GaGe

CompuGen SDK for MATLAB for Windows 95/98/NT

User's Guide

CompuGen SDK for MATLAB Version 1.00+
for MATLAB version 5.2

P/N: 0045118
Reorder #: MKT-SWM-CGML6
0404

Fifth Edition (December 1999)

How to reach Gage Applied Technologies, Inc.for Product Support
Toll-free phone: (800) 567-4243
Toll-free fax: (800) 780-8411

To reach Gage from outside North America
Tel: (514) 633-7447
Fax: (514) 633-0770

E-mail: prodinfo@gage-applied.com
Web site: www.gage-applied.com
On-line Support Request Form: www.gage-applied.com/support.asp

Table of Contents

Introduction

This manual provides information on the functionality of the CompuGen Analog SDK for MATLAB for Windows and the CompuGen Digital SDK for MATLAB for Windows.

Please note that this manual is not intended as a reference for any software other than the CompuGen SDKs for MATLAB. If you did not receive the correct guide, please contact the factory for a replacement.

It is assumed that the reader is familiar with using PCs, MS-DOS, Windows and PC - based cards. No description is included for these topics. If you are not comfortable with these areas, it is strongly recommended that you refer to guides such as the IBM PC AT reference manual before starting.

Gage's CompuGen Analog SDK for MATLAB for Windows and the CompuGen Digital SDK for MATLAB for Windows allows you to control one or more CompuGen 1100, CompuGen T30 and CompuGen 3250 boards from the MATLAB environment. The board(s) can be controlled either from an m file or interactively from the command prompt. Example m files are included on the distribution diskette.

The CompuGen SDK for MATLAB consists of a DLL (GGENCALL.DLL) which implements a MATLAB function that can be used from within MATLAB to control the board. This DLL communicates with the driver DLLs which come on the CompuGen Win NT or Win 95/98 Drivers distribution diskette and which in turn communicate with the actual hardware. Patterns can be created from within MATLAB and sent to the CompuGen board to be generated.

IMPORTANT NOTE: For the CompuGen SDK for MATLAB to work properly, you must have the CompuGen Win NT or Win 95/98 drivers installed, and your hardware must be configured properly. See *Installation* below for more information.

This guide assumes that the user is familiar with the use of MATLAB software; no information on this topic is included. For additional information, please refer to *Using MATLAB* or *Application Program Interface Guide*.

To maintain the accuracy of the information contained herein, we reserve the right to make changes to this manual from time to time.

Please note: For brevity, the following abbreviations are used in this manual:

- "CG1100" is used to refer to CompuGen 1100 ISA
- "CGT30" is used to refer to CompuGen T30 ISA
- "CG3250" is used to refer to CompuGen 3250 ISA
- "CompuGen SDK for MATLAB" is used to refer to both the CompuGen Analog SDK for MATLAB for Windows and the CompuGen Digital SDK for MATLAB for Windows.

What You Receive with the CompuGen SDK for MATLAB

The CompuGen SDK for MATLAB package consists of:

For CompuGen 1100 Boards:

- CompuGen Analog SDK for MATLAB for Windows.
- CompuGen SDK for MATLAB User's Guide.

For CompuGen 3250 and T30 Boards:

- CompuGen Digital SDK for MATLAB for Windows.
- CompuGen SDK for MATLAB User's Guide.

If you did not receive all of the above items, please contact the factory.

Running the Sample Programs

To use the sample programs, launch MATLAB and move to the directory where you've installed the CompuGen SDK for MATLAB files. The default directories are listed in the *Installation* section above.

You can run the example by typing in the name of the m file at the MATLAB command window prompt.

If you open the MATLAB Editor without changing the directory to where the example files are, MATLAB may not be able to find the files. If this is the case, the directory should be added to the MATLAB search path. This can be done in one of four ways:

1) Use the MATLAB Path Browser. This utility is accessed from the MATLAB command window in the File Menu\Set Path option. This will launch the Path Browser utility, which will allow you to change the MATLAB search path.

2) Edit the pathdef.m file. You can use any editor to edit the directory names in pathdef.m. This file is used by MATLAB on startup to determine the search path. The file is located in the \MATLAB\TOOLBOX\LOCAL.

3) Use the path command. If you wanted to add the directory d:\gage\test to the end of the current path, the syntax is:

```
path(path, 'd:/gage/test');
```

   Note the forward slashes.

4) Use the addpath command. The syntax is:

```
addpath d:/gage/test
```

   Again, note the forward slashes.

