应用指南

通过 IX/MXE 的控制端口(GPIOs) 对 IX 功率放大器及 MXE 混音矩阵的模拟控制

IX 功率放大器和 **MXE 混音矩阵**都配有**控制端口，**使用模拟控制连线和控制与其他系统对接。

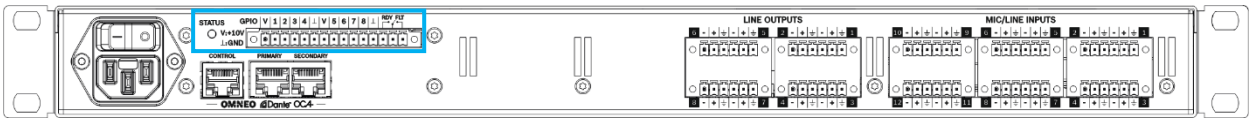


图 1： MXE 后面板视图

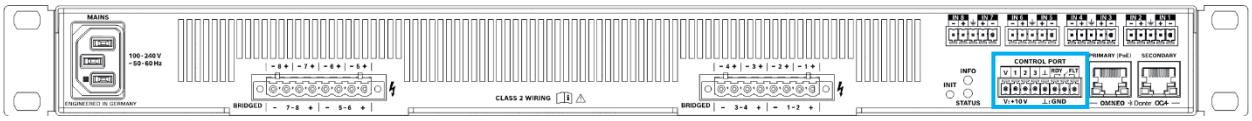
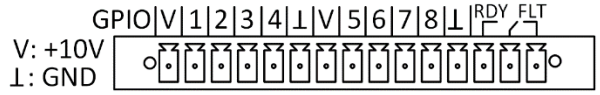


图 2：IX 后面板视图（图示为 8 通道型号）

控制端口（*GPIO* 或 *CONTROL PORT*）可以在 MXE 和 IX 的后面板上找到。提供包括：**8 个（MXE）**或 **3 个（IX）**可自由配置的 **GPIOs**（General Purpose Inputs and Outputs，通用输入和输出控制接口），**1 个 正常/故障（Ready/Fault）**继电器（*RDY/FLT*）以及 +10 V (*V*) 和 接地 ground (⊥)参考针。

GPIOs 可以通过 **SONICUE** 配置成 **Analog In（模拟输入），Digital In（数字输入），**或 **Digital Out（数字输出）。**



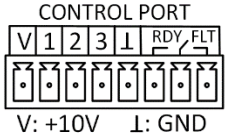


图 3 及图 4： MXE （左）及 IX（右）控制端口详图

MXE 任务引擎（MXE Task Engine）的 使用需求

MXE 混音矩阵固件版本 1.4.3119 （或更高）

SONICUE 声系统软件 1.3.0 （或更高）安装到电脑上

IX 任务引擎（IX Task Engine）的使用需求

IX 功率放大器 固件版本 1.0.0 （或更高）

SONICUE 声系统软件 1.5.0 （或更高）安装到电脑上

任务引擎（TaskEngine）快速使用指南

每台 IX 和 MXE 设备都包含了 1 个强大的**逻辑处理引擎（logic processing engine），**独立于音频数字信号处理（DSP）。**任务引擎（TaskEngine）**提供全部工具和 1 个宽泛的逻辑控制模块选择，可以用于创建复杂的集成控制系统。**提供不同的 SONICUE 硬件，面板设计（Panel Designer）用户界面**与**很多第三方设备之间的连接。**

IX 和 **MXE 任务引擎**不仅可以调用自身的控制参数，还能**调用相同网络中其他设备的控制参数。**

- MXE 可以控制其他 MXE, IPX, IX, OCA 设备以及第三方支持 API 的产品
- IX 可以控制其他 IX 以及第三方支持 API 的产品

配置输入和输出节点

在（MXE）**任务引擎**中 ***MXE 矩阵，IPX 功放***和 ***IX 功放***模块，默认拥有 ***1 个***输入节点用于***电源（Power）开关，***输出节点用于***电源（Power），故障（Error）***或***离线（Offline）***状态。

IPX 功放默认带有**额外用于报警（Alarm）**预设的输入和输出节点。

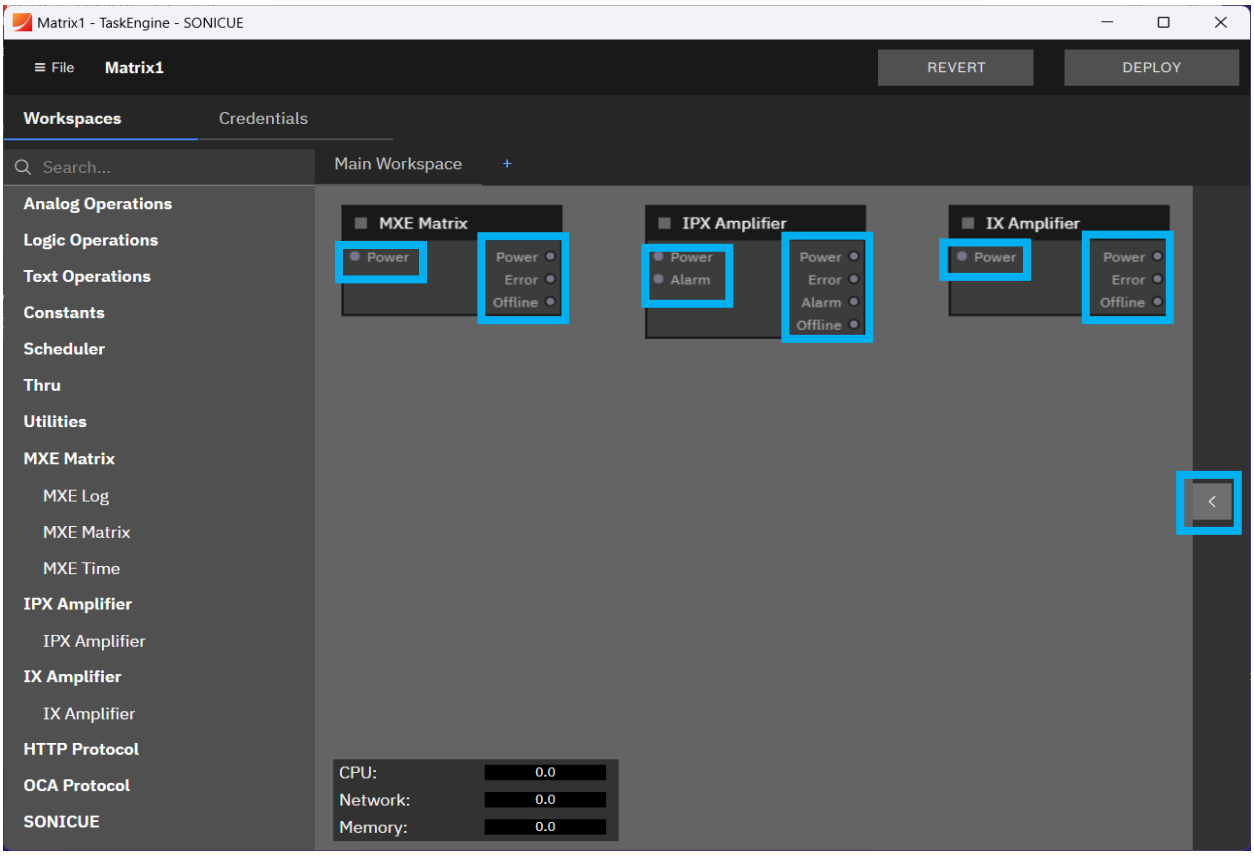


图 5：MXE 任务引擎（TaskEngine）带有 MXE 矩阵，IP 功放和 IX 功放模块

选择 1 个矩阵或功放**模块**，**单击**任务引擎工作区（TaskEngine workspace）右侧  **按钮**，就会打开该逻辑模块的属性窗口。

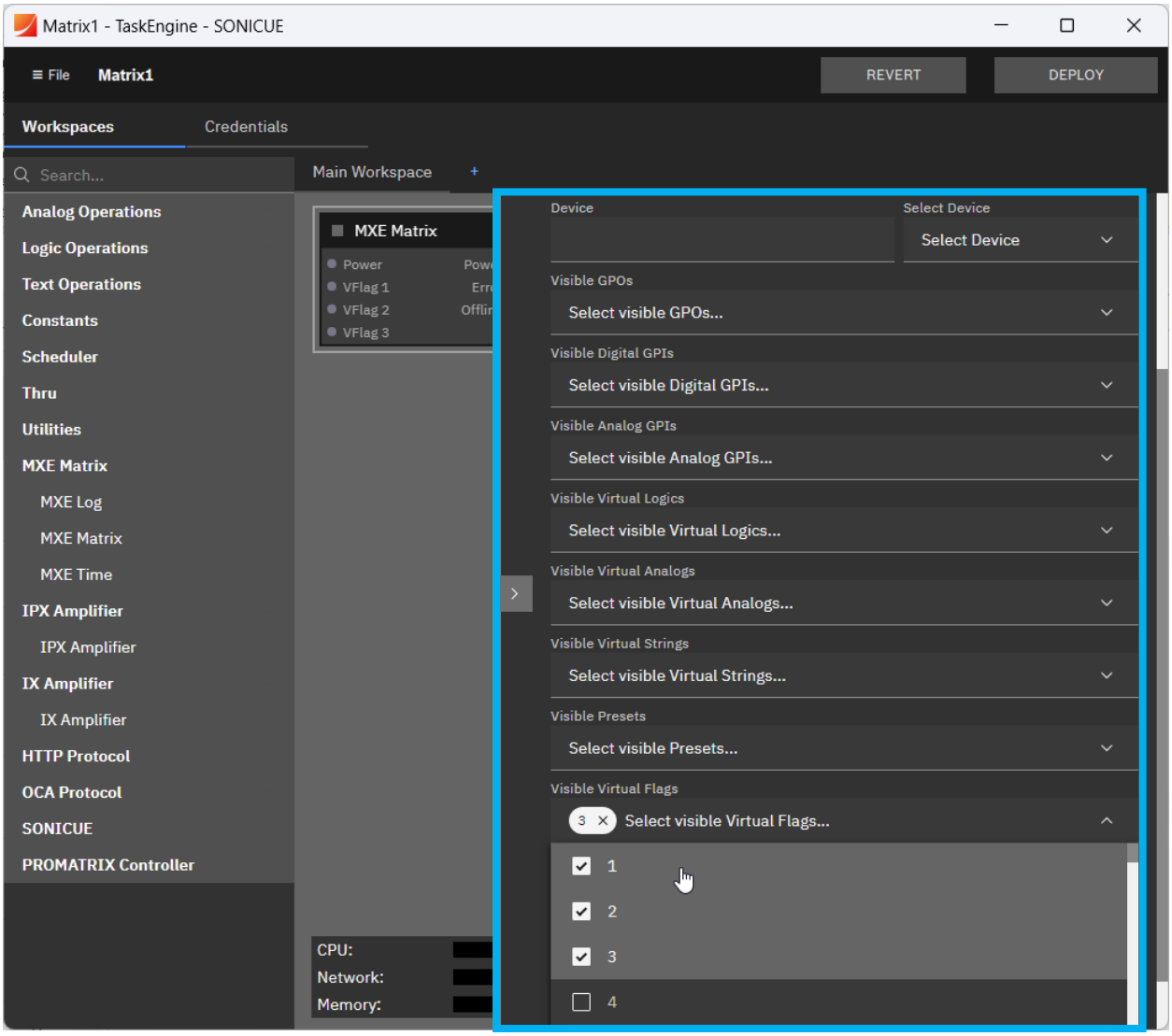


图 6： MXE 任务引擎工作区打开了 MXE 矩阵模块属性窗口，并选择其中 *Visible Virtual Flags 1-3*

MXE 混音矩阵 逻辑模块，可以添加如下节点：

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

*MXE 拥有 8 个 GPIOs 可以用作 GPO 或者数字 GPI 或者模拟 GPI。

IX 功率放大器 逻辑模块，可以添加如下节点：

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

*IX 拥有 3 个 GPIOs 可以用作 GPO 或者数字 GPI 或者模拟 GPI。

IPX 功率放大器逻辑模块，可以添加如下节点：

- 多达 **3 个** ***Visible GPOs***
- 多达 **3 个** ***Visible Digital GPIs***
- 多达 **3 个** ***Visible Analog GPIs***
- 多达 **20 个 User + 1 个 Factory** ***Visible Presets***

