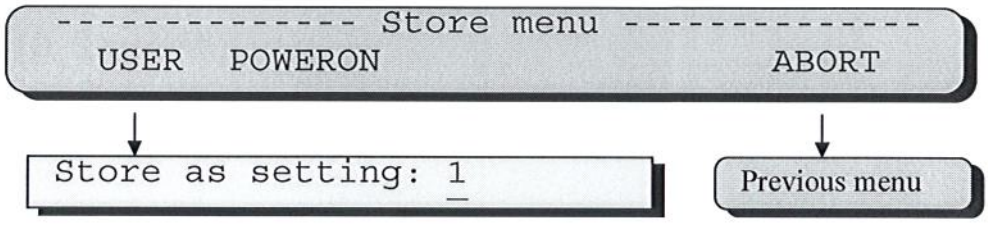
Display	Action	Comment
	Special	
	I-Act	
I-Act: Main menu	SEGMENT	We must first create our segment.
Arbitrary start address:	0 'Enter'	
Length of segment:	1000 'Enter'	
	Create	We are now going to create the sine-wave.
	Standard	Sine is a standard waveform.
	SINE	
Zeroline DAC-value:	0 'Enter'	
Peak DAC-value:	2000 'Enter'	After this the processor will start calculating. You can see this on the oscilloscope.
	Add	We are going to add something
	NOISE	
Zeroline DAC-value:	0 'Enter'	The noise shouldn't have any offset
Peak DAC-value:	200 'Enter'	
Calculating . . .		Noise is being programmed on the sine- wave.

This example shows how easy it is to program a waveform using I-Act. In the beginning it will take some time to find the right menu for the right operation, but by using

I-Act, just like every program on a computer, this will very soon change and you will find the right keys without even reading the display.

4.10 Store Menu

The **store** menu allows you store several types of settings into the non-volatile memory of the generator.



USER

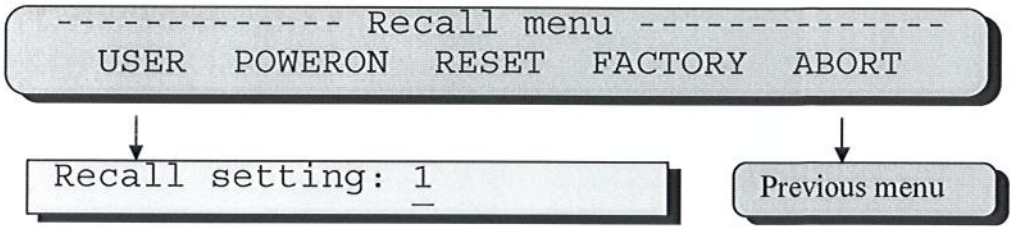
By **USER** you can store up to 10 user-defined generator settings. All hardware configurable parameters are stored. Arbitrary memory is not stored because there is only one arbitrary memory in the generator. All references to the arbitrary memory however are stored.
To store the current setting, select **USER** and enter the memory in which to store your settings (0 to 9).

POWERON

If you select **POWERON**, the current settings will overwrite the current power-on settings.

4.11 Recall Menu

The **recall** menu allows you recall several types of stored and preconfigured settings of the generator.



USER
Select **USER** to recall one of the previously stored configurations.

POWERON
You can recall the settings from the last power-on by selecting this option.

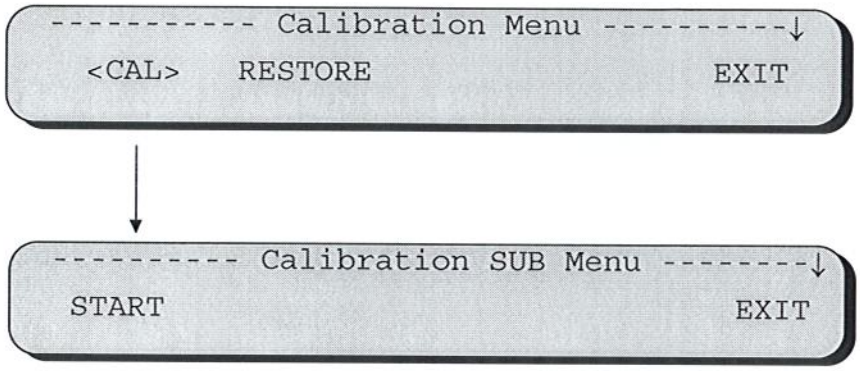
RESET
This recall configures the settings to defined settings as defined by IEEE488.2 (see IEEE command *RST)

FACTORY
This item sets all settings to the factory defined reset settings.

4.12 Calibration Menu

The ARB 1000 offers a self calibration that should be performed one time a year. All amplifiers are adjusted with an internal voltage source, so it is not necessary to connect any measurement instrument or voltage standard.

The calibration menu is opened if you press the CAL switch on the rear side of the instrument. In the display you see the following display



Before you start the calibration the output must be matched with a 50 Ohm shunt, otherwise the calibration will fail.
Be sure that you are not in an input menu, when depressing the CAL key on the rear of the instrument.

CAL, START

Calibration data will only be stored in the EEPROM when calibration was successfully performed, otherwise calibration data will remain unchanged.
The instrument should be allowed to warm up for one hour before calibration. Depressing START initiates calibration of the amplifiers both on amplitude and offset.

RESTORE

The RESTORE factory calibration is initiated by pressing the RESTORE key.
The factory calibration is the initial calibration performed with the first delivery of this function generator.

EXIT

Choosing EXIT exits the calibration menu, to return to normal operation.

5 Remote Control

5.1 Command syntax

You can control the instrument through both the GPIB and RS-232 interface using a large group of commands and queries. The instrument supports several industry standards and specifications (SCPI, IEEE 488.2, IEEE 488.1 and RS-232) in remote operation. This section describes the syntax which the commands and queries use. The commands and queries themselves are listed in a separate section called **Commands**.

The SCPI (Standard Commands for Programmable Instruments, say 'Skippy") standard, first published in 1990, is built on top of the IEEE 488.2 standard. SCPI defines program messages, response messages and data formats that are consistent across all instruments, regardless of manufacturer. The goal of these definitions is to reduce ATE program development time. The IEEE488.2 standard defines the syntax of a command language while SCPI defines the vocabulary.

Transmitting information to the instrument is done using the ASCII character encoding format. In this manual symbols of the Backus-Naur Form (BNF) notation and syntax diagrams are used to describe commands and queries.

Symbol	Meaning
< >	Defined element
: : =	Is defined as
|	Exclusive OR
{ }	Group (One element is required)
[ ]	Optional (Can be omitted)
...	Previous elements may be repeated
( )	Comment

Table: BNF Symbols and meanings

5.2 Command and query structure

SCPI commands consist of set commands (usually called commands) and query commands (usually called queries). Commands modify instrument settings or tell the instrument to perform an action. Queries are used to request data or status information from the instrument.
Most commands have a set form as well as a query form. The query form is usually the same as the set form but with a question mark on the end.

Set form: `AMPLitude 12E-1` (Sets the amplitude to 1.2Vpp)
Query form: `AMPLitude?` (Requests the current amplitude set ting)

Several standard elements are used defining a command message. A command message consists of words (mnemonics, the actual commands) and these elements.

Symbol	Meaning
<Header>	Basic command name. *May* begin with a colon (:). Multiple concatenated commands *must* begin with a colon. If the header ends with a question mark, the command is a query.
<Mnemonic>	A header sub-function. If a header has multiple mnemonics, they are separated by a colon.
< Space>	One or optionally more white space characters between command header and argument.
:	Colon; used to separate multiple mnemonics in a commands
<Argument>	A quantity, limit, quality or restriction associated with the header. Multiple arguments are separated from each other by a <Comma>
,	A single comma separates multiple arguments.
;	A semicolon is used to separate concatenated combinations of set commands and queries.

Table: Command message elements

Using the elements above a command can have the following structure:

[:]<Header>[<Space><Argument>[<Comma><Argument>]...][;<**Command**>]

Using the elements above a query can have the following structure:

[:]<Header>?[<Space><Argument>[<Comma><Argument>]...][;<**Query**>]

Commands

Command entries follow the following rules:

☞ You can enter commands in upper or lower case.

Example: AMPLITUDE 10
`amplitude 10`
Meaning: Set the amplitude to 10 Vpp.

☞ You can precede any command with white space characters.

☞ The command header can either use a long or a short form. Only the exact short and long forms are accepted.

Example: `AMPLitude 5`
`AMPL 5`
Meaning: Sets the amplitude to 5 Vpp.

☞ You can set maximum and minimum limit values of a parameter by giving the command the arguments MAX or MIN.

Example: `AMPL MAX`
Meaning: Sets the amplitude to 40 Vpp (Maximum limit).

Commands are hierarchical structured. This allows the same header to be used several times for different purposes.
See also chapter „Concatenating".

Queries

Query entries follow the following rules:

☞ You can enter queries in upper or lower case.

Example: `AMPLITUDE?`
`amplitude?`
Response: 1.000E+1
Meaning: The currently set amplitude is 10 Vpp.

☞ You can precede any query with white space characters.

☞ The query header can either use a long or a short form. Only the exact short and long forms are accepted.

Example: `:AMPLitude?`
`:AMPL?`
Response: 1.000E-1
Meaning: The currently set amplitude is 100 mVpp.

☞ The answer on a query will always be in the short form, e.g. if the answer on a query is TRIangle, TRI will be returned by the instrument.

Example: `:WAVE?`
Response: SQU
Meaning: The currently set waveform is a square.

☞ You can request maximum and minimum limit values of a parameter by giving the query the arguments MAX or MIN.

Example: `:AMPLitude? MAX`
Response: 40
Meaning: The maximum limit value of the amplitude is 40Vpp.

Queries are hierarchical structured. This allows the same header to be used several times for different purposes.
See also Chapter „Concatenating“.