The first two methods are persistent. The last two apply only to the current MATLAB session.

The CompuGen SDK for MATLAB expects to find a configuration file, GAGE_GEN.INC, in your Windows directory. To change and verify these values you can use the CGWIN.EXE application, See your *CompuGen for Windows User's Guide* or *Software User's Guide* for more information.

Files Included With the CompuGen SDK for MATLAB

The files described below are installed with your CompuGen SDK for MATLAB. In addition, there is a DLLSRC directory which contains the C source code for the GGENCALL.DLL.

The sample programs GGENDEM1.M and GGENDEM2.M are installed with CompuGen Analog SDK for MATLAB for Windows and the sample programs CNT_T30.M and CNT3250.M are installed with CompuGen Digital SDK for MATLAB for Windows.

GGENDEM1.M and GGENDEM2.M

GGENDEM1.M is a sample M file which demonstrates controlling a CompuGen 1100 board with MATLAB. A sample is shown below. GGENDEM2.M is a sample m file for two CompuGen 1100 boards. Note that if there is only one board you can access the board structure either as board(1).opmode or board.opmode.

```
clear
board(1).opmode = 2;
board(1).sample_rate = 80000000;
board(1).output_range = 3;
board(1).output_filter = 1;
board(1).generate_once = 0;
board(1).t_source = 1;
board(1).t_slope = 0;
board(1).t_level = 128;
board(1).t_range = 1;
board(1).external_clock_level = 128;
board(1).external_clock = 0;
board(1).loop_number = 0;


theta = -pi:0.001:pi;
x = ((sin(theta) * 2048) + 2048);
theta = -pi:0.01:pi;
y = ((sin(theta) * 2048) + 2048);
x = x + 49152;
ggencall(0, 0, -1, board, x);

b = ggencall(1, 0, -1, board, x);
ggencall(2, 0, -1, board, x);
ggencall(3, 0, -1, board, x);
a = ggencall(5, 0, -1, board, x);
pause (5);
board(1).output_range = 1;
ggencall(2, 0, -1, board, x);
ggencall(3, 0, -1, board, x);
pause (5);
board(1).output_range = 3;
board(1).sample_rate = 40000000;
ggencall(2, 0, -1, board, x);
ggencall(3, 0, -1, board, x);
pause (5)
```

```
y = y + 49152;
for I = 1:100:5000
   x(I) = x(I) - 49152;
end
ggencall(3, 0, 1, board, x);
ggencall(3, 0, 1, board, y);
ggencall(6, 0, 1, board, x);
```

All the calls can also be done interactively from the MATLAB command prompt.

CNT_T30.M and CNT3250.M

CNT_T30.M is a sample M file which demonstrates controlling a CompuGen T30 board with MATLAB. CNT3250.M is a sample M file which demonstrates controlling a CompuGen 3250 board with MATLAB. In both the sample programs, the input pattern is a counter.

GGENCALL.DLL

This is the DLL that communicates between MATLAB and the CompuGen driver files. It should be in your current working directory. Typing GGENCALL with no parameters will display a help screen with information about the GGENCALL command (see below, *Using the GGENCALL Command*).

Using the GGENCALL Command

The GGENCALL command is how MATLAB communicates with the drivers for the CompuGen board. The format of the GGENCALL command is as follows:

```
ggencall (command, flag, board number, board structure, patterns);
```

Calling ggencall with no parameters will provide the following help screen:

```
You must specify a full GGEN command.
The syntax is: ggencall(command, flag, board #, board_structure, patterns)
The commands are: 0 - Do Everything.
                  1 - Initialize.
                  2 - Set up boards.
                  3 - Load and generate patterns.
                  4 – Reserved for future use.
                  5 – Reserved for future use.
                  6 - Abort boards.
```

Not all commands require all the parameters, though it does no harm and is safer to include them all for each call.

Command

The commands are:

0: Do everything.

```
boards_found = ggencall (0, 0, -1, board, x)
```

This command will perform a complete initialization, set up the boards, and load the pattern and generate cycle. The return value is the number of CompuGen boards found. See the descriptions below for more information on what is performed.

The second parameter is a flag allowing you to specify the operating mode for multiple boards: a value of 0 means Master/Slave, while a value of 1 means Multiple/Independent. By default, the driver assumes that multiple boards are in a Master/Slave configuration. If your multiple boards are in a Multiple/ Independent configuration, you must specify it via this flag.