*IPX 拥有 3 个 GPIOs 可以用作 GPO 或者数字 GPI 或者模拟 GPI。

选择正确的硬件设备

选择一个矩阵或功放**模块**，然后**单击**任务引擎工作区右侧的  **按钮**，打开该模块的属性窗口

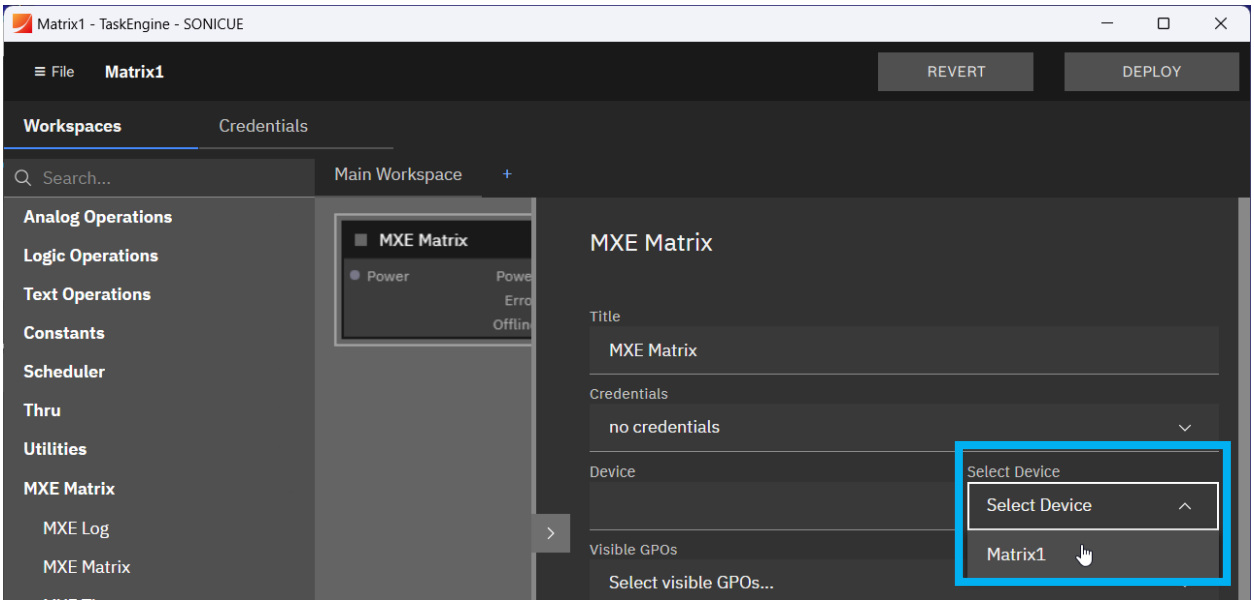


图 7： 通过属性窗口>设备选择（Select Device）为 MXE 矩阵模块选择 Matrix1 设备

由于网络上可能存在**多台** MXE **矩阵**或者 IX/IPX **功放**，通过***设备选择（Select Device）*****选择正确的硬件设备**非常关键。因此，在网络上的设备需要被显示出来。或者，如果设备并没有连接到网络中（当前），则可以在设备（Device）区域手动输入正确的设备名称。

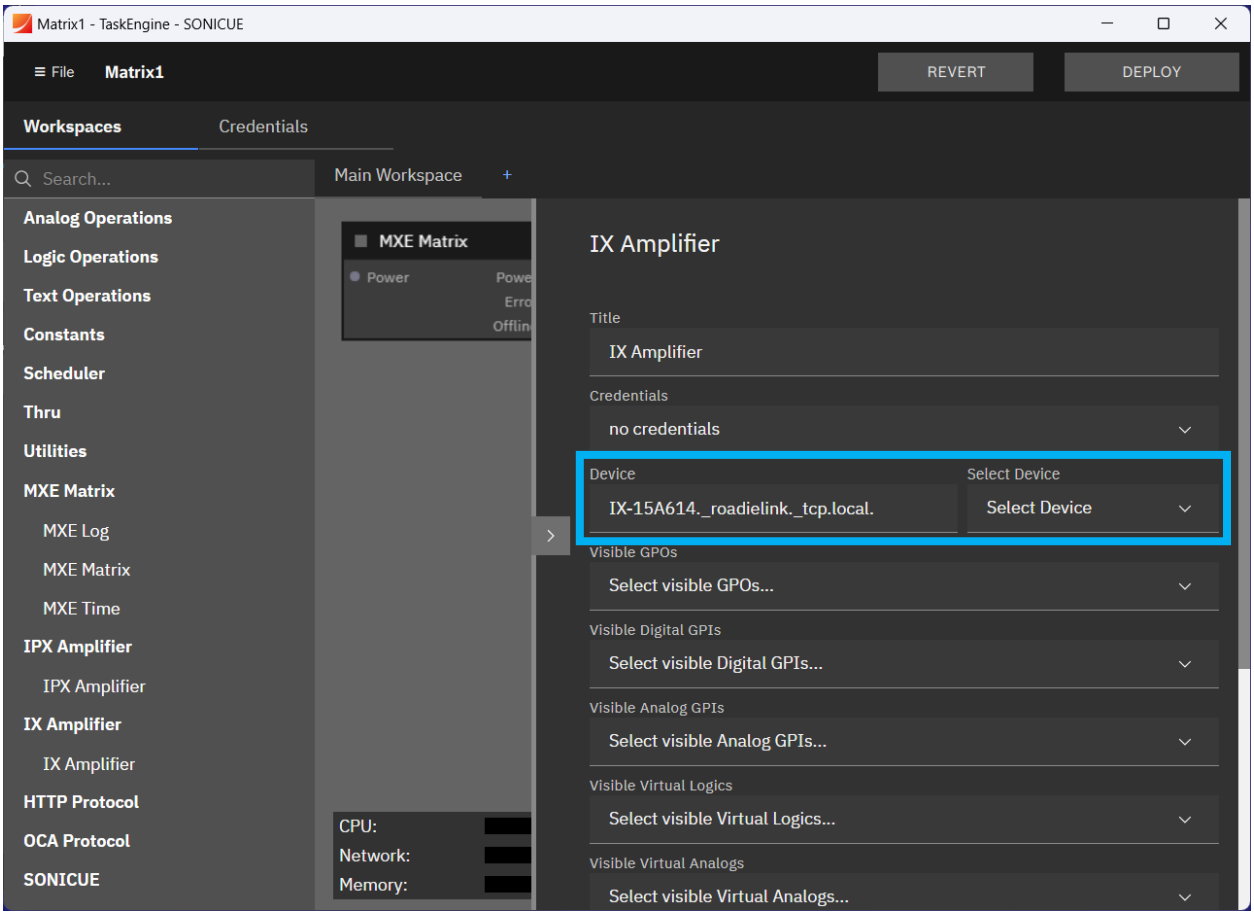


图 8: 通过 *Select Device 选择 Amp1*，独一无二的服务名称 *IX-15A614* 是*设备名称*的一部分

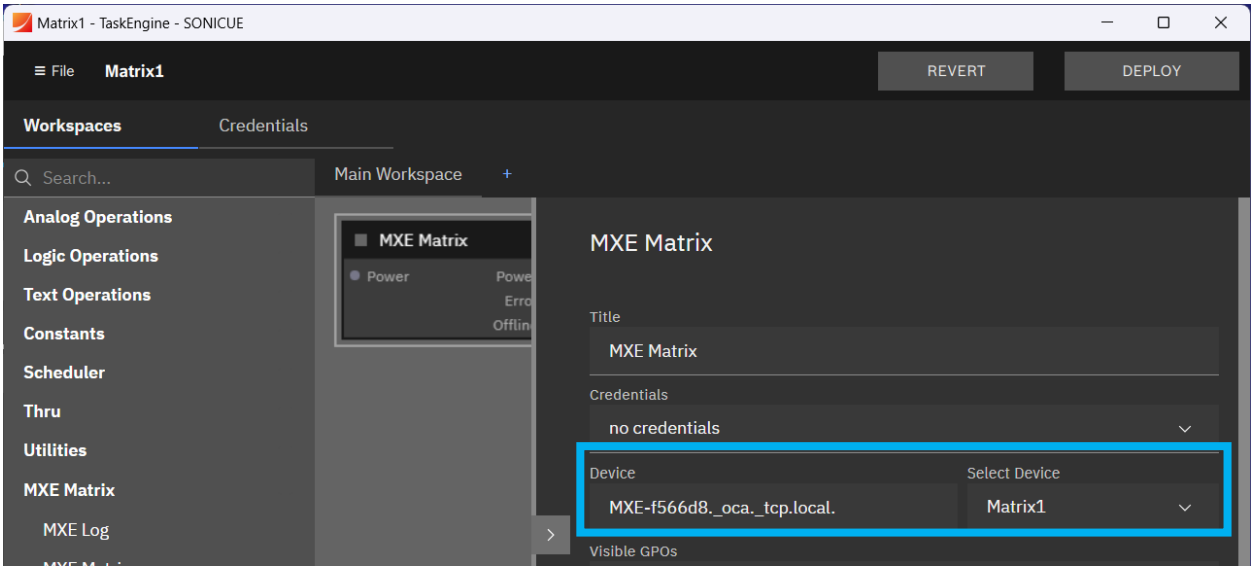


图 9: 通过 *Select Device 选择 Matrix1*，独一无二的服务名称 *MXE-f566d8* 是*设备名称*的一部分

添加 SONICUE 表达

SONICUE 版本 **1.3.0...1.4.0** 中，添加 ***Logic*** 和 ***Analog*** **表达**的按钮在任务引擎窗口的**右上角**

