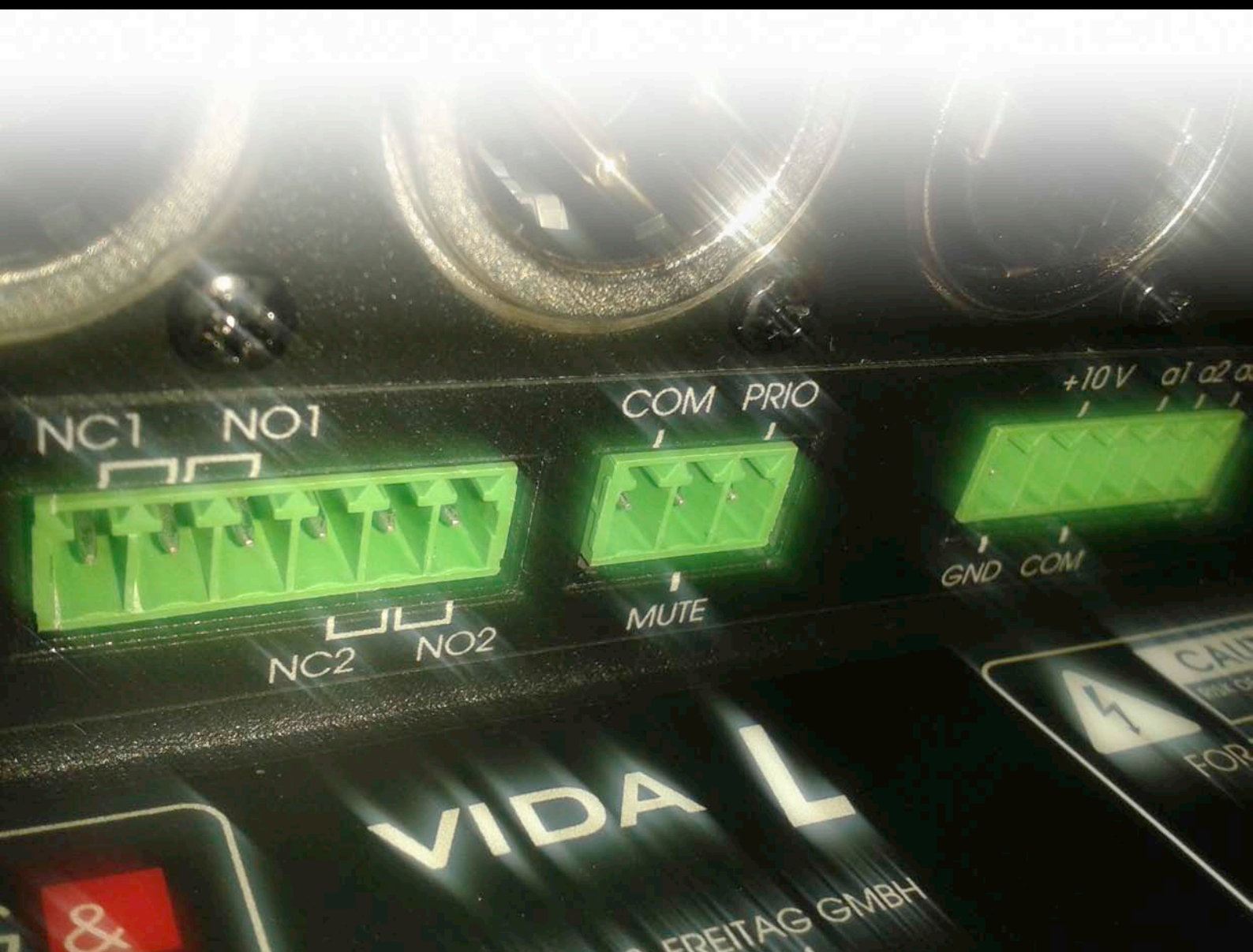


VIDA GPIO/SGPIO



Technische Information

Version 1.0
Stand: 08.09.2016



**Wichtige Informationen, vor
Inbetriebnahme lesen!**

KLING & FREITAG GmbH
Junkersstraße 14
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

Inhaltsverzeichnis

Kapitel / Abschnitt		Seite
1	Allgemeines zu VIDA GPIO/SGPIO	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	Anhang, JSON-Codebeispiel	10

1. Allgemeines zu VIDA GPIO/SGPIO

Im K&F VIDA-Lautsprechersystem gibt es fünf normale GPIOs und acht softwaregesteuerte GPIOs, sogenannte 'SGPIOs'.

Mit den GPIOs und SGPIOs können Steuerungsbefehle von z.B. einer Mediensteuerung empfangen und ausgewertet werden. Über die GPIOs und SGPIOs können definierte Zustände der VIDA abgefragt werden.