Concatenating

The last example shows a concatenated command. When concatenating commands the following rules must be followed:

☞ Completely different headers must be separated by both a semicolon and by the beginning colon on all commands but the first.

Example: `AMPL 10;:MOD AM 1`
Meaning: Set amplitude to 10Vpp and switch AM to "on".

☞ If concatenated commands have headers that differ only by the last mnemonic, you can abbreviate the second command by eliminating the beginning colon.

Example: `MOD:AM 1;AM 0`
Meaning: Switch AM to "on" and switch AM to "off".

☞ Never precede a star (*) command with a colon.

☞ When you concatenate queries, the responses to all queries are concatenated into a single response message.

Example: `MOD:AM?;FM?`
Response: 1;0
Meaning: AM is switched on.
FM is switched off.

☞ Set commands and queries may be concatenated in the same message.

Constructed Mnemonic

A constructed mnemonic specifies one of a range of mnemonics, f.i. a specific sequence. You can use these mnemonics just as you do with other mnemonics. Constructed mnemonics are abbreviated to the base of the mnemonic followed by <x>. Thus, if there is a SEQ1 and SEQ2 mnemonic this will result in SEQ<x>.

Argument Types

Arguments are represented in several forms. The following arguments are used in this manual and defined as follows.

Argument symbol	Meaning
<>	No argument.
<Dig>	A digit character, range 0-9.
< NZDig>	A non-zero digit character, range 1-9.
<DChar>	A character, range 0..255.
<x>	Header suffix used in constructed mnemonics, range <NZDig>.
<NR1>	Signed integer value (16 bits value).
<NR1a>	Signed long value (32 bits value).
<NR1b>	{<NR1> | UP | DOWN | DEF}
<NR2>	Floating point value without an exponent.
<NR3>	Floating point value with an exponent.
<NRf>	Flexible numeric argument {<NR1> | <NR1a> | <NR2> | <NR3>}
<NRfa>	{<NRf> | UP | DOWN | MIN | MAX}
<NRfb>	{<NRf> | MIN | MAX}
<Boolean>	{<NRf> | ON | OFF}
<Block>	#<NZDig><Dig>[<Dig>...][<DChar>...]<terminator>

Table: Arguments used in this manual

<Boolean> A boolean argument has a value of 0 or 1 and is unitless. On input, an <NRf> is rounded to an integer. The elements ON and OFF shall be accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0.
A nonzero result is interpreted as 1. Queries shall return 1 or 0, never ON or OFF.

<Block> A block argument is used to transfer large amounts of data. The first sub-argument of the block argument is preceded by #, the block header. The following <NZDig> specifies the number of length digits that follow. The following length digits (1 to NZDig) specify the number of data elements that follow.

<terminator> For a GPIB network, <EOI> can be used as terminator. The end-of-message terminator may be the END message (EOI asserted concurrently with the last databyte).
The ASCII code for line feed (LF) sent as last databyte can also be used as terminator.

UP The UP argument will cause the parameter on which it is applied to be increased by +1.

DOWN The DOWN argument will cause the parameter on which it is applied to be increased by -1.

DEFault The DEFault argument may be provided to allow the instrument to select a value for a parameter. When DEFault is sent, the instrument shall select a value which is deemed to be convenient to the customer. The use of DEFault is optional on a command-by-command basis.

Example: `SYST:TIME UP,DEF,DEF`
Meaning: The instruments clock will be set ahead one hour (Daylight Savings Time).

5.3 Commands

The commands which can be used for the instrument will first be listed in an alphabetical range. Following this list each command will be described.

Arbitrary Memory commands

Header	Parameter form	Notes
AMEMory		
:ENCode?	{ASC|INT8|INT16|SINT16};{0|1};{WAVE|ALL}	Query only
:FORMat	{ASCii|INT8|INT16|SINT16}	
:SIGNed	<Boolean>	
:TYPE	{WAVE|ALL}	
:LENGth	<NRfb>	
:STARt	<NRfb>	
:DATA	{<Block>|<NR1[,NR1...]>}	

Table: Arbitrary Memory Commands

Settings Commands

Header	Parameter form	Notes
AMPLitude		
[:VPP]	<NRfa>	
:VP	<NRfa>	
:STEP	<NRfb>	
FREQuency		
[:CW]	<NRfa>	
:STEP	<NRfb>	
MODulation		
:AM	<Boolean>	
:FM		
[:STATe]	<Boolean>	
:DEViation	<NRfb>	
:STARt	<NRfa>	
:POINts	<NRfa>	
:RATE	<NRfb>	
OFFSet		
[:LEVel]	<NRfa>	
:STEP	<NRfb>	
OUTPut		
[:STATe]	<Boolean>	
:INVert	<Boolean>	
WAVE		
[:TYPE]	{DC|FSINe|FSQuare|SINE|COSine|TRIangle|SQUare|SAWTooth|RECTangle|PRAMp|NRAMp|ARBitrary|TPULse|HSINe|GAUSsian	
[:WAVE]	|EXPonential|SINC<x>}	
:RECTangle		
[:DCYCLe]		
:STEP	<NRfa>	
:ARBitrary?	<NRfb>	Query only
:ARBitrary	<NR1a>;<NR1a>;{F60|F5K|F50K|F1M|OFF};{INT|EXT|OFF}	
:STARt		
[:ADDRess]		
:STEP		
:LENGth	<NRfa>	
[:SAMPles]	<NRfb>	
:STEP		
:FILTer	<NRfa>	
:CLOCk	<NRfb>	
:SRESet	{F60|F5K|F50K|F1M|OFF}	
TRIGger	{INTernal|EXTernal|OFF}	
[:MODE]		
:NOBurst		
[:CYCLes]	{CONTinuous|TRIGgered|GATed|NBURst}	
:STEP		
:SOURce	<NRfa>	
[:TRIGger]	<NRfb>	
:INTerval<x>	{INTerval|EXTernal|MANual}	
[:WIDth]		
:STEP		
	<NRfa>	
	<NRfb>	

Sweep Commands

Header	Parameter form	Notes
SWEep		
:MARKer<x>		
[:FREQuency]	<NRfb>	
:POSition	<NRfa>	
:UPMarker	{Boolean}	
:POINts	<NRfa>	
:SPACing		
[:TYPE]	{LINear|LOGarithmic|ARBitrary}	
:ARBitrary		
[:STARt]	<NRfa>	
:STEP	<NRfb>	
:STARt	<NRfb>	
[:STATe]	<Boolean>	
:STOP	<NRfb>	
:TIME	<NRfb>	
:TRIangle	<Boolean>	
:TRIGger		
[:MODE]	{CONTinuous|TRIGgered|NBURst}	
:NOBurst		
[:COUNt]	<NRfa>	
:STEP	<NRfb>	
:SOURce	{INTerval|EXTernal|MANual}	

System Commands

Header	Parameter form	Notes
SYSTem		
:BEEPer	<NRfb>	
:COMMunicate		
:SERial		
:BAUD	<NRfa>	
:BITS	<NR1>	
:EOS	{CR|LF}	
:SBITs	<NR1>	
:PACE	{XON|NONE}	
:GPIB		
[:ADDRess]	<NRfa>	
:DISPlay		
[:CONTrast]	<NRfa>	
:ERRor?	<NR1>,"string"	Query only
:KEY	<NR1>	
:VERSion?		Query only

5.4 Command Descriptions

Most commands can be used either as a set command or as a query command. Commands which only can be used as a set command have the words "No query form" included with the command name. Commands which only can be used as a query command have the words "Query only" included with the command name.

Headers, mnemonics and arguments are spelled out fully in this manual (long form) with the minimal spelling (short form) shown in upper case. When text is enclode in brackets{ [ ] }, this text is optional in the described command.

AMEMory:ENCode? Query only

The `AMEMemory:ENCode` command queries the currently set FORMat, SIGNed and TYPE settings used for arbitrary memory data transfers to and from the generator.

TYPE settings.
A dataword in the arbitrary memory can hold 16 bits, the upper 12 databits are used for arbitrary waveform programming and the full 16 bits can be used for sweep waveform definition. To select whether the dat is put in the upper 12-bits which are used for arbitrary waveform generation or for sweep waveform generation which can use the full 16-bits resolution the TYPE of the data structure can be defined.

WAVE	Data is applicable only on (transferred to) the upper 12 bits used for wave data.
ALL	Data is applicable on the full 16 bits.

SIGNED settings.
The signed setting defines whether the data is signed or unsigned.

e.g.	Signed	-32767	⇒	+32768
	Unsigned	0	⇒	+65535

FORMAT settings.
The following data-formats can be used to transfer data:

ASCII	ASCII representation of values.
INT8	IEEE 8-bit integer.
INT16	IEEE 16-bit integer.
SINT16	Swapped IEEE 16-bit integer.

When 8-bit data is transferred to fill the arbitrary memory data, this data is automatically put in the MSB (exploded to 16 bits dynamic range).
E.g. values ranging from -128 ⇒ +127 will be expanded to -32767 ⇒ +32768.

Syntax: AMEMory:ENCode?

returned data: {ASC|INT8|INT16|SINT16};{0|1};{WAVE|ALL}

Example: AMEMory:ENCode?
might return the string:
SINT16;1;ALL indicating that the currently set dataformat is a swapped IEEE 16-bit
signed integer ([LSB][MSB])which is interpreted as 12-bits arbitrary waveform data and 4 bits trigger data.

AMEMory:ENCode:FORMat

The AMEMemory:ENCode:FORMat command sets or queries the currently set FORMat setting used for arbitrary memory data transfers to and from the generator.

Parameter	Explanation
ASCii	ASCii representation of values.
INT8	IEEE 8-bit integer.
INT16	IEEE 16-bit integer.
SINT16	Swapped IEEE 16-bit integer.

Table: FORMAT settings

Syntax: AMEM:ENCode:FORMat {ASCii|INT8|INT16|SINT16}
AMEM:ENCode:FORMat?

