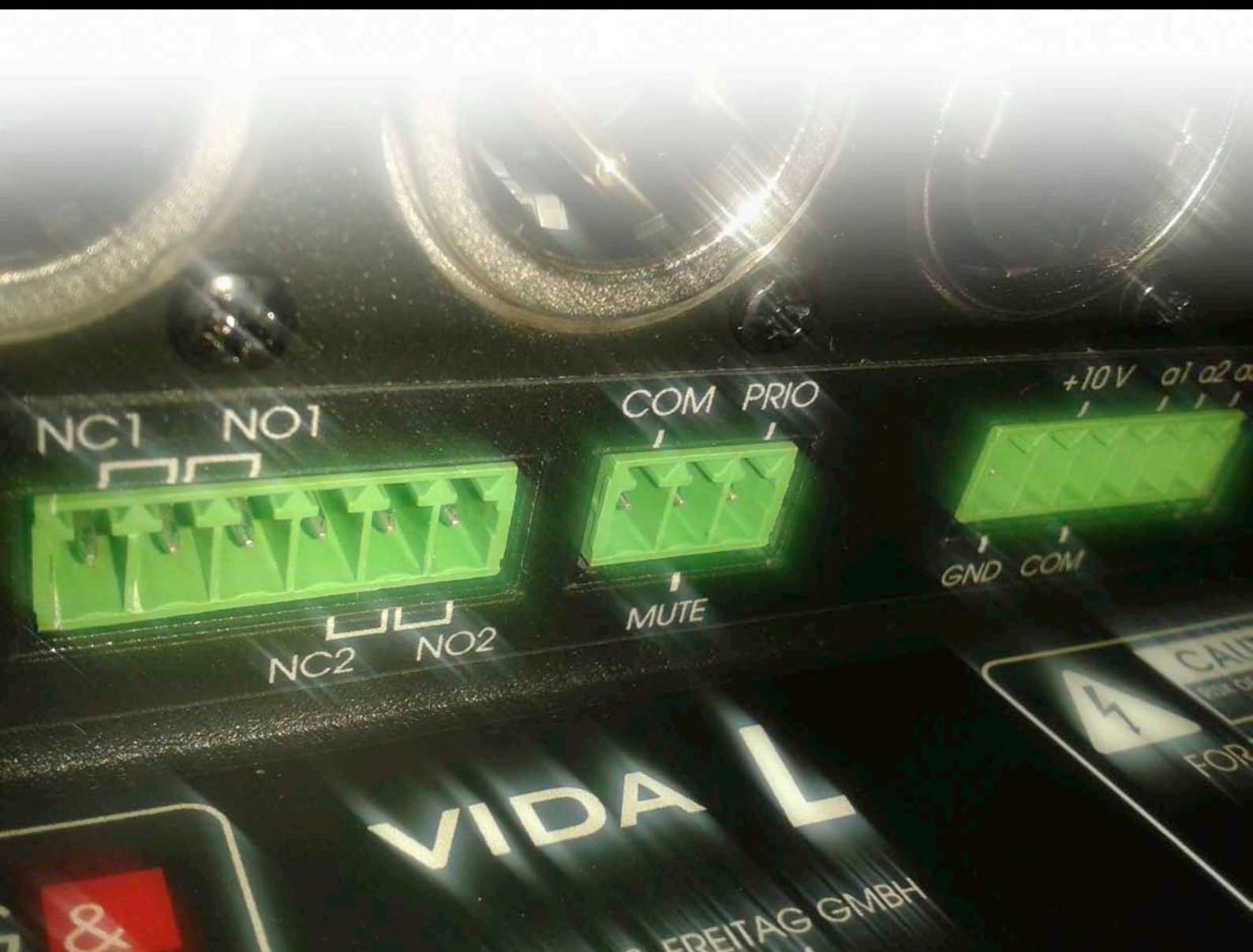


VIDA GPIO/SGPIO



Technical Information

Version 1.0
Released: 26.09.2016



**Important Information,
Please Read Before Use!**

KLING & FREITAG GmbH
Wohlenbergstrasse 5
D-30179 Hannover
TEL +49 (0) 511 96 99 70
FAX +49 (0) 511 67 37 94
www.kling-freitag.de

KLING &
FREITAG

SOUND SYSTEMS

Table of contents

1	General Notes	4
2	GPI / SGPI	4
2.1	GPI	5
2.2	SGPI	6
3	GPO / SGPO	9
3.1	GPO	9
3.2	SGPO	9
4	Appendix: JSON Sample Code	10

1. General Notes

The K&F VIDA speaker system offers 5 standard GPIOs plus 8 software-controlled GPIOs (“SGPIOs”).

