



TM Palletizing Operator Kit

使用说明书

翻译自原有指示

产品版本：1.0

文件版本：1.0

发布日期：2021-10-05

本文记述 Techman Robot 机器人产品系列 (以下简称 TM Robot) 信息 , 所有信息属于达明机器人 (股) 公司 (以下简称本公司) 财产 , 未经本公司事先授权不得以任何形式或方式转载及复制任何数据。本文任何信息不应视为任何要约或是承诺 , 日后如有变更 , 恕不另行通知。本说明书应定期审查 , 本公司不会对本文任何错误或是遗漏承担责任。

TM 标志为达明机器人 (股) 公司于台湾与其他国家地区之注册商标 , 本公司保留本说明书及其拷贝的所有权及其著作权。

修订表

Revision Code	Date	Description
1.0	2021-10-05	Official release

修订表	3
1. 总章	7
1.1 概说	7
1.2 如何得到协助？	7
1.3 安全警示符号	7
1.4 安全防范	8
1.5 确效与责任	8
1.6 责任限制	9
1.7 功能注记符号	9
2. 应用软件设定	10
2.1 TM Palletizing Operator Kit Software Package 简介	10
2.2 TM Palletizing Operator Software Configuration Workflow	10
2.3 TMstudio	10
2.3.1 TMstudio 堆栈精灵及场景建立器、功能及限制	10
2.3.1.1 步骤 1 – 汇入 CAD 檔	13
2.3.1.2 步骤 2 – 汇入 Pillar Base	13
2.3.1.3 步骤 3 – 设定 Pillar Base 参数	14
2.3.1.4 步骤 4 – 汇入手臂	15
2.3.1.5 步骤 5 – 汇入工具	16
2.3.1.6 步骤 6 – 插入堆栈并定位	17
2.3.1.7 步骤 7 - 更改栈板长出方向	19
2.3.1.8 步骤 8 – 设定母子关系	20
2.3.1.9 步骤 9 – 开始堆栈	21
2.3.2 在堆栈精灵设堆栈应用	22
2.3.2.1 更换夹爪	22
2.3.2.2 选择堆栈类型	22
2.3.2.3 开始堆栈设定	23
2.3.2.4 步骤 1 - 设置栈板大小和边界	23
2.3.2.5 步骤 2 - 设置箱体大小和标签方向	24
2.3.2.6 步骤 3 -建立层及摆放方式	25
2.3.2.7 步骤 4 -建立层的放置顺序	29
2.3.2.8 复制箱体	31
2.3.2.9 模拟页面 – 参数设定	32
2.3.2.10 验证	34

2.3.2.11 模拟.....	34
2.3.2.12 汇出	35
2.3.2.13 点位档格式说明.....	35
2.3.3 将项目设定为堆栈精灵预设项目	37
2.4 TMflow.....	39
2.4.1 汇入专案与 Text File 至 TMflow.....	39
2.4.2 项目停止/错误时对应之 IO 设定	42
2.4.3 更换立柱 TMflow 设定.....	42
2.4.3.1 限制量测结果	43
2.4.3.2 量测到伸长量	44
2.4.4 设定 TMOperator_TMROBOT_TMPLTZOP_V002_Pillar	44
2.4.4.1 降低立柱高度	47
2.4.5 设定 TMOperator_TMROBOT_TMPLTZOP_V002_Main	48
2.4.5.1 Step 1 – Import File	49
2.4.5.2 Step 2 – Recognizing Box Position	50
2.4.5.3 Step 3 – Teach Gripping Point	52
2.4.5.4 Step 4 – Visual Recognition.....	54
2.4.5.5 传感器输入注意事项.....	61
2.4.6 设定 TMOperator_TMROBOT_TMPLTZOP_V002_Grip 与 TMOperator_TMROBOT_TMPLTZOP_V002_Release	61
2.5 进阶堆栈设定	61
2.5.1 Advanced1_General_Parameters	62
2.5.2 Advanced2-8	64
2.5.3 TMOperator_TMROBOT_TMPLTZOP_V002_Main 项目中其他副流程及同步线程	65
2.5.3.1 Subflow Pallet_start.....	66
2.5.3.2 副流程 function.....	66
2.5.3.3 副流程 take	67
2.5.3.4 副流程 flow_in_take	68
2.5.3.5 副流程 flow_in_put.....	71
2.5.3.6 线程 thread_judge.....	72
2.5.3.7 线程 nonpause.....	73
2.5.3.8 副流程(SF)之间关系.....	74
2.5.4 TMOperator_TMROBOT_TMPLTZOP_V002_Main 项目中其他参数、点位、坐标系.....	76
2.5.5 TMOperator_TMROBOT_TMPLTZOP_V002_Pillar 专案中参数	78
2.5.6 TMOperator_TMROBOT_TMPLTZOP_V002_Grip 项目中变数	78
2.5.7 TMOperator_TMROBOT_TMPLTZOP_V002_Release 项目中变数.....	78

2.6 文件档案叙述	79
2.7 Base and Point 计算	81

1. 总章

1.1 概说

TM Palletizing Operator Kit 为专为堆栈应用设计的解决方案，支持以下 5 种不同的堆/拆栈模式：

- 相同箱体堆栈
- 不同箱体堆栈：同时处理两种不同箱体的堆栈任务
- 相同箱体拆栈：将栈板上的箱体做拆栈处理
- 不同箱体拆栈：将不同栈板上不同的箱体做拆栈处理
- 相同箱体更换栈板重新堆栈：将箱体从 A 栈板换到 B 栈板做堆栈，两个栈板可为不同规格

TM Palletizing Operator Kit 软件部分包含 TMflow 以及 TMstudio，TMflow 为手臂流程编程的软件，搭配 TM Palletizing Operator 提供的 TMflow 项目档，以及 TMstudio 针对 Palletizing 可自行布置 Palletizing 场景以及仿真 Palletizing 的功能，让用户可自行整合相关硬件，以建构不同手臂安装悬挂姿态，搭配 TM Robot，快速打造定制化的 Palletizing Operator。



重要：

由于 TMstudio 的 Palletizing Wizard 其产出之 Palletizing 点位档已带 TM Robot 基座与栈板坐标系之关系，可直接于范例项目利用此坐标关系。



重要：

以上关于 TMstudio Palletizing Wizard 以及 TMflow 项目中各点位及其运动种类之适用性，使用者仍需视状况自行调整以应用于自行设计之堆栈应用单元。

1.2 如何得到协助？

使用者可至敝公司网站以获得更多资讯：

<http://tm-robot.com/>

TM Robot 的用户、系统整合者于使用本机器人前必须先行详尽阅读并充分了解本章内容。此外，当使用者遵照本说明书进行机器人之任何操作前，需先行阅读并遵照产品软硬件版本对应之安全说明书，以及对应硬件版本之硬件设置说明书后，始能进行操作。

1.3 安全警示符号

下表定义本说明书中于各段落标示的安全警示层级符号，请于阅读各段落时详加注意并遵从，以免发生对人身或设备之危害。

**危险：**

此标记符号表示紧急之危险情况，如果不加以避免，将导致严重伤害，并可能导致死亡或严重财产损失。

**警告：**

此标记符号表示潜在的危险情况，如果不加以避免，将导致轻度或中度伤害，并可能导致严重伤害，死亡或重大财产损失。

**注意：**


此标记符号表示潜在的危险情况，如果不加以避免，可能会导致轻微伤害，中度伤害或财产损失。

安全警示符号**1.4 安全防范****危险：**

若不依循以下安全防范，错误的使用本产品可能将造成机体或其它设备损伤，或人员的严重受伤或致死。

- 所有安装，操作，教导，编辑或维护本产品之人员必须详阅并遵循对应软硬件版本之 *硬件设置说明书*，*软件说明书*，以及 *安全说明书*，并且受过本产品之充分训练。

**阅读说明书标示****撞击警示标示**

- 所有使用本产品设计机器人系统之人员应详阅并遵循对应软硬件版本之 *硬件设置说明书*，*软件说明书*，以及 *安全说明书*，并遵循本产品装置地之各地区 / 国家之安全法规。
- TM Robot 应根据其预期用途使用。
- 风险评估结果可能需要使用额外措施降低风险。
- 在进行任何维护之前，供应给机器人之电源必须被锁定及标示，换言之危险能量已掌握或实施能量隔离。
-  按照产品使用国家或地区的相关规定和规范处理产品。

1.5 确效与责任

本手册提供的讯息不包含如何设计、安装及操作一个完整的手臂应用，也没有涉及会影响整个系统安全性的外围设备。完整系统的设计及安装必须符合所在国家对于安全要求的标准与规范。手臂使用者或整合者需了解当地国家的安全法律及安规，并避免整个系统中出现重大危险。

这包括但不限于：

- 整个系统的风险评估
- 依风险评估的定义增加其他机器及额外的风险减低机制
- 于软件中架构适当的安全机制
- 确保使用者不会修改任何安全相关措施
- 确保所有系统皆正确设计与安装
- 标明使用指示
- 标明手臂安装相关标志和整合者之连络信息
- 提供相关文件，包括风险评估及本说明书。



注意：

本产品为机械半成品 (partly complete machine)。设计与安装完整系统时，用户须遵循并符合使用地区 / 国家之安全规范。机器人的使用者及整合者应充分了解当地国家之安全法规，并避免完整系统发生任何主要风险。

1.6 责任限制

任何安全相关之信息不被本公司视为对于本公司各系列机器人产品不会造成人员受伤或物品损害之保证。

1.7 功能注记符号

下表定义本说明书中于各段落标示的功能注记符号，请于阅读各段落时详加注意，有助于提升编程效率。



重要：

此标记符号表示相关功能细节，有助于编程及使用。



提示：

此标记符号表示相关功能使用技巧，有助于提升编程效率。

功能注记符号

2. 应用软件设定

2.1 TM Palletizing Operator Kit Software Package 简介

TM Palletizing Operator Kit Software Package 包含四个项目，这些项目经过专门编程，供使用者快速设定。

- TMOperator_TMROBOT_TMPLTZOP_V002_Pillar：Pillar calibration and result saved to global variable
- TMOperator_TMROBOT_TMPLTZOP_V002_Main：Control function for palletizing
- TMOperator_TMROBOT_TMPLTZOP_V002_Grip：Grip function
- TMOperator_TMROBOT_TMPLTZOP_V002_Release：Release function for using palletizing gripper



重要：

在进行项目操作前，须先安装 TMstudio 和 TMflow 应用程序。

2.2 TM Palletizing Operator Software Configuration Workflow

以下工作流程中列出 TMstudio 和 TMflow 已正确下载并安装完成之后的步骤。

步骤	任务
1	下载 TM Palletizing Operator Kit.zip 档并解压缩出 4 个专案。
2	将 USB 碟以 TMROBOT 命名并建立：TM_Export\TMPLTZOP\Projects，然后将 4 个项目放至其中。
3	在 TMstudio 设堆栈应用并产出 text 档案 (Text 文件内纪录箱体与栈板之间的关系)。
4	导出产出的 layout text file 到指定的 USB 路径 (细节在下一节)。
5	汇入 4 个专案和 text 档至 TMflow。
6	定义 Set I/O while project error/stop。
7	立柱校正 (仅需校正一次)。
8	设定视觉任务和教导吸取点。
9	根据需求进入进阶设定。
10	开始堆栈。

2.3 TMstudio

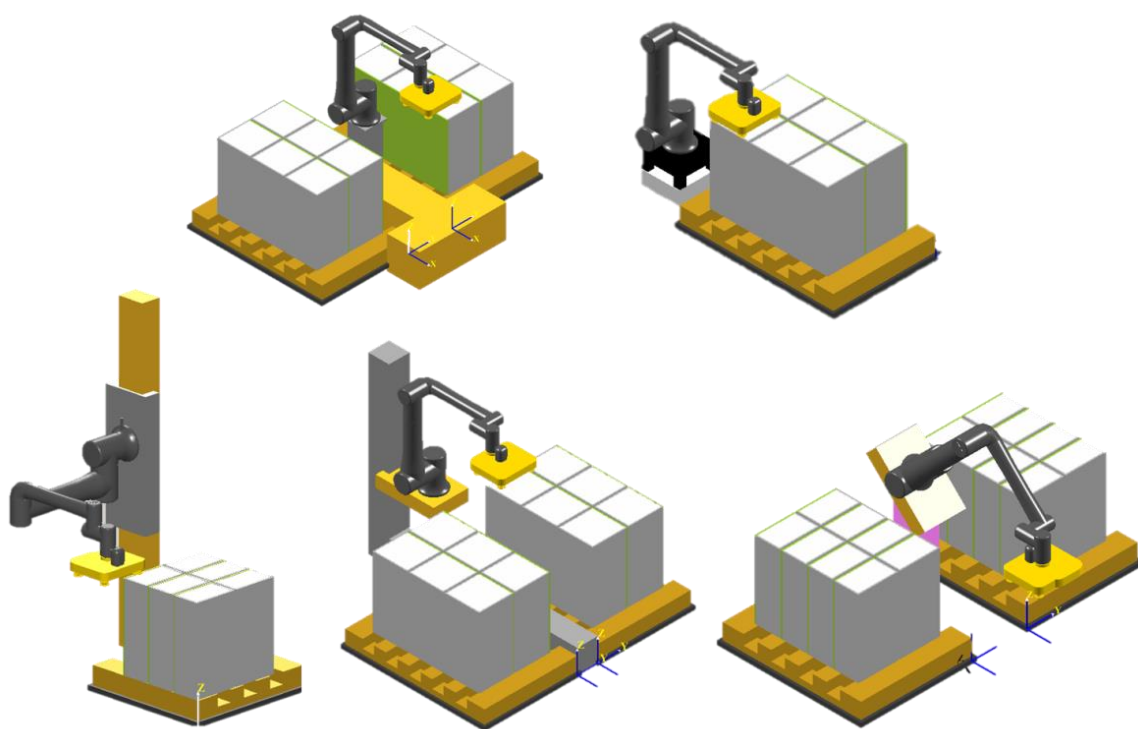
2.3.1 TMstudio 堆栈精灵及场景建立器、功能及限制



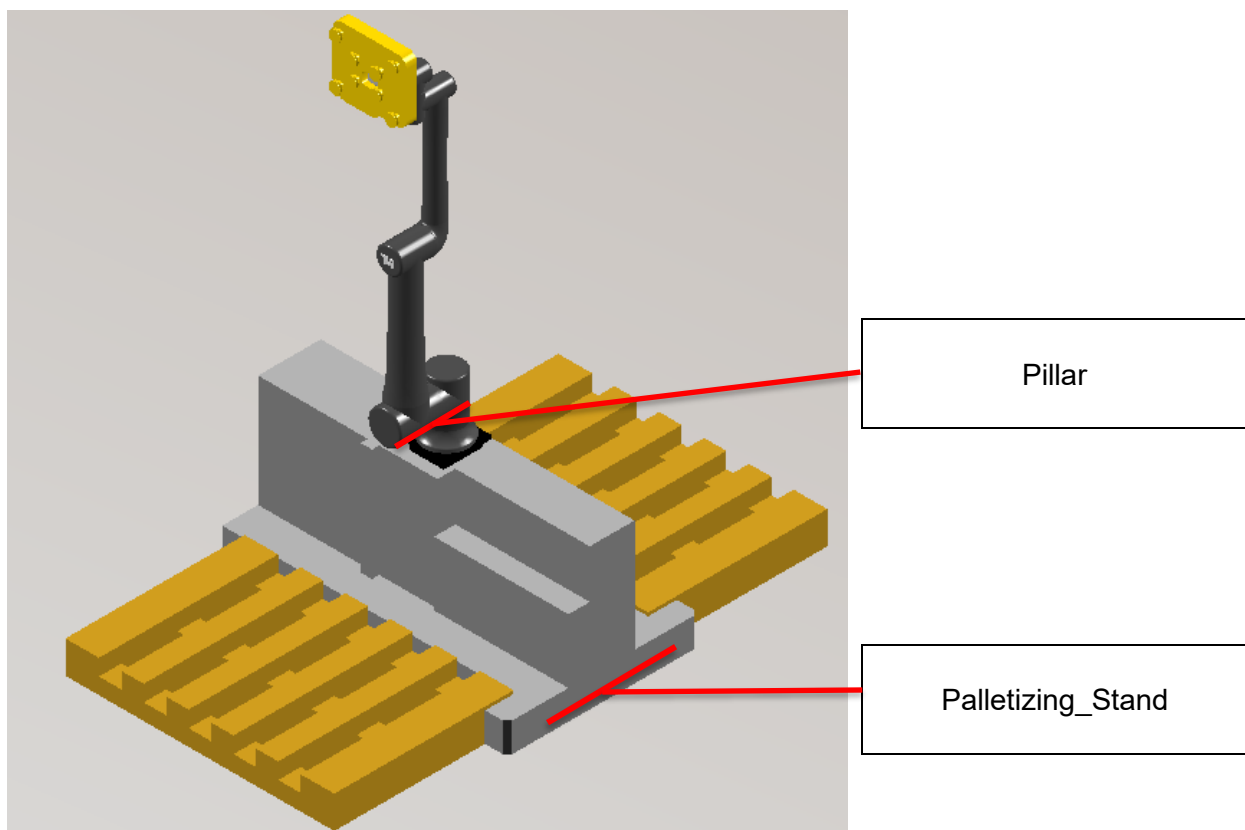
提示：

软件包附有 TM Palletizing Operator 1.1 的场景档供参考，详细操作说明请参阅 TM Palletizing Operator 1.1 说明书。

如下为可能的堆栈情境示意。



以下将以建立常见之左右堆放场景，进行范例说明。



IMPORTANT

重要：

在 TMstudio 堆栈精灵中，使用者可以建构各种姿态，供手臂安装，且输出档案会附上手臂底座与栈板底座间的关系。为达成项目中各点位及其运动类型之适用性，使用者需要依情况调整，以应用客制化设计之栈板应用。

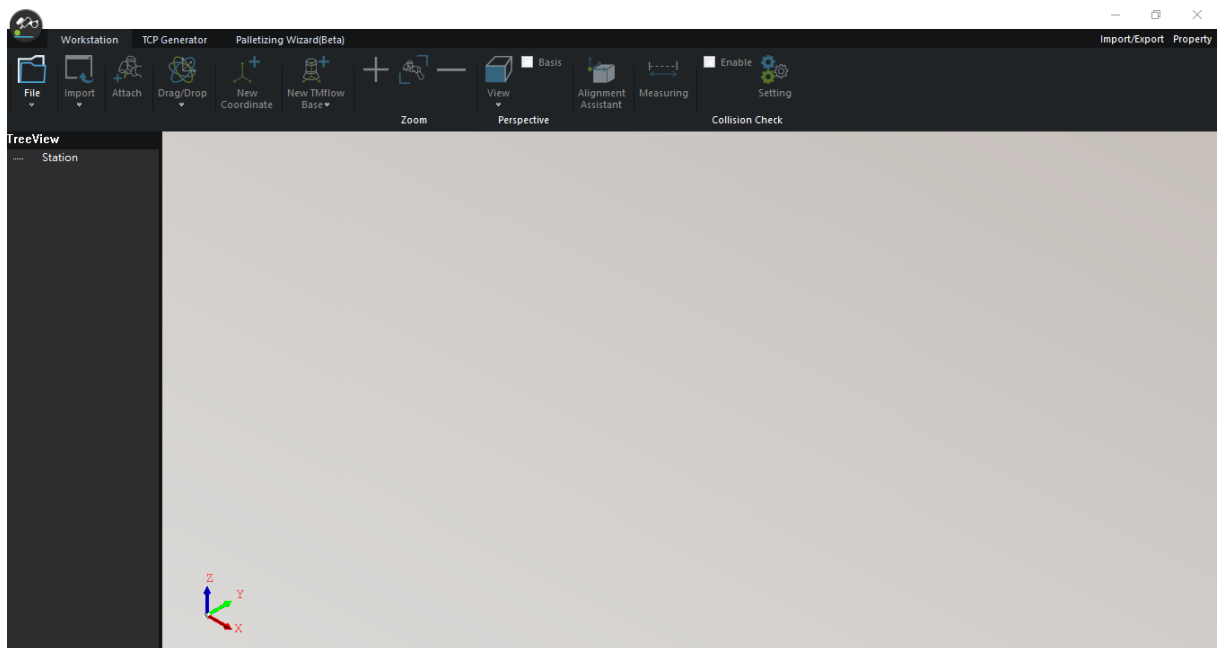
```
TM Palletizing OP.txt - [Title Bar]
(F) (E) (O) (V) (H)
//box_2
////////////////////////////////////
//boxx_info: box setting lx,ly,H(mm),weight(kg),label;
box2_info=290.0,180.0,278.0,5.0,0

//boxx_Elevated_Height :box taken point offset(mm)
box2_Elevated_Height=200

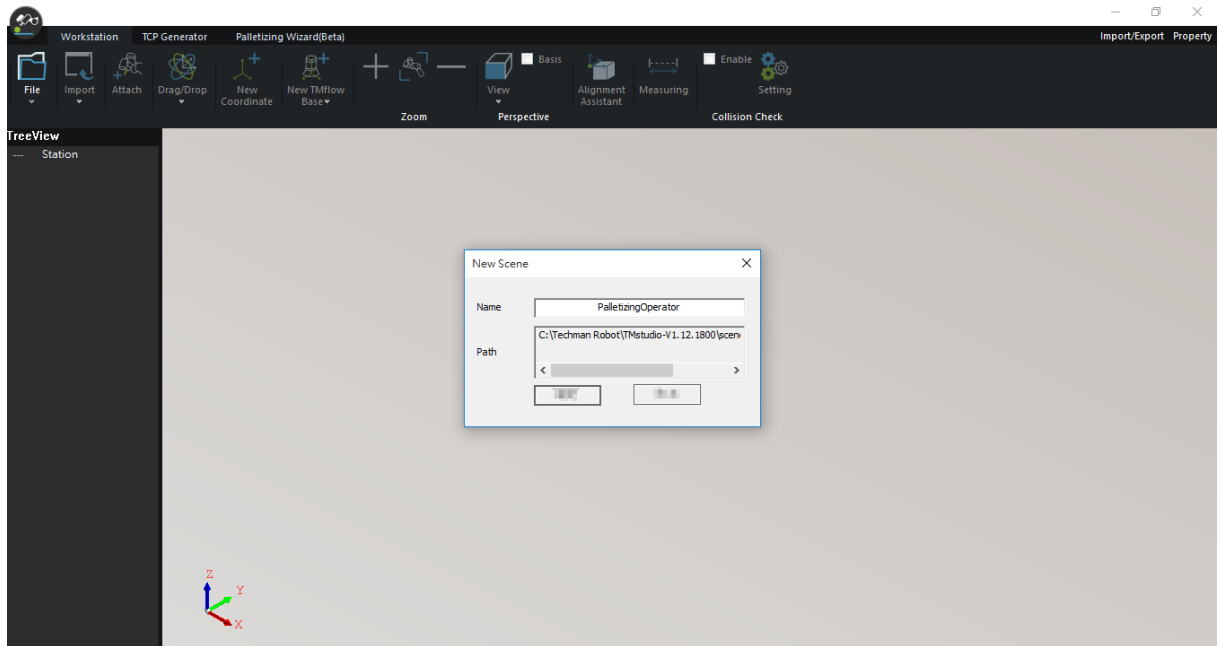
////////////////////////////////////
//Pallet_0
////////////////////////////////////
//palletx_base_xyzabc base relation : robot base to palletx base (without axis 7)
Pallet_0_base_xyzabc=600,-300,-680,0,-0,0

//pallet setting lx,ly,h(mm),weight(kg);
Pallet_0_info=800.0,600.0,144.0,0.0
```

以下列出 TMstudio 之 **Workstation** 页面。



点选 **File**，并点选 **New** 以建立新场景。新场景取名为 **PalletizingOperator** 如下所示。



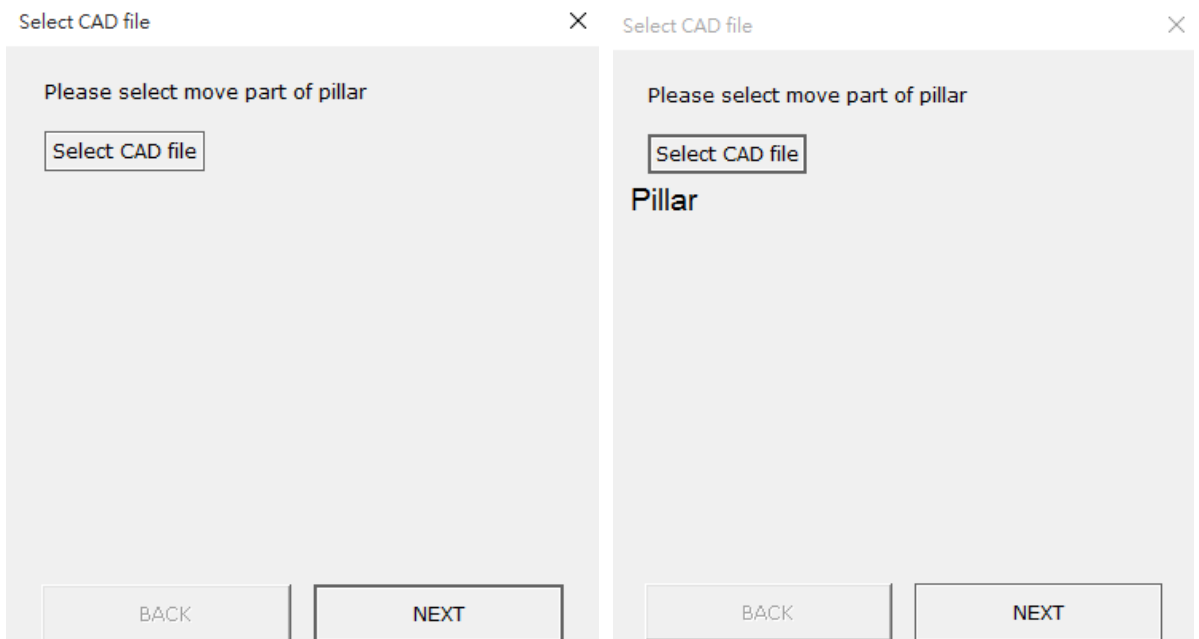
2.3.1.1 步骤 1 – 汇入 CAD 檔

點選 TMstudio0 上方 **Import**，選擇 **CAD**，將所繪制的 3D CAD 圖檔匯入 TMstudio 內，如下匯入了底座 Palletizing_Stand。



2.3.1.2 步骤 2 – 汇入 Pillar Base

再點選 Import 下的 Palletizing，插入特殊坐标系 Pillar Base，並選擇 CAD File (范例为 Pillar) 作为 Pillar Base 的子组件。



Note

提示：

在实际运行中，所有 Pillar Bas 下的对象，会根据设定项 Pillar Height (于模拟中对应的各层立柱高度)，进行移动。

Note

提示：

若于此步骤未将对象作为此坐标系之子组件，也可稍后透过汇入一般 CAD 档，再更改子母关系。

IMPORTANT

重要：

场景内最多仅支援插入一个立柱坐标系。

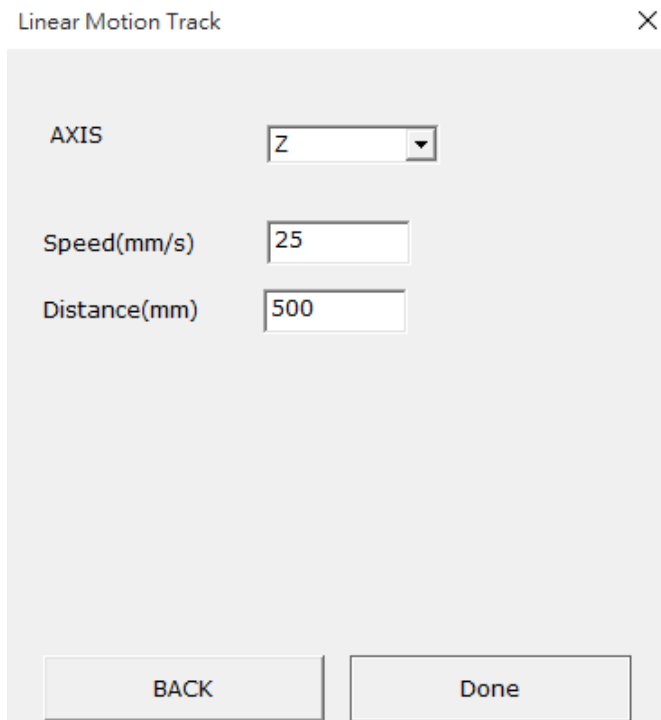
IMPORTANT

重要：

使用者无法使用 X 或 Y 方向的外部轴 (轨道)。仅立柱可以使用 Z 方向的外部轴。

2.3.1.3 步骤 3 – 设定 Pillar Base 参数

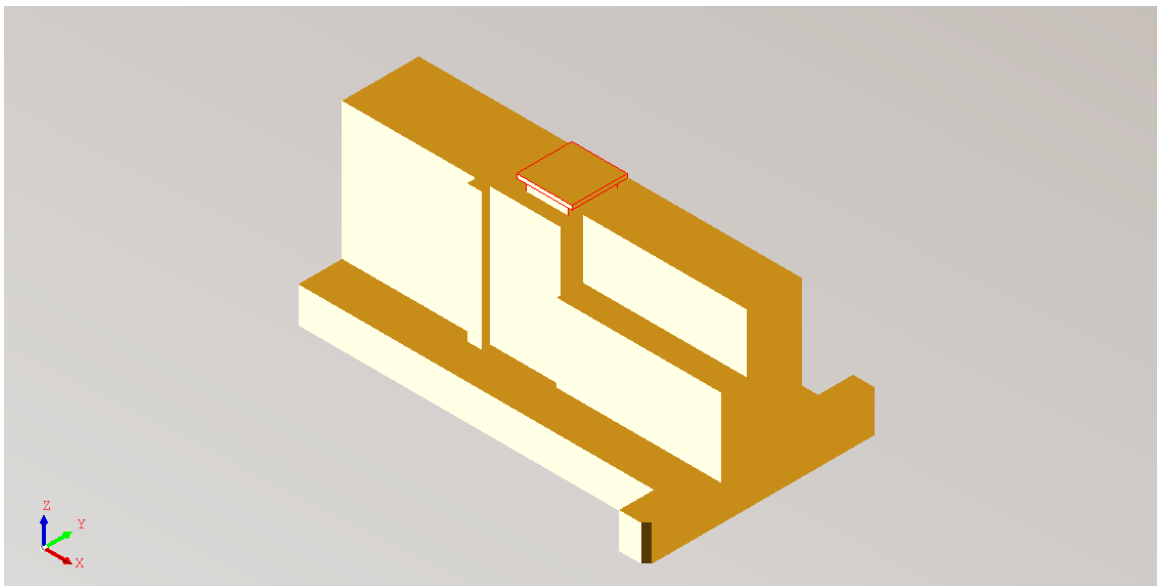
加载 Pillar Base 与 CAD 档后，可设定 Pillar Base 参数，假设立柱的规格是长度 500mm，且由底部升至顶部需要 20s，因此，设定速度为 25mm/s，高度为 500mm。距离可作为模拟时计算移动高度的阈值，而速度参数为作为仿真时，估算时间使用。



