

OMRON TM Operator Platform User Manual



Original Instruction

I662-E-01

This Manual contains information of the Techman Robot product series (hereinafter referred to as the TM Robot). The information contained herein is the property of Techman Robot Inc. (hereinafter referred to as the Corporation). No part of this publication may be reproduced or copied in any way, shape or form without prior authorization from the Corporation. No information contained herein shall be considered an offer or commitment. It may be subject to change without notice. This Manual will be reviewed periodically. The Corporation will not be liable for any error or omission.

W logo is registered trademark of TECHMAN ROBOT INC. in Taiwan and other countries and the company reserves the ownership of this manual and its copy and its copyrights.

Terms and Conditions Agreement

Warranty Limitations of Liability

Warranties

Exclusive Warranty

Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

Limitations

OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE. Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.

Buyer Remedy

Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron's analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

See http://www.omron.com/global/ or contact your Omron representative for published information.

Limitation of Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT,

WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Application Considerations

Suitability of Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.

Disclaimers

Performance Data

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron's test conditions and the user must correlate it to actual application requirements. Actual performance is subject to the Omron's Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Contents

Revision History	11
1. General	12
1.1 Overview	12
1.2 How to Get Help?	12
1.3 Warning and Caution Symbols	13
1.4 Safety Precautions	13
1.5 Validation and Liability	14
1.6 Limitation of Liability	14
1.7 Function Note Symbols	14
2. TM Operator Platform Concept	16
2.1 What is Robot Operator?	16
2.2 What is TM Operator Platform?	17
2.3 Systematic Structure	17
2.4 Building a Robot Operator by TM Operator Platform	19
2.4.1 Application Targeting and Conceptual Stage	19
2.4.2 Analysis and Verification of the Concepts	21
2.4.3 End-effectors	24
2.4.4 Software Conceptual Design Stage	27
2.4.5 Software Development and Trial Stage	33
2.4.6 Make it into a product	34
3. TM Operator Suite	35
3.1 License	35
3.2 Installation	
3.2.1 Builder Installation	
3.2.2 TMflow Installation	
3.2.3 TMstudio Installation	40
3.2.4 Program Entry	44
3.3 OperatorBuilder package – general structure of SW	44
3.3.1 Relationship of Builder, Dashboard Console, and Service Engine	46
3.3.2 Programming	46
3.3.3 Runtime	47
3.4 Environment	48
3.4.1 Introduction of Environment Settings	48
3.4.2 Software installation and version check	48

3.4.3 Network Connection Confirmation	
3.4.4 Dashboard Console Firewall Whitelist Setting	
3.4.4.1 SQL Server through the Firewall	49
3.4.4.2 Allow Service Engine apps through the firewall	51
3.4.4.3 Allow Dashboard Console apps through the firewall	51
3.4.5 Confirm that the firewall settings have been completed	51
3.4.6 IP Settings	51
3.4.6.1 Robot IP Setting	
3.4.6.2 Computer IP Setting	53
3.4.6.3 Set the Service Engine IP pointed by TMflow	54
3.4.6.4 Set Service Engine IP	55
3.4.6.5 Confirm that the items are ready	56
3.5 Builder	57
3.5.1 Builder's role	57
3.5.2 Service engine	59
3.5.2.1 Search and Screening Functions	59
3.5.2.2 Service Status List	60
3.5.3 Basic concept	61
3.5.3.1 Device	62
3.5.3.2 Tags	77
3.5.3.3 Action	96
3.5.3.4 Monitor	
3.5.3.5 Group	
3.5.4 Dashboard Designer	
3.5.4.1 Operation Interface Introduction	
3.5.4.2 Toolbox Introduction	
3.5.4.3 Design Window Introduction	
3.5.4.4 Property Window Introduction	
3.5.4.5 Toolbox Components and Properties	
3.5.4.6 Component	
3.5.4.7 Data Control	
3.5.5 Script Language	
3.5.5.1 Program structure	
3.5.5.2 Program Rules	
3.5.5.3 Operation Symbols	
3.5.5.4 Data Type	230

3.5.5.5 Properties	238
3.5.5.6 Command Introduction	239
3.5.5.7 System reserved characters	240
3.6 Dashboard Console	
3.6.1 Dashboard Advance Confirmation	240
3.6.2 Dashboard Console Function Explanation	241
3.6.2.1 Dashboard Deployment Teaching	244
3.6.2.2 Dashboard Deployment Warning	246
3.6.2.3 Dashboard Container Connection Interrupted	247
3.7 TMflow	
3.7.1 TMflow's role	
3.8 TMstudio	
3.8.1 TMstudio's role	249
3.8.2 TMstudio Palletizing Wizard and Scene creator, function and limitation	249
3.8.2.1 Step 1. Import CAD File	252
3.8.2.2 Step 2. Import Pillar Base	252
3.8.2.3 Step 3. Set Pillar Base Parameters	253
3.8.2.4 Step 4. Import Robot	254
3.8.2.5 Step 5. Import Tools	256
3.8.2.6 Step 6. Insert and align the pallet	256
3.8.2.7 Step 7. Change the extended direction of pallet	258
3.8.2.8 Step 8. Set parent-child relationship	
3.8.2.9 Step 9. Start palletizing	
3.8.3 Build your palletizing operator layout in TMstudio	261
3.8.3.1 Change Gripper	261
3.8.3.2 Choose Palletizing Type	
3.8.3.3 Start Palletizing Setting	
3.8.3.4 Step 1 - Set the Pallet size and boundary	
3.8.3.5 Step 2 - Set the box size and label direction	
3.8.3.6 Step 3 - Create layers and placement method	
3.8.3.7 Step 4 - Create the Layer Placement Sequence	270
3.8.3.8 Copy Box	271
3.8.3.9 Simulation Page - Parameter Setting	272
3.8.3.10 Verification	275
3.8.3.11 Simulation	
3.8.3.12 Export	276

3.8.3.13 Point File Format Description	277
3.8.4 Set the project as Palletizing Wizard default project	279
4. Common Techniques	
4.1 CTQ1 : Create a Operator UI	
4.2 CTQ2 : Login, Logout and Access Control of Robot	284
4.3 CTQ3 : Embedded Tutorial Video	
4.4 CTQ4 : Call External Executive File	287
4.5 CTQ5 : Call TMflow Automatically When Windows Begins	287
4.6 CTQ6 : Call TMflow and Send Index Accordingly	
4.7 CTQ7 : Read/Write Variable, Base and Point	289
4.8 CTQ8 : Open Designated Subflow	292
4.9 CTQ9 : Control I/O on robot	293
4.10 CTQ10 : Open, Copy and Save the TMflow Project	293
4.11 CTQ11 : Open the Specified View of TMflow	294
4.12 CTQ12 : Import TCP and Text File	294
4.13 CTQ13 : File Storage	305
4.14 CTQ14 : Virtual Keyboard	313
4.15 CTQ15 : Error Handling	314
4.16 CTQ16 : About Dashboard Console	316
4.16.1 Deploy UI to Dashboard Console	317
4.16.2 Lock the Toolbar on the Dashboard Console	317
4.16.3 Lock the Hot Key on Dashboard Console	318
5. Advanced techniques	
5.1 General description of the example	
5.2 ATQ 1: Protection of TMflow Project (Component)	
5.3 ATQ 2: How to establish Accessibility management system on builder	326
5.4 ATQ 3: Call vision job	
5.5 ATQ 4: Choose executing gripper on dashboard	
5.5.1 Choose Gripper by parameterize of subflow in TMflow	
5.5.2 Programming custom gripper logic	
5.5.3 Change TCP parameters	
5.6 ATQ 5: Move the robot and record the point	
5.6.1 How to establish robot controller on dashboard	
5.6.2 How to overwrite the exist point	
5.7 ATQ 6: Execute grip/release function on dashboard	
5.7.1 Create action to play robot	

37
38
38
40
40
40
41
42
44
48
00

Revision History

Revision	Date	Revised Content	
01	January 2021	Original release	

1. General

1.1 Overview

This manual is provided for the party who would like to build Robot Operator product based on TM Robot product and sell. TM Operator Platform is a development platform provided by TECHMAN ROBOT to build a Robot Operator-a highly flexible robot cell product. The major tool of the TM Operator Platform is the software suite – TM Operator Suite.

In this manual, at first, the concept of Robot Operator is introduced. Then, the guidance of how to start to build a Robot Operator- the conceptual phase of constructing a Robot Operator using TM Operator Platform is provided. In the later chapters, first the installation and usage of the major software tool TM Operator Suite is introduced. Then, chapter about the common techniques and other techniques are provided with examples.

Readers of this manual should first understand the functionality of TM Robot product. There are the related manuals for this purpose:

Manual Title	Description	
Safety Manual	Contains the safety information of TM Robot.	
TMflow	Instructions for using TMflow software.	
TMvision	The built-in functions of TM Robot with rich auxiliary and integration tools for TM Robot's including feature recognition, object positioning, image enhancement mode, barcode	
	steps.	

There are the advanced manuals that may relate to the Robot Operator design based on different applications, or as a design reference:

Manual Title	Description
Expression Editor & Listen Node	TMflow's Expression Editor
Script Language	Instruction for use of Builder Script Language

1.2 How to Get Help?

Users can access information sources on the corporate website:

http://www.ia.omron.com/

Users and system integrators of TM Robot must read and fully understand this chapter before using this robot. In addition, before users perform any operation on the robot in accordance with this manual, it is necessary to read and comply with the *Safety Manual* for the corresponding product's hardware

and software version, and the *Hardware Installation Manual* for the corresponding hardware version, before the operation can be performed.

1.3 Warning and Caution Symbols

The table below shows the definitions of the warning and caution levels used in our manuals. Pay close attention to them when reading each paragraph, and observe them to avoid personal injuries or equipment damage.



DANGER:

Identifies an imminently hazardous situation which, if not avoided, is likely to result in serious injury, and might result in death or severe property damage.



WARNING:

Identifies a potentially hazardous situation which, if not avoided, will result in minor or moderate injury, and might result in serious injury, death, or significant property damage.



CAUTION:

Identifies a potentially hazardous situation which, if not avoided, might result in minor injury, moderate injury, or property damage.

Danger, Warning, and Caution Symbols

1.4 Safety Precautions



DANGER:

This product can cause serious injury or death, or damage to itself and other equipment, if the following safety precautions are not observed:

 All personnel who install, operate, teach, program, or maintain the system must read the *Hardware installation Manual*, *Software Manual*, and *Safety Manual* according to the software and hardware version of this product, and complete a training course for their responsibilities in regard to the robot.



Read Manual Label; Impact Warning Label

- All personnel who design the robot system must read the *Hardware installation Manual*, *Software Manual*, and *Safety Manual* according to the software and hardware version of this product, and must comply with all local and national safety regulations for the location in which the robot is installed.
- The TM Robot must be used for its intended use.

- Results of the risk assessment may require the use of additional risk reduction measures.
- Power to the robot and its power supply must be locked out and tagged out or have means to control hazardous energy or implement energy isolation before any maintenance is performed.
- Dispose of the product in accordance with the relevant rules and regulations of the country or area where the product is used.

1.5 Validation and Liability

The information contained herein neither includes how to design, install, and operate a complete robotic arm system, nor involves the peripherals which may affect the safety of the complete system. The integrators of the robot should understand the safety laws and regulations in their countries and prevent hazards from occurring in the complete system.

This includes but is not limited to:

- Risk assessment of the whole system
- Adding other machines and additional risk reduction measures based on the results of the risk assessment
- Using appropriate software safety features
- Ensuring the user will not modify any safety measures
- Ensuring all systems are correctly designed and installed
- Clearly labeling user instructions
- Clearly marked symbols for installation of the robot arm and the integrator contact details
- Making accessible relevant documents, including the risk assessment and this Manual

CAUTION:

This product is a partly complete machine. The design and installation of the complete system must comply with the safety standards and regulations in the country of use. The user and integrators of the robot should understand the safety laws and regulations in their countries and prevent major hazards from occurring in the complete system.

1.6 Limitation of Liability

No safety-related information shall be considered a guarantee by the Corporation that a TM Robot will not cause personnel injury or property damage.

1.7 Function Note Symbols

The following table defines the functional note symbols used in this manual. Read the paragraphs carefully.



IMPORTANT:

This symbol indicates the relevant functional details to assist programming and use.



NOTE:

This symbol indicates the relevant functional use tips to assist programming efficiency.

Function Note Symbols

2. TM Operator Platform Concept

In this chapter, the basic concept of the Robot Operator, TM Operator Platform, and how to build a Robot Operator by TM Operator Platform, will be introduced.

2.1 What is Robot Operator?

Robot Operator is a flexible, application oriented robot cell/kit, integrates with many intelligent automation features, make it easy to use so the end-user can easily handle it, and doing the workpiece changeover by themselves, while this part is traditionally need to be done by system integrators.

The traditional robot cell, is also an application oriented solution that provided by system integrators, but always is customized for the automation requirements of each stations in the end-user site. It takes a lot of time and cost to build them according to the needs of different customers. The traditional robot cell also has an operational user interface for the product line operator of end-user to operator the cell, but once the application details, especially the workpiece, is changed, there is usually no such UI and functions for end user to do the relative change on it. End user needs to go back to the system integrator. The adjustment of logics of automation, the changes of the sequences of the application, and especially the handling of different workpieces, usually rely on the deep layer programming between the robots and other devices in the cell. The end user has no ability to handle it.

Robot Operator is more like a human operator that have built-in application techniques in their controller, and having a step-by-step application setting UI designed for end users to set the application in the first time, and do the reset by themselves when changeover needed. It becomes possible only when the provider of the Robot Operator having a brilliant idea about how to let end-user easy to do the changeover by themselves, which may need a lot of intelligent features from the robot itself, or integration with sensors. The provider of the Robot Operator need to wrap up all these intelligent features, and design an easy-to-use user interface, and provides it accompanied with the Robot Operator to the end user, instead of a customized deep layer programming like robot language, which the end user can't handle.

The provider of the traditional robot cell is always limited in system integrators, because only they can provide the related services and the re-construction of the cell when changeover happens. The provider of the Robot Operator, can be the system integrator that would like to provide a standard robot cell for specific application, with the ability to let the end-user doing customization and setting by themselves, and also other designers that original not an system integrator, but would like to provide this kind of revolutionary product to the industry.

2.2 What is TM Operator Platform?

TM Operator Platform is a development platform to build a Robot Operator product , It is constructed with 3 major parts:

- 1. TM Operator Suite: a software suite that can integrate the intelligent software of TM Robot, and build a step by step UI for the Robot Operator. The result of the step by step UI is implemented on an IPC separately from the robot controller.
- 2. TM Robot: a series of robot that have many built-in smart features, like all-graphical user interface, flow-chart-like programming UI, built-in eye-in-hand vision, and intelligent options, like external cameras, OCR, TM 3Dvision, and TM AI+.
- 3. TM Plug & Play eco-system: an eco-system that many automation peripheral device makers provide ready-to-use solutions for TM Robot, having software packages, ready-to-install hardware and wirings, like grippers, end-effectors, communication modules, and mechanical accessories.

Based on these major parts of TM Operator Platform, the one who would like to build a Robot Operator, can use the built-in intelligent features of TM Robot, and using TM Operator Suite to integrate all intelligent features of TM Robot into an application oriented, step by step user interface, and leverage many ready-to-use automation peripherals to quickly build a revolutionary easy-to-use Robot Operator and create the business of it.

2.3 Systematic Structure

The diagram below shows the classical Robot Operator built by TM Operator Platform.



Systematic structure of a Robot Operator created by TM Operator Platform

TM Robot and TM Robot's control box are the central part of the Robot Operator, while these parts transform the TM Robot to an Robot Operator:

 Operator UI: the step by step UI created by TM Operator Suite is running on an individual IPC, connected with TM Robot's control box via TCP/IP. A display module is usually arranged for the end-user, like a touch screen, or a TM Plug & Play Teach Pendant. The minimum configuration of the IPC is listed in the table below:

Minimum configuration	Maker's development environment (e.g. the Laptop used to develop)	The IPC in Robot Operator	
Minimum Running Target	TM Operator Suite*	TMflow Client Operator UI Runtime**	
Operating System	Windows 10 Service Pack 1		
CPU	Intel i3 4th generation		
Memory(RAM)	8GB	4GB	
Hard Disk	40GB of free hard disk space		
Interface	USB port, Ethernet port		
Display Resolution	1,366X768		
Additional Requirements	Additional Requirements1. 2010Redistributable_vcredist (x64, x86) 2. 2013Redistributable_vcredist (x64, x86) 3. 2015Redistributable_vcredist (x64, x86) 4. Microsoft. Net 4.52 or above		
*Include Builder, TMflow client, TMstudio			

**The dashboard console, and the dashboard content created by the maker.

Minimum configuration of the computer to run TM Operator Suite and the Runtime

- End-effector: Grippers or tools for the application, mounted on TM Robot's flange.
- External equipment: equipment may added to achieve the application scenario, like extension IO module to have more IO than the built-in 16in /16 out on robot control box, PLCs to leverage an existed subsystem, Safety controller for expanding safety design in a more complex case, External axes to expand the motion range of the robot...etc. The external equipment can connect to TM Robot's control box using the I/O port or communication ports provided by TM Robot.
- Sensors: Limit sensor of the conveyor, external cameras, 3D cameras....etc. They may connect to TM Robot's control box, or the subsystem in the external equipment.

2.4 Building a Robot Operator by TM Operator Platform

In this section, guidance for those who want to build a Robot Operator product (Makers) of how to start to construct a Robot Operator's product concept, and how to build it with TM Operator Platform, and finally make it into a product, will be introduced.

2.4.1 Application Targeting and Conceptual Stage

A successful Robot Operator for achieving the flexibility for end-user to change over the workpiece by themselves and the design for easy-to-implement by end-user, and make sure it can become a successful product, depends on the conceptual stage consideration. In this stage, makers should consider the most important topics: what is the targeted application?

These topics should be considered in the first step:

- For this application, is there a method to let end-user changeover the workpiece by themselves?
- How many parts inside the application that I can provide to the user?
- How customization will it be, when the workpiece is changed?
- In what range of the application can my Robot Operator product cover?

To check these questions, it is suggested to draw a conceptual diagram of the whole system first to assist the Maker to analysis, like below:



Conceptual layout diagram - Deburring robot operator example

No.	Content
1	TM5-900
1.1	Force sensor
1.2	Pneumatic Spindle
1.3	Deburring Tool
1.4	Robot Stick
1.5	Robot Control box
2	IPC for Operator UI
2.1	Touch Screen
2.2	Keyboard & Mouse
3	System Base
3.1	Protective Cover
4	Conveyer
4.1	Workpiece Input
4.2	Workpiece Output
4.3	Proximity Sensor
4.4	Fixture Module
5	Workpiece Tray
5.1	TM Landmark and QR code

Table xx: Conceptual layout diagram – Deburring robot operator example

This conceptual layout diagram of the Robot Operator should include all major parts to achieve the application, including at least

- Robot
- External equipment
- End-effector
- Workpiece input and output parts.
- User interface
- Sensors

Then, a Conceptual functionality diagram is suggested to be constructed based on the layout concept, to present your concept about how it works.

Concept of Automation Process



Conceptual automation process diagram – Deburring robot operator example

The process in the diagram above is:

- 1. Workpiece is fixed on the tray, loaded on the conveyor, send from the station before this station.
- 2. Workpiece detected by proximity sensor, and conveyer stops
- 3. Fixture fix the workpiece tray
- 4. Robot camera detects Landmark and QR code (on landmark) of workpiece, to localize the workpiece, and recognize the number of the deburring process to execute.
- 5. Deburring process is executed according to the workpiece part number.
- 6. Fixture released, the processed workpiece is send to the next station.

2.4.2 Analysis and Verification of the Concepts

Then, analysis can be made based on this conceptual layout. For example of the deburring operator example above:

Number	Topics	Analysis	Further idea
1	For this application, is	End user teaches the	If I can't service the end-user
	there a method to let	deburring process for each	directly, I can cooperate with
	end-user changeover the	workpiece through the	regional system integrator, or the
	workpiece by	provided step by step UI,	distributor of my Robot Operator to
	themselves?	which integrates with the force	provide the service of Tray
		sensor hand guiding path	manufacturing, while I can provide
		generation feature of TM	them the example 3D CAD files of
		robot, and register the	it.
		process by let the robot record	
		the workpiece name on the	
		QR code on the tray.	

Number	Topics	Analysis	Further idea
2	How many parts inside	Standard Package included:	-I can choose one of the TM Plug &
	the application that I can	-TM Robot (from selling	Play force sensor, discuss with
	provide to the user?	channels of the robot)	them for the business to include
		-Force sensor (from TM Plug	that standard into my Robot
		& Play device maker)	Operator product.
		-Pneumatic Spindle (from	
		spindle maker)	Does the size of the conveyor
		-IPC for Operator UI, touch	should have several standard
		screen.	option products in my lineup to
			match different workpiece sizes for
			different customers? Also take
		Option provided:	TM12 into consideration for
		-The conveyor and fixture	different working range?
		module	
		-System base	About the protective cover, should I
		-Deburring tool	make it into several options, or
			cooperated with regional system
		Need to customized by	integrator, or my distributor?
		service	
		-Protective cover(according)	
		Not provided (added by user if	
		they need)	
		-Keyboard and mouse	
3	How customization will it	The change of the workpiece	If the workpiece size has big
	be, when the workpiece is	will be limited in the objects on	change and the original tray, even
	changed?	the tray and the whole system	the conveyor can't be used, end
		is remain unchanged, and	user can't handle this by
		user can handle it by	themselves. It is better to think the
		themselves.	grades for bigger workpieces in my
			Robol Operator lineup, and provide
			the end user a clear recognition
			Operator in my lineup
4	In what range of the	-Tool on hand deburring	How about deburring for 3D edge?
-	application can my Robot	application	Maybe I should make it into an
	Operator product cover?	-Metal parts of automotive	add-on of the Operator III
		industry.	
		-2.5D deburring of the	
		machined surface edge	
		accompanied with a 45	
		degree shape deburring tool.	

Analysis of the conceptual stage

The most important action in this stage is, decides for modules that:

- standard provided in the Robot Operator

- As Options in the provision accompanied with the standard Robot Operator.
- User added parts.

For the example above, due to the Robot Operator design, it seems not much parts are out of standard package. But according to different application, the range of the part the Maker can provide will be significant different. For example, the graph below shows a conceptual diagram of a 3D random bin picking Robot Operator.



Conceptual example of a 3D random bin picking Robot Operator

The Maker may realize that considering the flexibility of this Robot Operator, the container is hard to be a standard one, while the end user may already have theirs and does not willing to change to the one that Maker provided. In addition, how should the input of the workpiece be solved in this Robot Operator? It is also highly related to the arrangement and existed method in the end-user's factory. If there is no any universal solution, it is better to keep the flexibility to end-user. But, as a Robot Operator product, Maker should provide and design the method for them to add the rest part of the robot cell, and design in the software to let end user can easily configure the modifications. The conveyor is also not a key part in this solution that end user can add it simply by themselves.

For the gripper, the black module in the graph is a TM Plug and Play electrical suction gripper, and is benefitting to the Robot Operator because it will eliminate the needs for air valves

integration and the requirement for the compressed air in the place end user install this Robot Operator, so it is a good choice to include it into the standard package. In the contrast, if the finger / suction cap should include inside the standard package, it will be another question to be discussed. In this example, finally the maker may decide only the Robot, the HMI (integrated with IPC for Operator UI in the back), the 3D camera, the electrical suction gripper module is included in the package, and user can use their own containers, and need to construct the frames to mount the 3D camera, and the container by themselves, and this part should be taken into consideration of the Maker's product and business strategy arrangement*.

*For example, TM Plug & Play ecosystem also includes some digital on-line design service provider that can let end user quickly construct the frame by themselves.

The consideration of the flexibility arrangement is important, and the list below is suggested to taken into consideration:

- Which part can be/need to be done by the user?
- Reducing customization effort by
 - A more universal design
 - Easy to replace to TM Plug and Play devices
 - Let user easy to construct according to their own needs.
 - Leveraging TM Plug & Play service provider.

If an application can only be achieved by highly customization, the range of case by case part is very high, and these parts are very hard for end user to build by themselves, then it is not so likely to become a successful Robot Operator and more like a traditional robot cell.

2.4.3 End-effectors

Many Robot Operators can be constructed simply by the combination of TM Robot, the end-effector, and the software (Operator UI), as a Robot Operator Kit and without other equipment. The end-effector is an important part in the Robot Operator, and is related to the product concept and the business model of the Robot Operator product. It should be taken into consideration in the conceptual stage of the Robot Operator. This section will introduce what should the Maker consider and check.

The first question that Maker need to analysis is:

Is the end effector need to be changed every time when workpiece change?

If so, Maker needs to arrange a series of options accompanied with the Robot Operator product to the end user and need to pay more effort on it. But, in most case, maker can separate the end-effector into 2 parts:

- A. Actuator
- B. Tool tip

There are some examples:



Examples of End Effectors for Different Application.

The modularizations of these 2 parts should be considered.

For Actuator, makers are recommended to view TM Plug & Play supported peripheral device list first. These gripper/actuators already has mechanical interface to TM Robot, and having the software packages for manipulating, which can be simply integrated into the Robot Operator you created. Examples of TM Plug & Play Devices as below:



TM Plug & Play Devices for Different Application.

In common, it is welcome to include the actuator in the package of the Robot Operator product, while the tool tip is highly related to the customer's work piece and is the part that need to consider about the business model of the Robot Operator.

Only the application that a universal end-effector can meet the need of application, that may

have no need of the consideration of actuator and tool tip, like the palletizing application Robot Operator case which the end effector is just a flat suction cap array with defined size– Maker may provide the end-effector by original design, or directly find it in the TM Plug & Play list, include both the actuator and tool tip, because in this type of end effector it usually provides both of them.

	Example	Consideration of the provision of end	Further consideration
1	Case Palletizing	effector Standard: Existed electrical area suction module in the market Standard: Original design adjustable suction cap array Options: Different suction caps	 Discuss with the device maker about includes their product in the designed Robot Operator. 1. Provision of the suction cap because they are consumables. 2. Universal mounting hole designs that the distributor, system integrator, or even end user of the Robot Operator product can replace the standard suction caps according to their application cases.
2	3D random bin picking	 Standard: a. TM Plug & Play electrical suction actuator b. A L shape tool tip constructed by sheet metal with a suction cap. This may cover 30% of the application, and is easy for the user to trial for the first time. Options: a. Different shapes of the sheet metal tool tips having the combination of the lateral shift distance and length to the actuator flange. b. An adjustable design to have suction cap array. c. A flange to include the flexibility of the combination of suction cap and gripper due to some needs of the random bin picking that some pose of the part may need external/internal gripping for hole structures . 	 The consideration of the consumable suction caps, and the design to let user to replace the type of suction caps by themselves. Discuss with the distributors of the Robot Operator products that can customize the tool tip in the local sales region.

Here are some examples of the actuator-tool tip consideration:

3	Deburring	Standard:	Discuss with the distributor of the Robot
5	Debuining	An electrical or pneumatic rotary tool as	Operator about beloing the end user to
		the actuator with jaw of sleeves that can	choose and purchase of the tool tin from
		clamp different kind of tool tin. User can	the providers in local region
		change the teel tip in the least market	
		choose the tool up in the local market	
		according to their workpiece material and	
		size, under the help of the distributor of the	
		Robot Operator.	
4	Assembly	Standard:	Discuss with the distributors of the Robot
		a. An electrical gripper actuator.	Operator products the possibility of the
		b. A tool tip (finger) for trial	customization of the tool tip in the local
			sales region.
5	Pick and	Standard:	Discuss for the most suitable business
	place	No standard gripper.	strategy with the gripper makers.
		Options:	
		a. Many electrical / pneumatic grippers as	
		an option lineup that supports the	
		Robot Operator, provided by the	
		Maker. They are pre-arranged in the	
		design of the software that user can	
		easily choose.	
		b. Let user to choose the gripper that	
		support TM Robot and buy them by	
		themselves, while in the software the	
		design of import and set the software	
		packages of the gripper are provided.	

Maker may consider building the software options in the UI of the Robot Operator if a series of options of end-effectors are provided accompanied with the Robot Operator products. If there are some part or all part of the tool tip is designed that need local manufacturing, the ability of setting of the local made tool tip should be taken into consideration of the Robot Operator's software design. Makers can find further guidance and example of this topic in *Chapter 4. Common Technique*.

2.4.4 Software Conceptual Design Stage

After the systematic concept building stage, the work of the software conceptual stage decides if the Robot Operator can really be used by an end user.

To start the software stage of work, there are 2 pre-works to do.

1. Understand the software structure and the working model of the TM Operator Suite.

2. Check the intelligent features and mapping them into the software design plan.

For the pre-work 1, first read the concept of the working model of the TM Operator Suite showed below, and read the rest part of this manual.



The structure and the working model of TM Operator Suite

The diagram above shows the relationship of the major software in the TM Operator Suite. User starts the Robot Operator and sees the "Operator UI" first, and then operates the step by step UI from beginning to end. In some steps, specified TMflow pages pop ups, like vision, subflow, and setting pages that Maker arranges. So, a seamless user experience is created, while the existed TMflow functions can be leveraged by the Maker. The mechanism behind this user experience is that the Operator UI start up the TMflow Client in the beginning, and send requests to manipulate it, while the TMflow feedback the information like variables and data of the robot to the Operator UI.

The request, or command from the Operator UI to TMflow includes:

- Show/hide TMflow client, pop up specific pages of TMflow.
- Log in the robot through TMflow client.
- Open the pre-arranged projects/vision jobs and pop up.
- Copy and create new projects/vision jobs.
- Set and get the value of each variables and data like Base, Point, and TCP in the project or global variables.
- Get robot status data.



NOTE:

For these major communication and command of the Operator UI with TMflow, refer to *Chapter 4/ Chapter 5.*

In addition, the Modbus/Ethernet function is sometimes established between Operator UI and the robot directly, to leverage the Modbus/Ethernet function to get more robot status data. For the table of Modbus/Ethernet function of TM Robot, refer to the manual of TMflow.



IMPORTANT:

In general, the Operator UI and its function is established by these 2 major concepts:

- 1. Initiating the Operator UI, and jump into/jump out specific TMflow client pages / third party software.
- 2. Using Operator UI to manipulate the pre-arranged variables in TMflow to set the behavior of the robot.

Finally, the dashed block and arrows showed in the diagram indicates the integration with TMstudio Palletizing Wizard. This will be further explained in *Chapter4.4 CTQ4. Call External Executive File* of this manual. For the integration of the Maker's original-built software, Maker can also refer to these dashed blocks, in TM Operator Platform, the 3rd party software works like the work model of TMstudio Palletizing Wizard.



NOTE:

The start-up command that the Operator UI can send out, is a general execution of EXE. File function with parameters. Refer to *Chapter4.4 CTQ4. Call External Executive File* for the call up of the 3rd party software.

The second pre-work of the software stage of work is to list and check the possible intelligent features that TM Robot provided, and consider how to use these intelligent features to build a flexible Robot Operator. Examples are listed below. For further understand these functions, refer to the related manuals listed in *Chapter 1*.

Features of TM Robot	Description
Landmark of TMvision	Locate the jig, tray, tables and machines in the environment by taking a
	picture by the built-in eye-in-hand camera, which provide the maximum
	flexibility to the arrangement of the Robot Operator.
	Removes the precise and expensive base structure of the robot.
TMvision	Localization of the workpiece by TMvision. Removes the needs of precise and
	expensive jigs. Includes powerful adjustment and enhancement of machine
	vision.
Smart-Pick	A step by step vision localization function that simplify the standard vision job

TM Operator Platform User Manual

	and using landmark only to create the vision workspace instead of using		
	calibration plate. Suitable for regular pick and place application that easy for		
	end user to change over the workpiece by themselves.		
TM 3D vison (licensed)	Solution for 3D random bin picking application.		
External cameras (licensed)	Eye-to-hand localization and picking, AOI and upward looking localization for		
	workpiece on hand application that add the fixed camera to the control box,		
	programming in TMflow.		
Communication functions	Modbus, Ethernet, RS232, Profinet(Option), to communicate with other		
	devices.		
Force Control Node	Integrates with force sensor to provide solution to force related application.		
Path recording function	Use the intrinsic hand guide function or force sensor hand guide function, to		
	record the hand guided path. Related application: spraying, deburring, and		
	polishing.		
Listen Node	Arrange Listen Node in the flow project to let the robot enters a passive mode		
	that execute the commands through Ethernet from Operator UI, or external		
	devices. This is usually used in direct external control of the robot like		
	producing motion paths or command in other devices.		

