

用于算法图形设计的 AlgoBuilder 应用程序入门

引言

AlgoBuilder 是一种图形设计应用程序，用于构建和使用算法。

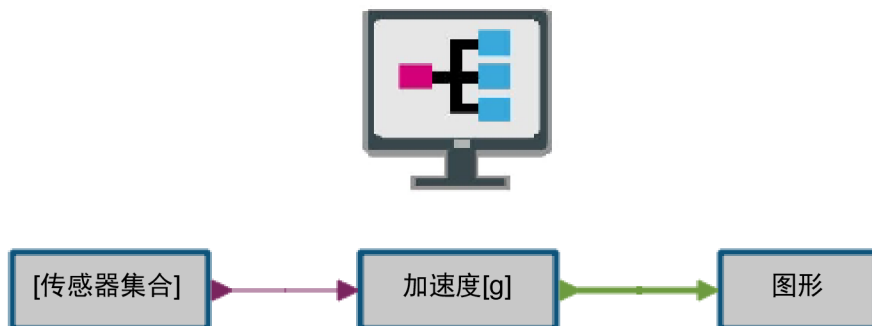
它可以快速开发 **STM32** 微控制器和 **MEMS** 传感器应用原型，其中已包含现有算法（例如传感器融合或计步器）、用户定义的数据处理功能块和其他功能。

该应用程序使用图形界面简化了理念验证的实现过程，无需编写代码。

AlgoBuilder 重复使用之前定义的功能块，将多种功能组合到一个项目中，并在 **Unicleo-GUI** 中使用绘图和显示功能实时实现数据可视化。

AlgoBuilder 使用 **STM32 ODE**（开放式开发环境）生态系统，该生态系统组合了 **STM32 Nucleo** 板（**NUCLEO-F401RE** 或 **NUCLEO-L476RG**）、**X-NUCLEO-IKS01A2** 或 **X-NUCLEO-IKS01A3** 扩展板等硬件、软件（**STM32 HAL** 驱动程序、**BSP** 结构、低和高层传感器驱动程序）和 **Unicleo-GUI**。

图 1. AlgoBuilder 应用程序框图



1 说明

1.1 概述

AlgoBuilder 的主要目标是：

- 快速开发 **STM32** 微控制器和 **MEMS** 传感器应用原型，其中已包含现有算法（例如传感器融合或计步器）、用户定义的数据处理功能块和其他功能
- 无需编写代码，使用图形界面简化理念验证的实现过程
- 重复使用之前定义的功能块
- 在一个项目中组合多种功能
- 使用绘图和显示功能在 **Unicleo-GUI** 中实时实现数据可视化

应用的主要特性包括：

- 简单的算法图形设计（拖放、连接、设置属性、构建和上传）
- 可选多级设计
- 库中提供了大量功能块，包括运动传感器算法（例如传感器融合、陀螺仪、磁力计校准、计步器等）
- 用于 **FFT** 分析的集成功能块
- 用于创建自定义功能块的功能块创建器
- 自动验证设计规则
- 从图形设计生成 **C** 代码
- 使用外部编译器（**STM32CubeIDE**、**IAR EWARM**、**Keil µVision®**和 **System Workbench for STM32**）
- 通过集成的数据输出监控器或 **Unicleo-GUI** 显示生成的固件输出
- 开放的 **XML** 格式，便于功能块和设计存储
- 能够使用 **MQTT** 协议将输出数据发送至 **AWS** 云
- 通过连接的 **X-NUCLEO-IKS01A2** 或 **X-NUCLEO-IKS01A3** 扩展板、**SensorTile STEVAL-STLKT01V1** 和 **SensorTile.box STEVAL-MKSBOX1V1** 支持 **NUCLEO-F401RE** 或 **NUCLEO-L476RG**
- 网络更新，并提供新版本自动通知功能
- 免费易用的许可条款

AlgoBuilder 使用 **STM32 ODE**（开放式开发环境）生态系统，该生态系统组合了 **STM32 Nucleo**（**NUCLEO-F401RE** 或 **NUCLEO-L476RG**）、**X-NUCLEO-IKS01A2** 扩展板、软件（**STM32 HAL** 驱动程序、**BSP** 结构、低和高层传感器驱动程序）和 **Unicleo-GUI**。

1.2 先决条件

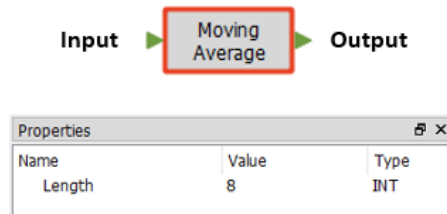
为了充分利用 **AlgoBuilder** 的功能，需用到下列软件和硬件。

- 以下 IDE 之一：
 - **STM32CubeIDE** 1.0.0 或更高版本
 - **System Workbench for STM32** (**SW4STM32**) v1.13.1 或更高版本
 - **IAR-EWARM** 8.11.0 或更高版本
 - **Keil µVision** 5.22 或更高版本
- **Unicleo-GUI**
- **STM32CubeProgrammer** (**STM32CubeProg**)
- **STM32** 虚拟 COM 端口驱动程序 (**STSW-STM32102**)
- **NUCLEO-F401RE/NUCLEO-L476RG** 与 **X-NUCLEO-IKS01A2/X-NUCLEO-IKS01A3/SensorTile STEVAL-STLKT01V1/SensorTile.box STEVAL-MKSBOX1V1**

1.3 术语和参考文献

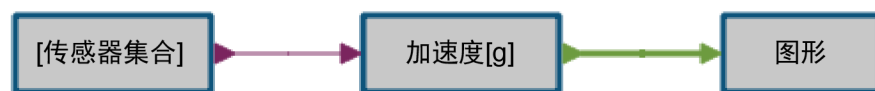
- 功能块是一个数据处理元件，具有一个或多个输入或输出。它处理输入并生成输出，可能有一个或多个属性。

图 2. AlgoBuilder 功能块



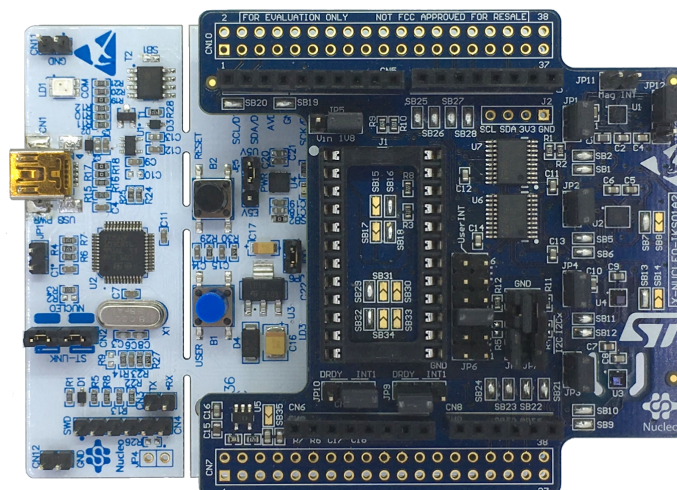
- 设计是一组多个彼此连接的功能块。

图 3. AlgoBuilder 设计



- 节点表示两个功能块之间的连接。
- STM32 微控制器的固件可以从设计开始构建。
- 使用具有 STM32 微控制器的 STM32 Nucleo 开发板进行设计测试。
- X-NUCLEO-IKS01A2 运动 MEMS 和环境传感器扩展板，嵌入了加速度计、陀螺仪、磁力计以及温度、湿度和压力传感器。

图 4. STM32 Nucleo (NUCLEO-F401RE) + X-NUCLEO-IKS01A2



- Unicleo-GUI 可用于显示固件输出。

图 5. Unicleo-GUI



1.4 工作原理

工作流程从使用简单的“拖放”法进行所需功能的图形设计开始。

您可以使用以库的形式提供的预定义功能块。

您还可以创建自定义功能块。为了运行功能块，可能或必须调整一些功能块属性（示例中，在滤波功能块属性中定义了滤波系数）。然后，您可以使用节点将兼容的功能块互连。

AlgoBuilder 自动检查输入和输出之间的兼容性，并且只允许连接具有相同类型和规格的端子。

当设计完成时，**AlgoBuilder** 从定义的图形设计生成 C 代码。

使用 C 代码生成器，结合预先准备的固件模板和二进制库，创建最终的固件项目。

可使用外部编译工具编译项目，并支持最常用的集成开发环境（IDE）（具有 GCC 编译器的 **System Workbench for STM32**、Keil μ Vision 和 IAR Embedded Workbench）。

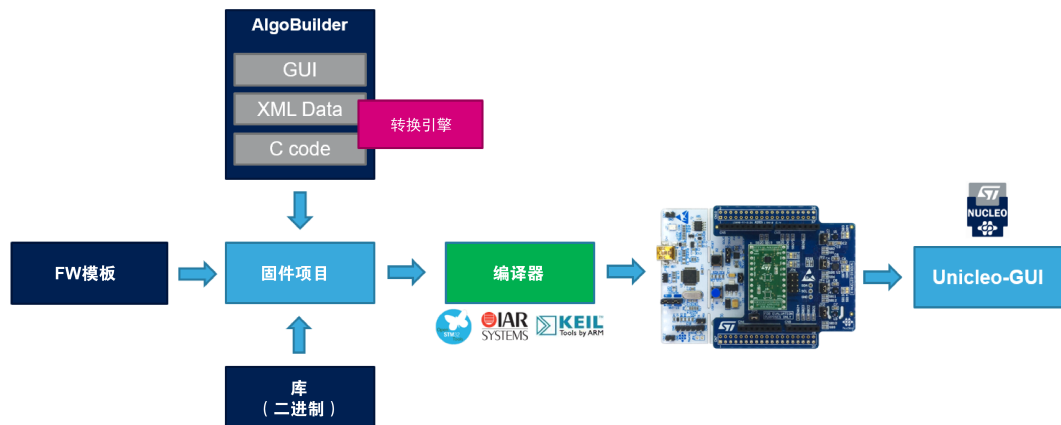
然后，通过生成的二进制文件对 **STM32 Nucleo** 板进行编程。在执行固件时，它从选择的传感器读取数据，通过算法处理数据，并将结果发送至 **Unicleo-GUI** 应用。

在图形设计期间，您可以选择结果的查看方式。支持图形、逻辑分析仪、柱状图、三维绘图、散布图、直方图、茶壶、FFT 绘图和文本值。

在启动期间，固件配置 **Unicleo-GUI**，以便以需要的格式进行显示。

图形设计和库都保存为 XML 文件。

图 6. Algebuidler 的工作原理

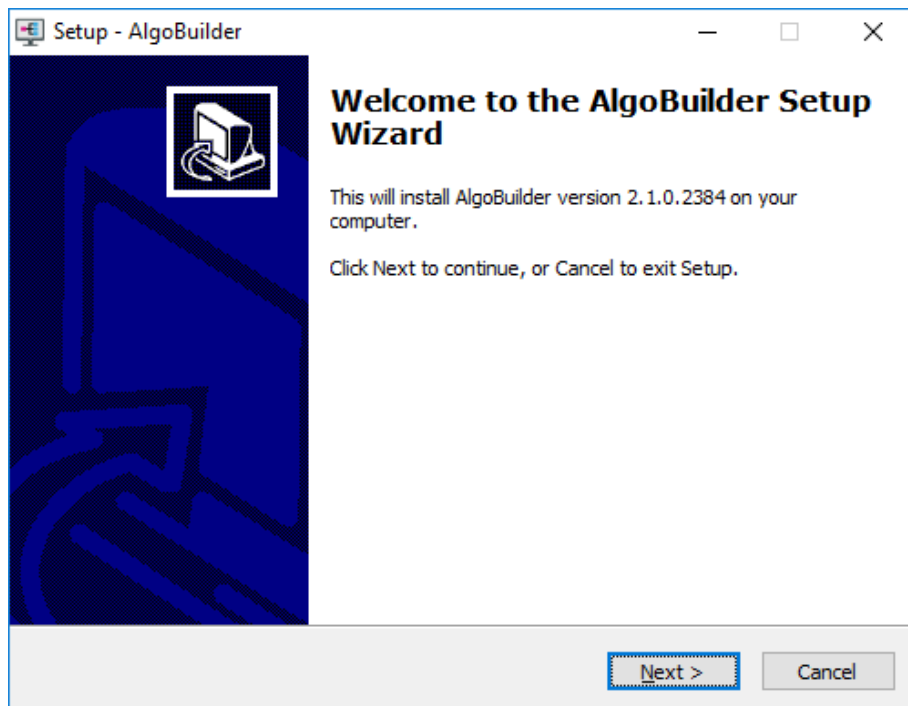


2 入门指南

2.1 安装软件

AlgoBuilder 软件被设计为在 Microsoft® Windows 操作系统中运行。为了安装该应用程序，运行 *Setup_AlgoBuilder.exe*，按照说明操作，并在安装完成后执行 AlgoBuilder。

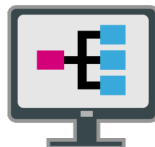
图 7. AlgoBuilder 安装程序



2.2 首次运行软件

安装程序可能在您的 Windows 桌面上和/或 Windows 开始菜单中创建了快捷方式。双击该快捷方式可以运行 AlgoBuilder。如果没有创建快捷方式，您可以通过执行 *AlgoBuilder.exe* 文件运行 AlgoBuilder，该文件位于应用的安装目录中（默认位置为 C:\Program Files (x86)\STMicroelectronics\AlgoBuilder）。

