Application Note

Analog control of IX amplifiers and MXE Matrix Mix Engines via the IX/MXE control port (GPIOs)

IX amplifiers and **MXE Matrix Mix Engines** are equipped with a **control port** for interfacing to other systems, using analog control wiring and controls.

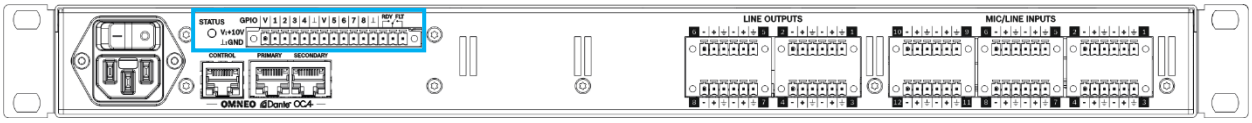


Image 1: MXE rear view

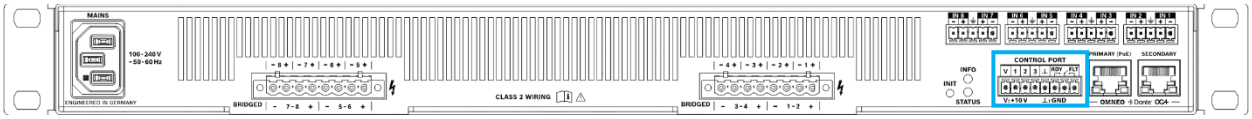
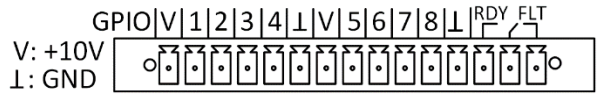


Image 2: IX rear view (8 channel model shown)

The **control port** (*GPIO* or *CONTROL PORT*) can be found on the IX's and MXE's rear panel. It offers in total **eight (MXE)** or **three (IX)** freely configurable **GPIOs** (General Purpose Inputs and Outputs), a **Ready/Fault** relay (*RDY/FLT*) and +10 V (*V*) and ground (⊥) reference pins.

The **GPIOs** can be configured via **SONICUE** as either **Analog In**, **Digital In**, or **Digital Out**.



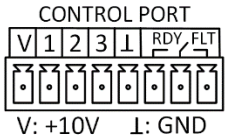


Image 3 and 4: MXE (left) and IX (right) control port detail view

Requirements for using MXE Task Engine

MXE Matrix Mix Engine with firmware version 1.4.3119 (or higher)

SONICUE Sound System Software 1.3.0 (or higher) installed on computer

Requirements for using IX Task Engine

IX amplifier with firmware version 1.0.0 (or higher)

SONICUE Sound System Software 1.5.0 (or higher) installed on computer

TaskEngine Quick Start Guide

Each IX and MXE device contains a powerful **logic processing engine** which is independent to the DSP used for audio. The **TaskEngine** provides all the tools, and an extensive selection of logic control blocks, necessary to build comprehensive integrated control systems. It provides the **link between** different **SONICUE hardware**, **PanelDesigner** user interfaces, and many **third-party** devices.

IX and **MXE TaskEngine** can not only **control parameters** in the device itself, but also **in other devices** on the same network.

- MXE can control other MXEs, IPX, IX, OCA devices and 3rd party products with http API.
- IX can control other IXs and 3rd party products with http API.

Configuring input and output nodes

MXE Matrix, ***IPX Amplifier*** and ***IX Amplifier*** blocks in (MXE) **TaskEngine** have by **default** an input node for switching ***Power*** and output nodes for ***Power***, ***Error***, and ***Offline*** status.

IPX amplifiers have by default **additionally** input and output nodes for the ***Alarm*** preset.

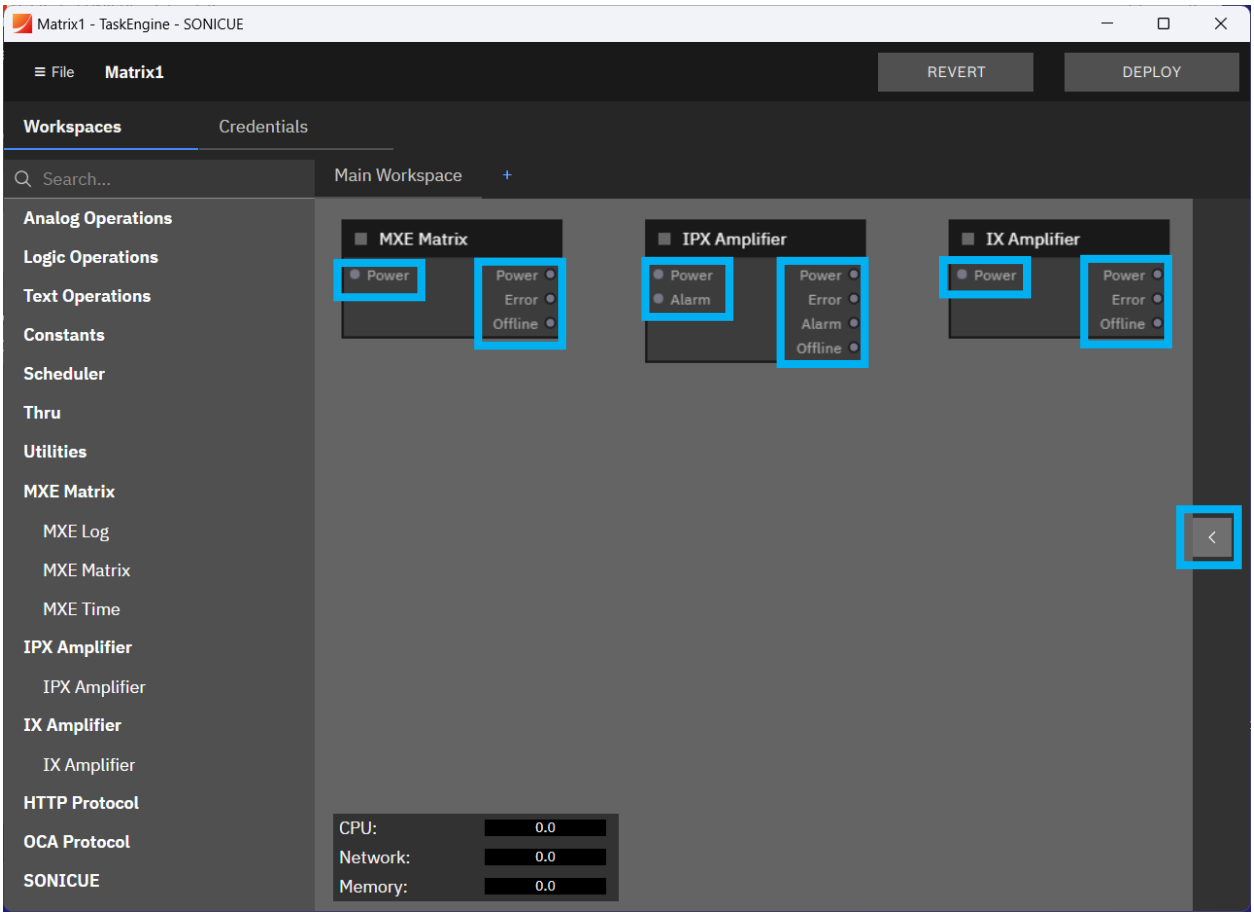


Image 5: MXE TaskEngine with *MXE Matrix*, *IPX Amplifier* and *IX Amplifier* blocks

By **selecting** a matrix or amplifier **block** and **clicking** the  **button** on the right-hand side of the TaskEngine workspace, the Property Pane of a logic block can be opened.

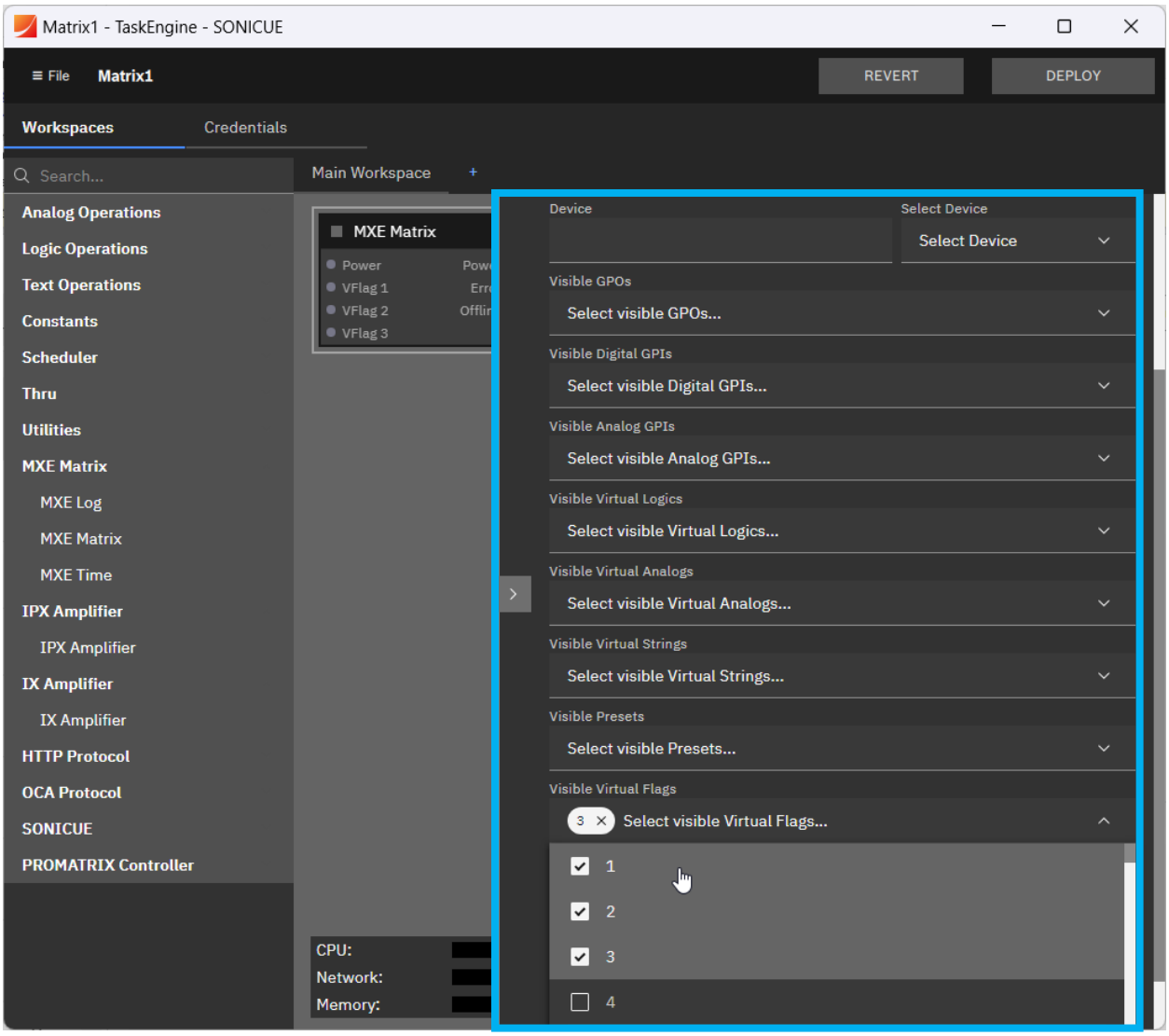


Image 6: MXE TaskEngine with Property Pane opened for MXE Matrix block, *Visible Virtual Flags 1-3* selected.

In case of the ***MXE Matrix*** block, it's possible to add nodes for:

- Up to **8** ***Visible GPOs***
- Up to **8** ***Visible Digital GPIs***
- Up to **8** ***Visible Analog GPIs***
- Up to **200** ***Visible Virtual Logics***
- Up to **200** ***Visible Virtual Analogs***
- Up to **200** ***Visible Virtual Strings***
- Up to **60 User + 1 Factory** ***Visible Presets***
- Up to **200** ***Visible Virtual Flags***

*MXE has 8 GPIOs which can be used as either GPO or Digital GPI or Analog GPI.