Examples of intelligent features of TM Robot

By reviewing the possibility of the integration of these intelligent functions of TM Robot,

arrangement and usage of these functions can be put into the design of the software. Then, the software conceptual stage can proceed.

The work of the software conceptual stage includes these 2 main parts:

- A. To clearly define the steps of the UI manipulation
- B. To check in each UI, what the TM Operator Platform related features are used.

The Operator UI is regularly separate into these 2 main functional branches in the first UI page, which can also see in the example of UI design diagram.

- 1. The operation UI for factory worker/operator to start the daily work of the Robot Operator. It regularly includes choosing projects, choose work items, start/pause/stop, and a dashboard for showing production information when running .
- 2. The application set up step by step UI for factory engineers to do the first-time setting, or change over. This is the main value of the Robot Operator, compares to traditional robot cell.



NOTE:

To separate the accessibility of these 2 branches, Makers can design a password login mechanism. The example is provided in *Chapter 5.11 ATQ 8: Protection of Builder Project*.

After constructing the concept of these 2 major UI branches, the Makers can start from the work item A. For work item A, the main target is to construct the UI structure to succeed the Concept of Automation Process that Makers made in *Chapter 2.4.1.* The technique of the UI design like the graph below is suggested.





The picture above shows a typical method of the UI design arrangement. Each blocks represent an UI page, and each wire with arrow represent a functional button on the UI. It also present the relationship of the main UI program with the TMstudio software, for the palletizing case that integrates with TMstudio palletizing wizard, that the UI creates by UI builder in TM Operator Suite jumps to TMstudio palletizing wizard in one of the steps, and jumps back to the UI after the manipulation sequence of the TMstudio palletizing wizard.

Then, for the work item B, after reads the rest part of this manual and understand the software

mechanism of the TM Operator Platform, Maker can proceed to add the techniques and check the feasibility of the UI, by adding the analysis of the techniques into the UI design diagram like the pictures below:



Adding techniques to the UI design diagram



NOTE:

The techniques can be found in Chapter 4/ Chapter 5.

IMPORTANT:

The definition of project is important for the user experience, while the Operator UI is an integration of multi-software. TMflow has its project files, and Operator UI has its setting variables that user would like to manage them between usages. TMstudio and the 3rd party software also have their files. Makers can use the function of the Operator Suite to make original designed project to link and establish the relation between these files, so the end user can have a more unified user experience.

For example, a Robot Operator's "Project" may be constructed by the combination of these three project types:

- Project: user project in this UI (as known as txt file)
- TMflow Project: Project in TMflow
- TMstuido Project: Project in TMstudio

Further description and example can be found in Chapter 4.13 CTQ13 : File Storage.

2.4.5 Software Development and Trial Stage

After the plan for the software is finished, the work of software development and trial stage can start.

In this stage, it is recommended to proceed according to the sequences listed below:

- A. Construct the TMflow project
- B. Trials the flow functions
- C. Application Validation
- D. Build the Operator UI
- E. Trial run and adjustment

The detailed descriptions for each work item are listed below:

No.	Work item	Description	
А	Construct the TMflow project.	Build the TMflow project with TM Robot first.	
		-Draw out the flows with motion, logics, and vision nodes.	
		-For the key parameters, set the variables that need to communicate	
		with Operator UI, and list them down.	
		-Establish the branches of user choices which will be provided in the	
		Operator UI and use logic nodes and set variables for the branches.	
В	Trials the flow functions	Makers can trial the flow with full function on a TM Robot. Although	
		the Operator UI is not constructed yet, Makers are suggested to	
		simulate the result of the total process designed for end user, by	
		jumping the function of pages and setting the variables manually on	
		TMflow. This will be a recursive working item to check the usability of	
		the Robot Operator.	
С	Application Validation	Adding the hardware and external equipment that other than TM	
		Robot itself, to validate the feasibility of the application, with real	



		workpiece
D	Build the Operator UI	After checking the feasibility of the application, Makers can start to create the Operator UI.
		In this stage, the developer works on his/her desktop computer or laptop with TM Operator Suite with Builder license installed on the computer. The computer can be connected to the robot to play as the IPC to have the initial trialing with the robot, using the Builder.
E	Trial run and adjustment	Integrate the Operator UI and TMflow project and fine tune the user experience on the prototype of the Robot Operator. In this stage, it is recommended to install the runtime on the IPC to connect to the robot, while the modification of the Operator UI is proceed on the desktop computer or laptop, and the result is downloaded to the IPC, for present of the real situation of the product.

Work items of Software Development and Trial Stage



IMPORTANT:

It is important to work according the order of the work items A~E. Make sure each step is valid and then moves to the next work item, or it may be hard to debug or may waste time on the process.



IMPORTANT:

During working item D, the developer should take the consideration of the final resolution of the planned display module of the Robot Operator product, to construct the Operator UI, to prevent wasting time to deal with the un-match situation with the real display module.



NOTE:

here are 2 licenses and 2 dongles of the runtime for Operator UI included in the standard package of the purchased TM Operator Suite, which can be used on the developer's desktop computer/laptop to use the TM Operator Suite, and installed on the IPC of the Prototype of the Robot Operator.

2.4.6 Make it into a product

In this stage, the Maker already has a prototype of the Robot Operator. Considering of make it into a product like business strategy with the device providers, the design of the packaging and test of it, and the standard and options of the Robot Operator products, should be decided and proceed.



IMPORTANT:

It is also important to lock the Operator UI content that Maker creates, to protect the content for selling. TM Operator Suite provides built-in functions for this purpose. Refer to *Chapter 5.11 ATQ 8: Protection of Builder Project* for this.

Note

NOTE:

TM Operator Suite also provides the possibility for Makers to establish their own license system to design the licensed function for advanced content of the Robot Operator. Refer to *Chapter 5.3 ATQ 2: How to establish Accessibility management system on builder* for this.

When selling the Robot Operator, each produced Robot Operator requires a runtime license of activating the created Operator UI. Contact the sales channel of your purchase of the TM Operator Suite for the purchase of these runtime licenses.



NOTE:

When selling the Robot Operator, if Maker integrates TMstudio Palletizing Wizard into the solution, it also requires licenses to sell them to the end user. Contact the sales channel of your purchase of the TM Operator Suite for the purchase of these runtime licenses.

3. TM Operator Suite

3.1 License

After purchasing this product, you will receive a hardware dongle and a product key. The authorization period is one year. During this period, you can enjoy software upgrade service. After the annual contract expires, you can choose whether to continue the contract renewal.

	Authorization Contents		Remarks
1.	1. Product Key: OperatorBuilder Package		Used by the developer computer of designing the
	• Builder: 1 unit		Dashboard
	 Number of Robots that can be 	2.	The authorized quantity of Builders, Number of Robots
	Connected: 1 unit		that can be Connected, Dashboard Console, Point can be
	 Dashboard Console: 1 unit 		added for purchase, please contact your supplier
	• Point: 100 units		
2. TMstudio Palletizing Wizard Exporting			
1.	Product Key: OperatorRuntime Package	1.	Used by the industrial computer of running the Dashboard
	 Number of Robots that can be 		on Robot Operator
	Connected: 1 unit	2.	The authorized quantity of Number of Robots that can be
	Dashboard Console: 1 unit		Connected, Dashboard Console, Point can be added for
	• Point: 100 units		purchase, please contact your supplier
2.	TMstudio Palletizing Wizard Exporting		



IMPORTANT:

- 1. After the authorization date expires, you can still use the Operator development kit, but cannot update it.
- 2. The product key and the hardware dongle will be paired during the software installation process. Once bound, they cannot be exchanged for use arbitrarily



IMPORTANT:

Point is the total quantity of Tags, Monitor and Action that you can use. TM Operator Suite provides you with 100 points to use. If the upper limit of use is insufficient, you can purchase additional Points from your supplier.

3.2 Installation

Before using the TM Operator Suite, prepare the database and SQL Server Management Studio first. It is recommended to install the database version as Microsoft SQL Server 2017 Express or above on Microsoft Windows 10 or above.

After the preparation is complete, users can install TM Operator Suite which includes the development tools and environment building of TM Operator. In TM Operator Suite - Setup, uses can install TMstudio, TMflow, and Builder in batches, and generate TMOperator Suite to launch as the unified entry point for the software packages.

To obtain TM Operator Suite, go to the official website to download **TM Operator Suite – Setup**. Decompress and execute TM Operator Suite - Setup.exe, check the desired items to install, and click **Install**.



NOTE:

If your computer comes with other versions of TMstudio and Builder, TM Operator Suite will start installing the correct version after the old version has been removed. If your computer has a newer version of TMflow, TM Operator Suite will start installing the correct version after the new version has been removed. If the version in the computer is an older version of TMflow, TM Operator Suite will execute the installation operation directly without removing the old version.

3.2.1 Builder Installation

Please insert one of the dongles as the builder dongle, and the other will be the runtime dongle.

Ste

р

1. Execute Installation File.
- 2. Click Next.
- 3. Check I accept the terms of the license agreement and click Next.
- 4. Fill the information in the fields below **User Name** and **Company Name**, and click **Next**.
- Click Browse to set the destination folder to install or leave it to the default path
 C:\Program Files (x86)\Techma and click Next.
- 6. Check the desired functions to install by demand and click **Next**. Check Demo to generate teaching settings and data for demonstration.
- Set the database connection between MainDB and LogDB and click Next. Click Browse to set thed database to connect. (The database is at the local site by default.) Input the credential for the database connection.

Refer to

<u>https://docs.microsoft.com/zh-tw/sql/relational-databases/security/choose-an-authenticatio</u> <u>n-mode?view=sql-server-ver15</u> for MSSQL Login verification instructions.

- 8. Input the Service Engine Name and IP, port number, administrator's email account, product key, and click **Next**.
- Input and confirm the Admin password for 8 alphanumeric characters or more including uppercase lowercase, numbers, and punctuations marks or symbols, check whether to enable Data Rights Control or not, and click Next.
- 10. Input Dashboard Console IP address and port number. Click Next.
- 11. Click **Install** to start the installation. Click **Finish** when done.

Activate License

Insert the hardware dongle before activating the License.

Step

- 1. Click the icon of view hidden icons at the bottom right, right-click the TM LOGO icon, click **Activate License**, and click **Activate**.
- 2. Select **Offline Certification** or **Online Certification**. For **Online Certification**, the system will complete the activation process directly. For **Offline Certification**, continue to the next step.
- Copy the Activation code and go to <u>ols-tri.azurewebsites.net</u> for Certification with the browser.
- 4. After clicking **Activate** on the webpage, a License code will be generated. Copy the license code.
- 5. Return to the Activate License window and paste the License code. Click Activate to

complete the Activation process.

TMmanager System Removal

Step 1: Go to the Control Panel and find the "Uninstall Program" screen.

Step 2: Find TMmanager and right click "Uninstall".

Step 3: Wait for the Removing screen, click "Yes (Y)".

Operator UI Builder - InstallShield	Wizard	
Do you want to completely remo	we the selected application	and all of
its features?		

Step 5: Wait for the removing screen. If there is data backup, the file will be under this path.

Step 6: Click Auto Restart and click "Finish" to complete.

3.2.2 TMflow Installation

TMflow is the software that can set and program the motion flow of TM Robot. The installation steps are as below.

TMflow - InstallShield Wizard

Welcome to the InstallShield Wizard for
TMflow

The InstallShield(R) Wizard will install TMflow on your computer.
To continue, click Next.

WARNING: This program is protected by copyright law and
international treaties.

Step 1. Start installing TMflow.



Step 2. Click **Change** to select the destination path to install or leave it to the default path and click **Next**.

🔂 TMflow	🛃 TMflow - InstallShield Wizard 📃 🗾 🔁							
Destinati Click Ne	と							
	Install TMflow to: C:\Techman Robot\TM flow\			Change				
InstallShield		< Back	Next >	Cancel				

Step 3. Click Install to start installation.

🚰 TMflow - InstallShield Wizard	×
Ready to Install the Program	
The wizard is ready to begin installation.	
If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.	
Current Settings:	
Setup Type:	
Typical	
Destination Folder:	
C:\Techman Robot\TM flow\	
User Information:	
Name: A0960032	
Company:	
InstallShield	
< Back Install Cancel	

Step 4. Click Finish when the installation completed.	
· _	



3.2.3 TMstudio Installation

TMstudio can simulate the Palletizing scenario you build, and conduct collision checking with the Collision Check node in TMflow. If your application is not palletizing, or does not need to check collisions, then it is not necessary to install this software. Refer to the following for related installation steps.

Step 1. First install C++ Package.

TMstudio	- InstallShield Wizard
۲ ۵	Mstudio requires the following items to be installed on your computer. Click Install to egin installing these requirements.
Status	Requirement
Pending	Microsoft Visual C++ 2010 Redistributable Package (x64)
Pending	Microsoft Visual C++ 2010 Redistributable Package (x86)
Pending	Microsoft Visual C++ 2017 Redistributable Package (x86)
Pending	Microsoft Visual C++ 2017 Redistributable Package (x64)
	Install Cancel

Step 2. Start installing TMstudio, select the path you want to install and click Next.

🕼 TMstudio - InstallShield Wiz	zard	×
ع	Welcome to the InstallShield Wizard for TMstudio	
	The InstallShield(R) Wizard will install TMstudio on your computer. To continue, click Next.	
	WARNING: This program is protected by copyright law and international treaties.	
	< Back Next > Cancel	

Step 3. Select the installation directory

🔂 TMstudi	o - InstallShield Wizard	— ×	
Destinatio	on Folder At to install to this folder, or click Change to install t	to a different folder.	
	Install TMstudio to: C:\Techman Robot\TMstudio\	Change	
InstallShield -	< Back	Next > Cancel	

Step 4. Click Install to start the installation.

🔂 TMstudio - InstallShield Wizard
Ready to Install the Program
The wizard is ready to begin installation.
If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.
Current Settings:
Setup Type:
Typical
Destination Folder:
C:\Techman Robot\TMstudio\
User Information:
Name: A0960032
Company:
InstallShield
< Back Install Cancel

Step5. Installation is complete.

🔀 TMstudio - InstallShield Wiz	zard
	InstallShield Wizard Completed The InstallShield Wizard has successfully installed TMstudio. Click Finish to exit the wizard.
	< <u>B</u> ack <u>Finish</u> Cancel

Step 6. Ask whether to restart the computer. Because TMflow and TMmanager also need to be installed, therefore, click No at this step. Restart after the installation of all the software to be installed are complete.



3.2.4 Program Entry

After the installation is complete, a program entry will appear on the desktop .



After opening the Program Entry, click Start to execute the software packages individually. For the software that has not been installed, the corresponding Start button will not be clickable. If software needs to be reinstalled, then use TM Operator Suite – Setup again to install the software correctly before enter.

Omron - TM Operator Suite - Setup	×
Omron TM Operator Suite provides the capability of designing an application-specific "operator" customized UI/X. An "operator" is a step-by-step wizard interface for end user setup, changeover simulation, and runtime monitoring.	with
\checkmark	
Builder 4.04.1640	
This tool provides capability of designing an easy-to-use, customized GUI for end user.	
	_
TMflow 1.80.5400	
This is the robot control software with graphical flow programming. Note that only V1.80.5400 and above is compatible with Operator Suite.	
	_
TMstudio 1.16.1000	
An offline tool for environment setting and simulation. Built-in application wizard streamlines specific application setup.	
Install V1.00.60	00

3.3 OperatorBuilder package - general structure of SW

Operator Suite is an integrated software package including Builder, TMflow Client and TMstudio Palletizing Wizard. The overall introduction is as below:.



Through Builder, you can program the Step by Step Dashboard according to the application and release it to the Dashboard Console for use. Dashboard has the ability to use TMstudio and TMflow, allowing you to develop and integrate the TM Robot simulation software and the TM Robot process to program software's interactive human-machine interface.



3.3.1 Relationship of Builder, Dashboard Console, and Service Engine

Database provides data storage and writing; Builder can create Devices, Tags, Monitors, and Actions and create a Dashboard. The completed Dashboard can be released by Builder to Dashboard Console. Service Engine will save the data in DB for Dashboard to display the data in charts and reports. The communication with the robot on the Dashboard can be connected to the Robot through the Service Engine's TCP/IP communication port and Modbus port. If you want to write the program revision such as Tag and other information content, it also can be communicated through Socket in the format specified by TM.



3.3.2 Programming

If it is in the development stage (when the developer programs the Dashboard), the developer must first install the database, Builder (including Service Engine, Builder), TMflow, and TMstudio in the computer where the Dashboard is developed (to decide whether to install the database according to the usage situation). After installation is complete, according to your application, use the TMflow to program the process of robot running, and then use the Builder to create a simple operation Dashboard for your process. You can also integrate the functions of TMflow and external resources (such as third-party software) into the Dashboard (refer to Appendix A - TMflow Command for the commands that can be used by TMflow). For the use method of Dashboard, please refer to the chapter 3.5.4 Dashboard Designer.



3.3.3 Runtime

If the Operator's human-machine interaction interface and process have been developed and the developer wants to sell, the database-related packages and Dashboard Console and Service Engine on the computer of the Robot Operator to be shipped to the customer must be installed first (refer to the chapters in the Instructions 3.5 Builder for the installation of Dashboard Console). Use the Builder to transfer the project in the developer's computer to the Dashboard Console of the target PC via the flash drive or the deployment function (refer to the chapter for detailed deployment methods 3.6 Dashboard Console). Then export the database in the developer's computer to a flash drive through the export function of the Builder, and import it from the Dashboard Console function in the Robot Operator computer for it can be used.



NOTE:

The Deploy function can only release project, and cannot release databases. Therefore, the data transfer of a database can only be imported and exported via a flash drive.



IMPORTANT:

Robot Operator's computer is not required to install the Builder.

Relationship between Dashboard/UI/Builder



3.4 Environment

3.4.1 Introduction of Environment Settings

This chapter introduces the preparations and steps required to implement this project.

3.4.2 Software installation and version check

Confirm the version of TM Operator Suite is correct (refer to 1.5 SW Constructing). In addition, please make sure that the Hardware Dongle has been inserted into the correct hardware.

3.4.3 Network Connection Confirmation

Use Ethernet to connect Robot Operator's IPC and Robot Control Box. And make sure they are in the same Local Area Network Domain (LAN).

As shown in the figure below, use the Ethernet port on the robot controller.



Such as in the following example, set the IPC of Robot Operator to the IP of Ethernet IPv4 to 192.168.133.253.



Enter ping 192.168.133.44 in the command prompt to check the connection between the IPC of Robot Operator and TM Robot.

Command Prompt - ping 192.168.133.44
Microsoft Windows [Version 10.0.17763.1039] (c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\jerry.chen>ping 192.168.133.44
Pinging 192.168.133.44 with 32 bytes of data: Reply from 192.168.133.44: bytes=32 time=2ms TTL=128 Reply from 192.168.133.44: bytes=32 time=1ms TTL=128 Reply from 192.168.133.44: bytes=32 time=1ms TTL=128

- 3.4.4 Dashboard Console Firewall Whitelist Setting
 - 3.4.4.1 SQL Server through the Firewall

Before setting up the firewall, record the path of the current SQL service execution file.

Step 1: After opening the "Task Manager", click "Details".

Step 2: Find "sqlservr.exe", and right click.

Step 3: Click "Open file location".

Step 4: Record the path name after opening the folder.

👰 Task Manager						-		×	
File Options View	N								
Processes Perform	ance App	history Startup	Users Details	Serv					
		1							
Name	PID	Status	User name	CPU	Memory (p Descriptio	n		^	
sqlservr.exe	5472	Running	MSSQLSER	00	180.460 K SOL Server	Windows	VT - 64 Bit		
sqlwriter.exe	2160	Running	SYSTEM		End task	SS Writer	- 64 Bit		
Ssms.exe	1424	Running	darien.wei		End process tree		C		
svchost.exe	704 920	Running	NETWORK		Set priority	for Wind	IOWS SERV		
svchost eve	940	Running	SVSTEM		Set affinity	for Wind	lows Serv		
svchost.exe	948	Running	NETWORK		Analyze wait chain	for Wind	lows Serv	~	
					Debug			_	
 Fewer details 					UAC with validation		End tas	k	
					Carata duran file				
		0 malia	ntion Tools Bir		Create dump file				
		Applic	ation loois bii		Open file location	3			
File Home	Share	View M	anage		Search online				
I	i k	Cut			Properties	item 🔻		💽 Open 🔻	🕂 Select all
		Copy path			Go to service(s)	access •		👌 Edit	Belect none
access	Paste	Paste shortcut	to to to	+	folder	_	Properties	History	💾 Invert select
Cli	ipboard		Orga	nize	New	r i i i i i i i i i i i i i i i i i i i	Ope	en	Select
$\leftarrow \rightarrow \vee \uparrow$	> This	PC → Local Disk	(C:) > Program	Files	> Microsoft SOL Server >	MSSOI 14.1	ASSOL SERVER	> MSSOL	> Binn
		- Local Disk	(ci) / region						
🖈 Quick access		Name			Date modified	Туре		Size	
Desktop	4	salscriptdow	ngrade.dll		8/22/2017 10:16 PM		ation extens	28	KB
L Downloads		salscriptupar	ade.dll		8/22/2017 10:16 PM	A Applic	ation extens	5,7321	KB
Documente	1	salservr.exe	1		8/22/2017 10:17 PM	A Applic	ation	474	<b< td=""></b<>
		Y_ salservr.exe.c	onfig		8/22/2017 10:16 PM	A XMLC	onfiguratio	11	KB
Pictures	A	salsvc.dll	5		8/22/2017 10:17 PM		ation extens	157	KB
1018		sgltoolsmailu	itilities.dll		8/22/2017 10:17 PM	A Applic	ation extens	33 1	KB
👌 Music		saltses.dll			8/22/2017 10:17 PM	A Applic	ation extens	9.314	KB
09 items 1 item	relected /	73 KR						-,	
so items i i item s	selected 4	13 10							

Step 1: Find "Allow an app through Windows Firewall" in the Control Panel.

Step 2: Click "Allow another app".

Step 3: Click "Change settings".

Step 4: Click "Browsers..." and select the SQL service execution file path recorded in the previous page.

Step 5: Click "Network types".

Step 6: Select "Domain, Private, and Public".

Step 7: Click "OK".

Step 8: Click "Add".

Step 9: Click "OK".



3.4.4.2 Allow Service Engine apps through the firewall

Step 1: Go to the Control Panel and find the "Allowed apps and features" screen.

Step 2: Click "Allow another app".

Step 3: Browse C:\Program Files (x86)\Techman\ Operator UI Builder\\Service Engine\TM Service Engine.exe.

Step 4: Click "Add".

3.4.4.3 Allow Dashboard Console apps through the firewall

Step 1: Go to the Control Panel and find the "Allowed apps and features" screen.

Step 2: Click "Allow another app".

Step 3: Browse C:\Program Files (x86)\Techman\Operator UI Builder\Dashboard Console\Dashboard_Console.exe.config.

Step 4: Click "Add".

3.4.5 Confirm that the firewall settings have been completed

Step 1: Confirm that the two services have been added successfully, and confirm that the domain and private have been checked.

Step 2: Click "OK".

3.4.6 IP Settings

In order for the external computer to obtain information related to the robot, confirm whether the computer to be connected is connected to the same domain as the robot. The following will introduce how to set the network IP of each device.

3.4.6.1 Robot IP Setting

Set the page from the TMflow Robot system, click Network to modify the IP of the robot.

≡			<u> </u>	🕸 0 mm/s	100 % 😌 F5C	.9		i	ahl.
			System	Setting					
	A x	$\langle 5 \rangle$	**			Ū,			
	Language	System Update	Group	User Account	Network	Import/Export			
	23	*P.	Ū,	(
	Date Time	Administrator Setting	Network Service	Backup\Restore	Input/Display Devices	Auto Remote Mode			

As below, we modify the IP to 192.168.133.44.

	👤 🖺 0 mm/s	100 % 😌 F5C9		(POSSING)	i	12 13 13 14
Network setting						
Local Area Connection 2						
Intel(R) Ethernet Connection I21	7-LM #2					
Static IP IP Address	102 168 133 44					
Subnet Mask	255.255.255.0					
Default Gateway			×.			
Obtain DNS conver address aut	matically					
 Use the following DNS server a 	ddress:					
Preferred DNS server:						
Alternate DNS server:						
	OK —		*			

3.4.6.2 Computer IP Setting

Step 1: Select the [Start] button, then select [Settings] > [Network and Internet].

Step 2: Execute one of the following operations.

Step 3: For Ethernet, select [Ethernet], and then select the Ethernet you are connected to.

Step 4: Under [IP Assignment], select [Edit].

Step 5: In [Edit IP Settings], select [Manual], and then open [IPv4].

Step 6: To specify an IP address, enter the IP address settings in the [IP address], [Subnet prefix length], and [Gateway] boxes. In the example below, we set it to 192.168.133.253.

Step 7: specify the DNS server address, enter the primary and secondary DNS server addresses in the [Preferred DNS server] and [Other DNS server] boxes.

網路功能 共用	General	
連線方式: 🥃 Realtek PCIe GBE Family Controller	You can get IP sett this capability. Othe for the appropriate	ings assigned automatically if your network supports erwise, you need to ask your network administrator IP settings.
設定(C) 這個連線使用下列項目(O):	Obtain an IP a	address automatically ing IP address:
✓ ▲ 網際網路通訊協定第4版 (TCP/IPv4) ✓ ● PROFINET IO protocol (DCP/II DP)	IP address:	192 . 168 . 133 . 253
 ✓ Microsoft LLDP 通訊協定驅動程式 	Subnet mask:	255 . 255 . 255 . 0
 ☑ ▲ SIMATIC Industrial Ethernet (ISO) ☑ ▲ 網際網路通訊協定第 6 版 (TCP/IPv6) 	Default gateway	
 Link-Layer Topology Discovery Responder PROFINET IO RT-Protocol V2.3 	Obtain DNS se	erver address automatically
<	> OSe the follow	ing DNS server addresses:
安裝(N) 解除安裝(U) 內容(R)	Preferred DNS se	erver:
描述	Alternate DNS se	erver:
傳輸控制通訊協定/網際網路通訊協定 (TCP/IP)。這是預設的廣 網路通訊協定,提供不同網路之間的通訊能力。	域	ngs upon exit Advanced
		OK Capce

3.4.6.3 Set the Service Engine IP pointed by TMflow

In TMflow:

- 1. Navigate to \equiv , and click **Setting**.
- 2. Click the **Builder** icon.
- 3. Check Enable Builder.
- 4. In the fields below **Server Setting**, fill the IP address and port number of Robot Operator that comes with service engine.
- 5. Click **OK** when done.

In the figure belows, the IP address field below **Server Setting** is set to 192.168.133.253 since the IP address of Robot Operator in our example comes withs 192.168.133.253.

	2 № 0 mm/s 100 % 😌 F5C9	
TMmanager		
	 Enable TMmanager Enable auto upload data to server Server Setting IP: 192 168 133 253 Port: 5430 	
	ОК	

3.4.6.4 Set Service Engine IP

Builder should be executed on static IP. Builder uses the name as the licensed system of Service Engine. When Windows starts, Service Engine will run automatically. You can find it in the Windows taskbar. The service engine icon turns from monochrome to color when the service engine is running.

- 1. Check the IP on Service Engine.
- 2. Right single click on Service Engine, then click Stop Service Engine.
- 3. Edit Service Engine configuration.

A structure l'annue	🛃 System File Config			-	□ ×
Activate License					
Edit MainDB Configuration	Service Engine Name :	ServiceEngine1	Socket Connect Wait Handle :	2000	
Edit LogDB Configuration	Service Engine IP :	192.168.133.36 ~	Run Action By Task :	True	~
Edit Service Engine Configuration	Service Engine Port :	5430	Robot Socket Use Task :	False	~
Start Service Engine (Windows level)	Socket Server Port :	5434	Socket Receive User Thread Pool :	False	~
Stop Service Engine (Windows level)	Management Port :	5431	Thread Pool Count :	10	
Restart Service Engine (Windows level) (Status: Stopped)	Modbus Slave :	False ~	SFC Statistical Analysis Loop Time :	60000	
Poload Service Data	Modbus Slave Port :	502			
Reload Service Data	Connect Retry Count :	0			
Trouble Shooting	Monitor Wait Time :	1000			
Error Message				Cancel	Check
	191 1				

IMPORTANT:

Builder should run on static IP.

- 1. If you unplug the PC's Ethernet, in some cases, the PC will lose the static IP. Connect PC to the LAN
- 2. Auto Assignment of IP, the Service Engine sometimes cannot start correctly. Use static IP instead.
- 3. If the IP on the IPC of the Robot Operator is changed, then the Service Engine IP needs to be changed.

3.4.6.5 Confirm that the items are ready

Please browse the list below to check if each item is ready.

Item	Reminder	Check
Check whether the developer's	Whether has the developer's computer installed	
computer software installation is	TMflow. Builder, (TMstudio).	
Has the Dongle been installed		
correctly		
The network connection between the	Requires Status IP	
IPC and ARM of Robot Operator		
requires static IP		
TMflow System Page - Whether the		
setting on the Network Setting Page		
is complete		
TMflow Setting Page - Whether the		
setting of IMmanager Page is		
Has the TMflow Setting Page,	G- ▲ № 0 mm/s	
turned en2	Modbus Slave	
	TCP Disable	
	STATUS: Enable	
Does TMstudio run normally.	(If it cannot run, execute SetENV.bat in the	
Lised in auto startun TMflow Client to		
check whether TMflow pops up		
Can the TMmanager connect with the		
robot		
Device Setting in Builder	Check whether the corresponding robot IP is	
<u> </u>	correct.	
Builder – Device TMservice	Check whether the TMservice Engine IP is	
	correct.	

Following is the setting example, the connection status can be confirmed through the Display on the top right of the Builder.



Target	Setting Page	IP
Robot	TMflow System Page Network	192.168.133.44
Robot	TMflow Setting Page TMmanager	192.168.133.253
Industrial Computer of Robot Operator	Builder –Device TMservice	192.168.133.253
Industrial Computer of Robot Operator	Builder –Device Robot	192.168.133.44





NOTE:

The connection status can be confirmed through the Display on the top left of the Builder.



IMPORTANT:

The Dashboard created by Builder can obtain TM Robot data from Ethernet via Modbus. Make sure that the Modbus settings of TMmanager and TM Robot are correct, otherwise the connection between Builder and TM Robot may fail.

3.5 Builder

3.5.1 Builder's role

Builder, Service Engine, Dashboard Console, and Microsoft SQL Server Express (required to be TM Operator Platform User Manual 57 installed separately); Builder provides a customizable human-machine interface, integrates the communication interface of TM robots, and provides real-time data of production, visual data chart functions etc. Service Engine is the background service of the data processing program. Dashboard Console is the Dashboard execution container, and the UI made by the Builder can be placed on the container to run;. Microsoft SQL Server Express is the database.

Builder is the environment for building Dashboard (UI is included in Dashboard). Developers can create a step-by-step Dashboard in the Builder, and integrate many functions from TMflow and external resources (such as third-party software) into the Dashboard. Dashboard can be deployed to the target PC (installed with Dashboard Console) and run.

Builder			- B X
Venetit huntit Heightin Optimistit	an	0	Saraca Engra Status Santus English
DODDOH 4.4 Sina Pata Jawa Beer Sawa Beer Sawa Beer Sawa Beer Sawa Beer Sawa Beer Sawa Beer Sawa Brand Beer Sawa Beer Sawa Br	Evence Securition A Toress Depresentation Evence Statement Evence Statement	Const Price Const Department Brance Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Department Departme	Properties * * * Province Control of Contro
ucentes === [] :::-] ::: ¥] ==: Onixet			Materian Ka Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Salar Sal

The project in Builder consists of two parts: UI editor and Code editor.