Examples: `AMEM:ENCode:FORMat ASCii`
sets format-setting for data-transfers to use ASCii representation of values.

`AMEM:ENCode:FORMat?`
might return the string:
INT16
indicating the currently set format for data-transfer is an IEEE 16-bit integer.

AMEMory:ENCode:SIGNed

The `AMEMemory:ENCode:SIGNed` command sets or queries the currently set SIGNed setting used for arbitrary memory data transfers to and from the generator.

Parameter	Explanation
0	Data is transferred as an unsigned value.
1	Data is transferred as a signed value.

Table: SIGNED settings

Syntax: `AMEM:ENCode:SIGNed <Boolean>`

`AMEM:ENCode:SIGNed?`

Examples: `AMEM:ENCode:SIGNed 0`
sets the signed-setting for data-transfers to use unsigned values.

`AMEM:ENCode:SIGNed?`
might return:
1
indicating that signed values are used during data-transfers to and from the arbitrary memory .

AMEMory:ENCode:TYPE

The `AMEMemory:ENCode:TYPE` command sets or queries the currently set TYPE setting used for arbitrary memory data transfers to and from the generator.

Parameter	Explanation
WAVE	Data is applicable only on (transferred to) the upper 12 bits used for wave data
ALL	Data is applicable on the full 16 bits.

Table: SIGNED settings

Syntax: `AMEM:ENCode:TYPE {WAVE|ALL}`

`AMEM:ENCode:TYPE?`

Examples: `AMEM:ENCode:TYPE WAVE`
sets the type-setting for data-transfers of arbitrary data to be applicable only on wave data.

`AMEM:ENCode:TYPE?`
might return:
ALL
indicating that data-transfers of arbitrary data are currently applicable to sweep data.

AMEMory:LENGth

The `AMEMemory:LENGth` command sets or queries the currently set LENGth setting used for arbitrary memory data transfers coming from the generator. The LENGth setting defines the number of arbitrary data elements which are transferred from the generator when an `AMEMemory:DATA?` command is used.

Syntax: `AMEM:LENGth <NRfb>`

`AMEM:LENGth ?`

Examples: `AMEM:LENGth 2000`
defines the number of arbitrary data elements to be transferred from the generator to 2000.

`AMEM:LENGth ?`
might return:
1231
indicating that the currently set number of data elements during a transfer of arbitrary data is 1231 words.

AMEMory:STARt

The `AMEMemory:STARt` command sets or queries the currently set arbitrary start-address, which indicates the first position in the arbitrary waveform memory, where the array of arbitrary words transferred, is positioned. The `STARt` setting is required when the `AMEMemory:DATA` command is used.

Syntax: `AMEM:STARt <NRfb>`

`AMEM:STARt ?`

Examples: `AMEM:STARt 32000`
defines the arbitrary memory start address during data transfer to be address 32000.

`AMEM:STARt ?`
might return:
21231
indicating that the currently set start address a transfer of arbitrary data is address 21231.

AMEMory:DATA

The `AMEMemory:DATA` command transfers arbitrary data to and from the arbitrary waveform memory of the generator.

Syntax: `AMEM:DATA {<Block>|<NR1[,NR1...]>}`

`AMEM:DATA?`

Examples:

```
AMEM:ENCode:FORM ASCii
AMEM:ENCode:SIGNed 1
AMEM:ENCode:TYPE WAVE
AMEM:STARt 100
AMEM:DATA -2048,-1593,-1138,-683,
          -228,227,682,1137,1592,2047
```

The first three commands define the data-structure to be signed ASCII which is applicable to the waveform data only.
Next, the start-address is defined to be address 100.
At last a stairwave of ten datapoints is transferred into the arbitrary waveform memory.

```
AMEM:ENCode:FORM INT16
AMEM:ENCode:SIGNed 0
AMEM:ENCode:TYPE WAVE
AMEM:STARt 100
AMEM:DATA #220
{Binary represented elements of values}
```

This string of commands performs exactly the same as the first example, however uses an unsigned binary data transfer.
After the #, 2 indicates there are two ASCII character following, defining the number of data**bytes** following. Because INT16 is used this will be twice the number of datawords. In the data-array each dataword will be [MSB][LSB].

```
AMEM:ENCode:FORM ASCII
AMEM:ENCode:SIGNed 1
AMEM:ENCode:TYPE WAVE
AMEM:LENGth 10
AMEM:STARt 100
AMEM:DATA?
```

returns:

```
-2048,-1593,-1138,-683,-228,227,682,
1137,1592,2047
```

when one of the previous examples is used before.

```
AMEM:ENCode:FORM INT16
AMEM:ENCode:SIGNed 0
AMEM:ENCode:TYPE WAVE
AMEM:LENGth 10
AMEM:STARt 100
AMEM:DATA?
```

returns:

```
#220 {Binary represented elements of values}
```

when one of the first two examples is used before.

AMPLitude[:VPP]

The `AMPLitude[:VPP]` command sets or queries the amplitude of the specified channel. Parameter values are measured in Vpp (Peak to peak voltage in matching impedance).

Syntax: `AMPLitude[:VPP] <NRfa>`

`AMPLitude[:VPP]?`

Examples: `AMPLitude 10`
sets the amplitude to 10 Vpp.

`AMPLitude:VPP DOWN`
decreases the current peak to peak amplitude setting with the current STEP value.

`AMPLitude?`
might return:
`1.0000E+0`
indicating that the amplitude is 1 Vpp.

AMPLitude:VP

The `AMPLitude:VP` command sets or queries the amplitude of the specified channel. Parameter values are measured in Vp (Peak voltage in matching impedance).

Syntax: `AMPLitude:VP <NRfa>`

`AMPLitude:VP?`

Examples: `AMPLitude:VP 5`
sets the amplitude to 5 Vp.

`AMPLitude DOWN`
decreases the current amplitude setting with the current STEP value.

`AMPLitude?`
might return:
`1.0000E+0`
indicating that the amplitude is 1 Vp.

AMPLitude:STEP

The `AMPLitude:STEP` command sets or queries the step-value of the amplitude setting.

Syntax: `AMPLitude:STEP <NRfb>`

`AMPLitude:STEP?`

Examples: `AMPLitude:STEP 1`
sets the amplitude-step (f.i. when AMP UP is used) to 1 V.

`AMPLitude:STEP?`
might return:
`1.0000E+0`
indicating that the amplitude-step is 1 Vp.

FREQuency[:CW]

The `FREQuency[:CW]` command sets or queries the frequency during Continuous Wave operation parameter. This is also the default command of the FREQuency command.

Syntax: `FREQuency[:CW] <NRfa>`

`FREQuency[:CW]?`

Examples: `FREQuency 1E3`
sets the frequency 1 kHz.

`FREQuency?`
might return:
`2.0000E+6`
indicating that the frequency of the waveform
is currently 2 MHz.

FREQuency:STEP

The `FREQuency:STEP` command sets or queries the frequency the step-value of the Frequency setting.

Syntax: `FREQuency:STEP <NRfb>`

`FREQuency:STEP?`

Examples: `FREQuency:STEP 1E3`
sets the frequency step to 1 kHz
(f.i. when FREQuency:CW DOWN is used).

`FREQuency:STEP?`
might return:
`2.0000E+6`
indicating that the STEP setting for the frequency is currently set to 2 MHz.

MODulation:AM

The `MODulation:AM` command sets or queries the current state of the AM-mode.

Syntax: `MODulation:AM <Boolean>`

`MODulation:AM?`

Examples: `MODulation:AM ON`
switches to AM-mode.

`MODulation:AM?`
might return:
OFF
indicating that AM-mode is off.

MODulation:FM[:STATe]

The `MODulation:FM` [:STATe] command sets or queries the current state of the FM-mode.

Syntax: `MODulation:FM[:STATe] <Boolean>`

`MODulation:FM[:STATe]?`

Examples: `MODulation:FM:STATe ON`
switches to FM-mode.

`MODulation:FM:STAT?`
might return:
OFF
indicating that FM-mode is off.

MODulation:FM:DEViation

The `MODulation:FM:DEViation` command sets or queries the current frequency deviation of the FM-mode (F-DELTA).

Syntax: `MODulation:FM:DEViation <NRfb>`

`MODulation:FM:DEViation?`

Examples: `MODulation:FM:DEViation 1e3`
switches the frequency deviation in FM-mode to 1kHz.

`MODulation:FM:DEV?`
might return:
1.200E+0
indicating that the frequency deviation in FM-mode
is currently set to 1.2 Hz.

MODulation:FM:STARt

The `MODulation:FM:STARt` command sets or queries the current frequency start address in the arbitrary memory of the arbitrary waveform defining the modulating signal (S-ADDR).

Syntax: `MODulation:FM:STARt <NRfb>`

`MODulation:FM:STARt?`

Examples: `MODulation:FM:STARt 1e3`
defines the arbitrary waveform starting at address 1000 as the modulating wave..

`MODulation:FM:STARt?`
might return:
1.200E+2
indicating that the modulating wave is starting at address 120.

MODulation:FM:POINts

The `MODulation:FM:POINts` command sets or queries the length (number of points) of the modulating waveform.

Syntax: `MODulation:FM:POINts <NRfa>`

`MODulation:FM:POINts?`

Examples: `MODulation:FM:POINts 1e3`
defines the length of the modulting wavefomr to be 1000 points.

`MODulation:FM:POINts?`
might return:
1.234E+3
indicating that the length of the modulating waveform is 1234 points.

MODulation:FM:RATE

The `MODulation:FM:RATE` command defines the duration of one memory point in arbitrary memory (update rate) specifying the F.M. frequency.

Syntax: `MODulation:FM:RATE <NRfb>`

`MODulation:FM:RATE?`

Examples: `MODulation:FM:RATE 1e-3`
defines the duration of one memory point in FM-mode to be 1ms.