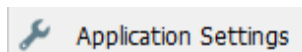
图 8. AlgoBuilder 图标



2.3 应用设置

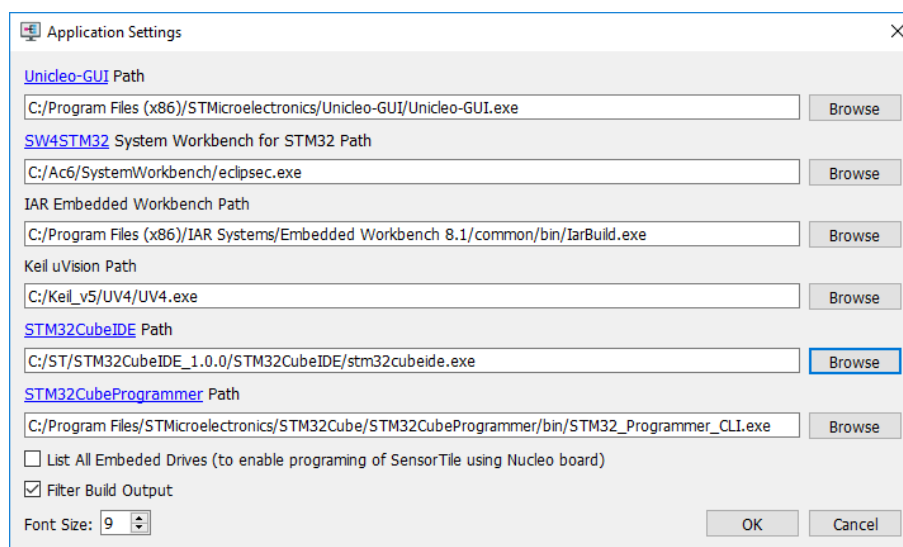
您可以在“文件→应用设置”中调整 **AlgoBuilder** 的配置。

图 9. 应用设置的菜单选项



Step 1. 指定 **Unicleo-GUI** 的路径。

图 10. 应用设置窗口



如果路径（**Unicleo-GUI.exe**）设置正确，可以快速地从工具栏或 **AlgoBuilder** 菜单执行 **Unicleo-GUI**。如果没有设置路径，工具栏中的相应图标和菜单项会被禁用。

Step 2. 指定至少一个 IDE 的路径。

对于 **STM32CubeIDE**，指定 **stm32cubeide.exe** 的路径；对于 **System Workbench for STM32**，指定 **eclipse.exe** 的路径；对于 **IAR Embedded Workbench**，指定 **IarBuild.exe** 的路径；对于 **Keil μVision**，指定 **UV4.exe** 的路径。

Step 3. 设置与 **SensorTile** 编程有关的选项。

如需在 **STM32 NUCLEO** 板上使用 **SensorTile** 并使用内置 **ST-LINK V2.1** 编程器进行编程，可选择列出所有嵌入式驱动。

Step 4. 指定 **STM32CubeProgrammer** 的路径。

如需使用单独的 **ST-LINK** 编程器或 **DFU** 编程模式，指定之前安装的 **STM32CubeProgrammer** 的路径。

Step 5. 设置应用程序行为。

如果使能了筛选构建输出，**AlgoBuilder** 将自动筛选外部编译器的输出，以方便在控制台上识读。还可以调整字体大小，以方便在高分辨率显示器上识读。

2.4 网络更新设置

该应用程序能够检查是否有新版本可用并发出通知。然后，您可以决定是否下载并安装新版本。必须在文件菜单中的网络更新设置中适当地设置一些功能网络参数。

图 11. 网络更新设置的菜单选项

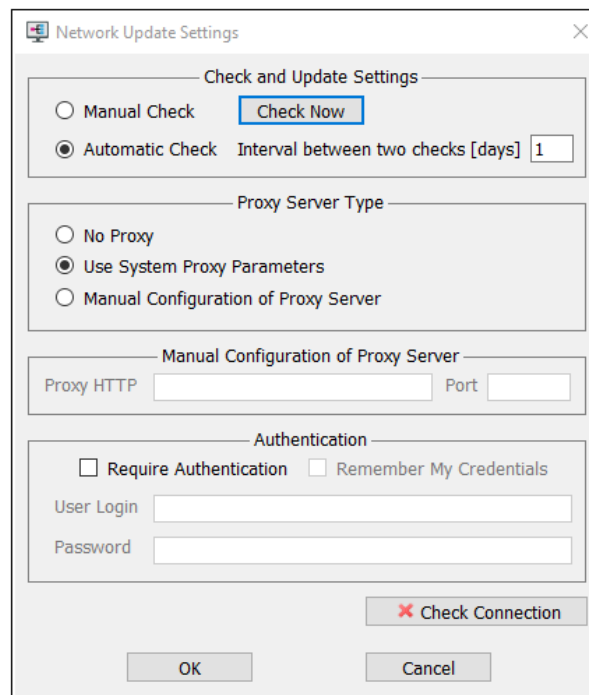


设置对话框分为几个不同部分：

1. 在第一部分中，您可以选择手动或自动检查。然后，您可以调整周期。如果设置的时间间隔为零天，将在每次应用程序启动时执行更新检查。如需立即检查更新，点击立即检查。
2. 第二部分包含代理服务器类型设置选项。
提示：如果您选择使用系统代理参数，则通常必须在运行更新检查前打开网络浏览器运行所有安全脚本。
3. 第三部分包含代理手动配置字段，在这里可以输入代理 HTTP 名称和端口号。
4. 最后一部分包含身份验证凭据字段（如果需要）。

检查连接按钮可用于检查更新服务器是否可访问。

图 12. 网络更新设置窗口


 A screenshot of the 'Network Update Settings' dialog box. It has a title bar with a close button. The dialog is divided into several sections:

- Check and Update Settings:** Contains two radio buttons: 'Manual Check' and 'Automatic Check' (which is selected). Next to 'Automatic Check' is a text field labeled 'Interval between two checks [days]' with the value '1'. There is a 'Check Now' button.
- Proxy Server Type:** Contains three radio buttons: 'No Proxy', 'Use System Proxy Parameters' (which is selected), and 'Manual Configuration of Proxy Server'.
- Manual Configuration of Proxy Server:** Contains two text fields: 'Proxy HTTP' and 'Port'.
- Authentication:** Contains two checkboxes: 'Require Authentication' and 'Remember My Credentials'. Below them are two text fields: 'User Login' and 'Password'.

 At the bottom right, there is a 'Check Connection' button with a red 'X' icon. At the bottom center, there are 'OK' and 'Cancel' buttons.

3 使用 AlgoBuilder

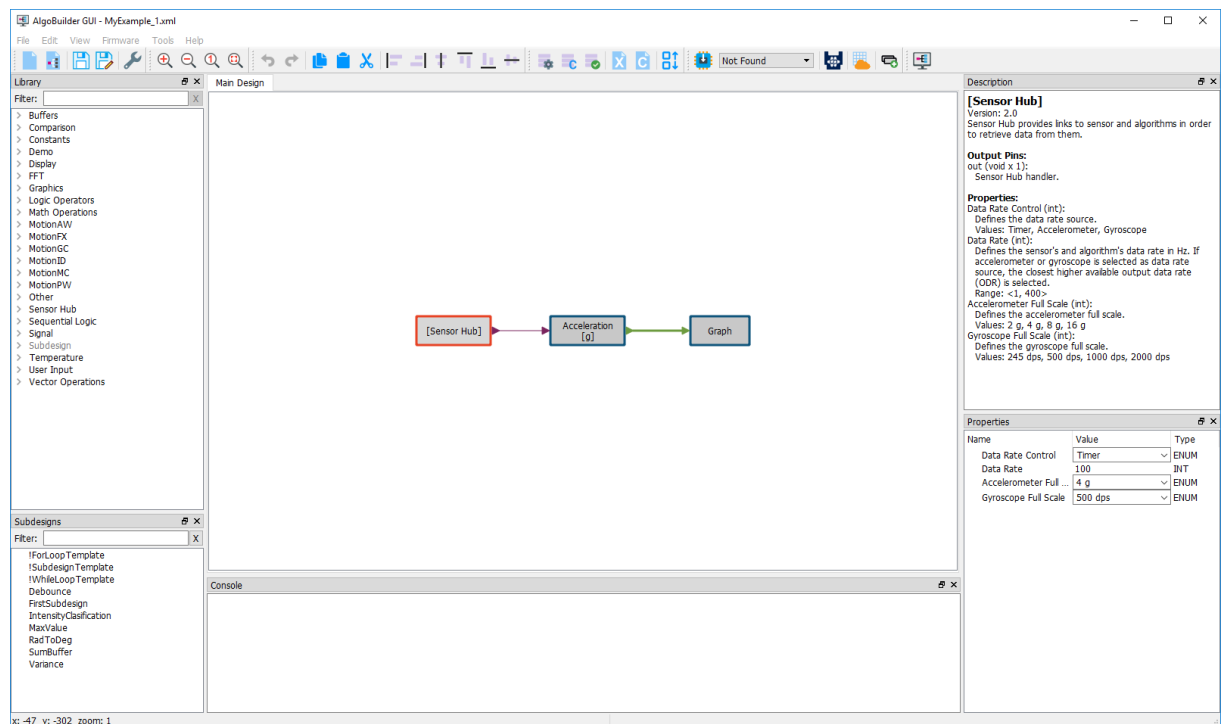
AlgoBuilder 主窗口包含：

- 中央工作区，用户在这里使用功能块设计算法；
- 库面板，显示可用库及其功能块的列表，功能块可以拖放到工作区窗口；
- 子设计面板，显示可用于设计和子设计模板的列表；
- 说明面板，显示选定组件（功能块、连接等）的相关信息；
- 属性面板，显示选定功能块的所有可用属性；
- 控制台面板，显示来自 AlgoBuilder 或外部编译器的消息。

AlgoBuilder 应用程序具有标准菜单和工具栏，方便用户快速访问频繁使用的功能。

注：您可以更改所有面板和工具栏的位置。面板可以在“视图”菜单中打开和关闭。

图 13. AlgoBuilder 主窗口





3.1 工作区

在工作区创建开发的算法设计。

Step 1. 将需要的功能块置于工作区

注：只需将库面板上的功能块拖放到工作区。

Step 2. 设置其属性

Step 3. 为了进行连接，点击要连接的输出并按住鼠标左键，将光标移动到应连接的输入。也可以反过来创建从输入到输出的连接。

注：您只能连接类型和规格相同的输入和输出。如果您尝试连接类型或规格不同的输入和输出，控制台将显示错误消息。

您可以修改一些功能块（如 MUX、Sum、And、Or 等）的输入数。

Step 4. 使用删除按钮删除任何组件，通过编辑菜单、工具栏或快捷方式剪切、复制和粘贴设计的任何部分。

Step 5. 将功能块上、下、左、右对齐。
最后选择的功能块决定了最终位置。

Step 6. 使用执行和撤销在工作区已执行的操作中前进和后退。

Step 7. 如需放大或缩小，使用 **Ctrl** 键和鼠标滚轮或视图菜单或工具栏中的相应功能。

Step 8. 选择全屏全屏显示整个设计。

Step 9. 选择 1:1 缩放将缩放因子设置为 1。

Step 10. 右键单击并长按工作区可探索设计内容。

3.2 库面板

您可以从库面板访问所有可用的库和特定库中的功能块。启动期间，AlgoBuilder 扫描[安装路径]/Library/和用户的根目录\STMicroelectronics\AlgoBuilder\Library，并加载其中的所有有效库。

图形库没有保存在 xml 文件中，而是由 AlgoBuilder 自动添加。

3.3 子设计面板

您可以从子设计面板访问所有可用的子设计。

启动期间，AlgoBuilder 扫描[安装路径]/Subdesigns/和用户的根目录\STMicroelectronics\AlgoBuilder\Subdesigns，并加载其中的所有有效子设计。子设计模板位于[安装路径]/Subdesigns/目录中。

3.4 说明面板

说明面板提供用户在工作区或库面板中选中的组件的相关信息。

如果您选中了一个功能块，将显示以下信息：

- 名称
- 版本
- 功能块功能的说明
- 所有输入的类型、规格和功能
- 所有输出的类型、规格和功能
- 所有功能块属性的说明



3.5 属性面板

如果功能块具有一个或多个属性，它们将显示在属性面板上。

每个属性都有名称、值和类型字段。

值可以修改。

AlgoBuilder 自动检查值是否有效，不允许设置无效的值（例如，超出可用范围的值）。

对于 **STRING** 类型，将禁止使用并自动删除%字符。

3.6 工具栏

工具栏提供对最常用功能的快速访问。工具栏的位置和功能的顺序可以调整。

表 1. AlgoBuilder 工具栏默认功能

工具栏图标	功能
	新设计
	打开现有设计
	保存设计（子设计）
	将设计（子设计）另存为其他文件
	打开“应用设置”窗口
	放大
	缩小
	将缩放比设置为 1:1
	全屏显示设计
	撤销
	恢复
	复制
	粘贴
	剪切
	左对齐
	右对齐
	上对齐
	下对齐

工具栏图标	功能
	水平居中
	垂直居中
	打开“固件设置”窗口
	从图形设计生成 C 代码
	构建固件/验证子设计
	在默认的文本编辑器中显示设计的 xml 源文件
	在默认的文本编辑器中显示 C 代码
	排列工具箱按钮
	编程目标
	运行 Unicleo-GUI 应用程序
	输入数据和输出数据监控器与 AWS 连接
	打开功能块创建器
	打开“关于”窗户

4 数据类型

AlgoBuilder 使用四种数据类型：

- **FLOAT** 表示实数，用于浮点运算（例如，在加速度功能块输出中）。C 代码中使用浮点型变量。大小为 4 字节。
- **INT** 表示整数（例如在计数器功能块中）。在 C 代码中，使用 `int32_t` 变量。大小为 4 字节。
- **VARIANT** 用于一组功能块的输入；**VARIANT** 数据基于连接到此输入的输出生类型改变其类型（例如，对比较功能块的输入使用 **VARIANT** 类型）。
- **VOID** 专用于传感器集合和其数据输出之间的连接。此类型不能可视化。

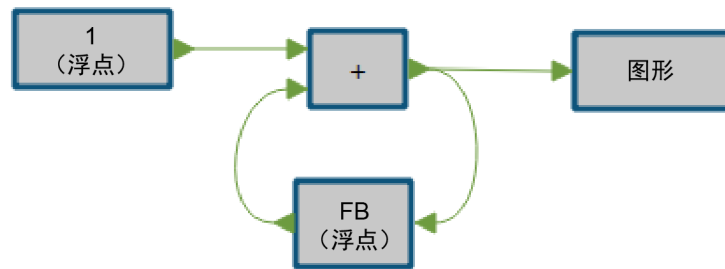
每个输入或输出都有各自的类型和规格：连接线的厚度表示输入和输出的规格，连接线的颜色表示类型。

规格值可以在属性中进行修改。

重要： 只能将具有相同类型和规格的输入和输出连接到一起。唯一的例外是 **VARIANT** 输入，它将获得所连接输出的类型。

不可能将同一功能块的输入和输出相连。如果需要这样做，需使用特定数据类型的反馈功能块。反馈功能块具有初始值，它定义了首次运行时功能块的输出值。

图 14. 使用反馈功能块



5 有条件执行

某些情况下，只有在特定条件有效时才需要执行功能块操作。对于这种情况，可以为选定功能块添加有条件执行输入。然后，此输入定义是否会执行功能块代码。这在生成的 C 代码中用 `if` 语句表示。如需添加或删除有条件执行输入，可使用鼠标右键点击功能块。