GPIOs and SGPIOs allow for receiving and evaluating control commands, for example, from media controllers. Using GPIOs and SGPIOs, you can query pre-defined VIDA statuses.

For this purpose, the GPIOs must be wired to apply voltages while the SGPIOs receive JSON messages through a network port. The VIDA evaluates the signals and/or JSON messages and triggers actions preconfigured by the user.

SGPIOs behave just like normal GPIOs but are accessed using JSON messages on the network. Their main benefit is that you don’t need to wire them individually.

To be able to use SGPIOs, be sure to connect the VIDA to the mains as well as to a computer network using a suitable network cable. You can send messages through the IP network to the VIDA using an Internet-enabled browser, some other piece of suitable software, or a suitable media controller.

2. GPI / SGPIO

You can separately configure the response to a voltage change for each GPI as well as to a JSON message for each SGPIO. On the left-hand side of the configuration window, set the action triggered by a change from low to high; those for high-to-low changes are set on the right-hand side.



Signal Interpretation:

Signal	GPIOs (hardware)	SGPIO (software)
Low	< 1.5 V	0
High	> 5 V	1

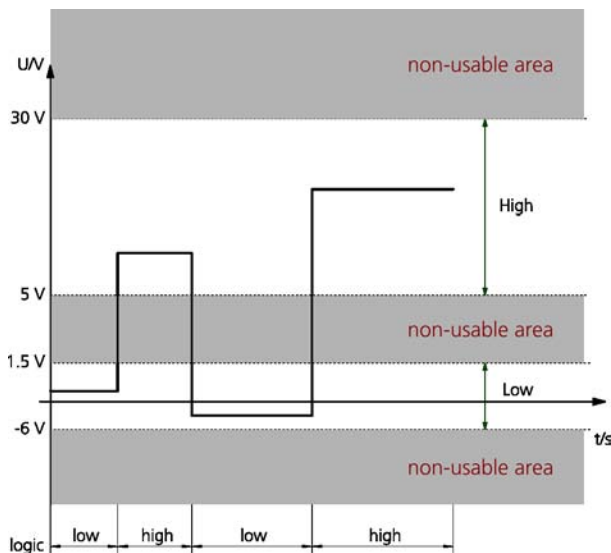
The GPIOs and SGPIOs allow for configuring 25 functions for each high-to-low and low-to-high changes.

#	Setting	Effect
1	Mute Toggle	Toggles the master-mute button: mute > unmute unmute > mute
2. – 11.	Change Volume (+0,1 dB, -0,1 dB, +0,5 dB, -0,5 dB, +1,0 dB, -1,0 dB, +5,0 dB, -5,0 dB, +10,0 dB, -10,0 dB)	Changes the master volume by the specified value.
12	Select Input Analog	Selects the analog input.
13	Select Input Dante 1	Selects Dante input 1.
14	Select Input Dante 2	Selects Dante input 2.
15	Select Input AES 3 L	Selects AES input L.
16	Select Input AES 3 R	Selects AES input R.
17	Select AES Input L+R	Selects AES input L+R.
18. – 25.	Load Preset (1–8)	Loads the respective preset.

2.1 GPI

The GPIs are designed as floating optocoupler inputs.

Voltages of less than 1.5 V reliably put the optocoupler into the Off state. The reverse voltage is -6 V. Always make sure not to exceed the maximum negative voltage of 6 V.



The current drawn at 10 V is 2.5 mA.

The 10-V output is a galvanically isolated (floating) DC/DC converter featuring a current limit at approx. 15 mA.

If no external voltage source is used, you can use the converter either for controlling the GPIOs or for indicating the GPIO statuses, for example, using a low-current LED.

In addition to the 3 standard GPIOs, there are 2 extra "emergency" GPIOs:

1. **Mute button:** Allows for adjusting whether mute is enabled by the high or low state. Enabling will mute the system – you cannot unmute it using the app nor the mute toggles of other GPIOs.
2. **Priority:** Here you can select the audio input that is switched to when the GPIO is enabled. This state disables both input gain and input mute, i.e. the signal is routed to the amplifiers at 0-dB level regardless of the gain setting and the mute status.