In case of the **IX Amplifier** block, it's possible to add nodes for:

- Up to **3** ***Visible GPOs***
- Up to **3** ***Visible Digital GPIs***
- Up to **3** ***Visible Analog GPIs***
- Up to **200** ***Visible Virtual Logics***
- Up to **200** ***Visible Virtual Analogs***
- Up to **200** ***Visible Virtual Strings***
- Up to **20 User + 1 Factory** ***Visible Presets***
- Up to **10** ***Visible Virtual Flags***

*IX has 3 GPIOs which can be used as either GPO or Digital GPI or Analog GPI.

In case of the **IPX Amplifier** block, it's possible to add nodes for:

- Up to **3** ***Visible GPOs***
- Up to **3** ***Visible Digital GPIs***
- Up to **3** ***Visible Analog GPIs***
- Up to **20 User + 1 Factory** ***Visible Presets***

*IPX has 3 GPIOs which can be used as either GPO or Digital GPI or Analog GPI.

Selecting the correct hardware device

By **selecting** a matrix or amplifier **block** and **clicking** the  **button** on the right-hand side of the TaskEngine workspace, the Property Pane of a block can be opened.

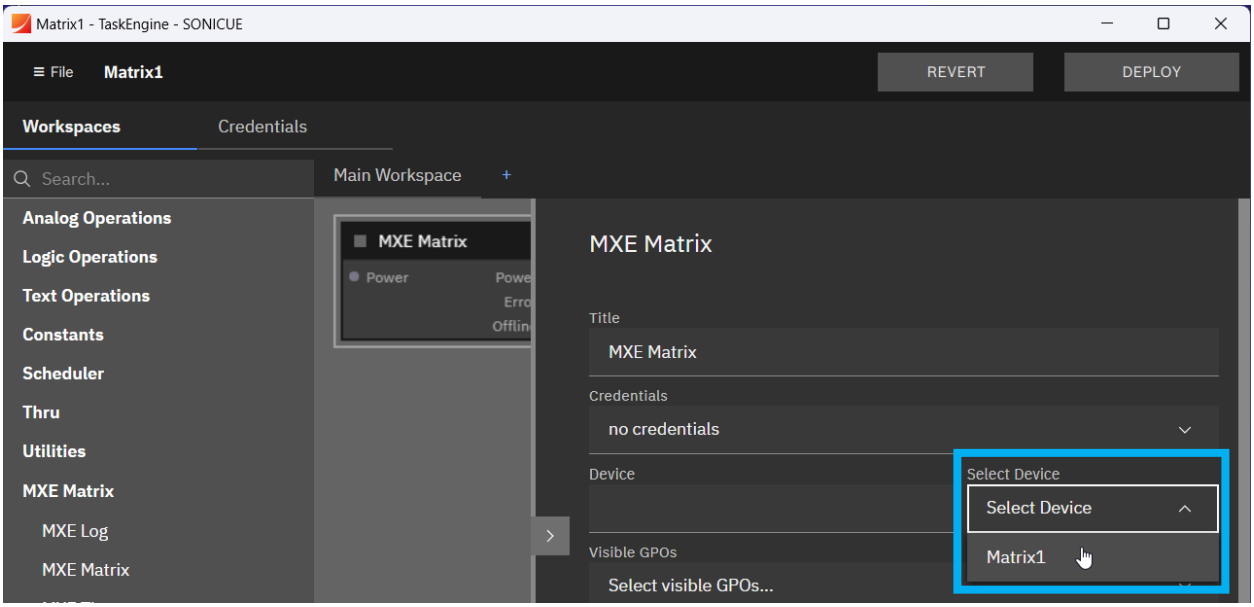


Image 7: *Device* selection of *Matrix1* for MXE Matrix block via Property Pane > *Select Device*

As there can be **multiple** MXE **matrices** or IX/IPX **amplifiers** on the network, it's crucial to **select the correct** hardware ***Device*** with ***Select Device***. Therefore, the hardware device needs to be present on the network. Alternatively, it's possible to enter the correct name in the *Device* field manually if the device is not connected to the network (yet).

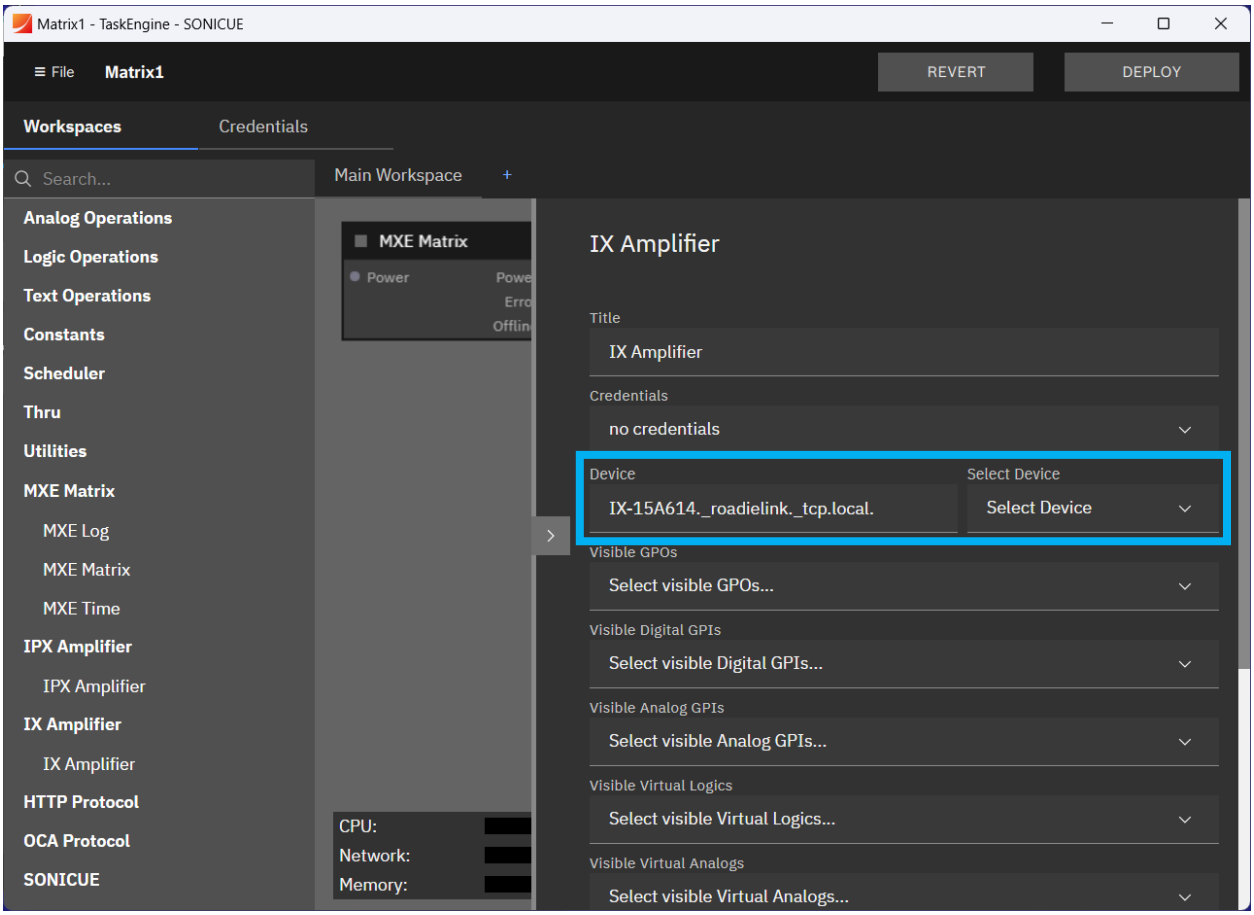


Image 8: *Amp1* selected via *Select Device*, unique service name *IX-15A614* is part of the *Device* name

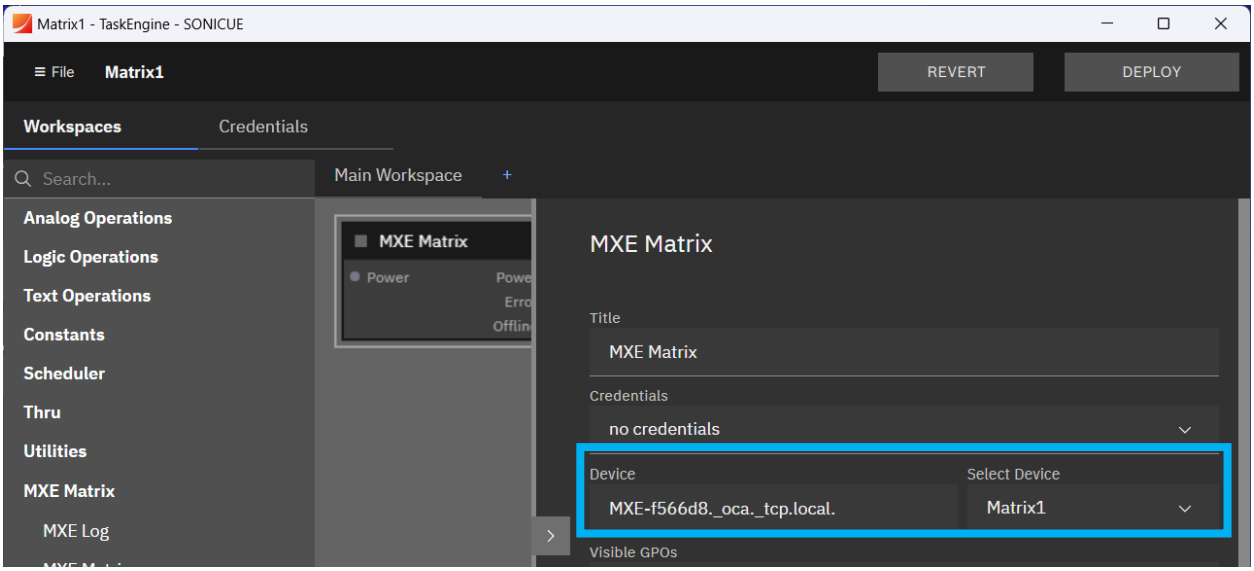


Image 9: *Matrix1* selected via *Select Device*, unique service name *MXE-f566d8* is part of the *Device* name

Adding SONICUE Expressions

In **SONICUE** versions **1.3.0…1.4.0** the buttons for adding ***Logic*** and ***Analog*** **expressions** are located at the **upper right corner** of the TaskEngine window.

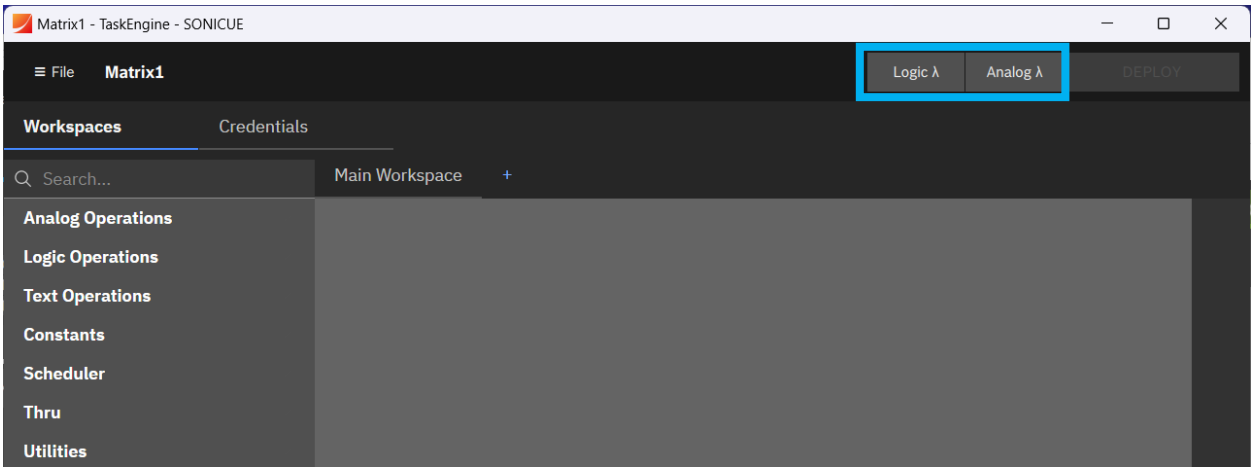


Image 10: *Logic λ* and *Analog λ* buttons for adding SONICUE expressions to the TaskEngine.