图 15. 有条件执行



6 库

最新安装的 AlgoBuilder 提供以下库。

表 2. AlgoBuilder 的库

Library	目录
缓冲区	用于操作数据缓冲区的功能块
比较	用于比较两个值的功能块（例如：>、<、=等）
常量	用于常量定义的功能块
显示	用于 Unicleo-GUI 应用程序中数据可视化的功能块
FFT	与 FFT 分析相关的功能块
图形	为设计提供注解的文本说明
逻辑运算符	用于逻辑运算的功能块（例如：与、或等）
数学运算	用于各种数学运算的功能块（例如：+、-、/等）
其它	辅助功能块（例如：mux、demux、类型转换等）
传感器集线器	可以从主要传感器集合功能块访问传感器、预构建算法和功能块，以便从连接的传感器采集数据。
顺序逻辑	用于顺序逻辑的功能块（触发器）
信号	用于信号处理的功能块（例如：滤波器等）
子设计	子设计的输入和输出端子
温度	用于温度单位换算的功能块
用户输入	允许用户通过 Unicleo-GUI 应用程序向正在运行的固件实时发送任意数据的功能块
矢量运算	用于矢量运算的功能块（例如：计算幅度等）

已准备好的二进制形式的算法也可以集成到用户设计中。

表 3. AlgoBuilder 支持的库

Library	功能
MotionAC	加速度计校准算法
MotionAW	腕戴式器件的活动识别算法
MotionEC	电子罗盘算法
MotionFX	传感器融合算法
MotionGC	实时陀螺仪校准算法
MotionID	运动强度检测算法
MotionMC	实时磁力计校准算法
MotionPM	移动设备的计步器算法
MotionPW	腕戴式器件的计步器算法
MotionTL	倾斜检测算法

仅当设计中使用了至少一个功能块时，固件中才包含二进制库。这将减少占用的 FLASH 和 RAM 存储器空间。

7 创建您的第一个设计

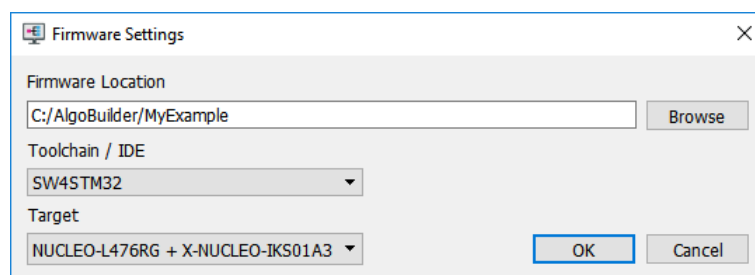
作为第一个示例，您可以创建这样的设计：以选定数据率从加速度计传感器读取加速度，并将数据发送至 **Unicleo-GUI**，以在时间图表中实现可视化。

Step 1. 从空白设计开始，点击工具栏或文件菜单中的图标 （新建设计）。

系统自动打开“固件设置”窗口，您也可以点击工具栏或固件菜单中的图标 （固件设置）打开它。

Step 2. 设置输出固件所在目录的路径、要用于构建固件的 **IDE** 和要用于测试的目标。必须选择正确的目标。

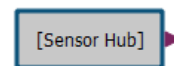
图 16. AlgoBuilder 固件设置窗口



Step 3. 将[传感器集合]功能块从传感器集合库拖放到工作区。

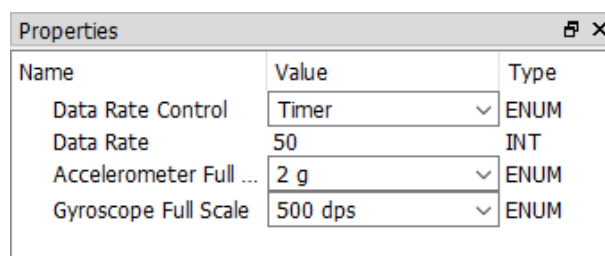
重要：必须从传感器集合功能块开始每个设计，通过它，您可以访问传感器和预构建的算法。

图 17. 传感器集合功能块



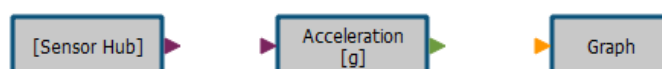
Step 4. 调整传感器集合的属性，选择“定时器”作为“数据率控制”的来源，将“数据率”设置为 50 Hz，将“加速度计满量程”设置为 2 g。

图 18. 传感器集合的属性



Step 5. 将传感器集合库中的加速度[g]功能块和显示库中的图形功能块添加到工作区。

图 19. 传感器集合、加速度[g]和图形功能块



Step 6. 调整图形属性。

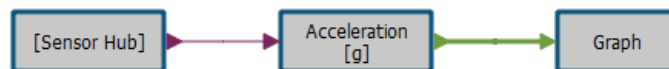
“图形中的曲线数”定义了其输入的规格。为了匹配加速度[g]输出规格，需将该值修改为 **3**。图形、波形和单位名称也可以修改。

图 20. 图形属性

Properties		
Name	Value	Type
Number of Curves	3	INT
Graph Name	Acceleration	STRING
Waveform 1 Name	X axis	STRING
Waveform 2 Name	Y axis	STRING
Waveform 3 Name	Z axis	STRING
Unit Name	[g]	STRING
Zero axis position	Middle	ENUM
Auto-scale	ON	ENUM
Full Scale	4	STRING

Step 7. 为了连接功能块，点击[传感器集合]输出，按住鼠标左键并移动至加速度[g]功能块的输入。

Step 8. 重复上一个步骤，将加速度[g]连接到图形功能块。

图 21. 传感器集合、加速度[g]和图形功能块


您的设计已就绪，可以使用它生成 C 代码。

Step 9. 点击工具栏或固件菜单中的图标 (生成 C 代码)。

此功能将固件模板复制到之前选择的文件中并创建 `algo_builder.c` 文件，该文件是图形设计的 C 代码表现形式。如果操作成功，控制台上将显示消息“代码生成已成功结束”。您可以通过点击工具栏或固件菜单中的图标 (显示 C 代码) 来检查生成的 C 代码。

图 22. `algo_builder.c` 文件中生成的代码

```

algo_builder.c - Notepad
File Edit Format View Help
#include "algo_builder.h"

void *Sensor_Hub_2_out;
float Acceleration_g_2_data[3];

sDISPLAY_INFO display_info_list[] = {
{INFO_TYPE_GRAPH,1,3,VAR_TYPE_FLOAT,0,"graph|Acceleration|[g]|1|19|X axis|Y axis|Z axis|1",0},
{0,0,0,0,0,0,0}};

void AB_Init(void)
{
    Sensor_Hub_Init(0, 100, 1, 1);
    Accelero_Init();
    Message_Length = 12;
}

void AB_Handler(void)
{
    Sensor_Hub_Handler(&Sensor_Hub_2_out);
    Accelero_Sensor_GetData(Sensor_Hub_2_out, Acceleration_g_2_data);
    Display_Update(Acceleration_g_2_data, &display_info_list[0]);
}
  
```

注：如果目标目录中的固件模板意外损坏或被删除，您可以通过点击固件菜单中的图标 (重新初始化固件) 启动固件模板的重新初始化。

Step 10. 点击图标 (构建固件) 调用外部 IDE，以便为 STM32 微控制器构建固件项目并生成二进制文件。

控制台显示编译器的输出。在编译结束后，屏幕上显示消息“构建过程结束”。如果没有来自编译器的错误消息，则在 STM32 Nucleo 板的固件编程就完成了。

图 23. System Workbench for STM32 的输出

```

Console
Generating binary:
arm-none-eabi-objcopy -O binary "STM32L4xx-Nucleo-Project.elf" "STM32L4xx-Nucleo-Project.bin" & arm-none-eabi-size --format=berkeley STM32L4xx-Nucleo-Project.elf
text      data      bss      dec      hex      filename
169920    7672     46628   224220   36bdc    STM32L4xx-Nucleo-Project.elf

11:49:17 Build Finished (took 5m:16s.454ms)
Invoking scanner config builder on project
SESSION 2018-01-01 11:43:55.059 -----
Build Process Finished
Exit Code: 0
  
```

注：在固件构建过程中，只编译修改过的文件。如需使用外部工具重新编译所有文件，在固件菜单中选择图标 (重新构建固件)。

Step 11. 点击工具栏或文件菜单中的图标 (保存设计) 保存设计。

8 目标编程

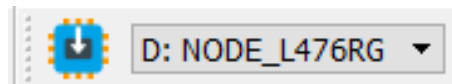
8.1 STM32 Nucleo 板

AlgoBuilder 自动扫描连接的 STM32 Nucleo 板。

连接的 STM32 Nucleo 板的列表显示在工具栏中“编程目标”图标旁边。

如果固件构建成功并选择了 STM32 Nucleo 板，则可以通过按下按钮 （编程目标）对板进行编程。

图 24. 编程目标图标和选择框



8.2 SensorTile

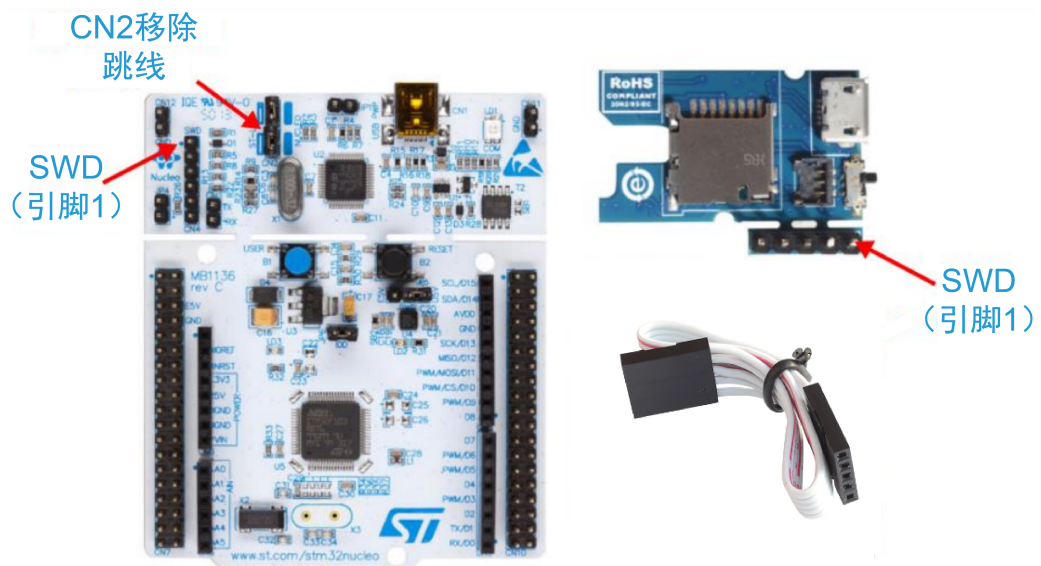
SensorTile 无内置编程器。为了对板进行编程，需使用 SensorTile 套件包中提供的 5 引脚扁平电缆，将外部 ST-LINK 编程器连接至 Cradle 上的 SWD 连接器。

获得 ST-LINK 器件的最简单方法是获得一个 STM32 Nucleo 板，它捆绑了 ST-LINK V2.1 调试器和编程器。必须移除 STM32 Nucleo 板上的两根 CN2 跳线，才能对 SensorTile 进行编程。

如需在 AlgoBuilder 选择框中显示所有 STM32 Nucleo 板，可在 AlgoBuilder 设置中选中列出所有嵌入式驱动。然后，您可以选择要用于 SensorTile 编程的 STM32 Nucleo 板，并按下编程按钮 .

如需使用单独的 ST-LINK 编程器，必须在 PC 上安装 **STM32CubeProgrammer**，并在 AlgoBuilder 应用设置中设置该工具的路径。完成后，ST-LINK 编程器将显示在工具栏的目标列表中。

图 25. STM32 Nucleo 板和 SensorTile cradle SWD 连接器



8.3 SensorTile.box

SensorTile.box 无内置编程器。为了进行器件编程，有两种选择：

1. 使用 ST-Link 编程器

如需对板进行编程，可将外部 ST-LINK 编程器连接到 SensorTile.box 板的 14 引脚 JTAG 连接器。

需使用 20 引脚（2.54 mm 间距）转 14 引脚（1.27 mm 间距）适配器。


必须在 PC 上安装 [STM32CubeProgrammer](#)，并在 AlgoBuilder 应用设置中设置该工具的路径。完成后，ST-LINK 编程器将显示在工具栏的目标列表中。