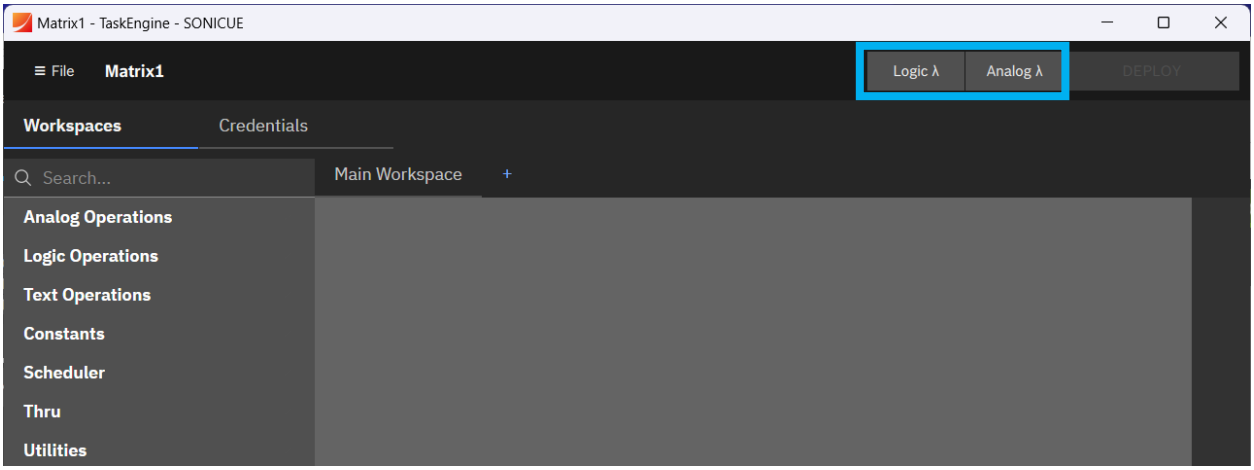


图 10: *Logic* / 和 *Analog* / 按钮，用来添加 SONICUE 表达到任务引擎（TaskEngine）

SONICUE 版本 **1.5.0 或更高，** 逻辑和模拟 ***SONICUE Expression*** 可以从**任务引擎目录**，**SONICUE** 菜单中添加。

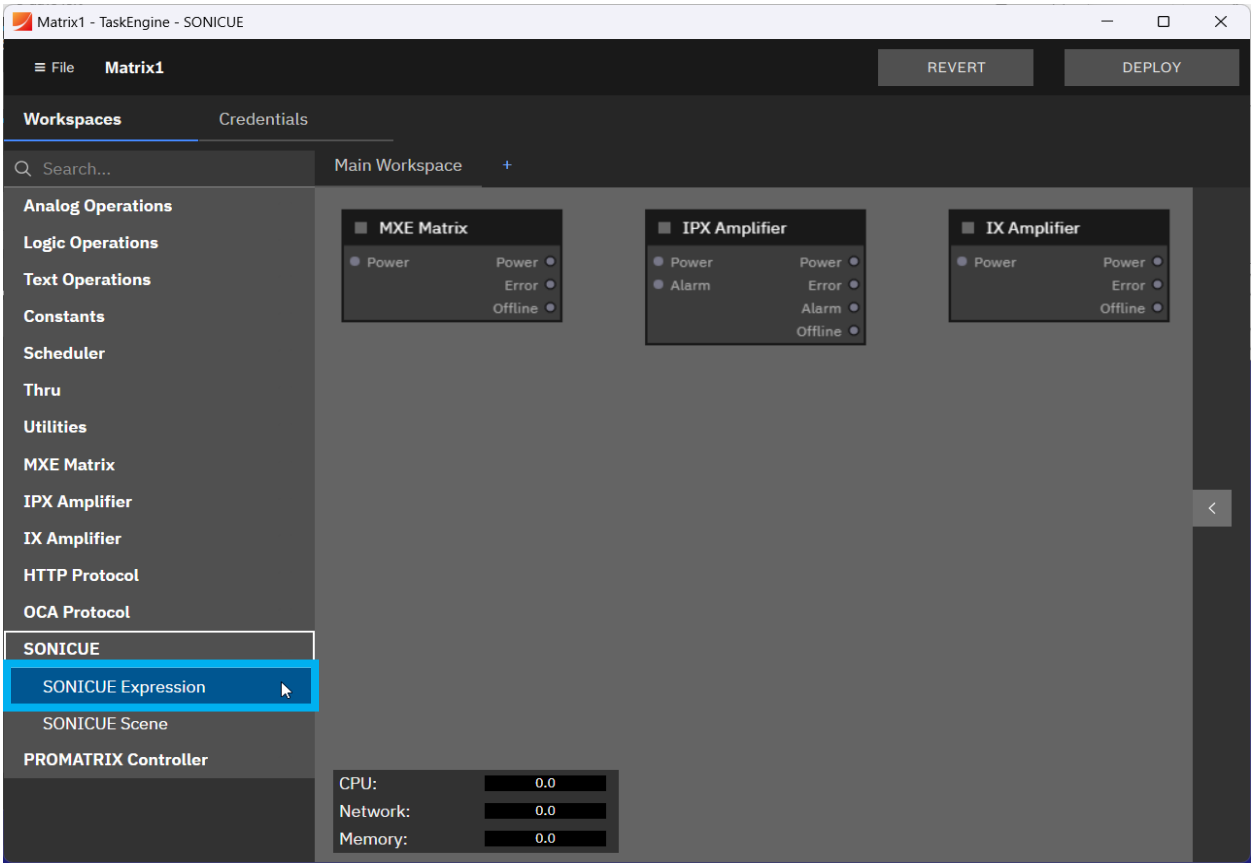


图 11: SONICUE 菜单中的 *SONICUE Expression*。

作为替代， **DSP 参数**可以直接从 **DSP 弹窗**中通过 **Ctrl + 拖/放**进行添加，甚至多重选择（=group）。

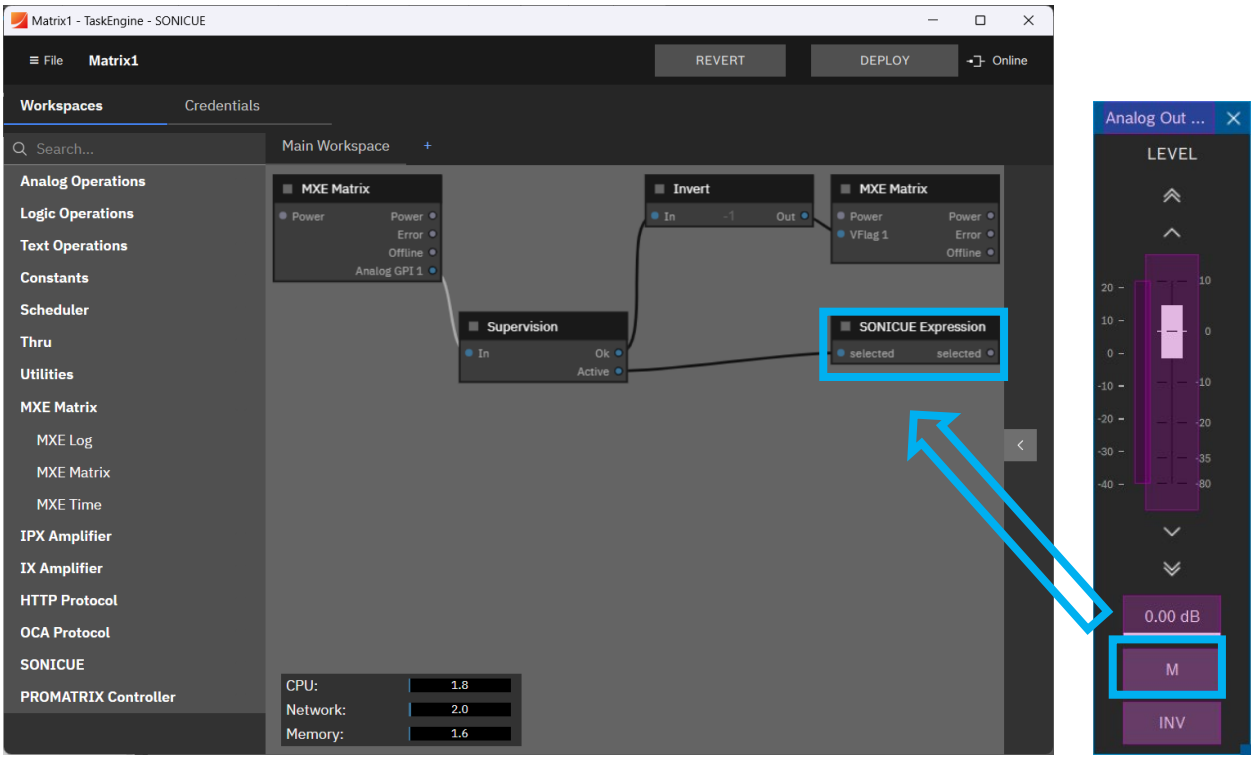


图 12 和 13: 从输出弹窗通过拖/放添加 *SONICUE Expression* 到任务引擎。

应用逻辑配置

在创建或修改**任务引擎逻辑**后**最重要的步骤**就是当**设备连线**后**应用（Deploy）逻辑**到硬件。而连线写入（WRITING）时逻辑是自动应用的。

图 14: 当任务引擎配置还没有被应用时 *DEPLOY* 按钮“激活”。

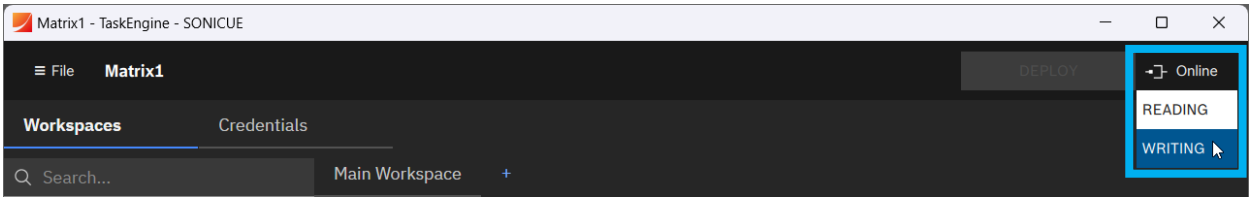


图 15: 通过 *WRITING* 连线将自动应用任务引擎配置。

图 16: 当任务引擎配置已经被应用时 *DEPLOY* 按钮“不激活”。

1. 模拟电平控制示例

本示例展示一个**模拟电位器**连接到 **IX 或 MXE 的控制端口（GPIO），**如何用来控制一个 **DSP 电平**。

连线图

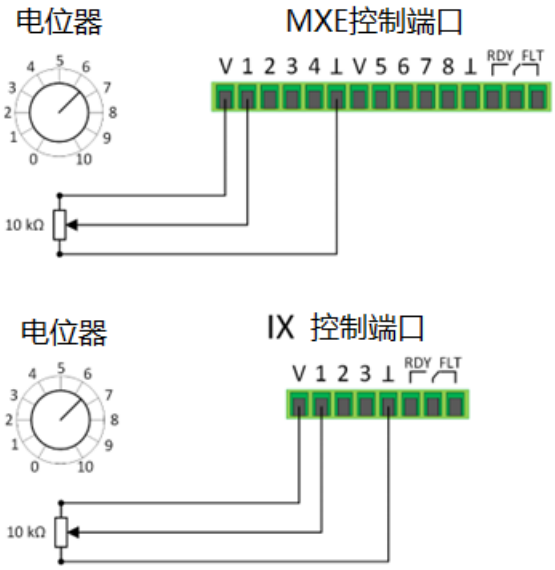


图 17：连接 1 个 10KΩ的模拟电位器到 MXE 或 IX 的控制端口

GPIO 配置

确保 SONICUE 中 **Setup>GPIO** 下的 GPIO 应设置成正确的类型，在我们的示例中我们需要 ***GPIO1*** 配置为 ***ANALOG IN（模拟输入）。***

