

总览

NION 信号处理功能由 NIONODE 负责，它是一个 2U 网络音频处理箱，带有四个用户定义的音频输入输出卡插槽。不同版本 NIONODE 可通用。

NION 控制和管理功能(没有信号处理功能)是由 CONTROLNODE 来负责，它是一个机柜电脑。注意，CONTROLNODE 不是构成系统所必须的，很多的控制和管理功能都可以由 NIONODE 来负责。

CAB 4N 是一个 2U 的 COBRANET 音频输入输出接口箱，带有四个厂家自定义音频卡。本文不对 CAB 4N 做说明。

NIONODE 硬件纵览

1. 信号处理：最多六个 21161N 芯片，每个可以在 400MFLOPs 下连续运行（最大 600MFLOPs）。
2. 音频扩展槽：每个 NIONODE 可以配置四个插入后面板的音频输入输出卡。种类包括：模拟 mic/线路 8 通道输入，模拟 8 通道输出，模拟 4*4 等，每个 NIONODE 可以扩展到最大 32 路本地输入/输出。扩展卡不支持热插拔。
3. COBRANET：标准 CM-1 芯片，每个 NIONODE 可提供 COBRANET 上 32 个输入通道和 32 个输出通道。使用 COBRANET，单独的单元或通过 XDAB 连接起来的单元串可以分布地处理几十成百甚至上千路高品质数字音频信号。NION 系统中不是必须使用 COBRANET 的。
4. XDAB：机柜空间内的 NIONODE 串可以通过 XDAB 环连接，可处理 512 个冗余延迟浮点音频。每个单元有两个 XDAB 口，标为 IN 和 OUT，用铠装 5 类线连接单元成物理环。一条 XDAB 线不得大于 4.5 米长。XDAB 的使用也不是必须的。
5. LAN：标准的 10/100 以太网 LAN 口用于从一个 WINDOWS 操作系统的 PC 机上对 NIONODE 进行配置，监控。也可用于独立的 NIONODE 网络内单元间无缝控制通信。一个工作 LAN 对于 NION 系统的某些操作来说是必须的。
6. 串口和 GPIO：两个串口，一个 RS232 和一个 RS485，用于外部控制。GPIO 口为报错，电平，开关，指示灯和编码器提供信号。
7. 前面板用户界面：NIONODE 前面板的用户界面包括一个单色位图显示器，一个数据轮，四个功能键，一个软电源开关一个警示键/灯和不同的状态 LED。这个界面用来做基本的配置任务，状态和错误指示，音频表，也可以被编程用于音频控制。

软件纵览

NWARE 是一个图形界面的应用软件，运行在标准的 WINDOWS PC 机上（XP 或 2000）。它是用来设计，部署和控制包括 NIONODE，CONTROLNODE 和 CAB 4N 的系统。一旦系统被配置和调整完后，NWARE 电脑就不再是系统操作所必须的了。

一个单独用于控制功能的 NWARE_CONTROL 可以使用户操作定义好的用户界面。

NIONODE 本身运行在强大的 LINUX 系统下。NIONODE 还运行 PYTHON 脚本，用于控制编程。

CONTROLNODE 运行 WINDOWS 操作系统，将来会使用 LINUX。绝大多数 NIONODE 中的控制设备和脚本在 CONTROLNODE 中也存在。另外 CONTROLNODE 支持 SNMP 管理协议和其它一些 NIONODE 不支持的控制功能。

设计

系统的设计从设计者创建一个 PROJECT 并输入图形化的输入输出和信号处理单元开始。设计者还可以创建自定义控制面板和自定义的交互控制脚本。

每个列入工程中的目标 NIONODE 和 CONTROLNODE 将被给予一个独一无二的角色名称。这个角色名称在随后的部署过程中配合物理上的目标 NIONODE 单元。

部署和仿真

当一个工程被部署，设计检查也完成了，DSP 代码已产生，接下来就是下载到目标 NIONODE 中，并且 PC 主机连接到目标 NIONODE 准备进行控制。设计检查的步骤将生成一个报告，显示硬件的使用情况以及设计中碰到的问题。

这个过程需要 PC 主机与需要安装工程的目标 NIONODE 连网进行。在部署的过程中，附属这个网络的 NODE 将被发现并显示给设计者。设计者将接着给所需要的 NIONODE 指派包含工程内容的角色。角色指派使得每个 NIONODE 可以下载并在本地存储配置数据，以便在运行时扮演好自己的角色。

如果需要的话，除了部署，可以在没有目标硬件的情况下，实现有限的 PC 仿真，来实现控制、预设的保存和加载，控制脚本练习甚至外部控制等功能。（注意：基于 PC 的 NIONODE 仿真并不能仿真 NIONODE 对音频

信号的处理。)

控制

一旦一个工程被部署完成，工程所设置的 NODE 就执行设计中的信号和控制处理功能。一个或者多个运行 NWARE-DESIGN 或者 NWARE-KIOSK 的 PC 机可以连接到目标 NODE 上来控制和监视它们。第三方的控制系统也可以配合。

NWARE 软件用户指南

这部分的内容包括 NWARE 软件和 NIONODE，CONTROLNODE 硬件的详细使用资料

NWARE 安装

NWARE 通过一个 WINDOWS 安装包进行安装。运行安装向导，将复制 NWARE 的程序数据到你的硬盘。安装文件可以在 MM 的网页上下载到。

最小运行要求

WINDOWS 2000 (带有 SP3 或更高的升级包)

WINDOWS XP HOME 或 PROFESSIONAL (带有 SP1 包或更高)

WINDOWS 2003 SERVER (如果非要这么做的话)

其他的版本的 WINDOWS 不被支持或推荐使用。除非是运行了更先进的东西。为了从软件上提供最先进的功能，我们被强制使用最新版的操作系统。事实上，为了在 WINDOWS 2000 上使用 NWARE，我们不得不安装额外的驱动。XP 的系统被认为是最恰当的配合 NWARE 使用的操作系统。