`MODulation:FM:RATE?`
might return:
1.000E-3
indicating that the duration of one memory point in FM-mode is 1 µs.

OFFSet[:LEVel]

The `OFFSet[:LEVel]` command sets or queries the output DC-offset (into a matching impedance).

Syntax: `OFFSet[:LEVel] <NRfa>`

`OFFSet[:LEVel]?`

Examples: `OFFSet 1.3`
sets the offset to 1.3 V (into a matching impedance).

`OFFSet?`
might return:
`-1.201E+0`
indicating that the offset is currently -1.201 V.

OFFSet:STEP

The `OFFSet:STEP` command sets or queries the step-value of the offset.

Syntax: `OFFSet:STEP <NRfb>`

`OFFSet:STEP?`

Examples: `OFFSet:STEP 1.3`
sets the offset step-value to 1.3 V
(e.g. when OFFSet DOWN is used).

`OFFSet:STEP?`
might return:
`1.201E+0`
indicating that the offset step-value is currently set to 1.201 V.

OUTPut[:STATe]

The `OUTPut[:STATe]` command sets or queries whether the output is switched on or off. This is also the default command for the `OUTPut` command.

Syntax: `OUTPut[:STATe] <Boolean>`

`OUTPut[:STATe]?`

Example: `OUTPut:STATe ON`
switches the output on.

`OUTPut:STATe 0`
switches the output off.

`OUTPut:STATe?`
might return:
1 indicating that the output is switched on.
0 indicating that the output is switched off.

OUTPut:INVert

The `OUTPut:INVert` command sets or queries whether the output signal is inverted or not.

Syntax: `OUTPut:INVert <Boolean>`

`OUTPut:INVert?`

Example: `OUTPut:INVert ON`
sets the signal to be inverted at the output.

`OUTPut:INVert 0`
sets the signal not to be inverted at the output.

`OUTPut:INVert?`
might return:
1 indicating the signal is inverted.
0 indicating the signal is not inverted.

WAVE[:TYPE]

The `WAVE[:TYPE]` command sets or queries the waveform. The specified parameter defines which waveform is set. This is also the default command for the `WAVE` command.

Argument	Waveform
DC	DC
FSINe	Fast Sine
FSQuare	Fast Square
SINE	Sine
COSine	Cosine
TRIangle	Triangle
SQUare	Square
SAWTooth	Sawtooth
RECTangle	Rectangle
PRAmp	Positive ramp
NRAMp	Negative ramp
ARBitrary	Arbitrary
TPULse	Triangular pulse
HSINe	Haversine
GAUSsian	Gaussian pulse
EXPonential	Exponential pulse
SINC<x>	Sine x/x

Table: Waveforms

Syntax: `WAVE[:TYPE]` `{DC|FSINe|FSQuare|SINE|COSine|TRIangle|SQUare|SAWTooth|RECTangle|PRAMp|NRAMp|ARBitrary|TPULse|HSINe|GAUSsian|EXPonential|SINC<x>}`

`WAVE[:TYPE]?`

Examples: `WAVE[:TYPE] PPULse`
sets the waveform to positive pulse.

`WAVE[:TYPE]?`
might return:
PRAM
indicating that the current waveform is the postive ramp.

WAVE:RECTangle[:DCYCLe]

The `WAVE:RECTangle[:DCYCle]` command sets or queries the duty cycle in % of the rectangle waves during rectangle waveform operation. This is also the default command of the `WAVE:RECTangle` command.

Syntax: `WAVE:RECTangle[:DCYCle] <NRfa>`

`WAVE:RECTangle[:DCYCle] ?`

Examples: `WAVE: RECTangle[:DCYCle] 50`
sets the duty cycle for the rectangle to 50%.

`WAVE: RECTangle[:DCYCle]?`
might return:
`20`
indicating that the duty cycle of the rectangle is currently 20%.

WAVE:RECTangle:STEP

The WAVE:RECTangle:STEP command sets or queries the step-value of the rectangle wave.

Syntax: WAVE:RECTangle:STEP <NRfb>

WAVE:RECTangle:STEP ?

Examples: WAVE:RECTangle:STEP 1
sets the rectangle wave step-value to 1 %
(f.i. when WAVE:RECT DOWN is used).

WAVE:RECTangle:STEP ?
might return:
1.000E+0
indicating that the rectangle step-value is currently set to 1 %.

WAVE:ARBitrary? Query only

The WAVE:ARBitrary command queries the currently set STARt, LENGth, FILTer and CLOCk settings as described in the following command descriptions.

WAVE:ARBitrary:STARt[:ADDRess]

The WAVE:ARBitrary:STARt[:ADDRess] command sets or queries the start-address of the arbitrary waveform. This is the default command for the WAVE:ARBitrary:STARt command.

Syntax: WAVE:ARBitrary:STARt[:ADDRess] <NRfa>

WAVE:ARBitrary:STARt[:ADDRess]?

Examples: WAVE:ARBitrary:STARt[:ADDRess] 1000
sets the arbitrary waveforms start address
to sample number 1000.

WAVE:ARBitrary:STARt[:ADDRess]?
might return:
65000
indicating that the arbitrary waveforms start address is set to sample number 65000.

WAVE:ARBitrary:STARt:STEP

The WAVE:ARBitrary:STARt:STEP command sets or queries the step-value of the arbitrary start address.

Syntax: WAVE:ARBitrary:STARt:STEP `<NRfb>`

WAVE:ARBitrary:STARt:STEP `?`

Examples: WAVE ARBitrary:STARt:STEP `10`
sets the arbitrary start address step-value to 10.

WAVE:ARBitrary:STARt:STEP `?`
might return:
`1.000E+0`
indicating that the arbitrary start-address step-value is currently set to 1.

WAVE:ARBitrary:LENGth[:SAMPles]

The `WAVE:ARBitrary:LENGth`[:SAMPles] command sets or queries the wavelength of the arbitrary waveform. The wavelength is specified in number of samples.

Syntax: `WAVE:ARBitrary:LENGth`[:SAMPles]`<NRfa>`

`WAVE:ARBitrary:LENGth`[:SAMPles]`?`

Examples: `WAVE:ARBitrary:LENGth`[:SAMPles] `1000`
sets the arbitrary waveform length to 1000 samples.

`WAVE:ARBitrary:LENGth`[:SAMPles]`?`
might return:
`65000`
indicating that the arbitrary waveform length is set to 65000 samples.

WAVE:ARBitrary:LENGth:STEP

The WAVE:ARBitrary:LENGth:STEP command sets or queries the step-value of the arbitrary waveform length.

Syntax: WAVE:ARBitrary:LENGth:STEP `<NRfb>`

WAVE:ARBitrary:LENGth:STEP `?`

Examples: WAVE:ARBitrary:LENGth:STEP `10`
sets the arbitrary waveform length step-value to 10.

WAVE:ARBitrary:LENGth:STEP `?`
might return:
`1.000E+0`
indicating that the arbitrary waveform length step-value is currently set to 1.

WAVE:ARBitrary:FILTer

The `WAVE:ARBitrary:FILTer` command sets or queries the arbitrary waveform filters. One of four 2-pole Bessel filters can be selected to filter the arbitrary waveform.

Argument	2-pole Bessel filter frequency
F60	60 Hz
F5K	5 kHz
F50K	50 kHz
F1M	1 MHz
OFF	Filters switched off

Table: Selectable 2-pole Bessel filters.

Syntax: `WAVE:ARBitrary:FILTer {F60|F5K|F50K|F5M|OFF}`

`WAVE:ARBitrary:FILTer?`

Examples: `WAVE:ARBitrary:FILTer F5K`
sets the arbitrary filter to a cut-off frequency of 5 kHz.

`WAVE:ARBitrary:FILTer?`
might return:
OFF
indicating that there is no filter switched on.

WAVE:ARBitrary:CLOCk

The `WAVE:ARBitrary:CLOCk` command sets or queries whether the internal clock, the external clock or neither of them is used for the arbitrary waveform generation.

Argument	Selected clock
INT	INTernal
EXT	EXTernal
OFF	None

Table: Clock selection for arbitrary waveform generation.

Syntax: `WAVE:ARBitrary:CLOCk {INTernal|EXTernal|OFF}`

`WAVE:ARBitrary:CLOCk?`

Example: `WAVE:ARBitrary:CLOCk INT`
switches the internal clock on.

`WAVE:ARBitrary:CLOCk EXT`
switches the external clock on.

`WAVE:ARBitrary:CLOCk?`
might return:
INT indicating that the internal clock is switched on.
EXTindicating that the external clock is switched on.
OFF indicating that the arbitrary clock is switched off.

WAVE:ARBitrary:SRESet

The WAVE:ARBitrary:SRESet command resets the arbitrary waveform generator to the currently set start-address.

Syntax: `WAVE:ARBitrary:SRESet`

Examples: `WAVE:ARBitrary:SRESet`
resets the arbitrary waveform generator to the currently set start-address.

TRIGger[:MODE]

The `TRIGger[:MODE]` command sets or queries the trigger mode. The specified parameter defines whether the trigger mode is continuous, triggered, gated or NBurst. This is also the default command for the `TRIGger` command.

Argument	Trigger mode
CONTinuous	Continuous
TRIGgered	Triggered
GATed	Gated
NBURst	NBurst

Table: Trigger modes

Syntax: `TRIGger[:MODE]`
`{CONTinuous|TRIGgered|GATed|NBURst}`

`TRIGger[:MODE]?`

Examples: `TRIGger:MODE TRIGgered`
sets the trigger mode to triggered.

`TRIGger:MODE?`
might return:
NBUR
indicating that the trigger mode is NBurst.

TRIGger:NOBurst[:CYCLes]

The TRIGger:NOBurst[:CYCLes] command sets or queries the number of cycles during NBurst mode operation.

Syntax: TRIGger:NOBurst[:CYCLes] <NRfa>

TRIGger:NOBurst[:CYCLes] ?