图 18 和 19：设置 *Matrix1*（左）或 *Amp1*（右） *GPIO1*为 *ANALOG IN*。

TaskEngine 编程

接下来的 TaskEngine 架构中， MXE 的 ***Analog GPI 1*** 用来控制一个 DSP **电平** (*SONICUE Expression*)。***Analog Scaler*** 模块将控制**电压范围** (0...10 V) 转换成通过模拟电位器控制的 DSP **电平范围**（比如-50...0dB）。

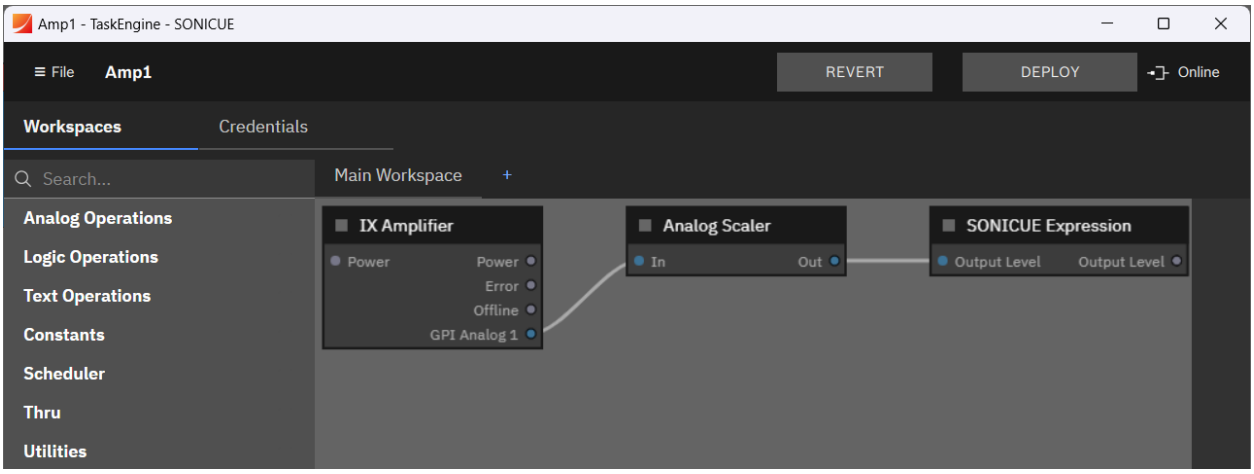


图 20: IX 任务引擎配置，使用 *Analog Scaler* 模块。

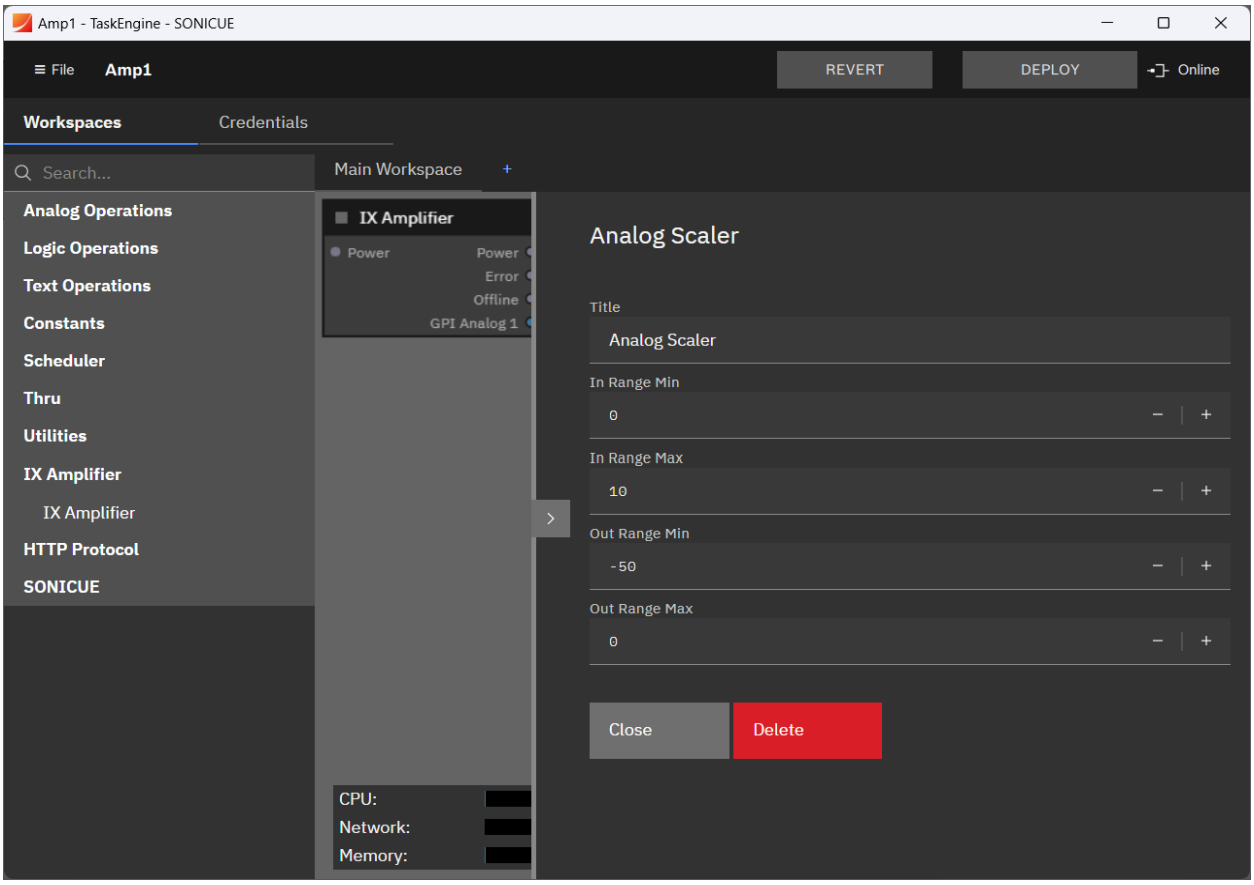


图 21: 打开 *Analog Scaler* 模块属性窗口，给一个分区电平控制使用典型值（-50...0 dB）。

2. 火警静音示例

本示例展示一个连接到 IX 或 MXE 控制端口（GPIOs）的无电势继电器开关如何用来激活 DSP 静音。这个连接使用 2 个阻抗进行监测。

连线图

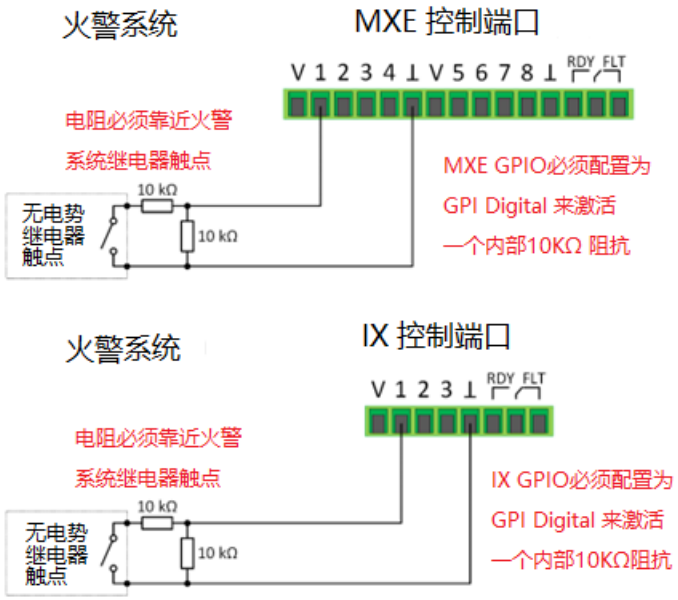


图 22: 连接火警系统的无电势继电器触点到 MXE 或 IX 控制器端口

GPIO 配置

确保 SONICUE 中 **Setup>GPIO** 下的 GPIO 应设置成正确的类型，在我们的示例中我们需要 ***GPIO1*** 配置为 ***Digital IN（数字输入）。***

图 23 和 24: 设置 Matrix1 （左）或 Amp1 （右） GPIO1 为 Digital IN。

TaskEngine 编程

接下来的 TaskEngine 架构中，一个 MXE ***Analog GPI 1*** 用来激活多个输出通道的 DSP **静音**。Analog GPI 1 的**电压范围**用一个**监测模块**来监看。**监测电压**的超出范围**故障**将报告给 MXE 的虚拟标志 ***VFlag 1***。请观察 ***Invert*** 模块，当出现 **not OK** 时用来触发 VFlag 1 (User Flag 1)。