Dazu müssen GPIOs über eine entsprechende Verkabelung Spannungen und SGPIOs über einen Netzwerkanschluss entsprechende JSON-Befehle empfangen können. Die Signale und/oder JSON-Befehle werden von der VIDA ausgewertet und lösen so vom Benutzer vorgegebene Aktionen aus.

SGPIOs verhalten sich wie normale GPIOs, sie werden jedoch über das Netzwerk mit 'JSON'-Befehlen angesprochen. Das hat den Vorteil, dass sie nicht wie GPIOs einzeln verkabelt werden müssen.

Um die SGPIOs nutzen zu können, muss die VIDA am Stromnetz und mit einem geeigneten Netzwerkkabel an ein Netzwerk angeschlossen sein. Über einen Internetbrowser, einer anderen geeigneten Software oder einer geeigneten Mediensteuerung können der VIDA über das IP-Netzwerk entsprechende Befehle gesendet werden.

2. GPIO/SGPIO

Für jeden GPIO und SGPIO lässt sich eine Reaktion auf eine Änderung der Spannung (GPIO) oder auf eine JSON-Befehlszeile (SGPIO) einstellen. Im Konfigurationsfenster kann auf der linken Seite die Aktion eingestellt werden, die bei einer Signaländerung von LOW zu HIGH ausgeführt wird. Auf der rechten Seite des Fensters kann die Aktion eingestellt werden, die bei einer Änderung des Signals von HIGH zu LOW ausgeführt wird.



Signalinterpretation:

Signal	GPIOs (Hardware)	SGPIO (Software)
LOW	weniger als 1,5 V	0
HIGH	mehr als 5 V	1

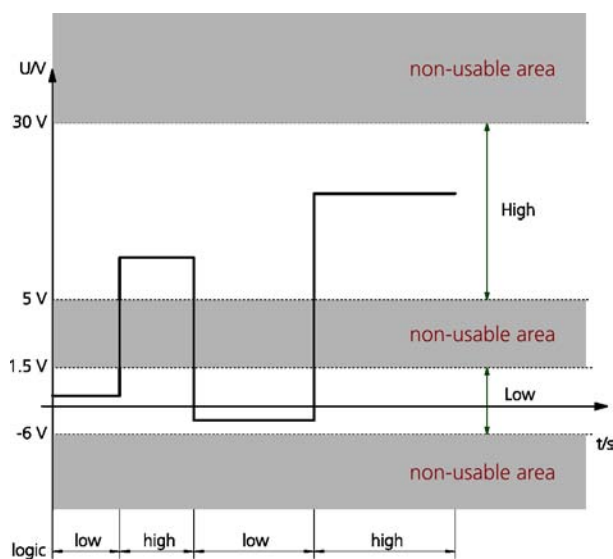
Für die GPIs und SGPIOs lassen sich für 'HIGH zu LOW' und 'LOW zu HIGH' jeweils 25 Funktionen einstellen:

Nr.	Einstellung	Auswirkung
1.	Mute Toggle	toggelt den Master Mute Schalter: wenn geMuted ist: unMute, wenn nicht geMuted ist: Mute
2. - 11.	Change Volume (+0,1 dB, -0,1 dB, +0,5 dB, -0,5 dB, +1,0 dB, -1,0dB, +5,0dB, -5,0 dB, +10,0 dB, -10,0 dB)	Ändert das Master Volume um den eingestellten Wert
12.	Select Input Analog	Auswahl Analog Input
13.	Select Input Dante 1	Auswahl Dante Input 1
14.	Select Input Dante 2	Auswahl Dante Input 2
15.	Select Input AES3 L	Auswahl AES Input L
16.	Select Input AES3 R	Auswahl AES Input R
17.	Select AES Input L+R	Auswahl Input AES L+R
18. - 25.	Load Preset (1 - 8)	Laden des eingestellten Presets

2.1 GPI

Die GPIs sind potentialfreie Optokopplereingänge.

Bei einer Spannung von unter 1,5 V schalten die Optokoppler sicher „aus“. Die Durchbruchspannung, ("Reverse Voltage") liegt bei -6 V. Die maximale negative Spannung von 6 V darf nicht überschritten werden.



Die Stromaufnahme bei 10 V liegt bei 2,5 mA.

Der 10 V-Ausgang ist ein galvanisch getrennter (potentialfreier) DC-DC-Converter mit einer Strombegrenzung von ca.15 mA.

Wenn keine externe Spannungsquelle verwendet wird, kann dieser zur Ansteuerung der GPIOs verwendet werden, oder auch um den Zustand der GPIOs zum Beispiel durch eine 'Low Current LED' anzuzeigen.