The UI editor is the area of building the Dashboard layout, and each item (such as buttons, picture boxes) is a component.

The Code editor is the area of Script programming in Builder, used to control the properties of components

Codename	Header	Content
(1)	Function toolbar	The functions provided by the Builder are opened by Icon or menu in this area
0	Dashboard editing toolbox	Dashboard of all components for users to edit
3	Set the item tree structure	Inspection of such items as Device, Tag, Monitor, Action, etc.
4	Function Setting	Set Device, Tag, Monitor, Action, etc. related content

5	Dashboard Editing Interface	Edit Dashboard, write script language interface
6	Properties of dashboard components	Set Dashboard component attributes

3.5.2 Service engine

Provides the list of Service Engine status, and can quickly check Service Engine information through simple query and screening.

*Note: This Service Engine status list is limited to the application level.

earch Service Engine Na	ame/ IP Address		Enable		Status	
					~	~
Service Engine Name	IP Address	Enable	Service Engine State	Download	Synchronize TMRobot Time	Start St
TMMService1	192.168.132.138	True	Start	Download	Synchronize	Start St

3.5.2.1 Search and Screening Functions

Service Engine - Manager	m ×							
Service Engine	(Applicatio	on lev	el)					
Search Service Engine N	ame/ IP Address		Enable		Status			
					~	\sim		
Service Engine Name	IP Address	Enable	Service Engine State	Download	Synchronize TMRobot Time	Start	Stop	
TMMService1	192.168.132.138	True	Start	Download	Synchronize	Start	Stop	

Search Service Engine Name/IP Address: input the Service Name or Service IP to query. Enable: Controls whether the Service Engine is enabled.

Connect: Service Engine is enabled.

Disconnect: Service Engine is not enabled.

Status: Service Engine Status.

Start: Service Engine is in Start Status.

Stop: Service Engine is in Stop Status.

Being Replaced: Service Engine is in Backup Status.

3.5.2.2 Service Status List

Search Service Engine Name/IP Address: input the Service Name or Service IP to query.

Controls whether the Service Engine is enabled.

Connect: Service Engine is enabled.

Disconnect: Service Engine is not enabled.

Status: Service Engine Status.

Start: Service Engine is in Start Status.

Stop: Service Engine is in Stop Status.

Being Replaced: Service Engine is in Backup Status.

Synchronize TMRobot Time: The time synchronizes connection to the Robot.

Step 1: Click "Synchronize"

Synchronize TM F	Robot Tin	ne
Synchronize TM Robot time with Ser	rvice Engine	
	Cancel	OK

Step 2: Click "Confirm", the message displays that the synchronization is successful; click "Cancel" to close this window.



Start: Click to Start the Service. Stop: Click to Stop the Service.

3.5.3 Basic concept

Builder basically consists of Devices, Tags, Actions, and Monitor. The Device can be PLC/Modbus Slave/TM Robot, etc.; The Tags can intercept the information of a certain Device, such as a Modbus Address, OPC UA node, a TM Robot variable, etc.; Action can issue command actions, such as sending email, downloading files, etc.; The Monitor can monitor data, and can write logic based on the monitored data, to issue Actions; and Script Language provides advanced users to write scripts to achieve the purpose of advanced applications.



3.5.3.1 Device

3.5.3.1.1 Device defines four types

TMRobot, Service Engine, Modbus Device, and Socket can communicate with the corresponding device, read or write device data according to different types. Device Form is as follows.

📖 Builder					- 🗆 ×
View(V) Tool(T) Help(H) Options(O)					
Designer Devices Tags	Monitors Actions	Groups	Service Engine Status TMMService1	•	Display
Devices 🛛 🕂 🗙	Device Form 🗙				<u> </u>
+ 🔤 🖶 🇉 – 💭 🖷 🍸 📗	N				
E 🚓 Default	Device Setting			Dashboard F	Picture
ModbusDevice	Device Name SocketDe	evice	Service Engine Status	Connect	Disconnect
SocketDevice	Device IP 127.0.0.1		🖌 Enable		
ن TMRobot	Device Type Socket	~]		
	Category Name Default		Set Category	Browse	Browse
	Description			Clear	Clear
	Saakat				
	SOCKEL				
	Service Engine	ServiceEngine	~		
	Socket Port	1111			

3.5.3.1.2 Device Explore Tree Diagram



Device Type Icon:



Service

Icon function (from left to right)



Dev	rices P ×
+	▥ ◨,ᅚ- ♡耳丁カ
+	<new> Add New Device</new>
Name	<rename> Rename Device</rename>
Ξ.	<addgroup> Add Group</addgroup>
Ē	<expand all=""> Expand Tree Diagram</expand>
•	<fold all=""> Fold Tree Diagram</fold>
	<refresh> Refresh</refresh>
	<copyitem> Copy</copyitem>
$\overline{\mathbf{Y}}$	<filter> Filter Query</filter>
1	<delete> Delete</delete>

3.5.3.1.3 Device Function

<New> Add New Device, Open New Form, Refer to "4. Device Form" for details.

<Rename> Rename Device

CAUTION: Before operation, the Device function or Service Engine Tray needs to be used to stop the Service Engine.

💀 Builder			- • ×	<
View(V) Tool(T) Help(H) Options(O)				
Designer Devices Tags Monito	rs Actions Groups Manag	Service Eng	ine Status ServiceEngine Display	
Devices 4 ×				
+ ฿, ᅚ⊢ 🕽 🖷 🍸 🛅				
∎–#n Default				
		Rename		
		Original Name Default		
		New Name		
			Cancel OK	



AddGroup>

Group Name: Fill in the new group name

Root Node: If the new group is the first layer, it is necessary to check for selection Parent Name: If the new group is a subgroup, it is necessary to select the Parent group name.

Add Category						
Category Name	Root Node					
Parent Category						
	Cancel OK					



Expand all subnodes.



)	<fold all=""></fold>
Fold	all subnodes,





Refresh Tree Diagram.



Detect the new robot is connected to the Service Engine.

After clicking Refresh (Refresh2), it will generate new robot items.

CopyItem>

*Device does not have a CopyItem function

Filter>

- 1. Input any condition
 - Device Type
 - Keyword
- 2. Select

The display results are as follows, Find displays the number of results.

Devices	5			-¤ X
R + 📼	□ 🖪, 🗉	💭 📲 🍸 i	Ì	
Device	Type TMRobot	-	Select	
			Cancel	
Keywo	rd		Find: 1	
■	Default 5 ³ TMRobot			

3. Cancel: Turn off Filter

Collete>

Click any Device item and press Delete, and a pop-up window will display Device and other related settings. Continue to delete after confirming the related items, all related settings will be deleted.

🔤 Builder									_		\times
View(V) Tool(T) Help(H)	Options(O)										
Designer Devices	Tags M	onitors Actions	Groups	Managemer	t	Service Engine Status	ServiceEngine	_	•	Displa	y
Devices	ب × الله 🔽 🛋			Assoc	iation						
∎ _ Default ☐ ∰ ModbusDevice	····			Name	TEST2			Total	2		
F ServiceEngine				#	Name	Туре	User	IsDirect			
E SocketDevice				1	TEST2	Devices		true			
∟ _p g Testrobot								Cancel	Next		

🔤 Builder								-		×
View(V) Tool(T) Help(H)	Options(O)									
Designer Devices	Tags M	onitors Actions	Groups	Management	Service Eng	ine Status ServiceEngine	_	·	Display	1
Devices	Ψ×									
+ 🔤 🖪, "E) 💭	🖫 🍸 🛅									
∎ n Default					Delete Item					
					The following items will b	be deleted	Total 2			
C. SocketDevice					Name	Туре				
r Testrobot					TEST2	Devices				
					Testrobot	Devices				
						Cancel	ок			

I. Device_Form

	Device Setting	g <u> </u>	CDashboard P	Picture
đ	Device Name	Service Engine Status	Connect	Disconnect
2	Device IP	€ Enable		
6	Device Type	Service ~		
6	Category Name	Default Set Category	Browse	Browse
G	Description		Clear	Clear

• Device: Input Device Name, Names are case-insensitive, so upper and lower case spellings will be regarded as the same name. Special characters and blanks cannot be used in names.

- Device IP: Input Device IP.
- Device Type: Four types of TMRobot, Service, Modbus Device and Socket.

• Enable: Checked represents the device is connected with Service Engine; if not checked represents the Device is not connected with Service Engine, make a record on the system.

G Category Name: Add / Modify to the Category (Set Category).

Devices $\mp \times$	Device Form	×		
+ ┉ 🖳 ་ㅌ▸ㅡ 🎧 🖷 🍸 🛅	🔁 🖾			
<u>∎ mar Default</u>	Device Settin	ıg —		
ModbusDevice	Device Name	TEST2	Service Engine Status	Stop
	Device IP	127.0.0.2	Enable	
TEST2	Device Type	Service	~	
Testgp	Cotogory Nama	Default	Ret Category	
		Delaun		
	Description			
	L			
		<i>∑</i> _{ <i>₹</i>	•	Service Engine Status
Designer Devices Tags Monitors	Actions	roups Managemen	t.	Cervice Engine olalus
DeviceName		x		
	7 🖻			
🕇 📼 🛛 🖳 📕 👘 Default				
Default Testgp				
- Servici		T2	Service Engine Status	Stop
r ⊨ 		.0.0.2	Enable	
TEST2		vice	~	
lestgr		ault	Set Category	
			out out og of y	
	· / (citorii)	oroopo managemen	·	
Newices	Device Form	x		
	1			
	-Device Settin			
ModbusDevice	Device Name	TEST2	Service Engine Statue	Stop
🗚 ErviceEngine	Device Name	107.0.0.0		Stop
SocketDevice	Device IP	121.0.0.2	Enable	
TEST2	Device Type	Service	~	
	Category Name	Testgp	Set Category	
	Description			
		L		

• Description: After setting the prompt, move the cursor to this device in the Device list and display the content of Description.

Dashboard Picture: Set the connection status icon of Device, displayed on Robot
 Control1, Robot Control2, DeviceActiveX components.

Device-Robot

C Device Setting	Dashboard Pict	ashboard Picture ——			
Device Name		Service Engine Status	Connect	Disconnect	
Device IP		Enable			
Device Type	TMRobot	~			
Category Name	Default	Set Category	Browse	Browse	
Description			Clear	Clear	
Service Engine		~			
IPC Connect Po	ort 5430				
Modbus Port	502				
☑ Use Modbus	TCP				
Nonsy	nchronous: keep th	process running without checking			
5 O Synch	ronous: occupy the	process until data received			
Modb	ous Retry count	1			
Modb	ous Retry interval	2000 ms			
Modb	ous Timeout	10000 ms			

• Service Engine: Set the Service of managing the Robot.

- **2** IPC Connect Port: The Socket Port provided by the device is fixed to 5430.
- Modbus Port: The Modbus Port provided by the device is fixed to 502.
- Modbus Asynchronous (default) (Read / Write two channels).

(Sends request, continues to send request without waiting for response)

(The order of receiving responses is not necessarily the same as the order of sending requests)

|--|

Modbus Synchronization

(Received the request. Must wait for the request to be completed before executing other requests)

- Modbus Retry count: Retry count (1~32767)
- Modbus Retry interval: Retry interval (100~32767)
- Modbus Timeout: Timeout (100~32767)



3.5.3.1.4 Device-Modbus

	Device Settin	g					Dashboard F	Picture ———
	Device Name	ModbusDev	ice		Service Engine	Status Stop	Connect	Disconnect
	Device IP	127.0.0.1			Enable			
	Device Type	ModbusDe	vice	\sim				
	Category Name	Default			Set Category		Browse	Browse
	Description				-		Clear	Clear
	ModbusDevic	;e						
D	Service Engine		ServiceEn	gine		\sim		
9	Modbus Port		502					
B	Nonsynchro	onous: keep t	he process ri	unning wi	ithout checking			
Ð	Synchronou	is: occupy th	e process un	til data re	ceived			
	Modbus R	etry count	1					
	Modbus R	etry interval	2000	ms	3			
	Modbus T	imeout	3000	ms	3			

• Service Engine: Set the Service Engine of Managing Robot

• Modbus Port: Modbus Port provided by the device (1~65535)

Modbus Asynchronous (default) (Read / Write two channels)
 (Sends request, continues to send request without waiting for response)

(The order of receiving responses is not necessarily the same as the order of sending requests)



Modbus Synchronization

(Received the request. Must wait for the request to be completed before executing other requests)

- Modbus Retry count: Retry count (1~32767)
- Modbus Retry interval: Retry interval (100~32767)
- Modbus Timeout: Timeout (100~32767)



3.5.3.1.5 Device-Socket

	g		Dashboard Picture —
Device Name		Service Engine Status Stop	Connect Disconnect
Device IP		Enable	
Device Type	Socket	~	
Category Name		Set Category	Browse Browse
Description			Clear Clear
Socket —			
Service Engine		~	

- Service Engine: Set the Service Engine of Managing Robot
- Socket Port: Socket Port provided by the device (1~65535)

3.5.3.1.6 Device-Service

Navian Managa	SonicoE	naine						Connect	Disconnect
Jevice Name	ServiceEl	igine		Service Engine	status st	ап		Connect	Disconnect
Device IP	192.168.1	132.189		Enable					
Device Type	Service		~						
Category Name	Default			Set Category				Browse	Browse
Description								Clear	Clear
Service —									
SQL Connectio	on String	12a3e902-2f4	4e-43fb-9	70d-a80288f31t	010				
Redundant sys	stem			~	,				
Service Engine	Loop	1000 🕀	ms	Monitor Loop	Time 50		ms		
Database				Modbus Set	ting				
DB Max Size	7.5	9 GE	3	🗌 Use Modb	us TCP				
DB Check Tim	e 00:00:	00		O No	onsynchrono	ous: keep the p	rocess runn	ing without cheo	cking
Maximum DB [Data prese	erved 90		🔘 Sy	nchronous:	occupy the pro	ocess until o	lata received	
Monitor			\sim		Modbus	Retry count	0		
					Modbus	Retry interval	1000 r	ns	
					Modbus	Timeout	3000 r	ns	

• SQL Connection String: LogDB database to be connected

Device Settin	g —							-Dashboard F	Picture —
Device Name	ServiceE	ngine	Service Engi	ne Status	Start			Connect	Disconnec
Device IP	192.168.	132.189	Enable						
Device Type	Service		\sim						
Category Name	Default		Set Catego	у				Browse	Browse
Description								Clear	Clear
Service —									
SQL Connection	on String	12a3e902-2f4e-43f	b-970d-a80288f3	1b10					
Service Engine	e Loop	1000 m	is Monitor Lo	op Time	50		ms		
Database			Modbus S	etting					
DB Max Size	7.5	GB	🗆 Use Mo	dbus TCF					
DB Check Tim	e 00:00:	:00 ‡	0	Vonsynch	ronous: ke	eep the pr	ocess run	ning without che	cking
Maximum DB I	Data pres	arved 90		Synchron	ous: occup	by the pro	cess until	data received	
Monitor		~		Mo	ibus Retry	count	0		
				Mo	ibus Retry	/ interval	1000	ms	
				Мо	dbus Time	out	3000	ms	

Click the red box button.
1		0
1	•	4

Device Form ×				
	Database Source			
Device Setting		New	Edit	Remove
Device Name TEST2	(local)\SQLEXPRE	SS [TMMLogDB]		
Device IP 127.0.0.2	(local)\SQLEXPRE	SS [TMMDemoDB]		
Device Type Service				
Category Name Testgp				
Description				
Service				
SQL Connection String 12a3e902-2f4e-43ft				
Redundant system				
Service Engine Loop 1000 m:				
Database				
DB Max Size 7.5 GB				
DB Check Time 00:00:00				
Maximum DB Data preserved 90				
Meniter				
Wonton V				
	Orest	0.410		014
	Cancel	GetiD		OK

Select the database for the service to write the log.

- 1.2.1 Cancel: Cancels and returns to Builder.
- 1.2.2 GetID: Gets the database ID code.
- 1.2.3 OK: Selects the DB as the logDB of the service.

If you want to add/modify the database, the New or Edit button can be clicked to add/modify the database.

- 1.2.4 New: New database source.
- 1.2.5 Edit: Edit database source.
- 1.2.6 Remove: Remove database source.

ctions Groups Managemen	Database Detail	4
[hot	Connection type	Microsoft SqlClient 🗸
ce Setting		
Name TEST2	The Microsoft So Recourse it does not be a set of the set of th	I Data Provider is specific to Microsoft SQL Server.
⇒IP 127.0.0.2	it is faster than th	e OleDb Data Provider.
3 Type Service	However, it can o	only be used with SQL Server 7.0 or later.
ory Name Testop		
intian	Parameters	
	Friendly nav	796' TM4M4 DD
ce	Server	
	Default	
Connection String	Time out	I MINILOGDB
undant system		0
lice Engine Loop 1000	Use Inte	grated Security
abase		
Max Size 7.5 GB	User na	ime: sa
Check Time 00:00:00	Passwo	rd: *******
imum DB Data preserved 90		
nitor v	Cancel	Try to test OK
	Capcal	CatilD
	Cancer	Gend OK

1.3.1 Cancel: Cancel New / Edit database source.

1.3.2 Try Test: Try to connect to the database.

1.3.3 OK: Complete New / Edit database source.

 Redundant system: Set up a backup device to make Service zero downtime (master, slave).

(1): First stop the Service of the backup (Master Server) and the backup (Slave Server).

(2): Click Service Engine(Master Server).

(3): Select the backup Service (Slave Server) in the Redundant System and save.

(4): Start the Service of the Master Server first, and then start the Service of the Slave Server.

Service Loop Time: Service returns data interval time, Range: 500ms~32767ms.

Monitor Loop Time: Monitors interval time, Range: 50ms~32767ms.

• DB Max Size: Sets the maximum capacity of DB data, if it exceeds, start Monitor.

DB Check Time Daily regular check (hours: minutes: seconds) : whether DB is > 7.5 GB.

 Maximum DB Data preserved: Data retention days (default 90 days), deletes old data automatically when exceeded.

- Monitor when conditions are met **9**, executes the selected monitor (must be set).
- Modbus Asynchronous (default) (Read / Write two channels).

(Sends request, continues to send request without waiting for response)

(The order of receiving responses is not necessarily the same as the order of sending requests)



Modbus Synchronization

(Received the request. Must wait for the request to be completed before executing other requests)

- Modbus Retry count: Retry count
- Modbus Retry interval: Retry interval
- Modbus Timeout: Timeout



1. Data capacity check, send mail notification automatically when exceeded (MailSMTPServer must be set first)

SMTPServer is opened through Windows programs and functions
It can be set to backup data to the backup database before deleting data

 Modbus Synchronization (default) × Can select whether to use ModbusTCP (Received the request. Must wait for the request to be completed before executing other requests)

- Modbus Retry count: Retry count (1~32767)
- Modbus Retry interval: Retry interval (100~32767)
- Modbus Timeout: Timeout (100~32767)

3.5.3.1.7 Add New Device

Designer -	CHART		Allera			• 📫						
Devices	12+-12			0	Device ED LD	Form	8					
Default					Device 1	ettings				Y Da	shboard P	Schute
					Dence No.	ne		Sence E	igne Status Stop		Connect	Disconnect
					Device P			Enude		10		
					Device Typ	a Thirty	HR.	~				
					Category 1	iane 🔄		Set Cate	gory .		Browte .	Browne
					Description	•					Onar	Clear
					TMRobo	t						
					Service 8	ingine	[~			
					IPC Cure	and Post	8430					
					Middus	Post.	602					
					[2] Used	Auditus TOP						
						horaynchron	ILL Heep Theproces	a running without	checking			
					0	Synchronizati	rockey the process	until data receive				
						Modbus Reb	Court	1				
						Mudbus Ret	y interval	D 000				
						Modius Tree	Ren 1	10000				

Step 1: Click Device

Step 2: Click " + ", Device Form tab appears

Step 3: Can pin the Device window

Designer Devices Tags Monitors	Actions Groups Management Service Engin	ne Status ServiceEngine1 Display
Devices 🕴 🕴 🗙	Device Form ×	<u> </u>
+ == ฿, '≣→− ⊖ �� ▼ カカ		
n 👘 Default	Device Setting	Dashboard Picture
	Device Name Service Engine Status Stop	Connect Disconnect
	Device IP	
	Device Type TMRobot ~	
	Category Name Default Set Category	Browse
	Description	Clear
	_ TMRobot	
	Service Engine	
	IPC Connect Port 5430	
	Modbus Port 502	
	Use Modbus TCP	
	Nonsynchronous: keep the process running without checking	
	 Synchronous: occupy the process until data received 	
	Modbus Retry count 1	
	Modbus Retry interval 2000 ms	
	Modbus Timeout 10000 ms	

Step 4: Input Device Name, IP Step 5: Device Type

Designer Devices	Tags	Monitors	Actions	Groups	Management		Service Engine	Status Service	îngine1	·	Display
Devices		ų×	Device Form	×							
+ 🖂 💷, 🖘		Ì	13								
Default			Device Setting	9 9				-Dashboard F	Picture ———	1	
			Device Name			Service Engine Status	Stop	Connect	Disconnect		
			Device IP			Enable					
			Device Type	TMRobot	~]					
			Category Name	Default		Set Category		Browse	Browse		
			Description					Clear	Clear		
]	
			TMRobot							1	
			Service Engine			~					
			IPC Connect Po	ort 5	430						
			Modbus Port	5	02						
			🖌 Use Modbus	TCP			-				
			Nonsy	nchronous	keep the process run	ning without checking					
			O Synch	ronous: occ	cupy the process until	data received					
			Modb	us Retry co	ount	1					
			Modb	us Retry in	iterval	2000 ms					
			Modb	us Timeou	ut	10000 ms					

Step 6: Service Engine Select its own Service Engine Step 7: Save

3.5.3.2 Tags

3.5.3.2.1 Tags Basic Introduction



Refresh Tree Diagram

3.5.3.2.2 Tags Function Bar Introduction



+	<new> Add New Tag</new>
Name	<rename> Rename Tag</rename>
Ε,	<add category=""> Add Category</add>
Ξ	<expand all=""> Expand all Lists</expand>
•	<fold all=""> Fold all Lists</fold>
$\langle \rangle$	<refresh> Refresh Lists</refresh>
	<copyitem> Copy Tag Item</copyitem>
$\overline{\mathbf{Y}}$	<filter> Filter</filter>
	<delete> Delete</delete>

Add Category

Add Category		
Category Name	Root Node	
Parent Category		
	Cancel OK	

- Category Name: Add New Category Name
- Select Parent, Add in the Parent Group

• Root Node: Checked represents this Category is the List Category of the highest layer, without Parent

➢ Filter		
Тад Туре	•	Select
Value Type	•	Cancel
Keyword		Find

- Tag Type: Can use Tag Type to Find
- Value Type: Can use Value Type to Find
- S Keyword: Can input Tag Name Keyword Like to Find
- Select: Select
- Cancel: Cancel Find

3.5.3.2.3 Tag Environment Introduction

1. Tag Form



2. Tag_Form

Tag Settings -					ashboard Pic	ture
, Tag Name					Within Range	Out of Range
-Service Engine	TMMService1	Capture Mode	Service Loop Time	~		
Tag Type	Get Modbus	✓ Data Type	int	~		
Category Name		Set	Keep Last Value		Browse	Browse
-Description					Clear	Clear
Save Mode					0 < Va	lue < 0
Get Modbus -						
Get Modb	us	Filter	Convert	Calculation	n and Statistics	Action After Value Chang
Device Name						
Function Code			~			
Start Address			Format Decimal			
Length						
Convert Type			~			

• Tag Name: Input Device Name, Names are case-insensitive, so upper and lower case spellings will be regarded as the same name. Special characters and blanks cannot be used in names.

• Service Engine: Define the Service Engine to which the Tag belongs

Get Modbus	Capture devices that support Modbus, such as PLC, Robot
	temperature, current etc.
Service Performance	Capture Service Engine performance
Device Error Code	Capture Device Error Code (Only applicable to the Device of
	TM Robot/Service Engine)
SQL Variable	Capture the Database Table value
Custom Variable	Custom Variables, when select Custom Variables, Capture
	Mode can only be Call Event.
Device Status	Capture device status, such as: whether the Robot is
	Running, whether the Service Engine has error
Robot Variable	Capture the variables on the Robot, understand the operation
	progress, calculate the production information etc. (Capture
	requires the Robot working with VarSync command)
Is File Exist	Confirm whether the file exists by Capture, in order to
	upload/download (Service Engine only)
Robot Groups Status	Capture the status of the Robot Groups, such as: whether the
	Robot Group is Running

• Tag Type:

Get Socket	Receive signals from devices that support Socket

• Capture Mode: Mode of Capture Data and Timing, Service Engine Loop Time, Monitor Schedule, Call Event

Service Engine Loop	Captures the Tag value regularly according to the polling time	
Time	defined by the Service Engine	
Monitor Schedule	Captures the Tag value actively only when it is judged	
	according to the Monitor conditions	
Call Event	Captures the Tag value actively only when the Tag name	
	appears during execution of Action, Script Language or	
	setting	

• Data Type:

int	Integer
float	Single-precision
	floating-point number
double	Double-precision
	floating-point number
string	string
byte	bit
bool	Boolean true/false
datatable	datatable
datetime	datetime

- **6** Category Name: Click "Set Category", add to new category
- Description: Cursor stays at the description of the List display Tag

Save Mode: Define Tag value whether to save data:

- Do not save: data that is only meaningful at the moment, for example: device status
- Save:

Last Data	Only keep the latest data, for example: Cumulative
	production volume, Pallet
Data changed	Save when the data changes, for example: record the
	production time of each product
Capture All	Save all when triggering Capture, for example: for analyzing
	data

• DashBoard Picture: Set Tag range Picture

Range: Set the Tag range, display the In Range Picture within the range, and display the Out Of Range Picture outside the range

3. Filter

Set the Filter conditions here to filter the Tag value

(Only when the Tag value changes, the set Filter will be functioning!)

Get Modbus		Filter	Convert	Calculation and Statistics	Value-Changed Action
Blank Data					
2 Content	O Value	O Length	Value		
Non Numeric					
Monitor after filterin	g				

• Filter out blank data

October Content

Filter range value or range length data

value	Filter out the specified range value	
	data	
Length	Filter out the length data of the	
	specified range	
~	>, <, >=, <=, ==, !=	
Value	Condition value to be filtered	

• Filter non-numerical data

Set the Monitor to be Started after filtering

4. Convert

Converts the received Tag value (the Value Type before and after conversion must be the same) (Only when the Tag value changes, will the set Convert be functioning!)

0	Get Modbus	Filter	Convert	Calculation and Statistics	Value-Changed Action
	+ 🖸 🛍				
	Value	2	Convert to		9
	Match Condition	Raw Data	Converted Data	1	
9					-
					-
					-

• Add, Update, Delete

Menu

>	More than	
۷	Less than	
>=	More than or	
	Equal to	
<=	Less than or	
	Equal to	
==	Equal to	
!=	Not Equal to	

- Value before Converting
- Value after Converting
- After input the Convert Value, click "Add" to add here
- 5. Calculation and Statistics

Calculation and Statistics

(Only when the Tag value changes, will the set Calculation and Statistics be functioning!)

Custom Varia	able	Filter	Convert	Calculation and Statistics	Value-Changed Action
Calculation Formula	test + 1				
Statistics Type	Maximum	~	Mode F	requency 100 times eriod ay	
Execution Order	Calculation First O Other Thing Thing	~			

• Calculation Formula: Click the menu on the right to select Tag, and input the Calculation Formula (Add spaces before and after Operation Symbols)

Statistics Formula: Statistics Formula

Maximum	Maximum
	maximum

Minimum	Minimum
Mean	Mean
Sum	Sum
Media	Media
SD	Standard Deviation
СРК	Process Capability Index

Mode

Set Statistics Mode

Frequency (Frequency)	Frequency 100 times
	When the Tag value changes several times,
	Statistics are made only once.
Period	Period 1 Minute
(Period)	Set time (Minutes), Statistics are made only
	once.
Day	Makes Statistics once a day.
(Day)	

• Execution Order: If there are both Calculation Formula and Statistics Formula at the same time, you can choose to execute the Calculation Formula first, or execute the Statistics Formula first

Save Mode: Calculation or Statistics Results can be saved in Other Tag or original
 Tag

Note: Mean, Media, SD, CPK can only be saved to Other Tag

6. Value-Changed Action

Custom Variable	Filter	Convert	Calculation and Statistics	Value-Changed Action
Action Execution Timing	Value Change	O Capture Value	Out of Range	
¥ + ⊠ × m ⊂ ⊂ ⊂ → Enable ActionType_	New Action Temp C Action Type	Diate	✓ ✓ Enable	

Refer to Action File.

II. Tag Type

3.5.3.2.4 Get Modbus

- Tag Setting —]	⊢Dashboard Pict	ture —	7
Tag Name						Within Range	Out of Range	
Service Engine	TMMService1	✓ Captu	re Mode	Service Loop Time	~			
Тад Туре	Get Modbus	✓ Value	Туре	int	~			
Category Name		Set (Category			Browse	Browse	
Description			U			Range	Clear	
Save Mode	Last D	ata 🔿 Data Cł	nanged	O Capture All		0 < Val	ue < 0	
Get Modbus —								
Get Modbu	us	Filter		Convert	Calcula	ation and Statistics	Action After Valu	e Change
Device Name								
Function Code			\sim					
Start Address				Format: Decimal				
Length			_					
Convert Type			~					
	Ĺ							
				Modbus tab	le			

• Value Type: int, float, double, string, bool

Device Name: Selects the device to be read (with Modbus support), such as:
 TM_Robot

B Refers to Modbus protocol to read data

Example: To read TMRobot Speed

Function Code: Refers to Modbus table "FC", (01~04 are R/Read, 05~06 are W/Write) Start Address: Refers to Modbus table "Address" Length: One Address is 1, two Addresses are 2

Convert Type: Refers to Modbus table "Type"

Robot Stick	FC.	Address ₁₀	Address ₁₆	Туре∘	R/W	Note∞
Project Running Speed.	04.0	7101.	1BBD.₀	Int16₀	R₽	%₀
M/A Mode _e	04.0	7102.	1BBE₽	Int16₽	R₽	A:1; M:2,
Play/Pause _*	05₽	7104	1BC0+	Bool₽	W٥	Triggered as 1 reserved
Stope	05.	7105.	1BC1~	Bool₽	W٥	inggered as 1 received.»
Stick+.	05.	7106	1BC2 _e	Bool₽	W٩	Triggered as 1 received.
Stick-₀	0 5¢	7107.	1BC3@	Bool₽	W٥	(Manual Mode only).