Examples: TRIGger:NOBurst[:CYCLes] 50
sets the number of cycles during NBurst operation to 50 cycles.

TRIGger:NOBurst[:CYCLes] ?
might return:
2
indicating that the number of cycles during NBurst operation is currently set to 2 cycles.

TRIGger:NOBurst:STEP

The TRIGger:NOBurst:STEP command sets or queries the step-value of the number of cycles in Nburst mode.

Syntax: TRIGger:NOBurst:STEP <NRfb>

TRIGger:NOBurst:STEP ?

Examples: TRIGger:NOBurst:STEP 10
sets the step value of number of cycles in Nburst mode to 10.

TRIGger:NOBurst:STEP ?
might return:
1.000E+0
indicating that the step value of number of cycles in Nburst mode is currently set to 1.

TRIGger:SOURce

The `TRIGger:SOURce` command sets or queries the trigger source. The specified parameter defines whether the trigger source is external, internal interval or manual.

Argument	Trigger source
EXT	External
INT	Internal interval
MAN	Manual

Table: Trigger sources

Syntax: `TRIGger:SOURce {EXT|INT|MAN}`

`TRIGger:SOURce?`

Examples: `TRIGger:SOURce INT`
sets the trigger source to internally triggered.

`TRIGger:SOURce?`
might return:
MAN
indicating that the trigger source is manual.

TRIGger:INTerval[:WIDth]

The `TRIGger:INTerval[:WIDth]` command sets or queries the period time of the current internal trigger interval.

Syntax: `TRIGger:INTerval[:WIDth] <NRfa>`

`TRIGger:INTerval[:WIDth]?`

Examples: `TRIGger:INTerval[:WIDth] 5.000E-3`
sets the period time for the internal interval timer to 5 ms.

`TRIGger:INTerval[:WIDth]?`
might return:
`2.0000E-6`
indicating that the period time of the internal interval timer is currently 2 µs.

TRIGger:INTerval:STEP

The TRIGger:INTerval:STEP command sets or queries the step-value of the trigger interval.

Syntax: TRIGger:INTerval:STEP <NRfb>

TRIGger:INTerval:STEP ?

Examples: TRIGger:INTerval:STEP 10
sets the step value of the trigger interval to 10 sec.

TRIGger:INTerval:STEP ?
might return:
1.000E+0
indicating that the step value of the trigge rinterval
is currently set to 1 sec.

SWEep:MARKer<x>[:FREQuency]

The SWEep:MARKer<x>[:FREQuency] command sets or queries the marker frequency of sweep frequency markers 1 to 4. This is also the default command of the SWEeep:MARKer<x> command.

Syntax: SWEep:MARKer<x>[:FREQuency] <NRfb>

SWEep:MARKer<x>[:FREQuency] ?

Examples: SWEep:MARKer1:FREQuency 1E3
sets the marker frequency of marker 1 to 1 kHz.

SWEep:MARKer2:FREQuency ?
might return:
2.0000E+6
indicating that the marker frequency of marker 2 is currently set to 2 MHz.

SWEep:MARKer<x>:POSition

The `SWEep:MARKer<x>:POSition` command sets or queries the marker position of sweep markers 1 to 4, to or from a specified position in the sweep data memory.

Syntax: `SWEep:MARKer<x>:POSition <NRfa>`

`SWEep:MARKer<x>:POSition ?`

Examples: `SWEep:MARKer1:POSition 125`
sets the marker position of marker 1 to the sweep memory address 125.

`SWEep:MARKer2:POSition ?`
might return:
`876`
indicating that the marker position of marker 2 is currently set to address 876.

SWEep:MARKer<x>:UPMarker

The `SWEep:MARKer<x>:UPMarker` (Use Position Marker) command sets or queries whether for a specified marker (1 to 4) the frequency settings or the position settings must be used. When the parameter is '0', the frequency setting is used. When the parameter is '1', the position setting is used.

Syntax: `SWEep:MARKer<x>:UPMarker <Boolean>`

`SWEep:MARKer<x>:UPMarker?`

Examples: `SWEep:MARKer3:UPMarker ON`
switches to use of the position marker for marker 3.

`SWEep:MARKer2:UPMarker?`
might return:
`1` indicating that the position setting is used for marker2.
`0` indicating that the frequency setting is used for marker2.

SWEep:POINTs

The `SWEep:POINts` command sets or queries the currently set number of points which are used during a sweep.

Syntax: `SWEep:POINts <NRfa>`

`SWEep:POINts ?`

Examples: `SWEep:POINts 50`
defines the number of sweep datapoints to sweep through to be 50.

`SWEep:POINts?`
might return:
`212`
indicating that the currently set number of sweep points is 212 points.

SWEep:SPACing[:TYPE]

The `SWEep:SPACing[:TYPE]` command sets or queries the currently set sweep spacing to be linear, logarithmic or arbitrary.

Argument	Sweep spacing
LINear	Linear sweep
LOGarithmic	Logarithmic sweep
ARBitrary	Arbitrary sweep

Table: Sweep Spacing

Syntax: `SWEep:SPACing[:TYPE] {LINear|LOGarithmic|ARBitrary}`

`SWEep:SPACing[:TYPE] ?`

Examples: `SWEep:SPACing[:TYPE] ARB`
defines the sweep spacing to be arbitrary, using the currently defined start-address and points settings .

`SWEep:SPACing[:TYPE] ?`
might return:
LOG
indicating that the currently set sweep spacing is logarithmic.

SWEep:SPACing:ARBitrary[:STARt]

The `SWEep:SPACing:ARBitrary[:STARt]` command sets or queries the currently set start-address of an arbitrary sweep.

Syntax: `SWEep:SPACing:ARBitrary[:STARt] <NRfa>`

`SWEep:SPACing:ARBitrary[:STARt]?`

Examples: `SWEep:SPACing:ARBitrary[:STARt] 1E6`
defines the start address of the arbitrary sweep to be address 1000000.

`SWEep:SPACing:ARBitrary[:STARt]?`
might return:
9
indicating that the currently set start address of the arbitrary sweep is address 9.

SWEep:SPACing:ARBitrary:STEP

The `SWEep:SPACing:ARBitrary:STEP` command sets or queries the currently set step value for the start-address of an arbitrary sweep.

Syntax: `SWEep:SPACing:ARBitrary:STEP <NRfb>`

`SWEep:SPACing:ARBitrary:STEP ?`

Examples: `SWEep:SPACing:ARBitrary:STEP 10`
defines the step value for the start address of the arbitrary sweep to be 10 addresses.

`SWEep:SPACing:ARBitrary:STEP ?`
might return:
9
indicating that the currently set step value for the start address of the arbitrary sweep is address 9.

SWEep:STARt

The `SWEep:STARt` command sets or queries the currently set start-frequency of a sweep

Syntax: `SWEep:STARt <NRfb>`

`SWEep:STARt?`

Examples: `SWEep:STARt 1E6`
defines the start frequency of the sweep to be 1 MHz.

`SWEep:STARt ?`
might return:
1.5000E+6
indicating that the currently set start frequency of the sweep is 1.5 MHz.

SWEep[:STATe]

The SWEep[:STATe] command sets or queries whether the sweep mode is on or off. When the parameter is '0', the sweep mode is off, When the parameter is '1', the sweep mode is on. This is the default command for the SWEEP command.

Syntax: SWEep[:STATe] <Boolean>

SWEep[:STATe] ?

Example: SWEep:STATe ON
switches the sweep mode to on.

SWEep:STATe 0
switches the sweep mode to off.

SWEep:STATe ?
might return:
1 indicating that the sweep mode is on.
0 indicating that the sweep mode is off.

SWEep:STOP

The SWEep: STOP command sets or queries the currently set stop-frequency of a sweep.

Syntax: SWEep:STOP <NRfb>

SWEep:STOP?

Examples: SWEep:STOP 1E6
defines the stop frequency of the sweep to be 1 MHz.

SWEep:STOP?
might return:
1.5000E+6
indicating that the currently set stop frequency
of the sweep is 1.5 MHz.

SWEep:TIME

The `SWEep:TIME` command sets or queries the currently set sweep step duration time.

Syntax: `SWEep:TIME <NRfb>`

`SWEep:TIME ?`

Examples: `SWEep:TIME` 1E-3
defines the sweep step time to be 1 msec.

`SWEep:TIME ?`
might return:
1.5000E-3
indicating that the currently set sweep step time is 1.5 ms.

SWEep:TRIangle

The `SWEep:TRIangle` command sets or queries whether the sweep traingle mode is on or off. When the parameter is '0', the sweep triangle mode is off, When the parameter is '1', the sweep triangle mode is on.

Syntax: `SWEep:TRIangle <Boolean>`

`SWEep:TRIangle ?`

Example: `SWEep:TRIangle ON`
switches the sweep triangle mode to on.

`SWEep:TRIangle 0`
switches the sweep triangle mode to off.

`SWEep:TRIangle ?`
might return:
1 indicating that the sweep triangle mode is on.
0 indicating that the sweep triangle mode is off.

SWEep:TRIGger[:MODE]

The `SWEep:TRIGger[:MODE]` command sets or queries the sweep trigger mode of the generator. The specified parameter defines whether the sweep mode is continuous, step, toggle, halt or NBurst.

Argument	Sweep mode
CONTinuous	Continuous
TRIGgered	Triggered sweep mode
NBURst	NBurst of sweeps after trigger

Table: Sweep modes

Syntax: SWEep:TRIGger[:MODE] {CONTinuous|TRIGgered|NBURst}

SWEep:TRIGger[:MODE] ?

Examples: SWEep:TRIGger[:MODE] CONT
sets the sweep mode to continuously sweeping.
SWEep:TRIGger[:MODE] ?
might return:
NBURst
indicating that the sweep mode is NBurst.

SWEep:TRIGger:NOBurst[:COUNt]

The `SWEep:TRIGger:NOBurst[:COUNt]` command sets or queries the number of sweeps during NBurst sweep mode. This is also the default command of the `SWEep:TRIGger:NOBurst` command.