重要：

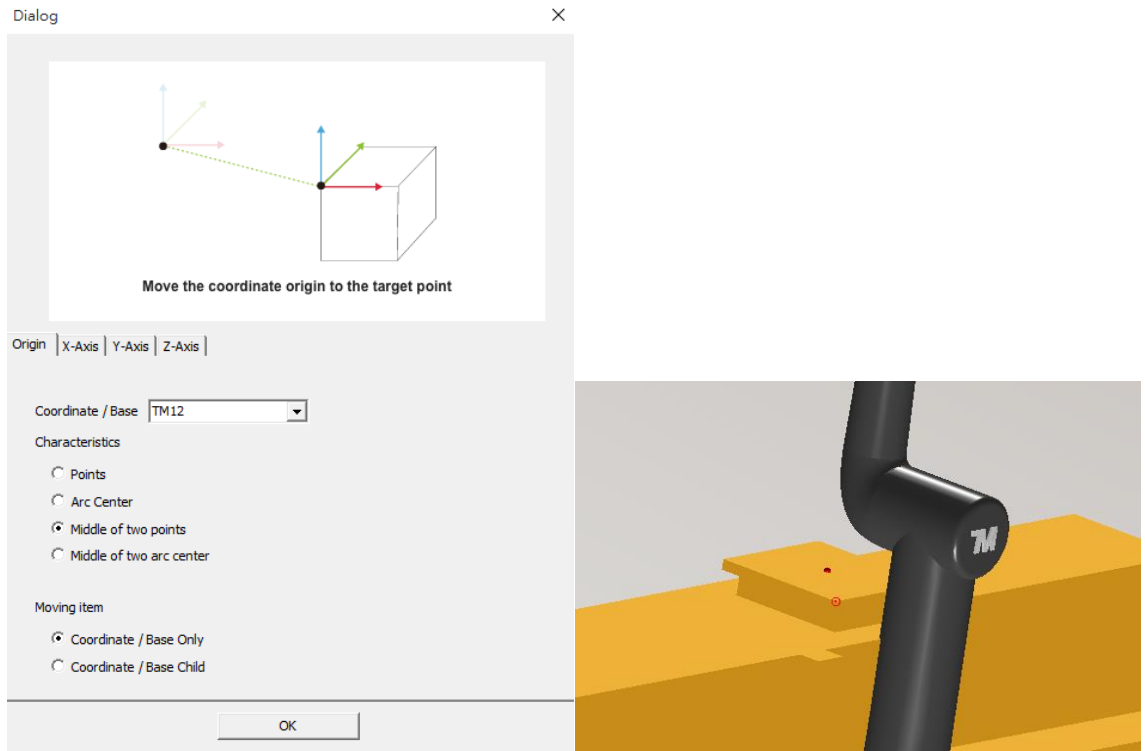
距离可作为仿真时，Pillar Base 下的子组件的最大距离。速度参数可作为仿真时，估算整体运行时间使用。

汇入后如下图所示。

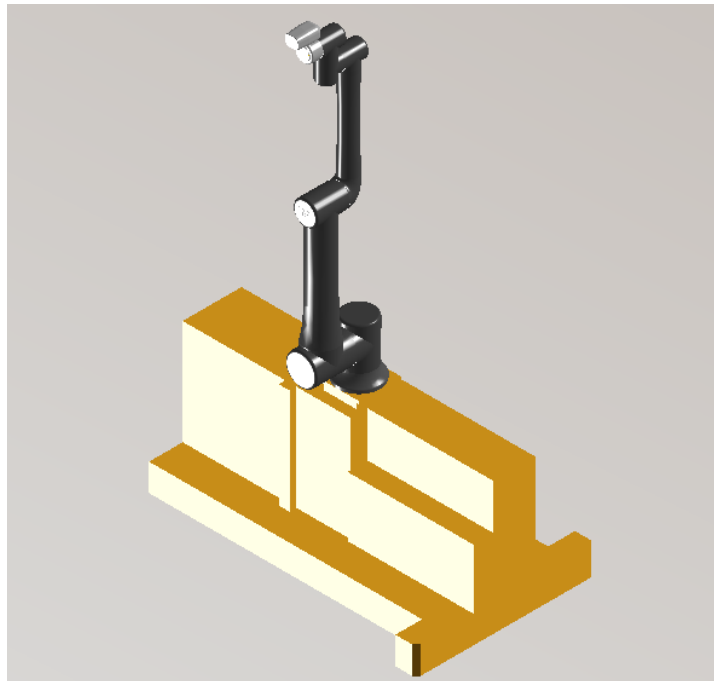


2.3.1.4 步骤 4 – 汇入手臂

點選 Import Load robot 可汇入 Palletizing 使用之手臂，若此应用使用 TM12，则點選 TM12 进行汇入，并點選上方列表 Alignment Assistant，将 TM12 复制于 Pillar Base 下 CAD 檔的中心。



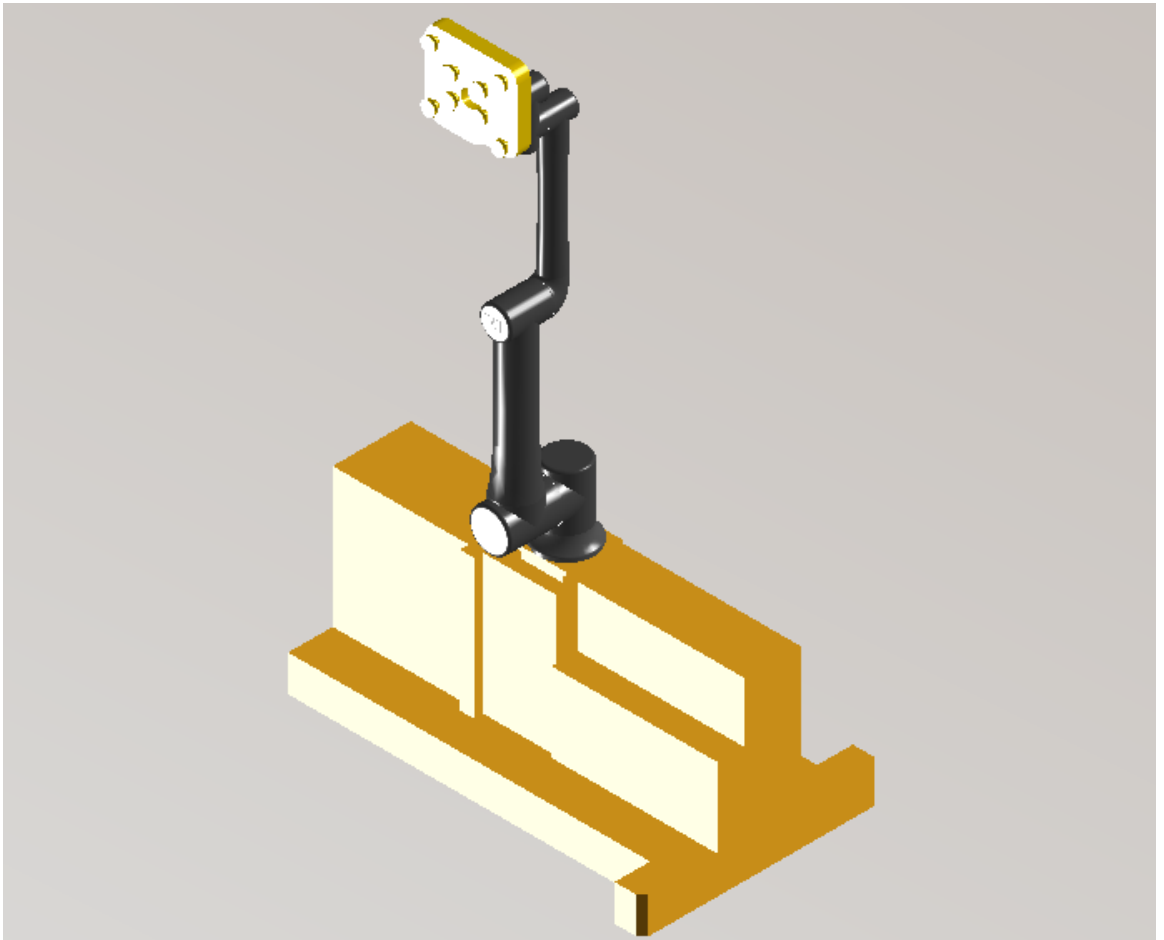
设定完成后如下所示。



2.3.1.5 步骤 5 – 汇入工具

点选 TMstudio 上方 **Attach**，选择 TCP 档案，并点选手臂，则工具与手臂将进行连接，关于 TCP 的设定，可参考 Software Manual TMstudio 手册中关于 TCP Generator 操作页面之

说明。

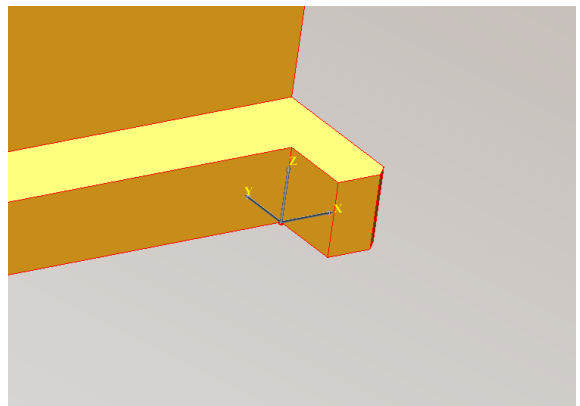
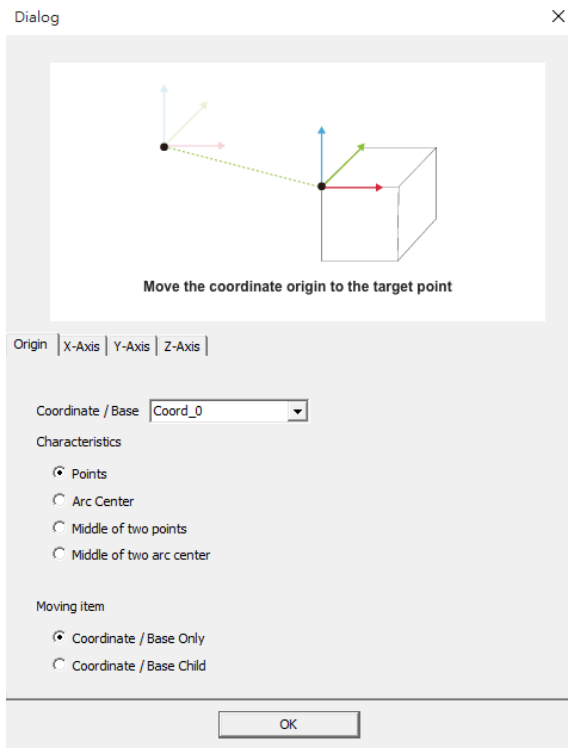


重要：

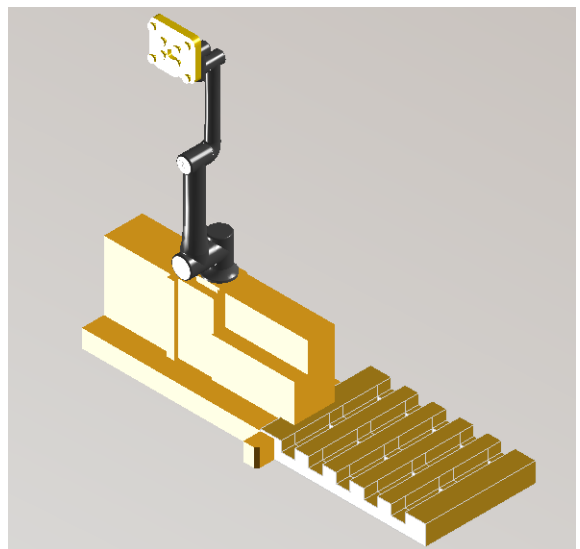
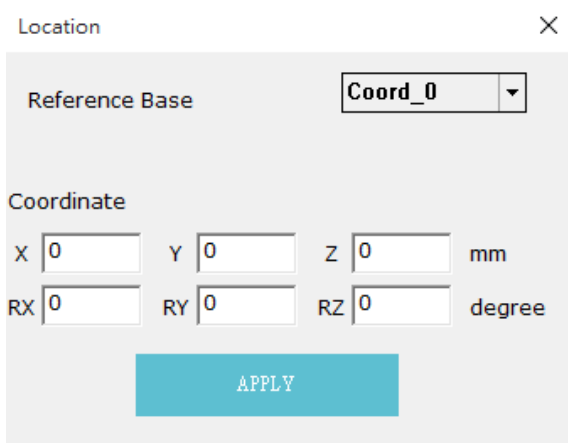
于场景中加载的工具，会变更为使用者使用堆栈精灵时的默认值。

2.3.1.6 步骤 6 – 插入堆栈并定位

为使栈板放置于 CAD 檔 Palletizing_Stand 之角落，先点选 New Coordinate 建立一个坐标系，再透过 Alignment Assistant，将坐标系 Coord_0 放置于 Palletizing_Stand 的角落。



點選 **Import** 加入特殊对象 **Pallet**，并设定其尺寸，按下 **OK** 后以产生栈板。产生的栈板可透过 **Location**，设定与 **Coord_0** 之间的距离为 0，达到 **Coord_0** 迭合之目的。



Note

提示：

點選左侧树形图目标，或直接點選 TMstudio 内的 CAD，透过鼠标右键，均可对其参数进行设定。

Note

提示：

栈板可设定范围如下：

- Length: 1200 mm [可设定范围：25~2000]
- Width: 800 mm [可设定范围：25~2000]
- Height: 144 mm [可设定范围：25~2000]

IMPORTANT

重要：

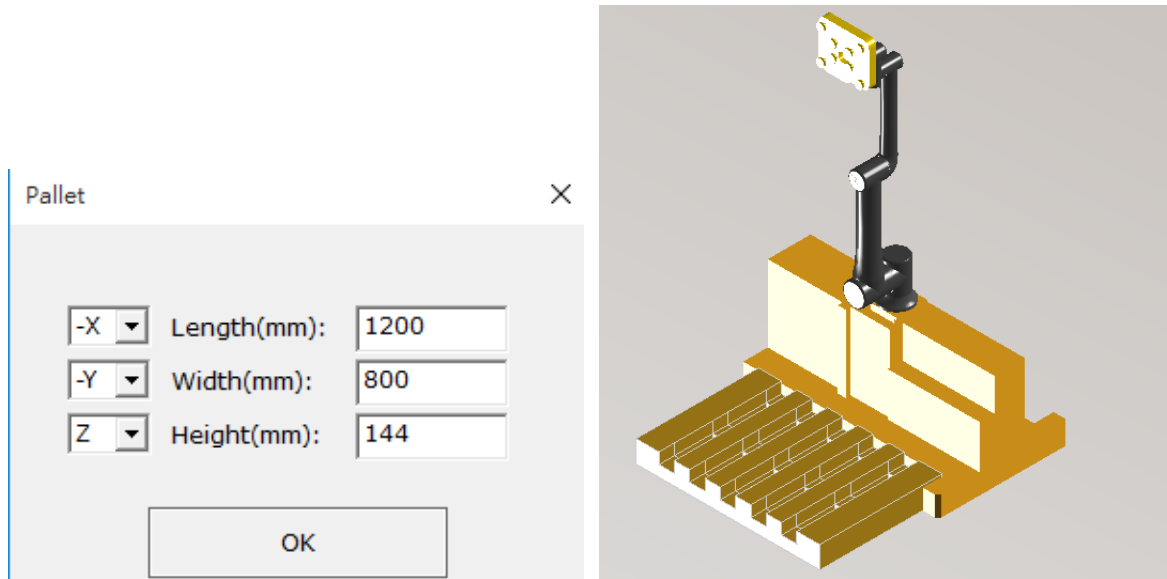
在场景中设定的栈板尺寸，会变更为使用者于堆栈精灵时的预设使用。

IMPORTANT

重要：

场景内最多仅支援插入两个栈板。

2.3.1.7 步骤 7 - 更改栈板长出方向

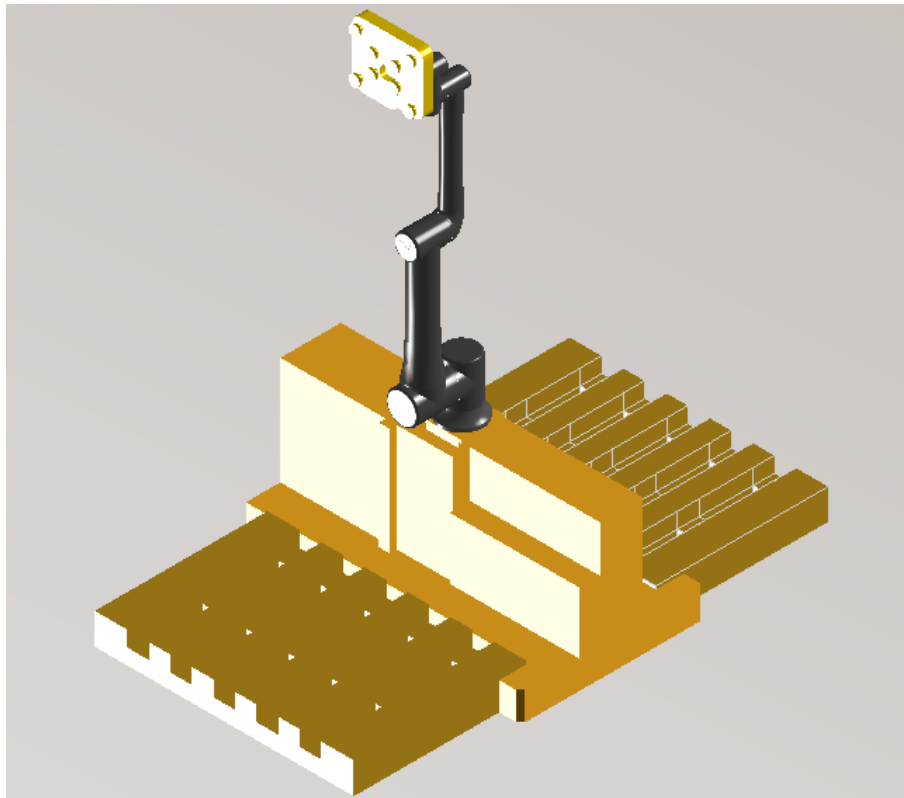


IMPORTANT

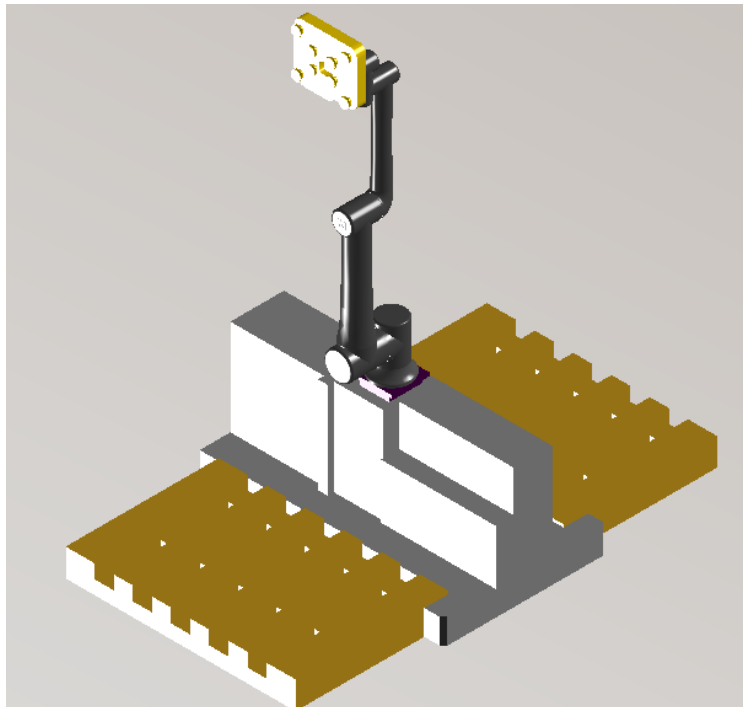
重要：

在场景中设定的栈板初始位置坐标，会成为用户更改栈板大小的长出方向。

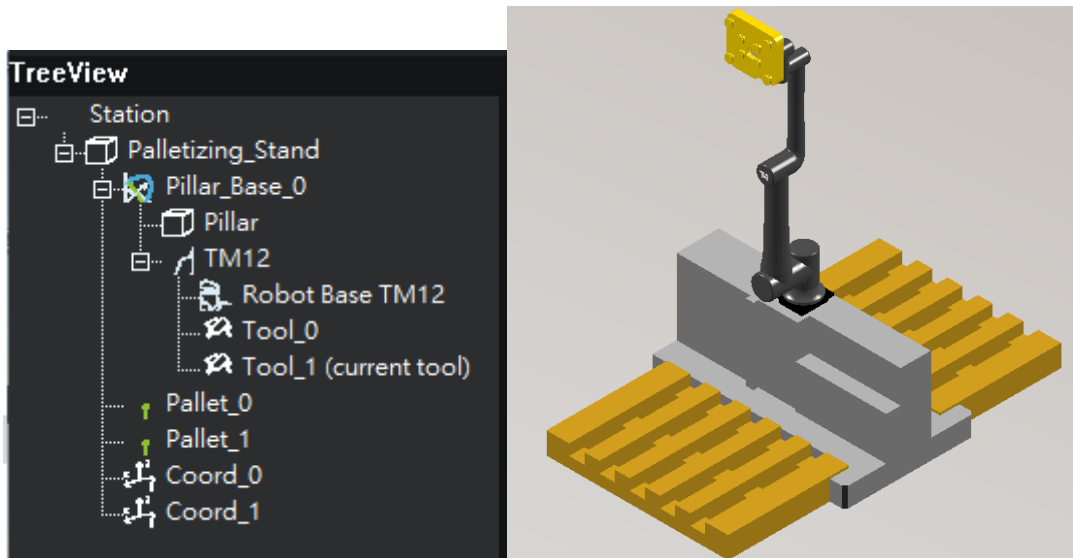
以此方式，重复上述步骤，则可建立另外一边的栈板。



用户也可以鼠标右键点选 CAD 档，改变颜色为所喜欢之颜色。

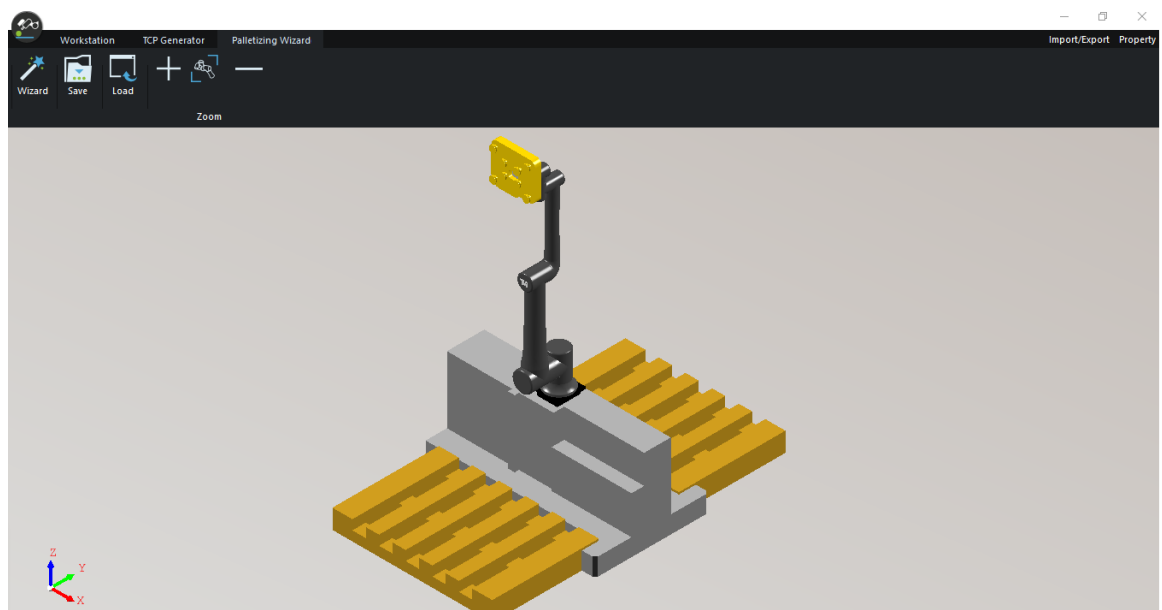


2.3.1.8 步骤 8 – 设定母子关系
相对子母关系如下所示。



2.3.1.9 步骤 9 – 开始堆栈

确认完成后，可切换至 Palletizing 页面使用自行设计之场景进行堆栈。



提示：

依据使用者定义场景，TMstudio堆栈精灵支持5种不同的堆/拆栈模式。

Note

1. 相同箱体堆栈
2. 不同箱体堆栈：同时处理两种不同箱体的堆栈任务
3. 相同箱体拆栈：将栈板上的箱体做拆栈处理
4. 不同箱体拆栈：将不同栈板上不同的箱体做拆栈处理
5. 相同箱体更换栈板重新堆栈：将箱体从A栈板换到B栈板做堆栈，两个栈板可为不同规格

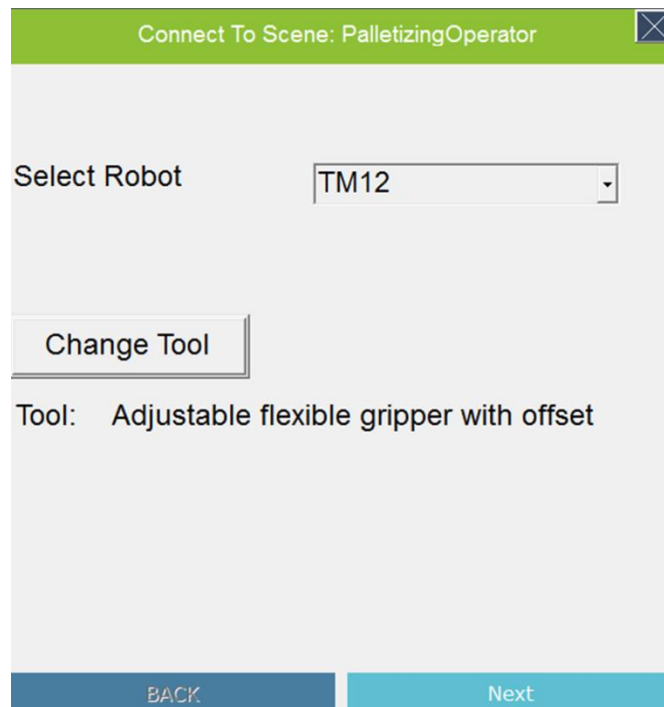
以下介绍如何使用堆栈精灵模拟与汇出点位文件。

2.3.2 在堆栈精灵设堆栈应用

堆栈精灵为 TMstudio 中针对堆栈应用的解决方案，此功能可让用户自行定义其偏好的堆栈模式，藉由不同的工具、参数、箱体排列方式等，验证其堆栈的可行性，并模拟及侦测潜在的错误。以下对该软件简要介绍。

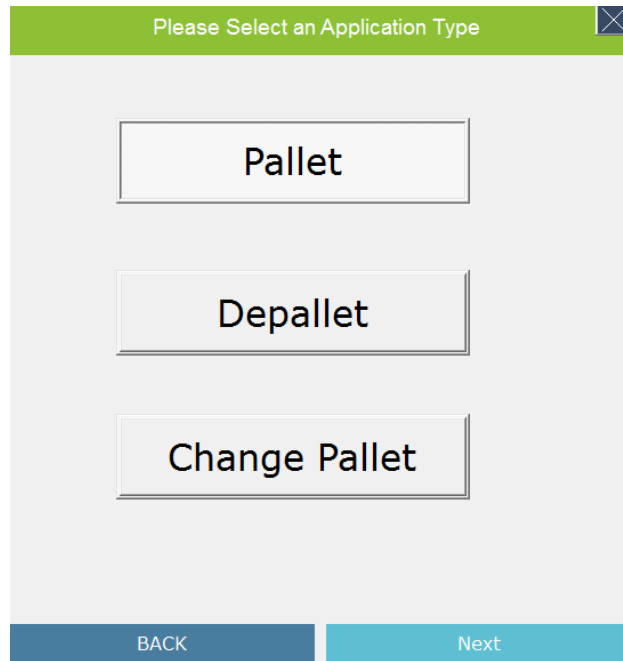
2.3.2.1 更换夹爪

若需使用不同夹爪，点击**更换工具**选择适合的夹爪。



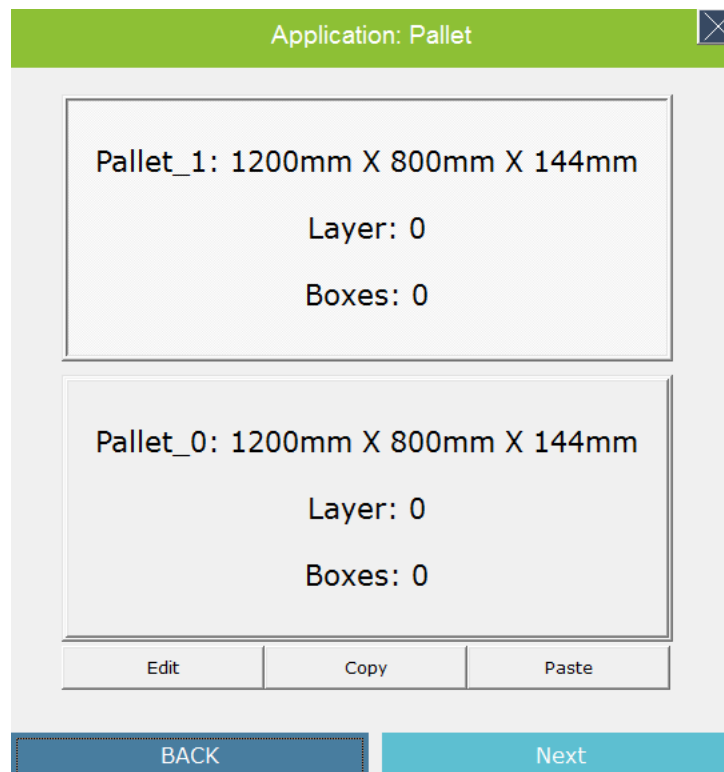
2.3.2.2 选择堆栈类型

堆栈精灵支持堆栈、卸栈、换栈等功能，用户可根据需求，选择对应的类型。



2.3.2.3 开始堆栈设定

选择对应要编辑的栈板后，点选 **Edit**，即可对栈板进行编辑。



2.3.2.4 步骤 1 - 设置栈板大小和边界

于步骤 1 中，使用者可透过下拉选单(参照下表) 选择预设的栈板模板进行使用，或自行输入栈板的长度(mm)，宽度(mm)和高度(mm)以自定义栈板，设定完栈板的尺寸后，透过边界设定，可以定义箱体于栈板上摆放的最大范围供，参照以下示意图：

- 以下示意图为自定义栈板设置

长度：1000 mm、宽度：800 mm、高度：144 mm

Note

提示：

使用者可将栈板的参数另存新檔或删除自定义的栈板参数。

模板	尺寸
Customize	N/A
EPAL 6 PALLET	800 x 600 x 144 (mm)
EPAL 3 PALLET	1200 x 1000 x 144 (mm)
EPAL 2 PALLET	1200 x 1000 x 162 (mm)
EPAL EUR	1200 x 800 x 144 (mm)
GMA	1219 x 1016 x 120.7 (mm)
UK standard	1200 x 1000 x 150 (mm)