安装故障排除

如果出于某些原因，你的安装失败了，首先确保已经完成了下载。我们在浏览器上的下载列表右边标明了文件的大小。如果文件没有被破坏，那安装失败的最大可能就是这个。如果重复出现此问题，先删除你的 INTERNET 临时文件后重新下载。

注意对于测试版本，你必须首先完全卸载之前的版本才可以安装新版本。我们将在随后发布相关补丁-或许在 1.0 版（不是保证）。

快速开始--创建和运行一个工程

本章提供一个简短的创建和运行一个工程的步骤的纵览，以方便那些太兴奋而不能阅读详细的文档的人，或者那些受 MTV 教育的看不懂详细文档的人？

快速开始--创建一个工程

运行 NWARE-DESIGN 程序，建立一个空白的工程，带有默认的名称，准备开始编辑。

工程设计创建，一般来说，就是从设备库中拖动设备到设计框的当前页面，然后把设备节点连线起来。

某些设备体现目标系统的硬件原理，有些设备体现基于 DSP 的信号处理算法，还有一些体现控制功能。

设备在编辑模式下被放置，在连线模式下被连接。

NIONODE 设备

NIONODE 设备体现一个单独的 NIONODE 的硬件组成。通过 NIONODE 的设备属性，可以看到角色名称，NIONODE 模型，是否使用 COBRANET CM-1，GPIO 口的配置，装在扩展槽上的卡的类型和一些其他的内容。

在 NIONODE 设备窗口里，是带标签的控制页面。包含通过设备属性配置的 NIONODE 组件的控制功能。例如，如果扩展槽 1 口被配置成一个模拟音频输入卡的话，这个卡的控制和表头将会在控制页面里头标题为扩展槽 1 口的页面上。

对应音频输入输出卡和 COBRANET 包的 FLYOFF(音频连线节点) 将会在功能框中的 FLYOFF 页面中找到。FLYOFF 被拖拽到设计页面，然后连线到信号处理设备。

快速开始部署一个工程（有硬件）

当文件--部署命令被调用（快捷键 F9），设计检查将会执行，DSP 代码也会生成，角色将会指派给所发现的 NIONODE，文件被下载到该 NIONODE，PC 主机已连接，准备控制。

在下载硬件之前，角色部署窗口将提示你将工程中的角色指派给网络中发现的 NIONODE。

一份关于设计检查的报告将生成，并且显示在输出框中。

快速开始仿真一个工程（没有硬件）

当你没有目标硬件（如 NIONODE，CAB 4N 等）时，或者不想弄乱当前运行的系统的时候，目标硬件可以在装有 NWARE 的 PC 机上被仿真。这使得手形控制，存储和调用预设，联系控制编程和检验外部控制等成为可能。这些属于目标硬件的运行功能，因此通常不在 PC 上运行。注意音频处理不能够被仿真。

要仿真，通过文件-仿真命令进行（快捷键 Ctrl+F9）。

NWARE 详细说明

本章细讲 NWARE 的用户界面概念。

用户界面窗口和它的组成部分

NWARE 的用户界面由一个菜单栏，一个按键工具栏，三个显示框和一个状态栏组成。

菜单栏和大多数 WINDOWS 应用程序一样，给出命令菜单。能够带出对话框的命令后面跟着三个。有键盘快捷键的命令会把快捷方式列在命令后头。

图标工具栏包括通往菜单命令的图标快捷方式，鼠标停留在图标上会显示图标所调用的命令。

右上的框是设计框，包括了工程系统设计。设计框默认的是打开一个页面，但是它可以包含很多的，由用户命名的带标签的页面。

低一点的框是输出框，包含了输出页面：报告（设计编译结果），日志（信息日志），远程日志（从 NIONODE 获得的日志数据）和远程状态（一个树形查看网络中所有的 NIONODE 的深入状态）

左上的框，功能框，包括了功能页面的设置：设备（显示设备库），FLYOFF（系统和用户创建的音频节点），脚本（用在控制脚本管理），媒体（用于媒体文件管理）和检察员（纠错级的设备检查员允许或不允许终止用户）。

功能框和输出框可以被关闭和开启，通过点分隔条上的图形柄或者通过视图菜单中的命令。

在 NWARE 窗口最下面有个状态栏，显示关于设计编辑过程和运行连接的信息。

用户参数选择

在工具--用户参数选择的菜单命令显示一个对话框，包含用户界面的全局设置。

工程

用于安装的 NION 系统设计包含在一个工程里。工程在磁盘上作为一个单独的文件存储，以 npa 为扩展名，但是实际上，工程文件是一个压缩存档，它包含一个目录和文件的树。这个树包括设备和控制布局和连线，控制预设，每个 NIONODE 和 CONTROLNODE 的配置数据和任意导入此工程的媒体文件（例如一个图片或一个波形文件）等。其中最重要的文件是 DATA.PDN，存在于文档的根目录，它包含了设备和控制的布局和连线。你不需要对工程文件的详细信息有所涉及。

工程的管理包括文件菜单下的如下命令：新建，打开，保存，另存为和属性。

工程属性对话框，可以通过文件--属性来访问，它包含影响当前工程的设置。

打开多个工程会导致多个用户界面（ie）被打开，每个 NWARE 用户界面处理一个单独的工程。你可以通过窗口菜单下的命令来切换其他开启的工程。

工程范围

一个独立的工程可以包含任意数量的 NIONODE，不管音频连接，只要这个节点是在同样的以太网控制网络里。这意味着一个工程可以包含由 XDAB 连接的多个 NIONODE 串，由 COBRANET 连接的 NIONODE，或者没有音频连接的 NIONODE。另外，任意数量的工程可以在相同的以太网内共存，并且 COBRANET 音频信号可以人工地配置在不同的工程中相互连接。但是，一个 XDAB 环，必须完全包含在一个单独的工程中。

