Goebel High Definition Analog Output Software Library and Utilities

User Manual

The Goebel Company

Revision: 1.2.1 **August 25, 2016**

Purpose

This manual describes the software for high definition analog output boards offered by The Goebel Company. This includes application programming interface library and management applications.

Notice

Information in this manual has been carefully reviewed and is believed to be accurate. The Goebel Company shall not be liable for errors contained herein. The Goebel Company reserves the right to make changes or additions to the software described herein.

Contact

For technical or other inquiries contact:

email: support@GoebelEtc.com

The Goebel Company
12486 Prowell
Leavenworth WA 98826
USA
Phone: 206-601-6010

Table of Content

1 Introduction

Goebel supplies a high definition analog output board with 8 channels. Each channel is configurable for variable gain, update rate, and waveform generation. The low gain setting provides highly accurate thermocouple output. The wave form generator provides continuous frequency control from zero to 50Khz for repeating waveforms.

This manual describes the API for part number GIO-HDAO-P8. These are universal PCI cards, meaning they operate in legacy 5v 32/64 bit PCI, 3.3v 32/64 bit PCI and PCI-X slots at 33 MHz. All channels are accessed by a DSUB 37 connector on the front panel of the card. Cards are numbered via dip switch for positive identification.

1.1 Advanced Features

Goebel provides an 8 channel high definition analog output suitable for multiple uses. Below are highlighted the main features of

- suitible for low voltage thermocouple outputs.
- general purpose analog output.
- waveform generation.

Card feature include:

- card ID via dip switch for positive idendtification

Software features include:

- shared software access to card channels from multiple application.

1.1.1 Configurable gain

Channels which are configurable as to gain with low gain settings sutable for thermocouple output. Gain are configurable from low voltage thermocouple levels to +/- 10V.

1.1.2 Card id

A dip switch block is configured with a card id. The benefit is that individual cards are positively identified. A card not seen on the bus can be positively identified with it's card id. There is never a danger of output going to a card other than the one intended for.

1.1.3 Adjustable Output Rate

The output rate is continuously adjustable from 1 usec to 1,000,000 usec.

1.1.4 Driver Features

The driver supports shared access to hardware by channel from multiple applications.

1.2 Change log

1.2.1 Revision 1.2.1-a

- Resolved issue with sine wave clamped at 0 or negative max for 1/2 cycle when updating frequency.

1.2.2 Revision 1.1.1-a

- ghdao routine will use first card found for test instead of hardcoded value.
- XML and code change to accept altitude_max instead of max_altitude per documentation.

1.2.3 Revision 1.1.0-a

First working version.
Add routines to work with the GHDAO-8 card

1.2.4 Revision 1.0.0-a

This is the initial revision for Linux drivers. The driver package name is ghdao and is avalable as Linux RPM.

2 HDAO Interface Overview.

The API interface device independent layer common to all Goebel devices and transportable between Windows and Linux.

2.1 Board Devices

Each output channel on a card can be accessed by the board device. To use a channel the board device must be opened and the channel configured.

2.1.1 Board

Opening the board device allows access to all channels.

2.2 Channel modes

2.2.1 Themocouple mode

In thermocouple mode the range is automatically set for the thermocouple type. Output can be specified as temperature, or voltage. The thermocouple temperature output is changed as often as a new teperature of voltage is specified.

2.2.2 Analog output mode

In analog output mode, the output voltage is changed each time the channel voltage is changed.

2.2.3 Waveform generation

In waveform generation mode, the waveform type and frequency are given. Frequency can be changed as often as desired. A new frequency takes immediate affect. Standard waveforms are available for sine wave generation. User specified waveforms can be specified as 32k data points.

2.3 Multi user interface

Multiple programs or threads can access board or channel devices simultaneously. This means multiple programs can run independently accessing only the channels they are interested in.

If multiple programs are accessing the board, they should not be accessing the same channels.

3 Application Programming Interface

The driver interface is largely the same whether the board device or channel device is opened.

3.1 Control Functions

3.1.1 gdev_open

Synopsis

```
#include "gdev.h"

gdev_hdl_t
gdev_open(char *board, char *device, char *options);
```

Description

This function establishes the connection to a hdao board. Boards are named “hdao/0” – “hdao/15”. The board number is configured in the boards dip switch set at install time. Boards are shipped as with board number 1 and need not be changed unless multiple boards are present. The device name is not used for this board.

Parameters

board:	board name “ghdao/N” where N is the dip switch number 0 to 15.
device	must be **NULL**.
options:	“w” as write mode is the only mode supported.

Returns

Handle **hdl** or GDEV_FAILED in case of an error.