Syntax: `SWEep:TRIGger:NOBurst[:COUNt] <NRfa>`

`SWEep:TRIGger:NOBurst[:COUNt]?`

Examples: `SWEep:TRIGger:NOBurst:COUNt 50`
sets the number of cycles during NBurst sweep operation to 50 sweep cycles.

`SWEep:TRIGger:NOBurst:COUNt?`
might return:
`2`
indicating that the number of sweep cycles during NBurst sweep operation is currently set to 2 cycles.

SWEep:TRIGger:NOBurst:STEP

The `SWEep:TRIGger:NOBurst:STEP` command sets or queries the step-value of the number of sweep cycles during NBurst sweep operation.

Syntax: `SWEep:TRIGger:NOBurst:STEP <NRfb>`

`SWEep:TRIGger:NOBurst:STEP?`

Examples: `SWEep:TRIGger:NOBurst:STEP1`
sets the step value to 1 sweep cycle
(f.i. when `SWEep:TRIGger:NOBurst:COUNt DOWN` is used).

`SWEep:TRIGger:NOBurst:STEP?`
might return:
`12`
indicating that the STEP setting for the number of sweep cycles during NBurst sweep operation is currently set to 12 cycles.

SWEep:TRIGger:SOURCe

The `SWEep:TRIGger:SOURCe` command sets or queries the currently set trigger source for triggered, stepped, toggled, halt and NBurst sweeps.

Argument	Sweep trigger source
EXT	External input
INTerval	Internal trigger generator
MANual	Manual by run/stop key

Table: Sweep Trigger Source

Syntax: `SWEep:TRIGger:SOURCe {EXTernal|INTerval|MANual}`

`SWEep:TRIGger:SOURCe ?`

Examples: `SWEep:TRIGger:SOURCe` EXT
defines the trigger source of triggered sweep modes to be the external trigger source.

`SWEep:TRIGger:SOURCe ?`
might return:
EXT
indicating that the currently set trigger source for triggered sweeps is the external input.

SYSTem:BEEPer

The `SYSTem:BEEPer` command sets or queries the internal beeper function of the generator. The argument defines which 'beep' is generated. When queried, the generator will indicate whether this function is supported or not.

Argument	Beep function
0	Off
1	short beep
2,3,....	long, longer,.....
16	Continuous beep

Table: BEEP functions

Syntax: `SYSTem:BEEPer <NRfb>`

`SYSTem:BEEPer?`

Example: `SYSTem:BEEPer 3`
triggers a BEEP of length 3.

`SYSTem:BEEPer?`
might return:
0 indicating that the BEEPer command is not supported.
1 indicating that the BEEPer command is supported.

SYSTem:COMMunicate:SERial:BAUD

The `SYSTem:COMMunicate:SERial:BAUD` command sets or queries the RS-232C baudrate of the generator, ranging up to 115200 baud.

Syntax: `SYSTem:COMMunicate:SERial:BAUD <NRfa>`

`SYSTem:COMMunicate:SERial:BAUD?`

Examples: `SYSTem:COMMunicate:SERial:BAUD 2400`
sets the generators RS-232C baudrate to 2400.

`SYSTem:COMMunicate:SERial:BAUD?`
might return:
`115200`
indicating that the generators baudrate is set to 115200 baud.

SYSTem:COMMunicate:SERial:BITS

The `SYSTem:COMMunicate:SERial:BITS` command sets or queries the number of databits used by the RS-232C port of the generator. This character length can be 7- or 8- databits.

Syntax: `SYSTem:COMMunicate:SERial:BITS <NR1>`

`SYSTem:COMMunicate:SERial:BITS?`

Examples: `SYSTem:COMMunicate:SERial:BITS 7`
sets the RS-232C character length to 7 bits.

`SYSTem:COMMunicate:SERial:BITS?`
might return:
`8`
indicating that the RS-232C character length is currently set to 8 bits.

SYSTem:COMMunicate:SERial:EOS

The `SYSTem:COMMunicate:SERial:EOS` command sets or queries the End Of String terminator used by the RS-232C port of the generator. This End Of String terminator can either be a 'Carriage Return (CR= Ascii(13))' or a 'Line Feed (LF= ASCII(10))' character.

Syntax: `SYSTem:COMMunicate:SERial:EOS {CR|LF}`

`SYSTem:COMMunicate:SERial:EOS ?`

Examples: `SYSTem:COMMunicate:SERial:EOS CR`
sets the RS-232C EOS to the 'CR'-character.

`SYSTem:COMMunicate:SERial:EOS ?`
might return:
`LF`
indicating that the RS-232C EOS is set to the 'LF'-character.

SYSTem:COMMunicate:SERial:SBITs

The `SYSTem:COMMunicate:SERial:SBITs` command sets or queries the number of stopbits used by the RS-232C port of the generator. This number of stopbits can either be 1 stopbit or 2 stopbits.

Syntax: `SYSTem:COMMunicate:SERial:SBITs <NR1>`

`SYSTem:COMMunicate:SERial:SBITs?`

Examples: `SYSTem:COMMunicate:SERial:SBITs 2`
sets the RS-232C number of stopbits to 2 stopbits.

`SYSTem:COMMunicate:SERial:SBITs?`
might return:
`1`
indicating that the RS-232C number of stopbits
is currently set to the 1 stopbit.

SYSTem:COMMunicate:SERial:PACE

The `SYSTem:COMMunicate:SERial:PACE` command sets or queries whether the Xon/Xoff protocol or hardware handshaking is used by the RS-232C port of the generator.

Syntax: `SYSTem:COMMunicate:SERial:PACE {XON|NONE}`

`SYSTem:COMMunicate:SERial:PACE?`

Examples: `SYSTem:COMMunicate:SERial:PACE XON`
sets the RS-232C handshake to Xon/Xoff.

`SYSTem:COMMunicate:SERial:PACE?`
might return:
`NONE`
indicating that the RS-232C handshake
is set to hardware handshake.

SYSTem:COMMunicate:GPIB[:ADDRess]

The `SYSTem:COMMunicate:GPIB[:ADDRess]` command sets or queries the GPIB address of the generator, ranging from 0 to 30. This is also the default command of the `SYSTem:COMMunicate:GPIB` command.

Syntax: `SYSTem:COMMunicate:GPIB[:ADDRess] <NRfa>`

`SYSTem:COMMunicate:GPIB[:ADDRess]?`

Examples: `SYSTem:COMMunicate:GPIB:ADDRess 20`
sets the generators GPIB address to 20.

`SYSTem:COMMunicate:GPIB:ADDRess?`
might return:
`9`
indicating that the generators GPIB address is currently set to 9

SYSTem:DISPlay[:CONTrast]

The `SYSTem:DISPlay[:CONTrast]` command sets or queries the contrast setting of the liquid crystal display. Valid settings are from 0 to 100% . This is also the default command for the `SYSTem:DISPlay` command.

Syntax: `SYSTem:DISPlay[:CONTrast] <NRfa>`

`SYSTem:DISPlay[:CONTrast]?`

Examples: `SYSTem:DISPlay[:CONTrast] 60`
sets the generators LCD contrast to 60%.

`SYSTem:DISPlay[:CONTrast]?`
might return:
`75`
indicating that the generators LCD contrast is set to 75%.

SYSTem:ERRor? Query only

The `SYSTem:ERRor` command is a request for the next entry from the instruments error/event queue. This queue contains an integer in the range [-32768,32767]. Negative numbers are reserved by the SCPI standard. Positive errors are instrument-dependent. An error/event value of zero indicates that no error or event has occurred.

The generator answers the `SYSTem:ERRor` query with:
`<Error/event number>,<Error/event description>`

As errors and events are detected, they are placed in a queue, and the most recent error/event is discarded. When the queue is full - 1, the next error will generate a "Queue overflow" message.

The error/event queue will be cleared when any of the following occur:

* Upon power up.
* Upon receipt of a *CLS command.
* Upon reading the last item from the queue.

Error No.	Error description
-102	Syntax error. An unrecognised command or data type was encountered; for example, a string was received when the device does not accept strings.
-108	Parameter not allowed. More parameters were received than expected for the header; for example, the DISPlay command only accepts one parameter, so receiving DISPlay 1,2 is not allowed.
-109	Missing parameter. Fewer parameters were received than required for the header; for example, the DISPlay command requires one parameter, so receiving DISPlay is not allowed.
-110	Command header error. An error was detected in the header.
-112	Program mnemonic too long. The header contains more than twelve characters.
-113	Undefined header. The header is syntactically correct, but it is undefined for this specific device; for example *XYZ is not defined for any device.
-114	Header suffix out of range. Indicates that a non header character has been encountered in what the parser expects is a header element.
-121	Invalid character in number. An invalid character for the data type being parsed was encountered; for example, an alpha in a decimal numeric or a "9" in octal data.
-128	Numeric data not allowed. A legal numeric data element was received, but the device does not accept one in this position for the header.
-130	Suffix error. This error is generated when parsing a suffix.
-138	Suffix not allowed. A suffix was encountered after a numeric element which does not allow suffixes.
-141	Invalid character data. Either the character data element contains an invalid character or the particular element received is not valid for the header.
-160	Block data error. An error was detected in a block data element.
-168	Block data not allowed. A legal block data element was encountered but was not allowed by the device at this point in parsing.
-350	Queue overflow. A specific code entered into the queue in lieu of the code that caused the error. This code indicates that there is no room in the queue and an error occurred but was not recorded.
0	No error
+201	Query only
+202	No query allowed

Table: Error numbers

Syntax: `SYSTem:ERRor?`

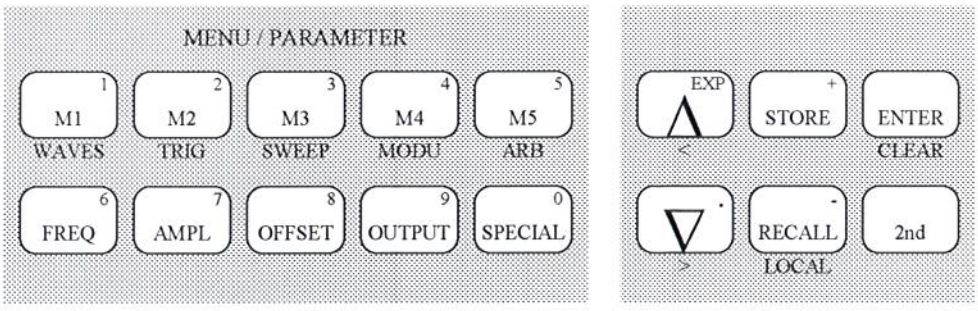
Examples: `SYSTem:ERRor?`
might return:
`-110,Command header error`
indicating that the generator detected an error
in a command header.