工程属性

需要添加。

编辑，连线，手形和上色模式。

有四种模式，其中总有一种模式是有效的。它影响用户界面中的设计框：编辑，连线，手形和上色。模式可以通过模式菜单命令切换，也可以通过键盘快捷方式或者右键关联菜单。

编辑模式 允许对页面上的设备的编辑。

连线模式 允许在线路节点间创建连线。

手形模式 运行时间内支持旋转旋钮等。

上色模式 用调色板上的颜色来修改设计中的物体颜色。

分级块，页面和层

使用分级块和页面，可以降低设计的复杂性。

分级块

分级块是可以包含设计元件的设计元件，如设备，控制，图形和其他分级块等。分级块的窗口可以通过鼠标双击进入它所包含的一个或多个页，这里可以粘贴其他设计元件。

一个分级块可以有音频输入输出的连线节点，这是通过它的设备属性中的输入数量和输出数量来进行配置的。连线节点同时出现在这个分级块的外部窗口内示意页面上，并且允许以绘画方式连线块内外。

某些的内建设备，和全部的用户定义设备，都是由分级块来创建的。

页面

设计框，以及分级块或者设备的窗口，都包含着一个或多个标签页面。如果页面标签是可用的，点击它的标签就可以让它显示出来，标签在页底突出的部分。页面也可以通过按组合键 `ctrl+tab` 循环地出现。

双击标签输入新名称，可以给标签改名。

通过拖拽标签往左或往右，可以重新排列标签顺序。

要插入新的页面，可以通过在页面标签区的窗口框底部右键单击选择添加，也可以通过在已有标签上右击选择插入。

一个页面可以被隐藏起来，通过在页面标签上点右键并设置成隐藏。但是至少要保证一个页面是显示的。在任意页面标签上或者标签区的空白处右击，选择显示对应需要显示的隐藏页面名称，即可取消该页的隐藏状态。

对于设备和分级块，页面标签本身也可以隐藏起来，或者显示出来。通过在其窗口标题条（或者窗口框顶部）右击鼠标，然后选择隐藏标签或显示标签。当标签被隐藏的时候，当前显示的页面将会填满窗口框，就好象没有其他页面存在一样，有时候这样挺好看的。用 `ctrl+Q` 也可以执行隐藏或显示标签的命令。比方说你开了个窗口没有标签，如果你怀疑标签是被隐藏了的，那就点一下 `CTRL+Q` 试试。

每个页面也还可以被指派到上边或者下边，也就是说可以在一个窗口里头分出框框。只要在某个标签上右击选择移动上边或者移动到下边就可以了。这里允许分出两个框框，或许控制方面在上头，示意图在下头。

每个页面都有可以访问的属性，通过在页面标签上右击鼠标选择属性。页面属性对话框允许你来改变页面背景颜色和风格。也可以选个图片做背景。例如什么度假胜地西部草原之类的。

信号可以通过连线 `FLYOFF` 的方式来在块和块之间传递。（后面会详细讨论这个）

层

这里讨论层

设备

对于一个工程的组成，最重要的应该就是设备。设备，是某些东西的一个象征，例如，象征一个硬件（如 NIONODE）或者一个基于 DSP 的音频算法（例如 NIONODE 的低通滤波器）。其他设备类型还包括控制处理和预设。分级块设备是其他设备的容器。

设备属性

许多设备都有设备属性，这是设计者可指定的值，例如混音输入数量输出数量。设备属性在设计和编译时起作用，也就是说，当程序运行的时候设备属性是不可以进行调整和改变的。要访问一个设备的设备属性，可以在选中当前设备的情况下选择工具菜单中的设备属性命令，也可以在设备上右击鼠标选择设备属性。

某些设备的属性会影响到设备本身输入输出节点和/或控制部分的数量。在这种情况下，改变属性会导致自动地生成正确的节点设置和控制面板。例如，你可以把一个 8*2 的混音台变成一个 9*3 的（或者其他你想要的）（注：这里可能是错误，因为输入通道不能是奇数---译者），那么输入输出节点的数量会自动更正，而且控制面板也会自动调整。

很多设备还有高级属性，这些高级属性是为了允许一些高级用户来控制更多的技术方面（这种控制更缺少人性化），或者是允许用户来检查某些编译中自动进行的分配。

自定义设备和预配置设备

在设备库中，某些设备类型有一个自定义选项，例如混音台。当你把这类设备拖到设计框的时候，这些设备会有一个设备属性对话框来提示你来定义它。其他一些不能自定义的设备都是为了方便应用而对自定义设备进行过简单的预设置产生的，这些设备在放进设计中后，都可以通过设备属性来编辑它们。

用户设备

你可以创建你自己的设备，并把它们放进设备库中去。选择一个设备或者一个分级块，然后选择工具--定义用户设备的菜单命令。你会被提示要输入一个唯一的设备名称，然后你的设备将会出现在设备库中的用户创建设备文件夹里头。

组合设备

组合设备包括其他的设备，但是可能依然有设备属性，来允许你配置它的内部。通常来说，你不知道哪个

设备是组合型的。例如，在设计中放入一个分频器，双击打开它的窗口，按快捷键 CTRL+Q 来显示它的隐藏标签页面，然后点击示意图页面标签，你会看到分频器是由低通滤波器和高通滤波器组成的。

连线

连线是设计过程中的一种处理方式，它可以以图形化的方式连接设备和控制的线路节点。用户参数选择对话框可设置对一些设计框中的线路外观的控制。每个线路节点都有一个信号类型，来决定什么样的连线可以作用于它。例如，你不可以把一个设备音频节点和一个控制节点连线起来，另外，一个设备的音频输出可以做扇形的扩散输出连接，但是音频输入节点就不可以。