Errors

EINVAL:
- **board**, **device** or **options** is invalid.
- **Board** must exist in the system.
- **Device**, must be null.
- **options** must be “w”.

ENODEV:
- No operational device was found with the given **board** name.

3.1.2 gdev_close

Synopsis

```
#include "gdev.h"

int
gdev_close(gdev_hdl_t hdl);
```

Parameters

hdl:	handle returned on gdev_open.

Returns

Return 0 or GDEV_FAILED in case of an error.

Errors

EBADF fildes is not a valid, active file descriptor.

3.1.3 gdev_command

Synopsis

#include "gdev.h"

gdev_hdl_t
gdev_command(gdev_hdl_t **hdl**, char ***fmt,** ...);

Description

This function passes command strings to the driver for a variety of device controls. See section 4 for device controls. The command strings are in an XML element like format. This means the format string results in the following general structure:

<command param1=value1 param2=value2 … paramN=valueN />

The handle returned by gdev_open is used to identify the board to apply the command to.

Parameters

hdl: value returned by gdev_open.
fmt See command strings from section 4.

Returns

GDEV_SUCCESS or GDEV_FAILED in case of an error.

Errors

EINVAL:
hdl or command is invalid.

4 Commands for gdev_command

This section describes the command strings passed to gdev_command. Command strings are used to provide a flexible method of parameter passing to the driver.

4.1 Syntax rules

Required option denoted by parameters within {}

Alternate option seperated by |

Optional parameter denoted by parameters within []

4.2 start/configure

Start and confugure have the same parameter set so that configureing and starting can be done with a single command. Start can also be done independently without specifying channel configure parameters.

Synopsis

For waveform generation:
<configure channel=C waveform=sine frequency=F amplitude=A amplitude_max=Amx />
<start channel=C />

For thermocouple output
<configure channel=C thermocouple=K units=U temperature=T />
<start channel=C />

For analog output
<configure channel=C range_min=R_{mn} range_max=R_{mx} volts=V/>
<start channel=C />

Description

This function string starts or configures the specified channel. The channel should be configured prior to start or include the configure parameters in the start command. The default state for unconfigured channels is zero volts output.

Parameters

channel=C **C** is a value between 1 and 8.

Thermocouple Output Parameters

thermocouple=K where K is the thermocouple type.
When thermocouple type is selected, temperature is specified instead of voltage.
Valid thermocouple types are B, E, J, K, N, R, S, T.
units=U U (celsius or fahrenheit), can be specified for thermocouples, (default=CELSIUS).
Temperature=T Temperature when thermocouple mode is selected.

Waveform Parameters

Waveform=W W is currenlty limited to sine.
Amplitude=A A specifies the current amplitude of the waveform in volts.
Amplitude_max=A_{mx} A_{mx} specifies the maximum amplitude of the waveform in volts.

frequency=F F is the frequency in hertz from 0 to 200K. Frequencies above 50k may show steps in the waveform. Frequency can be changed dynamically with the update command.

Analog Output Parameters

range_min=R_{mn} R_{mn} is a number between -10.0 and 10.0.
This is the minimum voltage that will be output.
Specifying a voltage below this value results in R_{mn} being applied.

range_max=R_{mx} R_{mx} is a number between -10.0 and 10.0.
This is the maximum voltage that will be output.
Specifying a voltage above this value results in R_{mx} being applied.

volts=V V is a value between -10.0 and +10.0.

Errors

EINVAL:
if start command and already started

4.3 stop

Synopsis

<stop [channel=C] />

Description

This function stops the channel if specified, or the board as a whole.

Parameters

channel=C **C** is a value between 1 and 8.

Errors

EINVAL:
if already stopped

4.4 update

Synopsis

For waveform updates:
<update channel=C frequency=F amplitude=A />

For thermocouple output
<update channel=C temperature=T />

For analog output updates:
<update channel=C volts=V />

Description

The frequency and voltage can be updated dynamically for waveform generation. Voltage updates happen at the next zero crossing.

Voltage updates of analog outputs take place when the update command is issued.

Parameters

Amplitude=A A specifies the current amplitude of the waveform.

frequency=F F is the frequency in hertz from 0 to 200K. Frequencies above 50k may show steps in the waveform. Frequency can be changed dynamically with the update command.

Temperature=T Temperature when thermocouple mode is selected.

volts=V V is a value between -10.0 and +10.0.