然后，您可以点击编程按钮  对 SensorTile.box 进行编程。

图 26. STM32ST-Link 编程器和 SensorTile.box 互连



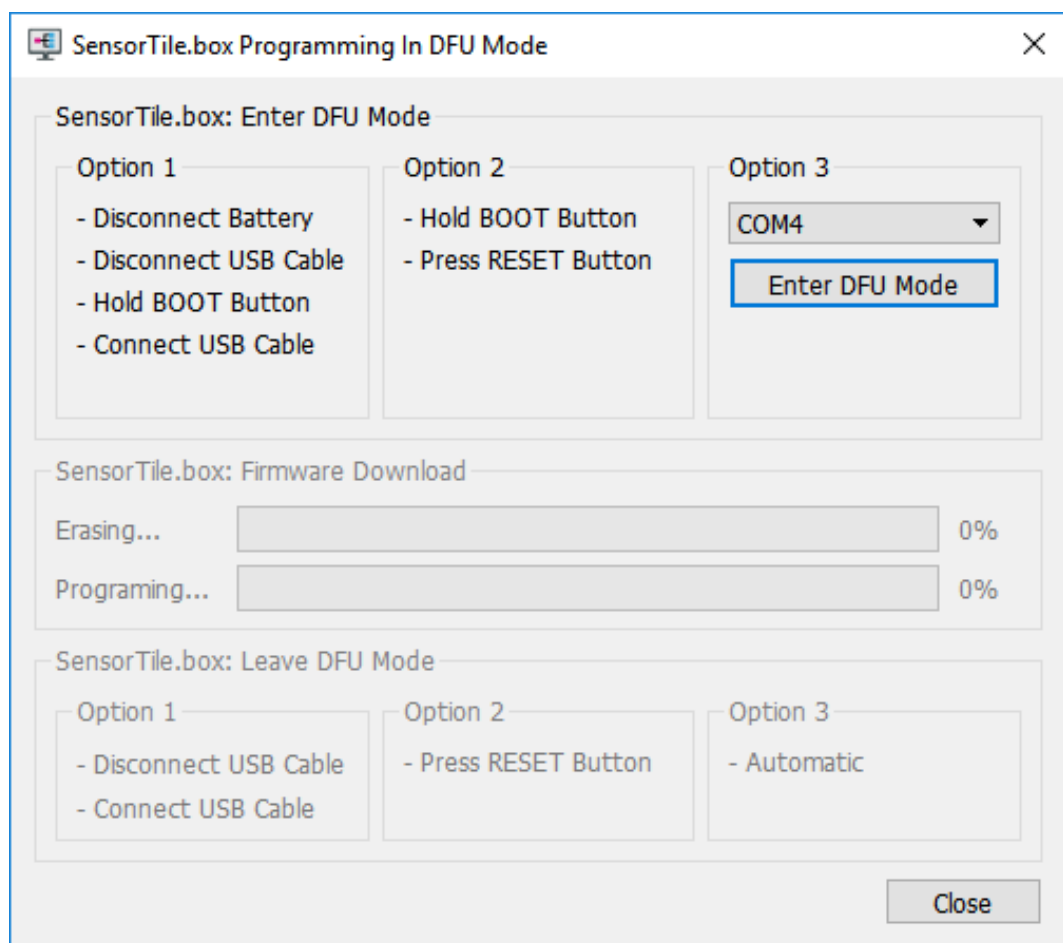
2. 使用 DFU（器件固件升级）

在 DFU 模式下，无需使用编程器即可对器件进行编程。只有在固件设置中将 **SensorTile.box** 选作目标时，才能使用 DFU 模式。如果选择了 **SensorTile.box**，DFU 模式将显示在工具栏的目标列表中。通过点击编程按钮 ，可打开 DFU 的专用窗口。为了将器件切换至 DFU 模式，有三种选择。窗口中描述了具体步骤。如果器件中已有 **AlgoBuilder** 固件，可通过点击“选项 3”下的进入 DFU 模式进入 DFU 模式。在器件切换至 DFU 模式后，新固件将自动擦除存储器并进行编程。

可以自动或手动退出 DFU 模式（选项 1 或 2）。

对于 DFU 模式，必须在 PC 上安装 **STM32CubeProgrammer**，并在 **AlgoBuilder** 应用设置中设置该工具的路径。

图 27. 在 DFU 模式下进行 **SensorTile.box** 编程



9 使用 Unicleo-GUI

Unicleo-GUI 可用于检查固件的功能。您可以将来自固件的数据可视化，并从 Unicleo-GUI 向正在运行的固件发送数据。

9.1 数据可视化

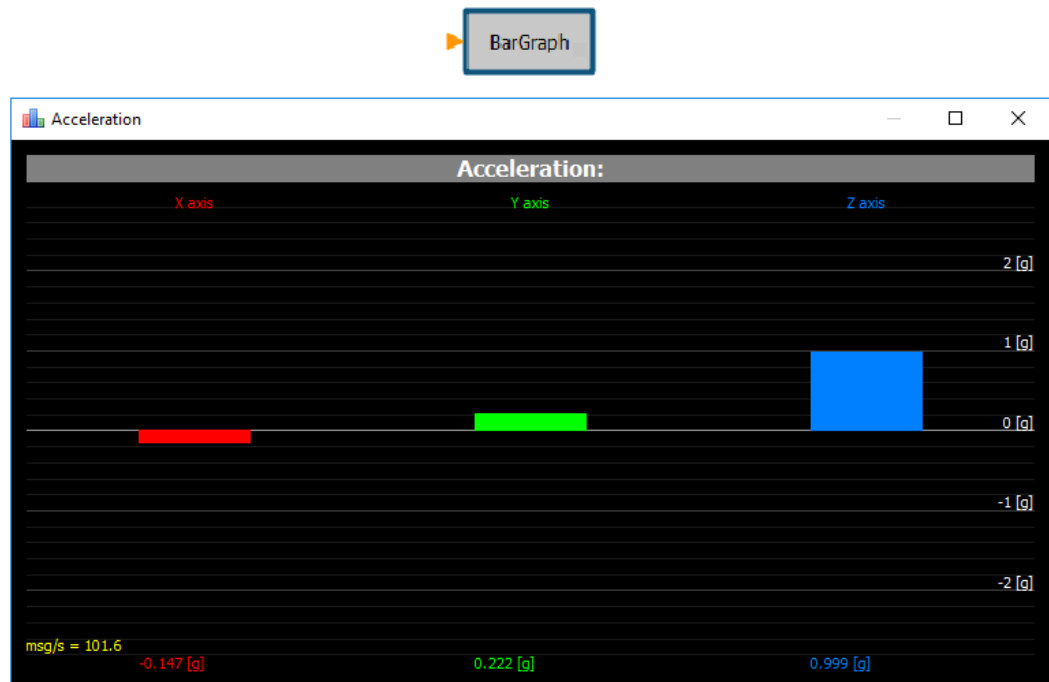
数据可视化有八种类型：

- 柱状图
- 三维茶壶模型
- 图形
- 直方图
- 逻辑分析仪
- 三维绘图
- 散布图
- 文本值

为了将数据发送到 Unicleo-GUI，需要为设计添加合适的功能块。

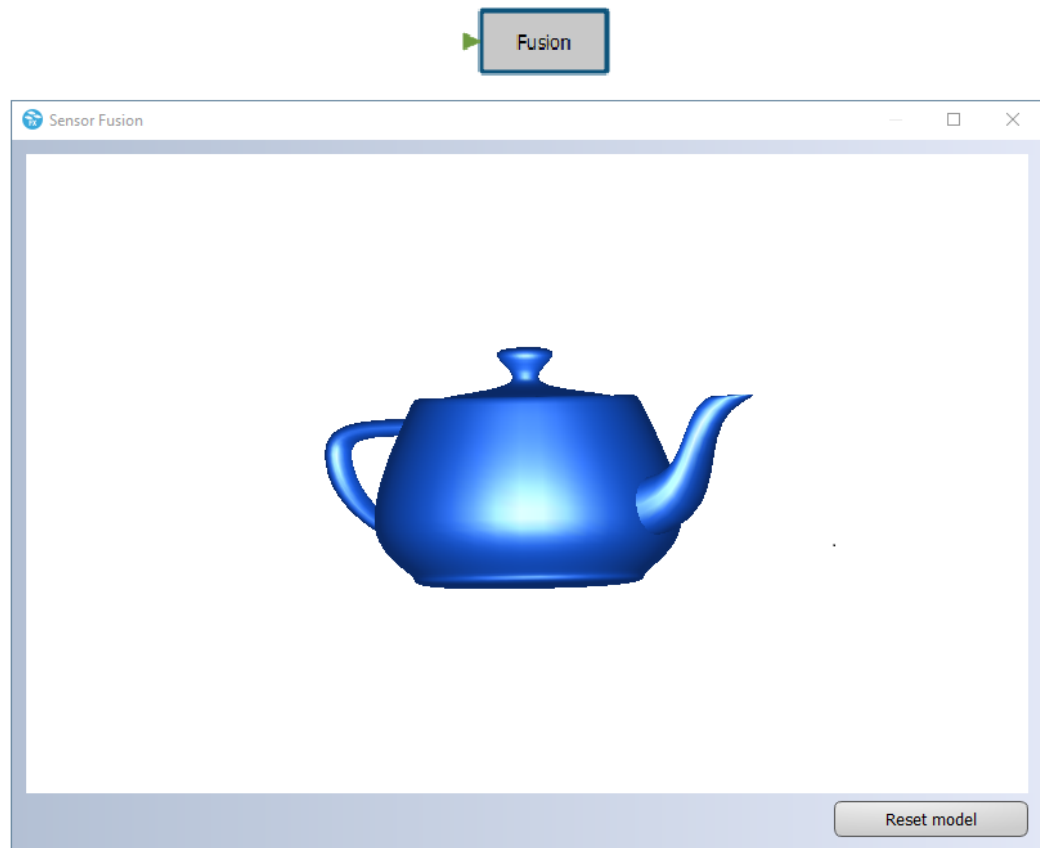
- 将显示库中的柱状图功能块添加到设计中，可以柱状图的形式显示数据。柱状图适合用来快速检查实际值，无需查看历史数据。此图适用于任何浮点或整数值，每个柱状图最多可包含 6 个条柱。在属性字段中，您可以设置图和每个条柱的名称、Y 轴上的单位、Y 轴上 0 的位置、满量程和自动缩放的使能或禁用。

图 28. 柱状图功能块和 Unicleo-GUI 中数据可视化的示例



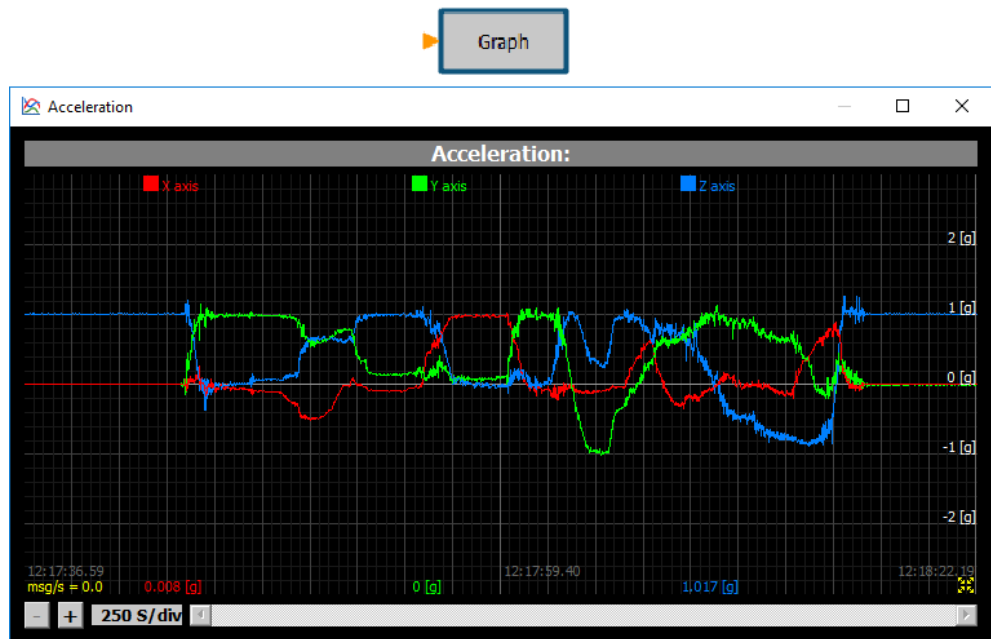
- 将显示库中的融合功能块添加到设计中，可以茶壶三维模型的形式显示四元数数据。三维茶壶模型通常用于检查器件在三维空间中的方向。此图需使用四元数，举例来说，可从传感器融合（MotionFX）算法获取四元数。

图 29. 融合功能块和 Unicleo-GUI 中的数据可视化示例



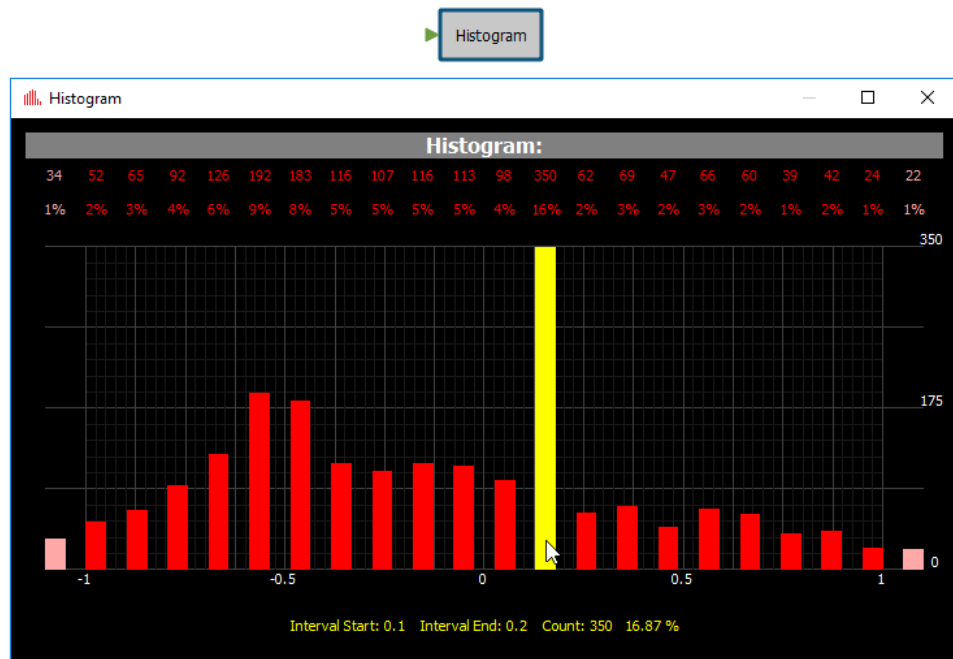
- 将显示库中的图形功能块添加到设计中，可以时间图的形式显示数据。
此图适用于任何浮点或整数值，每个图最多可包含 6 个波形。在属性字段中，您可以设置图和每个波形的名称、Y 轴上的单位、Y 轴上 0 的位置、满量程和自动缩放的使能或禁用。