3.5.3.2.5 Service Performance

Tag Setting —		Dashboard Picture
Tag Name		Within Range Out of Range
Service Engine	TMMService1 Capture Mode Service Loop Time	
Тад Туре	Service Performance 🗸 Value Type 🕕 float 🗸	
Category Name	Set Category	Browse Browse
Description		Range
✓ Save Mode	Last Data O Data Changed O Capture All	0 < Value < 0
Service Performa	ince	
Service Perform	ance Filter Convert Calcul	lation and Statistics Action After Value Change
Device Name		
Туре		

Value Type: int, float, double 0

0 Device Name: Selects the Device Name which the performance to be known.

€ Type:

CPU	CPU Utilization Rate, the capture time is the average value of
	30 times based on the Service Engine Loop Time. If the
	Service Engine Loop Time is set to 1 second, then capture the
	average value of the previous 30 seconds.
RAM	Memory Utilization Rate
HDSize	Hard Disk Size
HDIOTime	Hard Disk Input / Output Time
RMSLoopInterval	Service Engine Running Interval
MonitorLoopInteral	Monitor Monitoring Interval
TCPReceiveInterval	Packet Receive Interval

3.5.3.2.6 Device Error Code

Tag Setting —					CDashboard Pict	ture
Tag Name					Within Range	Out of Range
Service Engine	TMMService1	✓ Capture Mo	de Service Loop Time	\sim		
Tag Type	Device Error Code	∼ /alue Type	1 int	\sim		
Category Name		Set Categ	ory		Browse	Browse
Description					Range	Clear
✓ Save Mode	Last Data	O Data Change	d O Capture All		0 < Val	ue < 0
-Device Error Cod	de				•	
Device Error C	ode	Filter	Convert	Calcula	tion and Statistics	Action After Value Cha
Device Name						

- Value Type: int
- **2** Device Name: Selects the desired Device Name

3.5.3.2.7 SQL_Variable

Tag Setting —							┌ Dashboard Pict	ure —	1
Tag Name							Within Range	Out of Range	
Service Engine	TMMSen	vice1	~ (Capture Mode	Service Loop Time	\sim			
Тад Туре	SQL Vari	able	~ \	/alue Type 🛛 🕕	int	~			
Category Name				Set Category			Browse	Browse	
Description							Range	Clear	
Save Mode	۲	Last Data	OD	ata Changed	O Capture All		0 < Val	ue < 0	
- SQL Variable —							L		
SQL Variab	le		Filter		Convert	Calcula	ation and Statistics	Action After Value	e Change
2 SQL Connection	String								
3 SQL Language	Γ						^		
	L						V Nimer		
							SUL WIZard		

- Value Type: int, float, double, string, byte, bool, datatable, datetime
- SQL Connection String: Selects the location of the data source DB to be captured
- SQL Language: Inputs SQL syntax, can bring in by using "SQL Wizard"
- Step 1: Click "SQL Wizard"

Step 2: Select DB

Step 3: Select Table Name and Column

Databasa Nama	Г	Kev	Column	Table	DataBase		
TMMainDB			VolueTure	Things	TMMainDB		
Ishie Name			7 aloc Type	11migs	THHOLED		
Things	~						
Column	_						
□ ThingName							
RMSName							
□ ThingType							
Value Type							
Is_HistoryData							
Capture Type							
DeviceName							
ThingGroupName							
Modbus_FunCode							
☐ Modbus_StartAddr							
🗌 Modbus_Len							
☐ Modbus_ConvertType							
☐ Modbus_Value							
RMS_Capture Type							
SourceDefine							
SQL_ConnectType							
SQL_ConnectString	~						
Col	umn					Clear	Next

Step 4: Click Column

Step 5: Table Name, Column, DB Display

Step 6: Click Next



Step 7: The first 200 records of data are brought out by default, which can be modified Step 8: Set the data condition to be filtered Step 9: Next

🖳 S	SQLWizardForm				×
	Data Source	Data Filter	Advanced processing	SQL Command	
-	ing hay MAR 200				
Г	Value Type				^
•	2				
	2				
	3				
	4				
	4				
	4				
	1				
	2				
	4	_			
	4	_			
	1	-			
					¥
Fi	liter : (do not need to p	pre-position where)			
					0
					Next

Step 10: Select the Column, whether to display in Groups, whether to calculate Average,

Sum, Count, Maximum, Minimum

Step 11: Click Calculate

Step 12: The Data Results are shown below

Step 13: Next

💀 SQLWizardForm				×
Data Source	Data Filter	Advanced processing	SQL Command	
Select C		Group By	SQL Function (AVG) (MAX) (SUM) (MIN) (COUNT) (COUNT)	calculate
Value Type Value				
Original				Next

Step 14: Data Result

Step 15: SQL Command

Step 16: Whether to add external components

If yes, enter the condition and after clicking "Add", SQL Command will add the condition Step17: OK

💀 SQLWiz	ardForm					×
Data Sc	ource	Data Filter	Advanced processing	; SQL Command		
Data Result SQL Command	▶ 2 2 3 4 4 4 4 1 1 \$elect top /*Test_ve	Value Type	Type] from [TMMsinD	B].[dbo].[Things] where	• [Things].[ValueType	• •]= •
Whether Yes [Things]	to mput exte	nnai components Name)]	(Operator) = V	(Component N Test_vaniable	lame) V	Add

3.5.3.2.8 Custom Variable

Tag Setting —					Dashboard Picture
Tag Name					Within Range Out of Range
Service Engine	TMMService1	✓ Capture Mode	e Call Event	\sim	
Тад Туре	Custom Variable	√ \alue Type	1 int	~	
Category Name		Set Categor	У		Browse Browse
Description					Range
✓ Save Mode	Last Data	O Data Changed	O Capture All		0 < Value < 0
-Custom Variable)				
Custom Varia	ble	Filter	Convert	Calcula	tion and Statistics Action After Value Change
Default Value	2]		

- Value Type: int, float, double, string, byte, bool, datetime
- Default Value: Fills in the Default Value by Type

3.5.3.2.9 Device Status

Tag Setting —		Dashboard Picture
Tag Name		Within Range Out of Range
Service Engine	TMMService1 V Capture Mode Call Event V	
Tag Type	Device Status Value Type Dool ~	
Category Name	Set Category	Browse Browse
Description		Range
Save Mode	Last Data	0 < Value < 0
-Device Status -		
Device State	us Filter Convert Calc	ulation and Statistics Action After Value Change
Device Name	9	
Status	B	

- Value Type: bool
- Device Name: Selects the desired Device Name
- Status:

Running: Represents that the project is being executed

Connect: Represents that the Robot is connected to the Service Engine

Error: Robot report error

Pause: Project Pause

IsModeAuto: Whether the Robot is in Auto (Blue Light) Mode

IsModbusConnect: Whether the Modbus is Connected

	Robot Device Status	Service Engine Status	Modbus	Socket
Running	V			

Connect	v	V	v	v
Error	v			
Pause	V			
IsModeAuto	V			
IsModbusConnect	V	V		

3.5.3.2.10 Robot Variable

Tag Setting —] [Dashboard Pict	ure	
Tag Name				Within Range	Out of Range	
Service Engine	TMMService1	Service Loop Time	~			
Тад Туре	Robot Variable \sim Value Type	int	~			
Category Name	Set Category]		Browse	Browse	
Description				Range	Clear	
Save Mode	Last Data O Data Changed	O Capture All		0 < Val	ue < 0	
-Robot Variable			_			
Robot Varia	ble Filter	Convert	Calculatio	on and Statistics	Action After Value	Change
Device Name						
Variable	Global Variable O Variable					
Variable Name				~		
Default Value						

- Value Type: int, float, double, string, byte, bool
- **2** Device Name: Selects the desired Device Name
- Variable Type: Global Variable, Variable
- Variable Name: Variable Name (must be connected with the Robot first)
- Default Value: Default Value

If Variable Type is selected with Variable

Variable Type	O Global Variable	Variable	0
Project Name			✓ ☐ Limited to the Project
2 Variable Name			~
B Default Value			

- Project Name: Robot Project Name
- Variable Name: Variable Name, Format: var_xxx (must be connected with the Robot first)
- Default Value: Default Value
- Determines whether it is restricted to the project:

- ex. The Robot has two projects of A and B, and both have variables named x
- Restricted by Project: Only when project A is executed, will the Tag capture the X
 Variable
- Not Restricted by Project: Regardless project A or B is executed, , will the Tag capture the X Variable

3.5.3.2.11 Is File Exist

Tag Setting —		Dashboard Picture
Tag Name		Within Range Out of Range
Service Engine	TMMService1 Capture Mode Service Loop Time	
Тад Туре	Is File Exist Value Type Dool ~	
Category Name	Set Category	Browse Browse
Description		Range
✓ Save Mode	Last Data O Data Changed O Capture All	0 < Value < 0
-Is File Exist —		
Is File Exist	Filter Convert Calcu	lation and Statistics Action After Value Change
Device Name	2 TMMService1 .	
File Path	8	

- Value Type: bool
- Oevice Name: Selects Device Name
- File Path: File Path

3.5.3.2.12 Robot Groups Status

Tag Setting —					Dashboard Pic	ture —	1
Tag Name					Within Range	Out of Range	
Service Engine	TMMService1	✓ Capture Mode	e Service Loop Time	\sim			
Tag Type	Robot Groups Status	\sim Value Type	bool	\sim			
Category Name		Set Categor	y .		Browse	Browse	
Description					Range	Clear	
🖌 Save Mode	Last Data	O Data Changed	O Capture All		0 < Va	lue < 0	
Robot Groups S	tatus —						
Robot Groups	Status	Filter	Convert	Calcul	ation and Statistics	Action After Value	e Change
Croup Name		~]				
3 - Group Status							
O Play O Pa	ause 🔿 Error 🔿 Co	onnect 🔿 Auto Mode					
-Logical Condition	on						
O AND O C	DR						

• Value Type: bool

O Group Name: Robot Group Name

Group Status: Group Status

Running	Running
Pause	Pause
Error	Error
Connect	Connect
RobotModeAuto	Auto Mode

Logical Condition: Condition

AND	All Robots in the Group conform to Status, displaying true
OR	One Robot in the Group conforms to Status, displaying true

3.5.3.2.13 Get Socket

Tag Setting —					CDashboard Pict	ture	
Tag Name					Within Range	Out of Range	
Service Engine	TMMService1	✓ Capture M	ode Service Loop Time	\sim			
Tag Type	Get Socket	✓ Value Type	string	\sim			
Category Name		Set Cate	jory		Browse	Browse	
Description					Range	Clear	
✓ Save Mode	Last Data	O Data Chang	ed O Capture All		0 < Val	ue < 0	
Get Socket —					•		
Get Socke	et	Filter	Convert	Calcula	tion and Statistics	Action After Value	Change
Device Name	2 🔘 Service Engine	O Device					
Header	8						
Footer	0						
Socket Format	Header \t Content \t	Footer \n					

- Value Type: string
- Device Name: Selects Socket Device
- Header: Packet Header
- Footer: Packet Footer

Note: The Header/Footer of the received content must match the Header/Footer specified by the Tag, then the Tag will capture the Content.

3.5.3.2.14 Add New Tag

	Designer Devices	Monitors Actions	Groups	SFC Management				
•		🖡 🗙 🛛 Tag Form 🗙						
9	+▫।฿,ਞ▸- ฿ฅ♥	<u></u>						
	E grault	Tag Setting —				<u>ا</u> ا	Dashboard Pictu	ıre — ,
		Tag Name					Within Range	Out of Range
	copy_T3	Service Engine	TMMService1	✓ Capture Mode	Service Loop Time \sim			
		Tag Type		√ Value Type	~			
		Category Name		Set Category			Browse	Browse
		Description					Banga	Clear
	T3	Save Mode					0 < Valu	e < 0

Step 1: Click Tag Step 2: Click " + ", Tag_Form tab appears Step 3: Pin the Tag window

Designer Devices Tags Mo	onitors Actions	Groups	SFC Management				
Tags म ×	Tag Form 🗙						
+ 📼 🖶 🇉 – 🎧 🖷 🍸 🗋	5						
Default	<mark>┌</mark> Tag Setting —					Dashboard Pictu	ire
copy_T1	Tag Name					Within Range	Out of Range
copy_T3	Service Engine	TMMService1	∼ Capture Mode	Service Loop Time	2		
copy_T314	Tag Type		✓ Value Type				
— = copy_134 — = T — T	Category Name		Set Category			Browse	Browse
	Description					Range	
T3 T31 T31 T314	Save Mode					0 < Valu	e < 0

Step 4: Input Tag name, Service Engine, Capture Mode

Step 5: Input Tag Type, Value Type

• Tag allows Value comparison table

Тад Туре	Value Type
Get Modbus	Int, float, double, string, bool
Service Performance	Int, float, double
Device Error Code	int
SQL Variable	Int, float, double, string, byte, bool, datatable, datetime
Custom Variable	int, float, double, string, byte, bool, datetime
Device Status	bool
Robot Variable	int, float, double, string, byte, bool
Is File Exist	bool
Robot Groups Status	bool
Get Socket	string

Designer Devices Tags	Monitors Actions Groups SFC Management	
Tags # >	Tag Form X	
+ 📼 🖳 "E 🛏 🂭 🛅		
Default	Tag Setting	Dashboard Picture
copy_T1	Tag Name	Within Range Out of Range
copy_T3	Service Engine TMMService1 V Capture Mode Service Loop Time V	
copy_T314	Tag Type Value Type v	
	Category Name Set Category	Browse Browse
	Description	Range
	Save Mode	0 < Value < 0

Step 6: Click "Set Category" to select the new Group to be Added Step 7: Decide whether to Save

Designer	, Devices	Tags	Monitors Action	s Groups	SFC	Management				
Tags		ą	× Tag Form ×							
	B, 18 🕻) 📲 🍸 🗎	5					(3	
🔳 🙀 Defa	ault		┌ Tag Setting -					_	Dachbeard Piet	
	opy_T1 opy_T2		Tag Name						Within Range	Out of Range
	opy_T3 opy_T31		Service Engine	TMMService1		✓ Capture Mode	Service Loop Time	~		
	opy_T314		Tag Type			\sim Value Type		~		
Ţ	· ·		Category Name			Set Category		- 1	Clear	Clear
T.	2		Description						Range	orodi
	3 31		Save Mode						0 < Valu	ie < 0
	314									

Step 8: Check whether to set Tag Range and Icon

Tag Form 🗙							
Tag Setting —					Dashboard Pic	ture —	
Tag Name					Within Range	Out of Range	
Service Engine	TMMService1	✓ Capture M	ode Service Loop T	ïme ∨			
Tag Type	Get Modbus	√ Value Type	int	~			
Category Name		Set Cate	gory		Browse	Browse	
Description					Clear	Clear	
Save Mode					0 < Va	ilue < 0	
Get Modbus -							
Get Mode	ous	Filter	Convert	Calcul	ation and Statistics	Action After Value	Change
Device Name							
Function Code			\sim				
Start Address			Format: Decima	al			
Length							
Convert Type			\sim				

Step 9: Different Tag Types have different tab content. Refer to 3. Tag Type to set Step 10: Save

3.5.3.2.15 Edit Tag

	Tags # ×	Tag Form X	
	+ 📼 🖳 🇉 – 💭 🖳 🍸 🛅 🦲	6	
0	T3 T3 T3 T3 T34 T34 T34 T34 T34	Tag Setting — Tag Name Service Engine Tag Type Category Name Description Save Mode	CurrentDeviceModel Within Range Out of Range TMMService1 Capture Mode Service Loop Time Within Range Out of Range Custom Variable Value Type string Service Loop Time Browse Browse Common Set Category Clear Clear Clear Range 0 Value < 0 Common Set Category
	HourSpent	-Custom Variable	9
	InspectionResult	Custom Varia	ble Filter Convert Calculation and Statistics Action After Value Change
	LastLabel 	Default Value	

Step 1: In the Tag Form, double-click the Tag to be edited, and Tag_Form appears on the right

Step 2: Edit on the Tag_Form tab

Step 3: Save

3.5.3.3 Action

3.5.3.3.1 Action Basic Introduction

III. Through Action settings, users can produce in the templates of data in advance for

post-processing event

Action Form screen

View(V) Tool(T) Help(H) Options(O)	
Devices Designer Tags Monitors Actions Groups SFC	play
Actions # × Actions Form X Service Engine - Management Device Form	

3.5.3.3.2 Action Explore

1. Tree Diagram



2. Icon Function (From Left to Right)



•	<fold all=""> Fold Tree Diagram</fold>
()	<refresh> Refresh</refresh>
	<copyitem> Copy</copyitem>
$\overline{\mathbf{Y}}$	<filter> Filter Query</filter>
1	<delete> Delete</delete>

3.5.3.3.3 Function

3. Several Strain St

4. Rename> Rename Action

<u>CAUTION:</u> Before operation, the Device function or Service Engine Tray needs to be used to stop the Service Engine



5. Add Group>

Group Name: Fill in the new group name

Root Node: If the new group is the first layer, it is necessary to check for selection Parent Name: If the new group is a subgroup, it is necessary to select the Parent group name.

Add Group					
Group Name	Root Node				
Parent Name					
	Canc OK				





The display results are as follows, Find displays the number of results



iii. Cancel: Turn off Filter

11. 直 <Delete>

Click any Action item and click Delete. A pop-up window will display the Action and association with other settings. If no important association is confirmed, click Next, then confirm the item to be deleted and click OK.





	Action Sett	ing			
0	Action Name	test3	Service Engine	ServiceEngine1	\sim
₿	Action Type	Set Modbus 🗸	Category Name	Group2	Set Category
6	Description	test			

• Action Name: Input Action Name, Names are case-insensitive, so upper and lower case spellings will be regarded as the same name. Special characters and blanks cannot be used in names.

e Service Engine: Service name belongs to

Action Type:

Send Email	If there are any conditions on the production line, the
	Action function can be used to automatically send
	Email notifications
Send SMS	One-way/two-way SMS Notification
Set Modbus	Write in the Device that supports Modbus, such as:
	Let the Robot play, stop etc.
Enable Monitor	Enable/Disable Monitor
Pause Monitor	Pause MonitorSeconds
Set SQL	Change database data
Set Tag Value	Change Tag value
Call Dashboard Function	Call Dashboard Function
Add/Remove Group Member	Add/Remove Group Member
Set Monitor Property	Set Monitor Property
Enable Service	Enable/Disable Service
Change Robot Initial Project	Change Robot Initial Project
Upload/Download File	Upload/Download File
Upload/Download Directory	Upload/Download Directory
Set Socket	Send Socket Signal
Upload/Download Robot	Upload/Download Robot Project
Project	
Start Ticker	Start Ticker (Start Counting Time)
Stop Ticker	Stop Ticker (Stop Counting Time)
Call Script Language Function	Call Function

- Category Name: Click "Set Category", Add to New Group
- Description: Cursor displays Action Description

3.5.3.3.5 Action Type

1. Send Email

-Action Setti	ng					
Action Name	-		Service Engine	ServiceEngine1	~	
Action Type	Send Email	~	Category Name	Group1	Set Categor	y
Description						
- Send Email						
Email To					<	
~						
Subject						
	[Â	
B Email Body						
Linal Body					~	

- Email To: Recipient, multiple Recipients (separate with ;)
- Subject: Subject
- Email Body: Email Body, If to send the Tag value, input /* Tag Name */

3.5.3.3.6 Send SMS

-Action Sett	ina ———				
Action Name			Service Engine	ServiceEngine1	~
Action Type	Send SMS	~	Category Name	Group1	Set Category
Description					
- Send SMS					
One Way	Two Way (bi-dir	ectional)			
Tel No					
e Message				< >	
	L				

Tel No: Mobile phone number (for receiving SMS)

0

- Message: Text message content
- If select Two Way

	-Send SM	IS ——				
	One W	ay Two Way				
0	: + E	3 🛅				
	2 Tel No					
		3 Command	Action	Name	5 Description	
		Command	Action	Description		
	G)				

- Add, Edit, Delete
- Mobile phone number (for receiving SMS)
- Number
- Action Name
- G Content
- After input the Convert Value, click "Add" to add here

3.5.3.3.7 Set Modbus

	Action Setting ———		
	Action Name	Service Engine Ser	viceEngine1 ~
	Action Type Set Modbus	✓ Category Name Grou	up1 Set Category
	Description		
n	Set Modbus		
	Device Name		
	O Group Name		
	P unction Code	· · · · · · · · · · · · · · · · · · ·	-
	Btart Address		
	€ onvert Type	· · · · · · · · · · · · · · · · · · ·	~
	Salue		

• Select the device or group to write to (Modbus is supported), such as: TM_Robot

Device Name	Equipment name
Group Name	Group Name

• Function Code: Refer to Modbus Table

• Start Address: Start address (range: 0~65535), TM Robot use range is 0~8999, for customized address, addresses after 9000 can be used

• Convert Type: Because the Modbus storage data type is hexadecimal, so the original value type must be changed to byte

• Value: Input the Value to be written

3.5.3.3.8 Enable Monitor

Action Setting	I				
Action Name			Service Engine	ServiceEngine1	~
Action Type En	able Monitor	~	Category Name	Group1	Set Category
Description					
-Enable Monito	or				
Monitor Name]		
Choice	Stop	Start			

- Select Monitor
- **2** Select Stop or Start

3.5.3.3.9 Pause Monitor

Action Setting		
Action Name	Service Engine ServiceEngine1	~
Action Type Pause Monitor	✓ Category Name Group1	Set Category
Description		
-Pause Monitor		
Monitor Name		
Pause	Second	

- Select Monitor
- Input the number of seconds for Pause

3.5.3.3.10 Set SQL

h	-Action Set	ting ——						
	Action Name			Service Engine	ServiceEngine1	\sim		
	Action Type	Set SQL	~	Category Name	Group1	Set Category		
	Description							
ľ	- Set SQL —							
	SQL Conne	ction String						
							^	
	SQL Langua	age						
							.,	,
						SQL Wizard		

• SQL Connection String: Select the data DB to be changed

The displayed DB lists are all LogDB. Because MainDB is a setting file, it is not allowed for users to modify.

SQL Language: Enter SQL syntax, "SQL Wizard" can be used to add syntax

Step 1: Click "SQL Wizard"

Step 2: Select Database

Step 3: Select Table

🛃 SQL Wizard		_	×
Select Database Select Table	2 v 3 v		

Step 4: Select Insert (Insert Data) or Update (Update Data)/Delete(Delete Data)

🖳 SQL Wizard					_	×
Select Database	TMMainDB	~	SQL Mode	0.0.0.0		
Select Table	Robot_Error_Code	\sim		O Opdate / Delete		

Step 5: If Select Insert:

Input the data to be inserted in the Column. Click "Generate SQL Command" Step 6: Syntax appears here,. Click "OK"

SOI Wizard					_		×
Select Database	MMainDB		✓ SQL Mode ● Insert	e 🔿 Update / De	lete		~
Select Table R	obot_Error_Code		~				
Error_Group	Error_Dec	Error_Hex	Description_EN	Description_TW	Descrij		
						0	
<					>	Generate SQL Command	
G							
						ок	

If SQL Mode Selects Update/Delete



🖳 SQL Wizard					-		×
Select Database	TMMLogDB	~	SQL Mode				
Select Table	Things_History	~	O Insert	Update / Delete			
Update / Delete							
Batch field update	Custom data update						
SQL Command Ar	ea						
6							
Batch field update	e settings						
Column N	lame	Operator	Value				
	~	~					
Condition setting	(Where)						
Column N	lame	Operator	Value				
	~	· ·			Set		
0							
Whether to inpu	t external components	Select extra	a Control type —				
🔾 Yes 🖲 N	0	Tag Nam	ie 🔿 Das	hboard Controller			
Colur	nn Name	Operator	Colu	mn Name			
	\checkmark	\sim			~	Add	
					6		
						ОК	

- Batch field update (Update Data)
- **e** Batch field update settings:

Column Name	The data column to be updated
Operator	=
Value	Updated value

• Condition setting (where):

Column Name	Conditional data column
Operator	=, !=, >, <, >=, <=
Value	Conditional Value

After setting is complete, click set

- Whether to add other conditions
- SQL Syntax displays here
- Click OK

• Custom data update

🖳 SQL Wizard			– 🗆 X
Select Database Select Table	TMMDemoDB RobotStatusLog	SQL Mode Insert Update / Delete	
	Number of the second states (
Batch field update	Justom data update		
Delete	DeviceNo Temperature	Voltage Current	LogTime ^
	TM01 38.598317655278912	2 48.686662286609511 0.57283205425509165	7/22/2020 3
	TM02 38.50383102149857	50.898982016906416 0.66094821634118672	7/22/2020 3
	TM01 38.645123679079511	49.582921714825886 0.51909405091666927	7/22/2020 3
	TM02 38.550637045299169	49.795241471635016 0.60721021300276434	7/22/2020 3
	TM02 38.580151671772157	50.79351970412749 0.66249643902205235	7/22/2020 3
	TM01 38.6746383055525	48.581199973830586 0.57438027693595739	7/22/2020 3
<	TM01 38 714885523470208	49 033397576067394 0 58613422235551771	7/22/2020 3 *
Top 1000			
Where	Column Name	Operator Value	
8	~	~	Add
			Clear
			Filter
whether to incu	t external components	elect extra Control type	
🕒 🔿 Yes 💿 N	0	Tag Name O Dashboard Controller	
Column	Name Operato	r Control Name	Add
			ок
			6

- 0
- Custom data update (Delete Data)

2 Display the first 1000 sets of data, check Delete

• Set conditions, columns, expressions, and condition values. After clicking Add, the SQL syntax displays below

- Whether to add other conditions
- OK

2. Set Tag Value

-Action Set	ting —			
Action Name	test3	Service Engine	ServiceEngine1	~
Action Type	Set Tag Value v	Category Name	Group2	Set Category
Description	test			
Set Tag Va	lue			
Tag Name				
Operator				
Value	6			

• Tag Name: Select Tag Name

0	Expression
=	Equal to
+=	Cumulative Addition
-=	Cumulative Subtraction
*=	Cumulative Multiplication
/=	Cumulative Division
%=	Cumulative Remainder

Select Tag or enter custom value or enter calculation formula

3.5.3.3.11 Call Dashboard Function

	ing		
ction Name		Service Engine ServiceEngine1	\sim
ction Type	Call Dashboard Function	✓ Category Name Group1	Set Category
)escription			
all Dashb	oard Function		
Project Nam	oard Function		~
- Project Name: Select Project Name
- Function Name: Select Function Name

3.5.3.3.12 Add/Remove Group Member

Action Setting				
Action Name test3		Service Engine	ServiceEngine1	~
Action Type Add/Remov	ve Group Member 🧹 🗸	Category Name	Group2	Set Category
Description test				
Add/Remove Group M	/lember			
Group Name				
Choice On Add	O Add All	C) Remove	O Remove All
B Device Name				
(This operation is	only in memory, the da	tabase will r	not change.)	

Group Name: Select Group

Choice:

Add	Add Group Robot
Add All	Add all Robots controlled by Service to the Group
Remove	Remove Group Robot
Remove All	Remove all Robots out of Group

Note: The action taken by Action to the Group will not affect the storage value of the Group. It only changes the Group of memory. After the Service Engine is restarted, the Group will restore its original settings.

• Device Name: Select to Add/Remove the Robot member, only Add and Remove can be selected

3.5.3.3.13 Set Monitor Property

tion Name	test3	Service Eng	ine ServiceEngine1	~
tion Type	Set Monitor Property	✓ Category Na	ame Group2	Set Category
escription	test			
et Monito	r Property			
	0			
Monitor Nar	ne		B	

Monitor Name: Select Monitor

• Property:

Cycle Time	The Cycle Time in Monitor_Form sets how often to monitor
Pass Times	The Pass over Times in Monitor_Form sets the times of condition is met.
Fail Times	The Fail Times in Monitor_Form is the times of condition is not met
Start Time	The Schedule Time in Meniter, Form gets start and and of monitor time
End Time	

Set the value of Property

3.5.3.3.14 Enable Service

Action Sett	ing ———						
Action Name	test3			Service Engine	ServiceEngine1		\sim
Action Type	Enable Service		~	Category Name	Group2	Set (Category
Description	test						
Enable Ser	/ice						
Device Nam	e ServiceEngine	e1]			
Choice	Stop	Start					

- Device Name: Select Service
- Choice:

Stop	Stop Service
Start	Start Service

3.5.3.3.15 Change Robot Initial Project

Action Sett	ing				
Action Name	test3		Service Engine	ServiceEngine1	\sim
Action Type	Change Robot Initial Project	~	Category Name	Group2	Set Category
Description	test				
Change Ro	bot Initial Project ———				
Device Nam]		
Project Nam	ne		`	-	

Device Name: Select Robot

Project Name: Select Project

3.5.3.3.16 Upload/Download File

Action Set	ing	
Action Name	test3	Service Engine ServiceEngine1
Action Type	Upload/Download File	Category Name Group2 Set Category
Description	test	
Upload/Do	wnload File	
Choice	Upload	O Download
Source	File Path	
Destination	2 Device Name	
	Directory Path	
1		

• Choice: Select Upload or Download File

Source: File Source, File Location Path

Destination: Upload Destination, Device/Directory

If Choice selects Download

-Upload/Dow	Upload/Download File							
Choice	O Upload	Download						
Source	Device Name							
	File Path							
File Path	2 Directory Path							

• Source: File Source, Device and Path where the File is located

• File Path: Download Destination: Directory

Note: The file is only 128mb at most

3.5.3.3.17 Upload/Download Directory

Action Set	ting —		
Action Name	test3	Service Engine ServiceEngi	ine1 ~
Action Type	Upload/Download Directory	✓ Category Name Group2	Set Category
Description	test		
Upload/Do	wnload Directory —		
Choice	Upload	O Download	
Source	Directory Path		
File Path	B Device Name		
	Directory Path		
• CI	hoice: Select Uplo	ad or Download Directo	ory
e So	ource: Directory Se	ource, the Path of the D	Directory
6 D	estination: Upload	Destination, Device/Dir	rectory
➢ If C	hoice selects Dow	nload	
-Upload/Do	wnload Directory —		
Choice	O Upload	Download	

Choice	O Upload	Download
Source	Device Name	
	Directory Path	bt_15ToPath
Destination	Directory Path	bt_15FromPath

• Source: Directory Source: the Device and Path of the Directory

File Path: Download Destination, Directory

Note: The file is only 128mb at most

3.5.3.3.18 Set Socket

Action Setti	ng			
Action Name	test3	Service Engine	ServiceEngine1	\sim
Action Type	Set Socket	✓ Category Name	Group2	Set Category
Description	test			
Set Socket				
Device Name	 Service Engine 			
e	O To Device			
Header				
Content				
Footer				
Socket Forma	at Header \t Content \t Footer \n			

• Device Name: Select Socket Device (Service Engine or To Device), the port number will follow the port number set by the Socket Device

- Header: Header
- Content: Signal Content Source (Tag or Custom Value)
- Footer: Footer
- Message Format: Header\tContent\tFooter\n

3.5.3.3.19 Upload/Download Robot Project

Action Setting	
Action Name test3	Service Engine ServiceEngine1 ~
Action Type Upload/Download Robot Project	✓ Category Name Group2 Set Category
Description test	
Upload/Download Robot Project	
Choice Upload O Do	wnload
Source File Path	
Destination Device Name	

- Choice: Select Upload/Download Robot Project
- Source: Project File Source: the Path of the Project File
- Destination: Upload Destination, Device Name (Robot)

If Choice selects Download

- Upload/Down	load Robot Project	
Choice	O Upload	Download
Source	Device Name	
	Robot Project	~ ·
Destination	Directory Path	

• Source: Project File Source, the Device where the Project File is located, and the name of the Robot Project

Directory Path: Download Destination, Save Path

3.5.3.3.20 Start Ticker

-Action Settin	ng —				
Action Name			TMService	TMService1	\sim
Action Type	Start Ticker	~	Group Name	TMR_TM171421	Set Group
Description					
- Start Ticker					
Ticker Name					

• Ticker Name: Input Ticker Name

3.5.3.3.21 Stop Ticker

_ Act	ion Sett	ing				
Actio	on Name	test3		Service Engine	ServiceEngine1	\sim
Actio	on Type	Stop Ticker	~	Category Name	Group2	Set Category
Des	cription	test	·			
Sto	p Ticke	r ———				
Tio Tir	cker Nam	e D		~		
0	Tick	er Name: Se	elect Ticker Nam	e		
0	Tim	e Difference	: Select Tag			



	-Action Setting		
	Action Name test3	Service Engine ServiceEngine1	~
	Action Type Call Script Language Function 🗸	Category Name Group2	Set Category
	Description test		
	Call Script Language Function		
	Function Name	~	
0	Select Function Name		

3.5.3.3.23 Add New Action

View(V) Tool(T) Help(H) Options(O)	
Designer Devices Tags Monite Actions Groups	SFC Management
	Actions Form ×
+ 💷 🖳 ™ → 💭 🖫 🍸 🛅	13
Default	Action Setting
E w Root	Action Name test1 Service Engine ServiceEngine1 ~
∎	Action Type Set Tag Value Category Name Group1 Set Category
🖬 🚋 Group2	Description

Step 1: Click Action

Step 2: Click " + ", Action_Form tab appears

Step 3: Can pin the Action window

Designer Devices Tags M	nitors Actions Groups SFC Management
Actions 🛛 🛱 🗙	Actions Form ×
+ 🔤 🖳 'E +- 💭 🖳 🍸 🕅	
Default	CAction Setting
E ∰ Root ♥♥	Action Name Lest1 Service Engine ServiceEngine1 ~
test1	Action Type Set Tag Value ~ Category Name Group1 Set Category
g 👘 Group2	Description
	Set Tag Value

Step 4: Input the Name of the Action and its Service Step 5: Input Action Type

Step 6: Click "Set Category" to select the new Group to be Added

Designer Devices	Tags Monitors Action	ions Groups SFC Management
Actions + ▣, ᅚ: ♡ □	Image: specific constraints Image: specific constraints Image: specific constraints Image: specific constraints	m ×
 ⇒ Default ■ Root ■ Group1 ■ test1 ■ test2 ■ Group2 	Action Name Action Type Description	e test1 Service Engine ServiceEngine1 ✓ Set Tag Value ✓ Category Name Group1 Set Category
	Set Tag Va	alue
Designer Devices	Tags Monitors Action	ions Groups SFC Management
Actions + 📼 🖪, ᅚ) 🃿 🛙	Image: specific constraints Image: specific constraints Image: specific constraints Image: specific constraints Image: specific constraints Image: specific constraints	m ×
Default B & Root	Action Set	tting
test1	Action Type	Send Email V Category Name Group1 Set Category
n 🙀 Group2	Description	
	🔗 🔽 Send Emai	ill
	Email To	
	Subject	
	Email Body	ly

Step 7: Different Action Type has different tab content, refer to 3. Action Type for setting Step 8: Save

	Designer	Devices	Tags	Monitors	Actions	Groups	SFC	- X Manag	gement			
	Actions			4 × Actio	ons Form 🗙							
0	+ ፟፟∎.	™⊢¦C	- 7 i] <mark>⊡</mark> _Actio	n Setting -							
	Root	o1		Action	ame test1				Service Engine	ServiceEngine1		\sim
	te	st1 st2		Action	Type Send	l Email		\sim	Category Name	Group1	:	Set Category
	🚡 👾 Group	p2		Desci	iption							
				Send	Email —							
				Ema	il To						< >	
				Sub	ect]
				Ema	il Body						^]
											Ý	1

3.5.3.3.24 Edit Action

Step 1: In the Action Form, double-click the Action to be edited, the Action_Form appears

on the right Step 2: Edit in the Action_Form tab Step 3: Save

3.5.3.4 Monitor

- 3.5.3.4.1 Monitor Basic Introduction
 - **IV.** Use the Monitor Setting, the user can write logic and execute pre-programmed Action templates

3.5.3.4.2 Monitor Explore

Monit	tors	Υ
+ 📼	▣ 🖳 ᅚ⊢ 💭 🖳 🏹 🛅	
+	<new> Add Monitor</new>	
Narre	<rename> Rename Monitor</rename>	
e,	<addgroup> Create Group</addgroup>	
Ξ	<expand all=""> Expand all Lists</expand>	
•	<fold all=""> Fold all Lists</fold>	
()	<refresh> Refresh</refresh>	
	<copyitem> Copy Item</copyitem>	
$\overline{\mathbf{Y}}$	<filter> Filter</filter>	
	<delete> Delete</delete>	

Add Group

Add Group					
Group Name	□ Root Node				
Parent Name					
	Canc OK				

Group Name: Add Group Name

Select Parent, Add in the Parent Group

• Root Node: Checked represents this Group is the List Group of the highest layer, without Parent

➤ Filter



• Monitor Enable Status: Use Monitor Enable State to Filter (All, With Enable Function, Without Enable Function)

- Keyword: Can input Monitor Name Keyword Like to Find
- Select: Select
- Cancel: Cancel Find

3.5.3.4.3 Monitor Environment Introduction

7. Monitor Form

Monitors	Ψ×
┼┉│▋,ᅚ→┤∁┖┓Ÿ湔	
TMR_NCU Robot_TMR_NCU_Station_Status TMR_RMS_Project_ServiceEngine1 TMR_RMS_Project_ServiceEngine1	

8. Monitor_Form

(1) Monitor Setting

,	-Monitor Settin	na ————		
4	Monitor Name	6. 	Enable	Disable Monitor after Actions completed
¢	Schedule Time	00:00:00 🖨 ~ 00:00:00 🖨		CycleTime 1000 ms
¢	Category Name	TMR_NCU	Set Category	Pass 01 Times
þ	Service Engine	ServiceEngine1 ~		Fail 1 Times
8	Description			

Monitor Name: Input Monitor Name, Names are case-insensitive, so upper and lower case spellings will be regarded as the same name. Special characters and blanks cannot be used in names.