In **SONICUE** versions **1.5.0 and higher**, logic and analog ***SONICUE Expression***s can be added from the **TaskEngine catalog**, ***SONICUE*** menu.

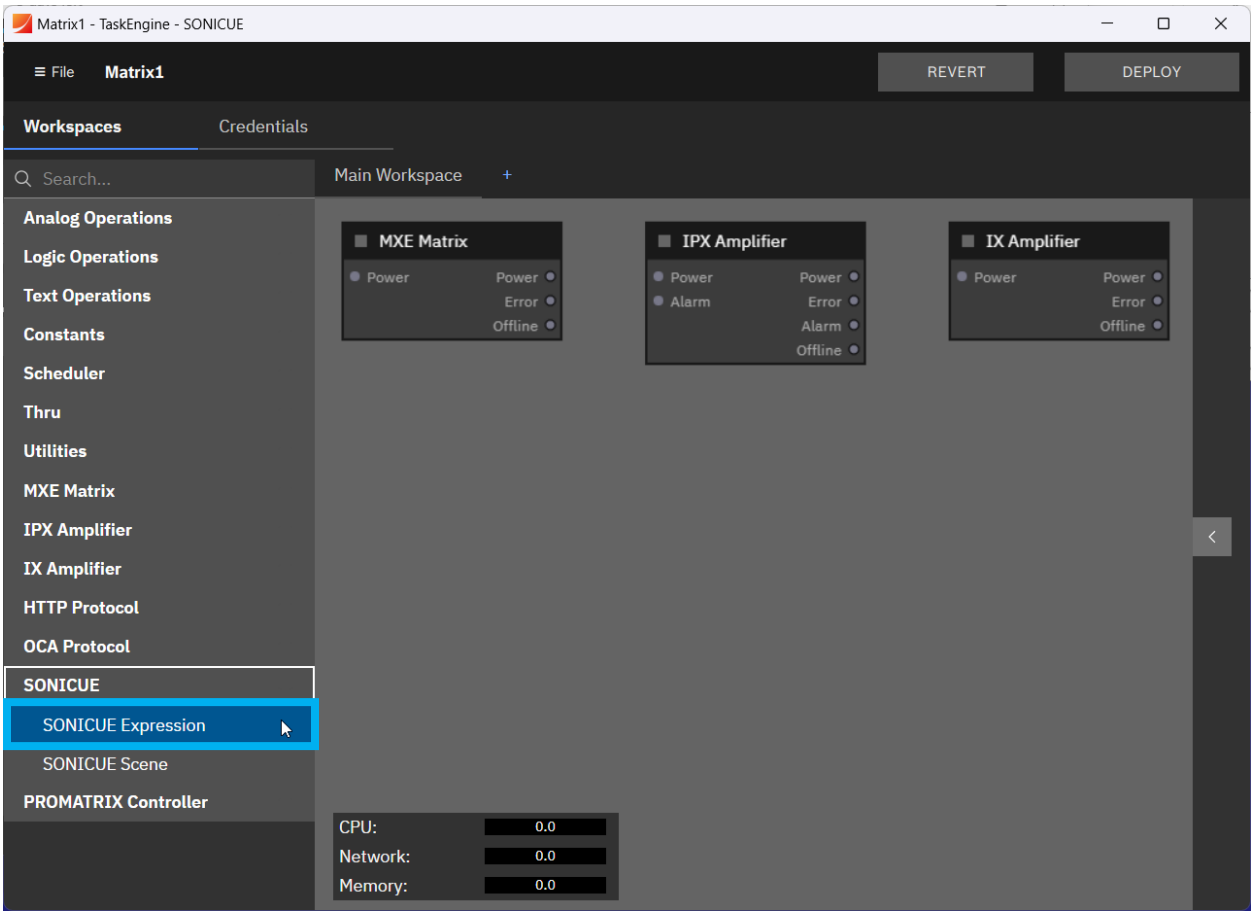


Image 11: *SONICUE* menu with *SONICUE Expression*.

As an alternative, for **DSP parameters** it's possible to add those via **Ctrl + drag&drop** directly from a **DSP flyout**, even with multi-select (= group).

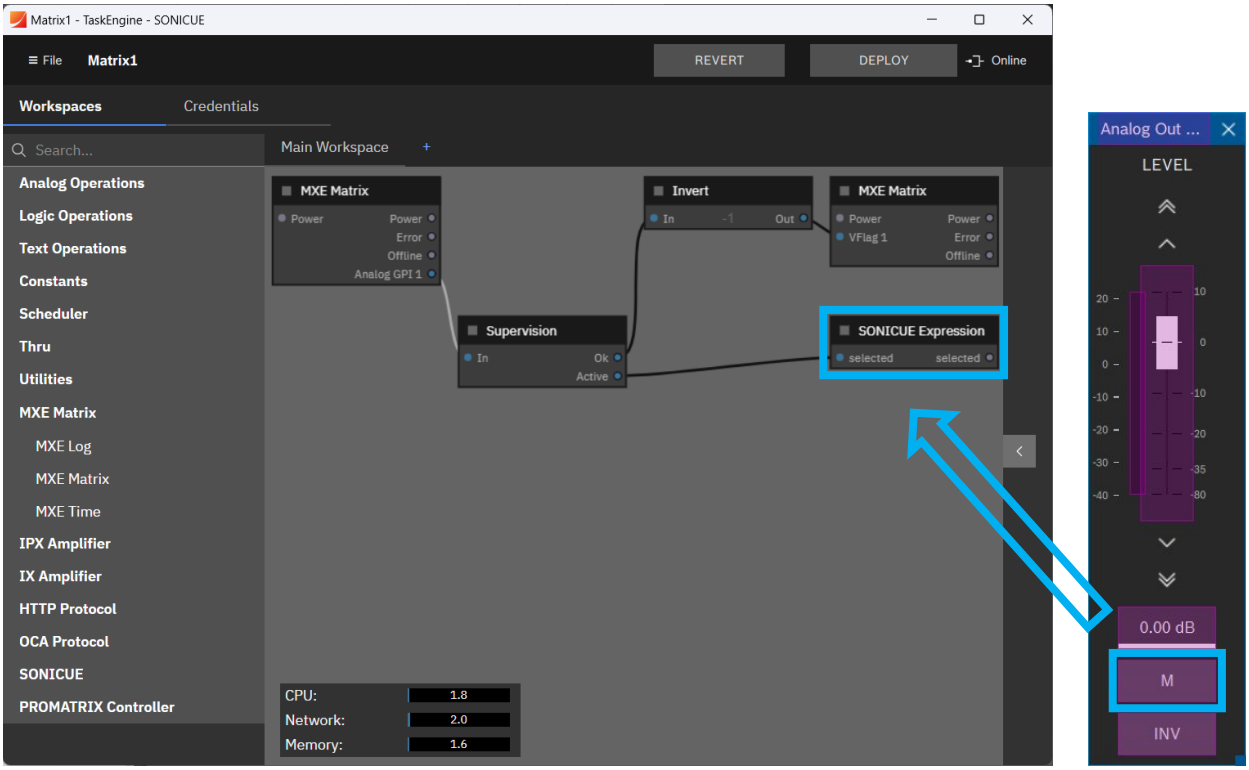


Image 12 and 13: Adding a *SONICUE Expression* to the TaskEngine via drag & drop from the *Output* flyout.

Deploying the logic configuration

The **most important step** after building or modifying **TaskEngine** logic is always to ***DEPLOY*** **the logic** to the hardware when ***ONLINE*** with the device. While going online writing a logic is automatically deployed.

Image 14: *DEPLOY* button "active" when the TaskEngine configuration has not yet been deployed.

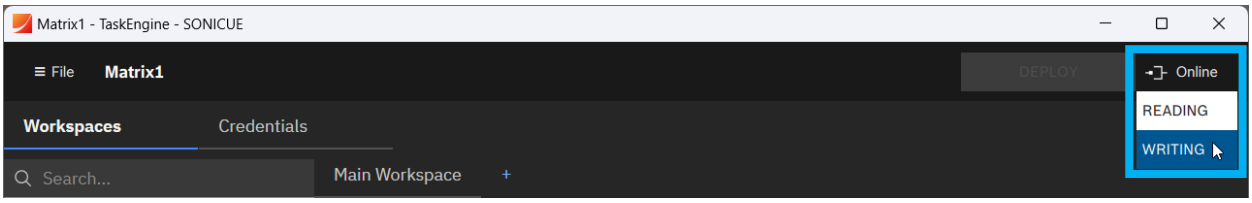


Image 15: Going online *WRITING* will automatically deploy the TaskEngine configuration.

Image 16: *DEPLOY* button "inactive" when the TaskEngine configuration has already been deployed.

1. Example Analog Level Control

This example shows how an **analog potentiometer**, connected to **IX** or **MXE Control Port (GPIO)**, can be used to control a **DSP level**.

Wiring diagram

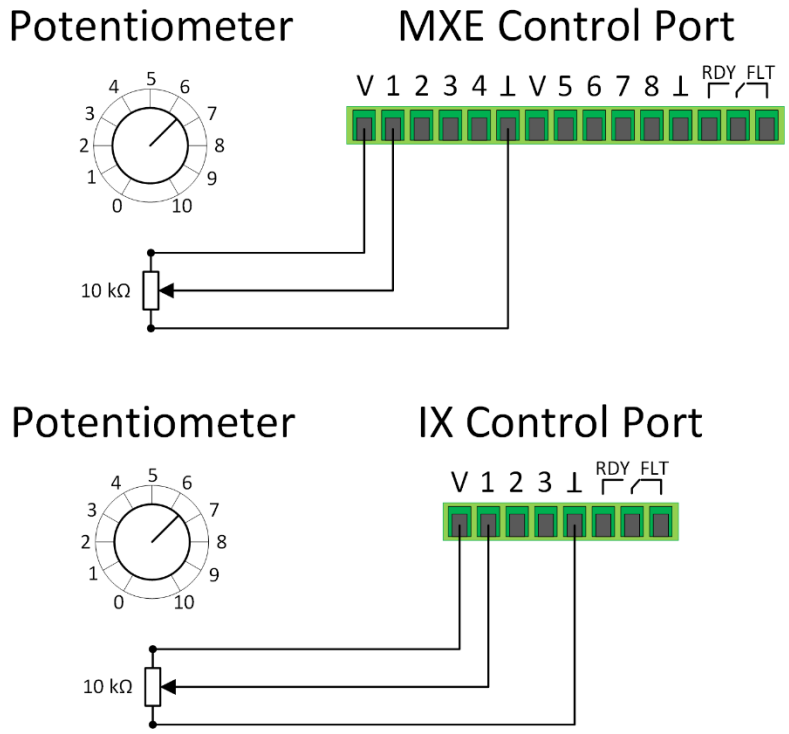


Image 17: Connection of a 10 kΩ analog potentiometer to *MXE* or *IX Control Port*

GPIO configuration

Make sure that in SONICUE under **Setup>GPIO** the GPIO that shall be used is set to the correct type. In our example we need ***GPIO 1*** configured as ***ANALOG IN***.

Image 18 and 19: Setting *Matrix1* (left) or *Amp1* (right) *GPIO 1* to *ANALOG IN*.

TaskEngine programming

In the following TaskEngine structure, an MXE's ***Analog GPI 1*** is used to control a DSP **Level** (*SONICUE Expression*). The ***Analog Scaler*** block **translates** the control **voltage range** (0…10 V) into the DSP **level range** (e.g. -50…0 dB) to be controlled via the analog potentiometer.