Zusätzlich zu den drei GPIOs gibt es noch zwei spezielle "Notfall"-GPIOs:

1. **Mute:** Einstellung, ob bei HIGH oder LOW gemutet wird. Das System wird gemutet und lässt sich weder durch die App noch durch Mute Toggle anderer GPIOs aktivieren.
2. **Priority:** Einstellung, auf welchen Audio-Eingang bei Aktivierung geschaltet werden soll. In diesem Zustand ist das Input-Gain und Input-Mute disabled, d.h. egal ob gemutet ist und wie der Gainregler steht, das Signal geht mit 0 dB auf die Verstärker.

2.2 SGPI

Die SGPIs sind dafür gedacht, die VIDA-Systeme in Mediensteuerungen einzubinden.

SGPIs werden genau wie GPIOs eingestellt. Diese können mit einem Browser oder mit einer anderen Software über HTTP gesetzt bzw. abgefragt werden. Das Webinterface der VIDA antwortet auf Anfragen immer in der 'JavaScript Object Notation' JSON.

Der Aufruf einer Anfrage über HTTP sieht wie folgt aus:

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

Dabei gilt:

- für '<IP>' muss die IP-Adresse der entsprechenden VIDA eingetragen werden,
- für '<ACTIONNUMBER>&<ATTRIBUTE>' müssen die Schlüsselwörter 'action=', 'number=' und/oder 'value=' und einer der angegebenen numerischen Werte eingetragen werden, siehe Tabellenspalte 'Attribut'

Anfragebeispiel:

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

("http://192.168.110.137/cgi-bin/vida_webservice.cgi?" steht immer am Anfang einer Anfrage, daher stehen in der Tabelle die Angaben der Schlüsselwörter mit den entsprechenden Werten)

ActNr.	Beschreibung	Attribut	Beispiel	Rückmeldung
201	Auslesen eines Software GPOs	Number(0-7) 0 ist SGPO 1 1 ist SGPO 2 ... 7 ist SGPO 8	action=201&number=0	{"SUCCESS": "0"} oder {"SUCCESS": "1"} oder Fehlermeldung
202	Auslesen eines Software GPIOs	Number(0-7) 0 ist SGPI1 1 ist SGPI2 ... 7 ist SGPI8	action=202&number=0	{"SUCCESS": "0"} oder {"SUCCESS": "1"} oder Fehlermeldung
203	Setzen eines Software-GPIOs	Number(0-7) value(0/1) 0 ist SGPI 1 1 ist SGPI 2 ... 7 ist SGPI 8 Value 0 ist LOW Value 1 ist HIGH	Action=203&number=0&value=1 oder action=203&number=0&value=0	{"SUCCESS": "0"} oder {"SUCCESS": "1"} oder Fehlermeldung

211	Auslesen eines GPOs	Number(0-1) 0 ist GPO 1 1 ist GPO 2	action=211& number=0	{"SUCCESS": "0"} oder {"SUCCESS": "1"} oder Fehlermeldung
212	Auslesen eines GPIs	Number(0-2) 0 ist GPI 1 1 ist GPI 2 2 ist GPI 3	action=212& number=0	{"SUCCESS": "0"} oder {"SUCCESS": "1"} oder Fehlermeldung
213	Auslesen des PRIO Mute	action=213		{"SUCCESS": "0"} oder {"SUCCESS": "1"} oder Fehlermeldung
214	Auslesen des PRIO Input		action=214	{"SUCCESS": "0"} = Aus {"SUCCESS": "1"} = Analog {"SUCCESS": "2"} = AES3 R {"SUCCESS": "3"} = AES3 L {"SUCCESS": "4"} = AES3 L/R {"SUCCESS": "5"} = DANTE 1 {"SUCCESS": "6"} = DANTE 2 oder Fehlermeldung

Fehlermeldung: {"ERROR": "[Fehlertext]"}
 {"SUCCESS": "0"}
 {"SUCCESS": "1"}
 {"SUCCESS": "2"}
 {"SUCCESS": "3"}
 {"SUCCESS": "4"}
 {"SUCCESS": "5"}
 {"SUCCESS": "6"}
 oder
 Fehlermeldung



Tipp

Damit eine mögliche Kommunikation sichergestellt wird, sollte der VIDA bei der Verwendung dieses Webservices eine feste IP zugewiesen werden.

Beim Setzen eines SGPIs ist zu beachten, dass je nach Einstellung die entsprechende Aktion beim **Wechsel** des Zustandes von HIGH zu LOW bzw. von LOW zu HIGH stattfindet.