• Whether the Monitor is always enabled as long as the Service Engine is running, the default is "Yes"

• Whether to turn off the Monitor immediately when the Actions are finished, the default is "No"

Schedule Time

Monitoring Frequency and Times

Cycle Time	How often to monitor once (ms)			
Pass over Times	The times of condition is met, execute Action When Pass			
Pass over Times	when the times of condition is met			
Eail Timos	The times of condition is not met, execute Action When			
	Fail when the times of condition is not met			

- G Click "Set Group", add to Group
- Belonged Service
- Cursor displays Monitor Description

3.5.3.4.4 Monitor Condition

	Monitor Condition					
_	Tag Name 2	Operator 🚯	Value 🕂		Logical Condition 5	
0	+		•			-
6	Actions Upon Pass Actions Upon Fail	Actions Upon Exception]		9	
6	十 🖻 mì × 🖻 🗐 🖂 New	Action T	emplate	~	🖌 Enable	
U	Enable ActionType	Action T	iype	~		

- Add/Delete
- Page Tag Name: Select Tag Name

Operator:

	•					
>=	More than or Equal to					
<=	Less than or Equal to					
!=	Not Equal to					
==	Equal to					
>	More than					
<	Less than					
lk	(Like) Strings that conform to					
	specific data					

Value: Input Conditional Value or Select Tag Name

• Logical Condition: If there are more than two Monitor conditions, set logical conditions

or	It is true as long as one of the conditions is met.
and	It is true when all conditions are met.

6

Actions When Pass	Action List to be executed when the conditions are met.		
Actions When Fail	Action List to be executed when the conditions are not met.		
Actions When Exception	Action List to be executed when error occurs during Action		
	execution.		

Function Bar and Action List

Function Bar:

+	<add> Add Action, Change to Add Mode after clicked</add>						
	<confirm> Confirm Add Action</confirm>						
	<delete> Delete Action</delete>						
\times	<cancel> Cancel Add Action, Change to Edit Mode after clicked</cancel>						
£	<move up=""> Move up</move>						
Ę.	<move down=""> Move down</move>						
\rightarrow	<expend fold=""> Expand/Fold</expend>						
New	<add edit="" mode=""></add>						
Edit	Note: In Add Mode, only <confirm>, <cancel> can be used</cancel></confirm>						

Action List

The newly added Action List will be listed here, **9** can be used to select whether to enable.

€+ E	🗄 🛅 🔀	∈ ∈ ⊡	Edit
	Enable	ActionType	
	\checkmark	Send Email	
۱.		Set Tag Value	

- Select Action, automatically brings in Action Type and Content
- Whether to Enable this Action

	Designer	Devices	Tag:	Monitors	Actions	Groups	SFC	Managen	nent .			Service Engine S	Status ServiceEngine	1	✓ Displ	lay
	Monitors			φ×	Monito	Form ×										- ,
	┼┉│閠	™⊢ C	- 7 🗋		1											
•	TMR_NC	U .			Monitor	Setting —										^
	TMR_RM	IS_Project_Servi	ceEnginel		Monitor N	ame			 Enable 	Disabl	e Monitor afte	r Actions completed				
					Schedule	Time 00:00:	00 🚖 ~ 00:0	00:00		CycleTime	1000	ms				
					Category	Name TMR_N	1CU		Set Category	Pass	1	Times				
					Service E	ngine Servio	eEngine1	~	,	Fail	1	Times				
					Descriptio	n						_]			
					Monitor	Condition							-			
						Tag Name			Operator			Value		Logical Condition		
					+ 👕	_					•					-
					Actions I	Upon Pass	Actions Upor	n Fail	Actions Upon Ex	ception						
					}+ ₽		= 🗔 New				.					
					En	ahle Action	Type				ction remplate		~	Enable		
										A	ction type		~			
																~

Step 1: Click Monitor.

Step 2: Click " + ", Monitor_Form tab appears.

Step 3: Can pin the Monitor Window.

Designer Devices Tags I	Monitors	Actions Groups	SFC Managem	ent .		Service Engine S	tatus ServiceEngine	1	Display
Monitors	Ψ×	Monitor Form 🗙							-
+ 📼 🖶 ᅚ - 💭 🖳 🍸 🛅		5							
TMR_NCU		Monitor Setting							^
IMR_RMS_Project_ServiceEngine1		Monitor Name		✓ Enable	Disable Monitor a	after Actions completed			
		Schedule Time 00:00:00	€ ~ 00:00:00 €	c	ycleTime 1000	ms			
		Category Name TMR_NCU	J	Set Category P	ass 1	Times			
		Service Engine ServiceEn	ngine1 \sim	F	ail 1	Times			
		Description]		
		Monitor Condition —							
		Tag Name		Operator		Value		Logical Condition	
		+				•			•
		Actions Upon Pass	ctions Upon Fail A	Actions Upon Excer	otion				
		¥+ 🖬 🕯 × 🖷 🗐	- New		Action Temp	plate	~	Enable	
		Enable ActionType	e		Action Type		~		
					_		I		_
									~

Step 4: Input the Monitor Name, select whether to continue to start the service while the Monitor is running, and whether to turn off the monitor immediately when the actions are completed.

Designer Devices Tags Monitors	Actions Groups SFC Manage	errient .	Service Engine Status ServiceEngine	• Display
Monitors 4 2	Monitor Form ×			<u> </u>
+ 🖳 ་≡ ▸− ଠ 📲 🎙 🛅	5			
TMR_NCU	Monitor Setting			^
	Monitor Name	Enable Disable Monitor afte	er Actions completed	
	Schedule Time 00:00:00 🗢 ~ 00:00:00 🕏	CycleTime 1000	ms	
	Category Name TMR_NCU	Set Category Pass 1	Times	
	Service Engine ServiceEngine1 ~	Fail 1	Times	
	Description			
	Monitor Condition	·		
	Tag Name	Operator	Value	Logical Condition
				•
	Actions Upon Pass Actions Upon Fail	Actions Upon Exception		
	1 + 🖸 🝵 🗙 📹 🖃 🕒 New	Action Templat	te 🗸	Enable
	Enable ActionType	Action Type	~	C more

Step 5: Enter the Monitor Time, click "Set Group", select the Group and Service to be added.

Designer Devices Tags Monitors	Actions Groups SFC Management	Service E	ngine Status ServiceEngine1
Monitors	Monitor Form ×		•
TMR_NCU TMR_RMS_Project_ServiceEngine1	Monitor Setting Monitor Name Image: Constraint of the set of the se	Enable Disable Monitor after Actions com CycleTime 1000 ms Pass 1 Times Fail 1 Times	pleted
	Actions Upon Pass Actions Upon Fail Action 3 + C > New	Action Template	Enable

Step 6: Input the Monitor Frequency, the number of successful Monitor conditions, and the number of failed Monitor conditions.

Step 7: Decide whether to set the Tag Name condition.

Designer Devices Tags Monitors	Actions Groups SFC Manager	r nent	Service Engine Status ServiceEngine	• Display
Monitors + □ □, "E>- C □ ▼ □ ■ ☆ TMR_NCU	Monitor Form × Monitor Setting			
n ्यू IMK_RMS_Project_ServiceEngine1	Monitor Name Schedule Time 00:00:00 ♀ 00:00:00 ♀ Category Name TMR_NCU Service Engine ServiceEngine1 ✓	Enable Disable Monitor after CycleTime 1000 Set Category Pass 1 Fail 1	r Actions completed] ms] Times] Times	
	Monitor Condition	Operator	Value	Logical Condition
	Actions Upon Pass Actions Upon Fail 5 + 10 * 12 New Enable ActionType	Actions Upon Exception Action Templat Action Type	e	₩ Enable
				~

Step 8: Select Action to be executed under the conditions of different results. Step 9: Save.

3.5.3.4.6 Edit Monitor

	Designer	Devices	Tags	Monitors	Actions	Groups	。 SFC	Managen	nent			Service Engine S	status ServiceEngine	ə1	▼ Display
0	Monitors ┼│฿	'≣)IC	- 7 i	₽×	Monito	r Form 🗙									-
	TMR_NC	J S_Project_Servi	ceEngine1		Monitor N Monitor N Schedule Category Service E Descriptie	Setting ame Time 00:00:0 Name TMR_N ngine Service	0 😧 ~ 00 CU Ængine1	:00:00	✓ Enable Set Category	Disablı CycleTime Pass Fail	e Monitor after 1000 1 1	Actions completed] ms] Times] Times	1		^
					Monitor	Condition									
						Tag Name			Operator			Value		Logical Condition	
					+						•				-
						able Action	Actions Upo	<u>n Fall</u>	Actions Upon Exe	A	ction Template ction Type		~	🖌 Enable	
															v

Step 1: In the Monitor list, double-click the Monitor to be edited, and Monitor_Form appears on the right Step 2: Edit on the Monitor_Form tab.

Actions Upon Pass Actions Upon Fail	Actions Upon Exception	
· 十日前 🗙 🖷 🖷 🗖 New	Action Template	✓
Enable ActionType	Action Type Set	Tag Value 🗸
🕨 🔽 Send Emsil		
	Tag Name TMR_NCU	J_Station_Status
	Operator =	\sim
	Value Start	

Step 3: To Edit an Action, click the item to be edited in the Action List, and edit in the Edit Mode.

Step 4: After Edit is complete, click <Confirm>.

Designer Devices Tags Monitors	Actions Groups SFC	Management	Service Engine Status ServiceEngine	1 Display
Monitors + □ □, 'E C □ ▼ 1 • + TMR_NCU • TMR_RMS_Project_ServiceEngine1	Monitor Form Monitor Setting Monitor Name Schedule Time 00:00:00	CycleTime	ter Actions completed	Ĵ
	Category Name TMR_NCU Service Engine Description Monitor Condition	Set Category Pass 1	Times Times	
	Tag Name	Operator	Value •	Logical Condition
	Actions Upon Pass Actions Upon \$ + 13 * x <= x <= x New Enable ActionType	IFail Actions Upon Exception Action Templ: Action Type	ate	🖌 Enable
				Ŷ



3.5.3.5 Group

3.5.3.5.1 Introduction

Group is mainly for the function of TM Robot. It is used to monitor multiple Robots. Group can contain multiple Robots and use Action or Script Language to issue commands to the Group. Service Engine will send signals to the Robots in the Group at the same time.

3.5.3.5.2 Group explorer Function Bar Introduction



+	<new> Add New Group</new>
Narra	<rename> Edit Group Name</rename>
Ε,	<addgroup> Create Group cannot use this function</addgroup>
E	<expand all=""> Expand</expand>
•	<fold all=""> Fold</fold>
ζ	<refresh> Refresh</refresh>
	<copyitem> Copy Item Group cannot use this function</copyitem>
$\overline{\mathbf{Y}}$	<filter> Search Group cannot use this function</filter>
	<delete> Delete</delete>

3.5.3.5.3 Group Environment Introduction

9. Group Form



- 10. Group_Form
- (1) Group Setting

	Group Setting	
0	Group Name	
8	Service Engine ServiceEngine1 ~	
8	Robot List 🕂 + 🛃 🗙 🛅 New 🕘	
6	Robot Name	Robot Name

• Group Name: Input Group Name, Names are case-insensitive, so upper and lower case spellings will be regarded as the same name. Special characters and blanks cannot be used in names.

Select the Service Engine belongs to

€	Function bar					
+	<add> Add Robot, Change to Add Mode after clicked</add>					
F	Confirm> Confirm Add Robot					
\times	<cancel> Cancel Add Robot</cancel>					
	<delete> Delete Robot</delete>					
Nev	<add mode=""></add>					
	Note: In Add Mode, only <confirm>, <cancel> can be used</cancel></confirm>					

- Select Robot Name
- Add Robot List

3.5.3.5.4 Add New Group

Vi	ew(V) To	ool(T) Help(H)	Options(O)		•					
I	Designer	Devices	Tags 8	Monitors	Actions	Groups	SFC	Managemer	• nt	
rtions	Devices	B. 15		₽× Ì	Groups Form	×				
Groups	∎ ÷ D ≩ گ	efault ServiceEngine1 TestDevice TestRobot00 TestRobot01			- Group Setting Group Name Service Engine	TEST ServiceEngine	1	~	6	
	2 6	TestRobot02			Robot List + Robot Nam	19 🗙 🛅 New De			Robot Name	

Step 1: Click Group.
Step 2: Click " + ", Group_Form tab appears.
Step 3: You may pin the Group window.
Step 4: Input the Group Name and its Service.
Step 5: Select the Robot Name to join in this Group.
Step 6: Click Confirm to Add to the list, if there are other Robots to be Added, click Add to add to the Robot List as well.

Step 7: Save.

3.5.3.5.5 Edit Group

View(V) Tool(T) Help(H) Options(O)	
Desig <mark>ar</mark> Devices Tags M	nitors Actions Groups SFC Management
	x Groups Form X
G Default Fr ServiceEngine1 Fr TestRobot00 √T TestRobot01	Group Setting Group Name TEST Service Engine ServiceEngine1 ~
ිස් TestRobot02	Robot List + 1 × 1 hew Robot Name

Step 1: In the Group list, double-click the Group to be edited, the Group_Form appears on the right

Step 2: Edit on the Group_Form tab



If to Change Robot

Step 1: Click the Robot Name to be Changed Step 2: Robot Name Select Robot to be Changed Step 3: Click Confirm

If to Add New Robot

	-	A .:				
lig	Devices 4 ×	Actions Form	Service Engine - Management	Device Form	Editor DataControl_EN	Groups
sdi	+ ┉ 🖳 ་≡ ↦- ⊖ 🖳 🏹 🛅	1				
	■ Default FrestPowice Control Control Contr	Group Setting Group Name Service Engine Robot List	TEST	Robot Name	2 TestRobot01	Ŀ

Step 1: Click Add Step 2: Robot Name Select New Robot Step 3: Click Confirm

3.5.4 Dashboard Designer

The Dashboard is a visual presentation of the equipment and collected data managed by the Service Engine, real-time display of equipment status, and control of the project process. The Dashboard Console provides multiple components and Script Language, allowing users to easily achieve customer's requirements. Also provides multiple module components, such as TM Robot exclusive components. Users only need to drag the Robot Controller component from the toolbox, place it in the design window, and then set the Robot to be monitored by the component from the property window. When executed after deployment, it will display the relevant information of this Robot and the control the Robot (must comply with safety regulations) on the Dashboard. In addition, the TM Robot Vision component can display TMRobot vision job pictures in real time, and with visual presentation for the collected data to provide Report, chart and other components. Data can be obtained in the DB, and with the graphical settings to connect with each Table, to select the desired data columns to be presented. Present the data in reports and charts, and provide a variety of components. Refer to Chapter 3: Tool Box Components and Properties.



The following Figure explains the information flow of the Dashboard, and the Dashboard built in Builder Dashboard Designer can be deployed to multiple computers for viewing in different fields. For special messages, passwords can be set to allow viewing only by specific people. The data sources of the Dashboard include Service Engine, DB, PLC, DLL, etc. (Refer to 2.5.5 Script Language for details); The components on the Dashboard can be set to retrieve specific data and present the data. Present the data on the components to achieve the effect of real-time presentation.



3.5.4.1 Operation Interface Introduction

1. Enter Dashboard Designer

Enter the Page to select Open Old Files or Add New Files

Designer \rightarrow Console UI \rightarrow New or Open

🔤 Builder				
View(V) Tool(T) Help(H)	Options(O)			
Designer Devices	Tags Monitors	Actions	S Groups	Management
Console UI(C) Web Dashboard(W)	New(N) Open(O)			
Data Controls	Toolbox(T)			
Common Controls	Properties(P)			

If select New, enter the Setting Page, as shown below.

New Proj	ect
Project Name	
Save Path	C:\Program Files (x86)\Techman\TMmanager\TMM Builder\Project
Description	
Password	Create DateTime 2020年 3月23日 🗐 🗸
	Cancel OK

- a. Project Name: Input Project Name
- b. Save Path: Select the file saving path (currently the default is the former folder under the execution directory)

- c. Description: Describe the Purpose and Precautions of this Project
- d. Password: Decide whether to set a Project Password (if it is set, the Password is required to open the Project later)

After entering the Dashboard Designer, as shown in the figure below, it is mainly divided into three areas, the Toolbox on the left, the Design Window and Script Language in the middle, and the Property Window on the right

Note Note Note Note Note Note Note Note	View(V) Tool(T) Help(H) Options(O)				
Dergine Zuroda Tage Mundar Autor Gorgen Dergine Zuroda Dergine Zuroda	💥 . 🔄 🔎 🖬 🕋	🗿 📾 . 🏟 .		Service Engine Status ServiceEngine	1 Display
Image: Control in the image: Contro in the image: Contro in the image: Control in the image: Control	Designer Devices Tags Monitors Actions C	roups SFC Management			
Production Production <td>The</td> <td>NEW MY</td> <td></td> <td>P contine</td> <td></td>	The	NEW MY		P contine	
Sector	Then there the test		1.1.20 00 04.0	Anin Window Main Window	
Important Product Product <td>Shed Fider Controls</td> <td></td> <td>s [m, [c] #1.</td> <td>Rulat I</td> <td></td>	Shed Fider Controls		s [m, [c] #1.	Rulat I	
Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview Interview <t< td=""><td>Device Controls</td><td>Wintow</td><td></td><td></td><td></td></t<>	Device Controls	Wintow			
Current Cur	Common Controls			Accessibility	
Link Descent	Common composis	Automatic Artister		Advented Setting	
I suit Sorbitolity <td>2: Label</td> <td></td> <td></td> <td>Backmendinana</td> <td></td>	2: Label			Backmendinana	
Control Contro Control Control Control Control Control Control Control Control Co	at Button			Background Inspel evout	Tile
Reduction Conce Order Be character Conce Conce	GroupBox			CloseFormParryout	
Prod Backborn Backborn Backborn Cassedbarn Backborn Backborn Cassedbarn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backborn Backbo	PictureBox			Center	Default
Image: Control Second Control Control<	Panel			RusCubrose	True
India Balcair Contal India Contal Contal <t< td=""><td>ComboBax</td><td></td><td></td><td>Appearance</td><td></td></t<>	ComboBax			Appearance	
Indication Rescale Rescale <td>ed TextBox</td> <td></td> <td></td> <td>BeckColor</td> <td>Control</td>	ed TextBox			BeckColor	Control
El decidado de consecuencia de la decidado de consecuencia de la decidado de decidado de decidado de la decidado de la deci	Tab Control			FourColor	ControlText
Control Contro Control Control Control Control Control Control Control Control Co	detaGridViev			Size	600, 400
Mithageneries (Second Parket Parke	Redisbutton		8	Text	Automatic Actioner
Image: Standard S	Cheokox			Event	
In Markets For the second	Advindowshedianayar			UserChange Target	
Monitor Solvigation Controls Tore Solvigation Controls Tore Monitor Tore Monitor Monitor Tore	Webbrowser			Contra Illing	
Society of the first sector of the Context The Context sector of the Context The	UConcillar			Control Close Days	Tree
Monacha Tra Manacha Tra Manacha Tra Manacha Tra Manacha Tra Defoces U Poloces U Reloces Tra Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Transcention Tran	SubProtect			FormBoolecStyle	Sizable
Monarche C. 9 Manarche C. 9 Manarc				MaximinaBox	True
Image: Control of the Control of th				Mooinunfize	0,0
Image: Control Line Contro Line Control Line Control Line Control Line Control Line Control				MinimizeBox	True
Benderska trv Several Several				MinimunSize	0, 0
Image: Section 1 Image: Section 2 Image: Section 2<				Shawin Teskber	Tree
The fact that The Color The Color The Color				StatLocation	0, 0
The Column The Column The Column The Column The Column The Column				IterPosition	Manual
Window AttaCool Polar Window AttaCool Name				TopMost	Felze
Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha Windowsha				WindowAutoCoroll	False
Under Solar Control Lines				WindowState	Normal
TI Educ Cole Educ				ButColor 456 the beiggmand color of the Coulto	
		UI Editor Code Editor			

3.5.4.2 Toolbox Introduction

Toolbox is divided into 4 major categories

- a. Common Control
- b. Device Control
- c. Data Control

There are all types of components in each major category, which will be detailed later.

3.5.4.3 Design Window Introduction

The Design Page is composed of Builder/Script Language. The Builder can generate Dashboard from the Toolbox components to present the colorful changes of Dashboard with the logic operation of Script Language. Design Window Layout is as follows (Related functions will be introduced in later chapters).

amWindow	
🖳 Automatic Actioner	

- ① Builder tab, displaying the Project Name
- ② Dashboard Designer Function Menu
- ③ Builder / Script Language Page Switch

Script Language Page

NEW test 🗙		•
i 🖬 💽 🗂 📓	▏▆▕▓▝▋▝▋▐▖▆▎▆▖▋▝▋▝▋▓ <mark>▕▓</mark> ▝▋▝▌▝▌▝▋ ▋ &	
Guide:	- X 🖻 🗈 X 🕫 🔍 🖃 🗰 1	-
1		~
2 define:	//Variable declaration area	
4		
5 -}		
7 main: //	Program start point	
8 🖓 {		
9		
11		
12 close: /	/Program execution point before the project leaves	
15 ^L }		
16 17 dashboar	derror: //Program execution point after system error	
18 🛱 {		
19		
20 }		
22 servicee	engineerror: //Service Engine return error	
23		~
UI Editor Code Edi	litor	
Citte Edi		

Dashboard Designer Function List

Number	Function Icon	Function Description
1	5	Save
2		Run
3		Save as
4		Open Error Prompt Window
5		Delete
6	1	Clear tool
7		Place to Top
8	D	Place to Bottom
9		Keep Down
10		Кеер Up
11		Keep Left
12		Keep Right
13	* *	Center Horizontally
14		Center Vertically
15		Lock Component

16		Select Components and Change to the Same Size
17	8	Cut
18	*	Сору
19	1	Paste
20	£	Restore
21	t	Cancel Restore
22	Ŷ	Modify Project Password
23		Add Sub-window
24	品	Deployment
25	0	Check program syntax and automatic typesetting
26	≣([#] =	Set Database
27		Script Log

3.5.4.4 Property Window Introduction

Pr	operties	<u></u> д	x
Ma	ainWindow:MainWindow		•
ŏ	≝ ≋⊎ I 📼		
	Role		^
\sim	Advanced Setting		
	BackgroundImage		
	BackgroundImageLayout	Tile	
	CloseFormPassword		
	Cursor	Default	
	RunOnlyone	True	
\sim	Event		
	UserChange Target		
\sim	Windows		
	ControlBox	Тгае	
	ControlCloseBox	True	
	FormBorderStyle	Sizable	
	MaximizeBox	True	
>	MaximumSize	0, 0	Ľ
	MinimizeBox	True	
>	MinimumSize	0, 0	
	ShowInTaskbar	True	
>	StartLocation	0, 0	
	StartPosition	Manual	
	TopMost	False	~
B	ekColor		
De	fine the background color c	of the Control.	

① All components in the Design Window can be selected, and Display the currently selected components.

⁽²⁾ Property Description Window, displays the Description of the currently selected Property.

3.5.4.5 Toolbox Components and Properties

1. Properties

The Properties are divided into the following categories: Identification, Accessibility, Appearance, Advanced Setting, among which the components of Advanced Setting are different, and they are introduced when introducing Dashboard Designer.

1.1 Identification

Identification	
Anchor	Top, Left
Description	
Dock	None
IsIndex	True
Name	Button1
TabIndex	0
TagDataAutoChange	False
TagName	

3.5.4.5.1 Anchor

The method of automatic resizing that can define when the size of parent control item of the control items is resized. Anchor the control item to its parent control item to ensure that the anchored edge will remain the same relative to the parent control edge when the size of parent control item is resized (Mutually exclusive with Dock properties).

3.5.4.5.2 Description

The description of the component will be displayed in Resource and Script Language to facilitate easy identification.

3.5.4.5.3 Dock

When the size of the parent control item of the control items is adjusted, the automatic resize method of control item can be docked on one edge of its parent container, or it can be docked on all edges and fill the parent container.

3.5.4.5.4 IsIndex.

Is the component displayed in Script Language.

ma ₽{	<i>in:</i> name//程式進入點	
L}	指令索引 ▶	
	元件案引 🕨	Label1 (顯示生產量)
⊟{		Label2
Τ.		Label3
5		MainWindow

3.5.4.5.5 Name

Component Name

Avoid the same variable name as in Script Language. Avoid the same name as a Tag/Monitor/Action.

3.5.4.5.6 TabIndex

Alignment order of control items in the control container

3.5.4.5.7 TagDataAutochange

When the Property Text value changes, whether to automatically update the value of Tag.

3.5.4.5.8 TagName

The paired Tag Name. When the Dashboard is running, the Data from the Service Engine will be updated. The latest Data of the Tag can be displayed in the control item. The component to mount the Tag is explained as in the Figure below, press the button in the TagName Property, and the Tag Explorer will pop up. Select the target to be mounted (Components are required to correspond to the Tag Type).

TagName	x Properties		ą×
┼┉╽┇┇┝╼╽╣┇╹┓┩╽	Button:Button1		•
E Demo	o∷ ≋♥ ■ ✓ Identification		
CurrentDeviceModel	Anchor Description	Top, Left	
	Dock IsIndex	None True	
AchievingRate	Name	Button1	
FailQty	TabIndex TagDataAutoChange	U False	
	TagName		

3.5.4.5.9 Accessibility

Accessibility	
Role	

3.5.4.5.10 Role

After executing the set Dashboard, the user accessibility authority of this control item After pressing Icon, the setting screen is as follows.



- 1 Add Group
- 2 Set Group Content
- 3 Save Group Content File
- (4) Select Set Target and Properties
- **(5)** Click to Write
- 6 Setting List
- Click OK to Complete

3.5.4.5.11 Event

Event	
ClickEvent	
MouseDown	
MouseEnter	
MouseLeave	
MouseUp	
Target	

Provides 6 types of Event Trigger, and the trigger methods:

- a. Call Script Language
- b. Call Other Projects
- c. Execute Action
- d. Start Project

Settings are shown in the Figure below.

	FunctionName	function: Europionium
C BATTAction / 1988 Monitor		
4	5	
L		

- Fill in the custom function name in Script Language, after clicked OK, Script Language will automatically generate the named Function.
- 2 Select Dashboard prog file to call.
- 3 Select to execute Action or start Monitor.
- (4) Check the Action to be executed.
- **(5)** Check the Monitor to be started.

ClickEvent : After clicked the control item, it will trigger the trigger action.

Target : When the value of Property text is changed, it will trigger the trigger action.

MouseEnter : When the mouse enters the control item, it will trigger the action. It is usually used for the Picture Box component, when the mouse enters, the Icon of the Picture Box is changed.

MouseLeave : After the mouse leaves the control item, it will trigger the trigger action.

MouseUp : Occurs when the mouse pointer is on the control item and the mouse button is released.

MouseDown : Occurs when the mouse pointer is on the control item and the mouse button is pressed.

3.5.4.5.12 Appearance

Appearance	
BackColor	Control
Enabled	Тгие
Font	PMingLiU, 9pt
ForeColor	ControlText
Left	382
Size	100, 50
Тор	406
Visible	True

BackColor: Control Item Background Color Enable: Whether to Enable the Control Item Font: Font Setting

ForeColor: Control item Foreground Color

Left: The distance between the control item in the container and the left edge

Size: The width and height of the control

Top: The distance between the control item in the container and the upper edge Visible: Whether to display the control item in the screen

3.5.4.5.13 Advanced Setting

Due to the differences in the properties of each component, the description is combined into the following chapters.