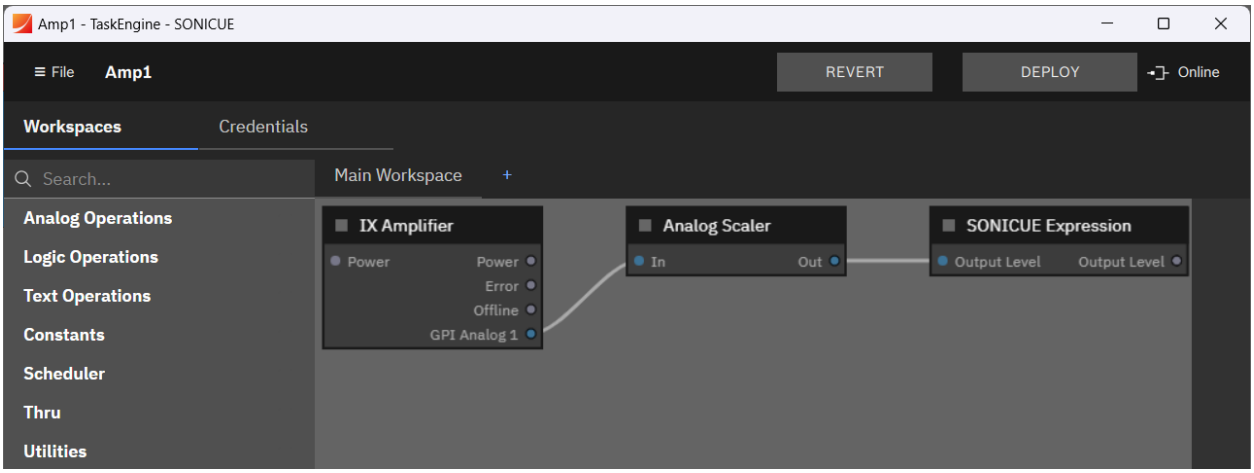


Image 20: IX TaskEngine configuration with *Analog Scaler* block.

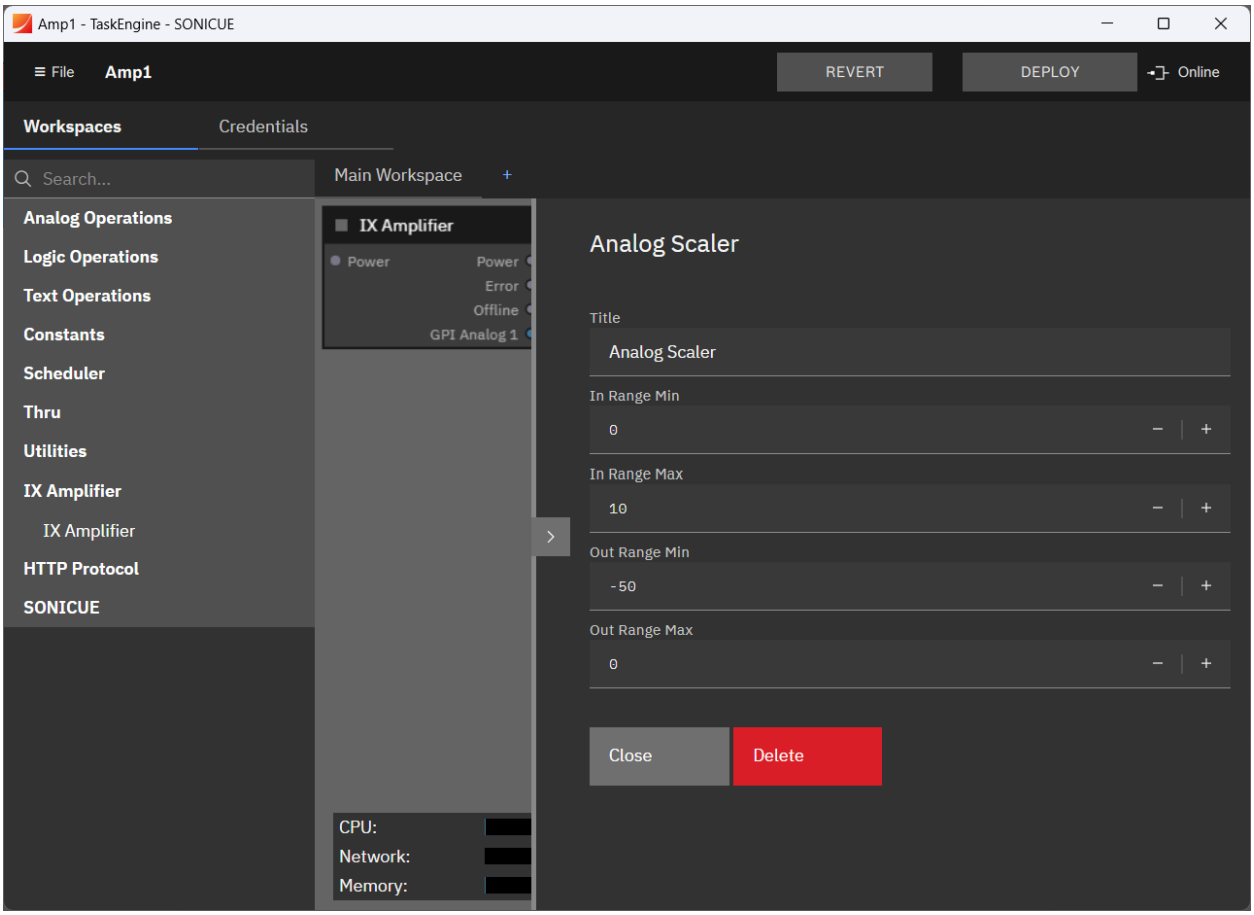


Image 21: *Analog Scaler* block Property Pane open, with typical values for a zone level control (-50…0 dB).

2. Example Fire-Alarm Mute

This example shows how a **potential-free relay contact**, connected to **IX** or **MXE Control Port (GPIOs)**, can be used to activate a **DSP mute**. The connection is supervised by two resistors.

Wiring diagram

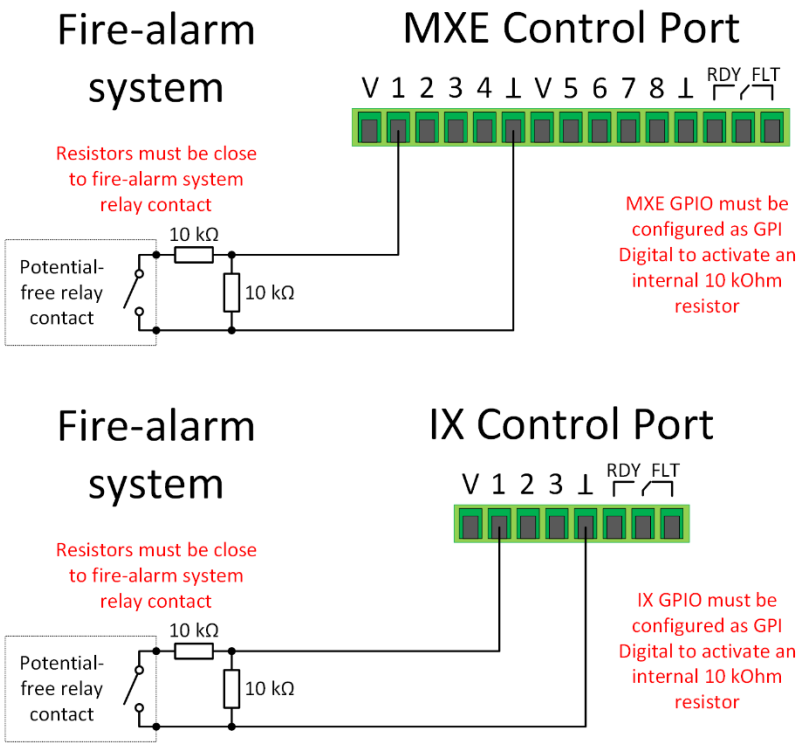


Image 22: Connection of a potential-free relay contact of a fire-alarm system to *MXE* or *IX Control Port*

GPIO configuration

Make sure that in SONICUE under **Setup>GPIO** the GPIO that shall be used is set to the correct type. In our example we need ***GPIO 1*** configured as ***DIGITAL IN***.

Image 23 and 24: Setting *Matrix1* (left) or *Amp1* (right) *GPIO 1* to *DIGITAL IN*.

TaskEngine programming

In the following TaskEngine structure, an MXE's ***Analog GPI 1*** is used to activate a DSP **mute** (*SONICUE Expression*) in multiple output channels. The **voltage range** of the Analog GPI 1 is monitored with a ***Supervision*** block. An out-of-range **fault** of the **supervised voltage** is reported to an MXE virtual flag ***VFlag 1***. Please observe the ***Invert*** block, to trigger the VFlag 1 (User Flag 1) when something is **not OK**.

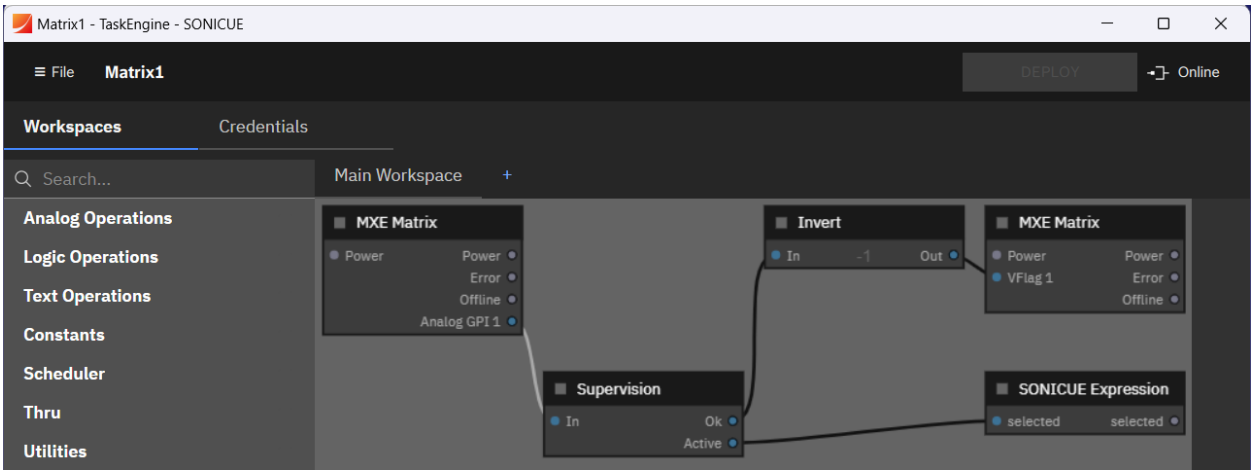


Image 25: MXE TaskEngine logic with *Supervision* block for supervising an analog control connection.