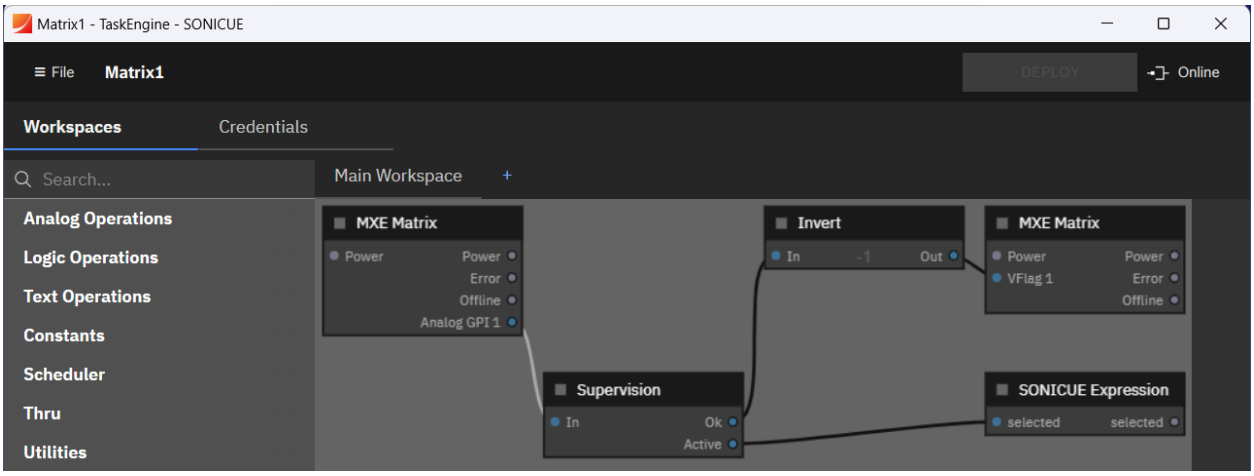


图 25： MXE TaskEngine 逻辑， 使用 *Supervision* 模块来监测一个模拟控制连接。

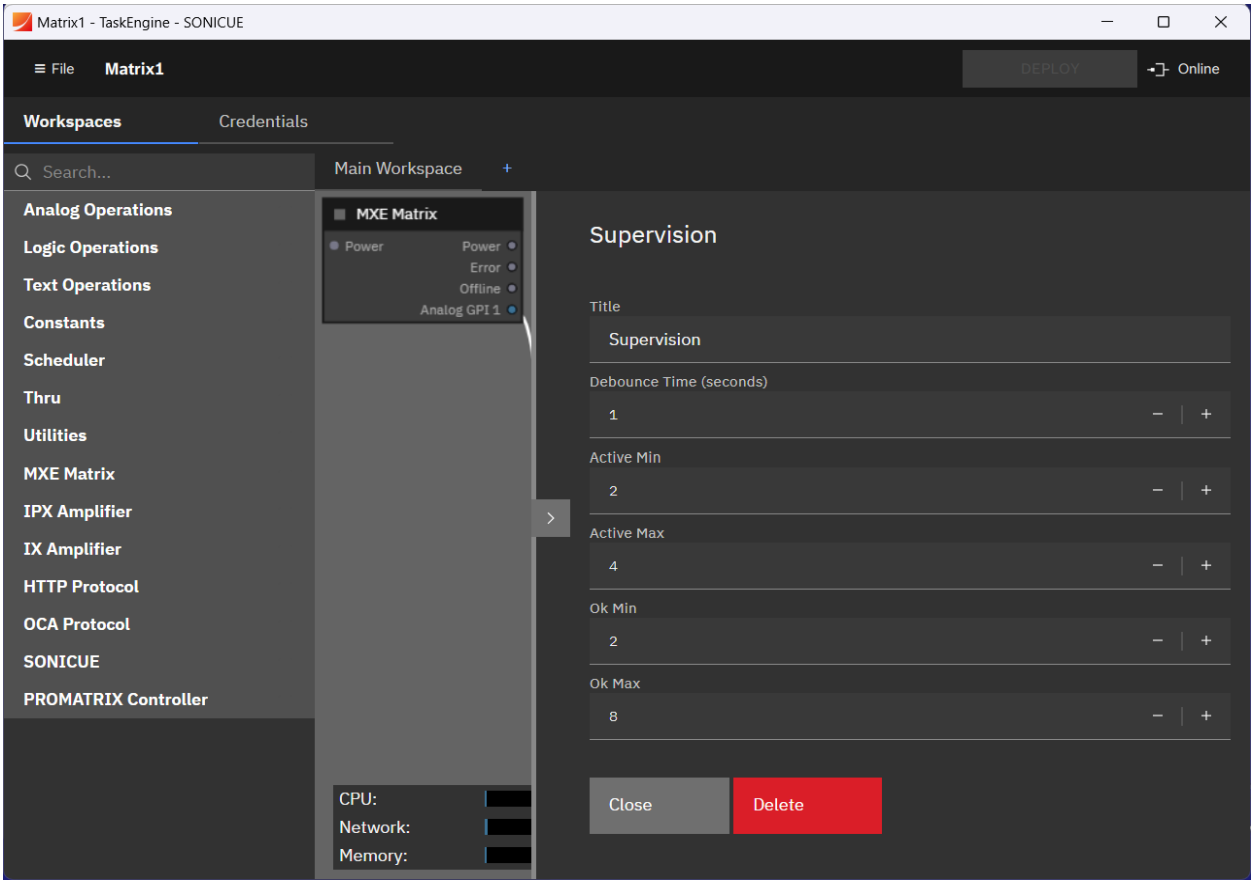


图 26： 监测模块的属性窗口，打开显示详细配置（全部默认值）。

3. 开机/待机切换示例

本示例展示连接到 **IX** 或 **MXE 控制端口（GPIOs）**的**拨动或瞬时开关**如何用来切换设备的**开机/待机**。

连线图

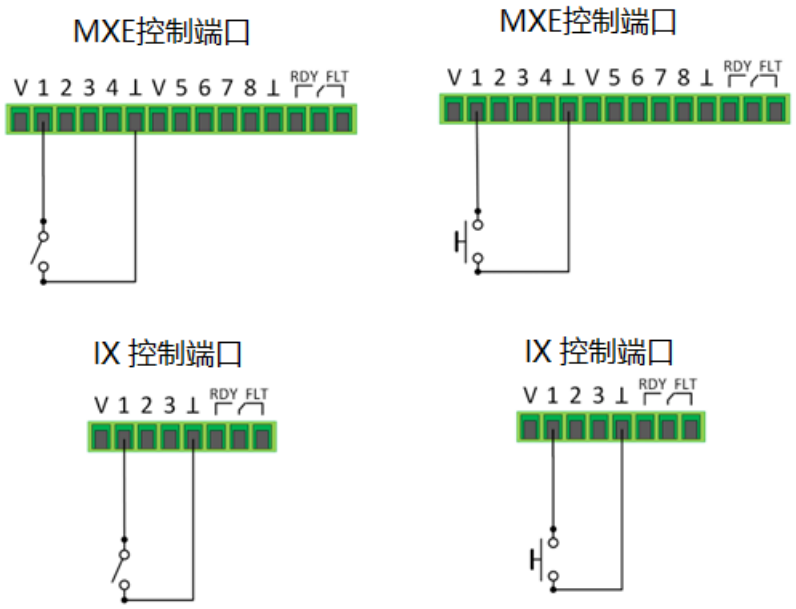


图 27 和 28：连接拨动开关（左图）或瞬时开关（右图）到 MXE 或 IX 的控制端口。

GPIO 配置

确保 SONICUE 中 **Setup>GPIO** 下的 GPIO 应设置成正确的类型，在我们的示例中我们需要 ***GPIO1*** 配置为 ***Digital IN（数字输入）。***

图 29 和 30： 设置 Matrix1 （左）或 Amp1 （右） GPIO1 为 Digital IN。

TaskEngine 编程

接下来的 TaskEngine 架构中，一个 MXE ***Digital GPI 1*** 用来切换 MXE ***Power***，使用拨动开关（锁定）连接到 GPIO1.

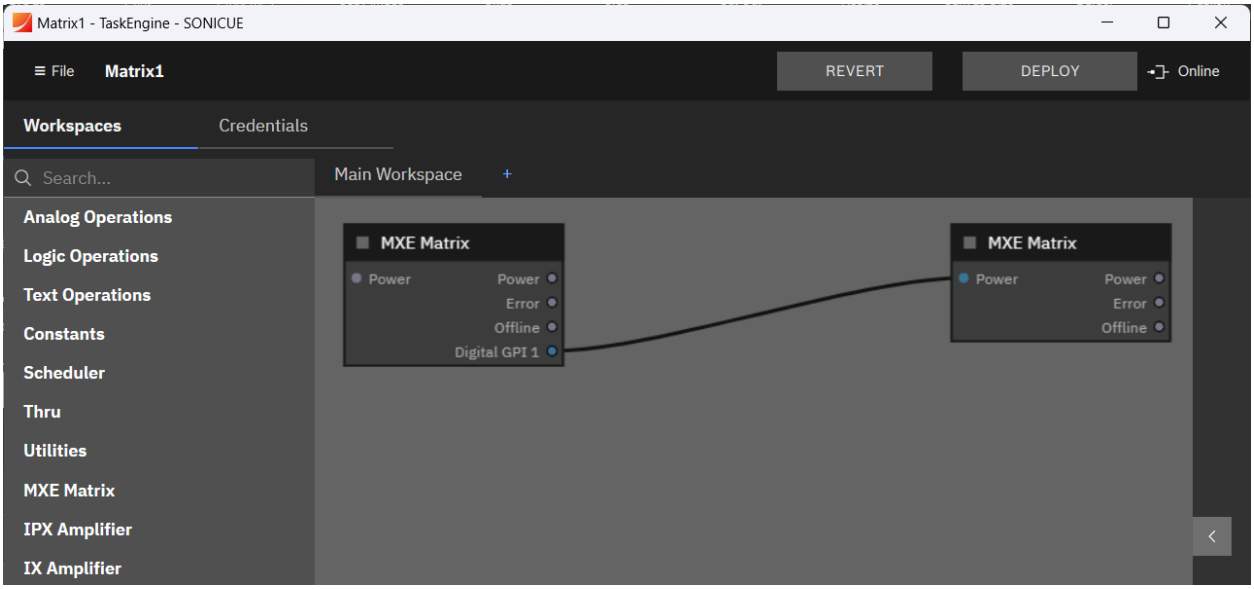


图 31： MXE TaskEngine 配置用来*通过* MXE *Digital GPI 1* (使用拨动开关) 开关 MXE *Power*。

下面这个修改过的 TaskEngine 架构中，一个 IX ***Digital GPI 1*** 用来切换 IX ***Power***，使用瞬时开关（无锁定）连接到 GPIO1。

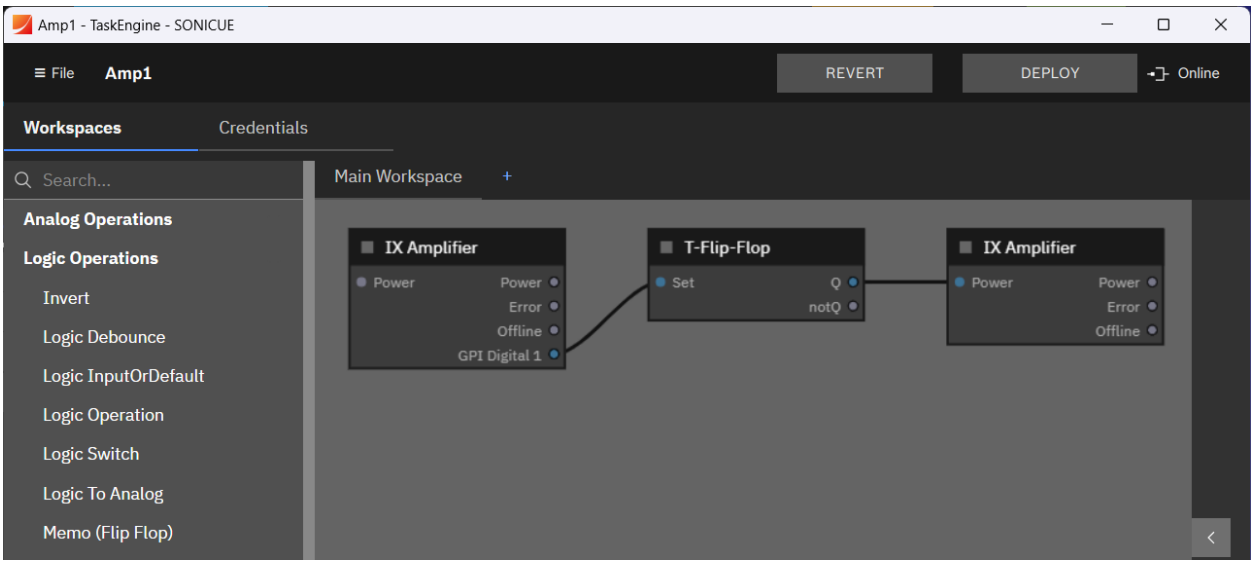


图 32： IX TaskEngine 配置使用 T 翻转触发器通过 IX *Digital GPI 1* （使用瞬时开关）来切换 IX *Power*。

4. 预设或场景调用示例

本示例展示连接到 **IX** 或 **MXE 控制端口（GPIOs）**的**瞬时开关**，如何用来**预设调用**或**场景调用**。

连线图

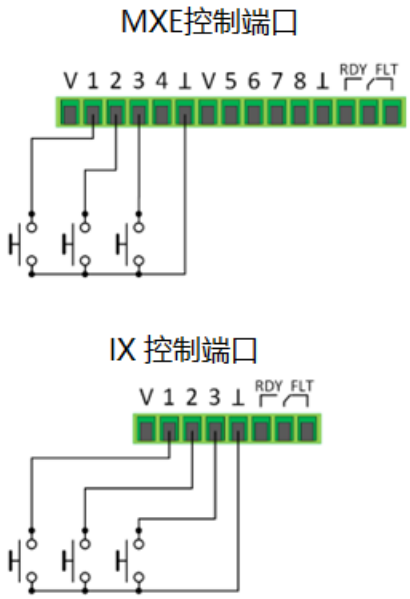


图 33：连接按钮开关到 MXE 或 IX 控制端口

GPIO 配置

确保 SONICUE 中 **Setup>GPIO** 下的 GPIO 应设置成正确的类型，在我们的示例中我们需要 ***GPIO1+2+3*** 配置为 ***Digital IN（数字输入）。***