FLYOFF

FLYOFF 是连线的终端标签，它允许信号穿过不同空间进行连接（可以在一个页面内，在不同的分级块之间，或者在不同页面间），它的连接是不需要有图形化的连线的。换句话说，连往相同名称的 FLYOFF 的节点是逻辑上（而不是图形上）的连接。FLYOFF 对设备和控制连线都适用。

FLYOFF 的创建是在连线模式下，从一个线路节点左击鼠标并拖拽出来后再点鼠标右键即可。

注意：FLYOFF 可以做出错误逻辑的连线，例如将两个音频输出口连接起来。当一个 FLYOFF 出错的时候，连到它上面的线会变成红色；当一个 FLYOFF 处在未完成状态，例如一个音频输出的 FLYOFF 没有连接到另一个音频输入时，它上面的连线会变成绿色。除此之外，它是黑色的。

命名 FLYOFF

在连线模式下，选中一个 FLYOFF 后输入文字，或者在 FLYOFF 上双击鼠标，可以打开一个列表式的对话框，显示已经存在的所有 FLYOFF 的名称。你既可以输入一个新的名称，也可以选择一个已经存在的名称。

你还可以自动命名一组 FLYOFF。要想这么做的话，首先你要人工地命名这一组开始的两个 FLYOFF，然后选中整个组，选择工具--自动命名，或者点自动命名的图标。所有这个组内的 FLYOFF 会被自动地以前两个名字间的关系做为基础而命名。例如，将前两个命名为 BUSS SEND 1 和 BUSS SEND 2，那么接下来的 FLYOFF 就会自动被命名为 BUSS SEND 3 等等。（注：对此处的验证发现，软件中的命名方式为表达式标签，也就是通过表达式来命名，没有所谓自动命名这个命令了，但是系统仍然能够识别前两个命名之间的逻辑关系，

所以只要输入两个名称后选择工具--表达式标签，然后弹出对话框中输入框内的文字其实就是系统判断出的逻辑表达式，然后直接点 OK 的话，也可以完成自动命名的功能----译者)

控制

大多数设备有控制部件，这是可以由用户调整的值，例如一个混音台的输入增益，输出增益和输入静音。设备控制部件可以在系统运行时进行调整。一个设备控制页面可以通过双击这个设备来开启窗口。只有在手形模式下，而且是在连接到硬件或仿真状态时，控制部件才可以被操作。

控制手势

一旦一个工程被编译部署（既可以是硬件的也可以是仿真的），就可以通过手形来操纵控制部件了。

详细手形

关于手形控制旋钮推子按钮等的详细内容，[在这里](#)（原文有超链接，但是点不出东西）。

文字录入

关于文字输入和显示的详细内容，[在这里](#)（原文有超链接，但是点不出东西）。

图形化控制编组

在手形模式下，临时性的控制部件编组可以通过同时选择多个控制部件来实现（控制部件的选择就跟标准的 WINDOWS 下的选择一样，无非是用 CTRL 或者 SHIFT 或者是拖拽框）。一旦编组，控制部件就可以在手性模式下统一动作。而且它们之间的相对值会被保留。这样的编组是暂时性的，一旦一个编组外的控制部件被选择，这个编组就会消失。

注意：图形化控制编组是 NWARE 用户界面的独有功能，它不会影响目标硬件，而且跟那种硬件内的永久性编组也没关系。

控制部件属性

控制部件有风格，标志和连线属性。一个控制部件属性可以通过选择当前控制部件后在工具--控制部件属性命令，或者对控制部件右击鼠标选择控制部件属性来访问。

风格

这是设置控制部件的外观和动作方式。

标志

这个标签是让你配置或编辑内建的关联控制部件的标志功能。

连线

控制节点（似乎现在叫 CONTROL WIRING）

这个选择框给当前选择的控制部件添加可连线的控制节点。

主节点可以连线到若干个隶属节点。主节点可以控制隶属节点，但是隶属节点不可以控制主节点。当一个主节点连线到特定的控制设备（即控制脚本编辑器，快照），这些设备可以发送控制值返回到该控制部件。这是双向的主控连线。隶属控制部件永远不会发送它们的设置到主控制部件的。

同级节点可以连接多个同级节点。有一个快捷方式来做这个连接，选择所有需要同级连接的控制部件，右击鼠标选择控制编组，同级的节点就会被自动添加上同级控制节点，而且所有节点会连到一起。

需要更多相关信息，请参阅后面的控制部件连线和编组标题内容。

控制别名

这是随意给控制部件输入任意外部控制名称。如果一个控制部件有这么一个名称，那就使通过串口或者 TCP/IP 协议的外部控制成为可能。

更多资料请参阅 TCP/IP 和串口外部控制。

自定义控制面板

控制部件可以被复制和粘贴到源设备控制页面外的位置。简单举一例，这就使创建一个混合了设计中所有的控制部件的用户自定义的面板成为可能。可以任意修改位置，风格，颜色和标签，还有背景图片，这样就可以创建一个更直观友好的用户界面，做系统设置或最终用户控制。

控制部件编程

控制部件编程可以在控制部件间创建自定义交互。记住这些交互不是在图形化用户界面上发生的，而是 NIONODE 的配置的一部分。这挺好的，因为这意味着它们不必通过 NWARE 电脑来工作，而且可以受控于第三方控制系统。

控制部件设置

控制部件的设置是包括如下三个方式：通过位置，字符串和值。位置是对应于用户界面中的图形化旋钮或推子位置的小数，范围是从 0-1。字符串是控制部件在用户界面中的显示文字串。值是控制部件的 32 位内部数，它虽然对用户界面不大有用，但是常用于控制部件编程。例如，一个增益设置位置可能是 0.55，它的字符串是 -45dB，它的值是 -45.0。

控制连线 and 编组

一个控制部件可以图形化地连线到另一个来形成控制编组，一个控制部件的连线属性可以通过它的连线属性对话框中的连线标签来访问。在控制连线对话框中，你可以独立地允许显示对这个控制部件有用的三种控制连线节点：隶属，平级和主控。这些节点从左到右地显示在控制部件的下面。