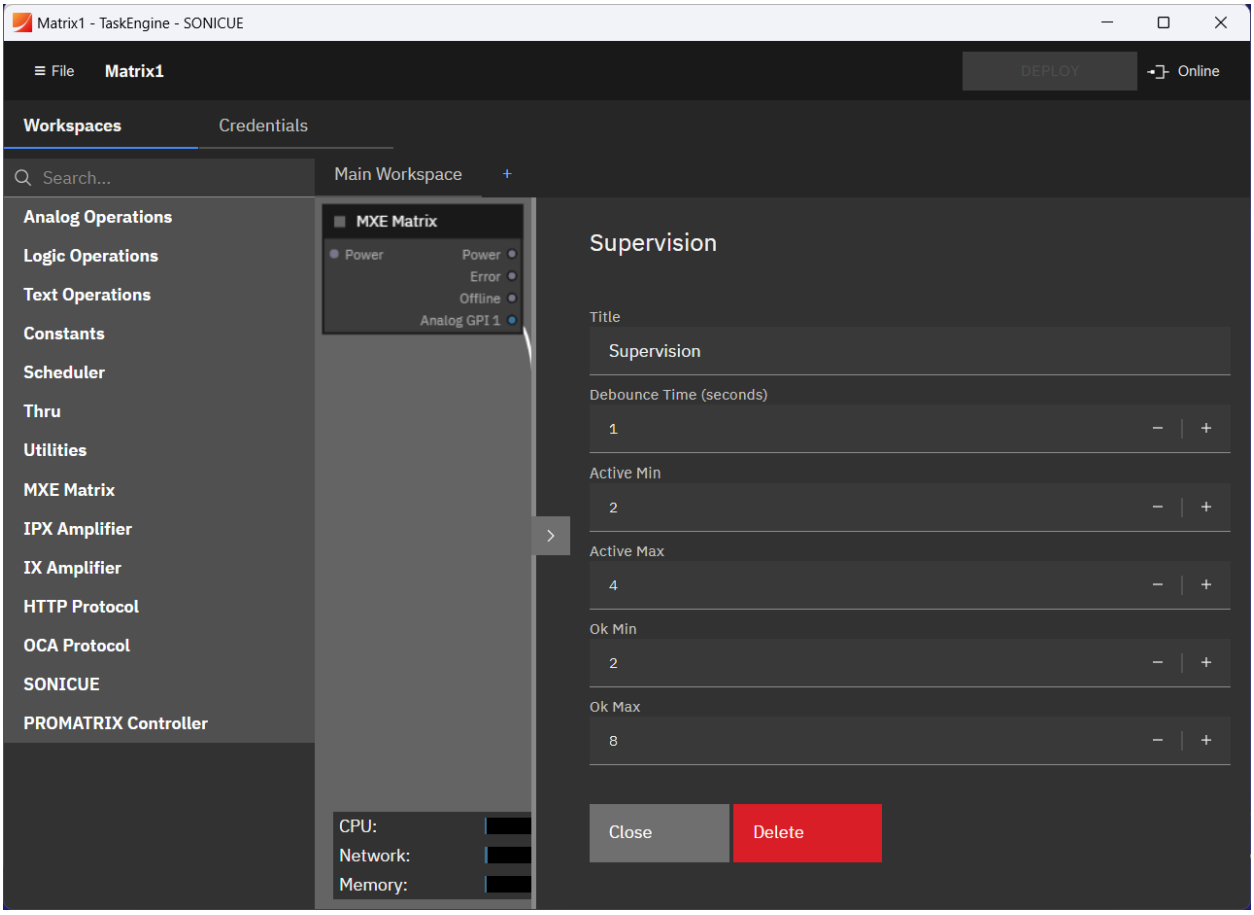


Image 26: Property Pane of the *Supervision* block open showing configuration details (all default values).

3. Example Power/Standby Switch

This example shows how a **toggle or momentary switch**, connected to **IX** or **MXE Control Port (GPIOs)**, can be used to switch the device **Power/Standby.**

Wiring diagram

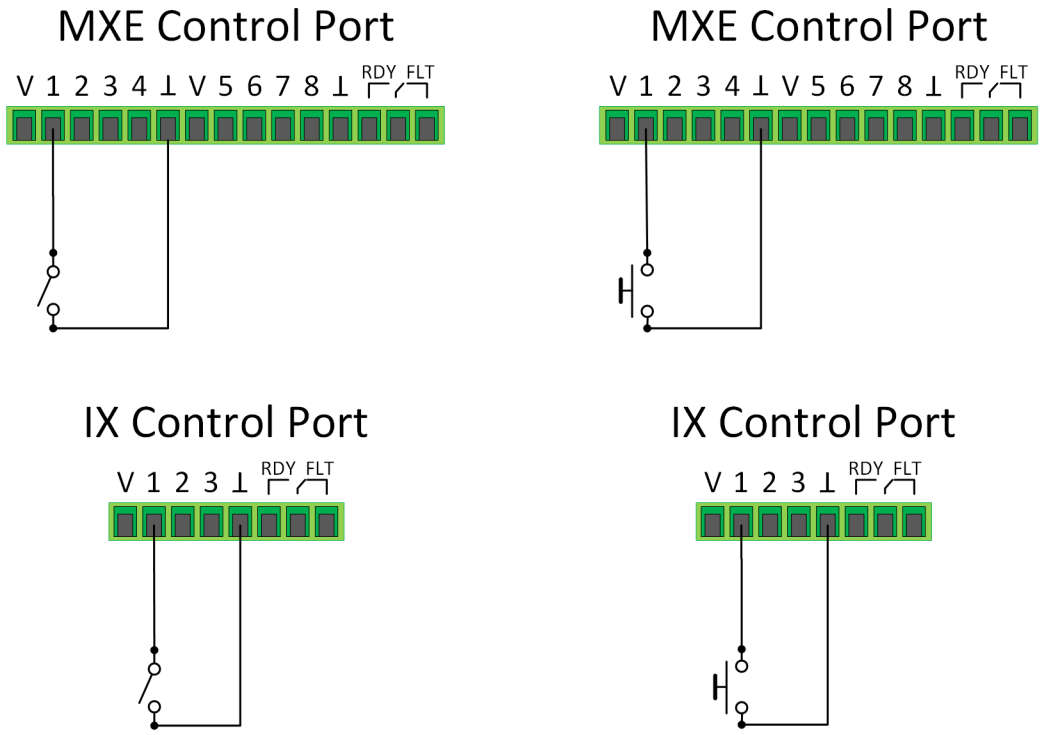


Image 27 and 28: Connection of a toggle switch (left) or momentary switch (right) to *MXE* or *IX Control Port*

GPIO configuration

Make sure that in SONICUE under **Setup>GPIO** the GPIO that shall be used is set to the correct type. In our example we need ***GPIO 1*** configured as ***DIGITAL IN***.

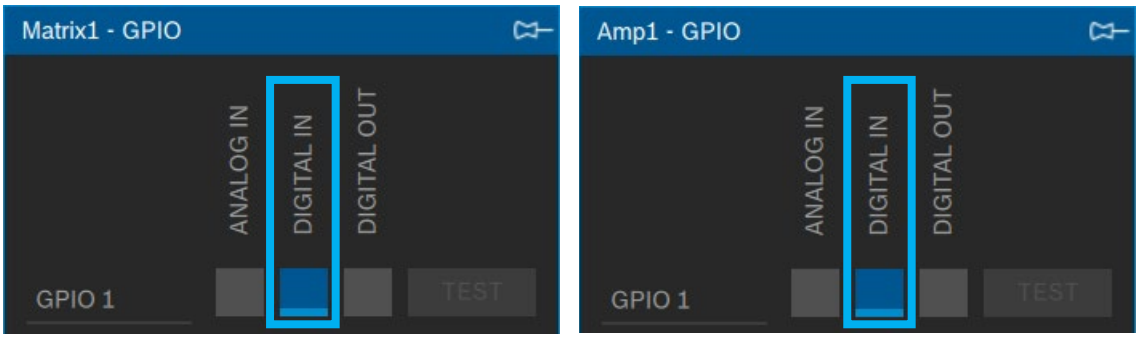


Image 29 and 30: Setting *Matrix1* (left) or *Amp1* (right) *GPIO 1* to *DIGITAL IN*.

TaskEngine programming

In the following TaskEngine structure, an MXE's ***Digital GPI 1*** is used to switch MXE ***Power*** with a toggle switch (latching) connected to GPIO 1.

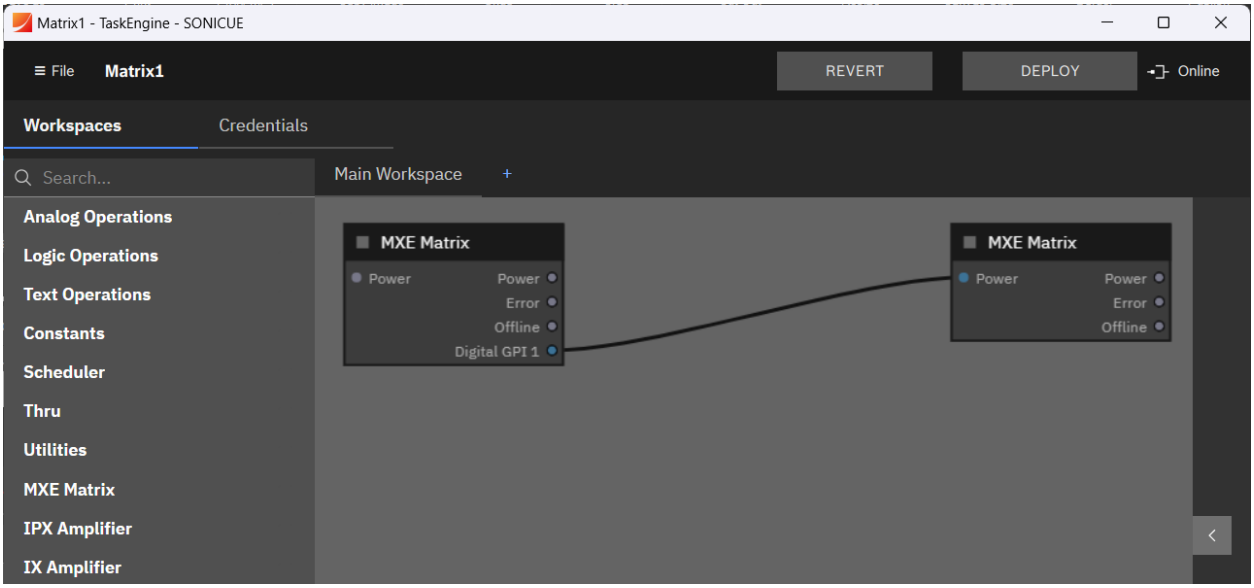


Image 31: MXE TaskEngine configuration for switching MXE *Power* via MXE *Digital GPI 1* (with toggle switch).

In this modified TaskEngine structure, an IX's ***Digital GPI 1*** is used to switch IX ***Power*** with a momentary (non-latching) switch connected to GPIO 1.

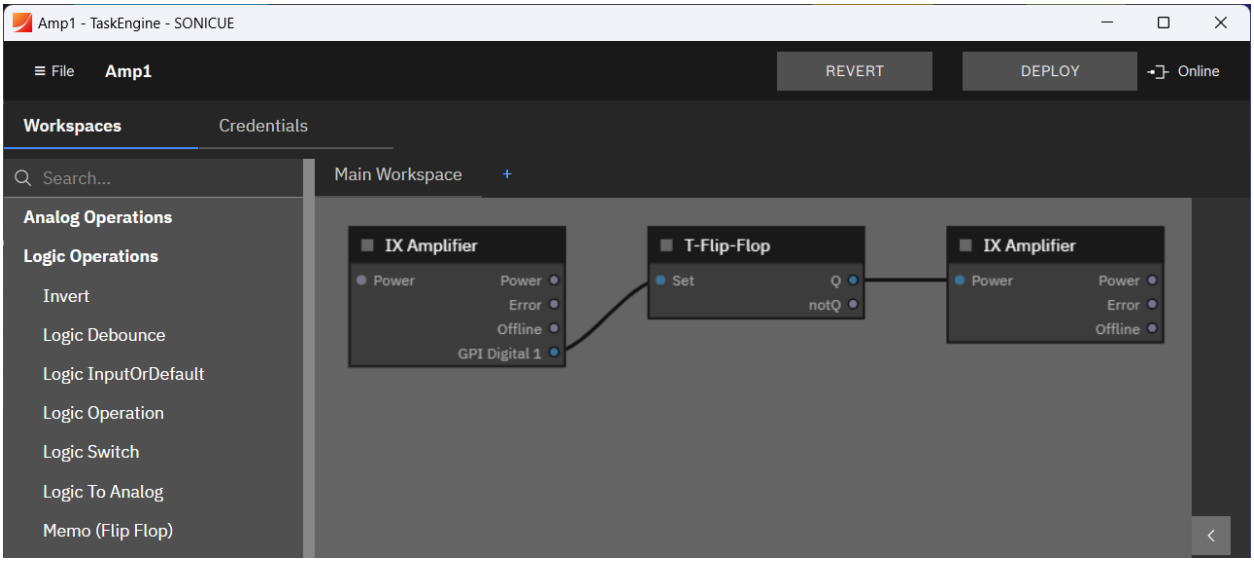


Image 32: IX TaskEngine config with *T-Flip-Flop* for switching IX *Power* via IX *Digital GPI 1* (with momentary switch).