3.5.4.6 Component

As mentioned earlier, the components are divided into Common Control, Data Control, and Device Control. This Chapter will introduce them in order.

3.5.4.6.1 Common Control

3.5.4.6.1.1 Label

o......o.....o oLabel o o......o

Purpose: Mark specific content, display Script Language variables, or display the content of mounted tag

Specific Properties:

Advanced Setting	
AutoSize	True
BorderStyle	None
FlatStyle	Standard
FormatStyle	
Image	
ImageAlign	MiddleCenter
RightToLeft	No
TextAlign	TopLeft

AutoSize: Automatically resizes the label size according to the font size

Borderstyle: Set the border style of the control item

FlatStyle: The appearance style setting when the mouse passes over or clicks on the control item

FormatStyle: Set the component display Format (p.s. general datatype property needs to be in numerical type)

Image: Set the Image displayed on the control item

ImageAlign: Set the Image Position displayed on the control item RightToLeft: Whether to use RTL language, that is, write from right to left RextAlign: Set the position of the Text

In Script Language Command:

In Script Language, use the components \rightarrow can set properties, go to Property Form to set other properties



3.5.4.6.1.2 Button

Q:	······	þ
ģ	Button 0	þ
ÓF	······O·······d	5

Purpose: Event triggered after Clicked

Specific Properties:

Advanced Setting	
BackgroundImage	
BackgroundImageLayout	Tile
BorderSize	1
Cursor	Default
FlatStyle	Standard
Image	
ImageAlign	MiddleCenter
TextAlign	MiddleCenter
TextImageRelation	Overlay

BackGroudImage: Background Image

BackgroundImageLayout: Background Image Layout

BorderSize: Set the control item border width

Cursor: Mouse cursor style

FlatStyle: Set the Flat Style appearance of the control item

Image: Set the Image displayed on the control item ImageAlign: Set the Image Position displayed on the control item TextAlign: Set the Text Alignment on the button control item TextImageRelation: Set the relative position of the Text and Image

In Script Language Command:

In Script Language, use the components \rightarrow can set properties ClickEvent \rightarrow Set the click event triggered functions of the button

3.5.4.6.1.3 GroupBox



Purpose: Multiple components of a Group can be moved together, this control item will display the frame around a Group of control items, and display selective header Specific Properties:

Advanced Setting	
BackgroundImage	
BackgroundImageLayout	Tile
FlatStyle	Standard

BackGroudimage: Background Image

BackgroundimageLayout: Background Image Layout

FlatStyle: Set the Flat Style appearance of the control item

3.5.4.6.1.4 PictureBox



Purpose: Picture Box control item for displaying images

Specific Properties:

Advanced Setting	
BackgroundImage	
BackgroundImageLayout	Tile
BorderStyle	None
Cursor	Default
Image	
ImageLayout	Normal

BackgroundImage: Background Image

BackgroundImageLayout: Background Image Layout BorderStyle: Set the border style of the control item Cursor: Mouse cursor style Image: Set the Image displayed on the control item ImageLayout: Set the Image displayed on the control item

In Script Language Command:

In Script Language, use the components \rightarrow can set properties ClickEvent \rightarrow Set the click event triggered functions of the button

3.5.4.6.1.5 Panel



Purpose: Used for the collection of Group control items Specific Properties:

Advanced Setting	
AutoScroll	False
BackgroundImage	
BackgroundImageLayout	Tile
BorderStyle	None

AutoScroll: Enables the scroll function when the content is larger than the visible area BackgroundImage: Background Image

BackgroundImageLayout: Background Image Layout

BorderStyle: Set the border style of the control item

In Script Language Command:

In Script Language, use the components \rightarrow can set properties HideSubWindow \rightarrow Hide the SubWindow embedded in the Panel component EmbedShowSubWindow \rightarrow Embed the SubWindow into the Panel component and show it

3.5.4.6.1.6 ComboBox



Purpose: Drop-down menu, select different options to trigger different events Specific Properties:

Advanced Setting	
DropDownStyle	DropDown
FlatStyle	Standard
Items	(Collection)
MaxDropDownItems	25
Sorted	False

DropDownStyle: Drop-down Box Style

FlatStyle: Set the Flat Style appearance of the control item

Items: Provide items can be selected, the input window as shown in the Figure below, and each option needs wrap

(The Figure below shows the addition of a, b, and c three options to the ComboBox)

MaxDropDownItems: Maximum Visible Count

Sorted: Whether the Options to be Sorted

String Collection Editor	2	x
Enter the strings in the collection (one per line):		
ab		^
		-
•		Þ
OK	Can	ıcel

In Script Language Command:

In Script Language, use the components \rightarrow can set properties

3.5.4.6.1.7 TextBox



Purpose: Text Box Control Item Specific Properties:

Advanced Setting	
BorderStyle	Fixed3D
CharacterCasing	Normal
MaxLength	32767
MinLength	0
Multiline	False
PasswordChar	
ReadOnly	False
ScrollBars	None
TextAlign	Left
UseSystemPasswordChar	False
Word Wrap	True

BorderStyle: Set the border style of the control item

CharacterCasing: Set input character ; Normal: Do not change the input character casing

Upper: input characters are automatically converted to uppercase, Lower: input

characters are automatically converted to lowercase

MaxLength: Maximum Length of control item

MinLength: Minimum Length of control item

Multiline: Whether to display in Multiple Lines

PasswordChar: Set Password Characters

ReadOnly: Read Only

ScrollBars: Scroll Style, None/Horizontal/Vertical/Double scroll

TextAlign: Alignment of Input Text

UseSystemPasswordChar: Whether to use the default password characters, the default is "False"

WordWrap: Whether to automatically wrap when the text is too long

In Script Language Command:

In Script Language, use the components \rightarrow can set properties

3.5.4.6.1.8 TabControl



Purpose: Multiple tab switching, control items can be placed Specific Properties:
Advanced Setting	
Alignment	Тор
Appearance	Normal
Change TabPageIndex	(Collection)
Cursor	Default
HotTrack	False
ItemSize	58, 18
Multiline	False
SizeMode	Normal

Alignment: Select Tab Position

Appearance: Tab Appearance

ChangeTabPageIndex: Change Tab Sequence

Cursor: The cursor style of the mouse on the component

HotTrack: Whether to change the appearance when the mouse is moved to the component

ItemSize: Index label display size

Multiline: Determines whether the index label can be displayed in multiple lines SizeMode: Index label size mode

In Script Language Command:

In Script Language, use the components \rightarrow can set properties SelectedTabPageIndex \rightarrow Specify to switch tab to a specific page

3.5.4.6.1.9 DataGridView



Purpose: Display Data Grid Specific Properties:

Advanced Setting	
AllowUserToAddRows	False 🗸
AllowUserToDeleteRows	False
AllowUserToOrderColumns	False
AllowUserToResizeColumns	True
AllowUserToResizeRows	True
AlternatingRowsDefaultCellStyle	DataGridViewCellStyle { }
AutoSizeColumnsMode	None
AutoSizeRowsMode	None
borderstyle	FixedSingle
CellBorderStyle	Single
ChangeRowRegisterlistName	
ColumnHeadersBorderStyle	Raised
ColumnHeadersDefaultCellStyle	DataGridViewCellStyle { BackColor=Color [Cont
ColumnHeadersHeight	18
ColumnHeadersHeightSizeMode	AutoSize
ColumnHeadersVisible	True
DataAutoUpdate	0
DataConnectID	
DataSqlCMD	
DefaultCellStyle	DataGridViewCellStyle { BackColor=Color [Wind
EditMode	EditOnKeystrokeOrF2
GridColor	ControlDark
ReadOnly	True
righttoleft	No
RowHeadersBorderStyle	Raised
RowHeadersDefaultCellStyle	DataGridViewCellStyle { BackColor=Color [Cont
RowHeadersVisible	True
RowHeadersWidth	41
RowHeadersWidthSizeMode	EnableResizing
ScrollBars	Both
SelectionMode	RowHeaderSelect

AllowUserToAddRows:

Get or set the value to indicate

whether the user has been shown

the option to add data rows

AllowUserToDeleteRows: Get or set the value to indicate whether the user is allowed to delete data rows in DataGridview

AllowUserToOrderColumns: Get or set the value to indicate whether the manual order adjustment of data columns is enabled

AllowUserToResizeColumns: Get or set the value to indicate whether the data columns can be resized

AllowUserToResizeRows: Get or set the value to indicate whether the data rows can be resized

AlternatingRowsDefaultCellStyle: Get or set the default cell style applied to the odd-numbered data rows of the DataGridView

AutoSizeColumnsMode: Get or set the value to indicate how to determine the width of the data columns

Borderstyle: The border style of get or set DataGridView

CellBoardStyle: The cell border style of get DataGridView

ChangeRowRegisterlistName: If Row has new, modified, or deleted event definitions, the system can temporarily store the Row in the list type variable for use by the function program.

ColumnHeardersBorderStyle: Get the border style applied to the data column header ColumnHeardersDefaultCellStyle: Get or set the style of default data column header ColumnHeadersHeight: The height of get or set data column header (Unit in pixels) ColumnHeardersHrightSizeMode: Get or set the value to indicate whether the height of the data column header can be resized, and whether the user can resize the height, or automatically resize the height according to the content of the header

ColumnHeardersVisible: Get or set the value to indicate whether to display the data row at the header of the data column

DataAutoUpdate: Update data automatically in a few seconds (0 means no update) DataConnectID: Database Connect ID

DataSqlCMD: T- SQL syntax

DefaultCellStyle: Get or set the default cell style to be applied to the cells in the DataGridView (The premise is that no other cell style properties are set) EditMode: Get or set the value to indicate how to start editing the cell GridColor: Get or set the grid line color of the partitioned DataGridView cells ReadOnly: Get or set the value to indicate whether the user can edit the cells of the DataGridView control items

Righttoleft: Get or set the value to indicate whether the items of the control item are aligned to support the locale setting using right-to-left fonts

RowHeadersBorderStyle: Get or set the border style of the first cell of the data row RowHeadersDefaultCellStyle: Get or set the default style applied to the first cell of the data row

RowHeadersVisible: Get or set the value to indicate whether to display the data column containing the header of the data row

RowHeadersWidth: Get or set the width of the data column containing the header of the data raw (Unit in pixels)

RowHeadersWidthSizeMode: Get or set the value to indicate whether the width of the header of the data row can be resized, and whether the user can resize the width, or automatically resize the width according to the content of the header

ScrollBars: Get or set the scroll bar type to be displayed in the DataGridView control items

SelectionMode: Get or set the value to indicate how to select the cells of the DataGridView

In Script Language Command:

CountColumn \rightarrow Get the count of columns DataBind \rightarrow Bind the Data source to control item DataSource \rightarrow Update the DataSource of control item DataSqlCMD \rightarrow T- SQL syntax DelValue \rightarrow Delete Datagridview Data GetCellValue \rightarrow Get the cell value of DataGridView component GetValue \rightarrow Get the data of the array or table, Need to use the extended property, (PS: If ->[..] extension property has not been added, the table returns all the values, and the columns are separated by',' symbols) LeavedRowIndex \rightarrow Get the number of the selected rows leave the DataGridView component CountRow \rightarrow Get the datatable type or the row count of the DataGridView component SelectedRowIndex \rightarrow Get or set the selected row number of the DataGridView component

3.5.4.6.1.10 RadioButton



Purpose: Allows the user to select a single option from a set of options and match it with other RadioButton control items. The control items can display Text, Image, or both

Specific Properties:



Checked: Get or set the value to indicate whether to select the control item

In Script Language Command:

Checked → bool type, set or get selected state of RadioButton

3.5.4.6.1.11 CheckBox

Q	·····O·····	·····0
ò 🗌	CheckBox1	ò
	·····	ò

Purpose: The user provides options, such as true/false. The CheckBox control item can display images or text or both

Specific Properties:



Checked: Get or set the value to indicate whether to select the control item

In Script Language Command:

Checked → bool type, set or get selected state of CheckBox

3.5.4.6.1.12 AxWindowsMediaPlayer



Purpose: URL property to get or set the name of the media item to be played Specific Properties:

Advanced Setting	
Ctlenabled	True
enableContextMenu	True
uiMode	full
URL	
volume	0

Ctlenabled: Define whether the component is enabled

enableContexMenu: Whether to enable the right-click menu function of the mouse uiMode: Set the player interface mode, Full/Mini/None/Invisible URL: Set the file path for play Volume: Set the volume

In Script Language Command:

EnableLoop → Whether to enable loop playing fullScreen → Whether to set to full screen Pause → Pause Play → Play GetState → Gets the State of control Item (Buffering, Last, MediaEnded, Paused, Playing, Ready, Reconnecting, ScanForward, ScanReverse, Stopped, Transitioning, Undefined, Waiting)



URL \rightarrow Set the File Path

3.5.4.6.1.13 WebBrowser



Purpose: Allows users to browse the components of the webpage, and link to

WebCam images

Specific Properties:

Advanced Setting	
bodyZoom	100
URL	
urlScrollPoint	0, 0

bodyZoom: Sets the display ratio of URL pages

URL: Sets the URL of the browsed web page

urlScrollPoint: the initial position of the WebBrowser Scroll Point

In Script Language Command:

Refresh → Refresh URL Path

URL \rightarrow Set component content file location

UrlScrollPoint \rightarrow Set the initial position of the Web Browser Scroll Point

3.5.4.6.1.14 HScrollBar / VScrollBar



Purpose: Scroll Function

Specific Properties:

Advanced Setting	
Cursor	Default
LargeChange	10
Maximum	100
Minimum	0
SmallChange	1

Cursor: Mouse cursor style

LargeChange: Gets or sets the value that will be added or subtracted from the Value property when the scroll box moves a large distance

Maximum: Upper limit of scrollable range

Minimum: Lower limit of scrollable range

SmallChange: Gets or sets the value that will be added or subtracted from the Value property when the scroll box moves a small distance

3.5.4.6.1.15 SubProject



Purpose: Can embed other Prog files

Specific Properties:

Advanced Setting	
AutoScroll	False
BorderStyle	Fixed 3D
ProjectFile	
ProjectFormBorderStyle	None

AutoScroll: Whether to allow Scroll Point to appear

BorderStyle: Control item border style

ProjectFile: The path of the project file to be executed

ProjectFormBorderStyle: The MainForm border embedded in the project

In Script Language Command:

ProjectFile \rightarrow The path of the project file to be executed RunProject \rightarrow Execute the set project StopProject \rightarrow Stop project running

3.5.4.6.2 Device Control

Device Control is mainly to link the Device and Tag created by the Builder to facilitate users to quickly deploy the Dashboard. As long as the DeviceActiveX/TagsActiveX component is mounted with a specific Device/Tag, the status of the Device/Tag can be displayed instantly when the Dashboard is running. In addition, RobotControl1/ RobotControl2 provides TM Robot modules. The user pulls out the component, mounts the Robot and runs the Dashboard to quickly monitor the Robert. The following chapters will introduce.

3.5.4.6.2.1 OperatorActiveX



Function: Establish a connection with TMflow Client, and can use it to link with TMflow Client to issue Operator commands.

Specific Properties:

Identification	
Anchor	Top, Left
Description	
Dock	None
IsIndex	True
Name	OperatorActiveX1
Accessibility	
Role	
Advanced Setting	
CMDTimeout	6000
TMFlowIP	127.0.0.1
TMFlowPort	3849
Appearance	
Left	90
Size	50, 50
Тор	61
Visible	True
Event	
HasErrorEvent	

CMDTimeout: TimeOut Seconds

ErrorCodeShowInt: As the Error code displayed by TMflow, True for the error code in decimal, and False, in hexadecimal.

TMflowIP: TMflow Client IP address (defaulted to the local IP: 127.0.0.1)

TmflowPort: The port number to communicatte with TMflow Client (Operator Command uses port 3849)

HasErrorEvent: When an Error occurs, execute a specific function

After pulling out the OperatorActiveX component from the toolbox, when the TMflow Client connection is normal, opening component will appear green, otherwise it will appear red; Operator Function can be used to drag this component. Refer to Appendix A for related commands: TMflow Command.

3.5.4.6.2.2 DeviceActiveX



Function: Show Robot status (Connect (green), Error (red), NoConnect (gray))

Specific Properties:

Advanced Setting	
Cursor	Default
DeviceName	
IsDisplayDeviceName	Trae

Cursor: The cursor style of the mouse on the component

DeviceName: After pressing the button, select TM Robot on the Explorer

IsDisplayDeviceName: Whether to display the Tag name on the control item

In Script Language Command:

DisplayPanelColor \rightarrow Set the panel color of the component

The following example explains how to transfer a Device from the Device setting Icon to Dashboard Designer.

 Set the Connect and Disconnect Icons on the Device page. Errors are always displayed with a red background, as shown in the Figure below. Click Browse in the red box and select the appropriate image

Tag Setting —					Dashboard Picture		
Tag Name	AchievingRate				Within Range Out of Range		
Service Engine	ServiceEngine	1 v Capture	Mode Service Loop Time	e v			
Тад Туре	Custom Variat	ole value Ty	rpe float	~			
Category Name	ProductionStat	tus Set Ca	ategory		Browse Browse		
Description					Bange		
Save Mode					85 < Value < 100		
- Custom Variable							
Custom Varia	able	Filter	Convert	Calcul	ation and Statistics Action After Value Change		
Default Value	0						

b. After pulling the DeviceActiveX component from the toolbox, go to the Device property bar and select the Robot with the image just set

Pro	perties	4 х		
Dev	iceActiveX:DeviceActiveX1	•		
	\$ ↓ ■			
	Identification			
	Anchor	Top, Left		
	Description			
	Dock	None		
	IsIndex	True		
	Name	DeviceActiveX1		
	Accessibility			
	Role			
	Advanced Setting			
	Curror	Defenit		
	DeviceName	A19500002NB		
L	DeviceName IsDisplayDeviceName	A19500002NB		
, V	DeviceName тартариауречисемание Арреатапсе	A19500002NB		
ľ	DeviceName IsDisplayDeviceName Appearance BackColor	A 19500002NB		
ľ	DeviceName INDISPRAYDEVICENAME Appearance BackColor Enabled	A 19500002NB		
ľ	DeviceName INDISPRAYDEVICENAME Appearance BackColor Enabled ForeColor	A 19500002NB		
×	DeviceName INDESPRAYDEVICENAME Appearance BackColor Enabled ForeColor Left	A 19500002NB 11ue Transparent True ControlText 324		
► ~	DeviceName INDESPRAYDEVICENAME Appearance BackColor Enabled ForeColor Left Size	A 19500002NB I lue Transparent True ControlText 324 50, 70		
► ~	DeviceName INDESPRAYDEVICENAME Appearance BackColor Enabled ForeColor Left Size Text	A 19500002NB True ControlText 324 50, 70 1		
-	DeviceName INDESPRAYDEVICENAME Appearance BackColor Enabled ForeColor Left Size Text Top	A 19500002NB 11ue Transparent True ControlText 324 50, 70 1 199		
• •	DeviceName INDESPIRATION INDES	A 19500002NB Transparent True ControlText 324 50, 70 1 199 True		
• •	DeviceName INDESPIRATE Appearance BackColor Enabled ForeColor Left Size Text Top Visible Event	A 19500002NB Transparent True ControlText 324 50, 70 1 199 True		
• •	DeviceName INDESPIRATE Constraine Appearance BackColor Enabled ForeColor Left Size Text Top Visible Event ClickEvent	A 19500002NB		

c. Click to execute Dashboard, the Robot status can be displayed instantly



3.5.4.6.2.3 TagsActiveX



Function: Tag can be linked. On the Tag setting page (the red box in the Figure below),

a reasonable range for the Tag can be set. If the Tag value exceeds the range to set one image, set another image within the range of the Tag value. When the Dashboard is running, from the change of the component icon to understand whether the monitored value exceeds the set range

Tag Setting —						- Dashboard I	Picture	1
Tag Name	AchievingRa	te				Within Range	e Out of Range	
Service Engine	ServiceEngi	ne1 v	Capture Mode	Service Loop Time	\sim			
Tag Type	Custom Vari	able \vee	Value Type	float	~			
Category Name	ProductionS	tatus	Set Category			Browse	Browse	
Description						Range	Ciear	
Save Mode						85 <	Value < 100	
Custom Variable								
Custom Varia	able	Filter		Convert	Calcula	ation and Statistic	Action After Value	e Change
Default Value	0							

Specific Properties:

Advanced Setting	
Cursor	Default
IsDisplayTagName	True

Cursor: The cursor style of the mouse on the component

IsDisplayTagName: Whether to display the Tag name on the control item

3.5.4.6.2.4 RobotControl1



Function: The RobotControl1 component can bind to the Robot, display the information of the Robot, and control the Robot. This component is divided into four pages, namely Total Info, Control, Total Chart, and Error List. The functions of each page are explained as follows

a. Total info



- 1) Robot light signal synchronization
- 2) Show Robot Status (Play / Edit / Pause / Error / Connect / NoConnect)
- 3) HMI version
- 4) Project Name
- 5) Running Time (Activity Rate)
- 6) Display Robot Speed

b. Control



1) Project deployment, Set initial Project

(The project to be deployed must first be uploaded from the Robot to Builder)

2) Robot Control (play, pause, speed up, speed down)

(Note that the use conditions are the same as using the Robot Stick)

c. Total Chart



- 1) Running Time, Activity Rate Chart
- d. Error List



1) Display Robot Error Code

Specific Properties:

Advanced Setting	
DefaultRobotName	
DefaultShowControl	True
IsChangeRobot	True
IsControl	True
Robot Group Name	
TagName	
WorkHour	Single 🛛 Аттау

DefaultRobotName: Set the Robot name to be mounted

DefaultShowControl: Whether to display the monitoring panel

IsChangeRobot: Whether to allow to change the monitoring Robot

IsControl: Whether the component is allowed to control Robot

RobotGroupName: The RobotControl1 component can select to mount the Group.

When executing the Dashboard, the user can select the Robot to be monitored on the component.

TagName: Mount the Tag, the value of this Tag will be synchronized with the Status of the selected Robot

WorkHour: In order to calculate the Activity Rate, this property needs to fill in the daily working hours

Robot Work Hour

Monday :	8	hr
Tuesday :	8	hr
Wednesday:	8	hr
Thursday :	8	hr
Friday :	8	hr
Saturday:	8	hr
Sunday:	8	hr
Cancel	ОК	

In Script Language Command:

ChangeRobotInitialProject → Set the Robot's initial project GetActivityRate → Get the ActivityRate information of the component GetDefaultProject → Get the initial project of the Robot selected by the component GetProjectListFromRobot → Get all projects of the Robot GetProjectListFromServiceEngine → Get the projects that the Robot has uploaded to the Service Engine GetRobotVersion → Get the version of the Robot selected by the component GetRobotIP → Get The IP address of the Robot selected by the component RefreshProjectList → Refresh Robot Project UploadRobotProject → Upload Robot project to Service Engine

Application Example:

Objective: Apply RobotControl1 component to deploy Robot project

- 1. Set the Service Engine IP at the Robot end
- 2. Device Explorer will automatically add new Robot



Can check whether there is a connection and whether the connection IP is correct in the Display Form Check. As shown in the Figure below, make sure that TM171421 is

connected to the Service Engine and the IP is 192.168.134.115.

-Connection List		
Connection Eist		
 Device Connection: 192.168.132.103:31203 192.168.134.115(TM171421) 	Refresh	

3. TM Robot Management downloads the project from the Robot to the Builder for management

TM Robot Management

Search TM Robot N	ame/ IP Address		Enabl	le		Status		Mod	ibus			
					~			~	_			\sim
TM Robot Name	IP Address	Enable	Service Engine Nar	me Ser	vice Engine State	Status	Modbus	Downlo	oad M	laintei	nance	Modify
A19500002NB	192.168.132.95	True	ServiceEngine1	Sta	rt	Disconnect	Disconnect	<u>Downlo</u>	oad Vi	ew	<u>Add</u>	<u>Edit</u>

Click Download as shown in above Figure, and the File Download window will pop up, as shown Figure below, select the "Project" tab on ①, all projects of the Robot will be displayed on ②, for example, select the project to be managed, ③ select Service Engine Folder, ④ click Download Key.

File Download			
Log Project			
Service Engine Folder Builder Folder			
	Set Default Project	Cancel	Download

4. Dashboard Designer pulls out the RobotControl1 component and sets up

mounting the Robot. Create new project, Toolbox → Device Control drags out RobotControl1 component on Dashboard Designer. The property DefaultRobotName selects TM171421



5. After executing the Dashboard Designer, enter the Control page and the project field will appear. For the project uploaded by the Robot just now, select the project to be deployed, and click "Upload Selected Project to Robot" to deploy the project to the Robot. (Note: This action can only be performed with the Robot in the blue light mode (Auto Mode), otherwise the project list will be reverse grayed)



3.5.4.6.2.5 RobotControl2

Connect Speed: 5%	<u>8</u> 1
Info Robot Name : TM171421 IP : 192.168.134.115 HMI Ver : 1.74.2100.18750 Project : e Health : ● Running Time 0 min/day Activity Rate : 0 % Total Hour : 8	Control MyTest MyTest Neil Neil2 Upload project
Robot Utilization	Play hour - Activity Rate
8 6 4 2 0 20200401 20200401 20200402 20200402	100 80 60 40 20 40 20 20200405 20200406
Error Statistic	View data for the past seven days
Robot Status	

Function: Same as the RobotControl1 function, the part of layout does different configuration and property control

Specific Properties:

Advanced Setting	
(Max) Control Visible	False
(Max) Error Statistic Visible	True
(Max) Robot Utilization Visible	True
(Max) Status Visible	True
(Min) Control Visible	False
(Min) Error Statistic Visible	False
(Min) Info Visible	False
(Min) Robot Utilization Visible	False
(Min) Status Visible	False
Default Expend	True
Default Robot Name	
Light Size	24, 24
Robot Image Path	
Robot Work Hour	Single 🛛 Array
Show Robot Image	False
Speed Label ForeColor	ControlDark
Tag Name	

(Max) Control Visible: Whether to show the Control Panel when expanded(Max) Error Statistic Visible: Whether to show the Error Statistic Panel when expanded(Max) Robot Utilization Visible: whether to show the Robot Utilization Panel when expanded

(Max) Status Visible: Whether to show the Status Panel when expanded

(Min) Control Visible: Whether to show the Control Panel when folded

(Min) Error Statistic Visible: Whether to show the Error Statistic Panel when folded

(Min) Info Visible: Whether to show the Info Panel when folded

(Min) Robot Utilization Visible: Whether to show the Robot Utilization Panel when folded

(Min) Status Visible: Whether to show the Status Panel when folded

Default Expend: Set whether to expand by default

Default Robot Name: Set the Robot name to be mounted

Light Size: Set the size of the display light

Robot Image Path: Control Visible must be set to true, the selected image will be displayed on the Control panel, and the Control panel cannot be used.

Robot Work Hour: In order to calculate the Activity Rate, this property needs to fill in the daily working hours

Show Robot Image: Set whether to show the selected image

Speed Label ForeColor: Set the fore color of the Robot speed display

TagName: Mount the Tag, the value of this Tag will be synchronized with the Status of the selected Robot



- ① Status Panel
- ② Info Panel
- **③** Control Panel
- **④** Robot Utilization Panel
- **⑤ Error Statistic Panel**

In Script Language Command:

Same as RobotControl1

3.5.4.6.2.6 WebApiButton

WebApiButton

Purpose: Dashboard gets data from the client through HTTP Request for application.



Specific Properties:

Advanced Setting	
Configuration	WebApiButton1_webapi.json

Configuration \rightarrow Open WebApiButton setting screen

🖳 Web Api Co	nfigurator	-		Properties	
	5			WebApiButton: WebApiButton1	
Setting Testin				8= 승규 1 📾	
Control Name	WebAp/Button1			 Identification 	
Method	Get		×	Anchor	Top, Left
LIPI				Description	
-				Dock	None
Token				Isindex	Irue Web 4 -: Detter 1
Туре	JObject			honoribility	webApiBullolli
	0		A	Pole	
				 Advanced Setting 	
				Configuration	WebApiButton1 webspi ison
				Appearance	
				BackColor1	Chartreuse
				BackColor2	📕 Aqua
				BorderColor	Gray
				ButtonStyle	Simple
				Enabled	True
Pequest				Font	Microsoft Sans Serif, 9pt
Request				ForeColor	Black
				Left	242
				> Size	100, 23
				Text	₩ebApiButton
				Top	148
				Visible	Irue
				Club Forest	
				ChickEven	
			T		
Message					
R	eset	Save			
1					

- (1). Open the Property Configuration page
- (2). Setting tab
 - Method: Provide Get / Put / Post / Delete
 - URL: HTTP Request URL
 - Token: Client provides Token for identity verification
 - Type: JObject (not accepted other than JSON)

(3). Testing tab

Submit - After submitting, the relevant message will be displayed on the screen

etting Testing						
	http://192.168.133.11/api/values	×	Name	Data Type	Value	
IRL						
		-		1	1	
	0	<u>^</u>	Name	Data Type	Value	
equest						
	{	¥				
esponse Header	"StatusCode": 200, "ReasonPhrase": "OK",					
	"Time": "2020-03-23T19:15:42.4371899+08:00 }	0"				
	{ "id": 1.					
asponse Rody	"name": "Test", "isComplete": true					
soponse body	}					
ception						
			-			

In Script Language Command:

(1). Enable

Whether the component is enabled for user interaction

- (2). GetWebApiValue Get client WebService HTTP Response
- (3). SetRequestParameters

Set HTTP Request to bring in parameters/conditions

(4). SetUrlParameters

Set HTTP Request Url to bring in parameters/conditions

(5). Submit Submit (Execute) HTTP Request

Application Example

- 1. Example Note
- The example explains the use of client WebService using Microsoft MVC example
- (2). HTTP Method return object must be JObject
- (3). HTTP Method Body must be JObject
- (4). If interested, refer to the Microsoft sample teaching course at the following link https://docs.microsoft.com/zh-tw/aspnet/core/tutorials/first-web-api?view=aspnet core-3.1&tabs=visual-studio-code#the-deletetodoitem-method

- (5). Git Download Url https://github.com/wrayz/DemoToDoltem
- 2. Create Dashboard
- (1). Open Builder and New Dashboard
- (2). Open ToolBox > Data Control
- (3). Use WebApiButton
- 1) WebApiButton Method Get
- 1. Method Get without Parameter

Imme WebApButton1		http://192.168.133.11/apyvaues	* Name	Data Type	Value
http://192.168.133.11/api/valued	URL				
JObject			* Name	Data Type	Value
	Request				
		6			
	Response Hea	der "FstatusCode": 200, "ReasonPhrase": "OK", "Time": "2020-03-23T19:15:42.4371899+08:00)			
		{ "d": 1, "name": "Test",			
	Response Bod	y "sComplete": true			
		-			
	Exception				

(1). Objective: Get data through HTTP Request

2. Method Get with Parameter



- (1). Objective: Get data through HTTP Request + URL bring in conditions
- (2). Set WebApiButton1 Configuration Setting
- (3). URL bring in query conditions:/*parameter*/

🖳 Web Api Co	onfigurator	-		×
Setting Testin	g			
Control Name	WebApiButton1			
Method	Get			*
URL	http://192.168.133.11/api/values//*id*/			
Token				
Туре	JObject			
Request	0			▲▼
Reset			Save	

- (4). Test WebApiButton1 Configuration Testing
- (5). Set parameter type & Value
- (6). The Submit result is shown as the Figure below

🖳 Web Api Config	gurator				-	
Setting Testing						
	http://192.168.133.11/api/values/1		Name	Data Type	Value	
URL		1	r id	Integer	1	
	0	×	Name	Data Type	Value	
				5557 MAG		
Request						
		*				
Response Header	{ "StatusCode": 200, "ReasonPhrase": "OK",					<u>^</u>
	"Time": "2020-03-24T11:52:29.0326374	+08:00"				•
Response Rody	"id": 99, "name": "Test Parameter",					
Kesponse body	scomplete: true					
						×
Exception						
						*
					Submit	

(7). WebApiButton creates a Button Event. Select a custom function, named as "submit"

Editor C:\Program Files (x8 ×	₹ Property		
			Set Event
	lentification Anchor Description Dock Inforce Nume Accessibility	Top, Left None True WebApiButton1	 自訂商式 [Main →] [mhmit] I whmit I who where we have a set of the left of th
C WebApiButton	Rolf Advanced Setting Configuration Event Clokfavnt	WebApiButton1_webapi.joon	○ 執行Action / 啟動 Monitor
			Cancel

- (8). After naming the Button Event, enter the Code Editor to set and write
- (9). NOTE

SetUrlParameters

The parameter name is the same as the WebApiButton Configuration URL

GetWebApiValue Use all lowercase for the property name



(10). Dashboard Execution Result

	- 1	~
	Result	
WebApiButton	Test Parameter	

- 2) WebApiButton Method Put
- 1. Objective: Update data via HTTP Request

🕡 Web Api Co	Setting	- 0 ×	👻 Web Api Confi	Testing			-	o x
Centrol Name Method	l WebApButtens Per		Setting Testing	http://192.168.133.11/api/values/5	Name	Data Type String	Value 5	
URL Token Type	http://192.168.133.11/ap/values/**d*/		URL		Name	Data Tuna	Value	
	("seek") "Tet". "Scalester": Ster		Request	'name': 'Test', 'scomplete': true }	PLEI - JE	Dem 1974	VINT	
Request			Response Header	{ "StatusCode": 200, "ReasonPhrase": "OK", "Tme": "2020-03-24T20:10:32.049267+08:00" }				
			Response Body	0				
			Exception					
R	set Sav	10					Submit	

- 2. Set WebApiButton1 Configuration Setting
- 3. URL adding parameters method:/*parameter*/
- 4. Request brings in JObject

Method	Put
URL	http://192.168.133.11/api/values/ /*id*/
Token	
Туре	JObject
	{ "name": "Test", "isComplete": true }
Request	

- 5. Test WebApiButton1 Configuration Testing
- 6. Set parameter type & Value
- 7. The Submit result is shown as the Figure below

	http://192.168.133.11/api/values/5	A	F	Name	Data Type		Value	
URL			۲	id	String		5	
Request	{ "name": "Test", "sComplete": true }	*		Name	Data Type		Value	
Response Header	{ "StatusCode": 200, "ReasonPhrase": "OK", "Time": "2020-03-24T20:10:32.049267+08:00"	×						
Response Body	0							
								v A
Exception								
]	Submit	

- 8. After naming the Button Event, enter the Code Editor to set and write
- 9. NOTE

SetUrlParameters

The parameter name is the same as the WebApiButton Configuration URL

GetWebApiValue Use all lowercase for the property name

The example shows that ID is the reason of Result

HTTP Server designs successfully and will return id = 0

fu	nction:submit
₽{	
	//設定URL Parameter Value
	WebApiButton1->SetUrlParameters = "id",1
	//提交 HTTP Request
	WebApiButton1->Submit = true
	//關閉按鈕互動機制(反灰)
	WebApiButton1->Enabled = false
	//HTTP Response 是否有值
	if (WebApiButton1->GetWebApiValue->["id"] != "")
Ė.	{
	//取出Reponse的name屬性值,並指定給Label元件顯示文字
	<pre>lblResult->Text = WebApiButton1->GetWebApiValue->["id"]</pre>
L_	}
	,/1. 愈用按鈕互動機制
	WebApiButton1->Enabled = true
L	

10. Dashboard execution result

2		-	×
	Result		
WebApiButton	0		

3) WebApiButton Method Post

1. Objective: Create or modify data via HTTP Request

🛃 Web Api Co	onfigurator Setting	- 0	×	💀 Web Api Confi	lesting	-	o x
Setting Testin	19			Setting Testing			
Control Name Method URL	WebApButton1 Post http://192.168.133.11/api/values		*	URL	http://192.168.133.11/a Nam	e Data Type	Value
Token Type	JObject { "name": "Test" }			Request	{ "name": "Test" } Nam	e Data Type	Value
Request				Response Header	{ "StatusCode": 200, "ReasonPhrase": "OK", "Time": "2020-03-24T20:33:22.96638	354+08:00"	×
				Response Body	"id": 99, "name": "Test", "isComplete": true }		- -
			w.	Exception			A.