下面是线路编组的类型关系。

*主控到隶属---一个控制部件的主控节点连线到一个或多个控制设备的隶属节点。隶属的控制部件会遵循主控，而且变成只读性质，这意味着它或它们不可以被直接用手形控制。

*平级到平级---一个控制部件的平级节点连线到一个或多个控制设备的平级节点。在这种情况下，任何一个控制设备可以受手形控制，而且它们互相遵循。

*隶属到隶属---将一个控制部件的隶属节点连线到另一个设备的隶属节点可以做到，但是没有任何意义，除非另外一个主控节点可以连入这个设置。

控制部件直接连线到另一个部件，这是一种位置链接，也就是说，图形位置，或上或下，链接上的控制部件都会遵循。

这里有一种在两个或者更多个控制部件间形成平级到平级链接的快捷方式。先选择所需要的控制部件，然后或者通过工具--控制编组菜单命令，或者通过右键单击选择控制编组。

控制处理

要创建控制部件间的交互作用，一个更有力的是利用控制部件处理器。在这种情况下，控制部件将连线到控制处理器设备，这是一个控制设备的主控节点连线到控制处理器，另外一个或多个控制设备隶属节点连线到控制处理器的输出节点。控制处理提供链接，并且处理控制设置的位置，字符串和值。（直接的控制部件到控制部件的连线只能提供基于位置的链接）
有两种类型的控制处理器：控制操作器和控制脚本编辑器。

控制操作器设备

控制操作器可以执行大量内建的控制处理操作，这在它的设备属性中可以选择。一些可能的操作包括：位置链接，字符串链接，值链接，最大值，平均值，位置比较，逻辑或，还有很多其他的。
注意：下面的部分大概该挪作设备参考部分的内容。

操作器类型	输入数量	输出数量	设备描述
翻转位置	1	1	输出位置与输入位置相反。由于位置的范围是 0 到 1，所以输出位置就是由 1 减去输入位置。
负面值	1	1	输出值是输入值的负数。注意输出的控制值可能会被钳位，这是为了保证输出范围保持在一定幅度内。
绝对值	1	1	输出值是输入值的绝对值。对于输出控制值的钳位也可能发生。
拐点	1	1	输入位置大于 0.5，输出为 1，否则输出为 0。
滞后	1	1	输出位置只会是 0 或 1，这取决于输入位置的拐点延迟。当输入位置下降到低于 0.3 时，输出从高变低；而当输入位置上升到高于 0.7 时，输出位置从低变高。
上升沿触发	1	1	使用拐点和滞后，输出位置在输入位置的每个上升沿都会触发输出的 0、1 翻转。
下降沿触发	1	1	使用拐点和滞后，输出位置在输入位置的每个下

降沿都会出发输出的 0、1 翻转。

值求和	2	1	输出值是所有输入值的和。输出值无法达到时会发生钳位。
值求积	2	1	输出值是所有输入值的积。有可能发生钳位。
平均值	2	1	输出值是所有输入值的平均值。
最小值	2	1	输出值是所有输入值中的最小值。
最大值	2	1	输出值是所有输入值中的最大值。
值比较	2	1	如果第一个输入值比第二个输入值大或相等，输出为 1，否则为 0。
值链接	1	1	输出值等于输入值。
位置求和	2	1	输出位置是所有输入位置的和。如果和超过 1，输出将被钳位到 1。
位置求积	2	1	输出位置是所有输入位置的积。
平均位置	2	1	输出位置是所有输入位置的平均。
最小位置	2	1	输出位置是所有输入位置中最小的一个。
最大位置	2	1	输出位置是所有输入位置中最大的一个。
位置比较	2	1	如果第一个输入位置大于或者等于第二个输入位置，输出位置为 1，否则为 0。
位置链接	1	1	输出位置等于输入位置。

对于下面六个逻辑操作器来说，输入都被处理成为布尔值。如果输入位置大于 0.5，布尔值为真，否则为假。

操作器	输入数量	输出数量	描述
逻辑 与	2	1	输出位置是对所有输入位置的布尔值做与运算的结果。

逻辑 与非	2	1	输出位置是对所有输入位置的布尔值做与非运算的结果。
逻辑 或	2	1	输出位置是对所有输入位置的布尔值做或运算的结果。
逻辑 或非	2	1	输出位置是对所有输入位置的布尔值做或非运算的结果。
逻辑 异或	2	1	输出位置是对所有输入位置的布尔值做异或运算的结果。如果输入数量多于 2 个，异或运算的定义为当奇数个输入为真时结果为真，否则为假。
逻辑 异或非	2	1	输出位置是对所有输入位置的布尔值做异或非运算的结果。也就是异或运算结果的补数。
字符串链接	1	1	输出字符串复制于输入字符串。
复用器	2-31	1	输入位置与 0.5 做比较后得出布尔值。这些布尔值将被作为一个二进制数的位。第一个输入做最低位。输出值是这个二进制数的值。
分离器	1	2-31	输入值会被作为一个二进制数。每个输出位置对应二进制数的一个位，输出一个 0 或 1 的位置。第一个输出是最低位。这是复用器的精确补体。

控制脚本编辑器设备

控制脚本编辑器涉及到 PYTHON 脚本功能，提供设备的完全可定制的控制处理。脚本功能在功能框的脚本页面创建。通过控制脚本编辑器的设备属性可以将功能指派给控制脚本编辑器设备。

参见 PYTHON 控制脚本参考以获得更多关于创建 PYTHON 脚本功能的信息。

通用控制部件

通用控制部件设备提供控制部件的设置，没有加入任何特殊功能，用于控制编程。通用控制部件的类型和

范围可以在设备属性页面中选择，风格也可以在每个控制部件的控制属性页面中选择。

NIONODE 前面板控制设备

任何一个控制部件都可以在通过控制连线到一个 NIONODE 前面板控制设备时，由 NIONODE 的硬件前面板来进行访问。每个 NIONODE 前面板控制设备都需要参考硬件角色名称的特性，指派到一个特定的 NIONODE 硬件。而且每个部件都需要赋予一个不同的标志特性来区分它们。这个标志就是显示在 NIONODE 硬件前面板屏幕上的文字。