4. Example Preset Recall or Scene Recall

This example shows how (a) **momentary switch(es)**, connected to **IX** or **MXE Control Port (GPIOs)**, can be used for **Preset Recall** or **Scene Recall.**

Wiring diagram

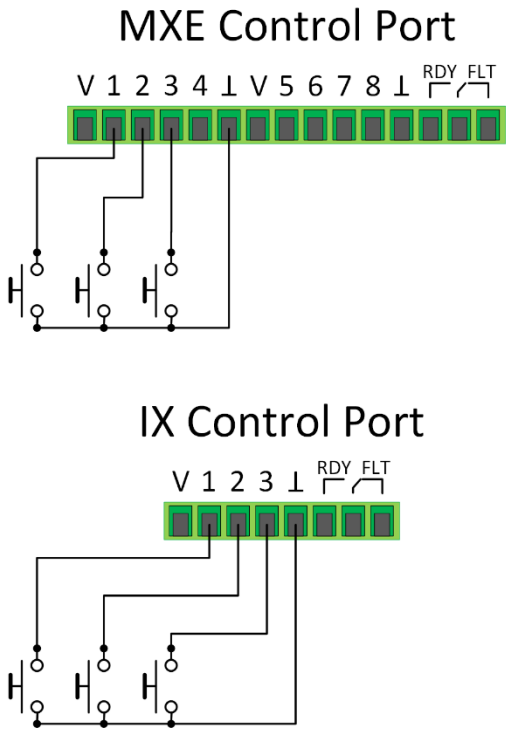


Image 33: Connection of pushbutton switches to *MXE* or *IX Control Port*

GPIO configuration

Make sure that in SONICUE under **Setup>GPIO** the GPIO(s) that shall be used are/is set to the correct type. In our example we need ***GPIO 1 + 2 + 3*** configured as ***DIGITAL IN***.

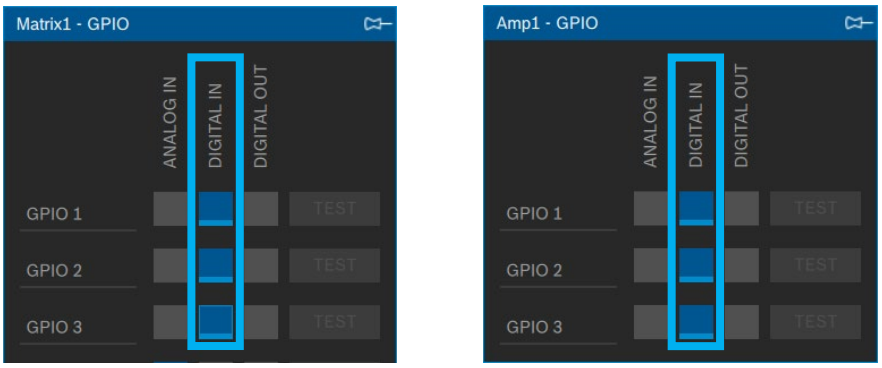


Image 34 and 35: Setting *Matrix1* (left) or *Amp1* (right) *GPIO 1, 2 and 3* to *DIGITAL IN*.

Presets to be recalled

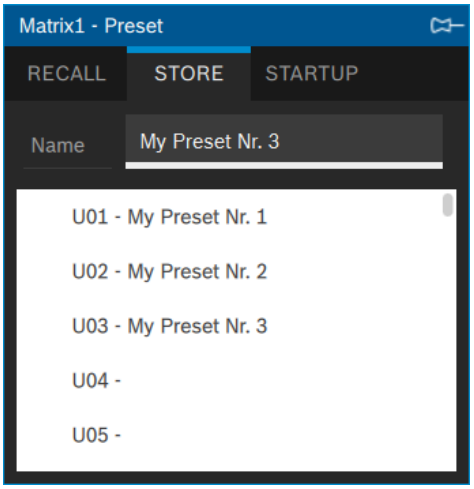


Image 36: Matrix1 – *Preset* flyout with presets *U01 – U03* stored.

TaskEngine programming for Preset Recall

In the following TaskEngine structure, an IX's ***GPI Digital 1***, ***2***, and ***3*** are used to recall User ***Preset U01***, ***U02***, and ***U03***.

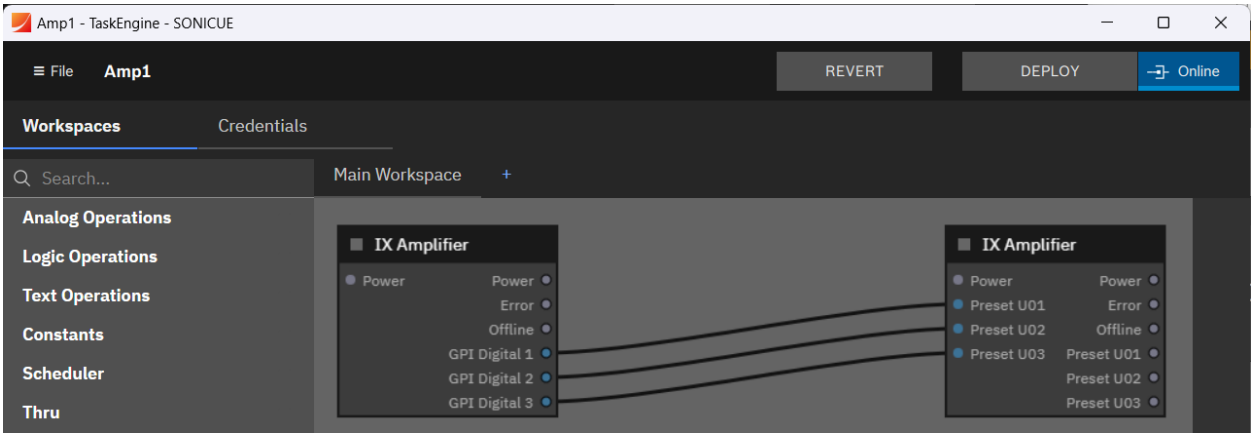


Image 37: IX TaskEngine configuration for recalling *Preset U01*, *U02* and *U03* via GPIOs (*GPI Digital 1*, *2* and *3*).

As an alternative, the ***VLogic 1***, ***2***, and ***3*** can be used for **3rd party interfacing via http API**.

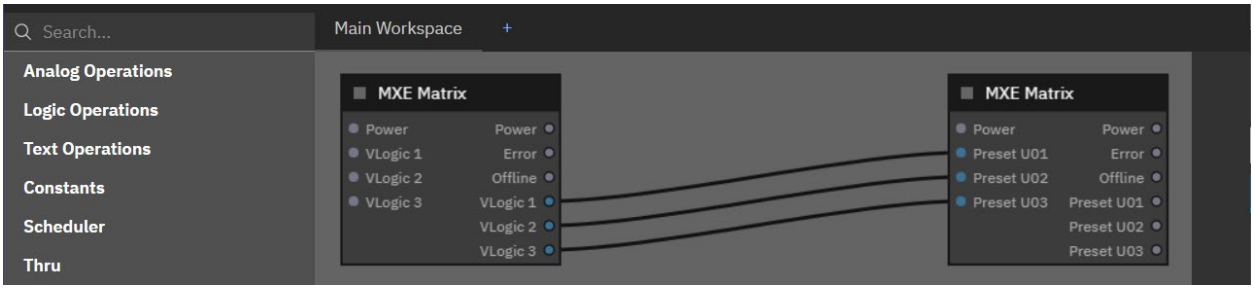


Image 38: MXE TaskEngine config for recalling *Preset U01*, *U02* and *U03* via Virtual Logics (*VLogic 1*, *2* and *3*).

TaskEngine programming for Scene Recall

In the following TaskEngine structure, an IX's ***GPI Digital 1***, ***2***, and ***3*** are used to **recall slots *Load 1***, ***2*** and ***3*** in a ***SONICUE Scene*** block.

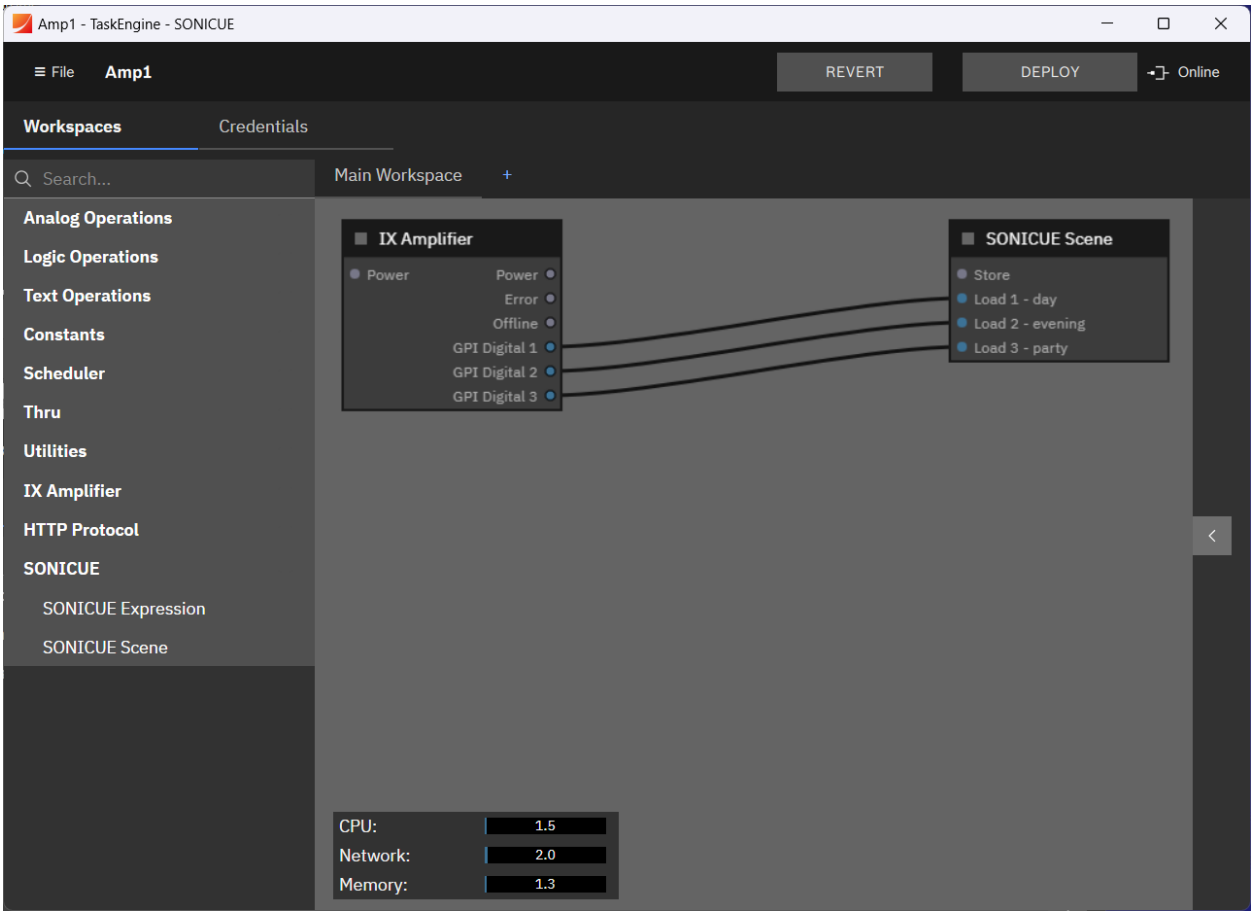


Image 39: IX TaskEngine config for recalling *Preset U01*, *U02* and *U03* via GPIOs (*GPI Digital 1*, *2* and *3*).