T

- 1. Set WebApiButton1 Configuration Setting
- 2. Request to Set JObject
- 3. Use /*parameter*/ in order to import data to Code Editor

Setting	Testing	
Control	Name [WebApiButton1
Method	[Post
URL	[http://192.168.133.11/api/values
Token	[
Туре	[JObject
		{ "name": "/*name*/" }
Request	t	

- 4. Test WebApiButton1 Configuration Testing
- 5. Test Data

6. The Submit result is shown as the Figure below

🖳 Web Api Confi	gurator					- 0	×
Setting Testing			2				
	http://192.168.133.11/api/values	-		Name	Data Type	Value	
URL		_					
		×					
	{ "name": "Test"	-		Name	Data Type	Value	
	}		Þ	name	String	Test	
Request		Ļ					
		~					
	{ "StatusCode": 200						
Response Header	"ReasonPhrase": "OK", "Time": "2020.03.24T20.37.04 9785254±08.00"					-	
	{						-
Response Body	"name": "Test",						
Response bouy	"isComplete": true						
							-
							-
Exception							

- 7. After naming the Button Event, enter the Code Editor to set and write
- 8. NOTE

SetRequestParameters

parameter is the same as the WebApiButton Configuration Request /*parameter*/

The example shows that id is the reason of Result

HTTP Server designs successfully and will return ID = 99

```
function:submit
∃{
    //設定Request Parameter Value
    WebApiButton1->SetRequestParameters = "name", "DemoTest"
    //提交 HTTP Request
    WebApiButton1->Submit = true
    //關閉按鈕互動機制(反灰)
    WebApiButton1->Enabled = false
    //HTTP Response 是否有值
    if (WebApiButton1->GetWebApiValue->["id"] != "")
Ė
    {
       //取出Reponse的name屬性值,並指定給Label元件顯示文字
       lblResult->Text = WebApiButton1->GetWebApiValue->["id"]
    }
    // 啟用按鈕互動機制
    WebApiButton1->Enabled = true
 -3
```

9. Dashboard execution result

2			-	×
		Result		
	WebApiButton	99		

4) WebApiButton Method Delete

1. Objective: Delete data via HTTP Request

Web Api Co	enfourator Setting X	🛃 Web Api Config	gurator Tes	sti	ng		- 🗆 ×
Theo Apreo		Setting Testing					
Setting Testin	9	_	http://192.168.133.11/api/values/99		Name	Data Type	▲ Value
Control Name	WebApiButton1	URL			▶ id	Integer	99
Method	Delete +				Name	Data Type	Value
URL	http://192.168.133.11/api/values/99				11001154	Units (Jpc	7.000 St.
Token		Request					
Туре	JObject						
	0	Response Header	<pre>{ "StatusCode": 200, "ReasonPhrase": "OK", "Time": "2020-03-24T20:47:56.6401! }</pre>	543+01	8:00"		
Request		Response Body	"id": 99, "name": null, "isComplete": true }				
		Exception					
Reset	Save						- Submit

- 2. Set WebApiButton1 Configuration Setting
- 3. URL brings in Delete Condition: /*parameter*/

Method	Delete
URL	http://192.168.133.11/api/values//*id*/
Token	
Туре	JObject
	0
Request	

4. Test WebApiButton1 Configuration Testing

5. Test Data

URL Request Image: StatusCode": 200, "ReasonPhrase": "OK", "Time": "2020-03-24T20:47:56.6401543+08:00" Response Header Image:	99				11ctp.//192.100.133.11/ap/(values/33	
Request Image: StatusCode": 200, "ReasonPhrase": "OK", "Time": "2020-03-24T20:47:56.6401543+08:00" Response Header Image: StatusCode": 200, "ReasonPhrase": "OK", "Time": "2020-03-24T20:47:56.6401543+08:00" Response Body Image: StatusCode": 200, "ReasonPhrase": "OK", "Time": "2020-03-24T20:47:56.6401543+08:00"		Integer	id	P		JRL
equest equest esponse Header esponse Body find:: 99, iname:: null, iname	Value	Data Tura	Alama	v	[D]	
equest esponse Header sponse Header sponse Body	Value	Data Type	Name			
esponse Header { "StatusCode": 200, "ReasonPhrase": "OK", "Time": "2020-03-24T20:47:56.6401543+08:00" } esponse Body { "id": 99, "name": null, "isComplete": true }						equest
esponse Header sponse Header esponse Body sponse Body sponse Body sponse Body sponse Body				-		
esponse Body "IsComplete": true			:00"	543+08:	{ "StatusCode": 200, "ReasonPhrase": "OK", "Time": "2020-03-24T20:47:56.6401; }	esponse Header
					<pre>''id": 99, "name": null, "isComplete": true }</pre>	esponse Body
cception						ception

6. The Submit result is shown as the Figure below

7. After naming the Button Event, enter the Code Editor to set and write

8. NOTE

SetUrlParameters

parameter is the same as the WebApiButton Configuration URL /*parameter*/

The example shows isComplete is the reason of Result

HTTP Server designs successfully and will return true

```
function:submit
₽{
    //設定Request Parameter Value
    WebApiButton1->SetUrlParameters = "id",5
    //提交 HTTP Request
    WebApiButton1->Submit = true
    // 關閉按鈕互動機制(反灰)
    WebApiButton1->Enabled = false
    //HTTP Response 是否有值
    if (WebApiButton1->GetWebApiValue->["isComplete"] != "")
Ė.
    {
       //取出Reponse的name屬性值,並指定給Label元件顯示文字
       lblResult->Text = WebApiButton1->GetWebApiValue->["isComplete"]
    }
    // 啟用按鈕互動機制
    WebApiButton1->Enabled = true
∟,
```

9. Dashboard execution result

2		-	×
	Result		
WebApiButton	True		

3.5.4.7 Data Control

The purpose of Data Control is to convert data into visual components, and the source of data is mainly from Tag and Database. Among them, CircularGaugeExpress, LinearGaugeExpress, DigitalGaugeExpress, ProgressBarExpress, can directly mount the Tag, and can see the change of visualization from the component. The data source of TagsChart, TagsPeriodChart, TagsPieChartPeriod, TagsAggregationChart is the database. The user must first set the Tag value storage method, and can select the Tag with the stored data to display the chart, even if the data is from the database, but to use these four components, there is no need to use SQL syntax. And Chart2Express and ReportViewer can be set by the user to present various reports and chart presentation styles

3.5.4.7.1 CircularGaugeExpress



Purpose: Tag value visual presentation, can reflect Tag Value in real time Specific Properties:

Advanced Setting	
1.StyleChooser	CircularGaugeExpress1_Style.xml
2.MinValue	0
3.MaxValue	100
4.MajorTickCount	11
5.RangeEnd1	33
6.RangeEnd2	66

StyleChooser \rightarrow Choose the style of presentation, there are many styles to choose from





Minvalue → To present the smallest value of this component
 MaxValue → To present the largest value of this component
 MajorTickCount → Components distinguish in several ticks

③ RangeEnd1 → The components can be divided into three stages, this is the maximum value of the first stage

4 RnageEnd2 \rightarrow This is the maximum value of the second stage

3.5.4.7.2 LinearGaugeExpress



Properties and purposes are the same as CircularGaugeExpress

3.5.4.7.3 DigitalGaugeExpress



Purpose: Thing Value Visual presentation, can reflect Thing Value in real time Specific Properties:

Advanced Setting	
1.StyleChooser	DigitalGaugeExpress1_Style.xml
2.DigitCount	5
3.DisplayMode	Matrix8x14
4.NumberRound	2

StyleChooser \rightarrow Choose the style of presentation, there are many styles to choose from

DigitCount \rightarrow The component presents several digits

 $\mathsf{DisplayMode} \rightarrow \mathsf{Choice of Digit style}$

NumberRound \rightarrow Decimal places

Application Example:

Objective: The Manipulator Power Consumption value is captured by TM Robot through Modbus Protocol, and the value is displayed on the Dashboard using the DigitalGaugeExpress component

(1). Set Tag to capture the Manipulator Power Consumption value of TM Robot through

Modbus.

Set Tag Type to select Get Modbus, Device Name to select one TM Robot Other settings depend on the information provided by TM Robot, as shown in the Setting Figure below

Tag Setting —				Dashboard Pict	ture	
Tag Name	TMPowerConsumption			Within Range	Out of Range	
Service Engine	ServiceEngine1	Mode Service Loop Time	\sim			
Tag Type	Get Modbus v Value Ty	pe float	\sim			
Category Name	Demo Set Ca	tegory		Browse	Browse	
Description				Range	Clear	
Save Mode				0 < Val	lue < 0	
Get Modbus —						
Get Modbu	us Filter	Convert	Calcul	ation and Statistics	Action After Value	Change
Device Name	A01060115PC					
Function Code	InputRegister(0x04)	\sim				
StartAddress	7344	Format: Decimal				
Length	2					
Convert Type	[byte]] Convert to Float[]	~				

- (2). Open new project, use DigitalGaugeExpress, and mount: TMPowerConsumption Create new project, drag out the DigitalGaugeExpress from ToolBox, select TMPowerConsumption for its property Tag Name, and select 2 for the property NumberRound
- (3). Execute Project



After clicking Execute, the Dashboard can display the Manipulator Power Consumption value of TM Robot

3.5.4.7.4 ProgressBarExpress

50 %

Purpose: Tag value visual display, can reflect Tag Value in real time (mostly used as

Progress Bar)

Specific Properties:

Advanced Setting	
EndColor	ForestGreen
StartColor	GreenYellow

EndColor: Component End Color selection

StartColor: Component Start Color selection

3.5.4.7.5 Chart2Express



Purpose: Use database data to present charts

Specific Properties

Advanced Setting	
1.DataSource	Please click button right.
2.SqlBuilder	Please click button right.
3.LayoutDesign	Please click button right.
4.RefreshCycleTime(Sec.)	0
5.TimeOut(Sec.)	1

 $\mathsf{DataSource} \rightarrow \mathsf{Data} \ \mathsf{Source}$

SqlBuilder \rightarrow Set by Sql LayoutDesign \rightarrow Layout Design RefreshCycleTime (Sec.) \rightarrow Refresh Cycle Time Timeout (Sec.) \rightarrow Timeout Seconds

Operation Example:

Drag Char2Express to the project, click the property DataSource

a. Select the data source DB, click Finish after selecting

Provider:	Microsoft SQL Server
Server name:	localhost
Authentication type:	Windows authentication
User name:	
Password:	
Database:	
	TMMainDB
	TMMLogDB
	TMMDemoDB

b. Click property SqlBuilder

c. There are two ways to select data under SQL syntax in Sql Builder, as follows:

(1). Use Allow Edit SQL (Check Allow Edit SQL on the top right)

Note: Only the syntax beginning with Select can be used
Query Builder			- 🗆 ×
			Allow Edit SQL
BoxPlotPoints	-ct "BoxPlotPoints".*		A
HarufacturingLog ManufacturingCoder MaterialMaster RobotStatusLog	irda "door."BoxflotFoints","DeviceNo" = @deviceNo)a: where (("BoxflotFoints","DeviceNo" = @deviceNo)a:	nd ("BoxPlotPoints","ColumnName" = N'Current'))	
	Data Preview	×	
	Device No Co	olumn Name Value1 Value2 Series Type Meaning Log Time	
	TM01 Cu	urrent 0.4488 0.4488 Main lower 02/26/2020 05:35:04 PM	
	TM01 Cu	urrent 0.5 0.5 MaxAvgMin min 02/26/2020 05:35:04 PM	
	TM01 Cu	urrent 0.55007 0.55007 Median median 02/26/2020 05:35:04 PM	
	TM01 Cu	urrent 0.55014 0.55014 MaxAvgMin avg 02/26/2020 05:35:04 PM	
	TM01 Cu	urrent 0.59999 0.59999 MaxAvgMin max 02/26/2020 05:35:04 PM	
✓ Columns of BoxPlotPoints	TM01 Cu	urrent 0.65169 0.4488 VerticalLine upper:lower 02/26/2020 05:35:04 PM	
DeviceNo String	TM01 Cu	urrent 0.65169 0.65169 Main upper 02/26/2020 05:35:04 PM	
ColumnName String			
Value1 Double Value2 Double		Close	
SeriesType String			
Meaning String			
LogTime DateTime			
	t		•
Preview Results Filter Edit Param	eters	OK	Cancel
ManufacturingOrder MaterialMaster RobotStatusLog	where ("BoxPlotPoints"."Devic and ("BoxPlotPoints"."Co	ceNo" = <u>KraviceNo</u> olumnNa <mark>s</mark> e" = N'Currer <mark>:</mark> '))	
	Query Parameter	rs	×
	Name	Type Expression Value	
	deviceNo	String TM01	
	Uticetty	Sung L 1901	
✓ Columns of BoxPlotPoints DeviceNo	String		
ColumnName	String		
Value1	Double		
valuez ForiorTuno	Double		
Meaning	String Preview	Add Remove OK	Cancel
LogTime	ate Time		
Preview Results	Edit Parameters		

• Write SQL syntax selection data in the blank space

• Click Preview Results to view the selected data content

If there is data change requirement, click Edit Parameters to set the parameters
 As shown in the Figure: Add deviceNo parameter, bring in @deviceNo in SQL syntax,
 Script Language must set *SqlParameters* (can refer to 4. Example Operation: Example 4).

(2). Do not use Allow Edit SQL (do not check Allow Edit SQL on the top right)

Query Builder				- 🗆 X
			6	Allow Edit SQL
BoxPlotPoints ManufacturingLog ManufacturingOrder MateriaMaster RobotStatusLog	0	BoxPlotPol * (Al Columns) DeviceIo ColumnName	<pre>select "BoxPlotPoints"." from "dow."BoxPlotPoints" "BoxPlotPoints" where (("BoxPlotPoints"."DeviceNo" = deviceNo) and ("BoxPlotPoints"."ColumnName" = N'Current'))</pre>	
		Value2 Value2 SeriesType LogTme	Fiker Editor Fiker Group Fiker And - + × BoxPlotPoint_DeviceNo = @ deviceNo BoxPlotPoint_Value1 = @ Enter a value	×
 Columns of BoxPlotPer DeviceNo ColumnName Value1 Value2 SeriesTyme 	oints String Double Double String	Table PoxPjotner	[BoxPlotPoints.DeviceNo] = ?deviceNo And [BoxPlotPoints.Value1] = ?	•
Meaning LogTime	String Date Trme		Select only 0 records starting with index 0 Select only distinct values OK	Cancel
Preview Results	Filter Edit Parameters		ОК	Cancel

• Drag the Table to the blank space in the middle, and check the data fields to be selected

- Can filter data through Filter
- SQL syntax is automatically generated
- d. Click property LayoutDesign

e. Select the chart data on the left, click Data to drag the SQL column data into the value below (Argument: X-Axis Value, Value: Y-Axis Value)



LayoutDesign Introduction

(This only lists the commonly used functions, refer to DevExpress official documents for more detailed explanations)

- Chart

(Set Chart Style, Color)



APPEARANCE

Appearance	Chart Appearance
Palette	Chart Color (Palette)
Back Color	Chart Background Color

Ø BORDER

Visibility	Whether the Chart has Border
Color	Border Color
Thickness	Border Thickness

O Appearance

Appearance Name	Chart Appearance Name
Back Color	Chart Background Color
Background Image	Insert Chart Background Image
Border	Chart Border Setting
Padding	Set the Chart Spacing
Palette Base Color	Select the Chart Color as Palette Color Number
Number	ex: Blue II , 5 is
Palette Name	Chart Palette Name

- Series

(Chart data, can set display style, label, data filter)





GENERAL

Visible	Decide whether this data is displayed in Chart
Name	Data Name
Show in Legend	Whether to Show in Legend
Labels Visibility	Check to show data value label
View	Data View Type (Bar, Line, Spline, etc.)

2 VIEW: GENERAL

(Take the Bar Graph as an example)

Bar Width	Bar Width
Color Each	Each Bar displays the color of each Palette

6 VIEW: LAYOUT

X-Axis	X-Axis corresponding to this data
Y-Axis	Y-Axis corresponding to this data

VIEW: BORDER

(Take the Bar Graph as an example)

Visibility	With or without Bar Border
Color	Bar Border Color
Thickness	Bar Border Thickness

TM Operator Platform User Manual

VIES: APPEARANCE

Color	Bar Color
Fill Mode	Bar Fill Mode
	Empty: Normal
	Solid: Solid
	Gradient: Gradient, with Gradient Layer
	Hatch: Hatch Lines

6 Data

Argument Data Member	Data value, X-Axis (Data tab setting)
Filter Criteria	Filter Data
Value Data Members	Data value, Y-Axis (Data tab setting)

- Label

(If the Data Table has added value Label, the style can be set in Label)





• GENERAL

(Take the Bar Graph as an example)

Visibility	Decide whether the value label is displayed	
Position	Label display position	
	Auto: Center of the Bar	
	Top: Top of the Bar, Beyond the Bar	
	Center: Center of the Bar	
	Top Inside: Top of the Bar, Inside the Bar	
	Bottom Inside: Bottom of the Bar	
Indent	If the label display position is Top Inside or Bottom	
	Inside,	
	The distance of Bar can be adjusted	
Line Visibility	If the label display position is Top, select whether or	
	not to display the label line	
Line Length	Set label line length	

@ TEXT OPTONS

Text Pattern	Add Text in Label ex.100 (ms)
Text Orientation	Label Orientation
	Horizontal: Horizontal
	Top To Bottom: From Top to Bottom
	Bottom To Top: From Bottom to Top

Text Alignment	Label Alignment Method
	Near: Align to Left
	Center: Center
	Far: Align to Right

● APPEARANCE

Text Color	Text Color
Background Color	Text Background Color
Fill Mode	Fill Mode

BORDER

Visibility	Decide whether the Label has a Border
Color	Border Color
Thickness	Border Thickness

6

Font	Label Font Size, Font, Bold, Underline
Shadow	With or without Shadow, Shadow color

6

Max Width	Maximum Width
Show for Zero Values	Decide whether to display 0

- Primary AxisX

X-Axis Setting





• GENERAL

Visibility	With or without X-Axis
Alignment	X-Axis Position
	Near: Bottom
	Far: Top
	Zero: Zero
	Center: Center
Minor Count	Number of Grid Lines
Interlaced	Whether the Background has an Interlaced Color

@ GRID LINES

Visible	Whether the Background has Major Grid Line
Minor Visible	Whether the Background has Secondary Grid Line

6 TICKMARKS

Visible	With or without Major Tick Marks
Minor Visible	With or without Secondary Tick Marks

d LABEL

Visible	With or without X-axis Label
Text Pattern	Add Text in Label ex.100(ms)

Visibility	With or without X-Axis Title
Word Wrap	Decide whether to Wrap Automatically
Max Line Count	Maximum number of Title lines
Text	Title Text

6 VISUAL RANGE

Auto	Decide whether the visible range of X-Axis is automatic
Min Value	If it is not automatic, set the Minimum Value
Max Value	If it is not automatic, set the Maximum Value

Ø WHOLE RANGE

Auto	Decide whether all ranges of the X-Axis are automatic
Min Value	If it is not automatic, set the Minimum Value
Max Value	If it is not automatic, set the Maximum Value

Appearance

Color	X-Axis Tick Mark Color
Interlaced	Whether the Background has an Interlaced Color
Interlaced Color	Background Interlaced Color
Thickness	X-Axis Line Thickness

Behavior

Crosshair Axis Label Options	Crosshair background color and text settings
Interlaced Fill Style	Background Interlaced Color Fill Mode
Show Behind	Whether there is Underline

- Primary AxisY

Y-Axis Setting



GENERAL

Visibility	With or without Y-Axis
Alignment	Y-Axis Position
	Near: Bottom
	Far: Top
	Zero: Zero
	Center: Center
Minor Count	Number of Grid Lines
Interlaced	Whether the Background has an Interlaced Color
Logarithmic	Whether the Y-Axis is logarithmic
Logarithmic Base	Logarithmic Base

@ GRID LINES

Visible	Whether the Background has Major Grid Line
Minor Visible	Whether the Background has Secondary Grid Line

6 TICKMARKS

Visible	With or without Major Tick Marks
Minor Visible	With or without Secondary Tick Marks

LABEL

Visible	With or without Y-Axis Label
Text Pattern	Add Text in Label ex.100(ms)

6 TITLE

Visibility	With or without Y-Axis Title
Word Wrap	Decide whether to Wrap Automatically
Max Line Count	Maximum number of Title lines
Text	Title Text

6 VISUAL RANGE

Auto	Decide whether the visible range of Y-Axis is automatic
Min Value	If it is not automatic, set the Minimum Value
Max Value	If it is not automatic, set the Maximum Value

WHOLE RANGE

/	Auto	Decide whether all ranges of the Y-Axis are automatic
---	------	-------------------------------------------------------

TM Operator Platform User Manual

Min Value	If it is not automatic, set the Minimum Value
Max Value	If it is not automatic, set the Maximum Value

Appearance

Color	Y-Axis Tick Mark Color
Interlaced	Whether the Background has an Interlaced Color
Interlaced Color	Background Interlaced Color
Thickness	Y-Axis Line Thickness

Behavior

Crosshair Axis Label	Crosshair background color and text settings		
Options			
Interlaced Fill Style	Background Interlaced Color Fill Mode		
Show Behind	Whether there is Underline		

- Secondary X-Axes

Click "+" to add Secondary X-Axes as the secondary X-Axis, and the property setting is the same as the X-Axis

- Secondary Y-Axes

Click "+" to add Secondary Y-Axes as the secondary Y-Axis, and the property setting is the same as the Y-Axis

Default Legend
 Legend Setting



GENERAL

Visibility	Decide whether there is a Legend
Text Visible	Decide whether there is Legend Text
Marker Mode	Legend Marker Style
	Marker: Series 1
	Check Box: Series 1

Check Box and Marker: 🔽 Series 1
None: None

❷ LAYOUT

Alignment	Legend Position
Direction	Arrangement

6 APPEARANCE

Marker Height	Marker Height
Marker Width	Marker Width
Background Color	Marker Background Color
Text Color	Text Color
Visibility	Decide whether the Legend has Border
Color	Border Color
Thickness	Border Thickness

TITLE

Visible	Decide whether there is a Legend Title		
Text	Title Name		
Text Color	Title Color		
Word Wrap	Decide whether to Wrap Automatically		

● APPEARANCE

Background Image	Legend Background Image	
Fill Style	Fill Mode	
Font	Legend Text Setting	
Shadow	Decide whether there is Shadow, Color of Shadow	

Behavior

Margins	Margins
Padding	Padding
Text Offset	Distance between Label and Legend Marker

- Title

Title Setting





• GENERAL

Visibility	With or without Title		
Word Wrap	Decide whether to Wrap Automatically		
Max Line Count	Title Line Count		
Dock	Title Position		
Alignment	Title Alignment Method		
Indent	Title Margins		
Lines	Title Text		

Appearance

Font	Text Setting
Text Color	Text color

Application Example:

Objective: Display Robot's Output and Average CycleTime at different times (8~18H)

(Output is Bar, CycleTime is Spline)



1. Set Database (example is TMMLogDB)

Connection Editor		×
Select the data	provider and specify the connection properties.	
Provider:	Microsoft SQL Server	Ŧ
Server name:	localhost	-
Authentication type:	Windows authentication	*
User name:		
Password:		
Database:	TMMLogDB	*
	Next	Finish

2. Set SQL capture value (the columns are Hourly, Output, Average CycleTime)



TM Operator Platform User Manual

SELECT DATEPART (HOUR, LOGDateTime) as Hourly, COUNT(*) as Output, ROUND (AVG (CAST (ThingValue as DECIMAL (18, 2))), 2) as Average FROM [dbo].[Things_History] WHERE ThingName='TMR2_CT' and LOGDateTime > CONVERT (CHAR (10), GETDATE (), 120) GROUP BY DATEPART (HOUR, LOGDateTime)

3. Click the Chart on the left, and select the properties of Bar color, background color, and whether or not there is a border on the right



4. Title- Modify the properties on the right according to the Legend (Title position, Title name, Title font size,



Chart	Cycle Time	1		Options Properties	
Series (1)	Cycle Time	2		Enter text to search	
> Series 1	55		Series 1	Appoaranco	
🗸 🚹 XY-Diagram	50 -		Jenes 1	> Font	Tahoma 18nt
> Default Pane				Text Color	White
Additional Panes (0)	CH			Dehavior	
> Primary AxisX	40 -			Alignment	Near
 Primary AxisV 	25			Dock	Тор
 Phillip Axist Considerable (0) 	35 -			Enable Antialiasing	Default
Secondary X-Axes (0)	30 -			Indent	0
Secondary Y-Axes (0)				> Lines	String[] Array
📃 Default Legend	25 -			Text	Cycle Time
Additional Legends (0)	20 -			Visibility	Default
Titles (1)				Word Wrap	No
Title	15 -			Misc	
	10			Tag	
Annotations (0)	5 -				

5. **Default Legend-** Modify the properties on the right according to the Legend (Legend background color, legend font color, font size, legend position)





6. **Primary AxisX-** Modify the properties on the right according to the Legend (number of grid lines, label added text, set maximum and minimum values, grid line color, Label font color)

•	í	Options Prop	erties
🗸 📲 Chart	Cycle Time	GENERAL	
 I Series (1) 	Cycle Hille	GENERAL	
> Series 1		Series 1 Visibility:	
🗸 📲 XY-Diagram	55	Alignment:	Near -
> Default Pane		Minor Count:	1 *
Additional Panes (0)	50 -		
Primary AxisX	45 -	Interlaced:	
> Primary AxisY		Logarithmic:	
Secondary X-Axes (0)	40 -	Logarithmic B	Base: 10 0
Secondary Y-Axes (0)			
Default Legend	35 -	GRID LINES	
Additional Legends (0)			
✓ 11 Titles (1)	30 -	Visible:	
Title		Minor Visible:	
Annotations (0)	25 -		
	20 -	TICKMARKS	
		Maible	
	15 -	Visible:	V
		Minor Visible:	×
	10 -		
		LABEL	
	5	Visible:	
		Task Datha	(4)/(-)
	8(h) 10(h) 12(h) 14(h	n) 16(h) 18(h)	: {A}(II) ····
			¥.



7. **Primary AxisY-** Modify the properties on the right according to the Legend (number of grid lines, label added text, set maximum and minimum values, grid line color, Label font color)





8. Secondary AxisY1- Click "+", Modify the properties on the right according to the Legend (number of grid lines, label added text, set maximum and minimum values, grid line color, Label font color)





9. **Series**- Modify the properties on the right according to the Legend (Series name, value display (check), drag data field)





10. Add New Series - Click "+" at Series, Select Line

•	4 + 4
0	il il
× alt Q	Cvcle 1
× .	Series (1)
	Side By Side Bar 3D Stacked 100%
	Point And Bubble Series
	> oo Point
	Bubble
	Line Series
Ý	Line Line
	Line Stacked 100%
	្វឹរ Step Line
	No Spline
	C Scatter Line
	w ^W Swift Plot

11. Modify the properties on the right according to the Legend (Series name, value display (check), Y-Axis, drag data field)

0	◆ →	*						Options Properties Da	ita
~ .1 0	hart	Cycle	Time					GENERAL	
× "I	Series (2)	Cych						GENEROL	
>	Output					Out	put	Visible:	
>	Cycle Time(ms)					— Cycl	le lime(ms)	Name.	Cycle Time(mb)
× 🚽	XY-Diagram							Show in Legend:	~
>	Default Pane	1000(11)					(0000()		
	Additional Panes (0)	1000(N) -					- 10000(ms)		
>	Primary AxisX							View:	Line -
>	Primary AxisY	900(N) -					- 9000(ms)		
	Secondary X-Axes (0)							VIEW: GENERAL	
~	Secondary Y-Axes (1)	800(N) -					- 8000(ms)	Color Fach:	
	Secondary AxisY 1								
	Default Legend	700(N) -					- 7000(ms)	VIEW LAYOUT	
	Additional Legends (0)	700(14)					7000(113)		
× 1	Titles (1)							Pane:	Default Pane 👻
	Title	600(N) -					- 6000(ms)	X-Axis:	Primary AxisX 🔹
	Annotations (0)								
		500(N) -					– 5000(ms)	1-70,05	occondary rocor 1
		400(N) -					- 4000(ms)	VIEW: MARKER OPTIONS	
								Marker Visibility:	
		300(N) -					- 3000(ms)	Color:	
								Size	10 1
		200(01)					2000()	5120.	
		200(N) -					- 2000(ms)	Kind:	Circle
		100(N) -					- 1000(ms)	VIEW: APPEARANCE	
								Color:	
			0(b) 10(1)	12(6)	14(b) 460	-) 10(1-)			
			8(n) 10(n)	12(h)	14(n) 16(i	1) 18(n)			



12. **XY-Diagram**- Modify the properties on the right according to the Legend (Background color, whether to have border lines)



More Chart related examples are in the Appendix.