图 30. 图形功能块和 Unicleo-GUI 中的数据可视化示例



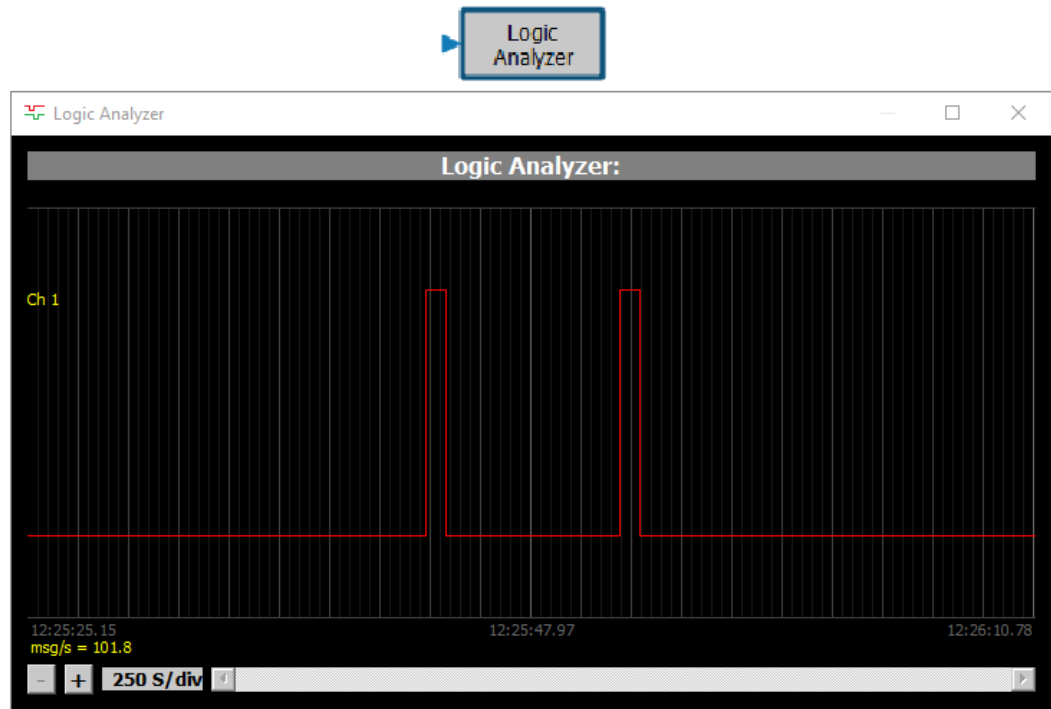
- 将显示库中的直方图功能块添加到设计中，可显示数据分布图。
直方图可用于查看选定值的分布情况。在属性字段中，您可以设置图的名称、间隔数、零轴位置、满量程和自动缩放的使能或禁用。

图 31. 直方图功能块和 Unicleo-GUI 中的数据可视化示例



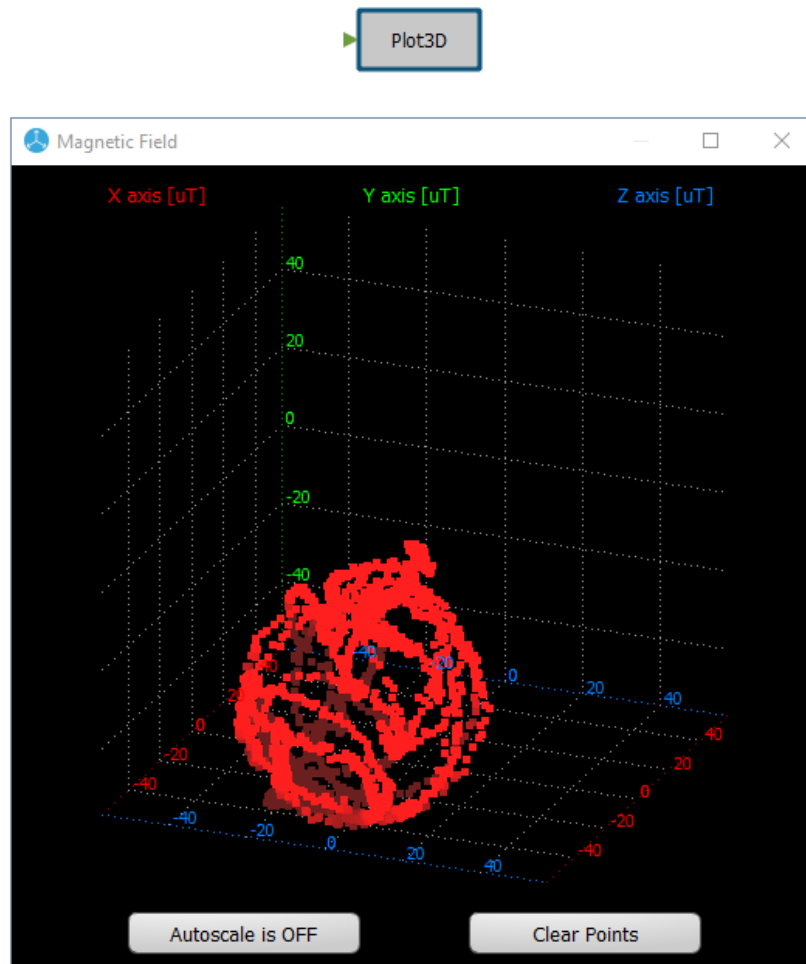
- 添加显示库中的逻辑分析仪功能块，可显示逻辑信号，逻辑信号的值只能是 0 或 1。
逻辑分析仪最多可以有 8 个通道。您可以在属性中修改每个通道的名称。

图 32. 逻辑分析仪功能块和 Unicleo-GUI 中的数据可视化示例



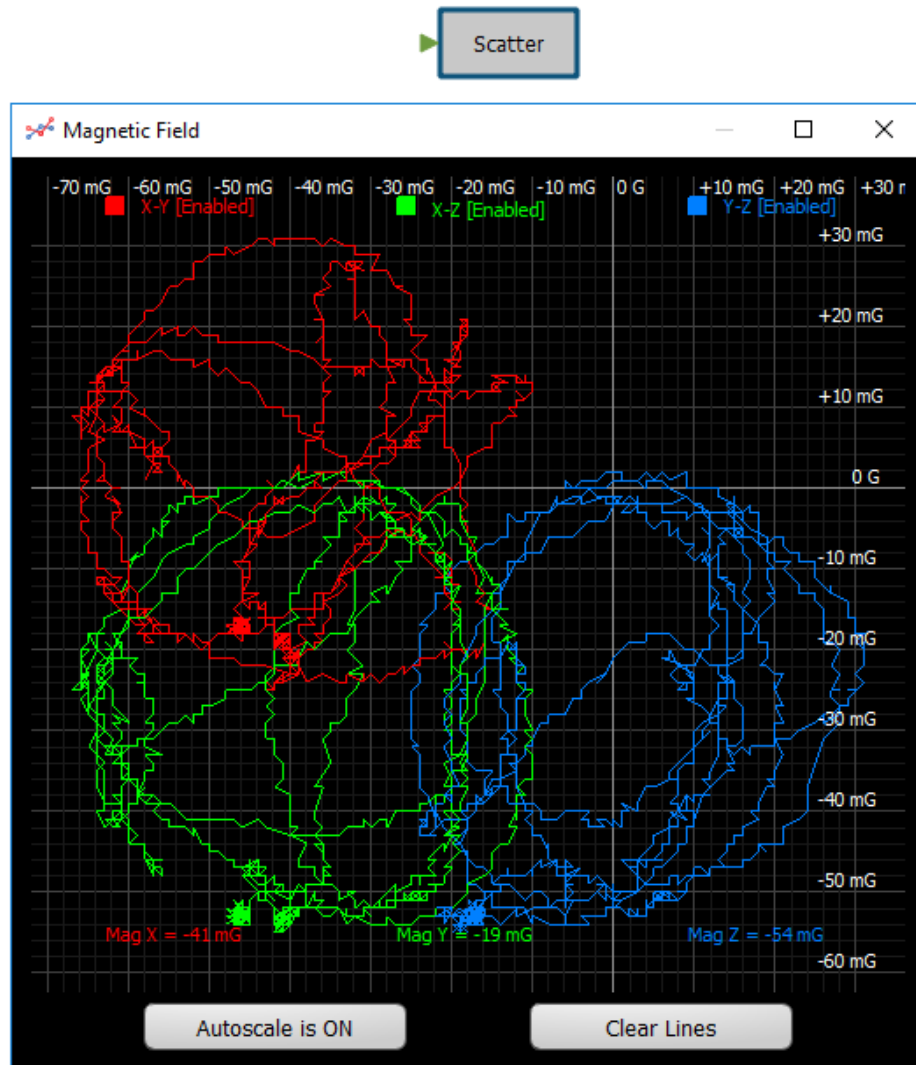
- 添加显示库中的三维绘图功能块，可在三维图中显示 X、Y 和 Z 数据。

图 33. 三维绘图功能块和 Unicleo-GUI 中的数据可视化示例



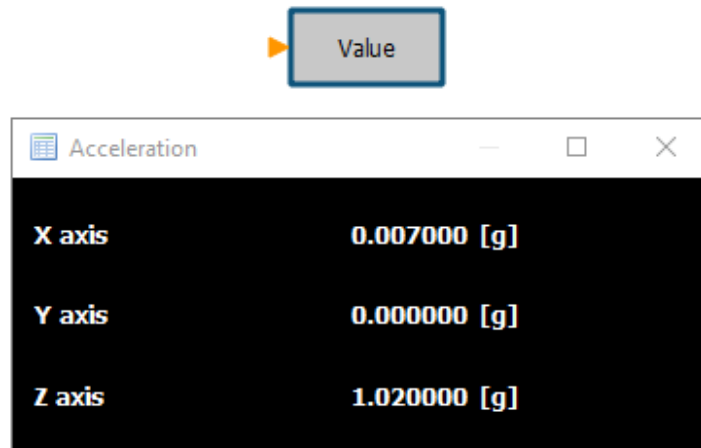
- 添加显示库中的散布图功能块，可在 X-Y 轴、X-Z 轴和 Y-Z 轴的二维图中显示 X、Y 和 Z 数据。

图 34. 散布图功能块和 Unicleo-GUI 中的数据可视化示例



- 添加显示库中的值功能块，可显示具体的浮点或整数值。
每个功能块最多可以显示 8 个值。在属性中，您可以修改值的名称和每一项的单位。

图 35. 值功能块和 Unicleo-GUI 中的数据可视化示例



9.2 数据输入

可以从 **Unicleo-GUI** 发送三种类型的数据到正在运行的固件：二进制、整型和浮点型。

将用户输入库中的输入值功能块（具有合适类型）添加到您的设计中。

每个输入值功能块最多可以显示 4 个值。在属性中，您可以设置每个值的名称和默认值。输入值功能块的输出规格由“值数量”属性定义。

图 36. 输入值功能块

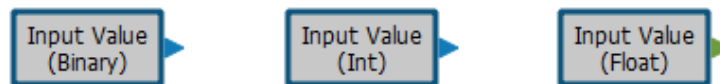
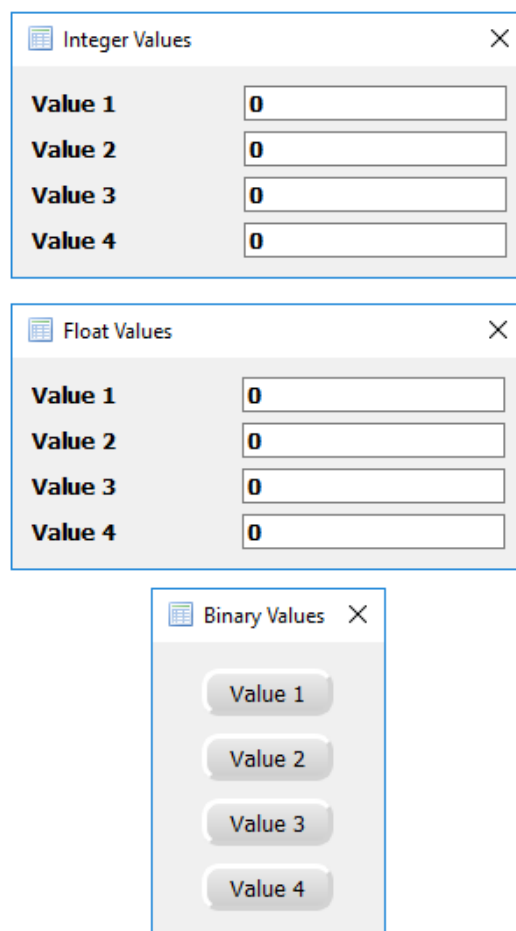


图 37. Unicleo-GUI 中的输入值



The figure displays three screenshots of the Unicleo-GUI configuration windows for input values. The first window, titled 'Integer Values', shows four input fields labeled 'Value 1', 'Value 2', 'Value 3', and 'Value 4', each containing the number '0'. The second window, titled 'Float Values', also shows four input fields labeled 'Value 1', 'Value 2', 'Value 3', and 'Value 4', each containing the number '0'. The third window, titled 'Binary Values', shows four buttons labeled 'Value 1', 'Value 2', 'Value 3', and 'Value 4'.



9.3 设置工具箱按钮顺序


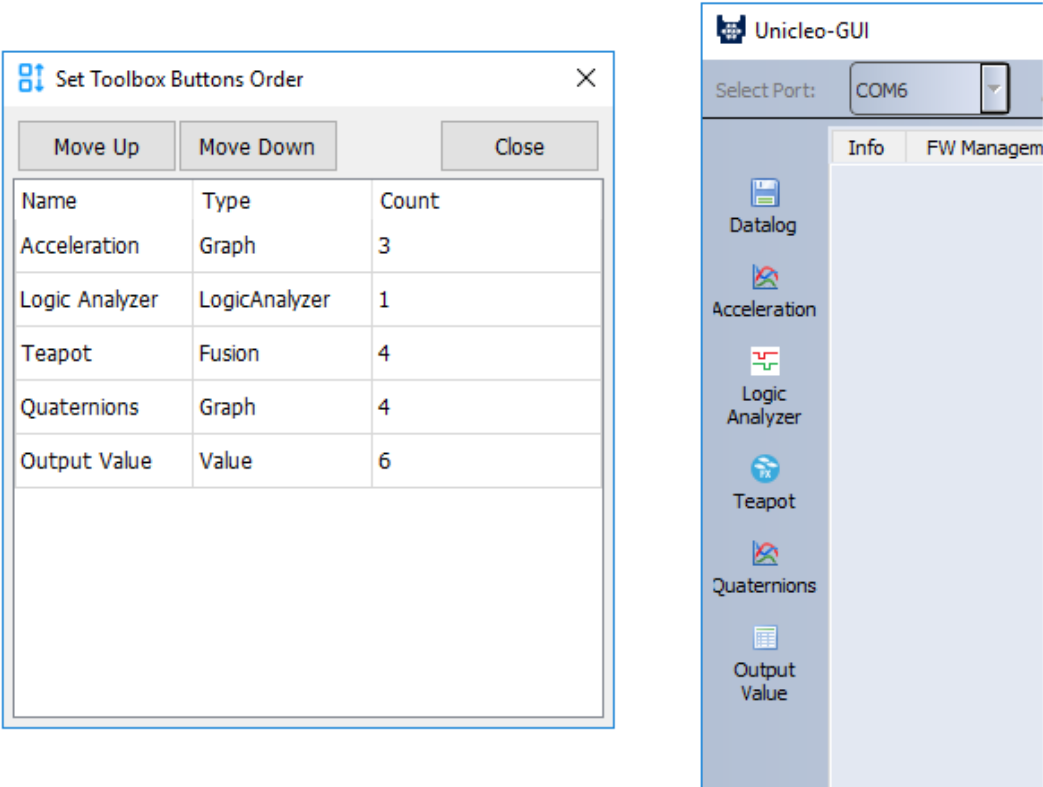
当 AlgoBuilder 设计中使用了很多显示和用户输入功能块时，Unicleo-GUI 上的工具箱按钮可能变得杂乱无章。通过点击工具栏或“固件”菜单中的图标（设置工具箱按钮顺序），可以对这些工具箱按钮进行整理。