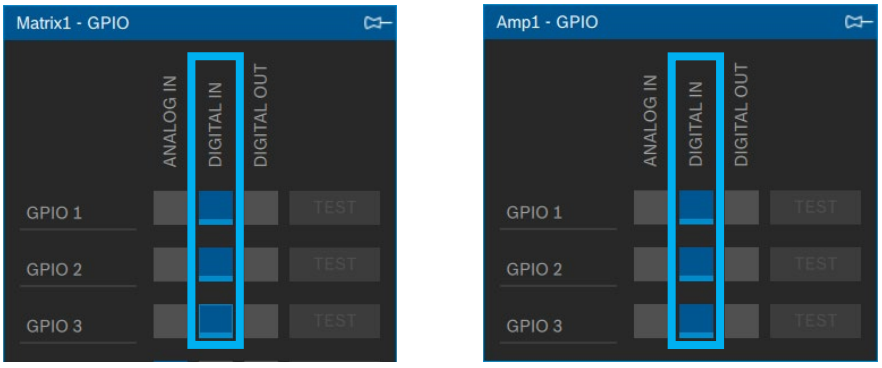


图 34 和 35：设置 Matrix1 （左）或 Amp1 （右） GPIO1、2 和 3 为 Digital IN。

计划调用预设

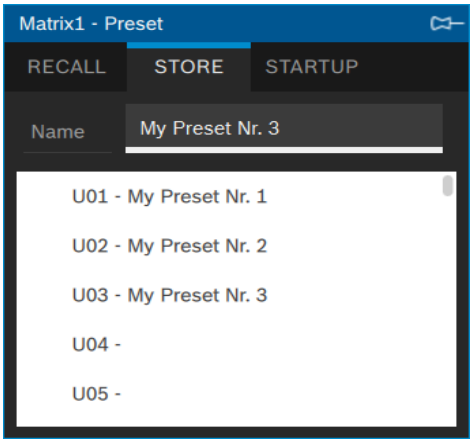


图 36: Matrix1 – *预设弹窗，预设 U01 – U03* 已保存

TaskEngine 编程用于预设调用

接下来的 TaskEngine 架构中，IX 的 ***GPI Digital 1***、*2* 和 *3* 被用来调用用户预设 ***Preset U01***、 *U02* 和 *U03*。

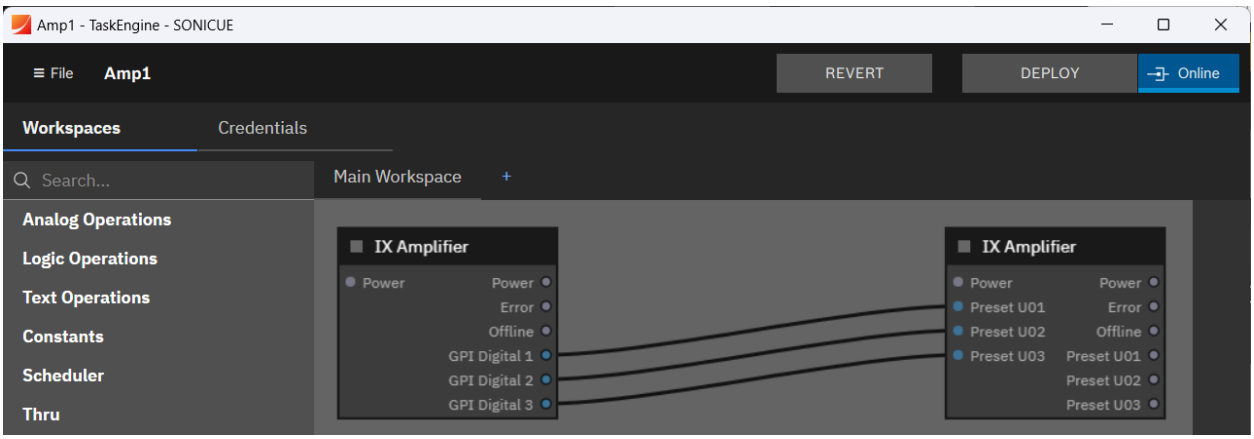


图 37: IX TaskEngine 配置，通过 GPIOs (*GPI Digital 1*、 *2* 和 *3*)来调用预设 *Preset U01*、*U02* 和 *U03* 。

作为替代， ***VLogic 1***、 *2* 和 *3* 可以用于**第三方设备**通过 **http API** 进行调用

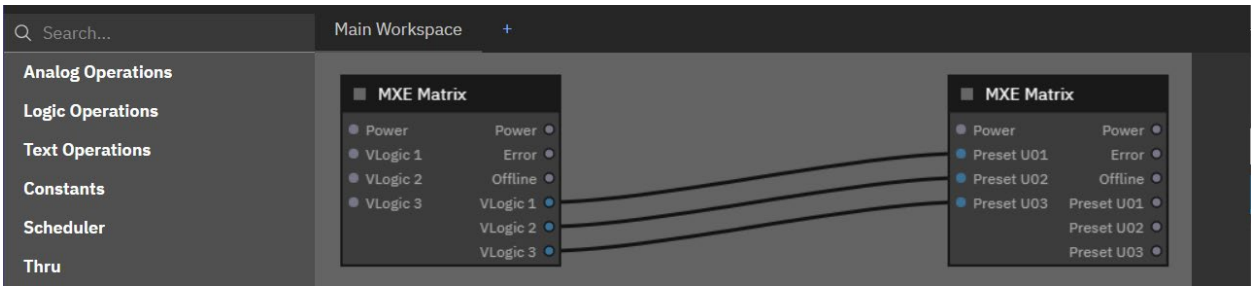


图 38: MXE TaskEngine 配置，通过 Virtual Logics (*VLogic 1*、*2* 和 *3*) 来调用预设 *Preset U01*、*U02* 和 *U03*。

TaskEngine 编程用于场景调用

接下来的 TaskEngine 架构中， IX 的 ***GPI Digital 1、2*** 和 ***3*** 用来调用 ***SONICUE Scene 场景模块中场景槽位 1、2 和 3.***

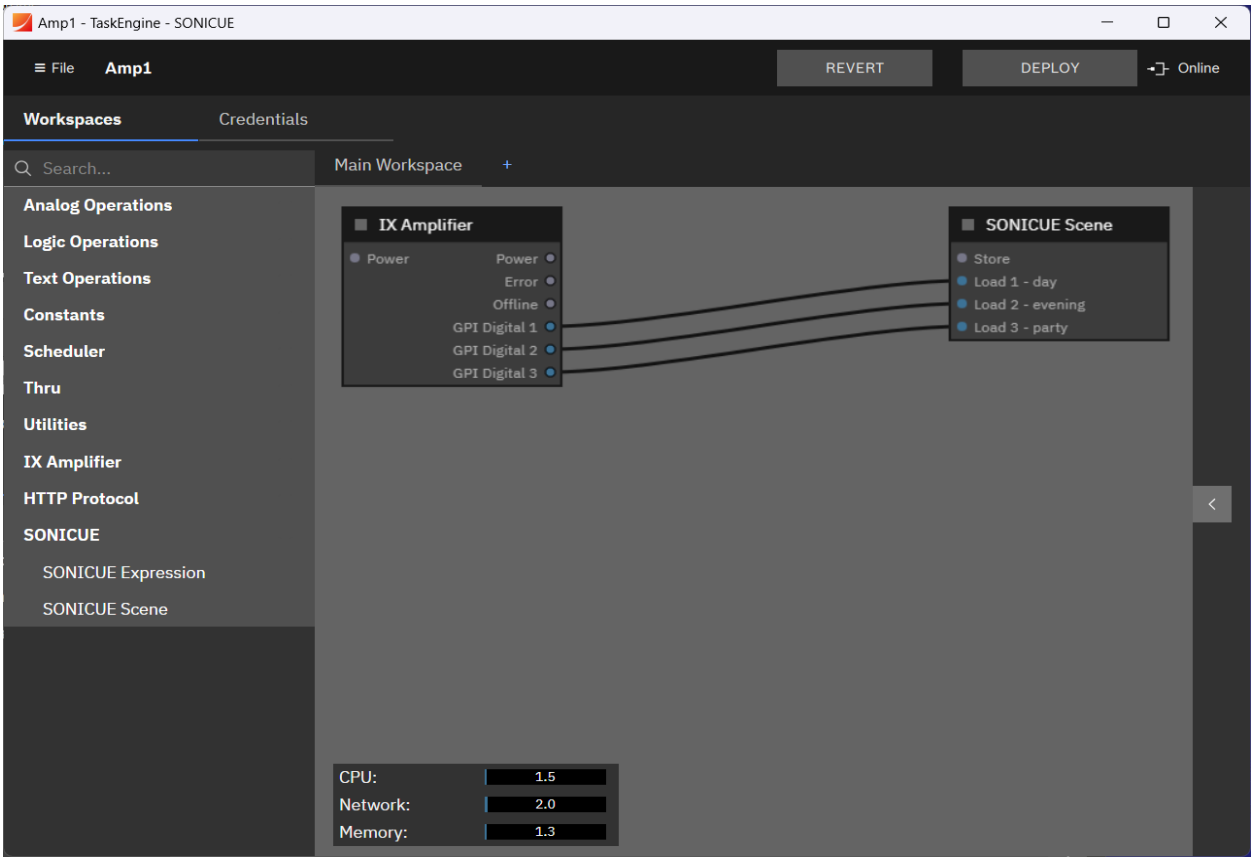


图 39： IX TaskEngine 配置，通过 GPIOs (*GPI Digital 1、2* 和 *3*)来调用预设 *Preset U01、U02* 和 *U03* 。

同样这里作为替代， ***VLogic 1、 2*** 和 ***3*** 可以用于**第三方设备通过 http API** 或在 SONICUE PanelDesigner 中创建的 **SONICUE Control** 用户界面中的按钮。