Hat man z.B. SGPI 1 bei „LOW zu HIGH“ auf „Change Volume +1,0dB“ eingestellt und will den Pegel um 3dB erhöhen, so müssen folgende Scriptzeilen gesendet werden:

Befehlszeile	Erklärung/Grund
<code>http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=203&number=0&value=0</code>	Um sicherzugehen, dass der Zustand LOW ist
<code>http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=203&number=0&value=1</code>	Wechsel von LOW zu HIGH (+1dB)
<code>http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=203&number=0&value=0</code>	Zurückstellung auf LOW
<code>http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=203&number=0&value=1</code>	Wechsel von LOW zu HIGH (+1dB)
<code>http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=203&number=0&value=0</code>	Zurückstellung auf LOW
<code>http://192.168.110.137/cgi-bin/vida_webservice.cgi?action=203&number=0&value=1</code>	Wechsel von LOW zu HIGH (+1dB)

Die VIDA ist jetzt 3 dB lauter!



Tipp

Zur Sicherheit sollte der SGPI wieder auf LOW gesetzt werden, damit mit einem LOW-HIGH-Wechsel eine weitere Aktion ausgelöst werden kann:

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

Vor dem Senden eines Befehls muss sichergestellt sein, dass die vorherige Aktion vollständig abgeschlossen ist, da das System ansonsten eine Fehlermeldung ausgibt.

Für das Auslesen der GPOs muss eine Programmschleife programmiert werden, die in regelmäßigen Abständen die VIDA abfragt. Auch hier gilt es den Abschluss der vorherigen Aktion abzuwarten.

Weitere Informationen zur Scriptsprache 'JSON' finden Sie im Internet unter

- https://de.wikipedia.org/wiki/JavaScript_Object_Notation

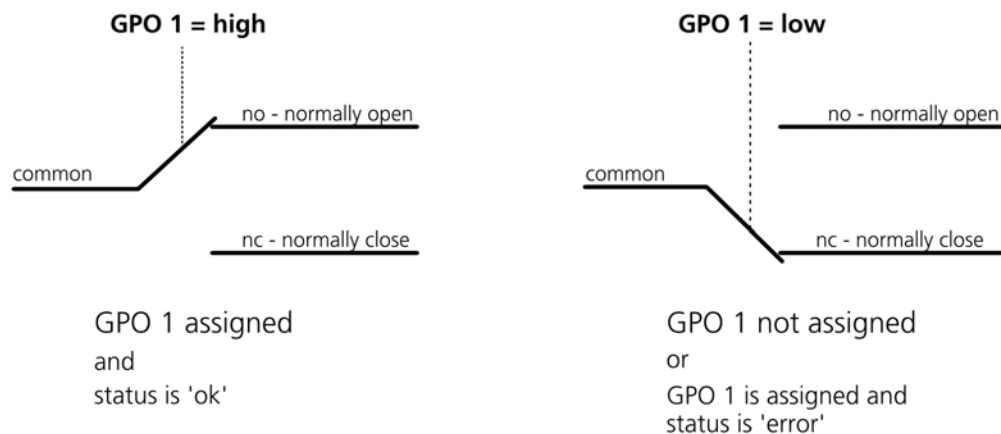
3. GPO/SGPO

3.1 GPO

Die GPOs sind potentialfreie MOSFET-Schalter (NO Normaly Open und NC Normaly Closed).

Wenn die Ausgänge in der Software aktiviert sind, ist NO niederohmig und NC hochohmig zu common.

Für einen Fehlerfall gilt, dass NO hochohmig und NC niederohmig ist.



Maximaler Schaltstrom ResistiveLoad:

- 2 A bis 30 V AC/DC
- 1 A bis 60 V AC/DC
- 600 mA bis 110 V AC

3.2 SGPO

Für die SGPOs lassen sich fünf Zustände abfragen:

1. **Off:** SGPO ist deaktiviert, der Zustand ist immer '0'
2. **System On:** SGPO ist '1', wenn die VIDA mit Strom versorgt und hochgefahren ist.
3. **System Status (High if config is wrong):** SGPO ist '0', wenn ein Fehler in der Konfiguration ist. Dieser Zustand wird durch die rote Front-LED signalisiert.
4. **System Overtemp:** SGPO ist '1', wenn die Temperatur eines Verstärkers über 70° C steigt, oder wenn die Temperatur des Netzteils 5° C vor der Abschalttemperatur liegt.
5. **System Hardware Fault:** SGPO ist '1', wenn
 - 'Protect' gemeldet wird oder der Lautsprecher nicht angeschlossen ist.
 - Amp DAC Error
 - Amp Error

4. Anhang, JSON-Codebeispiel

```

// 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);
}
//-----

```