前面板控制设备的唯一连线节点如果连线到一个隶属节点，则部件对隶属部件作为主控；如果连线到一个主控节点，则作为一个平级控制。

查找

编辑--查找命令用于定位设计中的元件。

[更多内容参见这里。](#)

预设

预设部件是存储和恢复控制部件状态设置数据的设备。每个预设部件可以存储和加载多个数据设置，具体数量可以在设备属性中的数据数量中进行配置。设备属性中的预设名称是为了区分包含多个预设部件的设计中的每个预设设备。如果预设设备都用相同的预设名，那么它们就相当于是一个设备，加载和存储同样的数据。

预设设备有两中类型：全局预设设备和子预设设备。

全局预设设备

全局预设设备存储和重建一个工程中的所有控制部件，无论它们在设计中被放在哪个层中。由于全局预设设备可以有完全满足需要的数据数量，所以一个工程中无须使用多个设备，而且如果使用多个可能会使程序发生混乱。

子预设设备

子预设设备存储和加载一个分级层树下的所有控制部件。例如，如果一个子预设设备被放在一个包含大量设备和另外一个分级块的分级块中，这个预设设备就会存储和加载这个块中的所有设备的控制部件，以及这个块所包含的块中所有设备的控制部件，以及沿树下行的所有类似情况。

默认数据

在一个名称为 global 的全局预设设备中，标志为 DEF 的数据，意味着是一个工程的默认设置。这些设置在后台会被周期性地存储，而且在重启后会被重建。这就意味着在程序运行过程中，无论通过 NWARE ,NIONODE 前面板还是外部控制的方式对控制部件的改变，都会被永久保留。另外，只要工程中存在一个全局预设设备，你就可以选择随时重建默认设置。

安全状态 0

安全状态 0 是为了给 NION 一个可接受的安全初始值。

导出页面

页面可以被导出后用于 KIOSK 模式的用户图形界面。要打开导出页面的对话框，可以在页面标签上右击选择导出页面。这个对话框包括一个允许使用导出页面的勾选框，和一个用来确定要导出的页面是属于哪个编组的编组名称区。编组名称区是一个组合框，这样就可以很方便地在已经定义的编组名称中进行选择，或者也可以自己输入一个新的名称。

在菜单的查找命令中，有一个查找类型叫做查找导出页面，通过指定编组名称来实现。编组名称可以包含一个*，一个空白名称或者一个*会查找到所有的导出页面。

KIOSK 模式

NWARE : KIOSK 或者仅仅 KIOSK，是仅用于控制的图形化用户界面。当 NWARE 程序运行如下命令行变量：
"c:\pgogram files\mediamatrix\nware\nware.exe"/personality=nware_kiosk 时，即可打开它。

你也可以添加下列可选变量到目标字符串。

/open=<文件名>

/user=<用户名>

/pass=<密码>

注意:上面用到的文件名是指以 npa 作为扩展名的工程文件名。不需要指定文件路径是因为 NWARE : KIOSK 可以搜索到此名称的可用文件。以上参数可以随意组合。没有要求必须用到它们中的一个或全部。唯一的要求是必须使用/personality 变量。

部署保护

NIONODE 可以被设置为需要一个用户名和密码来部署工程。部署对话框会要求输入用户名和密码,否则将不能部署工程。

要设置用户名和密码,必须使用 PANDEBUB 程序连接 NION 主机。

如果你没有运行 NWARE,那 PANDA 搜索程序就不会运行,你必须手动地运行它。

当你运行 PANDEBUB 时,会出现一个带有命令提示的窗口

```
pandebub built 10:03:07 Sep  1 2004
```

```
type 'exit' to shutdown or 'help' for commands
```

```
[local]
```

输入 discover 可以看到连接到的网络上存在的设备的列表

```
[local]discover
```

```
1      srs_controlnode/conman
```

```
2      srs_controlnode/pandad
```

```
3      sd106_1/mcp
```

```
4      sd106_1/piond
```

```
5      sd106_1/psud
```

- 6 sd106_2/mcp
- 7 sd106_2/piond
- 8 sd106_2/psud
- 9 sd106_3/mcp
- 10 sd106_3/piond
- 11 sd106_3/psud
- 12 sd106_4/mcp
- 13 sd106_4/piond
- 14 sd106_4/psud
- 15 sd106_5/mcp
- 16 sd106_5/piond
- 17 sd106_5/psud
- 18 sd106_spare/mcp
- 19 sd106_spare/piond
- 20 sd106_spare/psud
- 21 sd106_controlnode/conman
- 22 sd106_controlnode/pandad
- 23 sd106_spare_controlnode/conman
- 24 sd106_spare_controlnode/pandad
- 25 derek_bottom/mcp
- 26 derek_bottom/piond
- 27 derek_bottom/psud

找到你想连接到的单元，以及后面的你想用来连接它的程序。这里用的是 PIOND。输入 conect 后面跟着你想连接的单元的序号。

```
[local]connect 26
```

```
connecting...
```

```
piond>
```

如今，你开始和 NION 内部的 PIOND 程序联动。输入 SECURITY 进入安全菜单。

```
piond>security
```

```
piond/security>
```