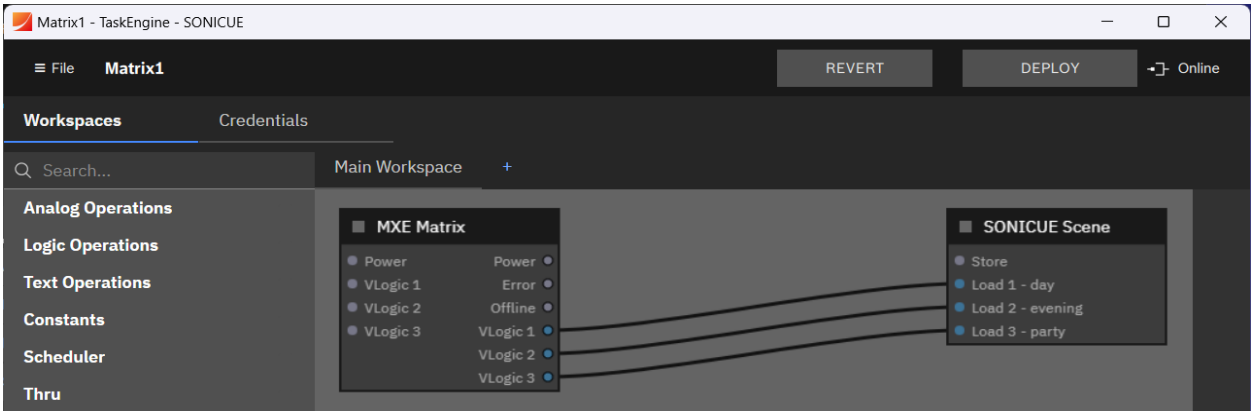


图 40： MXE TaskEngine 配置，通过 Virtual Logics (*VLogic 1、2* 和 *3*).来调用预设 *Preset U01、U02* 和 *U03*。

5. 外部继电器示例

本示例展示连接到 **IX** 或 **MXE 控制端口（GPIOs）**的**外部继电器**，如何用来**切换**比直接使用 GPIOs **更高的电流和电压**。

连线图

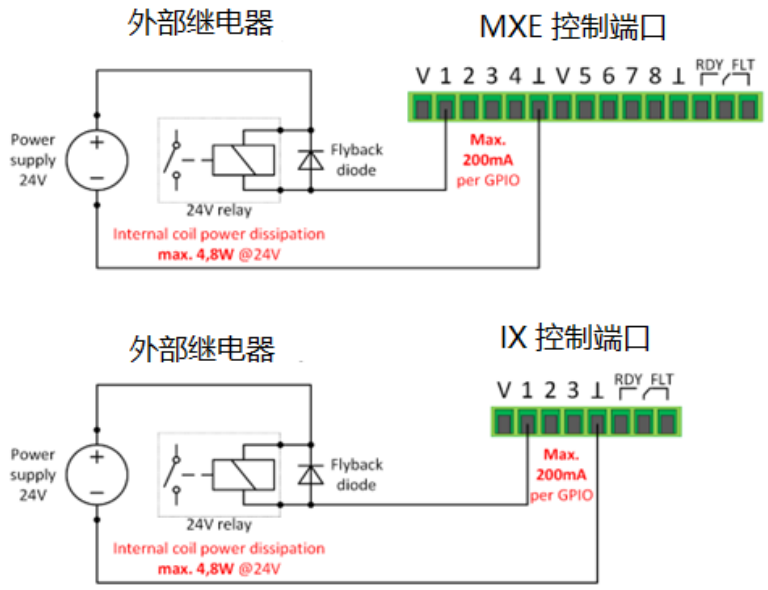


图 41：连接外部继电器到 MXE 或 IX 控制端口

GPIO 配置

确保 SONICUE 中 **Setup>GPIO** 下的 GPIO 应设置成正确的类型，在我们的示例中我们需要 ***GPIO1*** 配置为 ***Digital OUT（数字输出）。***

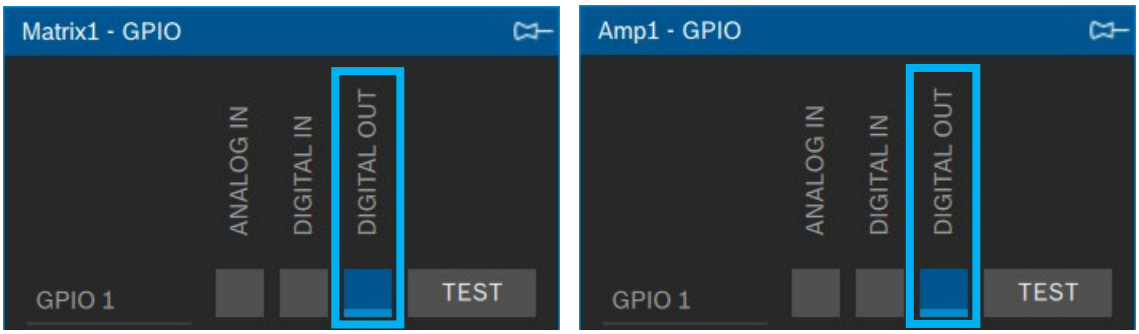


图 42 和 43： 设置 *Matrix1*（左）或 Amp 1（右）为 DIGITAL OUT。

TaskEngine 编程

接下来的 TaskEngine 架构中，IX 或 MXE 的 ***VLogic 1*** 用于激活 ***GPO 1*** 来控制 1 个外部继电器，IX 和 MXE 的 **virtual analog** 和 **logic 值**是连接 **SONICUE Control** 用户界面和 IX or MXE **TaskEngine** 的完美方法。

Virtual analog 和 **logic** 值还可以通过 **IX** 和 **MXE** 的 **http API** 进入。

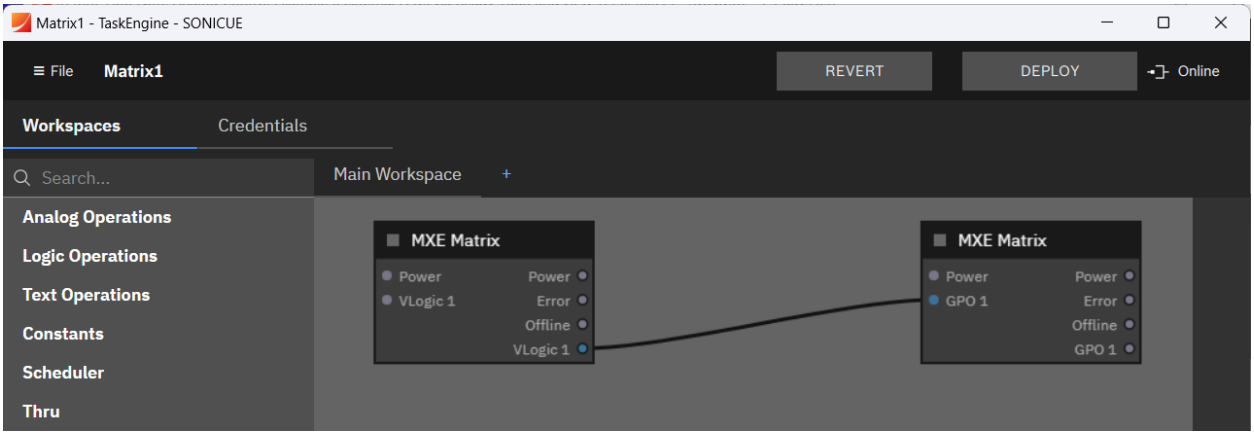


图 44： MXE TaskEngine 配置将外部继电器通过 *VLogic 1* 转换到与其连接的 *GPO 1*

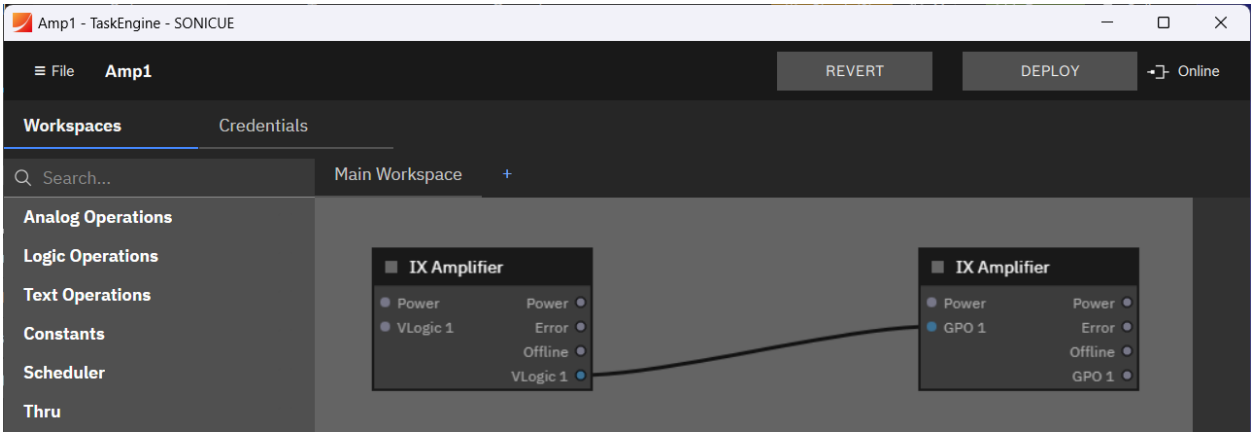


图 45： IX TaskEngine 配置将外部继电器通过 *VLogic 1* 转换到与其连接的 *GPO 1*

6. 外部 LED 示例

本示例展示于 **IX** 或 **MXE 控制端口（GPIOs** 或 **Ready/Fault** 触点**）**连接的**外部 LEDs**，如果用来作为**设备或系统状态**的信号指示。

连线图

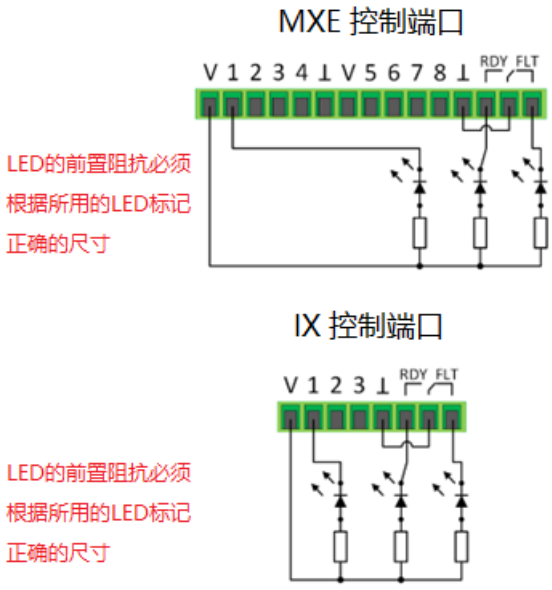


图 46: 连接外部 LED 到 MXE 或 IX 控制端口

GPIO 配置

确保 SONICUE 中 **Setup>GPIO** 下的 GPIO 应设置成正确的类型，在我们的示例中我们需要 ***GPIO1*** 配置为 ***Digital OUT（数字输出）。***

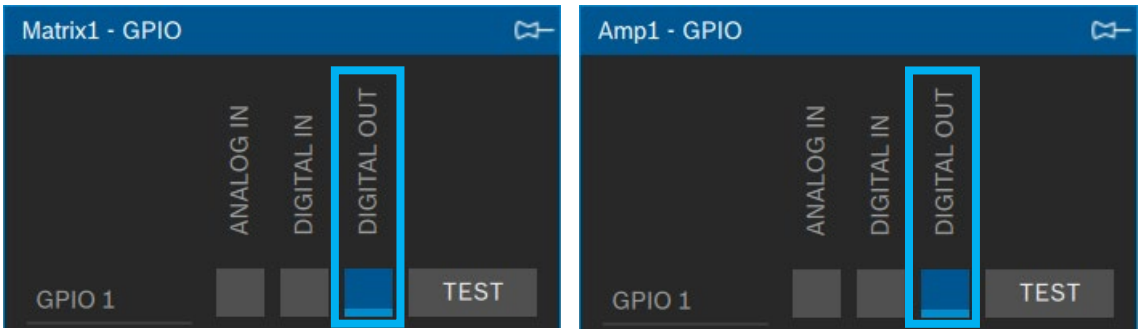


图 47 和 48: 设置 *Matrix1*（左）或 Amp 1（右）为 DIGITAL OUT。

Ready/Fault 配置

无电势 **Ready/Fault** 继电器触点的状态依赖位于 IX 或 MXE **Setup>Status** 下面 ***State Flags*** （状态标识）的配置，如果**设备**完整启动 =**ready**.