5 Example programs

5.1 Example waveform generation

```
#include <goebel/gdev.h>

gdev_hdl_t      hdl;
int       channel=1;
float     freq_min = 0.0;
float     freq_max = 50000.0;

hdl       = gdev_open("ghdao/1", NULL, "w");

status    = gdev_command(hdl, "<start channel=%d waveform=sine frequency=1.0
                range_min=-5.0 range_max=5.0 />", channel);

frequency       = 1.0;
update          = 1.0;
while (1) {
          usleep(10000);
          if (frequency >= freq_max)
                    update  = -1.0;
          if (frequency <= freq_min)
                    update = 1.0;
          frequency       += update;
          gdev_command(hdl, "<update channel=%d frequency=%f />",channel,frequency);
}
```

6 GHDAO Utilities

6.1 ghdao test program

This is an internal test program that demonstrates various features of a Goebel hdao PCI board. It is provided as a basic test program to validate board functionality. In addition source code is provided in the hope that it may prove useful as an example for programming. It is provided on an as-is basis, and is not intended for production use. As such this documentation is incomplete and not all features are present or functional. That said it is provided in the hope that it may prove useful for certain test uses.

> ghdao

```
enter test type:
        library   Show library revision
           regs   register dump
     freq_sweep   Frequency Sweep Test
          gtest1  Gain Test
          analog  Analog Test
         tcouple  Thermocouple Test
            off   All channels OFF
     analog_test  Test All channels in ANALOG_MODE
     tcouple_test Test All channels in TCOUPLE_MODE
      sine_test   Test All channels in WAVEFORM_MODE
```

This shows the options of the program. Entering an option, and you will be prompted for additional parameters. Default values are selected by entering <cr>.

All options may not be functional, and some options require the presence of data files and cabling.

6.2 Documentation.

/usr/local/goebel/docs.

6.3 Example programs

Example source code can be found in /usr/local/goebel/ghdao/examples.

7 Installation

7.1 Linux

Linux software distributions consist of rpm or srpm files.

Software is installed by default in:

/usr/local/bin	ghdao test programs.
/usr/local/lib	libghdao.a
/usr/local/include/goebel	hdao.h include file
/usr/local/goebel/docs	documentation
/usr/local/goebel/hdao/examples	example programs

rpm –U –force ghdao-<version>.rpm

```
>sudo rpm -U --force ghdao-1.1.0-a.centos.el5.rpm
>sudo rpm -U --force ghdao-i686-1.1.0-a.centos.el5.rpm
```

Once installed, the user can verify the package was installed using the rpm command.

```
>rpm -qi ghdao

[gsim:/usr/src/redhat/RPMS/x86_64]> rpm -qi ghdao
Name        : ghdao                         Relocations: /usr/local
Version     : 1.1.0                             Vendor: Goebel http://www.goebel.aero
Release     : a                              Build Date: Sat 20 Aug 2016 04:58:18 PM
MST
Install Date: Sat 20 Aug 2016 04:59:03 PM MST      Build Host: gsim
Group       : System Environment/Kernel     Source RPM: ghdao-1.1.0-a.src.rpm
Size        : 971313                           License: Proprietary
Signature   : (none)
Packager    : Goebel http://www.goebel.aero
URL         : http://support.goebel.aero/
Summary     : Goebel ghdao Linux driver
Description :
GA429 - Goebel High Definition Analog Output driver

This package is compiled against kernel %{kverrel}.
```

7.1.1 Installation verification

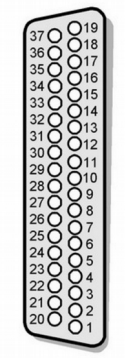
Verification of installation is accomplished by running the ghdao program (found in \usr\local\bin). Use the following tests to verify software functionality. No cable need be connected for this test.

```
>ghdao regs
```

7.2 Windows

Windows installers are available on your companies web support area, under Window/ghdao/ghdao-<rev>.exe

8 Pinout



DSUB 37 Pin Number	Signal	DSUB 37 Pin Number	Signal	Driver Channel
1	Analog Out 0	20	Analog Out Return 0	Channel 1
2	Chassis GND	21	Chassis GND	
3	Analog Out 1	22	Analog Out Return 1	Channel 2
4	Chassis GND	23	Chassis GND	
5	Analog Out 2	24	Analog Out Return 2	Channel 3
6	Chassis GND	25	Chassis GND	
7	Analog Out 3	26	Analog Out Return 3	Channel 4
8	Chassis GND	27	Chassis GND	
9	Analog Out 4	28	Analog Out Return 4	Channel 5
10	Chassis GND	29	Chassis GND	
11	Analog Out 5	30	Analog Out Return 5	Channel 6
12	Chassis GND	31	Chassis GND	
13	Analog Out 6	32	Analog Out Return 6	Channel 7
14	Chassis GND	33	Chassis GND	
15	Analog Out 7	34	Analog Out Return 7	Channel 8
16	Chassis GND	35	Chassis GND	
17	No Connect	36	No Connect	
18	Chassis GND	37	Chassis GND	
19	No Connect			