2.2 SGPI

SGPIs are provided for integrating VIDA components with media controllers.

The SGPIs are configured in the same way as their hardware counterparts. You can set and query them using a browser or a similar HTTP-enabled piece of software. The VIDA web interface will always respond with JavaScript Object Notation (JSON) messages.

With HTTP, a query call always looks like this:

```
<IP>/cgi-bin/vida_webservice.cgi?action=<ACTIONNUMBER>&<ATTRIBUTE>
```

Where:

- <IP> is replaced by the IP address of the VIDA to be queried
- <ACTIONNUMBER>&<ATTRIBUTE> are replaced by the "action=", "number=", and/or "value=" keywords and one of the specified numeric values (see the Attribute column)

Sample query:

```
http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=201&number=0
```

("http://192.168.110.137/cgi-bin/vida_webservice.cgi?" is always the start of a query—this is why the table includes the corresponding keyword values)

Action #	Description	Attribute	Example	Response
201	Querying a software GPO	Number(0–7) 0 = SGPO 1 1 = SGPO 2 ... 7 = SGPO 8	action=201&number=0	{"SUCCESS": "0"} or {"SUCCESS": "1"} or error message
202	Querying a software GPI	Number(0–7) 0 = SGPI 1 1 = SGPI 2 ... 7 = SGPI 8	action=202&number=0	{"SUCCESS": "0"} or {"SUCCESS": "1"} or error message
203	Enabling a software GPI	Number(0-7) value(0/1) 0 = SGPI 1 1 = SGPI 2 ... 7 = SGPI 8 Value 0 is low Value 1 is high	Action=203&number=0&value=1 or action=203&number=0&value=0	{"SUCCESS": "0"} or {"SUCCESS": "1"} or error message

211	Querying a GPO	Number(0-1) 0 = GPO 1 1 = GPO 2	action=211& number=0	{"SUCCESS": "0"} or {"SUCCESS": "1"} or error message
212	Querying a GPI	Number(0-2) 0 = GPI 1 1 = GPI 2 2 = GPI 3	action=212& number=0	{"SUCCESS": "0"} or {"SUCCESS": "1"} or error message
213	Querying PRIO mute	action=213		{"SUCCESS": "0"} or {"SUCCESS": "1"} or error message
214	Querying PRIO input		action=214	{"SUCCESS": "0"} = Off {"SUCCESS": "1"} = Analog {"SUCCESS": "2"} = AES 3 R {"SUCCESS": "3"} = AES 3 L {"SUCCESS": "4"} = AES 3 L/R {"SUCCESS": "5"} = DANTE 1 {"SUCCESS": "6"} = DANTE 2 or ERROR

Error message {"ERROR": "[message text]"}



Tip

To allow for secure communication, you might want to assign a static IP address to the VIDA before using the web service.

When enabling an SGPI, remember that the action will occur **at the time of the state change** from high to low or low to high depending on the configuration.

For example, if you have assigned the Change Volume +1.0 dB function to a change from low to high at SGPI 1, you will need to submit the following messages to increase the

Message	Details
<code>http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=203&number=0&value=0</code>	This ensures that the status is low.
<code>http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=203&number=0&value=1</code>	Changes from low to high (+1 dB)
<code>http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=203&number=0&value=0</code>	Resets to low.
<code>http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=203&number=0&value=1</code>	Changes from low to high (+1 dB)
<code>http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=203&number=0&value=0</code>	Resets to low.
<code>http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=203&number=0&value=1</code>	Changes from low to high (+1 dB)

Now the VIDA's level has been increased by 3 dB!



Tip

To be on the safe side, reset the SGPI to low again to make sure another action can be triggered by a low-to-high change.

`http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=203&number=0&value=0`

Before transmitting a message, you need to make sure that the previous action has been completed—otherwise, the system will issue an error message.

A loop for querying the VIDA in regular intervals needs to be programmed to stay up to date about the GPO statuses. Again, be sure to wait until the previous action has been completed.