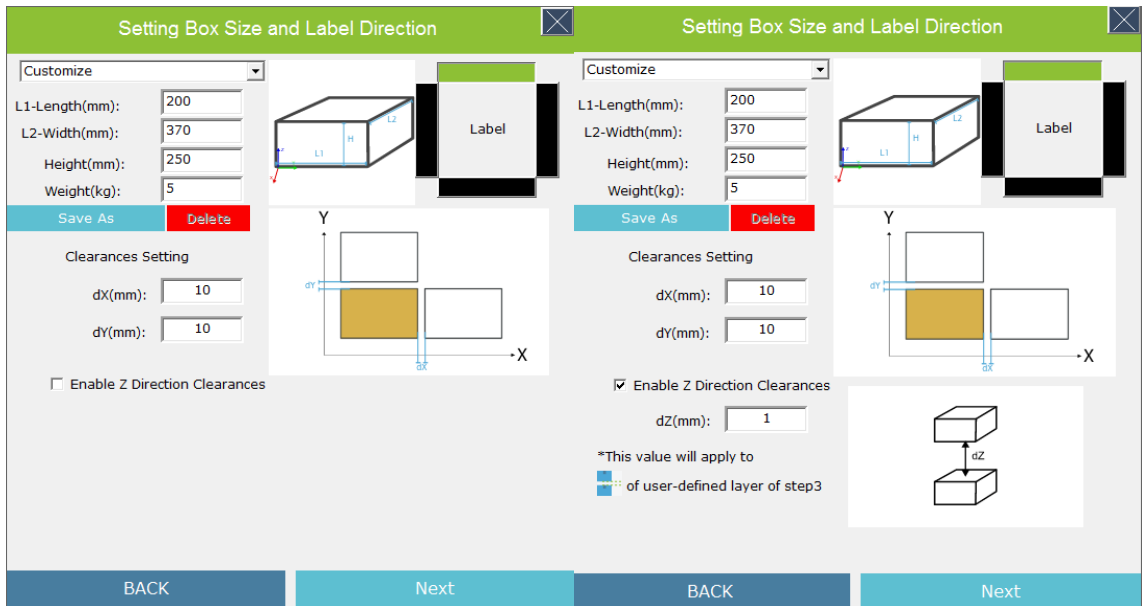
2.3.2.5 步骤 2 - 设置箱体大小和标签方向

在步骤 2，使用者可设定箱体长度(mm)，宽度(mm)，高度(mm)和重量(kg)与标签方向。另外也可启用 Z 方向间距 (dX/dY/dZ)。

- 以下示意图为自定义栈板设置

长度：200 mm、宽度：370 mm、高度：250 mm

dx : 10 mm、dy : 10 mm



Note

提示：

使用者可将栈板的参数另存新档或删除自定义的栈板参数。

Note


提示：

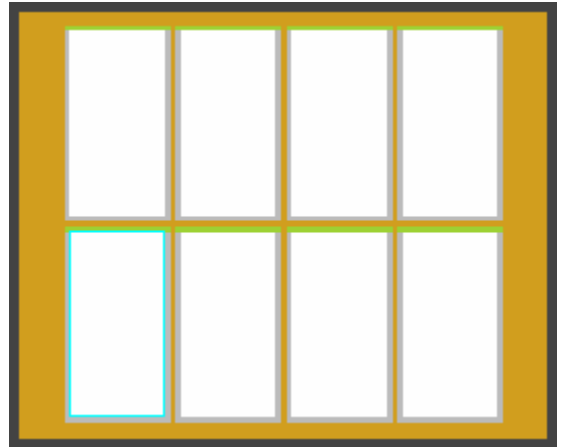
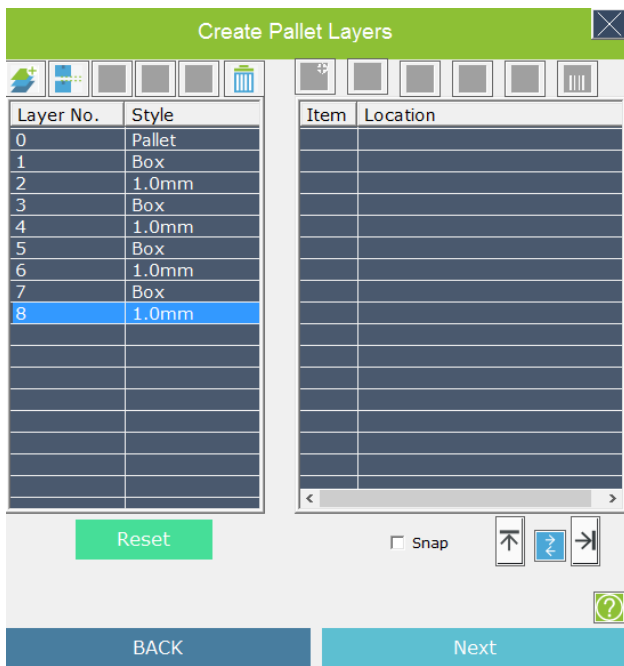
启动 Z 方向间距允许使用者在各层之间设置任务，例如插入隔板。当使用者启动 Z 方向间距，可以在步骤 3 插入各层数之间的间隔。

2.3.2.6 步骤 3 -建立层及摆放方式

在步骤 3，使用者可设置箱体的层数及其排列方式。以下会示范如何建立层与如何对箱体进行排列。


下面以四层为范例。

点击  布局箱体位置。注意箱体与栈板的尺寸以先前步骤设定为准。















Note

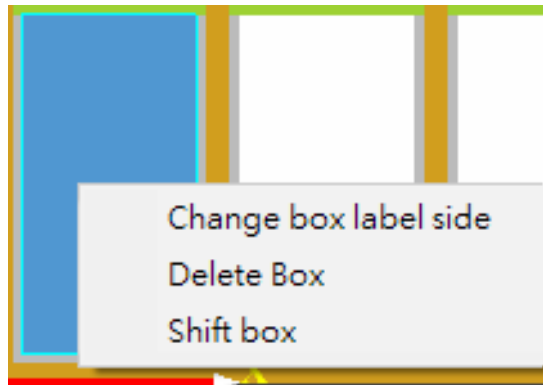
提示：

若使用者于步骤 2 中勾选并设置了 Z 方向间隙，则点击  将插入所选层至先前设置的间隙值中。

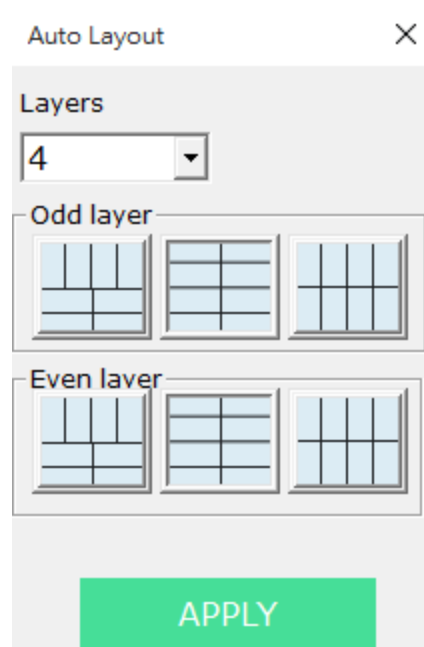
用户可透过问号钮显示图标对应之功能说明。有关功能图标的更多详细讯息，请参见下表：

图示	功能	图示	功能
	增加箱体层		新增箱体
	增加 Z 轴方向间距		旋转箱体
	复制		更改箱体标签面
	贴上		自动布局
	清除		移动箱体
	删除		删除箱体

用户可于箱体点选鼠标右键或按压屏幕上的箱体，进行**移动箱体**、**删除箱体**、或**更改箱体标签面**。




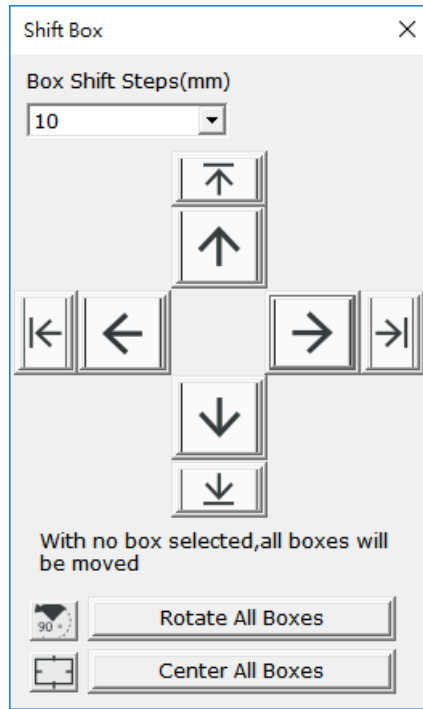
用户可透过自动布局图示 ，根据需求选择自动布局的类型。





Note







提示：
当奇数、偶数层皆为  时，可选择 **INVERT**，则偶数层将会与基数层摆放差 180 度。










用户可根据偏好透过鼠标点击  或按压箱体并点选移动箱体来移动这些箱体。



选项	功能叙述
箱体寸动 (mm)	设定箱体寸动，预设 10 mm。
旋转所有箱体	所有箱体旋转 90 度
置中所有箱体	将所有箱体置于中间位置 (中间位置以栈板边界定义)
	透过已定义的箱体寸动距离依照箭头方向移动箱体 <ul style="list-style-type: none"> ● 若选择单一箱体，点击/轻触此箭头将按定义的箱体寸动距离来移动选定的箱体。 ● 若无选择任一箱体，点击/轻触此箭头将按定义的箱体寸动距离来移动所有的箱体。
	将箱体依据箭头方向，使其根据根据边界设定(dx, dy) 靠近边界或先前放置的箱体 <ul style="list-style-type: none"> ● 若选择单一箱体，则点击/轻触此箭头仅移动选定的箱体。 ● 若无选择任一箱体，则点击/轻触此箭头将移动所有箱体。

Snap 可针对边界或箱体进行贴齐，快速完成 Layout，可透过勾选下方 Snap 的复选框启用该功能，相关功能及范例可参考如下。

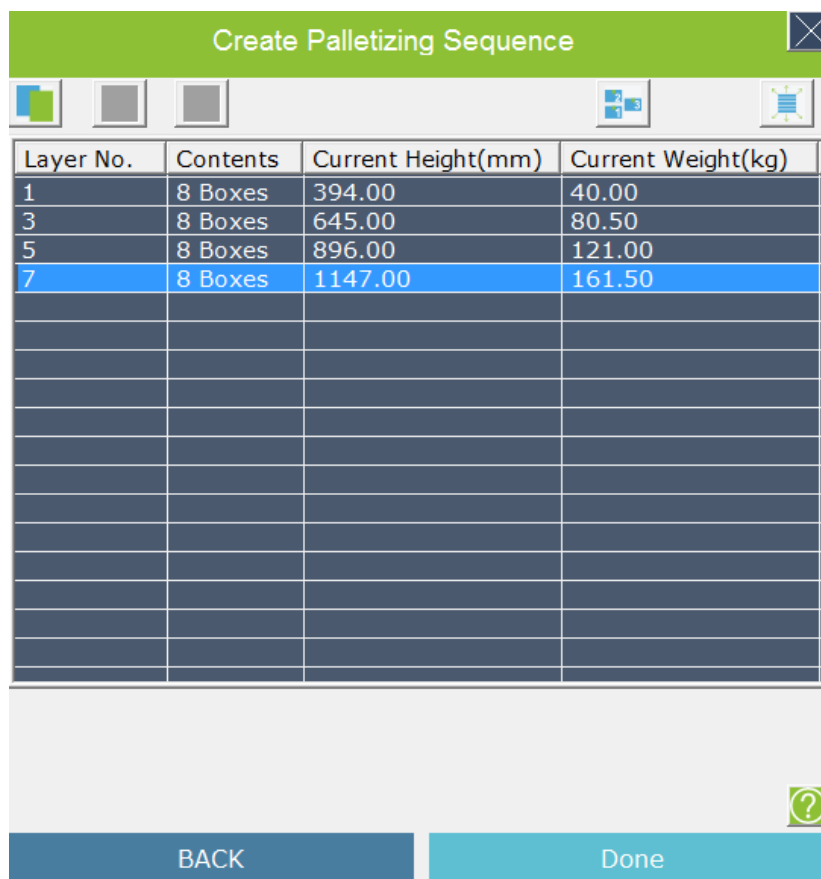
选项	功能叙述
 	放置后会往上进行贴齐，再往右进行贴齐
	点选后会变更为 
	点选后会变更为 

选项	功能叙述
	点选后会变更为 
	点选后会变更为 
	左右替换，如   ，点选后会变更为  
<input type="checkbox"/>	开启/关闭 Snap 功能

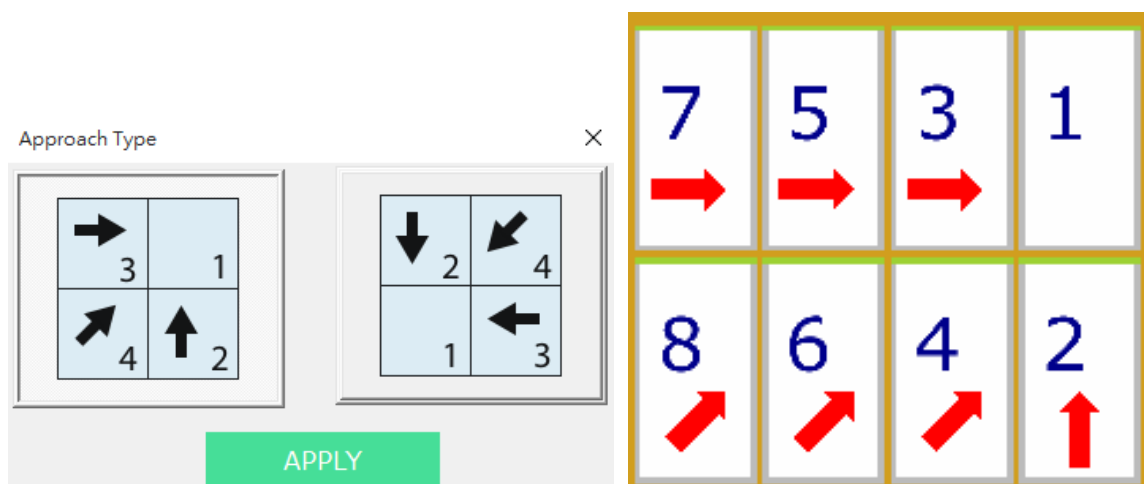


2.3.2.7 步骤 4 -建立层的放置顺序

在步骤 4 使用者可依在步骤 3 设置的堆栈建立堆栈顺序。以下步骤作为第一次建立堆栈顺序的范例，以**自动序列**示范。



自动序列让程序自动设置箱体堆栈的顺序。数字表示为箱体进行堆栈的顺序。

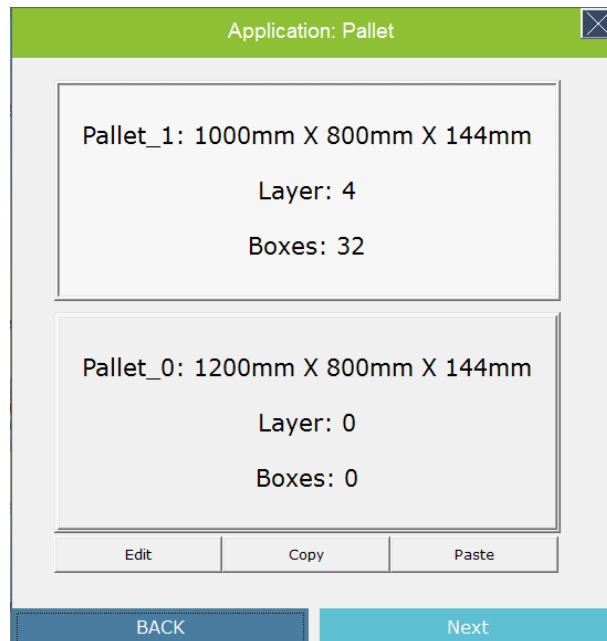


有关功能图标的更多详细讯息，请参见下表。用户可透过点选问号钮显示按钮对应的功能。

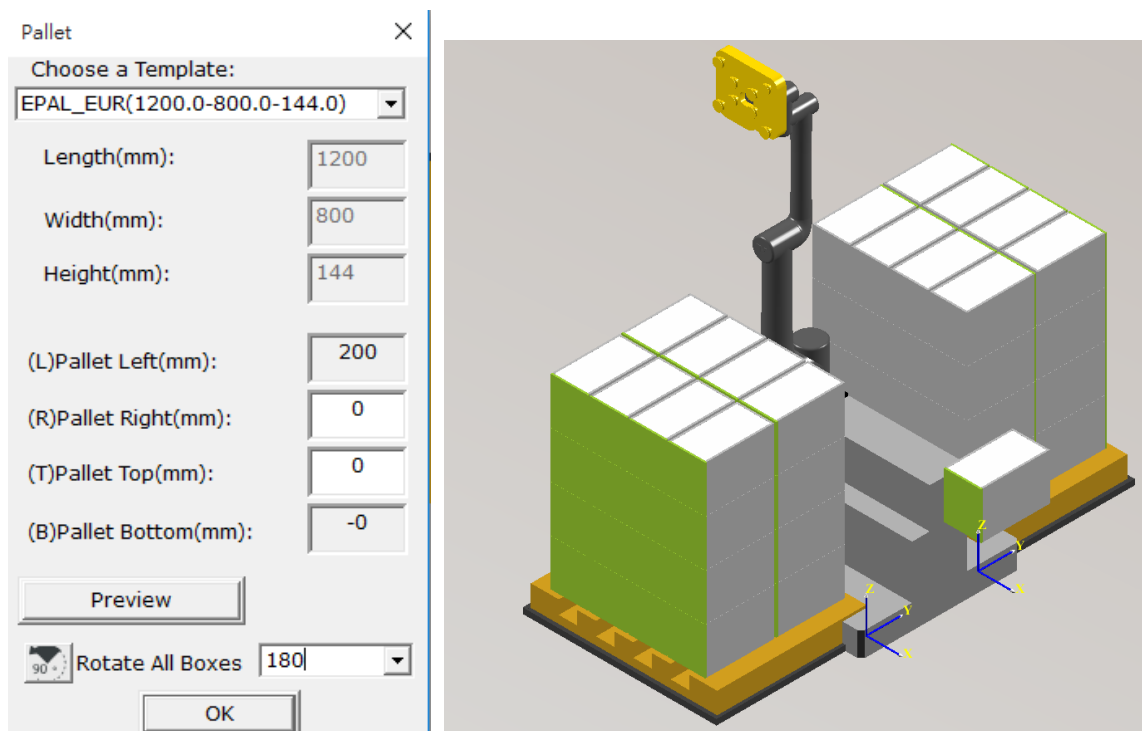
图示	功能	图示	功能
	复制箱体序列		自动序列
	贴上箱体序列		增加箱体序列
	上一步		

2.3.2.8 复制箱体

完成栈板的设定后，将返回前一页面，使用者可在对另外一栈板逐一进行编程。



或透过复制贴上，完成两边栈板的设定，点选 **Next** 进入下一步。



Note

提示：

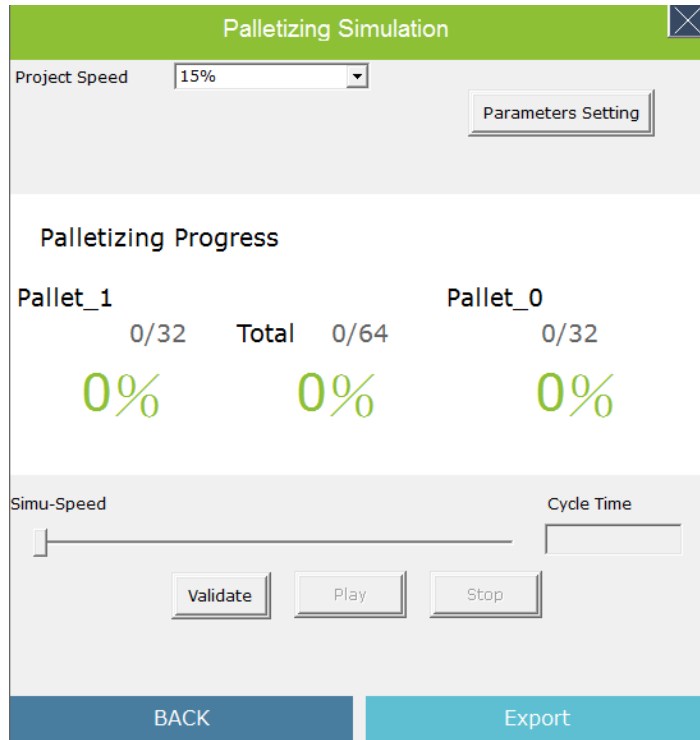
当将 Pallet_0 的箱体复制到 Pallet_1，若进行旋转，则除了箱体摆放位置外，放置向量也会被旋转。

Note

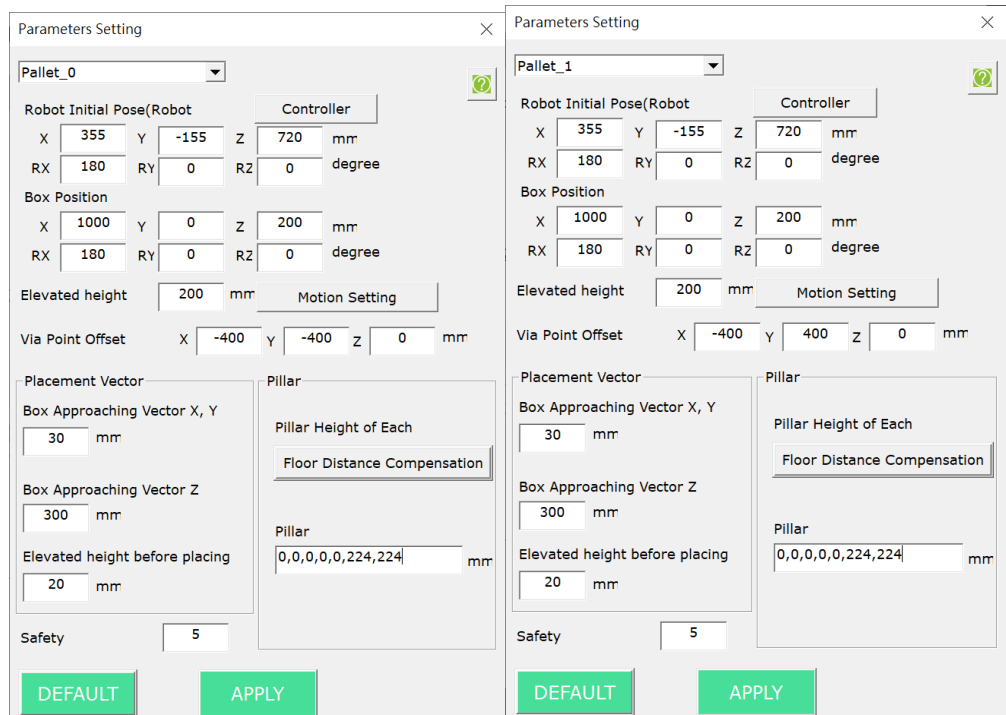
提示：

复制栈板时，由于复制的是实际的边界大小，因此不会有复制贴上后，箱体超过边界的问题发生。

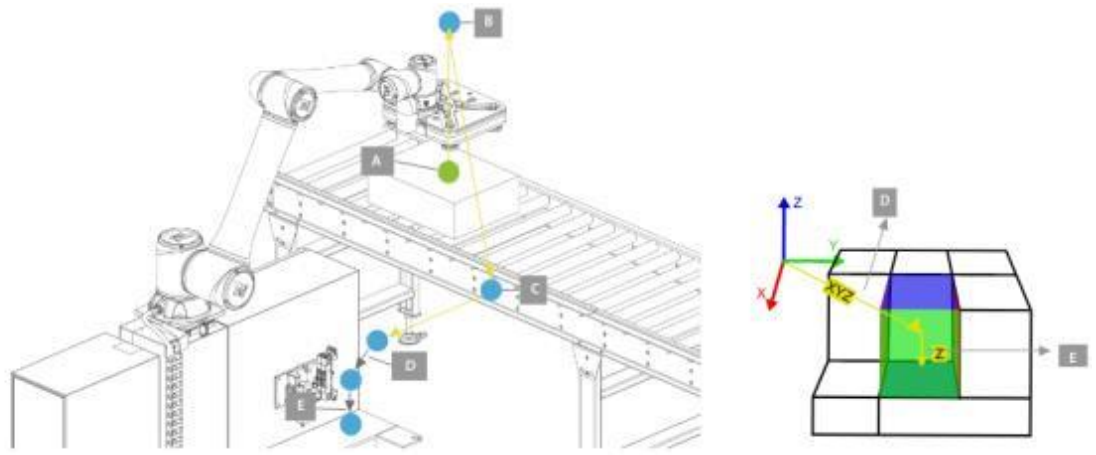
2.3.2.9 模拟页面 – 参数设定



点击参数设定可设定 Palletizing 的参数。如下图所示，可输入可向参数作为模拟时的参数。请参照以下相关参数的详细叙述。设定完成后点击套用。



参数		叙述
控制器		可透过控制器教导手臂进行堆栈作业前或立柱升降时对应的姿态
机器人初始位置		手臂进行堆栈作业前或立柱升降时对应的姿态
运动设定		可将 Elevated Height to Via Point Offset 以及 Via Point Offset to Placement Vector 的移动方式设定为 PTP 或 Line 。(建议设定为 PTP)
箱体位置 (A)		机器人夹取箱体的位置
取得箱体上升高度 (B)		当机器人夹取箱体后，沿 Z 轴的上升高度
中继点偏移 (C)		取得箱体后上升高度的相对位置
放置向量	箱体接近向量 X, Y, Z (D)	箱体接近向量 X, Y, Z 轴
	放置箱体下降高度(E)	放置箱体前的高度
立柱	各层数立柱高度 (Z 方向高度被视为层)	设定左右各层数对应的立柱高度 (此参数为立柱绝对高度，如设定 100, 300, 则代表手臂堆栈第一层箱体时，立柱绝对高度为 100 mm, 堆栈第二层时，立柱的绝对高度为 300 mm)
	平面距离补偿	如使用者有需求，可点选平面距离补偿。此时机器人将会根据当前页面的参数，并以不同立柱高度尝试进行机器人箱体的放置。
安全距离 (mm)		若侦测到手臂与其余对象间距离低于此阈值时，将视为碰撞。



2.3.2.10 验证

透过模拟，可以检测堆栈/卸栈/换栈过程中可能发生的潜在错误（如点位无法到达、空间中发生碰撞等），以及估算该项流程所需要的总时间。点选**验证**，验证完成后，则可点选**Play**以进行模拟。

Note

提示：
如果在模拟过程中发生了碰撞，则用户可以透过 **Play** 按钮，仿真到碰撞发生前的状态，以利堆栈参数的调整。

2.3.2.11 模拟

点选 **Play** 以进行模拟。

Palletizing Simulation

Project Speed 15% Parameters Setting

Palletizing Progress

Pallet_1	32/32	Total	61/64	Pallet_0	29/32
	100%		95%		90%

Simu-Speed Cycle Time

27m:29s:406ms

Validate
Play
Stop

BACK
Export

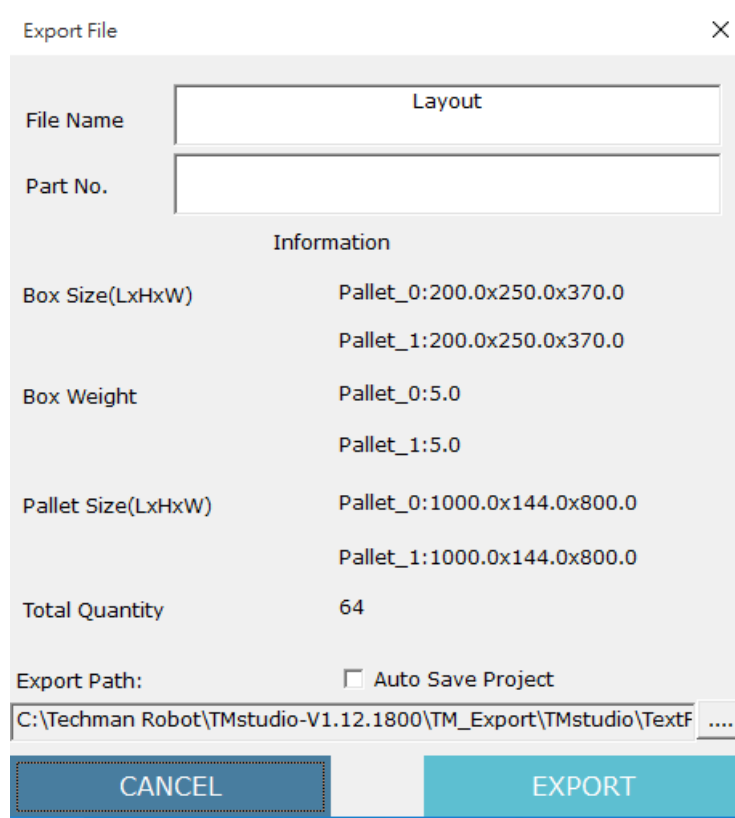
Note

提示：

- 若授权锁未连接，导出功能将无法使用。堆栈模拟循环时间不包含 I/O 等候时间与夹爪夹取箱体的时间。堆栈模拟的箱体夹取方式为 TCP 与箱体中心重合。
- 于 TMstudio 堆栈精灵仿真运动为直线运动（如：拾取/放置）与点对点 PTP 运动（如：通过点移动）之组合，具有 100%项目速度、精确到位、无轨迹混合，而实际运动以 TMflow 项目定义。

2.3.2.12 汇出

确认所有设定正确后，即可点选汇出以输出堆栈所需的档案；请将档案储存至以下路径：已命名为 **TMROBOT** 之 USB 装置建立：**TM_Export\TMPLTZOP\TextFiles**。



2.3.2.13 点位档格式说明

TMstudio 产生之点位档，可以搭配 TMflow 相关读取文件档案指令，完成箱体之取放，相关文件档案对应功能可参考如下。

名称	内容
Part_No	料号 (可于 TMstudio 存档 / 汇出点位时输入)
Total Quantity	箱子总数
date	产生此点位档的日期
pallet_version	此文件格式的版本
pallet_contorl	control = 1; 堆栈 control = 2; 卸栈 control = 3; 换栈

名称	内容
box_info	BoxN_Info, 对应第 N 个箱子的箱体信息 BoxN_Info [0]; L1-Length (mm) BoxN_Info [1]; L2-Width (mm) BoxN_Info [2]; Height (mm) BoxN_Info [3]; Weight (kg) BoxN_Info [4]; Label 方向 (上: 0; 左: 1; 下: 2; 右: 3)
box_Elevated_Height	boxN_Elevated_Height: 对应取完第 N 个箱子的上升高度 (mm)
Pallet_base_xyzabc	机器人坐标系到 pallet 坐标系的相对关系
Pallet_info	Pallet 参数 (名称 Pallet, 会随着场景设定的名称变动) Pallet_info [0]; Length (mm) Pallet_info [1]; Width (mm) Pallet_info [2]; Height (mm) Pallet_info [3]; Weight (kg) [此为保留项, 无法从 TMstudio 内设定]
Pallet_layer_info	数组对应 Pallet 各层数对应任务 Pallet_layer_info = 1; 该层为箱体层 Pallet_layer_info = 2: 该层为 Z 轴间隙 Pallet_layer_info = 3; 完成
Pallet_layer_wp_num	各层数对应箱体数量 Pallet_layer_wp_num [0]; 第一层箱体总数量 Pallet_layer_wp_num [1]; 第二层箱体总数量 Pallet_layer_wp_num [N]; 第 N 层箱体总数量
Pallet_layer_wp_sum	累计总箱体数量 Pallet_layer_wp_num [0]; 第一层箱体总数量 Pallet_layer_wp_num [1]; 第一到二层箱体总数量 Pallet_layer_wp_num [N]; 第一到 N 层箱体总数量
Pallet_Pillar_Height	数组对应 Pallet 各层数对应的立柱高度 (mm) [插入的 Z 轴高度会被视为层]
Pallet_Via_Point_Offset	相对于 Elevated_Height 位置进行 X、Y、Z 方向移动 Pallet_Via_Point_Offset [0]; 向 X 轴进行相对移动 (mm) Pallet_Via_Point_Offset [1]; 向 Y 轴进行相对移动 (mm) Pallet_Via_Point_Offset [2]; 向 Z 轴进行相对移动 (mm)
Pallet_Approach_XY_Z	相对于 Pallet_Elevated_Height_Before_Placing 的位置, 由 Pallet_Via_Point_Offset 到达的位置, 移动到该位置 Pallet_Via_Point_Offset[0], [1]; 向 X、Y 轴进行相对移动 (mm) Pallet_Approach_XY_Z [2]; 由 Z 轴反推放置向量 (mm)
Pallet_Elevated_Height_Before_Placing	放置箱体前的下降高度 (mm)
Pallet_point_wp	Pallet_point_wp_N; 栈板上第 N 个箱子的位置 Pallet_point_wp [0]; 箱子相对于栈板 X 轴的相对关系 Pallet_point_wp [1]; 箱子相对于栈板 Y 轴的相对关系 Pallet_point_wp [2]; 箱子相对于栈板 Z 轴的相对关系 Pallet_point_wp [3]; 箱子相对于栈板 RX 轴的相对关系 Pallet_point_wp [4]; 箱子相对于栈板 RY 轴的相对关系 Pallet_point_wp [6]; 箱子相对于栈板 RZ 轴的相对关系
Pallet_set_wp	Pallet_set_wp_N: 栈板上第 N 个箱子的设定 [1]: 箱体对应的栈板索引 [2]: 箱体对应层数 (从 1 开始)