你可以随时输入 HELP 来显示可用命令菜单

```
piond/security>help
```

```
//                comment - line is ignored

comment           comment - line is ignored

exit              exit this menu

fp_lock           lock the front panel

fp_password_check <user> <p1> <p2> <p3> <p4>

fp_password_set   <user> <p1> <p2> <p3> <p4>

fp_unlock         unlock the front panel

help              list available commands

ignore            <command> ignore errors returned by command

password_check    <user> <password>

password_set      <user> <new_password>
```

repeat <count> <command> execute command count times

require_logon <1/0>

user_add <user_name>

user_delete <user_name>

users_list lists all users

piond/security>

要登陆网络来部署一个角色到对应单元，输入：

piond/security>require_logon 1

设置角色登陆要求到 1。

现在，任何人如果进行登陆部署，就会看到一个登陆对话框。除非你创建了一个用户帐号并且授予它一个密码，否则将完全不能对这个机器进行部署。所以，接下来，让我们创建一个用户帐号，并给它设置一个密码。使用 users_list 命令来查看已经存在的用户帐号。

piond/security>users_list

default

derek

piond/security>user_add booger

添加用户 booger

piond/security>password_set booger yummy

piond/security>password_check booger gross

失败

piond/security>password_check booger yummy

成功

```
piond/security>
```

然后，下一次当你尝试部署时，就会发生这种情况。

（有个对话框，要求输入用户名和密码，还有一个勾选框是记住密码）

记住密码勾选框不是当前的功能。

前面板保护

NIONODE 可以被配置成要求使用数字密码来登入前面板。如果没有使用密码，那么锁定功能是不可用的。如果需要密码，那么锁定功能可用。如果锁定可用时，那么前面板可以处在锁定或者未锁定状态。当处在锁定状态下时，前面板只会显示需要用户输入密码的登陆界面。一旦输入了密码，所有前面板的功能才可用。 *还没实现：如果用户持续五分钟没有对前面板操作，用户就会被注销，前面板再次自锁。

前面板保护功能可以通过前面板本身进行配置，也可以通过 PANDEBUG 程序进行。

要在前面板上改变前面板密码，选择配置图标，然后一直按下一页，直到出现密码页面。输入旧密码和新密码，然后点应用键。默认的密码是 0000。

要在 PANDEBUG 中改变前面板密码，要在程序的安全菜单（security）下输入如下东西：

```
piond/security>fp_password_set default 17 26 35 40
```

要在前面板上允许锁定功能，选择锁定配置页（就在刚才的密码设置页上一页）。将锁定设为允许，并输入前面板密码，然后点应用。

要在 PANDEBUG 的安全菜单下允许锁定功能，请输入：

```
piond/security>fp_lock_enable 1
```

一旦锁定被允许，你就可以在主屏幕上直接选择锁定图标来锁住这个单元。要解锁则必须输入密码并按解锁键。你也可以通过 PANDEBUG 程序的菜单命令来进行锁定和解锁。

```
piond/security>fp_lock
```

```
piond/security>fp_unlock
```

系统错误对策

这个装置，是用来控制系统中如果发生某个节点出错或失效等情况。

工程部署和硬件安装

部署、连接、仿真和上传

部署

文件--部署命令会开启一连串事件：

- *执行一个对工程文件的检验和编译
- *生成 DSP 编码和控制配置文件
- *角色部署对话框显示网络中发现的所有 NIONODE
- *你指定工程文件中的角色给发现的 NIONODE
- *部署--配置文件被下载到指定的 NIONODE
- *NWARE 连接到 NIONODE，可以进行控制和监视

角色部署对话框动态地显示网络上发现到的 NIONODE，并且允许给它们指派当前工程中的角色。对于每个被发现的 NIONODE，相关的网络名称、工程、角色和状态都会被显示出来。角色指派是通过把一个 NIONODE 用鼠标拖到工程角色上来实现，但是在点下 OK 之前，什么都不会发生的。一旦对话框被认可，作出的指派就会生效，执行连接。这要花掉一段时间，因为 NIONODE 要下载工程文件到本机的存储器，然后开始音频处理。

通过文件--断开连接命令可以执行与正在运行的机器断开连接，失去控制和监视功能。

连接

文件--连接命令，用来连接到一个正在运行的硬件。基于当前打开的工程，NWARE 会自动寻找运行这个工程的硬件并连接到它，并使控制和监视功能生效。

文件--断开连接命令可以与正在运行的机器断开连接，失去控制和监视功能。

仿真

当你没有目标硬件（NIONODE、CAB 4n 等），或者不想打乱当前正在运行的系统，那么目标硬件是可以通过装有 NWARE 的 PC 机来进行仿真的。这种状态下允许进行手形控制、创建和加载预设以及练习控制编程。这些都是运行期间的硬件的功能，所以一般不会通过 PC 来运行。注意音频处理不能在仿真器中做到。

要仿真，调用文件--仿真命令（快捷键 ctrl+F9）

通过文件--取消仿真命令，可以断开和仿真器之间的连接并终止仿真器程序。

编译报告

在设计检查的步骤中，将会生成一个基于 HTML 格式的报告。此报告显示记录、警告、错误以及一个关于目标资源在此设计中的利用率的资源利用报告表。

很多记录、警告和错误可以被点击，使设计框中对应此记录的相关元件高亮显示。这个功能对资源报告表的某些地方也适用。要取消高亮显示，只要在报告中点击一下别的地方就可以了。

上传和连接

工程文件可以被保存到或取回自 NIONODE。当在部署时出现角色部署对话框的时候，可以勾选保存工程到目标选项，这会使全部的工程文件下载和存储在 NIONODE 上。注意，如果这个工程文件从来没有被保存过，那么这个选项是不可用的。例如 untitled2 就不可以被存储到目标硬件。

文件--上传和连接命令，它会列出当前网络上目标硬件中运行的程序列表。选择一个工程将会上传它，出现一个另存为对话框。在本地保存之后，将连接到运行的工程。

NIONODE 硬件监视

远程日志

输出框里的远程日志页包含了一个当前发现的 NIONODE 的列表。你可以允许连机检测一个 NIONODE 的日志，只要右击选择允许连机检测。通过点选 NIONODE 中的一个，可以显示它已经连机检测到的日志。在日志的显示区域里右击鼠标，弹出菜单，允许你配置显示内容的过滤。