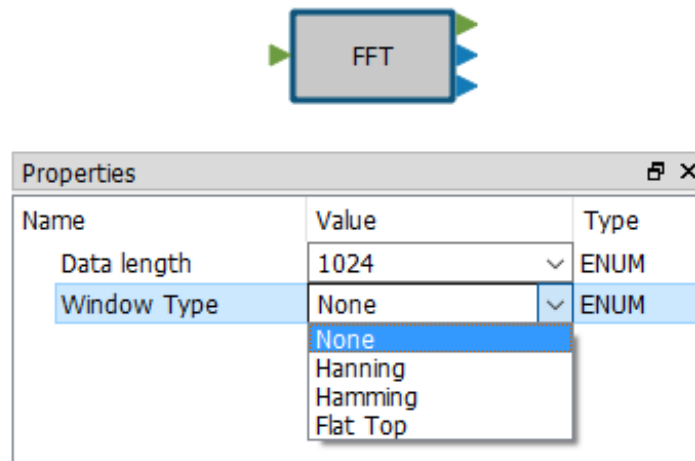
图 38. “设置工具箱按钮顺序”窗口



10 FFT（快速傅里叶变换）

AlgoBuilder 还提供了一个功能块，该功能块使用 FFT（快速傅里叶变换）进行传感器输出信号的频率分析。傅里叶分析将时域信号转换为频域信号。

图 39. FFT 功能块



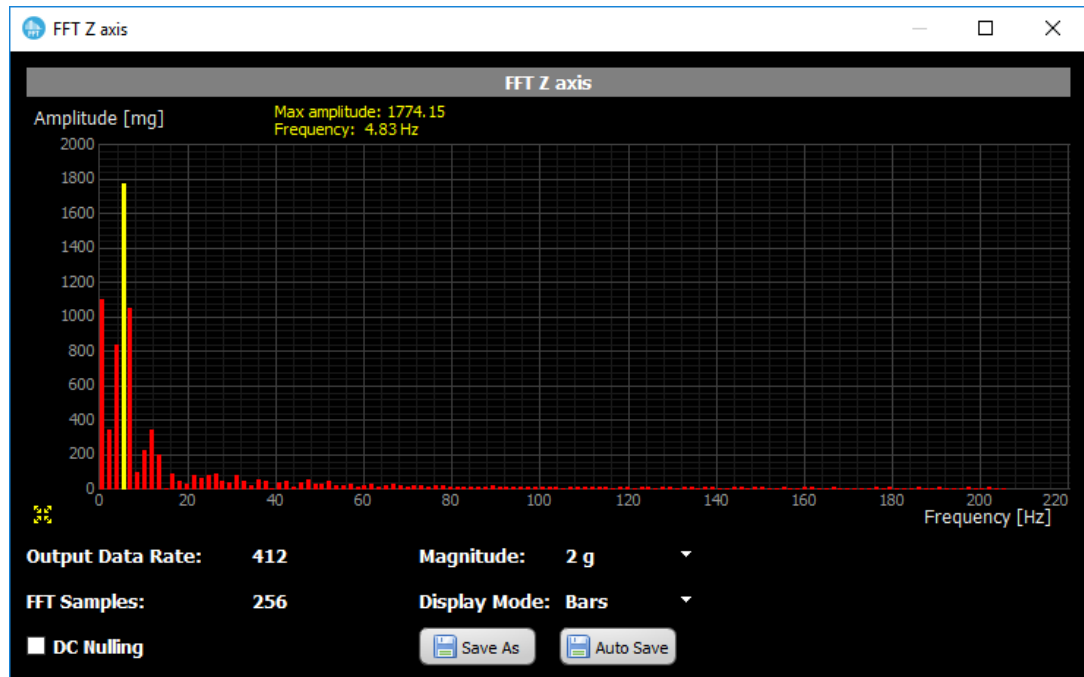
FFT 功能块提供 32、64、128、256、512 和 1024 个采样的频率分析。还可以利用窗函数来消除频谱泄漏，可以使用 Hanning、Hamming 和 Flat Top 窗。

FFT 功能块的输出可以连接到 FFT 绘图功能块，后者将结果发送至 Unicleo-GUI。Unicleo-GUI 随后将数据显示为频谱。为了只在 FFT 计算结束时向 Unicleo-GUI 发送数据，必须为 FFT 绘图添加有条件执行输入，并将该输入连接到 FFT 功能块。

图 40. FFT 和 FFT 绘图的连接



图 41. Unicleo-GUI 中的频谱



注：为了获取正确的频率值，必须选择将要在传感器集合的数据率控制器件中分析其数据的传感器。在固件初始化期间测量传感器的实际 ODR（输出数据率）。



11 创建您自己的功能块

在 XML 文件中，功能块被定义为 XML 记录。
每个 XML 文件代表一个 **AlgoBuilder** 库，其中可能包含一个或多个功能块。
库具有以下结构，在<Block></Block>标签中描述其中的每个功能块。

```
<?xml version="1.0" encoding="UTF-8"?>
<Library>
  <Version>1.0</Version>
  <Name>Comparison</Name>
  <Block>
  </Block>
</Library>
```

每个功能块必须具有下列 XML 标签。

表 4. 功能块 XML 标签

变量	说明
<Name></Name>	“库”面板的列表中显示的功能块名称。
<DisplayName></DisplayName>	放在工作区中的功能块矩形中显示的文本。在文本中，可使用两个井号（##）进行换行。
<Version></Version>	功能块的版本。
<Description></Description>	“说明”面板中显示的完整功能块功能的说明。
<Width></Width>	功能块矩形的宽度。
<Height></Height>	功能块矩形的高度。
<X></X>	功能块的 X 位置：必须为 0。
<Y></Y>	功能块的 Y 位置：必须为 0。
<Instances></Instances>	一个设计中特定功能块的最大数量。对于无限数量的实例，使用 0。
<Inputs></Inputs>	所有输入的列表。
<Input></Input>	包含<Type>、<Size>、<Name>和<Description>标签的输入定义。
<Type></Type>	特定输入的数据类型。可使用以下类型：INT、FLOAT 和 VARIANT
<Size></Size>	特定输入的数据大小。
<Name></Name>	特定输入的名称。
<Description></Description>	特定输入的说明。
<Outputs></Outputs>	所有输出的列表。
<Output></Output>	包含<Type>、<Size>、<Name>和<Description>标签的输出定义。
<Type></Type>	特定输出的数据类型。可使用以下类型：INT、FLOAT 和 VARIANT
<Size></Size>	特定输出的数据大小。
<Name></Name>	特定输出的名称。
<Description></Description>	特定输出的说明。
<Properties></Properties>	所有属性的列表。
<Property></Property>	包含<Type>、<Name>、<Value>和<Description>标签的属性定义。
<Type></Type>	特定属性的数据类型。可使用以下类型：INT、FLOAT 和 STRING



变量	说明
<Name></Name>	特定属性的名称。
<Value></Value>	特定属性的默认值。
<Description></Description>	特定属性的说明。
<Memory></Memory>	所有存储器项目的列表。
<Item></Item>	存储器项目的定义。
<Type></Type>	特定存储器项目的数据类型。可使用以下类型：INT 和 FLOAT
<Size></Size>	特定存储器项目的数据大小。
<Name></Name>	特定存储器项目的名称。
<Command></Command>	代表功能块的 C 语言指令。每个指令必须另起一行，并放在<Line></Line>标签之间。指令中可以使用任何输入、输出或属性的值。为此，在输入、输出或属性的名称之前和之后使用%。
<Line></Line>	
<Function></Function>	可以定义要在<Command>标签中使用的完整函数。每个指令必须另起一行，并放在<Line></Line>标签之间。
<Line></Line>	
<InitCode></InitCode>	在固件初始化期间执行一次的 C 语言指令。

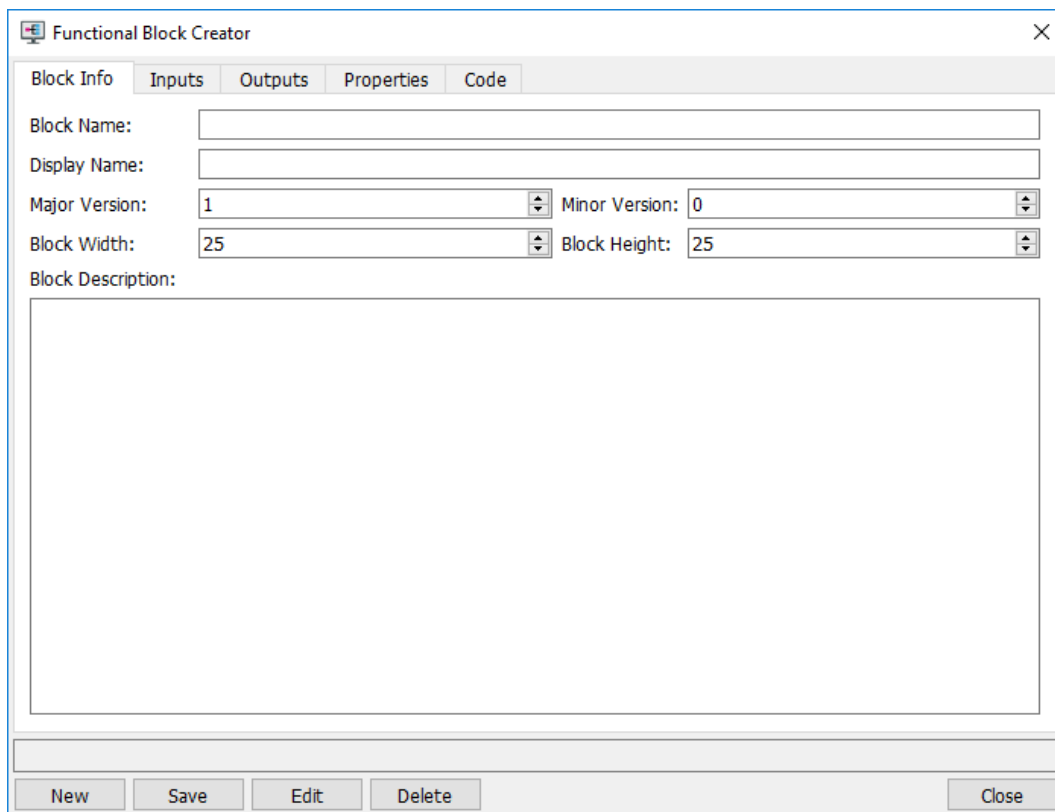
11.1 功能块创建器

AlgoBuidler 提供了一个功能块创建器，用户无需掌握 XML 标签的相关知识即可定义功能块。可以从“工具”菜单或从工具栏按下按钮 （功能块创建器）来执行功能块创建器。

用户定义的功能块保存在库中，库保存在用户的根目录\STMicroelectronics\AlgoBuilder\Library 中。

功能块创建器还提供编辑或删除现有功能块的能力。

图 42. 功能块创建器



Functional Block Creator

Block Info Inputs Outputs Properties Code

Block Name:

Display Name:

Major Version: Minor Version:

Block Width: Block Height:

Block Description:

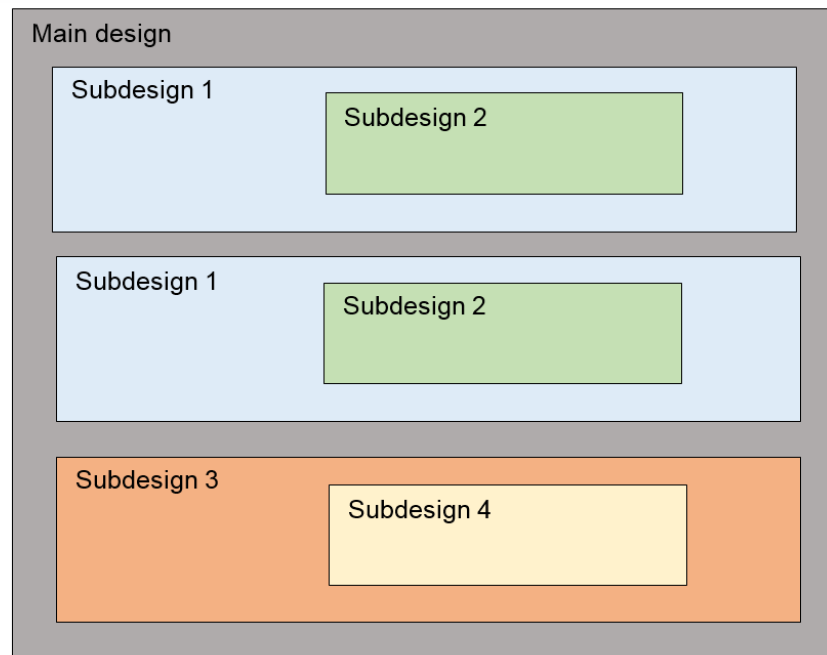
New Save Edit Delete Close

在代码部分，可以输入代表功能块功能的 C 代码。在 C 代码中，使用 %NAME%[index] 引用输入、输出和属性，其中的 NAME 为输入、输出或属性的名称，index 为项目在队列中的编号。用队列表示所有输入和输出，因此必须使用 index，即使队列中只有一个项目。属性是标量，因此不使用 index。

12 子设计

AlgoBuilder 具备创建多级设计的能力。此特性允许将设计的一部分封装成一个独立部分，称为子设计。此子设计用单个功能块表示，可以在主设计和所有其他设计中多次使用。用户之间可以共享子设计。子设计显著改善了高度复杂的项目的条理性。