Also here, as an alternative the ***VLogic 1***, ***2***, and ***3*** can be used for **3rd party interfacing via http API** or buttons in a **SONICUE Control** user interface created in SONICUE PanelDesigner.

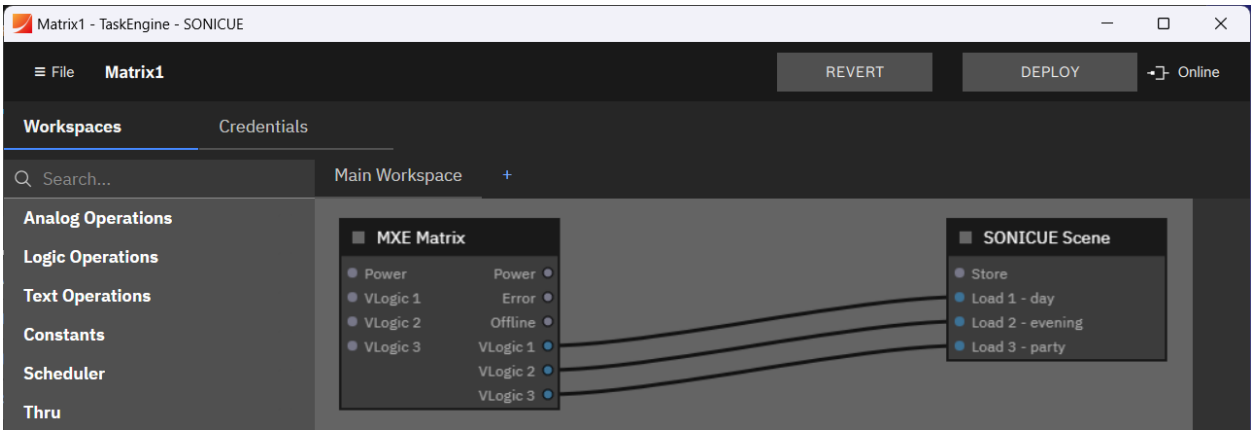


Image 40: MXE TaskEngine config for recalling *Preset U01*, *U02* and *U03* via Virtual Logics (*VLogic 1*, *2* and *3*).

5. Example External Relay

This example shows how an **external relay**, connected to **IX** or **MXE Control Port (GPIOs)**, can be used to **switch higher currents and voltages** than it is possible with GPIOs directly.

Wiring diagram

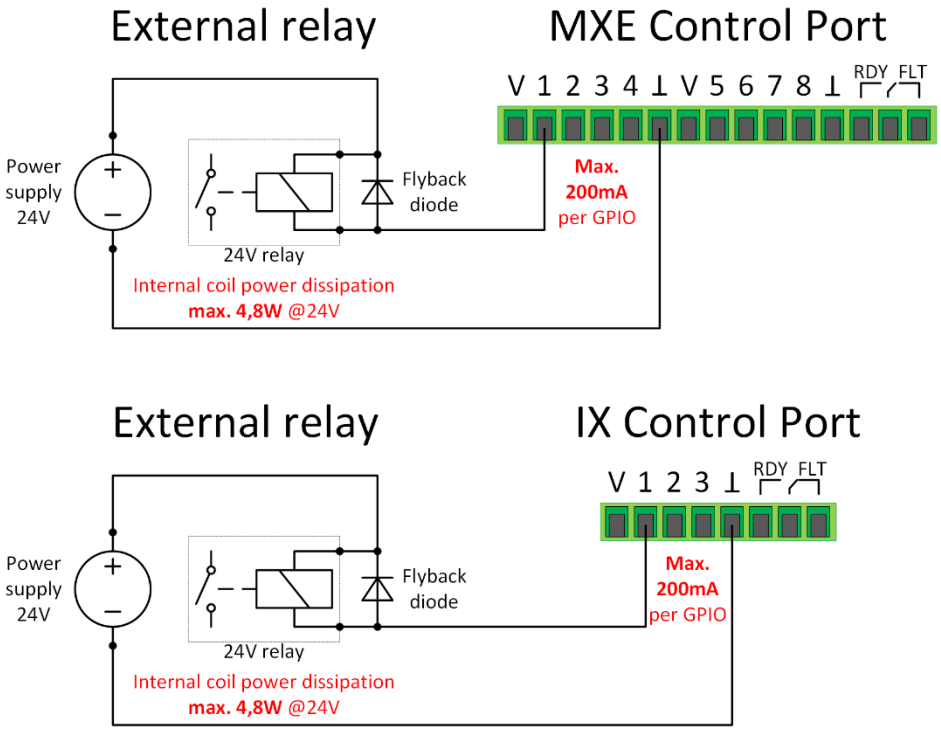


Image 41: Connection of an external relay to *MXE* or *IX Control Port*

GPIO configuration

Make sure that in SONICUE under **Setup>GPIO** the GPIO that shall be used is set to the correct type. In our example we need ***GPIO 1*** configured as ***DIGITAL OUT.***

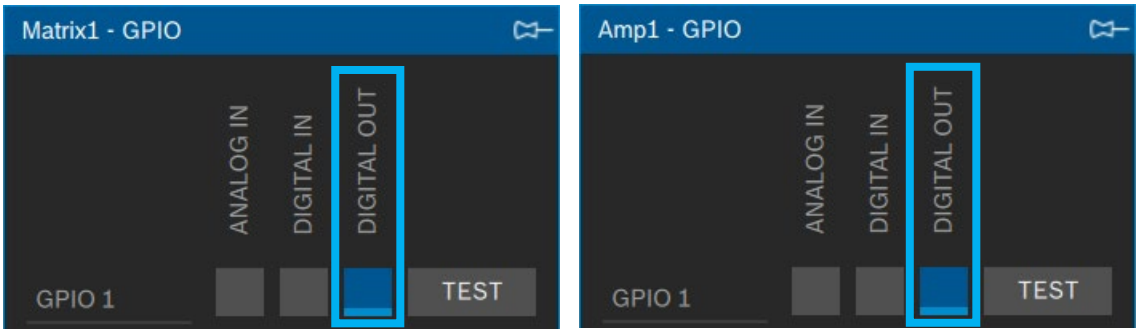


Image 42 and 43: Setting *Matrix1* (left) or *Amp1* (right) *GPIO 1* to *DIGITAL OUT*.

TaskEngine programming

In the following TaskEngine structure, an IX's or MXE's ***VLogic 1*** is used to activate ***GPO 1*** to control an external relay.

The IX's and MXE's **virtual analog** and **logic** values are a perfect way to link a **SONICUE Control** user interface to IX or MXE **TaskEngine**.

Virtual analog and **logic** values can also be accessed thru **IX**'s and **MXE**'s **http API**.

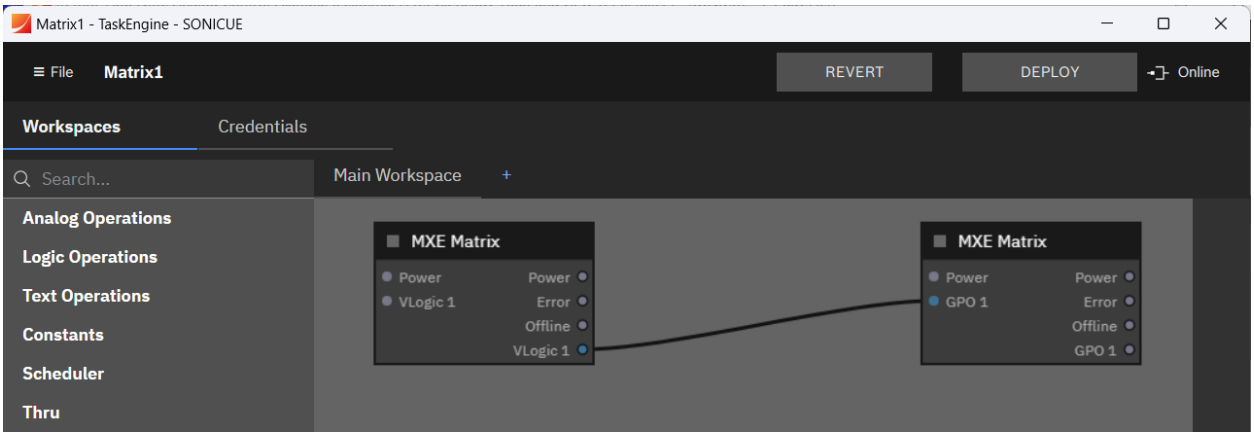


Image 44: MXE TaskEngine config for switching an external relay connected to *GPO 1* via *VLogic 1*.

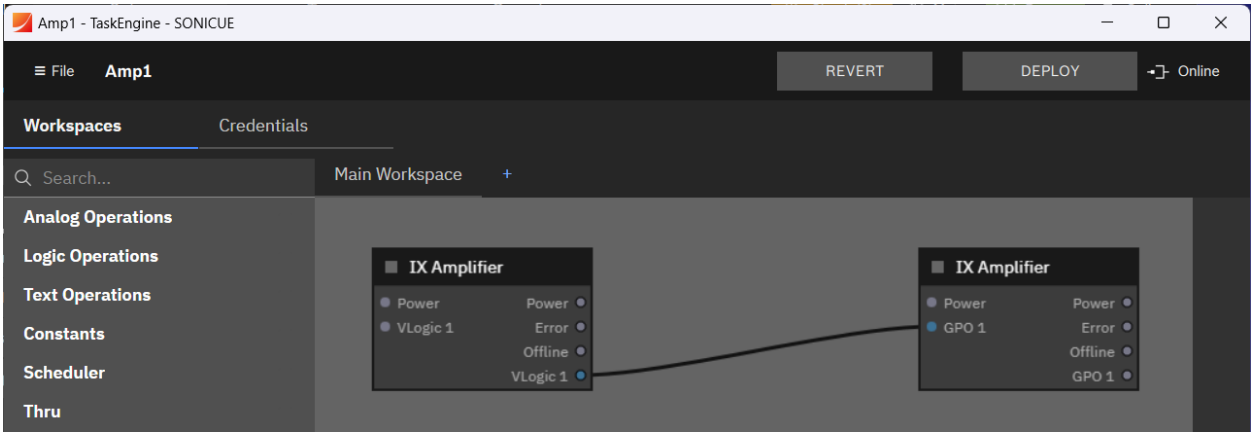


Image 45: IX TaskEngine config for switching an external relay connected to *GPO 1* via *VLogic 1*.

6. Example External LEDs

This example shows how **external LEDs**, connected to **IX** or **MXE Control Port** (**GPIOs** or **Ready/Fault** contact), can be used to signalize **device** or **system status**.