`0,No error`
indicating that there is no error event registered in the queue.

SYSTem:KEY

The `SYSTem:KEY` command sets a keypanel code, performing the same action as pushing the matching key, or requests the keypanel code of the last key pressed. Keypanel codes (numbers) are in decimal notation.

Key	Value
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
up	10
down	11
plus	12
min	13
enter	14
2^{nd}	15
standby	16
cal	17



Syntax: SYSTem:KEY <NR1>

SYSTem:KEY?

Examples: SYSTem:KEY 12
performs the same action as pressing the '+'-key.

SYSTem:KEY?
might return:
5
indicating that the generators last pressed key was the '5'-key.

SYSTem:VERSion? Query only

The `SYSTem:VERSion` command is a request for the generators firmware version.

Syntax: `SYSTem:VERSion ?`

Examples: `SYSTem:VERSion?`
might return:
`1.00.07`
indicating that the generators firmware version is version 1.00.07

6 Specifications

6.1 Waveforms

Standard

Fast (up to 20 MHz)	sine, square,
High Resolution	sine, cosine, triangle, square, sawtooth.
Pulses	rectangular, ramp up, ramp dn
Special Pulses	triangular, haversine, sin(x)/x, exponential, Gaussian pulse
Other	dc.

Arbitrary

Waveform length	2 to 130 000 points
Amplitude resolution	12 bits (4096 points)
Sample rate	100 mS/s to 40 MS/s
Non volatile memory	130 000 points

6.2 Frequency Characteristics

Fast Sine	100 mHz to 20 MHz (step 100 mHz)
Fast Square	100 mHz to 20 MHz (step 100 mHz)
High Resolution Waveforms	100 µHz to 40 kHz (step 100 µHz)
Rectangular Pulse	1 mHz to 400 kHz (step 1 mHz)
Ramp Up / Down	1 mHz to 200 kHz (step 1 mHz)
Special Pulses	100 µHz to 40 kHz (step 100 µHz)

Arbitrary Waveforms	100 mS/s to 40MS/s
Resolution	7 digits
Accuracy (1 year)	10 ppm, 18° C to 28° C
Temperature Coefficient	< 2 ppm / °C
Aging	< 10 ppm / year

6.3 Output Characteristics

Max. Output Voltage	10Vpp into 50 Ω, 20V into open circuit
Impedance	Zo = 50 Ω
Load capability	short circuit proof
Max external voltage	cut-off protection at ± 15 V
Output Current	I = 200 mA max.
Output Options	switchable on/off or inverted
Amplitude (into 50 Ω)	10 mVpp to 10 Vpp
Accuracy (at 1 kHz, sine Hi Res)	± 1% or 10 mV of specified output
Flatness (sine wave rel. to 1 kHz)	
<100 kHz	± 0.1 dB
100 kHz to 1 MHz	± 0.15 dB
1 MHz to 15 MHz	± 0.3 dB
Offset (into 50 Ω)	± 5Vpk ac+dc
Accuracy	± 2 % or 10 mV of setting
Resolution	3 digits, amplitude and offset
Rise/Fall time	< 12 ns typ.
Overshoot	< 5%

6.4 Signal Characteristics

Sinewave

Harmonics (dc to 100 kHz)	-45 dBc

Square

Rise / Fall time	< 12 ns typical
Overshoot	< 5%
Asymmetry	< 1%

Rectangular Pulse (Pos. /Neg.)

Rise / Fall Time	< 12 ns typical
Duty Cycle	1 % to 99 % selectable

Triangle / Ramp up / Ramp down

Linearity Error	< 0.5 % (between 10 % and 90 % duty cycle)

6.5 Operating Modes

Continuous		Output continuous at programmed frequency, amplitude and offset
Trigger	fmax = 5 MS/s	Triggered int. / ext. / remote / manual Gated ext. NBurst int. / ext. / remote / manual Number of cycles N = 1 to 65535
	Internal Trigger	Rate 1 µs to 1000 s Resolution 4 digits

Sweep	Linear (up, down or triangular) Logarithmic (up, down or triangular) User defined, arbitrary (up, down or triangular)
Markers	4 (absolute or relative)
Modes	Continuous, Triggered, NBurst (N = 1 to 65535)
Sweep points	1 000 000 max.
Point Duration	1 ms to 1000 s
Carrier Wave	All waveforms (except for arbitrary spacing which is sine wave only)
Start- / Stop Frequency	depending on carrrier waveform (see frequency specifications)
Modulation AM ext.	
Modulating signal	Connection on AM IN connector (rear side)
Input Voltage	- 0.5 V ... + 0.5 V for 100 % modulation
Bandwidth	DC ... 20 kHz
Carrier Waveform	all
Carrier Frequency	depending on carrier waveform (see frequency specifications)
Modulation FM int.	
Modulating signal	arbitrary (defined in arbitrary memory)
Carrier Waveform	Sine wave
Carrier Frequency	100 mHz ... 20 MHz
	Other modulations possible with arbitrary waveform creation.

6.6 Arbitrary Functions

Point duration	25 ns to 10s
Transition Time	< 12 ns typical
Start address	Selectable through whole memory
Length	2 ... 130 000 points
Edit functions	Wave form creation and editing with: • Create / Add standard waveforms, const, noise or user defined waveforms • Scale, Invert, Rotate, Shift, Smooth, Copy left / right • DSO waveform downloading

6.7 Arbitrary Filters

2-pole Bessel Filter	User selectable low-pass filters at: 60 Hz, 5 kHz, 50 kHz and 5 MHz

6.8 IEEE-488 Interface

OUTPUT INFORMATION	waveform, frequency, amplitude, offset, sweep mode, trigger mode, arbitrary and other instrument settings
INPUT INFORMATION	waveform selection, setting of frequency, amplitude, offset, sweep mode, trigger mode and other device settings, arbitrary waveform generation.
LANGUAGE	Conforming to SCPI
ADDRESS	selectable from 0 to 30, in the menu "Special, Aux menu, IEEE-488".
TERMINATOR	EOI Line and Line Feed
KEYBOARD	can be de-activated over REN, can be activated over GTL and over LOCAL key (lockable over LLO)
COMPATIBILITY	IEEE-488.1 and IEEE-488.2
BUS CONNECTOR.....	24-pin as per IEEE-488

PIN ASSIGNMENTS IEEE-488 INTERFACE

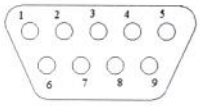
Data Bus:		
DIO 1-DIO 8	Data Bits 1-8	I/O
Data Transfer Control Bus		
DAV	Data Valid	I/O
NRFD	Not ready for Data	I/O
NDAC	No Data Accepted	I/O
Interface Control Bus:		
IFC	Interface Clear	I
ATN	Attention	I
SRQ	Service Request	O
REN	Remote Enable	I
EOI	End or Identify	I/O

Attention! Avoid any discharge of electrostatic voltages over the IEEE-488 connector, since it can lead to damage or destruction of the unit.

6.9 RS232 Serial Interface

BAUD RATE	selectable between 38 400 Bd and 110 Bd
WORD LENGTH	7-bits or 8-bits
STOPP BITS	1 bit or 2 bits
EOS	CR = 13 or LF = 10
HANDSHAKE	selectable: - Xon/Xoff - no handshake
STANDARD SETTING Xon/Xoff	19 200 Bd / Frame N, 8, 2 / EOS = LF /
CONNECTOR TYPE	9-pin Sub-D connector

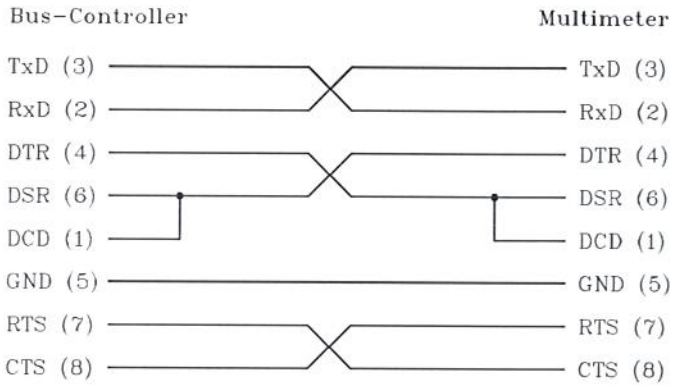
PIN ASSIGNMENTS



PIN-No.	Orientation	Signal	Description
1	Input	DCD (Data Carrier Detect)	
2	Input	RD (Receive Data)	Received Data
3	Output	TD (Transmit Data)	Sent Data
4	Output	DTR (Data Terminal Ready)	
5		GND	Signal Ground
6	Input	DSR (Data Set Ready)	
7	Output	RTS (Request to Send)	
8	Input	CTS (Clear to Send)	
9	Input	RI (Ring Indicator)	

Table: PIN assignment RS232

PIN Assignment RS232 Cable (Accessory No. 3018)



6.10 BNC Connectors

CONNECTOR TYPE	BNC Connector
FRONT CONNECTORS	Signal Output (50 Ω) TTL SYNC Output
REAR CONNECTORS	5
Output	Mrk-OUT (TTL), Clk-OUT (TTL)
VOL	Output low voltage (IOL = 8 mA) max. 0.4 V
VOH	Output high voltage (IOH = -4 mA) min. 2.4 V
Input	Trig IN (TTL), Clk-IN (TTL)
VIL	Input low voltage min. 0 V max. 0.8 V.
VIH	Input high voltage min. 2.0 V max 5.0 V
AM-Input	AM IN
Voltage	- 0.5 V ... + 0.5 V for 100 % modulation

6.11 EU Conformity

The EU Declaration of Conformity for the ARB 1000 certifies that this instrument conforms to the pertinent requirements of the relevant EU directives and standards.