图 43. 多级设计的原理

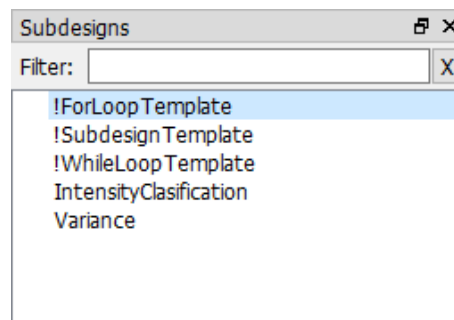


子设计允许实现循环。

AlgoBuilder 默认提供三个模板用于子设计的创建：

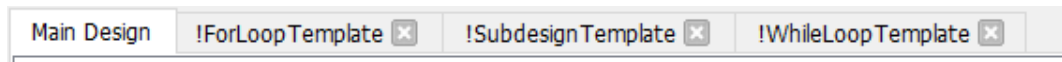
- **!Subdesign Template** 是通用的空的子设计
- **!ForLoopTemplate** 是用于循环的子设计，在进入循环前已知迭代次数时使用
- **!WhileLoopTemplate** 是用于循环的子设计，在定义的条件有效前执行

图 44. 子设计面板



通过双击特定模板，可以创建子设计。子设计面板上列出的或设计中使用的现有子设计可以用相同方式打开。如果打开了新的或现有子设计，**AlgoBuilder** 将在工作区创建专用选项卡。

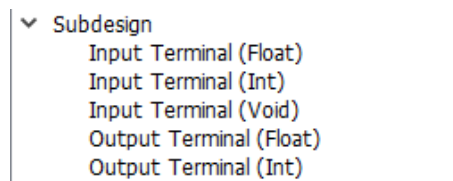
图 45. 主设计和子设计选项卡



通过点击图标 (保存设计) 或 (设计另存为)，可以用与主设计相同的方式保存子设计。然后，可以在主设计或其他子设计中使用该子设计。为了使用子设计，可以将特定项目从子设计面板拖放到工作区。

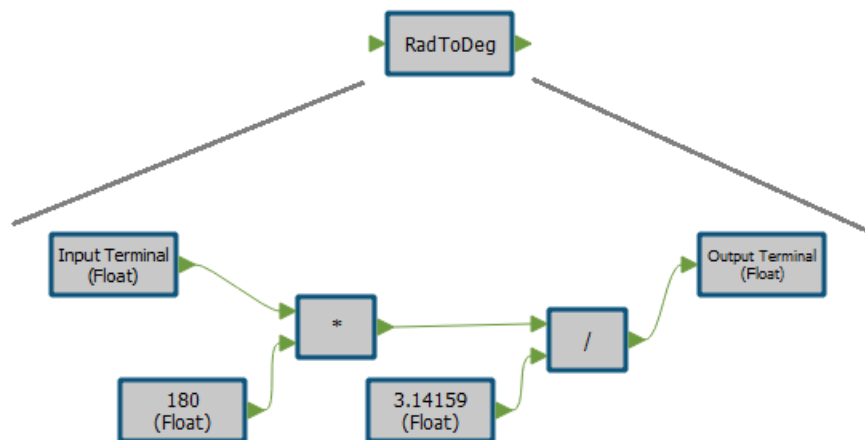
为了能够连接子设计与其他功能块，必须为每个子设计定义输入和输出端子。可以从子设计库添加端子，端子只在修改子设计时激活。

图 46. 子设计端子



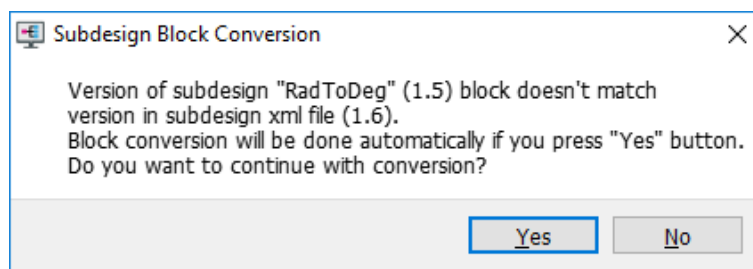
每个端子有三个属性：规格、名称和说明。必须定义规格和名称，此外名称在该子设计中必须是唯一的。说明为选填项。在图形设计中，子设计用相同的矩形表示，具有功能块作为输入和输出。输入和输出的垂直顺序由子设计内部的端子纵坐标给出。

图 47. 子设计功能块及其内容的示例



如果用户修改了特定子设计，AlgoBuilder 能够检测到它，并为子设计的每个实例提供最新版本的更新。在主设计的子设计验证或 C 代码生成期间执行这项检查。

图 48. 子设计更新



注：子设计禁止出现下列两种情况：

1. 子设计不应嵌入其自身，否则会构成递归调用和无限循环。通过检查子设计名称来防止这种情况。

2. 允许将子设计 2 嵌入子设计 1（请参考图 43），但不应将子设计 1 进一步嵌入子例程。将在设计确认期间执行此项检查，将结果通知用户并终止代码生成。

每个子设计都保存在单独的 XML 文件中。

for 和 while 循环的模板包含多个强制使用的功能块。这些功能块无法删除。

强制功能块如下：

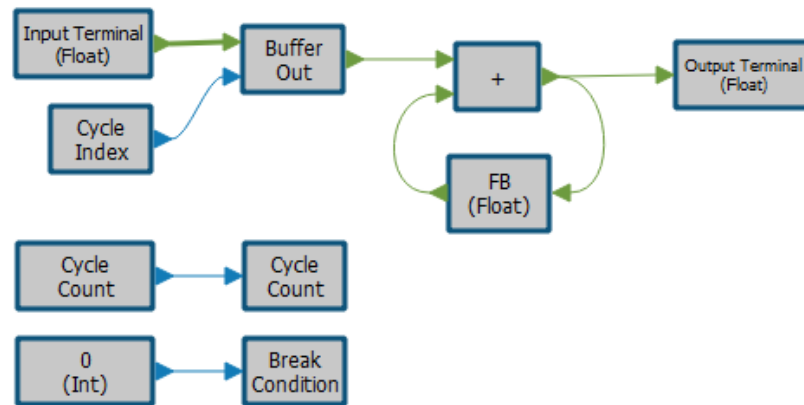
For 循环：

- Cycle Count（输入）... 定义将执行的迭代次数
- Cycle Index（输出）... 返回当前迭代次数
- Break Condition（输入）... 允许在达到循环计数前终止循环的执行

While 循环：

- Cycle Index（输出）... 返回当前迭代次数
- Break Condition（输入）... 终止循环的执行

图 49. 循环示例（计算输入缓冲区中所有项目的和）



13 输入数据和输出数据监控器与 AWS 连接

AlgoBuilder 能够快速检查正在运行的固件的输出，无需执行 Unicleo-GUI。它还允许使用离线数据和将输出数据发布至 AWS 云。通过点击工具栏或“工具”菜单中的图标 （输入数据和输出数据及 AWS 连接）可以打开一个专用窗口，用户可以在该窗口中使用所有这些功能。

13.1 输出数据监控器

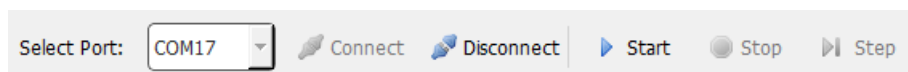
使用输出数据监控器，可以查看文本形式的输出数据。输出数据监控器窗口具有自己的用于控制功能的工具栏。“连接” 和“断开” 按钮可用于建立或关闭 AlgoBuilder 与连接的 STM32 器件之间的连接。“开始” 和“停止” 按钮控制从连接的器件到 AlgoBuilder 的数据流传输。输出数据显示在“输出数据”选项卡上。

图 50. 输入数据和输出数据监控器与 AWS 连接

Time Stamp [h:min:s]	X axis [g]	Y axis [g]	Z axis [g]	AND	q1	q2	q3	q4	q1	q2	q3	q4	Magnetic Field [uT]	Temperat [C]
0:01:15.78	0.01200	-0.009000	1.031	0.0000	0.001300	-0.006700	-0.6760	0.7369	0.001300	-0.006700	-0.6760	0.7369	49.50	25.78
0:01:15.80	0.01200	-0.008000	1.031	0.0000	0.001300	-0.006700	-0.6760	0.7369	0.001300	-0.006700	-0.6760	0.7369	50.09	25.78
0:01:15.80	0.01100	-0.008000	1.032	0.0000	0.001300	-0.006700	-0.6760	0.7369	0.001300	-0.006700	-0.6760	0.7369	49.47	25.74
0:01:15.82	0.01200	-0.01000	1.031	0.0000	0.001300	-0.006700	-0.6760	0.7369	0.001300	-0.006700	-0.6760	0.7369	49.17	25.74
0:01:15.83	0.01100	-0.009000	1.030	0.0000	0.001300	-0.006700	-0.6760	0.7369	0.001300	-0.006700	-0.6760	0.7369	49.41	25.74
0:01:15.83	0.01100	-0.008000	1.030	0.0000	0.001300	-0.006800	-0.6760	0.7368	0.001300	-0.006800	-0.6760	0.7368	49.90	25.74
0:01:15.85	0.01100	-0.008000	1.029	0.0000	0.001300	-0.006800	-0.6760	0.7368	0.001300	-0.006800	-0.6760	0.7368	49.32	25.74
0:01:15.85	0.01200	-0.009000	1.030	0.0000	0.001300	-0.006800	-0.6760	0.7368	0.001300	-0.006800	-0.6760	0.7368	50.16	25.74
0:01:15.87	0.01100	-0.009000	1.032	0.0000	0.001300	-0.006700	-0.6760	0.7369	0.001300	-0.006700	-0.6760	0.7369	49.29	25.74
0:01:15.87	0.01100	-0.008000	1.031	0.0000	0.001300	-0.006800	-0.6760	0.7368	0.001300	-0.006800	-0.6760	0.7368	49.24	25.74
0:01:15.89	0.01100	-0.008000	1.030	0.0000	0.001300	-0.006800	-0.6760	0.7368	0.001300	-0.006800	-0.6760	0.7368	49.49	25.74
0:01:15.89	0.01100	-0.009000	1.029	0.0000	0.001300	-0.006700	-0.6760	0.7369	0.001300	-0.006700	-0.6760	0.7369	49.68	25.74
0:01:15.91	0.01000	-0.009000	1.031	0.0000	0.001300	-0.006700	-0.6760	0.7369	0.001300	-0.006700	-0.6760	0.7369	49.23	25.74

FW Info: Example01_Demo.xml; FW_Mode: On-line; FW_ODR: 100

图 51. 输出数据控制部分

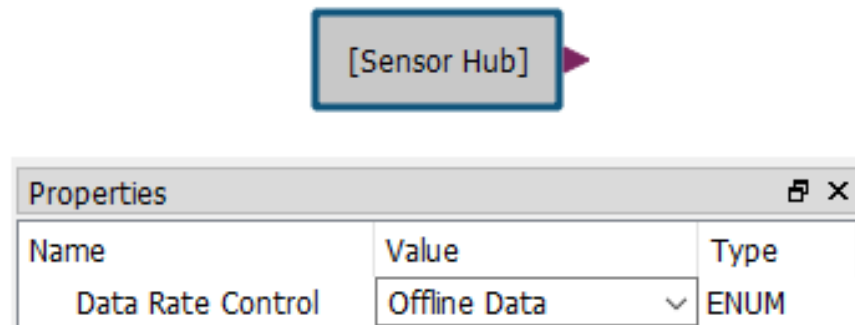


13.2 离线数据处理

AlgoBuilder 可以将之前记录的数据发送至连接的 STM32 器件，该器件将处理这些数据，而不是来自传感器的实际数据。这可以用于调试总是使用相同输入数据作为算法激励的算法。可以使用 Unicleo-GUI 中的 Datalog 功能记录和保存数据。

为了使用之前记录的数据，必须在“传感器集合”功能块的“数据率控制”属性中选中“离线数据”选项。

图 52. 传感器集合中的离线数据设置



通过点击图标 （打开），可以加载输入数据。如果以高于传感器集合要求的数据率保存文件中的数据，



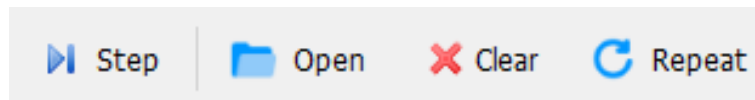

AlgoBuilder 将自动对该数据进行重新采样。如果点击图标 （重复），将以无限循环重复数据。可以自动处理所有数据或通过点击图标 （步骤）逐一处理数据。输入数据和相应的输出数据显示在“输入数据”和“输出数据”选项卡中。

图 53. 输入数据控制部分



13.3 AWS (Amazon 网络服务) 连接

AlgoBuilder 还允许向 AWS 云 (Amazon 网络服务)，准确地说是 IoT Core Service 或任何 MQTT Broker 发送数据。为了设置连接参数，点击图标  (AWS 连接选项) 并输入所有需要的参数。



AlgoBuilder 支持三种不同接口 (数据结构)：

- 通用 MQTT：数据采用可以被任何其他系统使用的通用 JSON 格式
- 仪表板：数据采用便于与 ST 仪表板协作的结构
- 共生：数据采用便于与 ST 内部工具协作的结构

可基于选定接口调整连接参数。必填参数如下：

- 通用 MQTT：配置名称、主题、Broker 地址和证书
- 仪表板：器件名称、MAC 地址、仪表板地址和证书
- 共生：事物、名称、端点和证书

其他参数为选填。

如果参数设置正确，可通过点击图标  (开始 AWS 通信) 与 AWS 云建立连接。如果连接已建立，红点将变为绿色。通过点击同一图标，可以断开连接。发布周期定义了向云发送数据的频率。通过点击图标  (记录向 AWS 发布的消息)，可以将发送至云的数据保存为用户本地硬盘上的文件。这些数据记录位于用户目录 \STMicroelectronics\AlgoBuilder\DataLogs 中。

可以对监控器上的数据进行筛选，以便只显示发送至云的数据，点击图标  (只显示向 AWS 发布的消息) 可使能此筛选功能。

图 54. AWS 控制部分

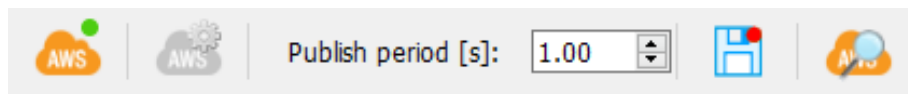


图 55. AWS 连接选项 (ST 仪表板接口)