Wiring diagram

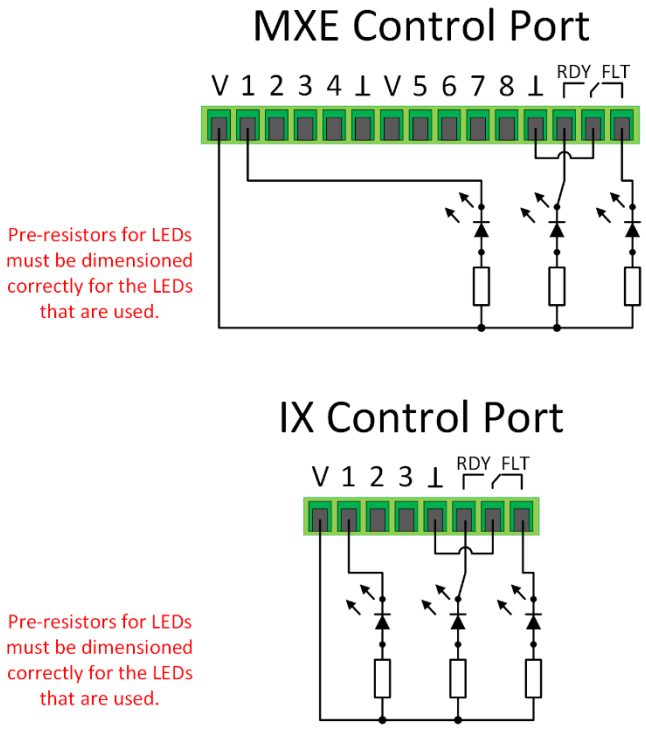


Image 46: Connection of external LEDs to *MXE* or *IX Control Port*

GPIO configuration

Make sure that in SONICUE under **Setup>GPIO** the GPIO that shall be used is set to the correct type. In our example we need ***GPIO 1*** configured as ***DIGITAL OUT.***

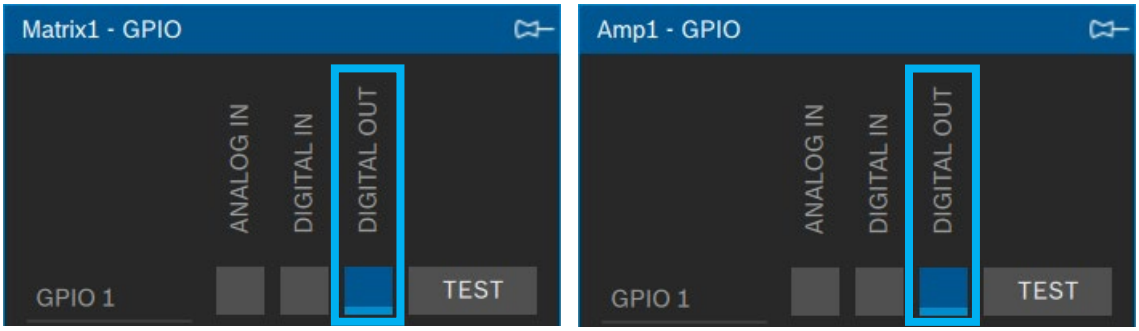


Image 47 and 48: Setting *Matrix1* (left) or *Amp1* (right) *GPIO 1* to *DIGITAL OUT*.

Ready/Fault configuration

The status of the potential-free **Ready/Fault** relay contacts depends on the configuration of the IX or MXE ***State Flags*** under **Setup>Status** and if the **device** has booted completely = **ready**.

If either the ***Device Not Ready*** flag, or **any other flag** that has been selected in the ***Collect* column** as part of the ***Collected Error State*** is active, the IX or MXE will show a fault on the front panel with the Fault LED being active and the Fault relay contact will be closed.

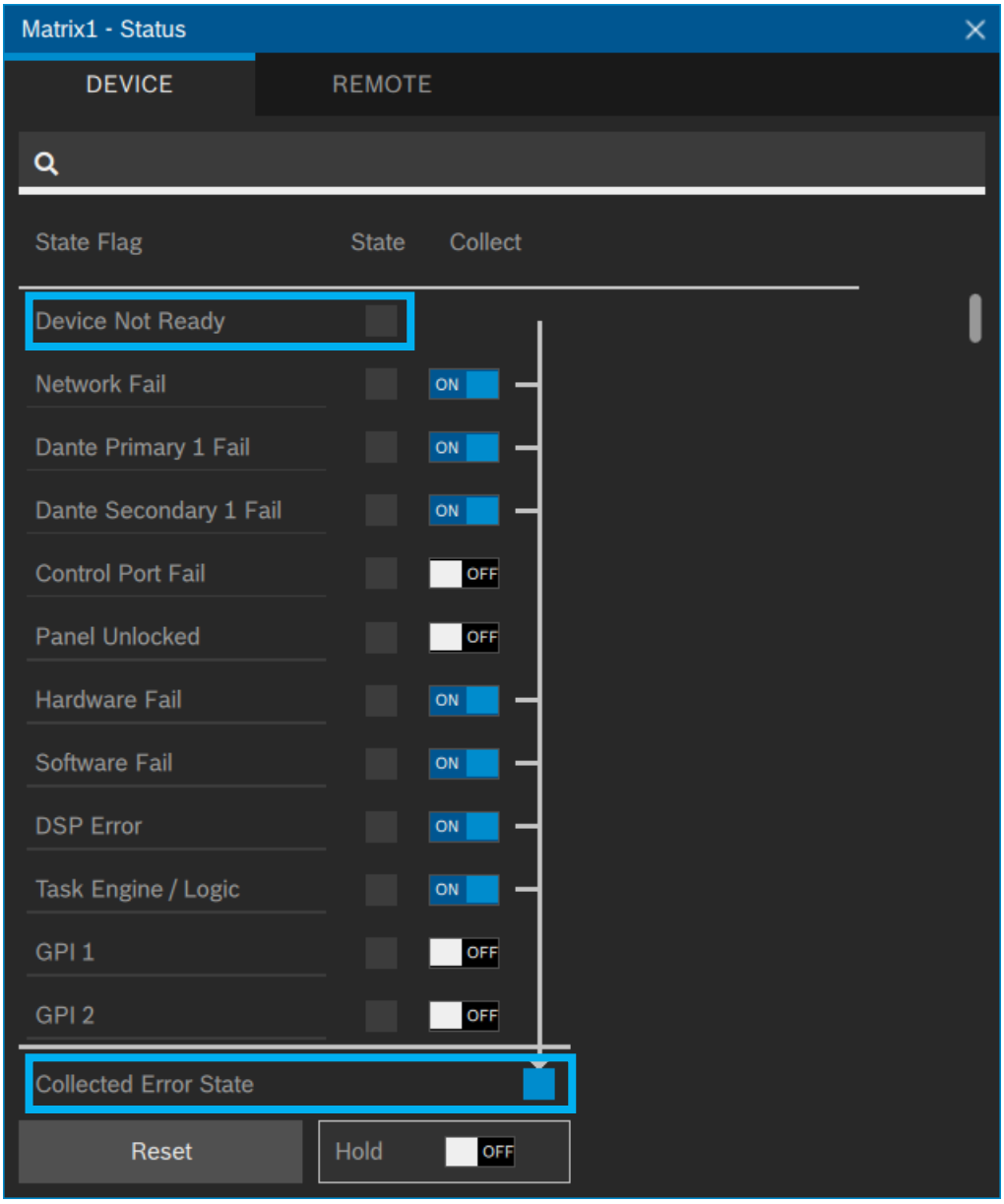


Image 49: MXE *Status* flyout – *DEVICE* tab.

In the same ***Status*** flyout, under the ***REMOTE*** tab, the **status of other devices** (MXE, IPX, IX) on the same network can be included in the ***Collected Error State***. By doing this, the **MXE's Fault relay** can be used to signalize not just a **device fault**, but also a **system fault**.

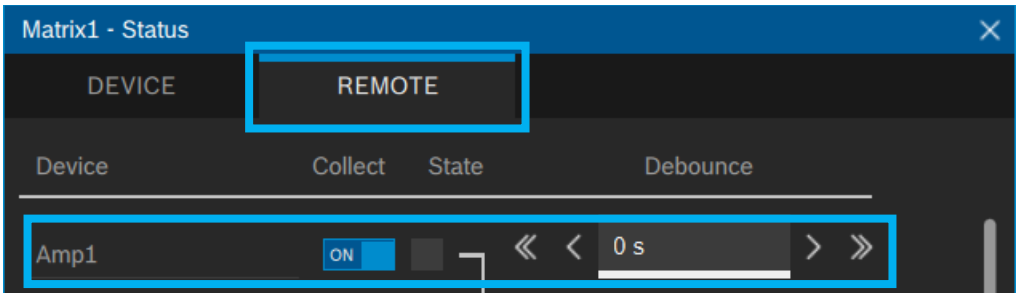


Image 50: MXE *Status* flyout – *REMOTE* tab.

TaskEngine programming

If **other information** than device or system status (via Ready/Fault) shall be signalized, a **GPIO** can be set as **DIGITAL OUT**. Task Engine logic can then be configured to trigger the GPIO with an LED connected.

In the following example, device ***Power*** status is reported via ***GPO 1***.

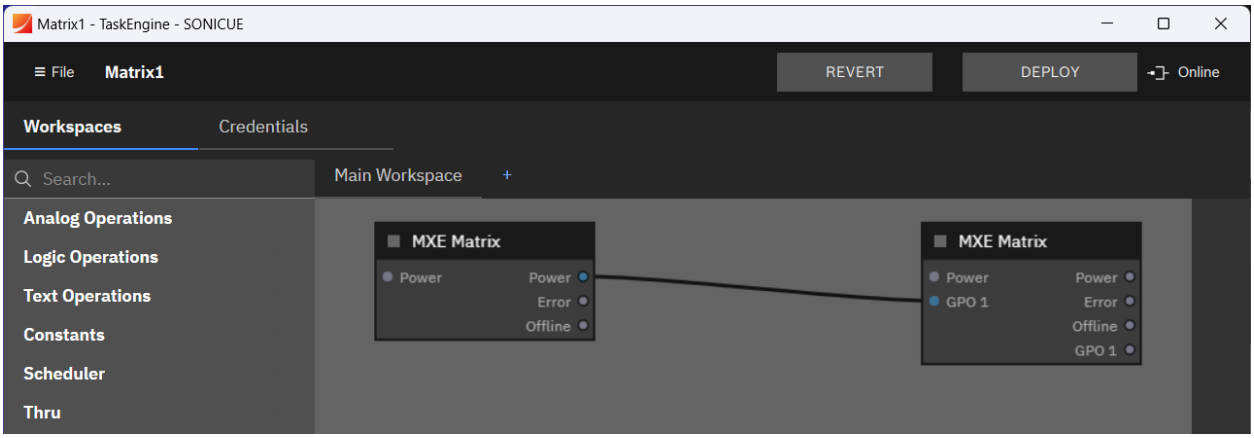


Image 51: MXE TaskEngine config for signalizing *Power* status via an external LED connected to *GPO 1*.

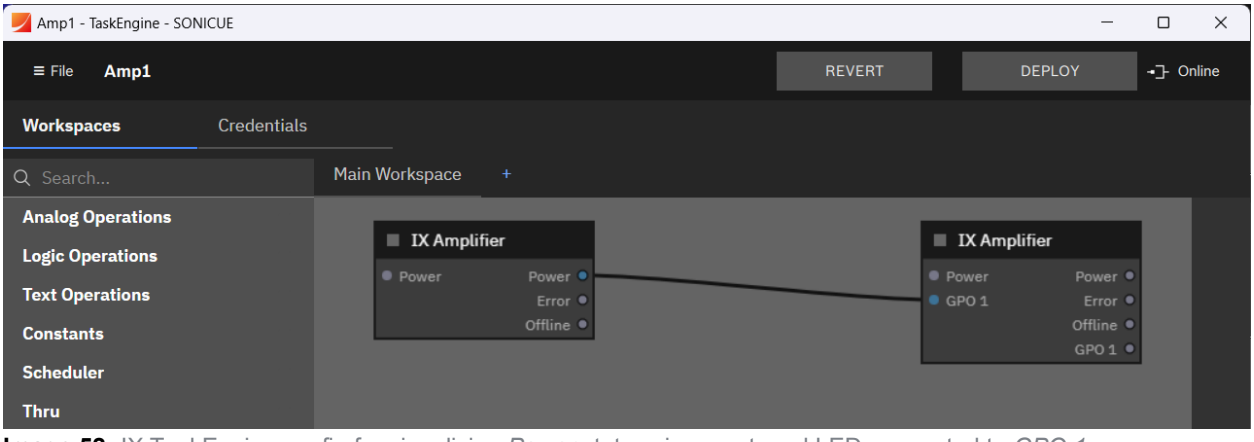


Image 52: IX TaskEngine config for signalizing *Power* status via an external LED connected to *GPO 1*.

Third party product disclaimer:

Dynacord does not take responsibility for the warranty, quality, or availability of standard electronic components (potentiometers, resistors, relays, LEDs, etc.). The standard electronic components contained within this document were tested successfully at the time of publication. However, Dynacord cannot guarantee the compatibility or availability of such standard electronic components in the future.