名称	内容
	<p>[3]: 箱体对应该层数之索引 (从 1 开始)</p> <p>[4]: 箱体对于整体栈板之索引 (从 1 开始)</p> <p>[5]: 放置向量 (范围 0~7)</p> <div data-bbox="734 324 1396 985" style="text-align: center;"> </div> <div data-bbox="662 1019 790 1131" style="text-align: center;"> </div> <p>[6]: 放置箱体方式 (0 : 正常 ; 1 : 工具 180 度抓取)</p> <p>[7]: 取得箱体方式 (0 : 正常 ; 1 : 工具 180 度抓取)</p> <p>[8]: 对应的箱体 (1 = box1; 2 = box2)</p>

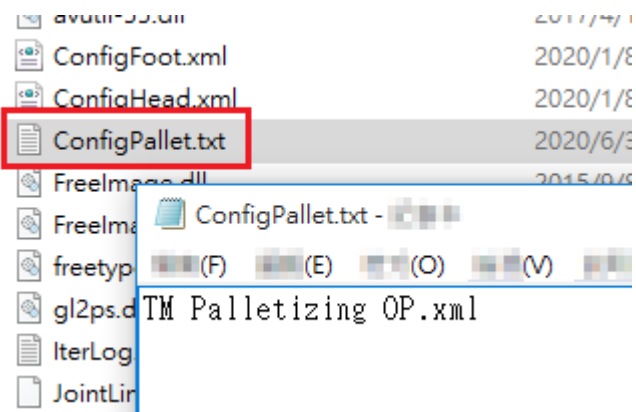


提示：

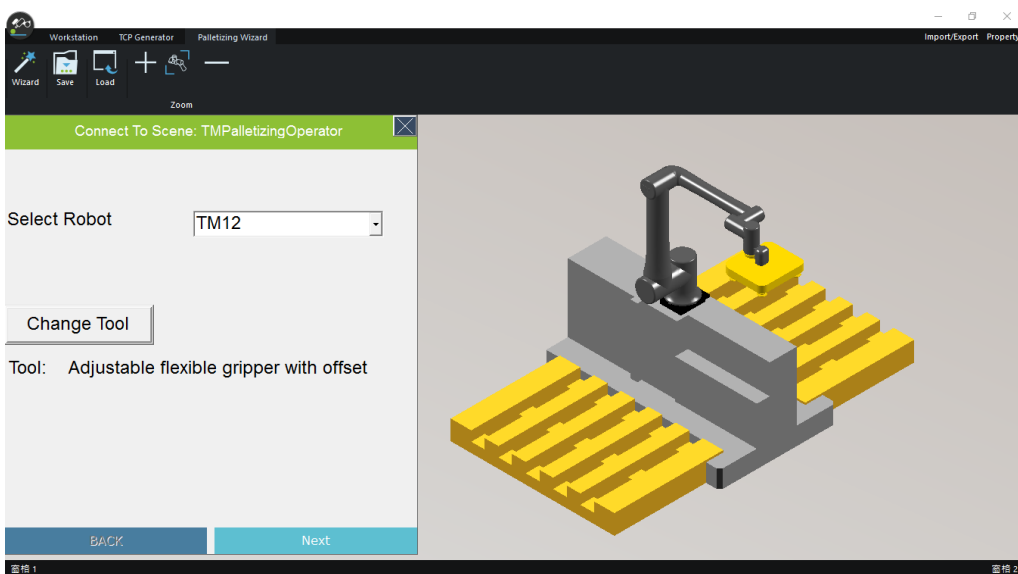
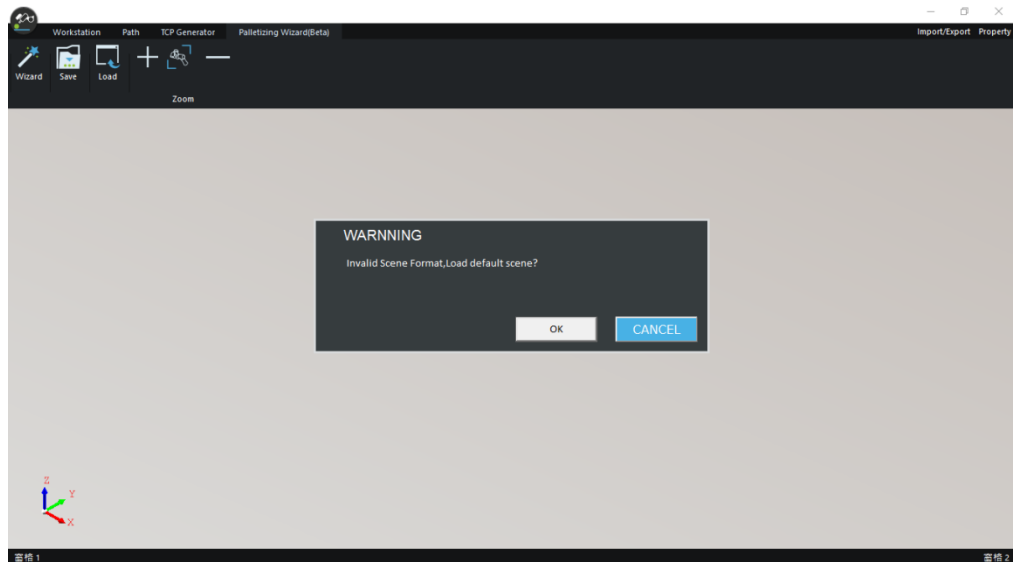
文件档案内各参数，可参考 2.3.2.9 模拟页面 – 参数设定之内容与图示。

2.3.3 将项目设定为堆栈精灵预设项目

若用户希望以其硬件作为堆栈精灵页面之预设项目，可于 TMstudio 文件夹内修改 ConfigPallet.txt 档案内的信息，为其堆栈项目。(此文件档案可透过 TMstudio 右上角按钮，进行汇入导出)。



若场景内无栈板或无场景，则直接切换至堆栈精灵点选 Wizard，会跳出以下窗口，点选 **OK** 后开启预设场景。





重要：

当堆栈精灵项目档以预设场景的方式开启时，堆栈精灵会读取 XML 内之各设定参数，如栈板大小、箱体大小、与放置向量作为默认值。因此，开发者可决定自己硬件之默认值。



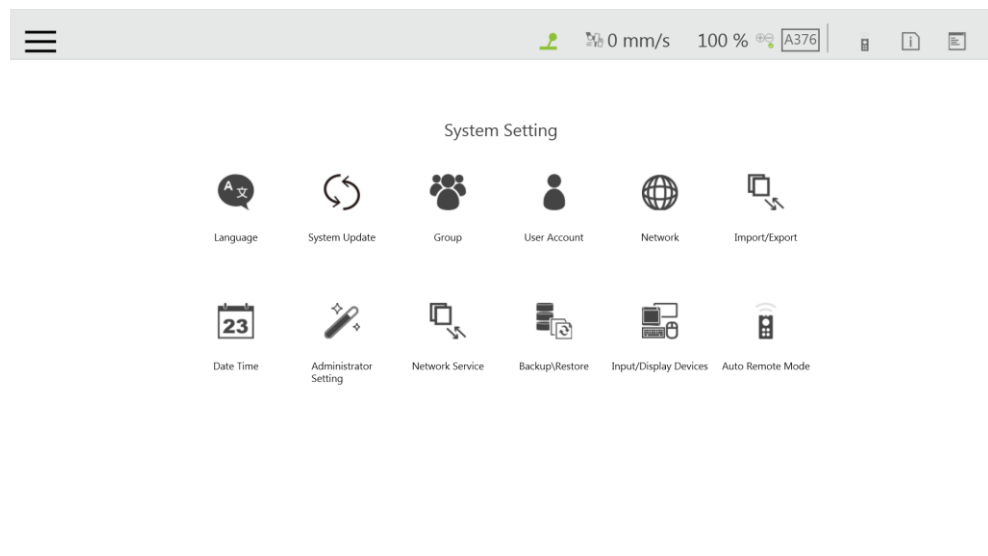
重要：

堆栈精灵项目档以预设场景方式开启时，栈板上的箱体及其摆放顺序都会被清空

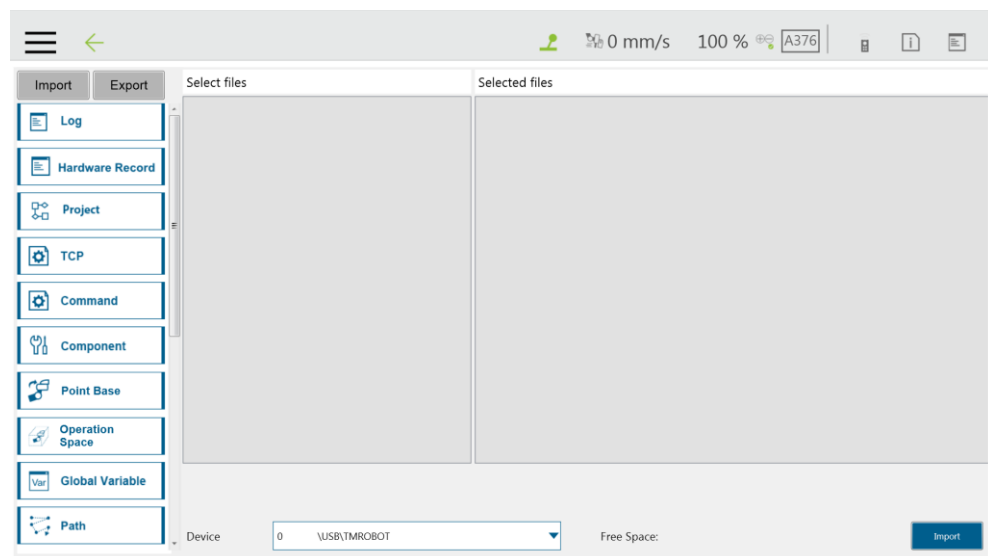
2.4 TMflow

2.4.1 汇入专案与 Text File 至 TMflow

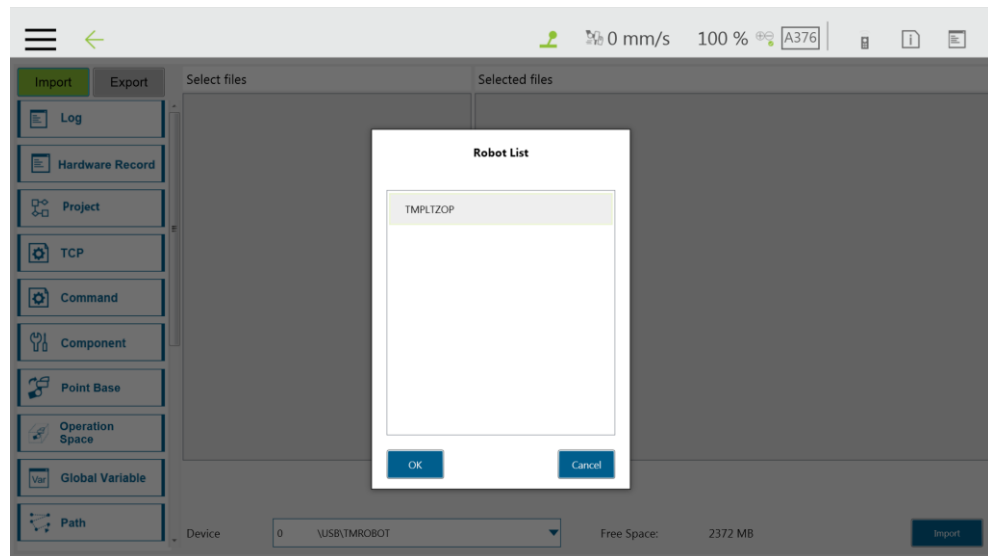
1. 于 TMflow，点击左上角的☰，再点击**系统 > 汇入/导出**。



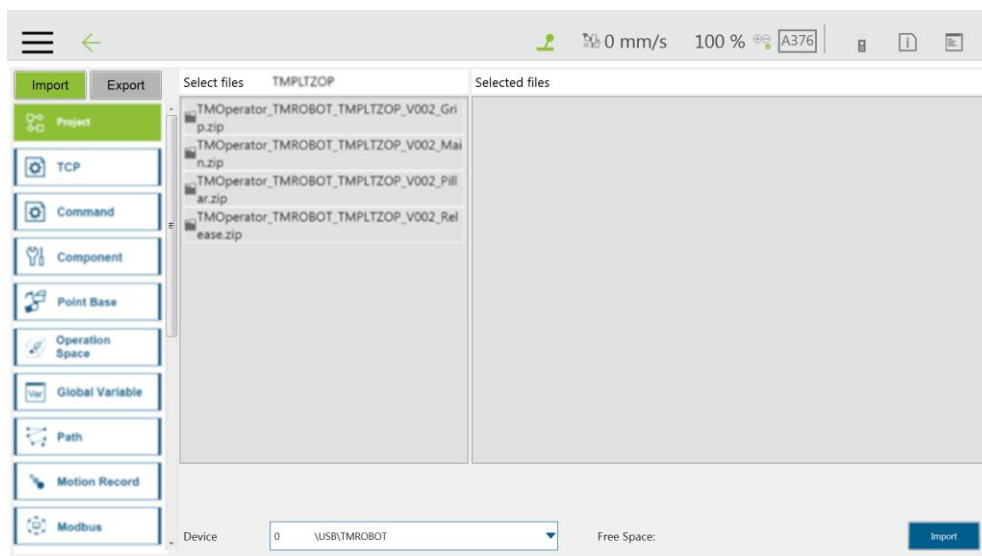
2. 点击画面左上的**汇入**。

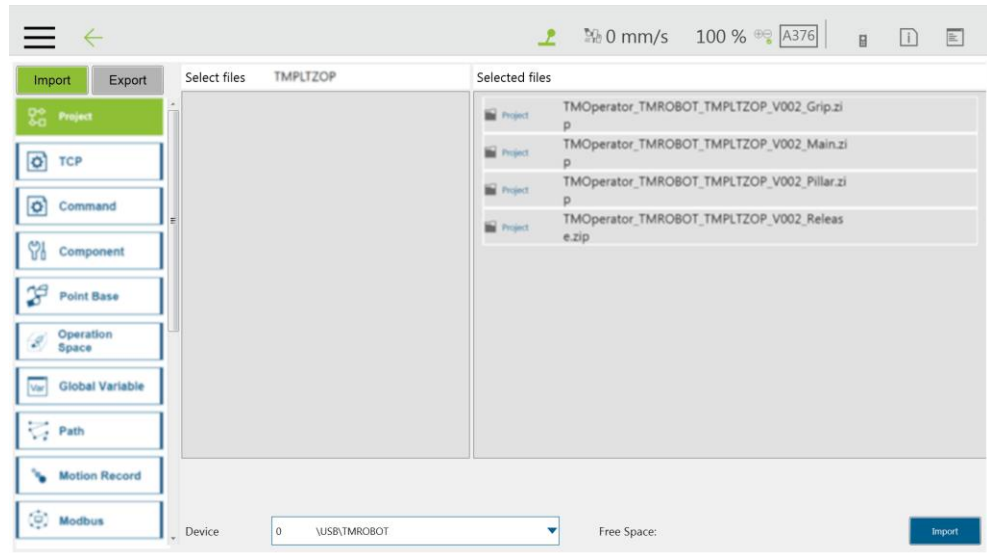


3. 从**机器人清单**提示中，选择机器人名称（于此处为 **TMPLTZOP**），再点击 **OK**。

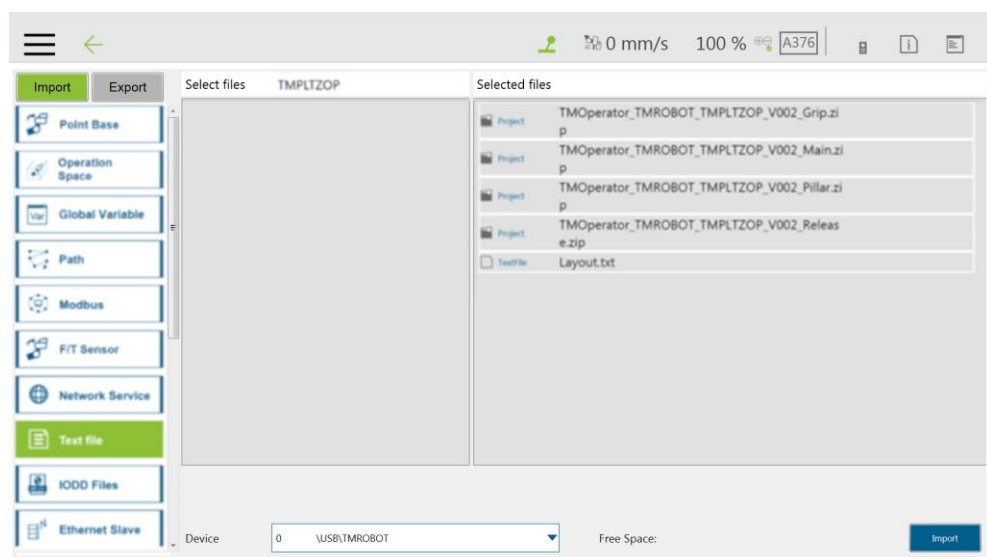
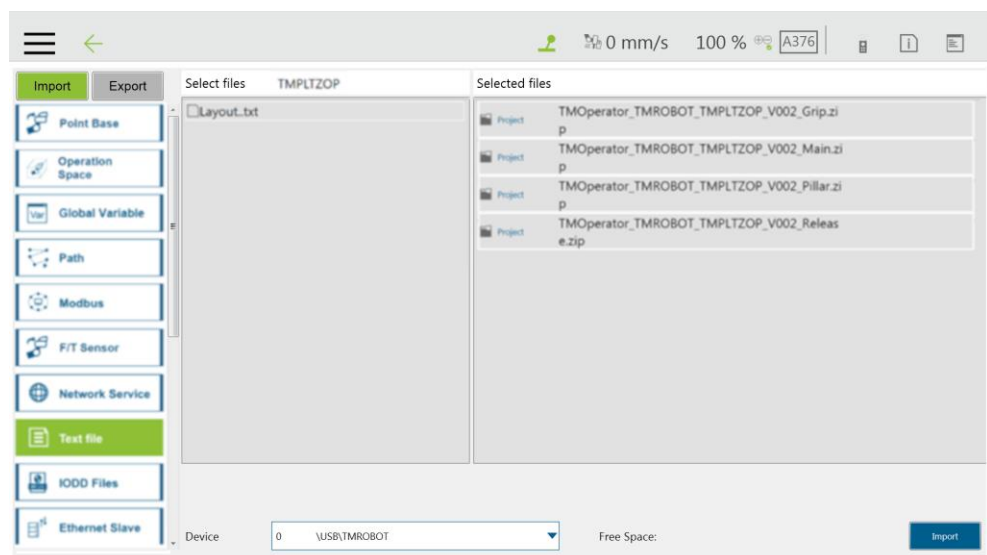


4. 点击 **Project**，点选清单上可选项之 **Project**。点选所有 **Project** 后，**Project** 会出现在**已选取清单**。





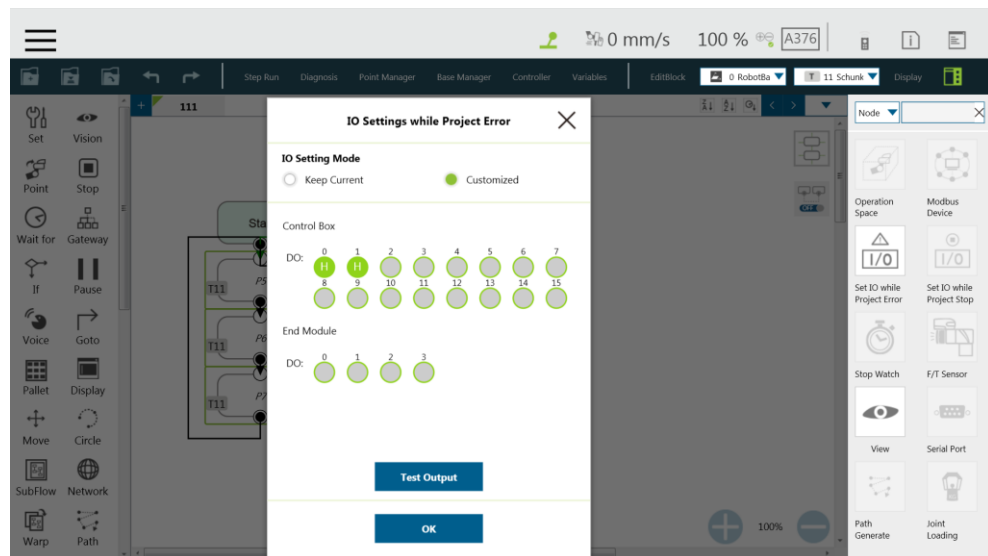
5. 汇入在堆栈精灵产出的本文文件以设定堆栈项目。点击本文文件，点选 **Layout.txt**，再点击画面右下的汇入。



2.4.2 项目停止/错误时对应之 IO 设定

使用汇入之组件前，须先设定对应之 IO 以确保立柱在项目停止或错误时暂停。请按照以下步骤设定。

1. 于 TMflow，点击建立新项目图示或点击开启旧项目图示。
2. 于右方工具侧栏，点击项目错误 I/O 设定 > 使用者自定义。
3. 在电控箱字段中，将 DO0 与 DO1 设定为 L。



4. 于右方工具侧栏，点击项目停止 I/O 设定 > 使用者自定义。
5. 在电控箱字段中，将 DO0 与 DO1 设定为 L。

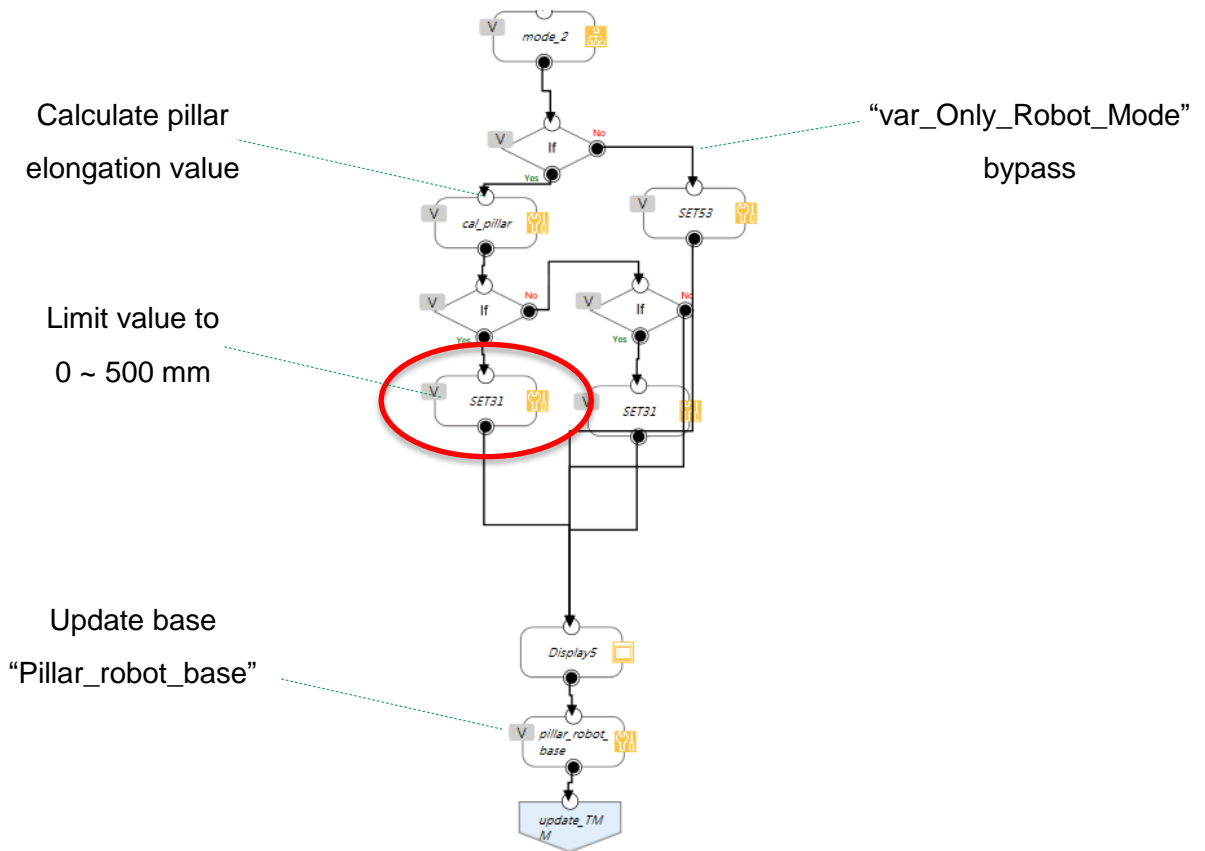


2.4.3 更换立柱 TMflow 设定

更换立柱时 TMflow 中设定

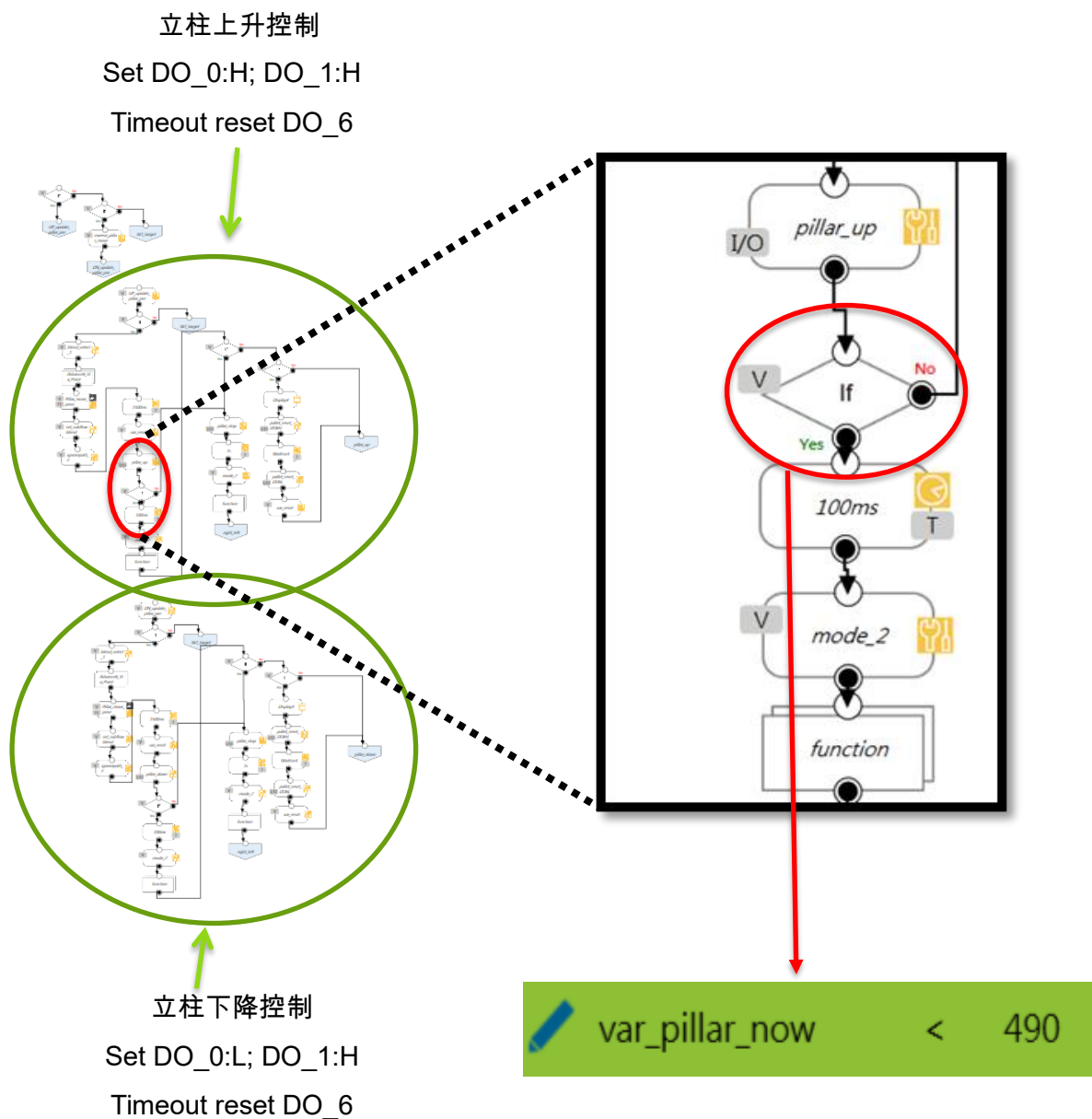
2.4.3.1 限制量测结果

副流程：function，mode_2 下方红圈所指之 SET 节点。预设值为 500mm，其数值为立柱行程，使用者仍需视状况自行调整以应用于自行设计之堆栈应用单元。



2.4.3.2 量测到伸长量

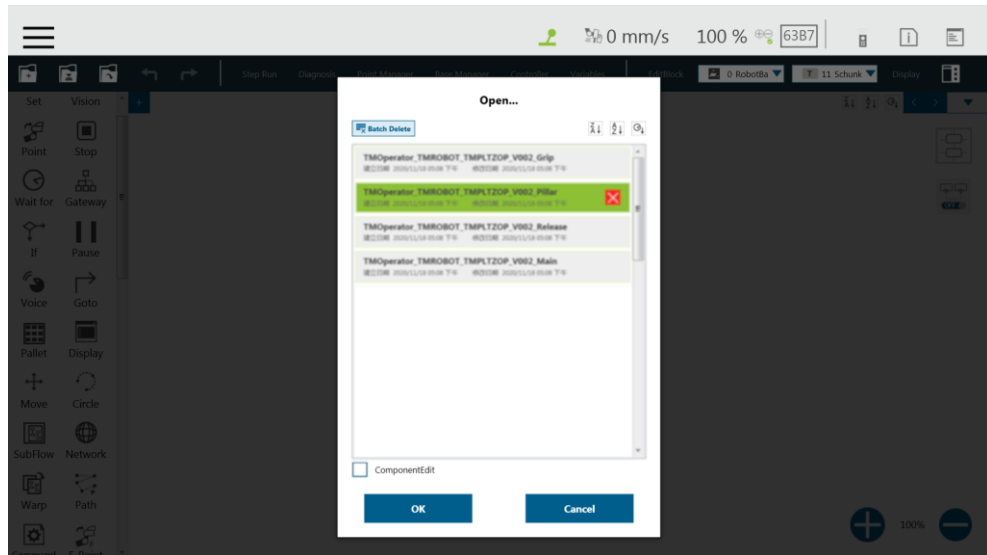
副流程：mid_put，红框中 If 节点，预设为 490mm 就停止，使用者仍需视状况自行调整以应用于自行设计之堆栈应用单元。