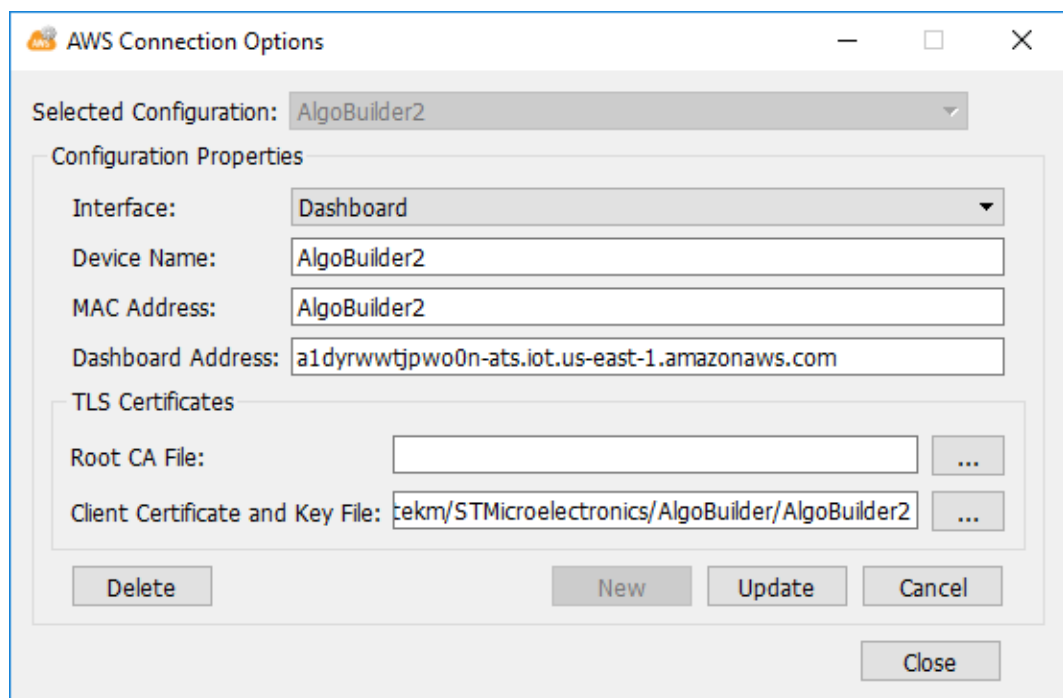
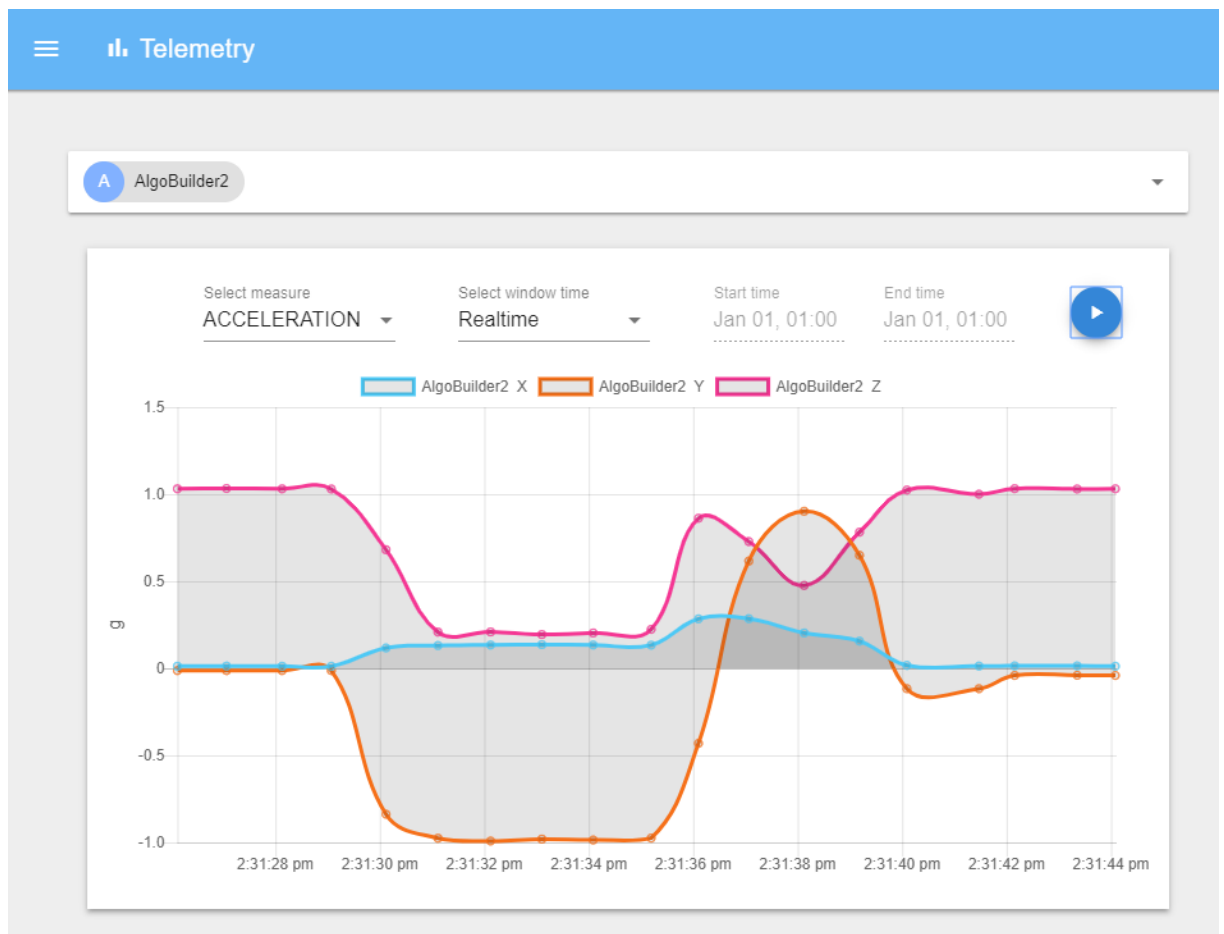


图 56. ST AWS 仪表板



版本历史

表 5. 文档版本历史

日期	版本	变更
2018 年 3 月 26 日	1	初始版本
2018 年 7 月 25 日	2	<p>更新了“图 1. AlgoBuilder 应用程序示意图”、“图 3. AlgoBuilder 设计”和“图 7. AlgoBuilder 安装程序”</p> <p>在 1.1 节“概述”和 1.2 节“必备条件”中增加了 SensorTile STEVAL-STLKT01V1</p> <p>更新了 2.3 节“应用设置”和图 10。应用设置窗口</p> <p>更新了图 13。AlgoBuilder 主窗口</p> <p>更新了 3.2 节“库面板”</p> <p>更新了表 1。AlgoBuilder 工具栏默认功能</p> <p>更新了表 2。AlgoBuilder 的库</p> <p>更新了第 6 章“创建您的第一个设计”</p> <p>更新了第 7 章“目标编程”</p> <p>更新了 7.2 节“SensorTile”</p> <p>更新了第 8 章“使用 Unicleo-GUI”</p> <p>增加了 9.1 节“功能块创建器”</p>
2018 年 9 月 27 日	3	<p>更新了图 7。AlgoBuilder 安装程序</p> <p>更新了图 13。AlgoBuilder 主窗口</p> <p>增加了图 14。使用反馈功能块</p> <p>增加了第 5 节。有条件执行</p> <p>更新了表 2。AlgoBuilder 的库</p> <p>更新了图 18。传感器集合属性和图 22。algo_builder.c 文件中生成的代码</p> <p>增加了第 10 节。FFT（快速傅里叶变换）</p> <p>少量文本更新</p>
2018 年 12 月 4 日	4	<p>增加了子设计特性：</p> <p>更新了图 7。AlgoBuilder 安装程序</p> <p>更新了图 13。AlgoBuilder 主窗口</p> <p>增加了 3.3 节“子设计面板”</p> <p>更新了“表 1. AlgoBuilder 工具栏默认功能”和“表 2. AlgoBuilder 的库”</p> <p>增加了第 12 章“子设计”</p>
2019 年 4 月 5 日	5	<p>增加了 X-NUCLEO-IKS01A3 扩展板</p> <p>更新了 1.1 节“概述”</p> <p>更新了“图 7. AlgoBuilder 安装程序”</p> <p>更新了 2.3 节“应用设置”</p> <p>更新了第 3 章“使用 AlgoBuilder”</p> <p>更新了表 1。AlgoBuilder 工具箱默认功能</p> <p>更新了 8.2 节“SensorTile”</p> <p>增加了 9.3 节“设置工具箱按钮顺序”</p> <p>更新了 11.1 节“功能块创建器”</p> <p>增加了第 13 章“输出数据监控器与 AWS 连接”</p>

日期	版本	变更
2019 年 6 月 18 日	6	增加了 SensorTile.box STEVAL-MKSBOX1V1 更新了 2.3 节“应用设置” 更新了表 3。AlgoBuilder 支持的库 增加了 8.3 节“SensorTile.box” 增加了 13.1 节“输出数据监控器” 增加了 13.2 节“离线数据处理” 增加了 13.3 节“AWS（Amazon 网络服务）连接”
2019 年 8 月 19 日	7	增加了 STM32CubeIDE 编译器（1.1 节“概述”和 1.2 节“必备条件”） 更新了 2.3 节“应用设置” 更新了表 1。AlgoBuilder 工具栏默认功能 更新了第 7 章“创建您的第一个设计” 更新了 8.1 节“STM32 Nucleo 板” 更新了 8.3 节“SensorTile.box”
2019 年 10 月 24 日	8	在整个用户手册中用 STM32CubeProgrammer 替换了 STSW-LINK004 更新了 第 1.2 节 先决条件



目录

1	说明	2
1.1	概述	2
1.2	先决条件	2
1.3	术语和参考文献	3
1.4	工作原理	5
2	入门指南	6
2.1	安装软件	6
2.2	首次运行软件	6
2.3	应用设置	7
2.4	网络更新设置	8
3	使用 AlgoBuilder	9
3.1	工作区	10
3.2	库面板	10
3.3	子设计面板	10
3.4	说明面板	10
3.5	属性面板	11
3.6	工具栏	12
4	数据类型	14
5	有条件执行	15
6	库	16
7	创建您的第一个设计	17
8	目标编程	20
8.1	STM32 Nucleo 板	20
8.2	SensorTile	20
8.3	SensorTile.box	21
9	使用 Unicleo-GUI	23
9.1	数据可视化	23
9.2	数据输入	30
9.3	设置工具箱按钮顺序	31



10	FFT（快速傅里叶变换）	32
11	创建您自己的功能块	34
11.1	功能块创建器	36
12	子设计	37
13	输入数据和输出数据监控器与 AWS 连接	40
13.1	输出数据监控器	40
13.2	离线数据处理	41
13.3	AWS（Amazon 网络服务）连接	42
	版本历史	44



表一览

表 1.	AlgoBuilder 工具栏默认功能	12
表 2.	AlgoBuilder 的库	16
表 3.	AlgoBuilder 支持的库	16
表 4.	功能块 XML 标签	34
表 5.	文档版本历史	44

图一览

图 1.	AlgoBuilder 应用程序框图	1
图 2.	AlgoBuilder 功能块	3
图 3.	AlgoBuilder 设计	3
图 4.	STM32 Nucleo (NUCLEO-F401RE) + X-NUCLEO-IKS01A2	3
图 5.	Unicleo-GUI	4
图 6.	AlgoBuilder 的工作原理	5
图 7.	AlgoBuilder 安装程序	6
图 8.	AlgoBuilder 图标	6
图 9.	应用设置的菜单选项	7
图 10.	应用设置窗口	7
图 11.	网络更新设置的菜单选项	8
图 12.	网络更新设置窗口	8
图 13.	AlgoBuilder 主窗口	9
图 14.	使用反馈功能块	14
图 15.	有条件执行	15
图 16.	AlgoBuilder 固件设置窗口	17
图 17.	传感器集合功能块	17
图 18.	传感器集合的属性	17
图 19.	传感器集合、加速度[g]和图形功能块	17
图 20.	图形属性	18
图 21.	传感器集合、加速度[g]和图形功能块	18
图 22.	algo_builder.c 文件中生成的代码	19
图 23.	System Workbench for STM32 的输出	19
图 24.	编程目标图标和选择框	20
图 25.	STM32 Nucleo 板和 SensorTile cradle SWD 连接器	20
图 26.	STM32ST-Link 编程器和 SensorTile.box 互连	21
图 27.	在 DFU 模式下进行 SensorTile.box 编程	22
图 28.	柱状图功能块和 Unicleo-GUI 中数据可视化的示例	23
图 29.	融合功能块和 Unicleo-GUI 中的数据可视化示例	24
图 30.	图形功能块和 Unicleo-GUI 中的数据可视化示例	25
图 31.	直方图功能块和 Unicleo-GUI 中的数据可视化示例	25
图 32.	逻辑分析仪功能块和 Unicleo-GUI 中的数据可视化示例	26
图 33.	三维绘图功能块和 Unicleo-GUI 中的数据可视化示例	27
图 34.	散布图功能块和 Unicleo-GUI 中的数据可视化示例	28
图 35.	值功能块和 Unicleo-GUI 中的数据可视化示例	29
图 36.	输入值功能块	30
图 37.	Unicleo-GUI 中的输入值	30
图 38.	“设置工具箱按钮顺序”窗口	31
图 39.	FFT 功能块	32
图 40.	FFT 和 FFT 绘图的连接	32
图 41.	Unicleo-GUI 中的频谱	33
图 42.	功能块创建器	36
图 43.	多级设计的原理	37
图 44.	子设计面板	37
图 45.	主设计和子设计选项卡	38
图 46.	子设计端子	38
图 47.	子设计功能块及其内容的示例	38
图 48.	子设计更新	38
图 49.	循环示例（计算输入缓冲区中所有项目的和）	39
图 50.	输入数据和输出数据监控器与 AWS 连接	40
图 51.	输出数据控制部分	40
图 52.	传感器集合中的离线数据设置	41



图 53.	输入数据控制部分	41
图 54.	AWS 控制部分	42
图 55.	AWS 连接选项 (ST 仪表板接口)	42
图 56.	ST AWS 仪表板	43



重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 标志是 ST 的商标。关于 ST 商标的其他信息，请访问 www.st.com/trademarks。其他所有产品或服务名称是其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2019 STMicroelectronics - 保留所有权利