EMC Compliance Tests

The following EMC (EMC = Electromagnetic Compatibility) compliance tests have been carried out conforming to the EMC directive 89/336/EWG. The compliance has been documented.

Measurement of the EMI Emissions

EN 50081-1
Electromagnetic Compatibility
Generic Standard Class: Residential, Commercial and Light Industry

EN 55022
Class B, Limits and methods of measurement of Radio Interference Characteristics of Information Technology Equipment (CISPR 22, modified), European Standard
Emission radiated, Frequency Range 30 MHz to 1 GHz.

EN 55022
Class B, Limits and methods of measurement of Radio Interference Characteristics of Information Technology Equipment (CISPR 22, modified), European Standard
Emission conducted, Frequency Range 150 kHz to 30 MHz.

Measurement of EMI Immunity

Conforming EN 50082-1, **European Standard,**
Electromagnetic Compatibility - Generic Immunity Standard
Generic Standard Class: Residential, Commercial, or Light Industry.
Additional 5017 specifications under EMC test conditions

ENV 50140
EMC Immunity against radiated EMI (Electromagnetic Interference).
EMC Basic Directive, 30-1000MHz. Residential environment
EMI Radiation: 3V/m, 80%AM - f.e. Handy in a distance of 3-5m

ENV 50141
EMC Immunity against conducted EMI
EMC Basic Directive, 150 kHz - 80 MHz
EMI Source: $3V_{RMS}$ on open wires.

EN 61000-4-2
ESD Immunity Part 4-2
ESD: Air or Contact Discharge +/-8kV / 4 kV - Protection Class 2.

EN 61000-4-4
Burst and electrical fast transient immunity Part 4-4
Environment: Industrial Area 2kVpk - Protection Class 3.

prEN 61000-4-5
Surge Immunity Test, Part 4-5
Environment: Industrial Area 2kV non symm. - Protection Class 3.

6.12 General Information

Voltage	115V ±10% or 230V ±10% for 50 and 60Hz
Environment Temperature	10°C to 40°C in operation -25°C to 60°C in storage
Power Consumption	max. 54 VA
Case	aluminum case for good EMC / EMI protection
H x W x D:	89 mm (2HU) x 225 mm (½19") x 375 mm
Weight	about 5 kg / 11 lbs
Rel. Humidity	< 25 °C: up to 75% rel. > 25 °C: up to 65% rel.
Warm-up Time	1 hour
Safety	EN 61010 • CE
EMC	EN 50 081, EN 50 082, EN 55 011 • CE
Warranty	2 years
Software	***Wavemaster*** for Windows (available as an accessory, see Chapter 'Accessories')

7 Accessories

7.1 RS232 Cable (3018)

RS232 cable for controlling the ARB 1000 from a PC with RS232 interface.
Zero Modem Cable with RTS/CTS and SD/RD lines crossed.

7.2 Carrying Case (4100)

Flexible carrying case for ARB 1000 and accessories with handle and shoulder strap.
Dimensions in cm : about 27 x 39 x 15 (W x H x D)

7.3 Accessories for the IEEE488 Bus

An interface card in the PC is necessary to remote control the ARB 1000
PREMA offer an IEEE488 interface card for PC/AT/XT and compatibles:

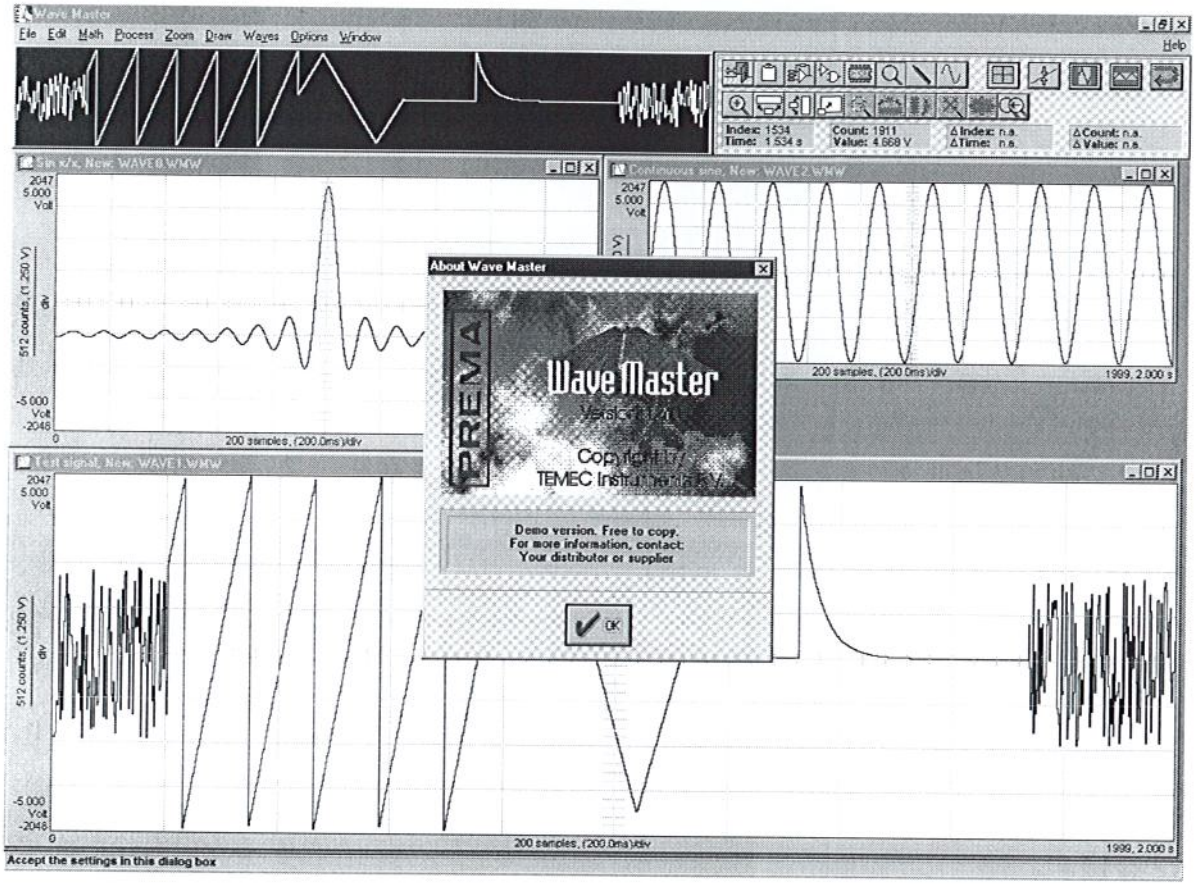
5025	IEEE-488 Interface Card PC2A for PC XT/AT incl. Drivers for C, Basic, QuickBasic, VisualBasic. More Drivers on request.
5023K	IEEE-488 Interface Cable, shielded, Length: 2m.

7.4 19-inch Rack Mounting Kit (5021 G)

Complete Slide-In module to mount the ARB 1000 into a 19-inch rack.
Height 2 HU.

The rack mounting kit is delivered with all essential screws.
An extra shield is also delivered if there is only one ARB 1000 in use.

7.5 Software Wavemaster



The way to create waveforms

Create waveforms easily using Wavemaster for Windows. Sketch, calculate, combine or load arbitrary waveforms in just a few minutes. Wavelengths up to 1048576 points x 16 bits are now available. Open as much windows as your computer has memory available.
Applications including biomedical signals, educational instruction, disk drive testing, power supply testing, noise, glitches, vibration waveforms and telephone / communication waveforms can be created easily from your PC using the windows environment you are familar with.
Loading from and to generators and Digital Storage Oscilloscopes was never this easy with the included IEEE-488 driver. As quickly as waveforms come to mind, you can create with wavemaster for windows.

Functional Overview

File Functions	EDIT Functions	MATH Functions	PROCESS Funct.	ZOOM Functions
New Wave	Undo	ADD	Normalize	Zoom In
Open	Copy	Subtract	Filter	Zoom In horizontal
Save	Cut	Multiply	Clip	Zoom In vertically
Save as	Paste	Divide	Duplicate	Window zoom
Save all	Insert paste	Invert	Expand to new	Zoom out
Load from device	Delete	AND	Change DAC type	Zoom out horiz.
Download to device	Select area	OR	Change wave set.	Zoom out vertically
Print	Select all	XOR		Unzoom
Printer setup	Hide selection	Rotate		Pan
Exit		Shift		Previous zoom

DRAW Functions	WAVES Functions	OPTIONS	WINDOW	HELP
Diagonal line	Constant	Devices	Cascade	Index
Orthogonal X-line	Noise	IEEE-488 setup	Tile	Using Help
Orthogonal Y-line	Sine	Adjust setup	Arrange icons	
Parabola	Cosine	Tracking	Close all	
	Triangle	Limit horizontal		
	Square	Limit vertical		
	Sawtooth			
	Ramp			
	User define			
	Rectangle pulse			
	Triangle pulse			
	Gaussian pulse			
	Havesine			
	Sinc pulse			
	Exponential pulse			

Index

—T—

—U—

—V—

—W—