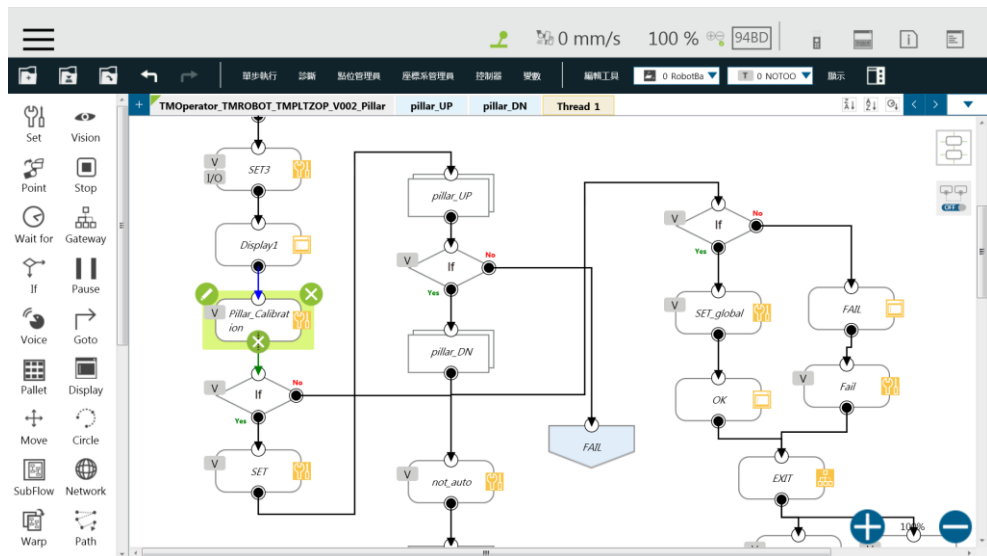
2.4.4 设定 TMOperator_TMROBOT_TMPLTZOP_V002_Pillar

进行堆栈前，请先进行立柱校正，并按照以下步骤设定。

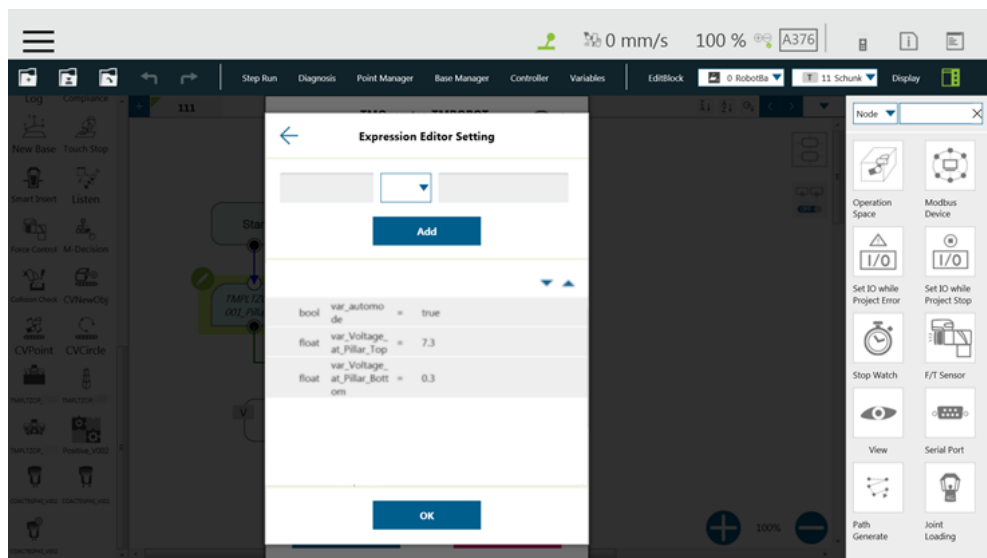
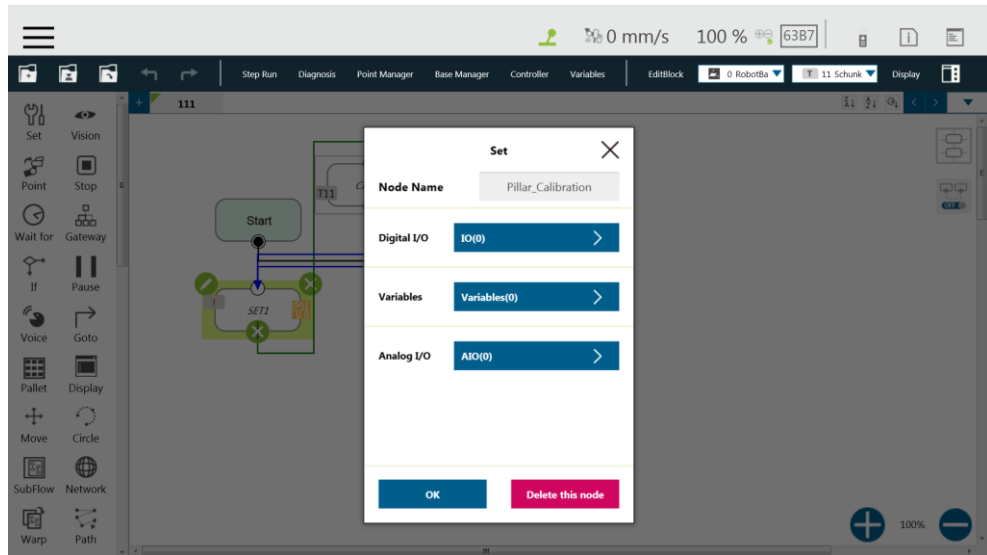
1. 开启 TMOperator_TMROBOT_TMPLTZOP_V002_Pillar 专案。



2. 点击名为 **Pillar_Calibration** 节点上的铅笔图示。



3. 点击 **Variables** 进入设定。确认 **bool** 值为 **true**。

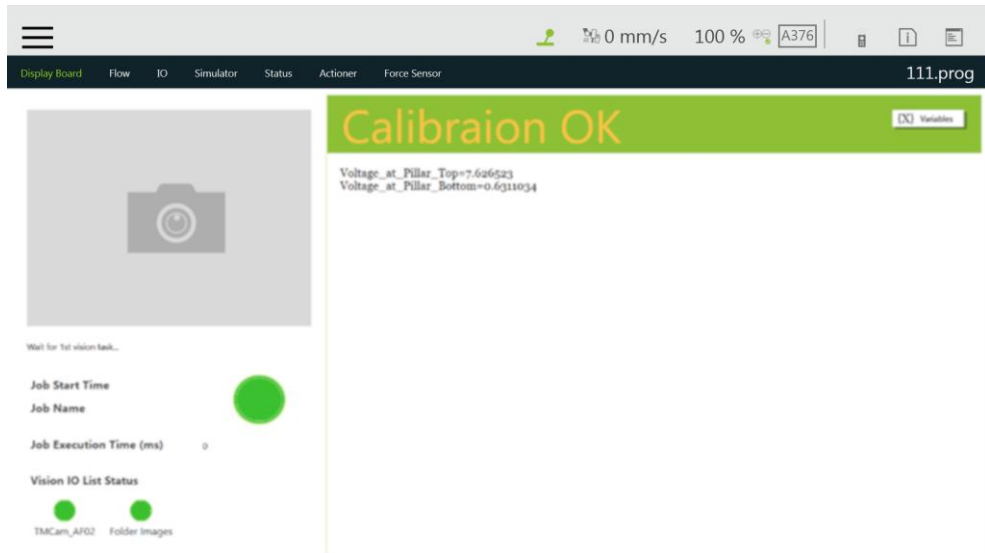


Note

提示：

当 **bool** 值为 **true** 时，立柱将执行自动更正。若 **bool** 为 **false**，则立柱将根据其它变量设定中的用户输入值为立柱最高点及最低点对应之电压值。

4. 校正测试。按下示教器上的 **PLAY/PAUSE** 钮开始测试。若成功，画面会出现 **Calibration OK** 讯息与相关电压讯息。测试后如需退出，可按 **Stop** 钮。



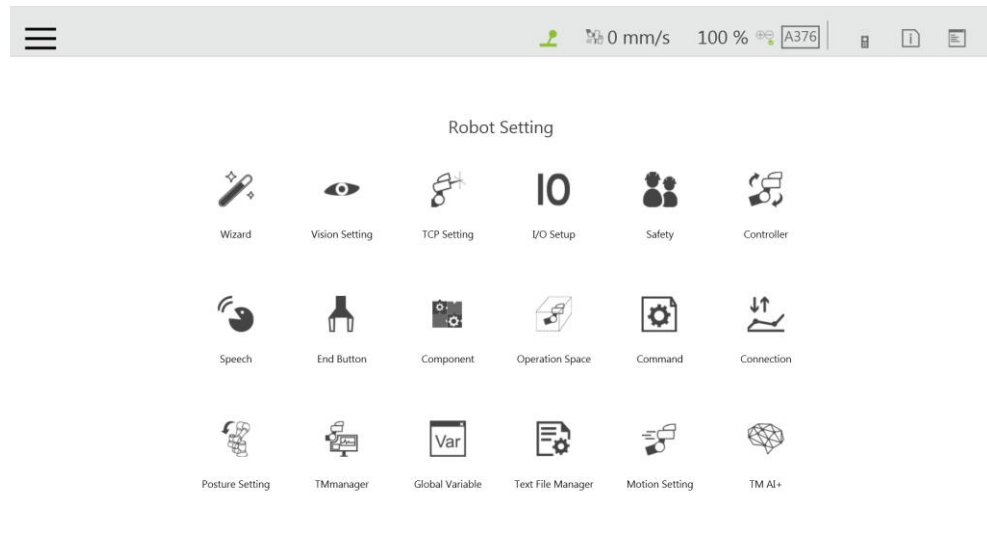
危险：

- 进行立柱校正时，请确保立柱不受外力干扰或碰撞。
- 请使用立柱校正项目来确保立柱电压值及对应的立柱高度正确。

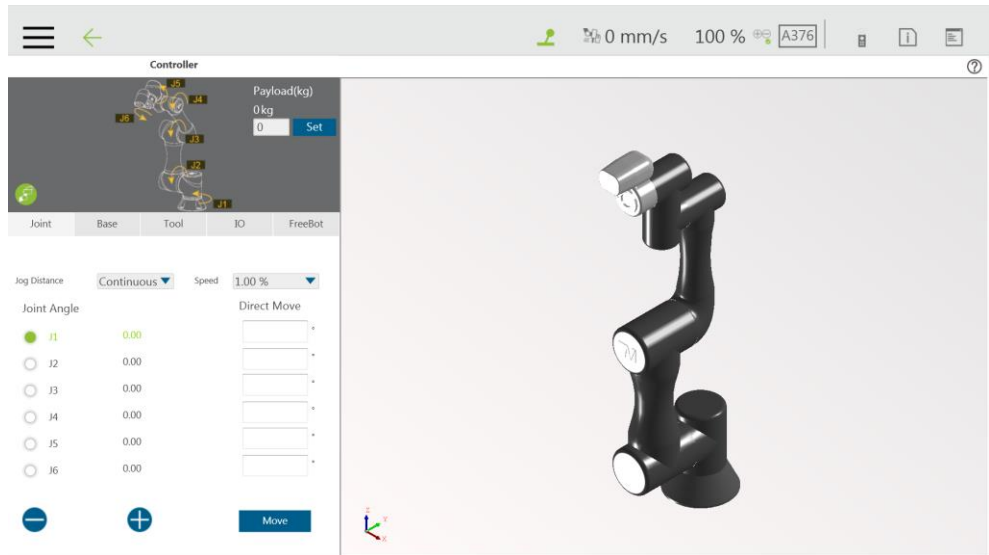
2.4.4.1 降低立柱高度

进行视觉相关任务编程时，必须先降低立柱到最低高度。步骤如下：

1. 于 TMflow，点击☰再点击**设定 > 控制器**。



2. 点击**输出**页签。



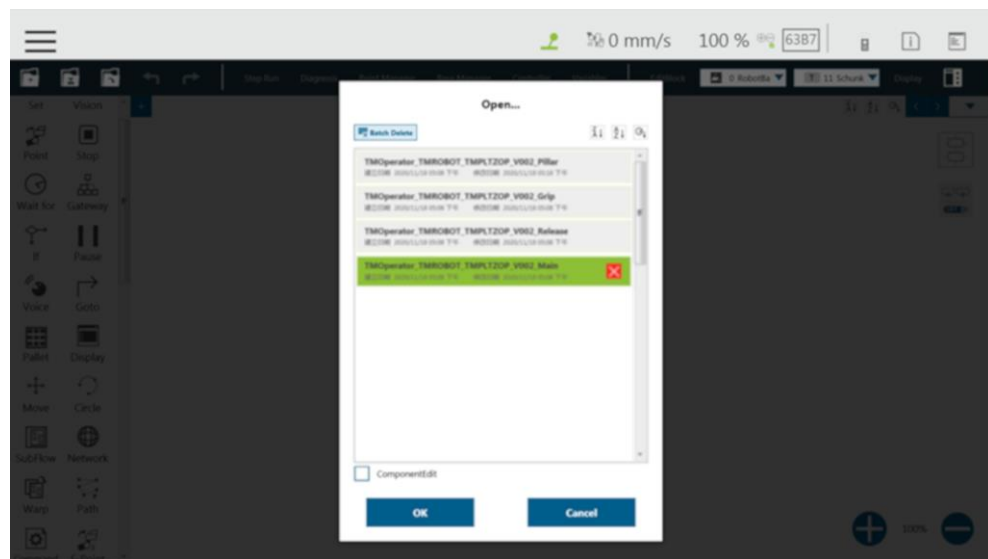
3. 在**电控箱**以下对应处进行设定：

- 设 DO_6 为 H (high) 之后停顿一下，再设 DO_6 为 L (low)，使立柱重置。
- 设 DO_0 为 L
- 设 DO_1 为 H，使立柱降到最低。

2.4.5 设定 TMOperator_TMROBOT_TMPLTZOP_V002_Main

以下为设定 TMOperator_TMROBOT_TMPLTZOP_V002_Main 项目之步骤与选项，此项目可针对堆栈所需参数进行主要设定，并执行堆栈应用。请按照以下步骤设定：

1. 于开启旧选单中，开启 **TMOperator_TMROBOT_TMPLTZOP_V002_Main** 专案。



此项目提供 4 个快速步骤设定主要堆栈任务。

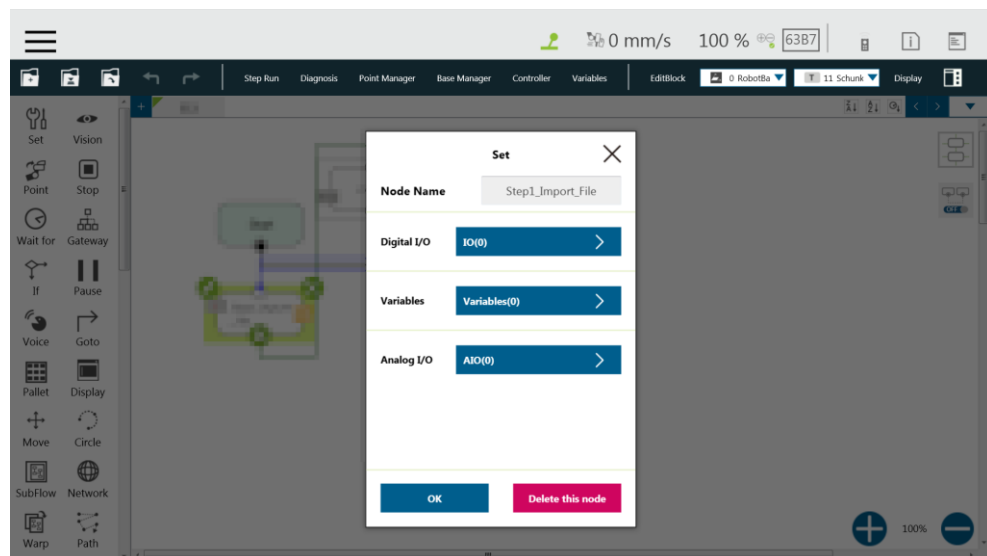
- **Step 1 – Import File**

- Step 2 – Recognizing Box Position
- Step 3 – Teach Gripping Point
- Step 4 – Visual Recognitions

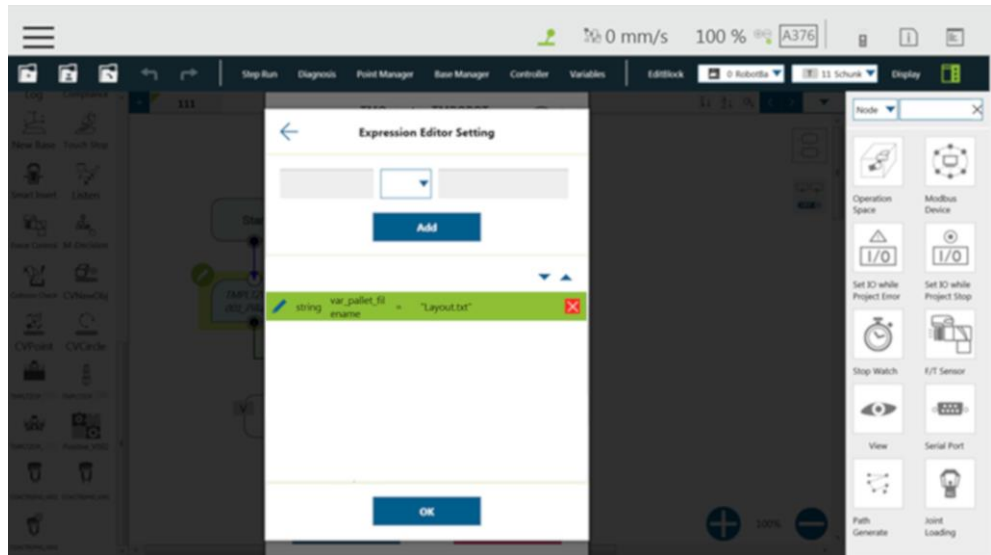


2.4.5.1 Step 1 – Import File

1. 点击 **Step1_Import File** 然后点击 **Variables**.



2. 于**表达式逻辑器**设定，点击于 2.4.1 汇入之本文文件名称（预设档名为 Layout.txt）。然后点击**确认**。



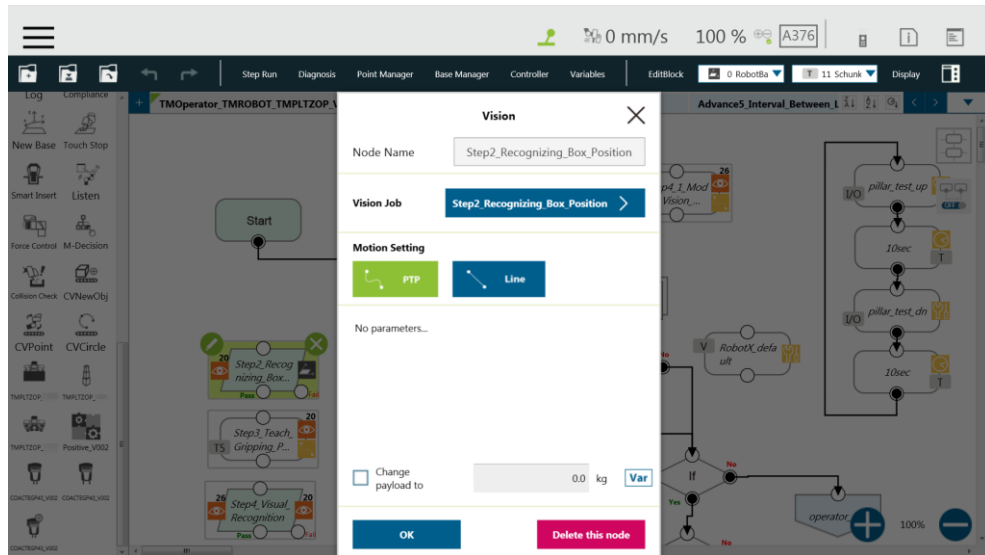
以下步骤 2 至步骤 4 会有 box1 与 box2 两个选项，若两个栈版的箱体相同，只需设定 box1，反之则需要再设定 box2。因程序相同，以下步骤仅以 box1 示意说明。

2.4.5.2 Step 2 – Recognizing Box Position

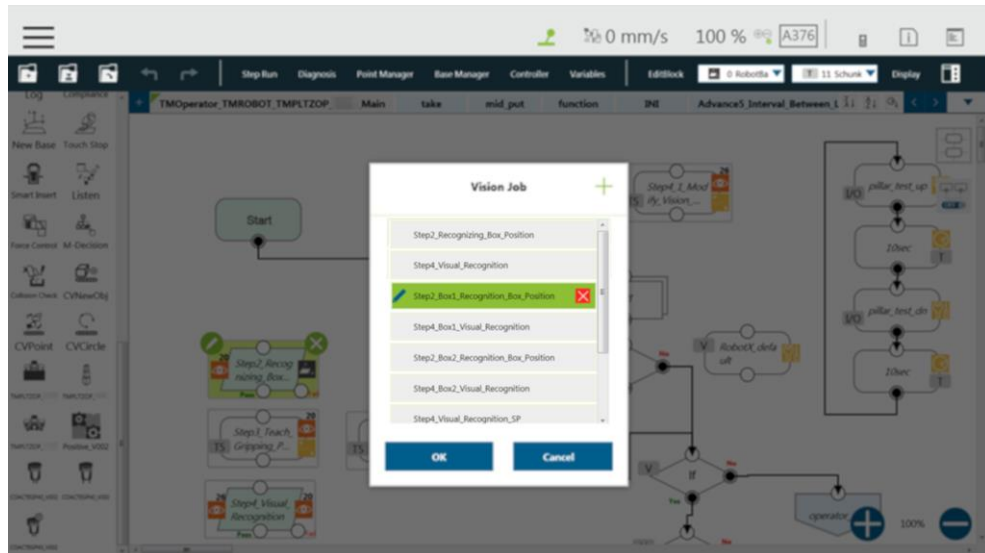
1. 降低立柱到最低高度。相关操作请参考 2.4.4.1 降低立柱高度。
2. 选择红框中 **Step2_Box1_Recognizing_Box_Position** 并点击节点上的铅笔图示。



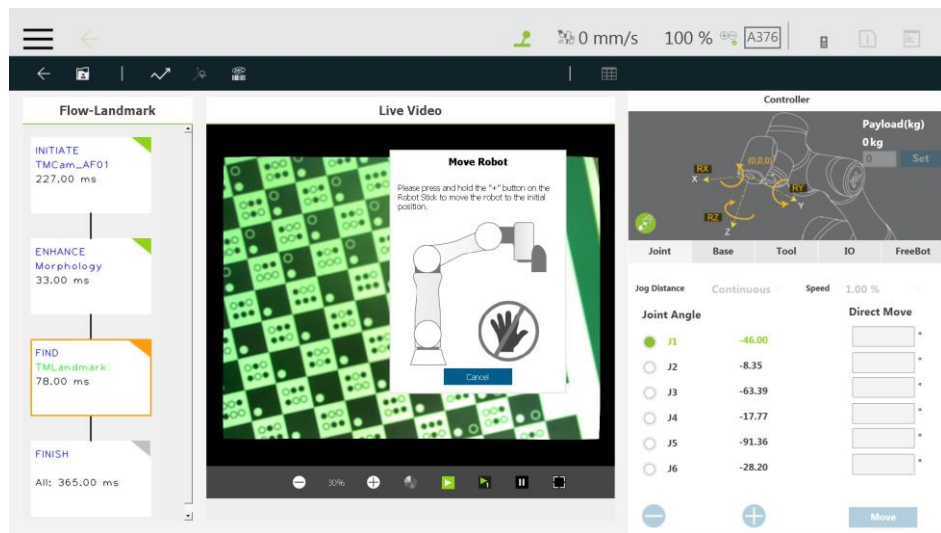
3. Click **视觉任务 > Step2_Box1_Recognizing_Box_Position**.



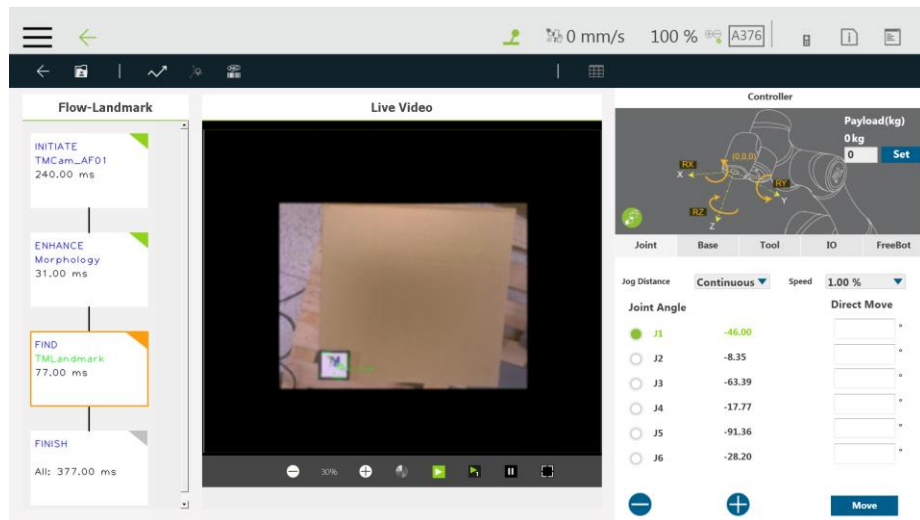
4. 于清单中，点击 **Step2_Box1_Recognizing_Box_Position** 之铅笔图示。



5. 进入 TMvision 编辑流程后：
- A. 若先前已教导初始位置，则可以按压示教器之+钮，直到机器人抵达初始位置。
 - i. 先将箱子放好，并于箱子左下角放上 TM Landmark。
 - ii. 确保摄影机可以看到 TM Landmark，同时此位置适合作为吸取箱子时的初始点位。
 - B. 若为第一次使用尚未教导初始位置，应点选取消，进行初始位置的教导。
 - i. 先将箱子放好，并于箱子左下角放上 TM Landmark。
 - ii. 手拉机器人到每次取箱子时的初始位置，同时确保摄影机可以看到 TM Landmark。
 - iii. 决定位置后，并进到 **Initiate** 里设定初始位置，点选 **Reset Initial Position**，将此位置设定为初始位置。



6. 机器人辨识 TM Landmark 后，点击左上角储存图示。



提示：

若机器人因为画面清晰度而无法侦测到 TM Landmark，可点击 **Initiate > Edit**，进行摄影机参数调整。

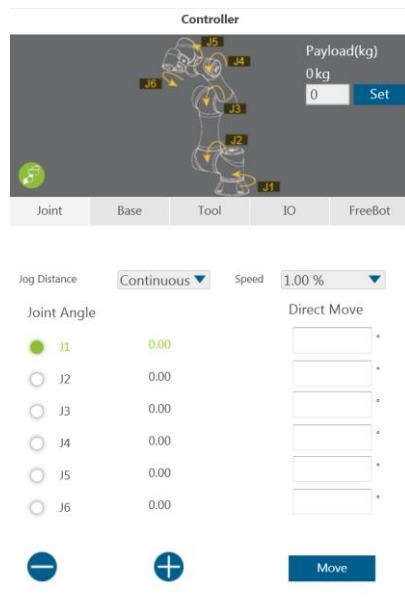


重要：

于此步骤中，请勿移动或触摸箱体。否则识别可能会失败。

2.4.5.3 Step 3 – Teach Gripping Point

1. 降低立柱到最低高度。相关操作可参照 2.4.4.1 降低立柱高度。
2. 于 TMflow 使用以控制器页面（强烈建议）或机器人之 FREE 按钮，引导夹爪到夹取点。



危险：

当以 FREE 按钮释放无电动力之煞车时，机器人可能会因重力而下垂。请准备好支撑机器人。如果发现无法支撑机器人，请立即释放 FREE 按钮，此将锁定机器人关节制动器。

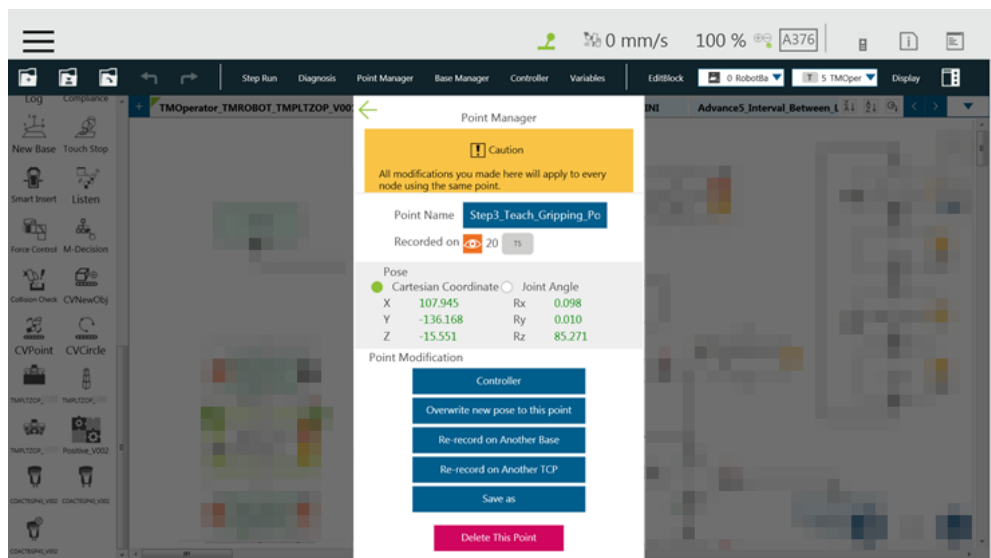
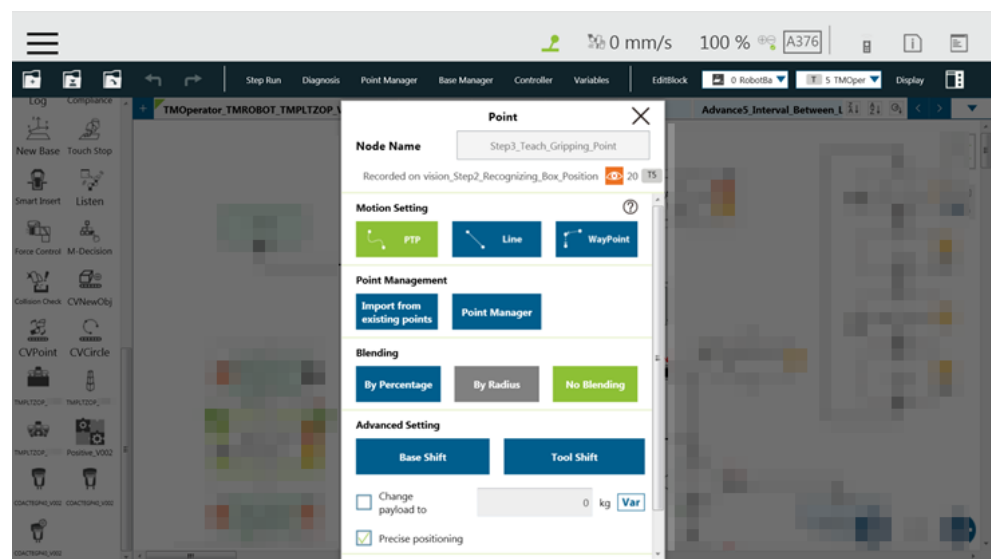
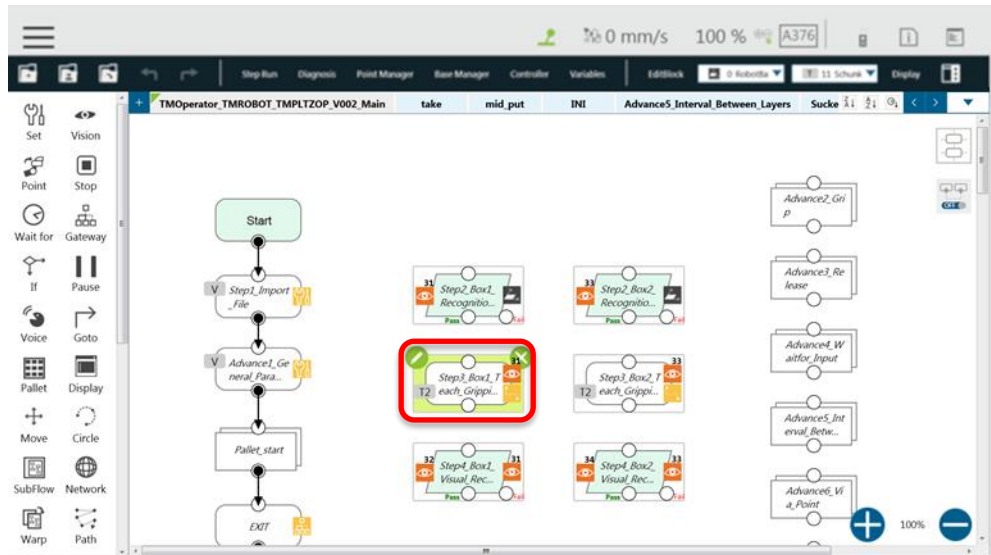