For more information on JSON, refer to

- <https://en.wikipedia.org/wiki/JSON>

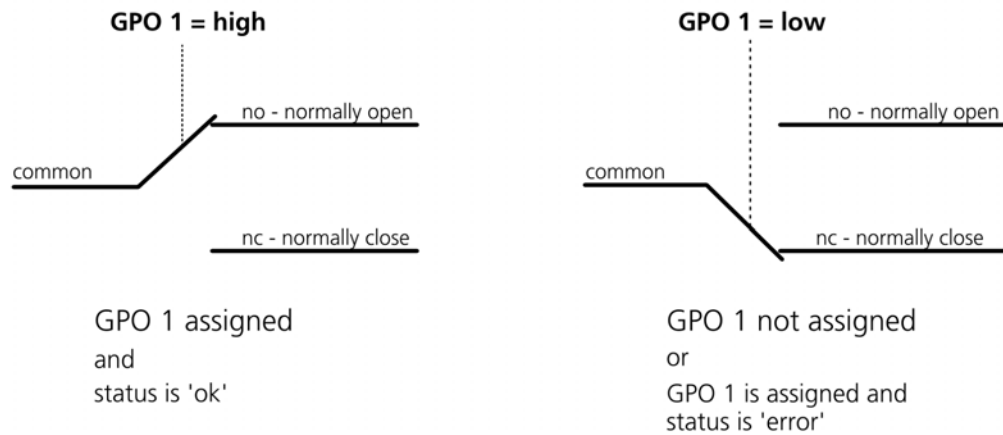
3. GPO / SGPO

3.1 GPO

The GPOs are floating MOSFETs supporting NO (normally open) and NC (normally closed).

If the outputs have been enabled using the software, NO is low-resistance and NC is high-resistance to common.

In case of errors, NO is high-resistance and NC is low-resistance.



Resistive Load:

- 2 A at up to 30 V AC/DC
- 1 A at up to 60 V AC/DC
- 600 mA at up to 110 V AC

3.2 SGPO

The GPOs allow for querying 5 statuses:

1. **Off:** The SGPO is disabled. The state is always "0."
2. **System On:** SGPO becomes "1" when the VIDA has powered up successfully and is ready for operation.
3. **System Status (high if config is wrong):** SGPO becomes "0" when the configuration is faulty. When enabled, the red front-panel indicator will light.
4. **System Overtemp:** SGPO becomes "1" when the temperature of an amp exceeds 70 °C or the power-supply temperature is 5 °C or less within the cutoff temperature.
5. **System Hardware Fault:** SGPO becomes "1" if
 - 'Protect' is reported or the speaker is not connected
 - at amp-DAC error
 - at amp error

4. Appendix: JSON Sample Code

```

// Remarks
// The JSON Component implements a standard JSON client.
// The component contains a number of properties that map directly to HTTP request headers.
//-----

void __fastcall TMainForm::GetGPI(BYTE Nr)
{
    JSON->Post("http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=212&number=" + IntToStr(GpiNr));
    JSON->DoEvents();
    if(JSON->TransferredData.Pos("SUCCESS\\": \\\"0\") > 0)
        CheckBox[GpiNr]->IsChecked = false;
    if(JSON->TransferredData.Pos("SUCCESS\\": \\\"1\") > 0)
        CheckBox[GpiNr]->IsChecked = true;
}
//-----

void __fastcall TMainForm::GetPrioMute(BYTE Nr)
{
    JSON->Post("http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=213");
    JSON->DoEvents();
    if(JSON->TransferredData.Pos("SUCCESS\\": \\\"0\") > 0)
        CheckBox[GpiNr]->IsChecked = false;
    if(JSON->TransferredData.Pos("SUCCESS\\": \\\"1\") > 0)
        CheckBox[GpiNr]->IsChecked = true;
}
//-----

void __fastcall TMainForm::GetPrioInput(BYTE Nr)
{
    JSON->Post("http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=200");
    JSON->DoEvents();
    if(JSON->TransferredData.Pos("SUCCESS\\": \\\"0\") > 0)
        CheckBox[GpiNr]->IsChecked = false;
    if(JSON->TransferredData.Pos("SUCCESS\\": \\\"1\") > 0)
        CheckBox[GpiNr]->IsChecked = true;
}

//-----
// Timer wird alle 50ms getriggert
void __fastcall TMainForm::Timer1Timer(TObject *Sender)
{
    if(!JSON->FIdle())
    {
        return;
    }
    GpiNr++;
    if(GpiNr > 4)
    {
        GpiNr = 0;
    }
    if(GpiNr <= 2)
        GetGPI(GpiNr);
    if(GpiNr == 3)
        GetPrioMute(GpiNr);
    if(GpiNr == 4)
        GetPrioInput(GpiNr);
}
//-----

```