1: Initialize.

```
boards_found = ggencall (1, 0, -1, board, x)
```

This command will initialize the CompuGen boards in the system. The return value is the number of CompuGens found and initialized. The DLL expects to find the configuration file, GAGE_GEN.INC, in the Windows directory. You can use the application CGWIN.EXE to configure the CompuGen boards.

The second parameter is a flag that allows you to specify the operating mode for multiple boards: a value of 0 means Master/Slave, while a value of 1 means Multiple/Independent. By default, the driver assumes that multiple boards are in a Master/Slave configuration. If your multiple boards are in a Multiple/ Independent configuration, you must specify it via this flag.

2: Set up the boards.

```
return_value  = ggencall (2, 0, 1, board, x)
```

The command takes the board structure and passes the values contained in it to the DLL, which sets up the hardware accordingly. The board variable is either a structure (for one board) or an array of structures (for more than 1 board), with the following fields:

```
board(1).opmode = 2;
board(1).sample_rate = 80000000;
board(1).output_range = 3;
board(1).output_filter = 1;
board(1).t_source = 1;
board(1).t_slope = 0;
board(1).t_level = 128;
board(1).t_range = 1;
board(1).external_clock_level = 128;
board(1).external_clock = 0;
board(1).loop_number = 0;
```

Note that if there is only one board, the structure can be accessed as board.opmode, etc. It should be initialized with valid values before calling **ggencall**. Though the DLL will use default values if none are provided, it is safer to initialize all the fields beforehand. The different fields of the structure will be explained in the next part of this manual.

If everything was set up correctly, the return value will be a 1. A return value of -1 indicates an error in either the operating mode or the sample rate. A return value of -2 indicates an error in the output range or filter. A return value of -3 indicates an error in one of the trigger parameters. An appropriate error message and error code will also be reported. Possible error codes are:

```
GGEN_NO_SUCH_MODE          0x01
GGEN_NO_SUCH_SAMPLE_RATE   0x02
GGEN_INVALID_SAMPLE_RATE   0x03
GGEN_NO_SUCH_BOARD         0x04
GGEN_NO_SUCH_GAIN          0x05
GGEN_NO_SUCH_TRIG_SLOPE    0x06
GGEN_NO_SUCH_TRIG_SOURCE   0x07
GGEN_LOAD_INVALID_CHANNEL  0x08
```

Note that the returned error codes should be converted to hexadecimal. The high byte indicates the board causing the problem and the low byte indicates one of the above error messages. For example, an error code of 263 is 107 in hexadecimal. This indicates board 1 has been set up with an invalid trigger source.

3: Load pattern into buffer and generate.

```
% Load board 1 with pattern x and generate.
ggencall (3, 0, 1, board, x);
% Load all boards with following patterns and generate.
% Board 1 will be loaded with pattern x, board 2 with pattern y and board
3 with pattern z.
ggencall(3, 0, -1, board, x, y, z); % Load all
```

The load and generate command is used to send a pattern to the CompuGen board's internal buffers and generate it. If a 0 or less is used for the board number parameter, there should be one pattern for each board. If the number of boards and the number of patterns differ, a message will be generated.

4: Reserved for future use.

5: Reserved for future use.

6: Abort the board(s).

```
ggencall(6, 0, 0, board, x)
```

This command is used to abort signal generation for all the CompuGen boards in the system.

Board Number

The number of the board on which to perform the command. A value of 0 or less means to perform it for all the boards. Note that some commands will be performed on all boards regardless of the board number. For example, command 0 (initialize) and command 1 (set up boards) are done for all boards. Other commands, for example command 7 (external clock level threshold), are performed only on the Master board. Currently, only command 3 (load buffer and generate) can be used with individual boards.

The Board Structure

The board structure is used to pass information on how to set up the board(s) and generation to the CompuGen drivers. The board structure should be defined and initialized either at the MATLAB command line or in an m file. The values don't take effect until ggencall is used with a command number 2. The format of the structure is as follows:

```
board.opmode = 2;
board.sample_rate = 80000000;
board.output_range = 3;
board.output_filter = 1;
board.t_source = 1;
board.t_slope = 0;
board.t_level = 128;
board.t_range = 1;
board.external_clock_level = 128;
board.external_clock = 0;
board.loop_number = 0;
```

board.opmode

This sets the operating mode of the CompuGen. Allowable values are:

2 - 12 bit mode (only for CompuGen 1100)
4 - 16 bit mode (only for CompuGen T30 and 3250)

board.sample_rate

This sets the sample rate for the CompuGen. The frequency generated will be the sample rate divided by the number of sample in the pattern. Allowable internal sample rates are:

1 Hz	1 kHz	1 MHz
2 Hz	2 kHz	2 MHz
5 Hz	5 kHz	5 MHz
10 Hz	10 kHz	10 MHz
20 Hz	20 kHz	20 MHz
50 Hz	50 kHz	40 MHz
100 Hz	100 kHz	80 MHz
200 Hz	200 kHz	
500 Hz	500 kHz	

If an invalid sample rate is chosen, an error message will be reported.

board.output_range

This field determines the range of the generated signal. Allowable values are:

Volts	Value
±5	1
±2	2
±1	3
±0.5	4
±0.2	5
±0.1	6

board.output_filter

This field sets the value of the internal output filter. Allowable values are:

Filter	Value
NO FILTER	0
20 MHz	1
5 MHz	3

board.generate_once

This field has been replaced by board.loop_number.

board.t_source

This field is used to set the trigger source. Allowable values are:

0 - External trigger
1 - Software trigger

External trigger means that signal generation will start when the trigger conditions have been met. Software trigger will cause a trigger in software to start the signal.

board.t_slope

This field is used to set the slope for an external trigger. This field is not used if the source is software trigger. Allowable values are:

0 - Positive slope
1 - Negative slope

board.t_level

This field is used to set the trigger level for an external trigger. This field is ignored if the trigger source is software. Allowable values are from 0 to 255, where 0 is the least positive value of the trigger range and 255 is the most positive. To convert from integer to Voltage, the following formula can be used:

Voltage = ((value - 128.0) / 128.0) * output range

If the trigger source is software, this field is ignored.

This field is valid only for the CompuGen 1100.

board.t_range

Used to set the external trigger range. Allowable values are:

Voltage	Value
±5 Volts	1
±1 Volt	3

The ±1 Volt range is valid only for the CompuGen 1100.

board.external_clock_level

This field is used to set the external clock level threshold. Allowable values are from -255 to 255. This represents a clock range of ±2.5 Volts. Anything below the external clock level threshold will not clock the board. The default value when the board is initialized is 1.2 Volts.

board.external_clock

This field is used to tell the driver that an external clock is being used. A one indicates that an external clock is being used, and a zero indicates that the internal clock is being used. If this field is set to 1, the board.sample_rate field is ignored.

board.loop_number

This field sets the number of times for the pattern to loop. A zero represents infinity, or continuous mode. A value of 4, for example, will generate the pattern 4 times.

Pattern

There should be one pattern buffer variable for each board found. Note that all boards can use the same buffer if you wish. The format of the buffer is integers from 0 to 4095, where 0 represents the most positive value in the output range and 4095 the least positive. For example, an array of all 0's in the ±1 Volt range would generate a +1Volt DC signal.

The length of the pattern to be generated must be divisible by 4. If it is not, the DLL will round the length down to the next highest number which is divisible by 4.

For the CompuGen 1100, the pattern is converted from doubles into unsigned 16 bit integers. For the CompuGen T30 and 3250, the pattern is converted into 32 bit integers.

Note that the current version of MATLAB (version 5.2) can only generate an array of doubles. The array should still be generated as integers from 0 to 4095, but they are represented in MATLAB as doubles (8 bit values). The

GGENCALL.DLL translates each of these buffers to an array of unsigned 16 bit integers and passes these buffers to the CompuGen board. Future versions of the drivers will allow for passing an unsigned integer array to the DLL as soon as MATLAB allows it. This is expected in the next version of MATLAB.

The CompuGen 1100 board provides a SYNC output to synchronize external hardware with the generated signal. The SYNC output is controlled by bits 14 and 15, which are called the SYNC bits. A zero in bits 14 and 15 of the pattern will cause the SYNC output to be high. To set the SYNC output to low, bits 14 and 15 should be set to one. That can be done by OR'ing the pattern with 0xC000 in the positions where you want the SYNC output to be low. In MATLAB this can also be accomplished by adding 49152 (0xC000 in decimal) to the pattern in the positions you want the SYNC output to be low. If they are low, subtracting 49152 will turn the SYNC output high again. For example:

```
% Turn off the synch bits in buffer x.
x = x + 49152;
% Turn on the synch bits every 100 samples in pattern x
for I = 1:100:5000
x(I) = x(I) - 49152;
end
```

Technical Support

Gage Applied Technologies, Inc. offers free technical support for all its drivers.