重要：

于此步骤中，请勿移动或触摸箱体。否则识别可能会失败。



3. 选择 **Step3_Box1_Teach_Gripping_Point** 并点击节点上的铅笔图示 > **点位管理员** > 将目前的姿态写入本点位 > **确认**，以设定并记入夹取点。

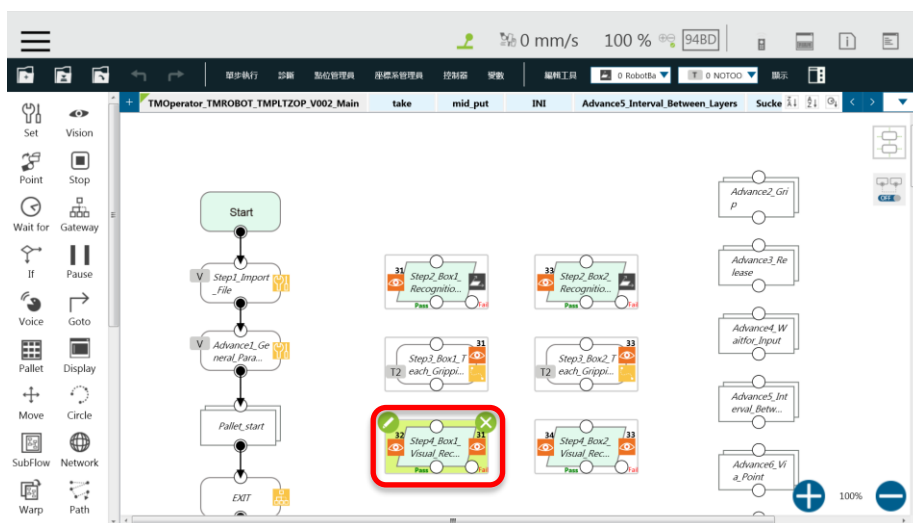


2.4.5.4 Step 4 – Visual Recognition

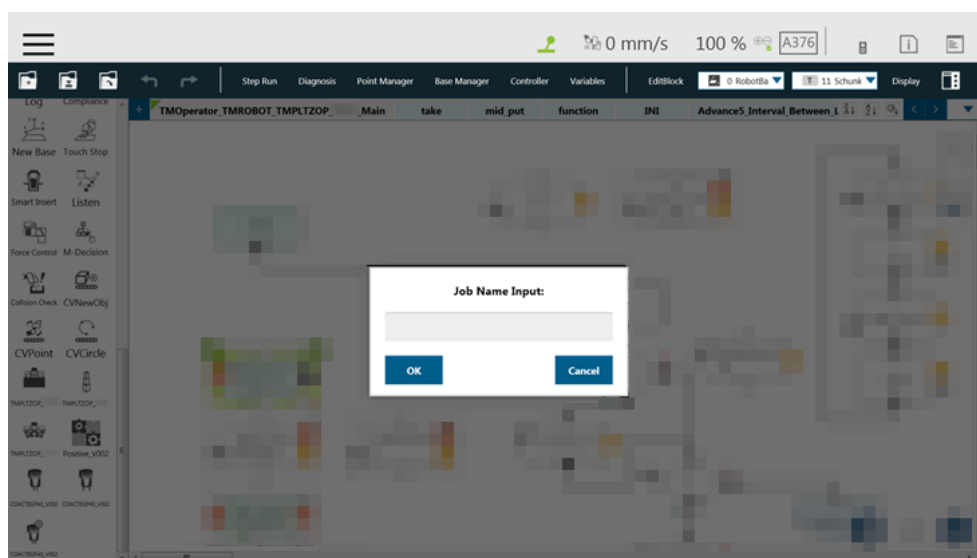
此步骤应用视觉作业来补偿产生偏移的箱体。若无须此功能可于进阶设定中关闭使用视觉进行

定位。若须更换夹取的箱体、或重新教导工作平面，请继续以下步骤：

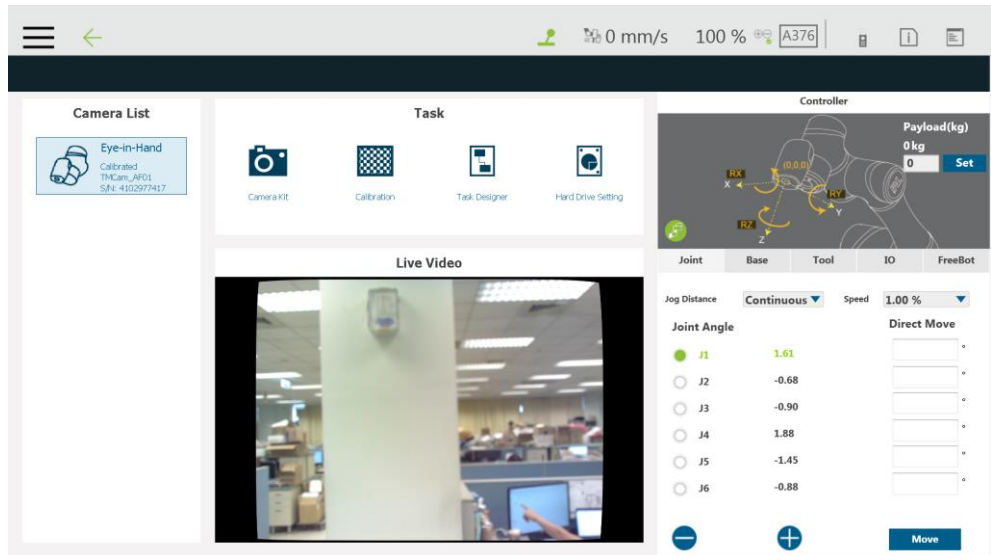
1. 降低立柱到最低高度。相关操作请参照 2.4.4.1 降低立柱高度。
2. 点击 **Step4_Box1_Visual_Recognition** 节点上的铅笔图示。



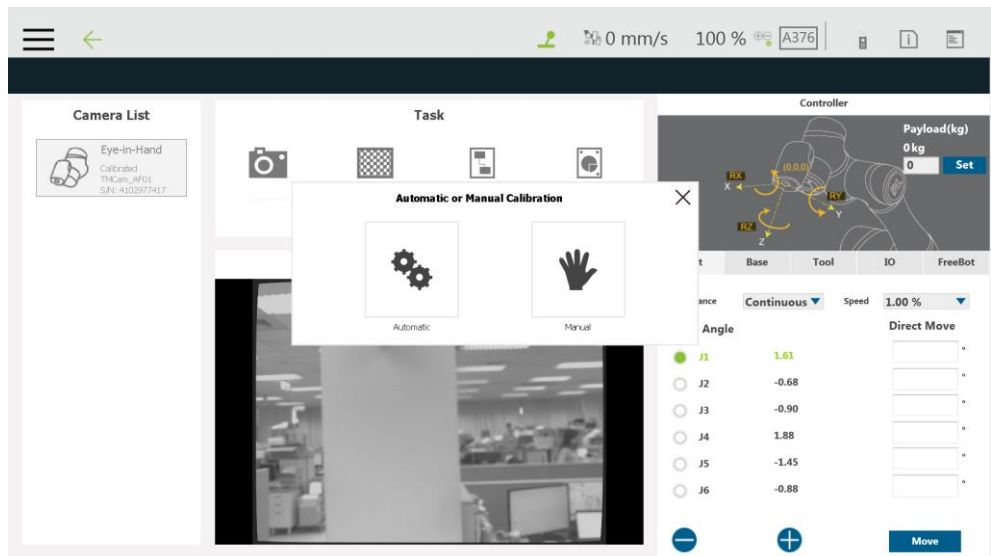
3. 点击视觉任务 > **Step4_Box1_Visual_Recognition**，再点击右上角的+。
4. 于视觉任务名称字段输入名称，再点击确认。



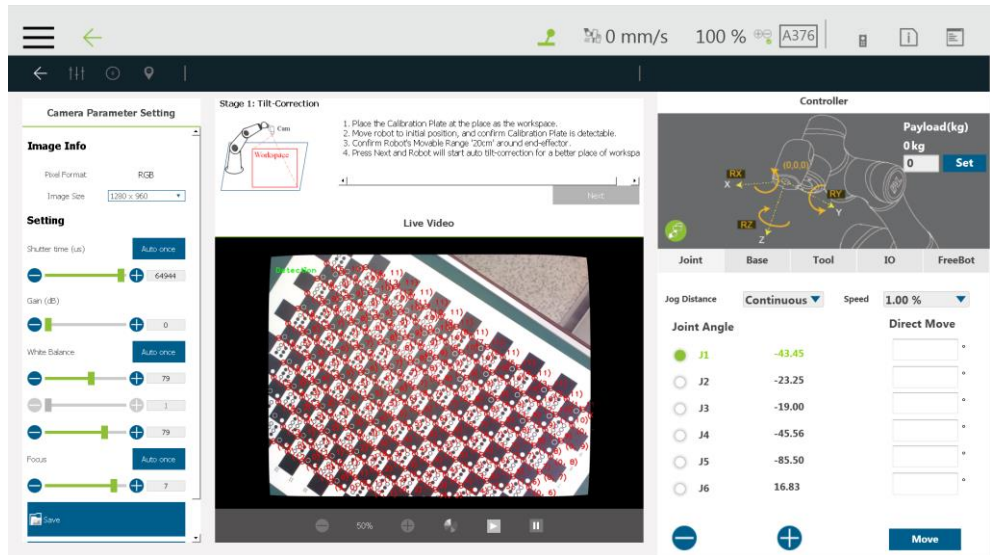
5. 进入视觉任务后，将校正版放置于机器人摄影机下，然后点击手眼校正。



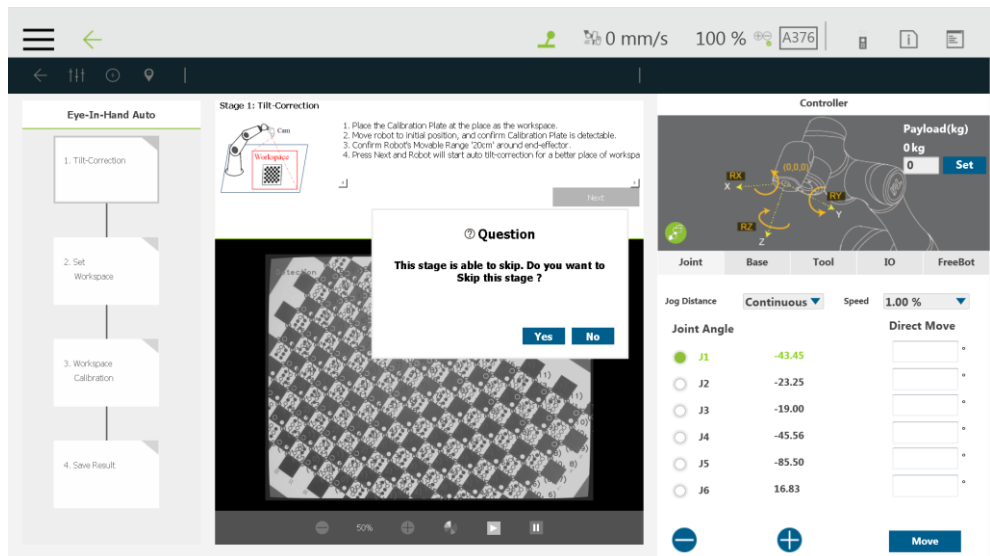
6. 点击自动 (建议), 以缩短教导流程。



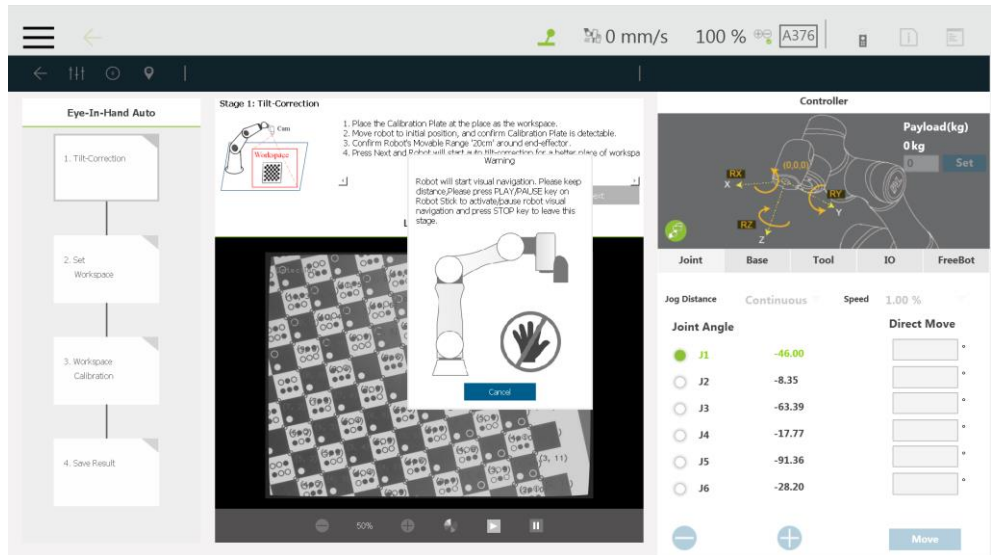
7. 点击左上角的摄影机参数调整图示。根据周遭因素或应用需求设置最佳快门时间、白平衡、与焦距。使用者也可选择自动调整让机器人摄影机侦测并自行调整参数以缩短设定流程。



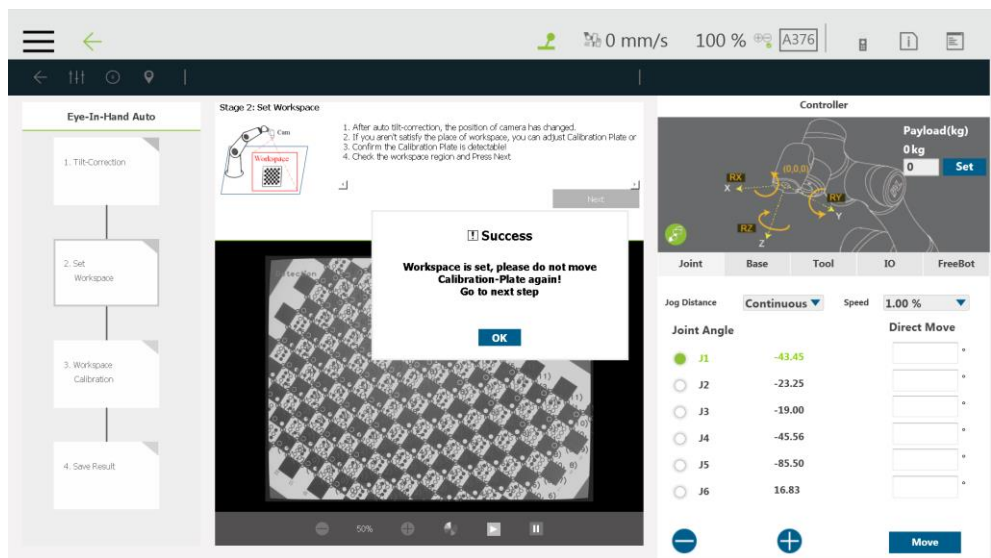
8. 于步骤 1：倾斜校正，确定画面中之说明都确实遵守。然后点击下一步。



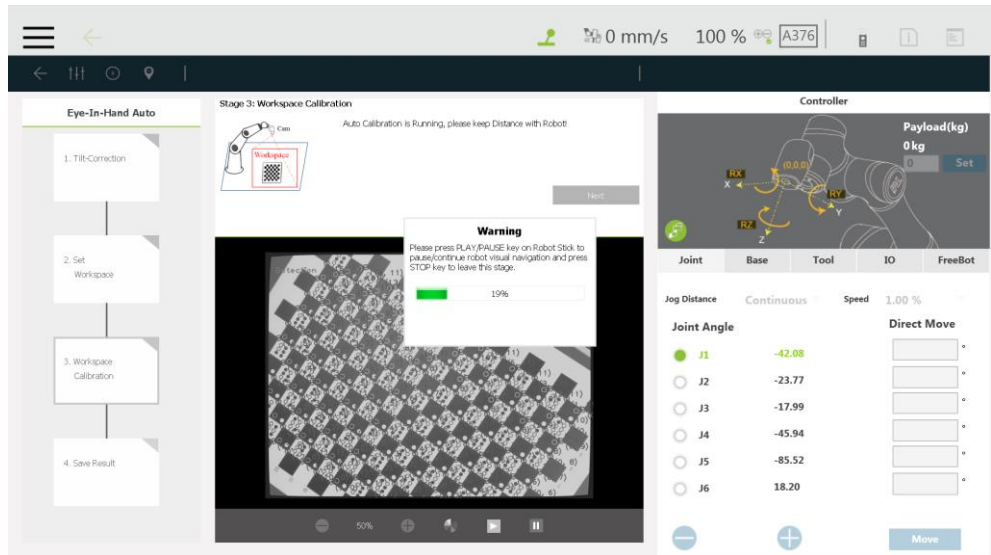
- 于画面提示时，选是，则略过自动倾斜校正，直接进入下一步。
- 于画面提示时，选否，将开始自动倾斜校正。建议选此选项确保所有后续步骤能确实执行。使用者将被要求轻触示教器上的 **PLAY/PAUSE** 钮，同时机器人会自动视觉导航并校正。



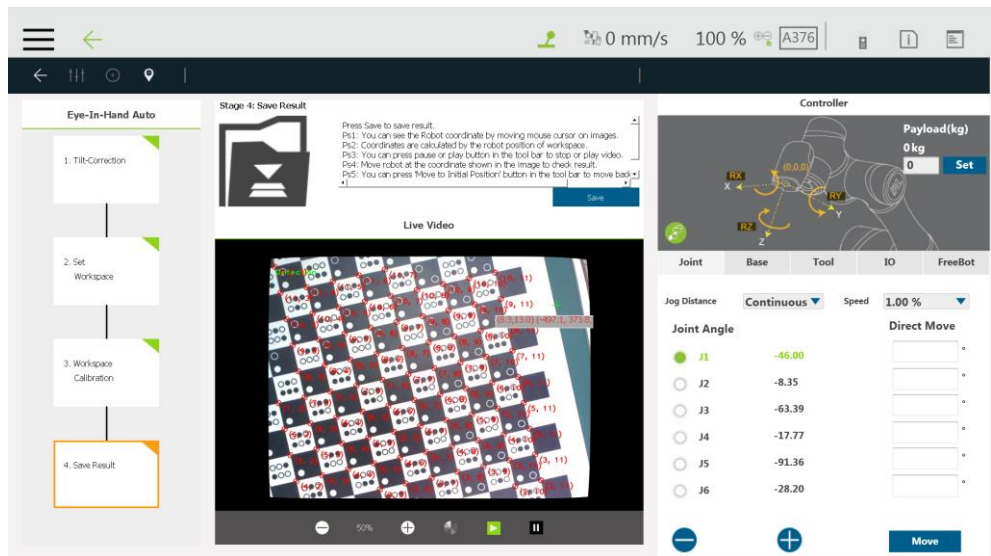
9. 于**步骤 2**：设置工作平面机器人会自动设置工作平面。请确认校正版仍维持原位。然后在成功讯息出现后，点击**确定**。



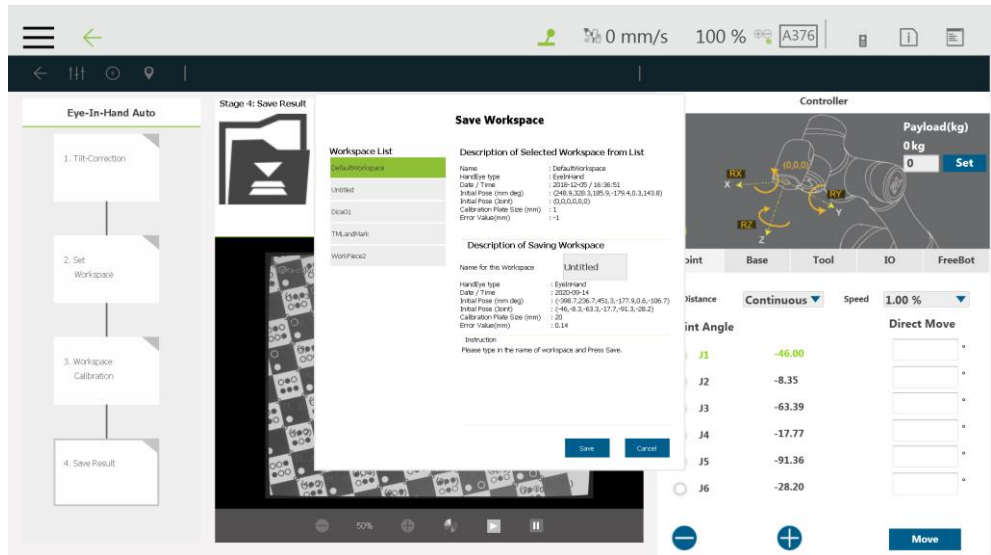
10. 于**步骤 3**：工作平面校正，轻触示教器上 **PLAY/PAUSE** 钮开始视觉导航。机器人会开始校正工作平面。



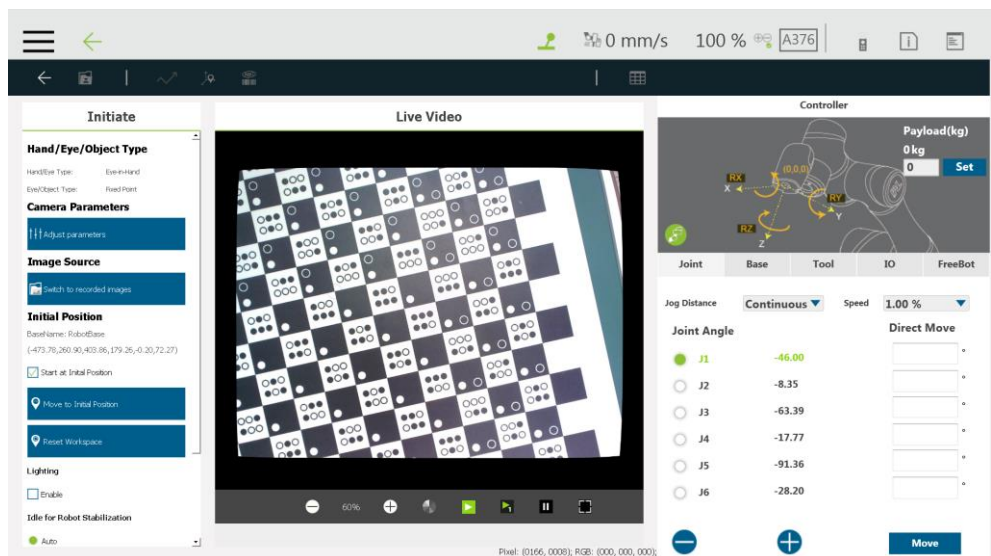
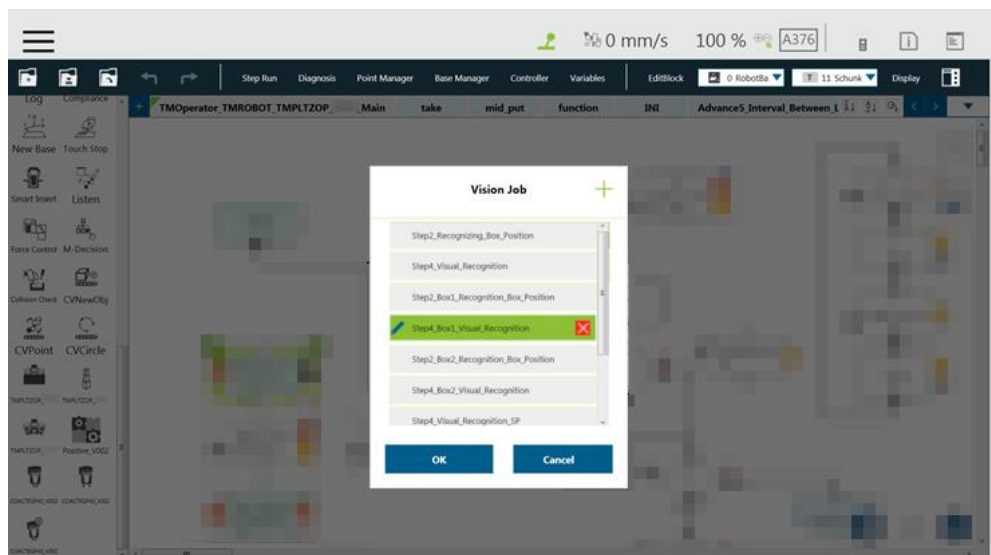
11. 于步骤 4： 储存结果，完成校正后，点击储存。



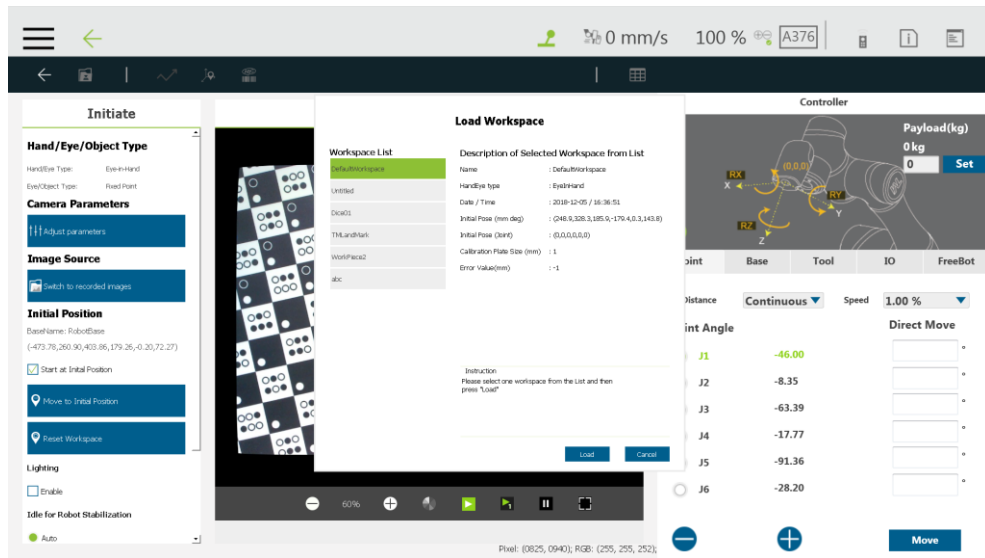
12. 点击储存结果后开启储存工作平面窗口。输入新增之工作平面名称，然后点击储存。



13. 回到视觉任务窗口后，点击 **Step4_Box1_Visual_Recognition** 之铅笔图示，然后点击启动。



14. 点击重新设定工作平面。于读取工作平面窗口出现时，选择需求之工作平面并点击读取。



2.4.5.5 传感器输入注意事项

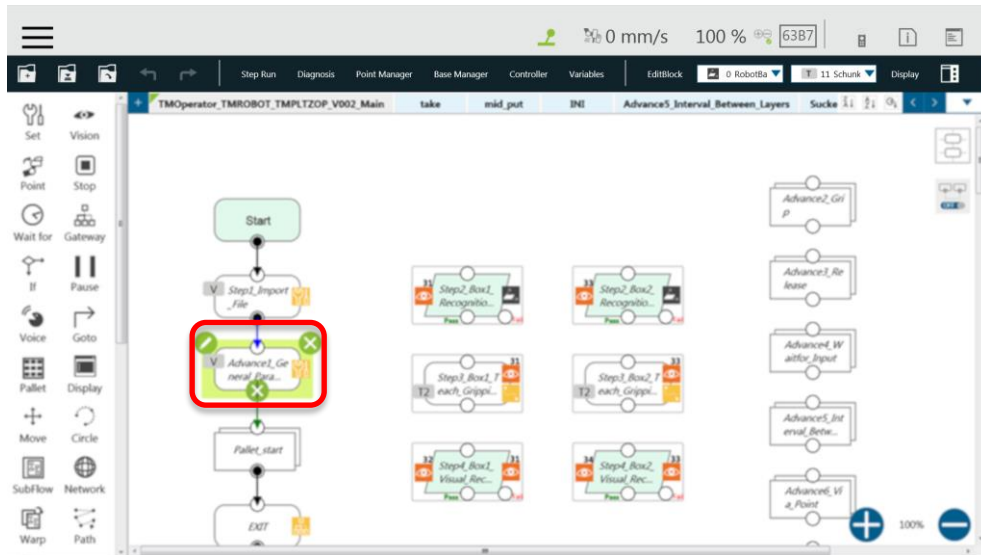
请将输入带上的传感器输出连接至 DI14，以便 DI14 触发为 High 时，通知机器人箱体已达定位，可进行夹取。

2.4.6 设定 TMOperator_TMROBOT_TMPLTZOP_V002_Grip 与 TMOperator_TMROBOT_TMPLTZOP_V002_Release