如果 ***Device Not Ready*** 标识，或任一其他 ***Collect column*** 里面选择作为 ***Collected Error State （错误状态收集）*** 的标识被激活，IX 或 MXE 前面板的故障 LED 将被激活显示故障，同时故障继电器触点闭合。

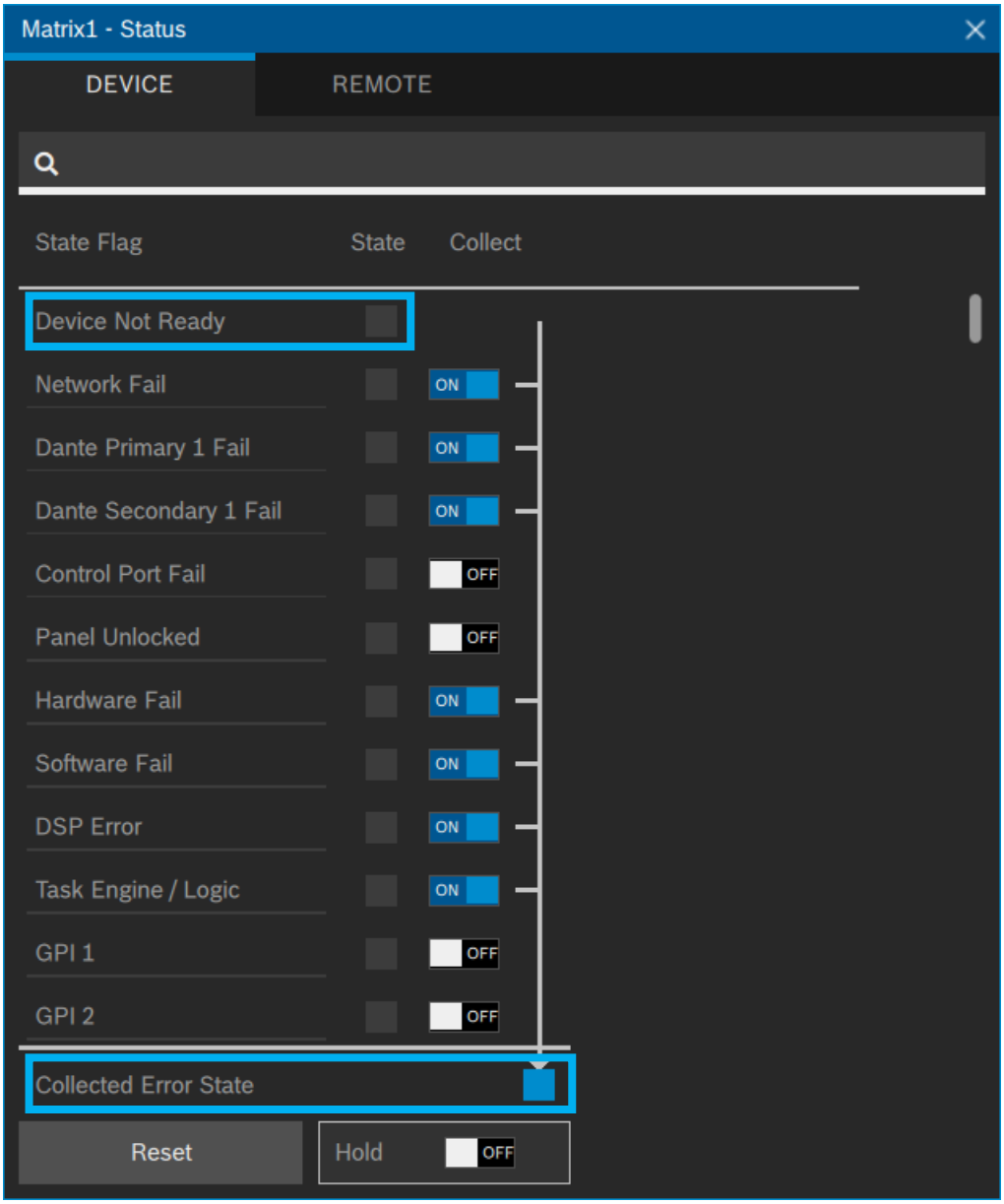


图 49: MXE 状态弹窗 – DEVICE 页

在同一个 ***Status*** （状态）弹窗 ***REMOTE*** （远程）页中, 位于同一网络上**其他设备**（MXE，IPX，IX）的**状态**都会被收集在 ***Collected Error State***. 这样， **MXE 故障继电器**不仅可以用作**设备故障，**还可以用作**系统故障**的信号指示。

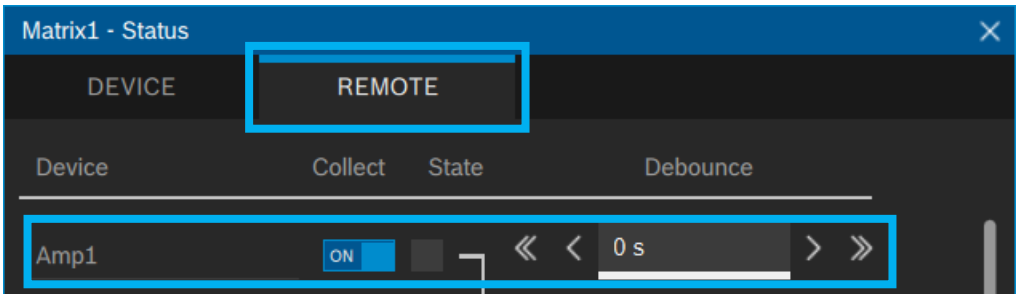


图 50： MXE *状态弹窗 – REMOTE* 页

TaskEngine 编程

如果除了设备或系统状态（通过 Ready/Fault）还有**其他信息**应该被信号指示，可以设置 1 个 **GPIO** 为 **DIGITAL OUT**。然后 TaskEngine 可以配置来触发与该 GPIO 相连的 LED。

接下来的示例，设备 ***Power*** 状态通过 ***GPO 1*** 报告。

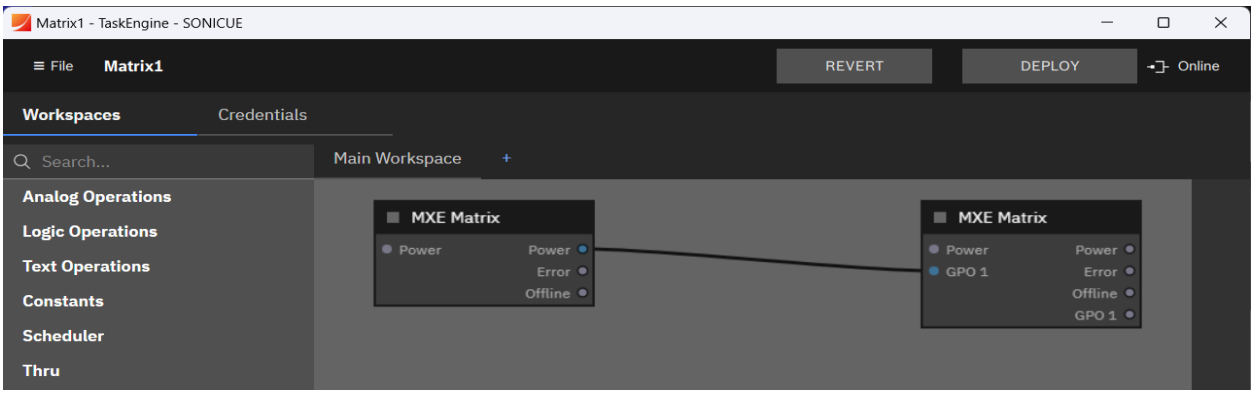


图 51： MXE TaskEngine 配置用来通过外部连接到 GPO1 的 LED 进行 *Power* 状态的信号指示。

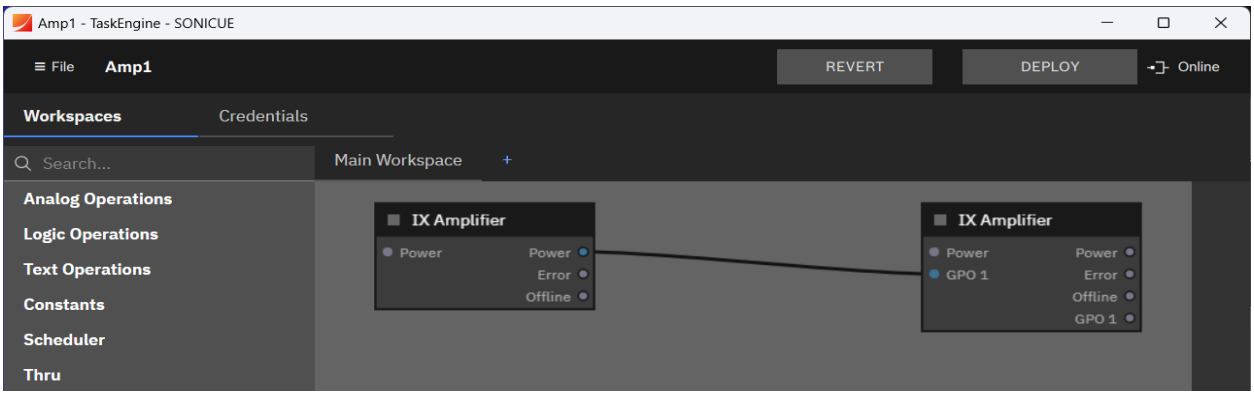


图 52： IX TaskEngine 配置用来通过外部连接到 GPO1 的 LED 进行 *Power* 状态的信号指示。

第三方产品免责声明：

Dynacord 不对标准电子元件（电位计、电阻器、继电器、LED 等）的保修、质量或可用性负责。本文档中包含的标准电子元件在发布时已成功通过测试。然而，Dynacord 无法保证未来此类标准电子元件的兼容性或可用性。