Technical support is available by phone at:

(800) 567-4243 (within North America)
(514) 633-7447 (all other locations)

from 9:00 A.M. to 5:00 P.M. Eastern Standard Time, Monday to Friday.

Support is also available by fax at

(800) 780-8411 (within North America)
(514) 633-0770 (all other locations)

or by e-mail at

prodinfo@gage-applied.com

Updated drivers are available at Gage's Web site:

http://www.gage-applied.com

When calling for support we ask that you have the following information available:

1. Version and type of your Gage drivers.
(The version number can be obtained at the top of any of the driver source files or on the distribution diskette.)

2. Type, version and memory depth of your Gage card.

3. Type and version of your operating system.

4. Type and speed of your computer and bus.

5. Contents of your CONFIG.SYS and AUTOEXEC.BAT files.

6. Any extra hardware peripherals (i.e. CD-ROM, joystick, network card, etc.)

7. Were you able to reproduce the problem with CGWIN.EXE or CGT30.EXE?

If the problem is with an application program you are writing, the simplest approach is often to send us some of the code you are having problems with, along with other details such as sample rate, trigger source, input signal, etc., either by fax or e-mail. This way, we can try to reproduce the problem.

Gage Products

For ordering information, see Gage's product catalog, or visit our web site at http://www.gage-applied.com

CompactPCI Bus Products	CompuScope 85GC	8 bit, 5 GS/s Analog Input Card
	CompuScope 82GC	8 bit, 2 GS/s Analog Input Card
	CompuScope 14100C	14 bit, 100 MS/s Analog Input Card
	CompuScope 1610C	16 bit, 10 MS/s Analog Input Card
	CompuScope 3200C	32 bit, 100 MHz Digital Input for CompactPCI Bus
PCI Bus Products	CompuScope 1610	16 bit, 10 MS/s Analog Input Card
	CompuScope 1602	16 bit, 2.5 MS/s Analog Input Card
	CompuScope 14200	14 bit, 200 MS/s Analog Input Card
	CompuScope 14105	14 bit, 105 MS/s Analog Input Card
	CompuScope 14100	14 bit, 100 MS/s Analog Input Card
	CompuScope 1450	14 bit, 50 MS/s Analog Input Card
	CompuScope 12100	12 bit, 100 MS/s Analog Input Card
	CompuScope 1250	12 bit, 50 MS/s Analog Input Card
	CompuScope 1220	12 bit, 20 MS/s Analog Input Card
	CompuScope 85G	8 bit, 5 GS/s Analog Input Card
	CompuScope 82G	8 bit, 2 GS/s Analog Input Card
	CompuScope 8500	8 bit, 500 MS/s Analog Input Card
	CompuScope 3200	32 bit, 100 MHz Digital Input for PCI Bus
CompuGen	CompuGen 1100	12 bit, 80 MS/s Analog Output Card
	CompuGen 3250	32 bit, 50 MHz Digital Output Card
Application Software	GageScope	World's Most Powerful Oscilloscope Software
	GageBit	Digital Input/Output Software
	CompuGen for Windows	Arbitrary Waveform Generator Software for Windows
Software Development Kits	CompuScope SDK for C/C++ for Windows	
	CompuScope LabVIEW SDK for Windows	
	CompuScope MATLAB SDK for Windows	
	CompuScope LabWindows/CVI SDK (for CompactPCI/PXI bus CompuScope cards)	
	CompuGen Analog SDK for C/C++ for Windows	
	CompuGen Digital SDK for C/C++ for Windows	
	CompuGen Analog LabVIEW SDK for Windows	
	CompuGen Digital LabVIEW SDK for Windows	
	CompuGen Analog MATLAB SDK for Windows	
	CompuGen Digital MATLAB SDK for Windows	
Instrument Mainframes	Instrument Mainframe 7000 Instrument Mainframe 2000	Instrument Mainframes for Housing CompuScope and CompuGen Products.
	Instrument Mainframe 8000C	Instrument Mainframes for Housing CompactPCI/PXI CompuScope Products.

GaGe

GAGE APPLIED TECHNOLOGIES, INC.

Head Office

Tel: 514-633-7447
Fax: 514-633-0770

Toll free tel: 800-567-4243
Toll free fax: 800-780-8411

E-mail: prodinfo@gage-applied.com
Web: http://www.gage-applied.com