以上项目为弹性调整式夹爪的夹取和释放，若您在 Advanced2-8 中自行插入流程需使用到该夹爪，可于流程中插入此组件进行使用。

2.5 进阶堆栈设定

TM Palletizing Operator 为用户提供了进阶设置选项，以实现更灵活的应用程序。以下介绍进阶设定的相关功能：



2.5.1 Advanced1_General_Parameters

#	名称	类型	预设	叙述
1	Use_Vision_or_Not	Bool	TRUE	True: 按照视觉任务夹取箱体 False: 按照固定位置夹取箱体
2	Control_Mode	Int[]	{1,1}	Control_Mode = {M, N} M = 1, 使用 TMstudio 设定; 忽略参数 N M = 0, 使用 TMflow 输入值 N N = 控制模式设定 [1~3](1: palletizing; 2: de-palletizing; 3: change pallet)
3	Box_Sum	Int	0	单一专案运行中夹住的箱体总数.达到定义的总和后, 软件将通过“确定”节点退出. 0 = No limit
4	Start_Direction	Int	1	1: 从右侧栈板开始堆栈 2: 从左侧栈板开始堆栈
5	Pallet1_Start_LayerIndex	Int []	{1,1}	Pallet1_Start_LayerIndex = {M, N} 使用 M 层中的 Box N 从右侧托盘开始堆垛 M= 从右侧的栈板开始的层数 N= 对应层数的索引
6	Pallet2_Start_LayerIndex	Int []	{1,1}	Pallet2_Start_LayerIndex = {M, N} 使用 M 层中的 Box N 从左侧托盘开始堆垛 M= 从右侧的栈板开始的层数 N= 对应层数的索引
7	Retry_Count_Depth	Int []	{3,3}	Retry_Count_Depth = {M, N} M =无法夹取箱体重试的次数 N = 每次重试时高度下降(mm)
8	Box1_Weight	Float[]	{1,5}	Box1_Weight = {M, N} M = 1, 使用 TMstudio 设定; 忽略参数 N M = 0, 使用 TMflow 输入值 N N = 箱体重量 (kg)
9	Box2_Weight	Float[]	{1,5}	Box2_Weight = {M, N} M = 1, 使用 TMstudio 设定; 忽略参数 N

#	名称	类型	预设	叙述
				M = 0 , 使用 TMflow 输入值 N N = 箱体重量 (kg)
10	Pallet1_Elevated_Height	Float	{1,200}	Pallet1_Elevated_Height = {M , N} M = 1 , 使用 TMstudio 设定; 忽略参数 N M = 0 , 使用 TMflow 输入值 N N = 夹取箱体后升高的高度 (在 Z 方向)
11	Pallet2_Elevated_Height	Float	{1,200}	Pallet2_Elevated_Height = {M , N} M = 1 , 使用 TMstudio 设定; 忽略参数 N M = 0 , 使用 TMflow 输入值 N N = 夹取箱体后升高的高度 (在 Z 方向)
12	Pallet1_Via_Point_Offset	Float []	{0,-400,-400,0}	Pallet1_Via_Point_Offset = {M , X , Y , Z} M = 1 , 使用 TMstudio 设定; 忽略参数 X , Y , Z M = 0 , 使用 TMflow 输入值 X , Y , Z X , Y , Z = 偏移的 X , Y , Z 方向与 Elevated_Height 关联
13	Pallet2_Via_Point_Offset	Float []	{0,-400,400,0}	Pallet2_Via_Point_Offset = {M , X , Y , Z} M = 1 , 使用 TMstudio 设定; 忽略参数 X , Y , Z M = 0 , 使用 TMflow 输入值 X , Y , Z X , Y , Z = 偏移的 X , Y , Z 方向与 Elevated_Height 关联
14	Pallet1_Approach_XY_Z	Float []	{1,30,300}	Pallet1_Approach_XY_Z = {M , X , Y , Z} M = 1 , 使用 TMstudio 设定; 忽略参数 X , Y , Z M = 0 , 使用 TMflow 输入值 X , Y , Z X , Y , Z = 箱体放置时的 X , Y , Z (mm) 接近向量
15	Pallet2_Approach_XY_Z	Float []	{1,30,300}	Pallet2_Approach_XY_Z = {M , X , Y , Z} M = 1 , 使用 TMstudio 设定; 忽略参数 X , Y , Z M = 0 , 使用 TMflow 输入值 X , Y , Z X , Y , Z = 箱体放置时的 X , Y , Z (mm) 接近向量
16	Pallet1_Elevated_Height_Before_Placing	Float[]	{1,100}	Pallet1_Elevated_Height_Before_Placing = {M , N} M = 1 , 使用 TMstudio 设定; 忽略参数 N M = 0 , 使用 TMflow 输入值 N N = 放置箱体后下降的高度(mm)
17	Pallet2_Elevated_Height_Before_Placing	Float[]	{1,100}	Pallet2_Elevated_Height_Before_Placing = {M , N} M = 1 , 使用 TMstudio 设定; 忽略参数 N M = 0 , 使用 TMflow 输入值 N N = 放置箱体后下降的高度(mm)
18	Pallet1_Pillar_Height	Int[]	{1,.....}	Pallet1_Pillar_Height = {M , H1 , H2 , H3..... } M = 1 , 使用 TMstudio 设定; 忽略参数 H M = 0 , 使用 TMflow 输入值 H H = 放置在右栈板上每一层的对应于升高支柱的相对高度 (mm)
19	Pallet2_Pillar_Height	Int[]	{1,.....}	Pallet2_Pillar_Height = {M , H1 , H2 , H3..... } M = 1 , 使用 TMstudio 设定; 忽略参数 H M = 0 , 使用 TMflow 输入值 H H = 放置在右栈板上每一层的对应于升高支柱的相对高度 (mm)
20	Pallet1_Offset	Float []	{0,0,0,0,0,0}	Pallet1_Offset = {X , Y , Z , RX , RY , RZ }

#	名称	类型	预设	叙述
				X, Y, Z, RX, RY, RZ = 右栈板 X, Y, Z, RX, RY, RZ 位置偏移量的修整值 (mm, mm, mm, 度, 度, 度)
21	Pallet2_Offset	Float []	{0,0,0,0,0,0}	Pallet2_Offset = {X, Y, Z, RX, RY, RZ} X, Y, Z, RX, RY, RZ = 左栈板 X, Y, Z, RX, RY, RZ 位置偏移量的修整值 (mm, mm, mm, 度, 度, 度)
22	Pallet1_Name	String	" Pallet_0"	参考使用 TMstudio 设定右侧栈板名称
23	Pallet2_Name	String	" Pallet_1"	参考使用 TMstudio 设定左侧栈板名称
24	Interval_To_Leave	Int	300	放置箱体后, 等待时间 (毫秒) 返回中继点.
25	Air_Pressure_Detection	Bool	FALSE	True: 侦测气压源压力 False: 无侦测气压源压力 (如夹爪未使用, 设定量为 False)
26	Movement_Protection	Bool	FALSE	TRUE: 侦测到栈板另一侧的箱体是否高于底座 False: 未侦测到栈板另一侧的箱体是否高于底座
27	Movement_Protection_Layer	Int	4	如 Movement_Protection = True, 当前栈板正堆栈到一定层数时, 侦测会被触发.
28	Gripper_Sensor_Feedback	Bool	Ture	True: 启动夹爪传感器回馈侦测 False: 未启动夹爪传感器回馈侦测

2.5.2 Advanced2-8

Advanced2-8 设置使用可编辑的副流程进行编程, 可提供其他灵活的配置, 供用户编辑夹爪使用, 传感器, 层之间的间隔, 偏移点, 位置, 放置向量或扫描 QR Code。

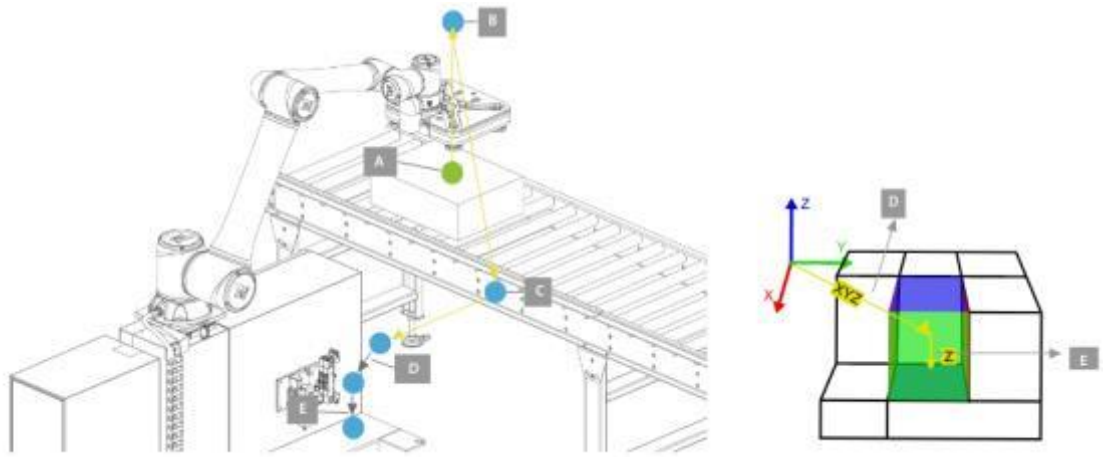
Advanced2-8 副流程定义:

名称	叙述
Advanced2_Grip	该副流程允许用户为夹爪定义夹取指令。
Advanced3_Release	该副流程允许用户为夹爪定义释放指令。
Advanced4_Waitfor_Input	该副流程允许用户输入等待输送带的传感器讯号, 以确保正确放置箱体。 预设: 由 DI14 触发 注意: 若使用者希望更改此设置, 则必须建立一个新的副流程来替换该副流程。
Advanced5_Interval_Between_Layers	此副流程允许用户在箱体的各层之间增加间隔, 例如插入隔板。
Advanced6_Via_Point	该副流程允许用户替换 Via Point 节点。
Advanced7_Box_Placement	该副流程允许用户替换接近向量节点。
Advanced8_Further_Vision	该副流程允许用户增加其他与视觉相关的应用程序, 例如条形码扫描。



警告:

若拿取的盒子需经过同层盒子, 请确定接近向量 Z 部份需高于箱体。



A	盒子取物点	C	避障点	E	放置点向下路径
B	取物点上方经过点	D	放置时的接近路径		

Relevant Parameters

参数	叙述
var_now_layer_inder	当前层数 (包括 Z 轴间隔)
var_box_info[2]	箱体高度 (mm)
var_pillar_now	当前立柱高度 (mm)
var_now_right_left	当前堆栈方向 1: 右、2: 左
var_via_point_way	中继点方向 1: 去、2: 返回

2.5.3 TMOperator_TMROBOT_TMPLTZOP_V002_Main 项目中其他副流程及同步线程

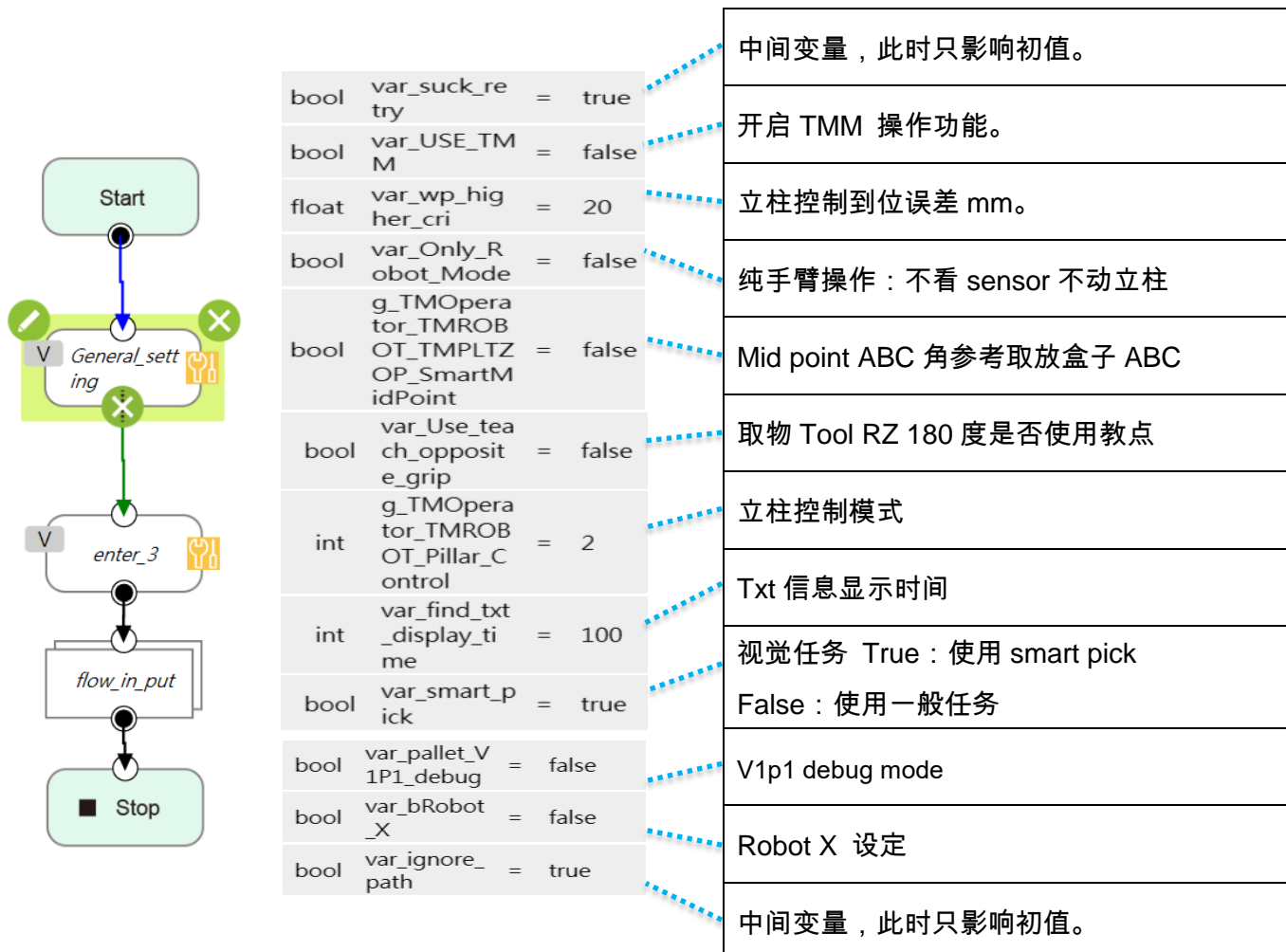
其他可编辑的副流程及同步执行续，可提供其他灵活的配置。

其他副流程及同步执行续定义：

种类	名称	叙述
subflow	take	take process
subflow	mid_put	put process
subflow	ini	initial process
subflow	Sucker_ON	gripper on
subflow	Sucker_OFF	gripper off
subflow	function	collect several repeat function
subflow	Via_Point	do Advanced6_Via_Point
subflow	flow_in_take	take process
subflow	flow_in_put	put process
subflow	Pallet_start	general setting and start
subflow	waitsensor	do Advanced4_Waitfor_Input
subflow	Take_Judge	grip status and retry action
thread	Thread_judge	several judgement in thread

种类	名称	叙述
thread	nonpause	several judgement in non pause thread
thread	varsync	variable sent to TMM

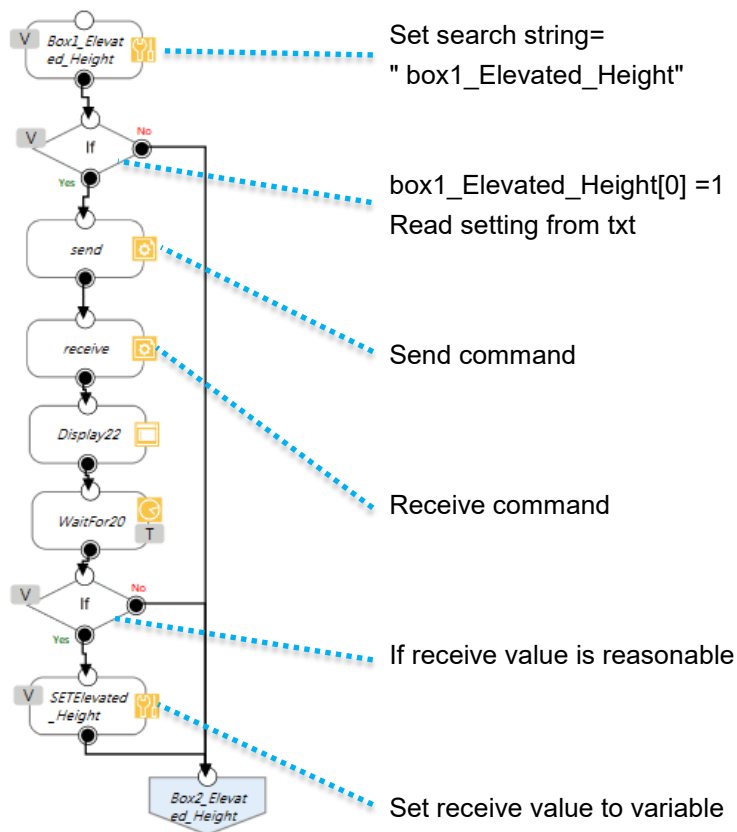
2.5.3.1 Subflow Pallet_start



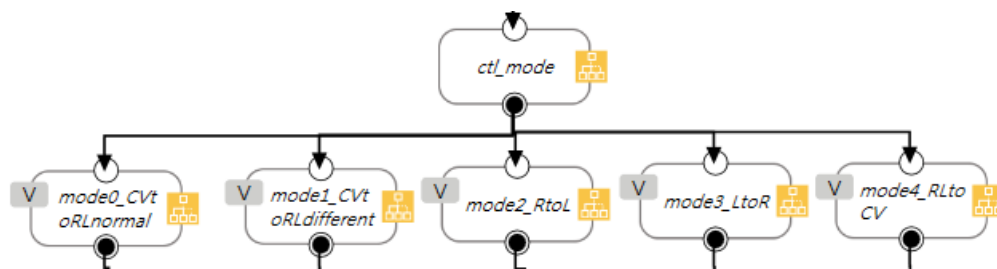
2.5.3.2 副流程 function

- Mode0: 初始执行一次，读出 txt 里除了 point/set 的信息。
- Mode1: 每放一盒子执行一次，读出该盒子的 point & set。
- Mode2: 更新立柱读值。
- Mode7: 初始执行一次，找出第 M 层第 N 个盒子，是算是第几个盒子。
- Mode8: 初始执行一次，检查不合理输入值。
- Mode9: 立柱读出该层要升缩到多少 mm。

- 读出 txt 讯息范例



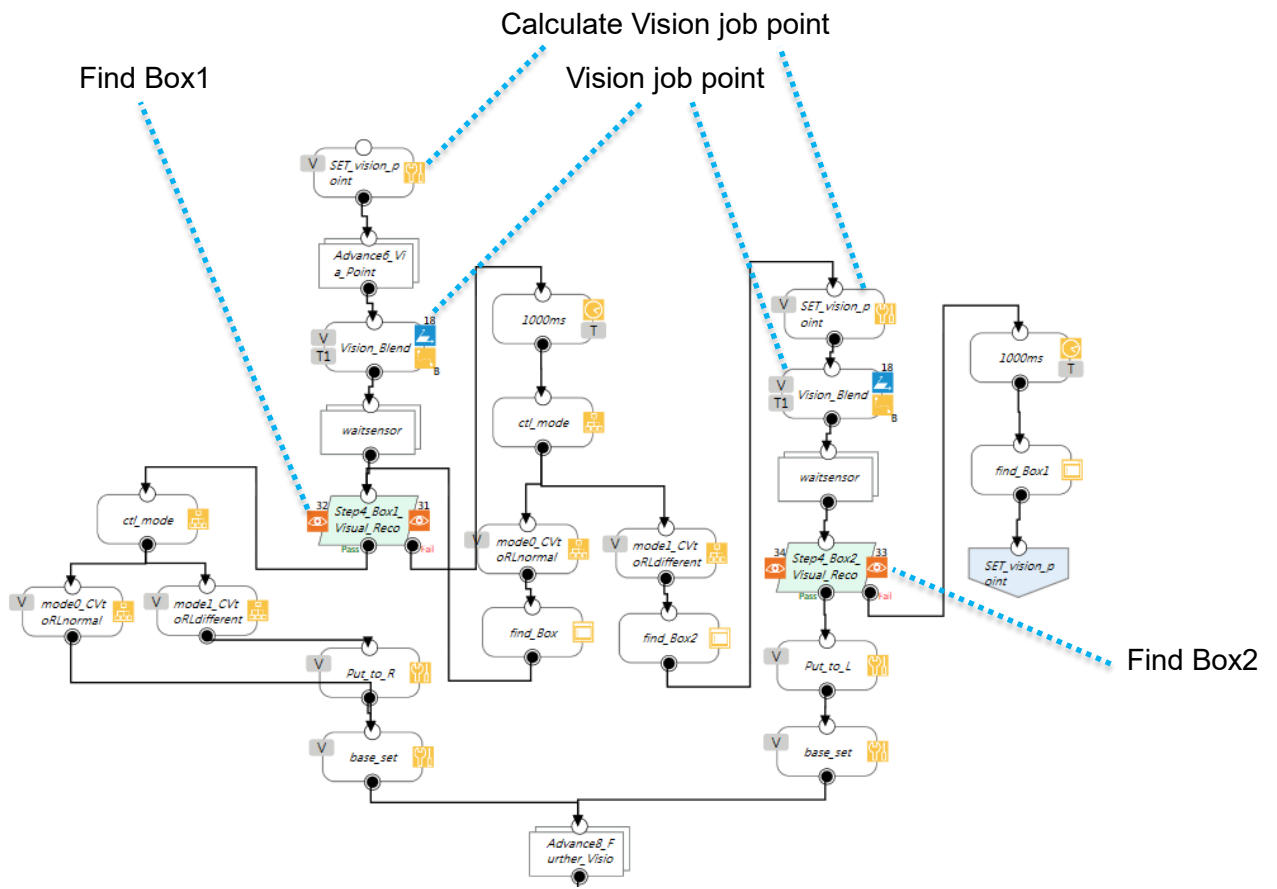
2.5.3.3 副流程 take

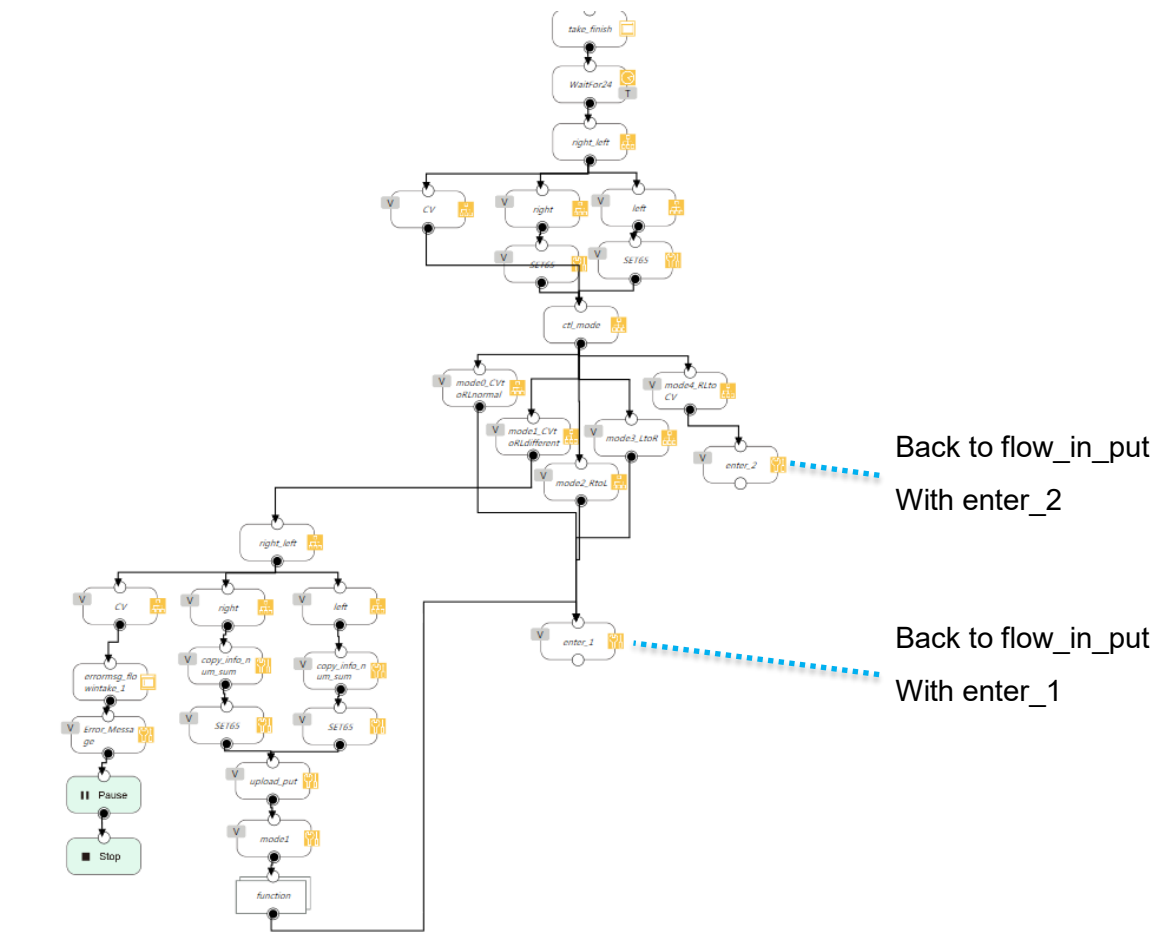


Gateway: var_pallet_ctl

- 0: [pallet] take from conveyor put to Right or Left with same box
- 1: [pallet] take from conveyor put to Right or Left with different box
- 2: [change pallet] take from Right and put to Left with same box
- 3: [change pallet] take from Left and put to Right with same box
- 4: [de-pallet] take from Right or Left with same box and put to conveyor