远程状态

输出框里头的远程状态页包含了单前发现的 NIONODE 的列表。你可以显示选择一个 NIONODE 来显示关于它的动态状态信息树。

NIONODE 硬件

好了，插电连线。

LAN

XDAB

EXPANSION SLOTS

CM-1

FRONT PANEL SERIAL PORT

BACK PANEL SERIAL PORT

GPIO

参见附录 B：前面板和 GPIO 外部控制。

NIONODE 配置

网络配置

部署，需要所有的目标硬件和 NWARE 电脑在同一个网络里，并且有一个前面板界面页用于进入网络设置。

主机名称

NIONODE 可以给出主机名称，以方便互相识别。而且所有基于 NWARE 的界面都支持显示主机名称。但是目前还不支持 DSN ,所以当使用基于 PING 或者 TCP/IP 协议的外部控制时 ,你还不能通过主机名称来和 NIONODE 通信，而是必须使用 IP 地址。注意，主机名称与角色名称不同，如果你愿意，

你可以给主机相同的名称命名。但是有些情况下你可能不会希望它们有相同的名字。

什么字符是合法的？？？

模式

DHCP

静态 IP

IP 地址

NIONODE 既可以被指派一个静态 IP 地址，也可以使用 DHCP 来从 DHCP 服务器获得一个 IP 地址。

IP 子网

当然，子网掩码无法更改，默认的是 255.255.255.0。

固件升级程序

NIONODE 中的固件可以通过升级固件功能来进行检查和升级。这个功能通过工具--升级 NIONODE 固件的菜单命令来启动。对话框中显示被发现的 NIONODE 列表，和它们的状态。右击需要升级的 NIONODE，选择升级包。它就会被下载和安装进这个单元里。

日期

为了支持某些功能，NIONODE 需要知道现在是什么时间。前面板上有个界面页可以设置时区和时间。通过这个界面设置了一台 NIONODE 后，所有在这个网络内的其他 NIONODE 的时间都会和它同步。但是时区设置不会影响到其他机器。

大容量存储设备替换程序

如果 NIONODE 中的大容量存储设备（CompactFlash 或者 IDE 硬盘）需要更替，那么可以通过 NIONODE BIOS 对它进行格式化和初始化。

详细资料参见[这里](#)。

检测和问题解决

前面板 LED 指示

前面板上有六个发光二极管来帮助指示单元状态。

FAULT

*红色：音频没有工作

*不亮：音频在工作

FAULT 指示灯遵循错误继电器的状态。（错误继电器的信号在 GPIO 接口的针脚上。）注意：红色并不一定代表是出现故障，它只是表示目前音频信号没有被处理和通过。这个状态经常在单元重启或者没有配置好运行一个工程时出现。当然，在单元出现故障，不能处理和通过音频信号时，这个状态也会产生。

X-DAB

*绿色：主备环已经在运转。

*绿色闪烁：只有一根线缆，运行在备份模式下，不丢失音频信号。

*红色闪烁：多个线缆/节点丢失，运行在备份模式下，丢失部分音频信号。

*红色固定：单元没有链接？？？

*不亮：(在这个 NIONODE 的当前工程/角色中) 没有使用 XDAB。

注意 XDAB 的 LED，只有在 NIONODE 的当前工程中使用了 XDAB，并且这个 NIONODE 已经启动并传输音频信号的情况下，它才会起作用。

LAN

*绿色：以太网链接

*不亮：没有以太网链接

COBRANET

*绿色：以太网链接

*绿色闪烁：以太网链接，并且 NIONODE 是 CobraNet conductor。

*红色：没有链接，或者 CM-1 无效，或者故障。(NIONODE 配置在使用 CM-1 模式下时。)

*不亮：没有链接，或者 CM-1 无效，或者故障，(NIONODE 配置在不使用 CM-1 模式下时。)

注意，即使 NIONODE 没有配置使用 CM-1 (在工程设计时)，链接和 conductor 状态仍然会被报告，单元在其所属网络中依然被作为一个 COBRANET

节点被访问。只不过，只有在配置为使用 CM-1 的模式下时，当单元的链接或者 CM-1 模块出现问题时，才会亮红灯。

IDE

*绿色：大容量存储器 (CompactFlash) 处在读/写状态。

*不亮：不活动。

IDE 指示灯跟 PC 机或者笔记本电脑上的硬盘状态指示灯很相似。

Attn

*不亮：没有重要的状态改变

*红色闪烁：表示发生了错误或者警告。这个指示灯和一个按钮是一体的，当发光二极管闪烁时，按这个键就可以直接导航液晶屏幕到问题由来的页面。

Power

*不亮：单元关机

*橙色：单元启动或关闭中

*绿色：机器已启动

以上 LED 的颜色在完成本文前尚未确定，无法预知上述内容是否精确。

后面板 LED 指示灯

在后面板 XDAB、LAN 和 cobranet 模块的 RJ45 口上都嵌入了 LED 管。正如你所期待的一样，这些 LED 指示连接状态。

XDAB IN and OUT 左 LED

*不亮：

*绿色：

*红色：

XDAB IN and OUT 右 LED

*不亮：

*绿色：

*红色：

LAN 左 LED

*不亮：

*绿色：

*红色：

LAN 右 LED

*不亮:

*绿色:

*红色:

CobraNet 模块(CM-1) Primary/Secondary 左 LED

*不亮: ???

*绿色: 链接

*绿色闪烁: 活动

*红色: 错误

CobraNet 模块(CM-1) Primary/Secondary 右 LED

*不亮: ???

*绿色: 使用中

*橙色: conductor

前面板状态和错误日志

安全模式下启动

数字音频时钟

详细资料参见[这里](#)。

独立运转

一旦一个工程部署到一个或多个 NIONODE 上，那些节点在给电后就会加载和运行该工程。