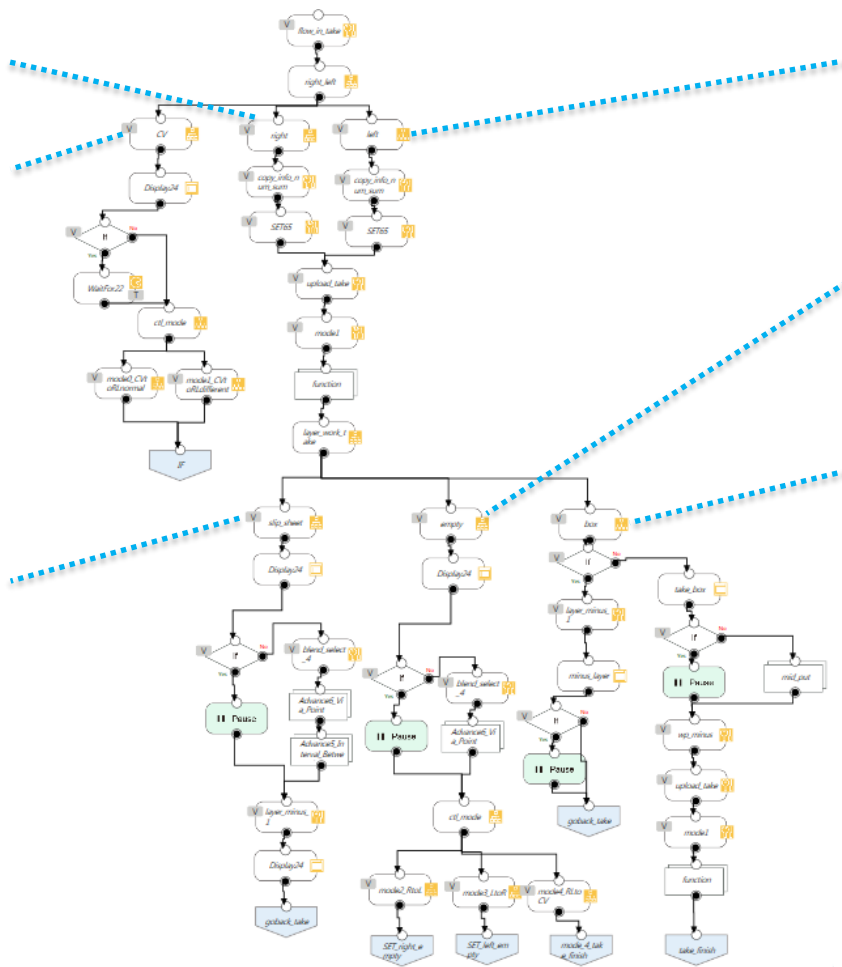
2.5.3.4 副流程 flow_in_take





Take from
right pallet
Take from
conveyor

Take from left
pallet

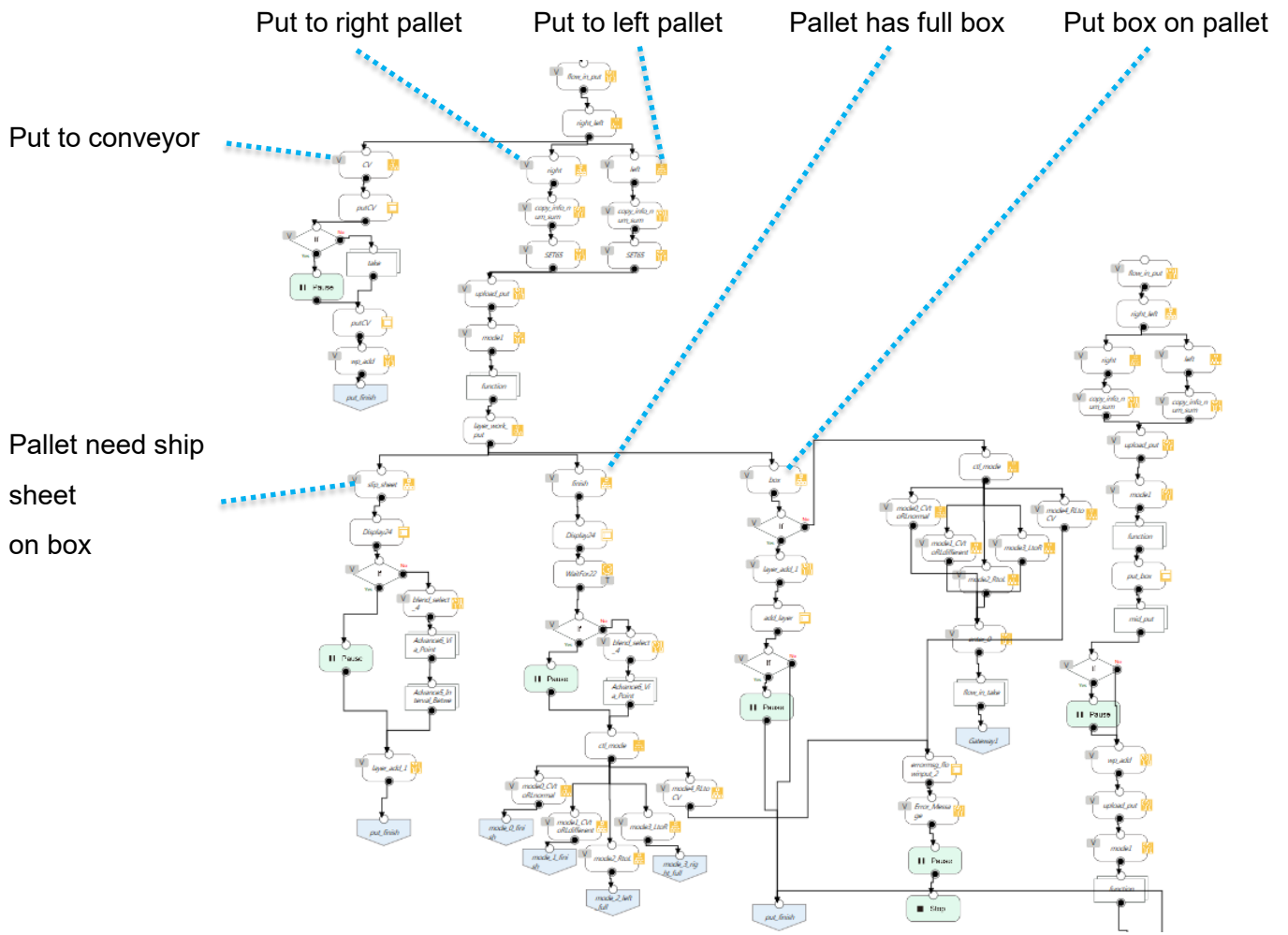


Pallet has no
box

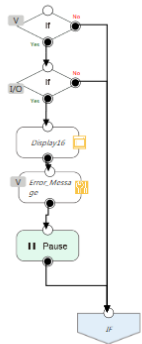
Pallet has ship
sheet on box

Pallet still has
box

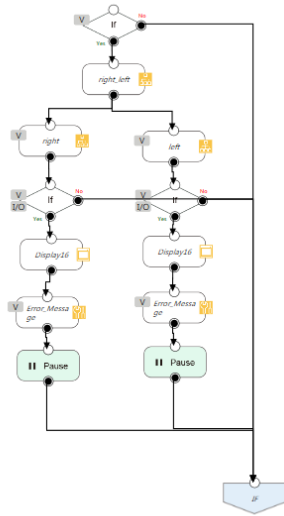
2.5.3.5 副流程 flow_in_put



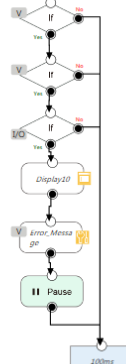
2.5.3.6 线程 thread_judge



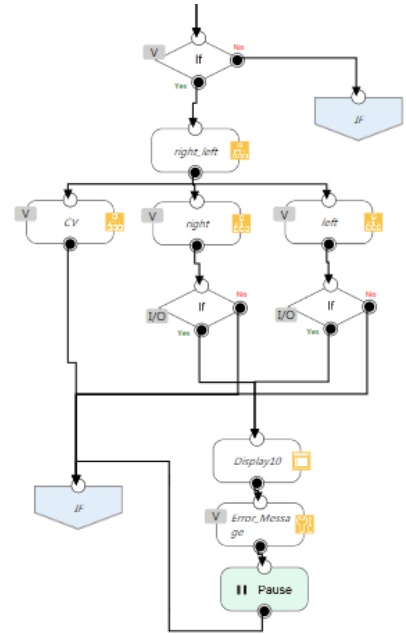
Detect DI_4 (L)
Air pressure error



Detect DI_2, DI_3
Layer protection

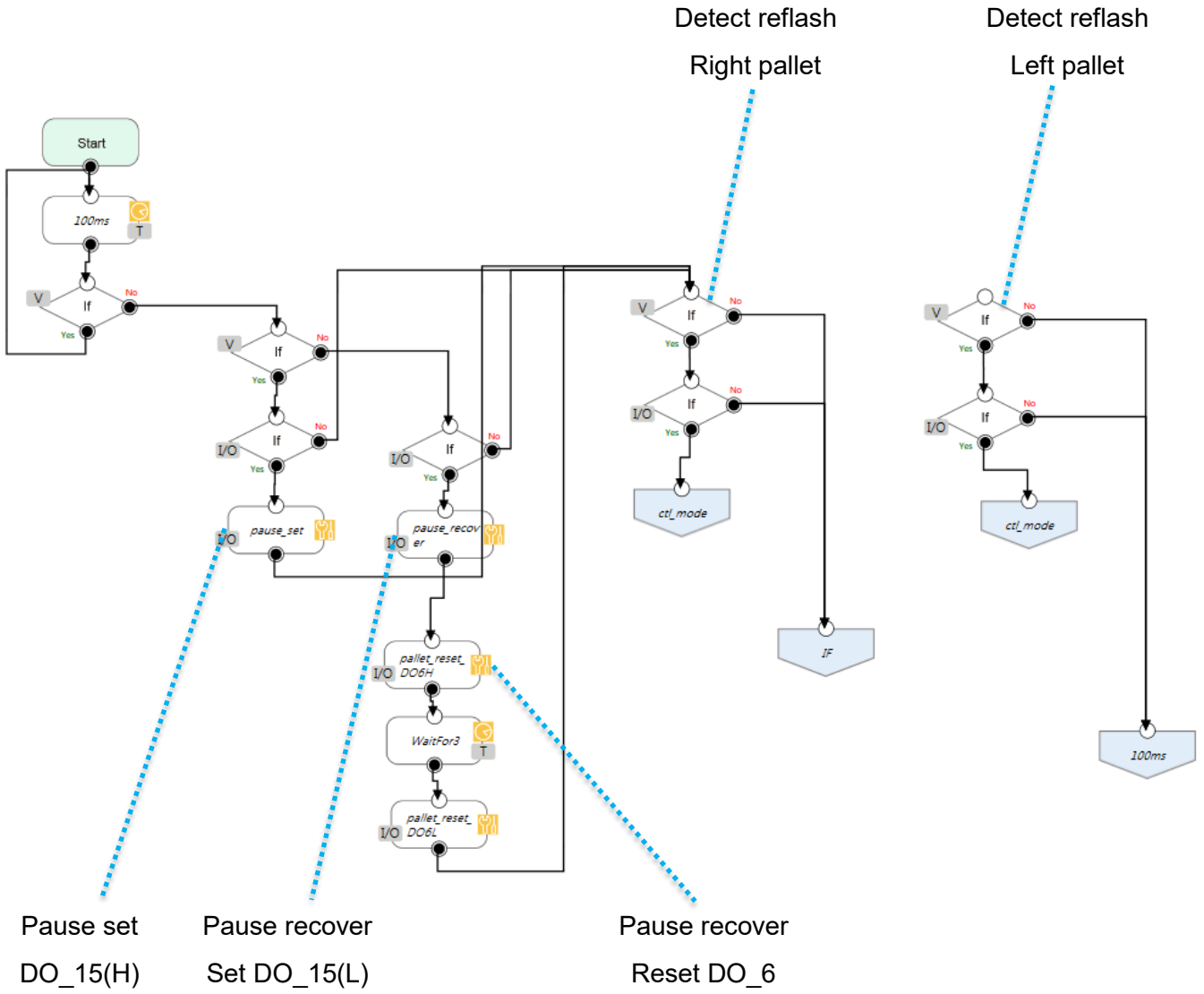


Detect end
module
DI_0, DI_1
Grip protection



Detect
DI_0, DI_1
Pallet detection

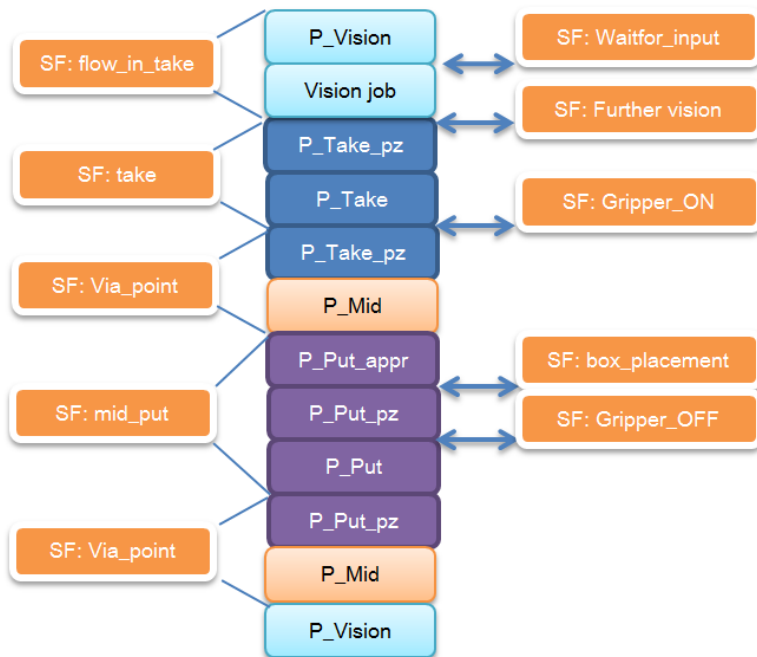
2.5.3.7 线程 nonpause



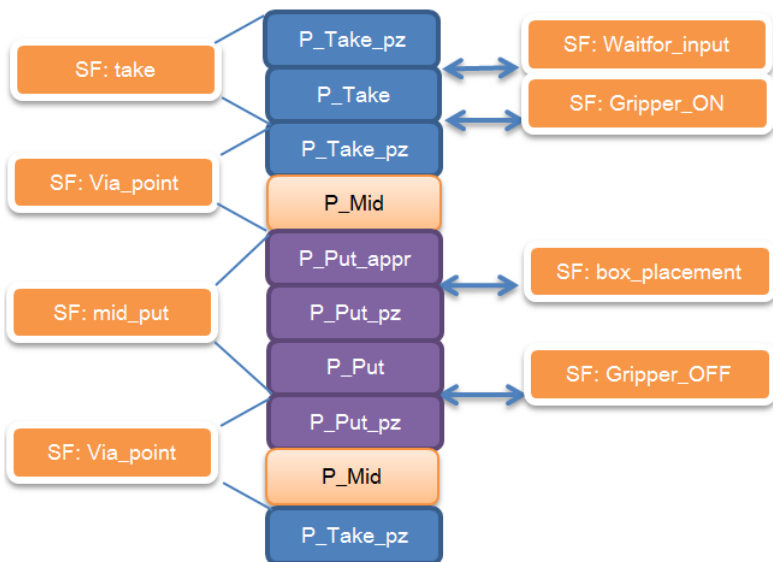
2.5.3.8 副流程(SF)之间关系

- Palletizing

- Palletizing 有视觉任务各副流程之间关系

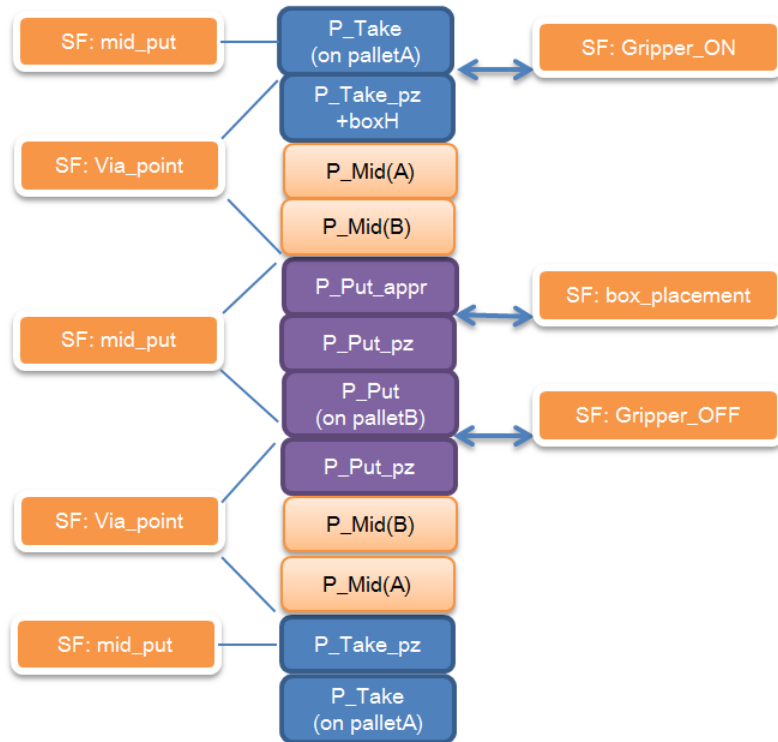


- Palletizing 无视觉任务各副流程之间关系



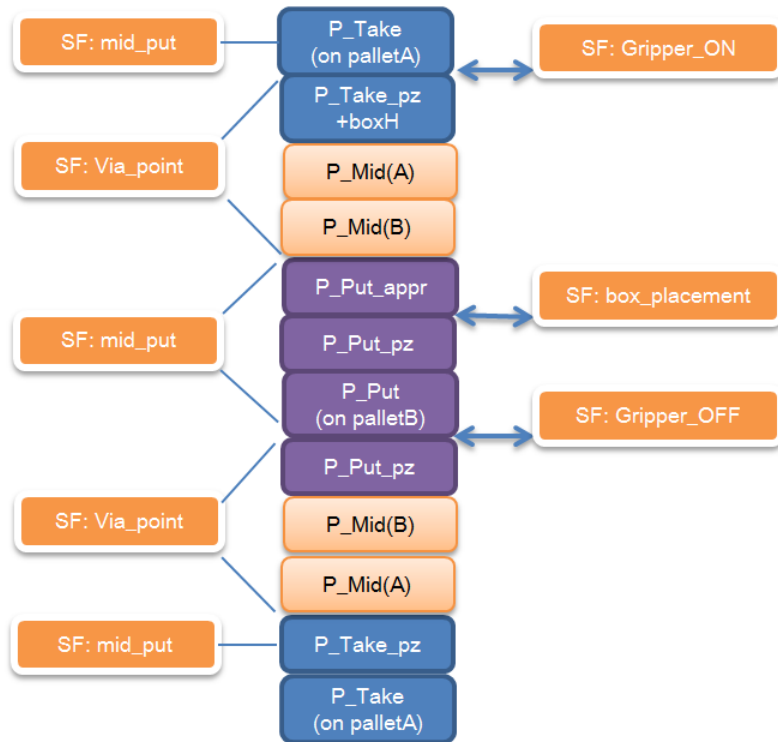
- Change Pallet

- Change Pallet 时各副流程之间关系



- De-Pallet

- De-Pallet 时各副流程之间关系



2.5.4 TMOperator_TMROBOT_TMPLTZOP_V002_Main 项目中其他参数、点位、坐标系

其他副流程及线程定义：

全局(global)变量名称前段皆以 g_TMOperator_TMROBOT_TMPLTZOP_命名，故下表中全局(global)变量名称仅显示 g_TMOperator_TMROBOT_TMPLTZOP_之后的名称。

#	类型	名称	叙述
1	global	Box_num	Box number that have been finish
2	global	right_left	Currently process right(1) or left(2)
3	global	start_R_index	process index of right pallet
4	global	start_L_index	process index of left pallet
5	global	Pillar_height	currently pillar elongation
6	global	Right_full	flag for right pallet is full
7	global	Left_full	flag for left pallet is full
8	global	SmartMidPoint	Middle point oriation ABC refer to box placement ABC
9	global	Pillar_control	pillar control selection
10	global	Mark_Golden	vision mark base for the teaching box
11	global	Mark_Golden2	vision mark base for the teaching box2
12	global	Mark_To_Box	vision mark base to 3DLM base for the teaching box
13	global	Mark_To_Box2	vision mark base to 3DLM base for the teaching box2
14	global	3DLMbase	3DLM base for the teaching box
15	global	3DLMbase2	3DLM base for the teaching box2
16	local	var_flow_in_take	var_flow_in_take = "TRUE", while taking box var_flow_in_take = "TRUE", while putting box
17	local	var_take_from_CV0_R1_L2	var_take_from_CV0_R1_L2 = 1 while box taking from conveyor var_take_from_CV0_R1_L2 = 2 while box taking from right pallet var_take_from_CV0_R1_L2 = 3 while box taking from left pallet
18	local	var_put_to_CV0_R1_L2	var_put_to_CV0_R1_L2 = 1 while box putting to conveyor var_put_to_CV0_R1_L2 = 2 while box putting to right pallet var_put_to_CV0_R1_L2 = 3 while box putting to left pallet
19	local	var_pillar_high	pillar max elongation Analog input value
20	local	var_pillar_low	pillar min elongation Analog input value
21	local	var_pillar_now	pillar now elongation value(mm)
22	local	var_pillar_length	pillar max elongation value(mm)
23	local	var_now_right_left	Currently process right(1) or left(2)
24	local	var_total_wp_max	max working box arrange
25	local	var_now_wp_index	index for box number refer to the pallet
26	local	var_now_layer_index	index for the layer number refer to the pallet

#	类型	名称	叙述
27	local	var_now_layer_wp_index	index for box number refer to the layer on the pallet
28	local	var_wp_higher_cri	max control error for pillar moving
29	local	var_pillar_pre	last pillar elongation value(mm)
30	local	var_box_info	box information
31	local	var_slipsheet_info	slip sheet information
32	local	var_now_set_wp	set information for box
33	local	var_now_point_wp	point information for box
34	local	var_suck_fail_retry	retry for gripping box fail
35	local	var_box_hold	recognize for box is hold
36	local	var_pallet_ctl	var_pallet_ctl = 1 for palleting AA box var_pallet_ctl = 2 for palleting AB box var_pallet_ctl = 3 for changing pallet box right pallet to left pallet var_pallet_ctl = 4 for changing pallet box left pallet to right pallet var_pallet_ctl = 5 for de-palleting AA/AB box
37	local	var_smart_pick	var_smart_pick = TRUE for using smart pick function in vision job var_smart_pick = FALSE for not using smart pick function in vision job
38	local	var_vision_find	var_vision_find = TRUE for using vision job to grap box var_vision_find = FALSE for using teaching point to grap box
39	point	suck_P	take point on conveyor
40	point	suck_P_Zup	z shift of suck_P
41	point	suck_P_Zupup	z shift of suck_P_Zup
42	point	mid_pos	middle point
43	point	subflow_blend	middle point
44	point	subflow_blend2	middle point
45	point	tar_p	target point of putting box
46	point	tar_pz	z shift of tar_p
47	point	tar_appr	with approach point of tar_pz
48	point	Pillar_move_pose	protect robot pose before pillar will activate
49	point	Vision_Blend	point before doing vision job
50	point	Opposite_Box1_Grip	another teach point of taking box1
51	point	Opposite_Box2_Grip	another teach point of taking box2
52	base	pallet_base_p	base of point tar_p
53	base	pallet_base_pz	base of point tar_pz
54	base	pallet_base_appr	base of point tar_appr
55	base	base_mid	base of point mid_pos
56	base	subflow_blend	base of point subflow_blend
57	base	subflow_blend2	base of point subflow_blend2
58	base	vision_Step2_Box1_Recognition_Box_Position	base of 3DLM on box1
59	base	vision_Step4_Box1_Visual_Recognition	base of vision mark on box1

#	类型	名称	叙述
60	base	vision_Step4_Box1_Visual_Recognition_SmartPick	base of SmartPick vision mark on box1
61	base	vision_Step2_Box2_Recognition_Box_Position	base of 3DLM on box2
62	base	vision_Step4_Box2_Visual_Recognition	base of vision mark on box2
63	base	vision_Step4_Box2_Visual_Recognition_SmartPick	base of SmartPick vision mark on box2

2.5.5 TMOperator_TMROBOT_TMPLTZOP_V002_Pillar 专案中参数

全局(global)变量名称前段皆以 g_TMOperator_TMROBOT_TMPLTZOP_命名，故下表中全局(global)变量名称仅显示 g_TMOperator_TMROBOT_TMPLTZOP_之后的名称。

#	类型	名称	叙述
1	global	pillar_AD_H	pillar max elongation Analog input value
2	global	pillar_AD_L	pillar min elongation Analog input value
3	global	pillar_pass	flag to recognize calibration pass or fail
4	local	var_pillar_AD	Analog input value
5	local	var_pillar_AD_pre	Last analog input value
6	local	var_pillar_move_cri	judge pillar is moving or not
7	local	var_timeout100ms	timeout value
8	local	var_Voltage_at_Pillar_Top	pillar max elongation Analog input value
9	local	var_Voltage_at_Pillar_Bottom	pillar min elongation Analog input value
10	local	var_AD_nochange	flag to recognize judge pillar is moving or not
11	local	var_automode	calibration mode auto mode or manual mode

2.5.6 TMOperator_TMROBOT_TMPLTZOP_V002_Grip 项目中变数

The prefix of the global variables is all named after g_TMOperator_TMROBOT_TMPLTZOP. Therefore, the table below lists the global variables with their names after the prefix.

#	类型	名称	叙述
1	global	Pallet_Grip_Status	judge grip status is ok or not
2	global	Detection_Drop_Flag	Detection_Drop_Flag = TRUE, user can detect box drop status in thread
3	local	var_count	retry count

2.5.7 TMOperator_TMROBOT_TMPLTZOP_V002_Release 项目中变数

全局(global)变量名称前段皆以 g_TMOperator_TMROBOT_TMPLTZOP_命名，故下表中全局(global)变量名称仅显示 g_TMOperator_TMROBOT_TMPLTZOP_之后的名称。

#	类型	名称	叙述
1	global	Pallet_Grip_Status	judge grip status is ok or not
2	global	Detection_Drop_Flag	Detection_Drop_Flag = TRUE, user can detect box drop status in thread
3	local	var_count	retry count

2.6 文件档案叙述

```

////////////////////////////////////
//version & control
////////////////////////////////////
pallet_version=1.1
//1.0;1.1;2.0
pallet_control=1
//pallet_control(start from 1)
//control = 1 ;Palletizing
//control = 2; Depalletizing
//control = 3; Change pallet
    
```

Num.	Name	Type	Default
1	Use_Vision_or_Not	Bool	TRUE
2	Control_Mode	Int []	{1,1}
3	Box_Sum	Int	0
4	Start_Direction	Int	1
5	Pallet1_Start_LayerIndex	Int []	{1,1}
6	Pallet2_Start_LayerIndex	Int []	{1,1}

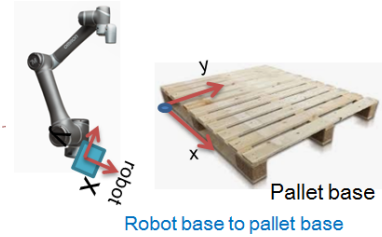
Advance setting in main page
 Control_Mode[0]=1 : read from txt pallet_control value
 Control_Mode[0]=0 : pallet_control = Control_Mode[1]

```

////////////////////////////////////
//Pallet_1
////////////////////////////////////
    
```

```

//palletx_base_xyzabc base relation : robot base to palletx base (without axis 7)
pallet1_base_xyzabc=-600,-1100,-680,0,0,0
    
```



```

//pallet setting lx,ly,h(mm),weight(kg);
pallet1_info=1216,1200,150,0
    
```

```

//layer number + layer type 0:box; 1: slipsheet; 2: finish;
pallet1_layer_info=3,0,1,0,2
    
```

Total 3 layers, layer 1:0(box), layer 2:0(slip sheet), layer 3:0(box),

```

//total object number , layer1 object number ,layer2 object number ....(slipsheet layer : object
number =0) max 20 layer
    
```

```

pallet1_layer_wp_num=8,4,0,4
pallet1_layer_wp_sum=8,4,4,8
    
```

Total 8 box, layer 1:4(4 box), layer 2:0(0 box), layer 3:4(4 box),

Total 8 box, layer 1:4(4 box), layer 1~2:4(4 box), layer 1~3:4(8 box),

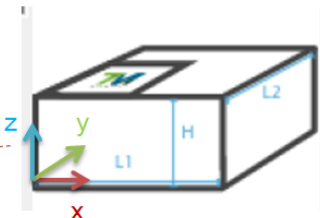
```

//palletx_Pillar_Height=pillar move offset(mm)
pallet1_Pillar_Height=100,200,300
    
```

Pillar height : layer 1:100(mm), layer 2:200(mm), layer 3:300(mm)

```

////////////////////////////////////
//box_1
////////////////////////////////////
//boxx_info: box setting lx,ly,H(mm),weight(kg),label;
box1_info=22,10,12.5,5,0
    
```



```

//boxx_Elevated_Height :box taken point offset(mm)
box1_Elevated_Height=300
    
```

L1 indicate x direction;
 L2 indicate y direction;

Box elevated height(mm)

```

////////////////////////////////////
//box_2
////////////////////////////////////
//boxx_info: box setting lx,ly,H(mm),weight(g),label;
box2_info=22,10,12.5,100,0
    
```

```

//boxx_Elevated_Height :box taken point offset(mm)
box2_Elevated_Height=300
    
```

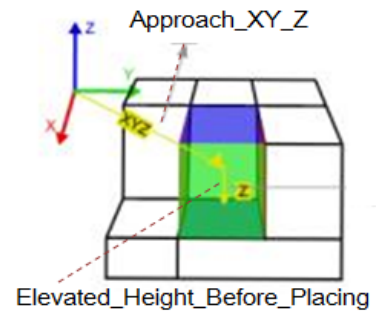
```
//palletx_Via_Point_Offset(mm) robot base
pallet1_Via_Point_Offset=200,200,100 -----
```

Via point define x,y,z shift value from box Elevated_Height point
Refer to robot base

```
//palletx_Approach_XY_Z(mm) pallet base
pallet1_Approach_XY_Z=30,150
```

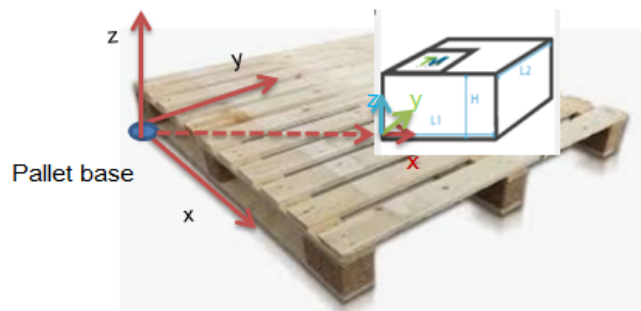
```
//Pillarx_Elevated_Height_Before_Placing pallet base
pallet1_Elevated_Height_Before_Placing=20
```

```
//slipsheet setting lx,ly,H(mm),weight(g);
pallet1_slipsheet_info=1200,1200,10,100
```



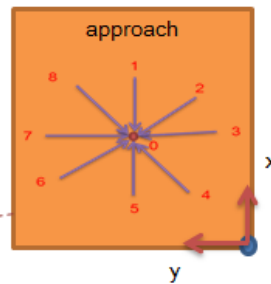
```
// point xyzabc (float)
pallet1_point_wp_1=-825,-340,140,0,0,0
pallet1_point_wp_2=-615,-340,140,0,0,0
pallet1_point_wp_3=-835,-540,140,0,0,0
```

Box position base on pallet base

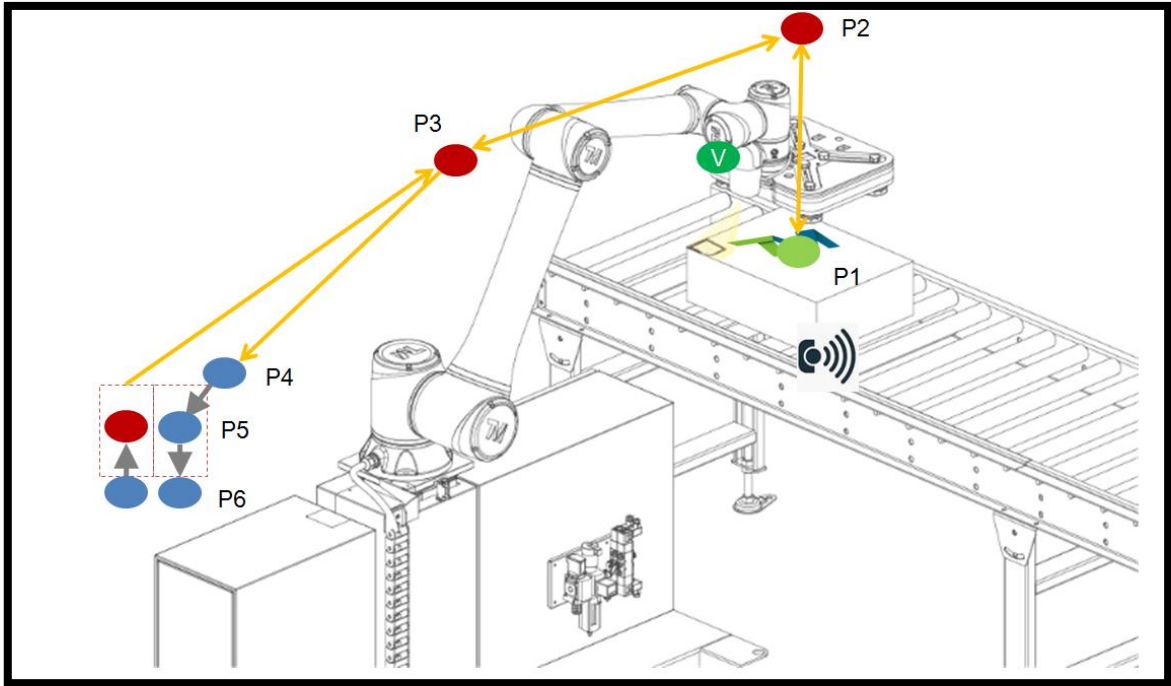


```
//palletx_set_wp_y
pallet1_set_wp_1=1,1,1,1,0,0,0,1
pallet1_set_wp_2=1,1,2,2,3,0,0,1
pallet1_set_wp_3=1,1,3,3,5,0,0,1
```

- [1]: pallet x number(reserved)
- [2]: layer number(start from 1)
- [3]: index(for layer)(start from 1)
- [4]: index(for pallet)(start from 1)
- [5]: approach(range 0~8) pallet base
- [6]: put way (0: normal)
- [7]: take way (0: normal; 1: tool RZ 180 degree take)
- [8]: (reserved)



2.7 Base and Point 计算



	base	point
P1	Infeed	suck_P
P2	Infeed	suck_P_Zupup
P3	base_mid	mid_pos
P4	pallet_base_appr	tar_appr
P5	pallet_base_pz	tar_pz
P6	pallet_base_p	tar_p

P1, P2 calculation:

Without vision mark find:

Base:

- B1(from 3DLM)
- B2(from vision mark)

With box height
Calculate to

- B3(box corner robot base)
- B4(box corner base on B2)

Point:

- P teach(base on B1)

With box height
Calculate to

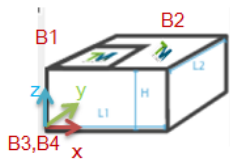
- P take(base on B3)

1.Base Infeed = B3

2.Point suck_P = P take;

3.Point suck_P_Zupup = suck_P + Elevated_Height

With vision mark find:



Base:

1. B1 (from 3DLM)
2. B2 (from vision mark)
With box height
Calculate to
3. B3 (box corner robot base)
4. B4 (box corner base on B2)

Point:

1. P teach (base on B2)
With box height, B4
Calculate to
2. P take (base on B3)

1. Running vision mark base B2r
2. $B3_{new} = B2r * B4$ (box corner base on B2)
3. Base Infeed = B3new
4. Point suck_P = P take;
5. Point suck_P_Zupup = suck_P + Elevated_Height

P3 calculation:

Point mid_pos = suck_P

Base base_mid = Infeed + Via_Point_Offset

P4, P5, P6 calculation:

Point:

$tar_p = tar_{pz} = tar_{appr} = P \text{ take (base on B3)}$

Base:

$pallet_base_p(B6) = Robot \text{ base (B4)} * Pallet \text{ base (B5 base on B4)} * Box \text{ base (B6 base on B5)}$

$pallet_base_pz = pallet_base_p + Elevated_Height_Before_Placing$

$pallet_base_appr = pallet_base_pz + Approach_XY_Z$

Base name mapping to flow variable

1. B1(3DLM):
vision_Step2_Box1_Recognition_Box_Position
vision_Step2_Box2_Recognition_Box_Position
2. B2(vision):
vision_Step4_Box1_Visual_Recognition

vision_Step4_Box2_Visual_Recognition

3. B3(box corner)

var_base_3DLM_A

var_base_3DLM_B

4. B4(box corner base on B2)

g_TMOperator_TMROBOT_TMPLTZOP_Mark_To_Box

g_TMOperator_TMROBOT_TMPLTZOP_Mark_To_Box2

Point name mapping to flow variable

1. P teach(base on B1)

Step3_Box1_Teach_Grippint_Point

Step3_Box2_Teach_Grippint_Point

2. P take(base on B3)

var_point_take_A

var_point_take_B

TECHMAN
ROBOT



www.tm-robot.com