3.5.4.7.6 ReportViewer

C)						 	 	 0	 				 			Q
		File	Print	: Previ	iew												^
	~	~			~ •	~	Find		∎	▶	⊳ I		*	~	~	~	8
																	-
Ċ)																Ó
																,	-
	•																•
è	Not	hing					 	 	 	 		1	00%	 			- +

Purpose: Use the data in the data chapter to present the report Specific Properties:

Advanced Setting	
1.ReportDesigner	ReportViewer1_Report.repx
2.RefreshCycleTime(Sec.)	0
3.TimeOut(Sec.)	3
Template	

Report Templates None

ReportDesigner \rightarrow Report Designer RefreshCycleTime(Sec.) \rightarrow Refresh Cycle Time TimeOut(Sec.) \rightarrow TimeOut Seconds Report Templates \rightarrow Report Templates

ReportDesigner:

- Home	Layout Page View	«					🔟 Designer 🕞 Prev	ew 🕡 Scr
E Save	Add Data Data Add Parame	ter Paste 🛞 Cop	Times New Roman • 9	.75 ▼ = = ×		Extract Style	valable styles.	
Report	Data	Clipboard	Font	Alignment	Borders	Styles	G	
XtraRe	eportExtension* ×						- Report Explorer	4 1
	¥ 🖹 (¹						r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r	eld List
Group	and Sort					4	× Backgroun	00.00 -
E Ad	d a Group * 🐁 Add a Sort *	X Delete O Move L	Ip 🔘 Move Down				Border Color	O -
Field Na	eme	Sort Order Show	Header Show Footer				Border Das	
		Intell Pop Vert Intell Add Cockedes Met Intell for the term forman 1975 Intell and I		Border Wid	1 :			

1. Left Toolbar

Can be used to design screen

	Pointer	Pointer
Α	Label	Label
	check Box	Check Box
	Rich Text	Text Box
	Picture Box	Picture Box
	Panel	Panel
	Table	Table
ab	Character Comb	Character Comb
~	Line	Line
	Shape	Shape
	Bar Code	Bar Code
	Chart	Chart
2	Cross tab	Cross tab
	Gauge	Gauge
	Sparkline	Sparkline
	Sub-Report	Sub-Report
	Table of Contents	Table of Contents
	Page Info	Page Info
	Page Break	Page Break
÷	Cross-band Line	Cross-band Line
	Cross-band Box	Cross-band Box

2. Report data source setting (picture uses Database)

Hor	ne Pag	ge View		_							
H Save	Add Data	dd Calculated Field dd Parameter	Paste 🖓 Copy	Times New Roman → B I <u>U</u> S I →	9.75 • <u>A</u> •			Extract Style	There are no available st Add a new style to unlock th	vies. • gallery.	
Report		Data	Clipboard	Font		Alignment	Borders		Styles	G.	
Xtral	ReportExtension*	×									
	a no da si si		1	2			5 6	· · · · · ·	2 4 7 8 8 92		
A					Data Sou	rce Wizard				×	
		label1									
						Select the data s	burce type.				
		T I 1									
ab											
							Database	Entity Frame	ework Object		
6											
-											
							X				
							Excel File	1500	YRO		
Σ							LACOTTO	35011	AFO		
										Next Finish	
- 000	n and Cost										

d

Set the database (the Provider example is Microsoft SQL Server) (the DB example is TMMainDB)

Data Source Wizard		×
Select the data pr	ovider and specify the connection properties.	
Provider:	Microsoft SQL Server	-
Server name:	localhost	-
Authentication type:	Server authentication -	
User name:		
Password:		
Database:	TMMainDB -	-
	Next Fin	ish

Select data source (example is RobotActivityRate)

Data Source Wizard	×
Create a query or select a stored procedure. Data columns selected from specific tables and/or views will be automatically included into a separate query.	
✓ ■ Tables	^
> DeviceStatusTimeRecords	
> IPCError_History	
astStationStatus	
> 🔽 RobotActivityRate	
RobotProjectList_History	
> SFC_CodingABC_History	
> SFC_StationData	
> SFC_StationDetail	
> SFC_StationIDLink	
> StationRobotDevice	
> StationStatusDetails	
> Things_History	
Views	
> Stored Procedures	
Queries	0 -
Manage Relations Next	Finish

Pull the data table from the Field List on the right to the report, and then click to set the specified data source from the upper left corner of the report.

• * -												
E- Hor	<mark>ne L</mark> ayout Page View									🔰 Designer	Preview 🖸	Scripts
믭	R Add Calculated Field	📄 💥 Out	Times New Roman × 9.75 ×	= = =			There are no weighted styles					
Save	Add Data 🛄 Add Parameter Source	Paste 🕥 Copy	В І <u>Ш</u> S 🔤 ~ <u>А</u> ~	E = = =	☞ - = -	Extract Style	Add a new style to unlock the gallery.					
Report	Data	Clipboard	Font	Alignment	Borders		Styles	6				^
Xtra	ReportExtension* ×									Field List		3 # X
	tere en terefereter	2	3 · · · 1 · · · 4 · · · 1 · · · 5							 sqDataSouri RobotAct 	iel NdvQata	
A	Report Tasks									Parameters	on grant of the second s	
	Design in Report Wizard											
	Edit Bindings		bot.0 [Robot.0 [Robot.0 [Robot.1 [Ro	botil [Robot.0 [Rob	otil							
	Data Source SqiDataSour	cel •										
	Data Member 🔍 None				<							
	Filter String	taSource1										
	Detail Count at Design Time											
1	Measure Units				1							
	Chula Chaot											
	Watermark											
~												
	📵 Add Nev	w DataSource										
										👬 Report Explor	er 🚦 Field List	
										Properties	0	1 # X
-1										XtraReportExte	nsion Report	*
÷.										🥕 💣 🌻	1 2 1	k -
2										Calculated	Collection)	
()										Data Adap	none)	٠.
-										Data Mem	RobotActMtyRate	e - 0
<u> </u>										Data Source	qDataSource1	- 0
										Filter String		0

After setting is complete, click Preview on the upper right

Norme Home				
E Print Quick Parametes Scale Margins Orientation Size Find Thumbnals	Bookmarks Editing Fields Page Page Page Page	I Nany Pages Zoom Out Zoom Zoom In	Page Color Watermark Export E-Mail	
Document Print Page Setup rs	Navigation	Zoom	Page Background Export	
	2/21/202 A110900 38318 TMMSe robot1 0 12NB rvicel	1.77.160 Blue 1oLyJgS 0 0 Conn 0.27519 WHSz0v uXskt6Y PA==	iect S	
	2/24/202 A110900 217 TMMSe robot1 0 12NB rvicel	1.77.160 Blue 1oLyJgS 0 0 Comm 0.27519 WHSzOv uXakt6Y PA==	iect S	
	2/25/202 A110900 741 TMMSe robot1 0 12NB rvicel	1.77.160 Blue KZAM7 0 0 Edit 0.27519 Vk5cj5E vy3Wha qVw-	8	
	2/26/202 A110900 0 TMIMSe robot1 0 12NB rvicel	1.77.160 Blue KfAM7 0 100 Conn 0.27519 Vk5cj5E vy3tWha qVw-	aect S	

The save close screen is as follows

View(V) Tool(T) Help(H) Options(O)		
Device Thing Monitor Action Group	STC Toolbox Propeny	TMM Service Status TMM/Service1
Years Carbon Parka	Note: 1000000000000000000000000000000000000	Pagarithe and gardfeed Pagarithe and gardfeed Pagarithe and gardfeed Pagarithe and gardfeed Pagarithe and gardfeed Pagarithe and pagarithe and pagarithe and pagarithe and pagarithe Pagarithe and Pagarithe and Pagarith
原題 46 年 49 년 18 국		
Destoy Not1) Service/		

3. File Location

After ReportViewer is complete, a File will be generated under the default installation path: Builder\Project\[Dashboard Name] \ [ReportViewer1_Report] .repx

Application Example

Objective: The objective of the example is to add two buttons in the Dashboard to bring in different conditions, and for the Dashboard to display different reports.

Step 1: Set the Parameters

		Extract Add a new Style	are no available styles. style to unlock the gallery.	* * *					
Alignment	Borders		Styles	G					1
					 Field L 	list		07	×
	. 6				, , ,	RobotActivityRate colDataSource2 Parameters			
Manage Qu	eries			×					
Name									
▶ RobotA	ctivityRate								
					🗛 R	eport Explorer 🛛 🛼 Field List			
					Prope	rties		□ #	×
					sqlD	ataSource1 SqlDataSource			-
	~			_	<u> </u>	Enter text to search			^
Add	Remove	E	OK Cancel		Cor	nnection Name	localhost_TMMLogDB_Connection		
					> Cor	nnection Options	(Connection Options)		•
					> Cor	nnection Parameters	(Connection Parameters)		•
					> Qu	eries	(Collection)		
					Rel	lations	(Collection)		•
				# >	(Na	ame)	sqlDataSource1		

Query Editor				×
Configure query p	parameters and preview the result.			
Name	Туре	Expression	Value	
Par	String			
Preview	Add Remove		Next Fini	sh

Step 2: Set the Syntax and paste the SQL Script below



SELECT [DeviceName], [Robot_Status] FROM [TMMLogDB].[dbo].[RobotActivityRate] WHERE[Robot_Status]=@Par

Step3: Drag Query (RobotActivityRate) to the Report



Step 4: Design two Buttons and create incidents

Window				
window.				
	Time	Name	Robot Status	RMSName
			Connect	Edit
			·······	

Step 5: Click open Code Editor and write function Script to import the Instructions of SqlParameters Property

- (1) con: Report Display Field Robot_Status is the Query Results of Connect
- (2) edit: Report Display Field Robot_Status is the Query Results of Edit



Step 6: Execute the program and use the Button just set to select the required data

Time	Name	Robot Status	RMSName
2/2/2020 12:00:00 AM	robot0	Connect	TMMService1
2/2/2020 12:00:00 AM	robot2	Connect	TMMService1
2/2/2020 12:00:00 AM	robot3	Connect	TMMService1
2/2/2020 12:00:00 AM	robot4	Connect	TMMService1
		Connect	Edit
<			>

Time	Name	Robot Status	RMSName
2/2/2020 12:00:00 AM	robot1	Edit	TMMService1
2/2/2020 12:00:00 AM	robot5	Edit	TMMService1
2/2/2020 12:00:00 AM	robot6	Edit	TMMService1
2/2/2020 12:00:00 AM	robot7	Edit	TMMService1
		Connect	Edit

When click (con) Button, Dashboard will execute ReportViewer1->SqlParameters = "Par", "Connect"

Will receive the following SQL Syntax

SELECT [DeviceName], [Robot_Status]

FROM [TMMLogDB].[dbo].[RobotActivityRate]

where [Robot_Status]='connect'

<

>

When you click (edit) the Button, the Dashboard will execute ReportViewer1->SqlParameters = "Par", "Connect"

Will receive the following SQL Syntax SELECT [DeviceName], [Robot_Status] FROM [TMMLogDB].[dbo].[RobotActivityRate] where [Robot_Status]='edit'

3.5.4.7.7 DateTimeExpress



Purpose: Select time and date components (In use, the time selected by the user can be obtained by Script Language as the basis for obtaining data from the database)

			*				
•		A	oril 20	020		•	
SU	МО	TU	WE	TH	FR	SA	
29	30	31	1	2	3	4	
5	6	7	8	9	10	11	
12	13	14	15	16	17	18	
19	20	21	22	23	24	25	\checkmark
26	27	28	29	30	1	2	
3	4	5	6	7	8	9	
ОК		Toda	y	Clear		Cancel	12:00:00 AM 🗘

Specific Properties:

Advanced Setting	
Show Time	True

Show Time \rightarrow Whether to Show Time

[True]

•		A	oril 2	020		•
SU	МО	TU	WE	TH	FR	SA
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9
ок		Toda	y	Clear		Cancel


[•										
	•	April 2020									
	SU	МО	TU	WE	TH	FR	SA				
	29	30	31	1	2	3	4				
	5	6	7	8	9	10	11				
	12	13	14	15	16	17	18				
	19	20	21	22	23	24	25				
	26	27	28	29	30	1	2				
	3	4	5	6	7	8	9				
	Today Clear										

3.5.4.7.8 TagsChart



Purpose: Draw a chart based on Tag's historical data (Tag value must have record in order to be used in this component, as set in the Figure below)

- Tag Setting —			
Tag Name	TMPowerConsumption		
Service Engine	ServiceEngine1	✓ Capture Mode	Service Loop Time 🗸 🗸
Tag Type	Get Modbus	\sim Value Type	int ~
Category Name	Demo	Set Category	
Description			
Save Mode	🔘 Last Data 📿) Data Changed	O Capture All

Specific Properties:

Advanced Setting	
AxisX	Minute
AxisY	(Collection)
AxisY_Field Type	Count
AxisY_Title	
ComponentStyle	TMMStyle
DataAutoUpdate	False
IsDisplay3D	False
Palette	Light
ShowCount	60000
Title	

AxisX \rightarrow Set the interval of X-Axis, and can set month, week, day, hour, minute AxisY \rightarrow Select the data source and presentation method of the Y-Axis, the drop-down menu of the ThingName field will have the Thing of the stored data to choose from, and the SeriesType can select the line style

Set	y-axis				
	Tag Name			Series	Туре
•			~		~
			Can	cel	ОК

AxisY_FieldType \rightarrow The mode of data to be presented, provides Maximum, Sum, Average, and Count.

AxisY_Title \rightarrow Y-Axis Title ComponentStyle \rightarrow Chart Style DataAutoUpdate \rightarrow Whether to Auto Update Chart Data IsDisplay3D \rightarrow Whether to Display 3D Chart Palette \rightarrow Line Color ShowCount \rightarrow How many sets of data to be displayed by chart Title \rightarrow Chart Title

Application Example:

Objective: Following the example of DigitalGaugeExpress, change the data of Manipulator Power

Consumption into Chart

1. Open TMPowerConsumption Tag Form to save Tag Data

- Tag Setting		
ing coming		
Tag Name	TMPowerConsumption	
Service Engine	ServiceEngine1	Service Loop Time 🛛 🗸
Tag Type	Get Modbus 🗸 Value Type	float \checkmark
Category Name	Demo Set Category	
Description		
Save Mode	🔿 Last Data 🔘 Data Changed	O Capture All

2. Open New Project, use TagsChart, and set it to display the TMPowerConsumption data chart After Dashboard Designer pulls out TagsChart, set the properties as follows

AxisY Property Setting:

Set the SeriesType of TMPowerConsumption to Line

3. Execute Dashboard

TMPowerCo			
	hum	 many	
20			8
	- 17:07	-10:41	

After execution, the chart will display TMPowerConsumption related information. In just a few simple steps, the data can be intercepted by the database to generate the chart

3.5.4.7.9 TagsPeriodChart



Purpose: Same as TagsChart, with more time intervals to choose from, and update button Specific Properties: Same as TagsChart



3.5.4.7.10 TagsPieChartPeriod

Purpose: Draw a pie chart based on the historical data of Tags. When the Data is under multiple tags, it is partitioned by the Tags to show the value to be displayed. When it is one Tag, it is partitioned by the value of the Tag (the Tag value needs to be recorded in order to be used in this component, set as shown in the Figure below)

Specific Properties:

Advanced Setting	
ComponentStyle	TMMStyle
Data.	(Collection)
Data_Field Type	Count
DataAutoUpdate	Null
Date1	2020-04-08
Date2	
IsDisplay3D	False
IsDisplayPercent	False
Palette	SeaGreen
Title	

ComponentStyle \rightarrow Chart Style

Data \rightarrow Select the data to be presented in the pie chart

If only one Tag is selected, the Tag value will be used as the basis for partitioning the pie chart. If multiple

Tag values are selected, the Count of each Tag value will be used as the basis for partitioning

Select 3 Tags to distinguish by Tag count



Select 1 Tag to distinguish by the value of Tag



DataAutoUpdate → Set time for data actively update (Null represents not actively updated)

Data1 \rightarrow Set data start date

Data2 \rightarrow Set data end date (default Now)

IsDisplay3D \rightarrow Whether the chart is presented in 3D Palette \rightarrow Line Color ShowCount \rightarrow How many sets of data to be displayed by chart Title \rightarrow Chart Title

3.5.4.7.11 TagsAggregationChart



Purpose: Draw a bar chart based on the historical data of the Tag. When the Data is multiple tags, it is partitioned by the Tag to show the value to be displayed. When it is one Tag, it is partitioned by the value of the Tag (the Tag value needs to be recorded in order to be used in this component, set as shown in the Figure below)

Specific Properties:

Same as TagsPieChartPeriod

4. Dashboard Designer Function



Provide users with tools to create Dashboard

The Functions have

- a. Save
- b. Execution

- c. Save New File
- d. Open Error Prompt Window
- e. Delete
- f. Clear All
- g. Place to Top
- h. Place to Bottom
- i. Keep Down
- j. Keep Up
- k. Keep Left
- I. Keep Right
- m. Center Horizontally
- n. Center Vertically
- o. Lock Component
- p. Select Components and Change to the Same Size
- q. Restore
- r. Cancel Restore
- s. Copy
- t. Paste
- u. Search and Replace
- v. Modify Project Password
- w. Add Sub-window
- x. Check program syntax and automatic typesetting
- y. Deployment
- z. Set up connection DB
- 4.1 Deployment

🔒 🎟 | ← ← 📲 🕒 | . 💡 | 💷 | 🔠 📑 |

After the Dashboard editing the project is complete, Dashboard Console can be deployed to multiple computers. Production line Management Personnel can see the production line information without going to the production line, or they can see the complete Dashboard on their own computer. Deployment Screen is as follows.



- ① Destination IP: The IP to be deployed to Dashboard Console
- 2 Port: The Port to be deployed to Dashboard Console
- ③ Timeout: Set the connection Timeout between Builder and Dashboard Console
- Attached: Add attached files. When building Dashboard, if external files such as graphics files are used, the attached files need to be delivered to Dashboard Console
- (5) Remove: Remove the attached files that have been added
- 6 Automatic claw attached file: Automatically add files, the system automatically searches for external files used by Dashboard and adds them to attached files
- ⑦ Attached Files List
- 8 Auto Run: Decide whether to set Auto Run when the Dashboard Console is opened
- (9) Execute Immediately: Execute Immediately after Deployment
- Deploy: Deploy Project and Attached Files
- 4.2 Advance Setting



The main function of Advance Setting is to set DB and voice function settings

4.2.1 Selection of DB

To save a project you may select the DB used by Builder later for this project, or the DB set by Dashboard Console (the DB set when the Dashboard was installed). The project may be built on the computer nearby, and use the computer nearby to build the DB. In case of deploying project to the production line, since different MainDB is used, therefore, the Prog used by the project must be selected when save the project.



The relationship between Dashboard Console and Builder to DB is shown as in the Figure above

4.2.2 Voice Settings

If the Script language command "systemVoice" is used, it must be set here

The settings provided are

- ♦ Play volume
- ♦ Play speed
- ♦ Play language

3.5.5 Script Language

3.5.5.1 Program structure

Script Language is composed of 6 program structures

define: variable declaration procedure

main: the initial starting point of the program

close: the final ending procedure of the program

dashboarderror: the procedure will be automatically triggered when an error occurs in the program

tmmserviceerror: a procedure will be automatically triggered if an error occurs in the action executed by the TMM Service

function: custom procedure (subprogram)

1 define:

What is Variable: During the operation of the program, a lot of data needs to be calculated, also a lot of data needs to be stored. The place where the data is placed, is called Variable, in Script Language.

Declared variables are all global variables, which can be used in various procedures and influence each other.

Reference Example DOC_GlobalVariable.prog.

2 main:

The entry point of the program, the program running starts from here

Reference Example DOC_main.prog.

③ close:

The last procedure will be executed after the program ends. For example: close all open connections in the program or clear unnecessary data. Do not control or set ui components in this program

dashboarderror:

If an error occurs during the operation of the program, the system will automatically trigger this procedure

Reference Example DOC main.prog.

(4) tmmservieerror:

If the program has an action to be executed by the TMM Service but an error occurs in the TMM Service, the system will automatically trigger this procedure

function: Name

For custom procedure, such as subprogram, a Name must be customized during declaration to represent the name of the subprogram.

Refer to Projects\Demo_EN in the installation directory for the actual sample program,

3.5.5.2 Program Rules

A procedure needs to be enclosed with { } symbols. Terms like if, else, switch, while, for, etc. also need to be enclosed with { } symbols.
 Reference Example: DOC ProcessControl.prog.



IMPORTANT:

Do not leave the execution statement of the while loop empty. Otherwise, the while loop will exit directly.

② For each line of program description, line needs to be skipped between lines, and cannot have two programs written in the same line

For Example:

```
if (_orderList->Length < _displayLimit)
{
    _displayLimit = _orderList->Length
}
else
{
    _displayLimit = 5
}
```

```
if (_orderList->Length < _displayLimit)
{
    _displayLimit = _orderList->Length
}else
{
    _displayLimit = 5
}
```

③ The commands or variables or setting values or Operation Symbols described in the program must be separated by blanks

Example: int I = 0

messagebox 🔪 message

 If the content of the set value contains special symbols or reserved words or displays blank, it must be enclosed in double quotation marks

Example: messagebox "Displayed Value of + Sign "

The setting values of blank and + sign in the middle of the displayed "and" must be enclosed in double quotes

Reference Example: DOC LanguageRules.prog.

(5) When setting the program operation calculation, the form of both sides of the "=" sign must be consistent, except for the following cases string type = can be int type, float type, double type, string type double type = can be int type, float type, double type float type = can be int type, float type, double type
 float type = can be int type, float type
 For Example:
 define:

```
{

int i = 0

float f = 0

string s = empty

}

main:

\{i = 5

f = i

f = 7.54

s = "ABC"

}

TM Operator Platform User Manual
```

nate

227

int type float type

f = s

float type string type

 Only one conditional symbol (&&, ||) can exist at the same time For Example:

Correct way

if (A==B && A==C && A==D)

Conditional judgment only exists at the same time &&

if (A==B || A==C || A==D)

Conditional judgment only exists at the same time ||

Incorrect Way

if (A==B && A==C|| A==D)

&& and || conditional judgment cannot exist at the same time

Conditional judgments and expressions do not support "(parentheses)" symbol judgments

For Example:

s =1+(2-3)*6

- (8) Variable name and function name cannot be duplicated. In addition, if you use Device, Thing, Monitor, Action, the name cannot be duplicated, and the name cannot be named with numbers or special symbols during naming
- (9) The use of the command must conform to the syntax of the command

For Example:

List when using GetValue to get the value of a specific field, it is necessary to add -> [Value]

function:exec	ute_robot_test	
_orderList->		
-}	AddValue	To obtain the data of the array or data table, you need to use the extended attribute (PS: if you do not add -> [] extended attribute, the table returns all values, the line and the line are separated by the 🖓 symbol).
	ChangeValue	Example 1: listaray->GetValue->[column, line (if the line is omitted, the entire column data will be returned, there will be // separated between lines)]
	DelValue	Example 2: datable>/SetValue>/Ecolumn, row (If the line is omitted, the entire column of data will be returned, there will be '; separated between lines)]
a Code Editor	FindIndex	and the second se
- 亍 錯誤提示	GetValue	
	Length	
	Target	

(1) The representation of the array is separated by the "," symbol

```
For Example:
define:
{
array a = string, int, string
}
main:
{
```



- / Divide
- % Remainder
- + Add
- Substract

+= Add the set value on the right to the original value on the left and put it back on the left For Example:

i = 5

The value of i +=4//i is 9

-= Subtract the set value on the right from the original value on the left and put it back on the left

*= Multiply the set value on the right by the original value on the left and put it back on the left

/= Divide the set value on the right by the original value on the left and put it back on the left

- (2) Comparison Operation Symbols
 - > More than
 - < Less than
 - >= More than or Equal to
 - <= Less than or Equal to
 - == Equal to

i= Not Equal to

3 Logical Operation Symbols

&& and(and)

|| or(or)

Reference Example DOC_Conditional.prog

3.5.5.4 Data Type

1 int

Signed 32-bit integer

Range -2147483648 to +2147483647

Syntax: int variable name = initial value

2 float

Value with 7-digit decimal point

Range ±1.5 x 10⁻⁴⁵ to ±3.4 x 10³⁸

Syntax: float variable name = initial value

3 double

Range ±5.0 × 10⁻³²⁴ to ±1.7 × 10³⁰⁸

Double floating point value

Syntax: double variable name = initial value

```
(4) string
```

Can store any value. Special symbols and defined names must be marked with double quotes

For Example:

define:

```
{
```

```
string s1 = ABC
```

string s2 = "Text of ABC + 123"

If put s2 into ui display, it will display as follows

```
Text of ABC + 123
```

string s3 = "ABC"+123

Display as follows

ABC123

}

Syntax: string variable name = initial value

5 bool

Store the value of true or false, true can also be represented by 1, false can also be represented by 0

Syntax: bool variable name = initial value

6 array

One-dimensional array, the array can be of different types with the length of the array fixed. When the array is converted to text, it will use the "," symbol to separate For Example:

define:

{

array a = int, string, float, string

// represents that a has 4 array values available for use, and each array value can be stored separately

```
[0] = int type value
```

[1] = string type value

[2] = float type value

[3] = string type value

}

Syntax: array variable name = data type... (the number of arrays, the data type declares the same number)

List One-dimensional array

The values in the array are of the same type, and the length of the array is not fixed For Example: define: { list a = strong // The values stored in a array are all string types } main: { $a \rightarrow AddValue = 1, 3, 5$ //represents a[0]=1 a[1]=3 a[2]=5 a→AddValue =100 · 200 // represents that 100 will be placed into a[3]=100, a[4]=200 } Syntax: list variable name=string ⑦ listarray two-dimensional array

The number of columns is fixed, the number of rows is not fixed, and can place different types of values

Syntax: listarray variable name = data type(How many number of columns are needed, the data type declares the corresponding number)

For Example: There are 3 columns

define:

1

```
{

listarray a = int, int, int

}

main:

{

a→AddValue = 1, 2, 3, 4, 5, 6, 7

// value will be placed as follows

a[0][0]=1 a[0][1]=2 a[0][2]=3

a[1][0]=4 a[1][1]=5 a[1][2]=6

a[2][0]=7

Table Notation
```

2

3



The system will generate 3 rows and 3 columns

If executing the following program

a→AddValue =100 · 200 · 300 · 400

Table Notation

1	2	3
4	5	6
7	100	200
300	400	
		1 1

Blue is generated in the previous row

a->AddValue = 1, 2, 3, 4, 5, 6, 7

The generated red is generated in the next column

}

⑧ file

The type of device generating the file

Syntax: file variable name = file (the file path can be an absolute path or a relative path) Example automatically)

_file->WriteLines = "TEST" txt Write Text

(9) file path

Type of device that generated the catalog

Syntax: file path variable name = directory path (path can be described as a pair path or

a relative path)

Example

filepath _ filepath = E:\filepath_Demo Generate path

_filepath->CreatePath Create path

10 datetime

Declare variable as datetime type

Syntax: datetime variable name = format of the declared datetime

Example

datetime _datetime = "yyyy/MM/dd HH:mm"

_datetime->GetNow

(1) datetable

Declare variables as data table type

Syntax: Reference Example

// The string generated by taken from the Device or Thing connecting to DB

- Thing Setting -			TMM Dashboard
Thing Name	FailQty		Within Range
TMM Service	TMMServ	ce1 V Capture Mode Service Loop Time V	
Thing Type	SQL Varia	ible Value Type int V	Browse
Group Name	Productio	nStatus Set Group	Clear
Description			Range
Save Mode			0 < Thing ,
SQL Variable —			
SQL Variab	le	Filter Convert	Calculation and
SQL Connection	String e	125401-2a12-4a68-aaeb-a58dcd718a20	^
SQL Language	S 7	ELECT COUNT(*) AS FailQty FROM ManufacturingLog WHERE CurrentOrderNo*/ AND InspectionResult = 0	OrderNo =

datatable dbs = "107000fd-488f-4012-b46f-09a6ca3b10da"

dbs->SetRemoteData = "SELECT *"

dbs->GetRemoteData = " INSERT INTO"

12 stopwatch

Declare variable as stopwatch timer device

Syntax: Reference Example

stopwatch _stopwatch = empty

_stopwatch->Start Start execution

_stopwatch->Stop Stop counting time </33062

Label1->Text = _stopwatch->GetSeconds

13 timer

Declare a timer device is generated

After the timer is started, the system will repeat execution function according to time [Note:=]

Syntax: Reference Example

timer _ timer = Flush, 2000

_timer->OpenStart Execution

timer->CloseClose

function:Flush

```
Label1->Text += "!"
```

}

{

CAUTION: The timer in use needs to be closed, otherwise it will continue to execute

```
2 define: //Variable declaration area
 3 ⊟{
 4
       timer _ timer = your_function,1000
 <sup>4</sup> _}
 6
7
    main: //Program start point
8 📮 {
       _timer->Open
9
10 L}
11
    close: //Program execution point before the project leaves
12
L3 🖵 {
       _timer->Close
L4
15 L
۱6
L7
    dashboarderror: //Program execution point after system error
L8 🖵 {
21
    tmmserviceerror: //TMM Service return error
22
23 ₽{
       _timer->Close
24
25 L}
```

thread (is the same as timer which can be executed repeatedly, and can adjust the number and weight of error execution)

Declare a multitasking executor is generated

Syntax: Reference Example

thread _thread = Flush, 1000, 0, 0

_thread ->OpenStart Execution

_thread ->CloseClose

function:Flush

{

Label1->Text += "!"

CAUTION: The thread in use needs to be closed, otherwise it will continue to execute

```
2
    define: //Variable declaration area
 3 🕀 {
 4
        thread _thread = your_function,1000,0,0
   L}
5
 6
7
     main: //Program start point
8 📮 {
9
        _thread->Open
10 L}
11
     close: //Program execution point before the project leaves
12
L3 🗏 {
        _thread->Close
14
15 L}
16
L7
     dashboarderror: //Program execution point after system error
L8 🖵 {
19
       _thread->Close
20 L}
21
    tmmserviceerror: //TMM Service return error
22
23 🕀 {
        _thread->Close
24
   L
25
```

(15) serialport

}

Create a serial port interface device

Syntax: serialport variable name = port number, rate, parity check bit, data bit, stop bit [, Timeout, Handshake Protocol, Event, Reading Mode, End Character, Waiting Time] (The parameters in PS : [] can be Omitted)

Example:

```
define : //Variable declaration area
```

```
{
 serialport sp = COM12 · 9600 · n · 8 · 1
}
main://Program start point
{
 sp->Target = reciver
                             //Define the procedure in the received value
}
function:Send
{
 sp->Open
                       //Equipment connection
 if (sp->IsWork == true) //Check whether the connection is successful
 {
   sp->Send = 2 · ABC · hexLine
 }
}
```

function:reciver

```
{
Label1 = sp->Text
}
```

(16) network

Create a network Socket interface device

Syntax: Reference Example

```
network _Socket = 127.0.0.1, 20108, 2, 2
```

_Socket->Open

_Socket->*Target* = Read_Data_Analysis Continuously trigger actions when the value is received

1 modbusRtu

Create a modbusRtu interface device

Syntax: modbusRtu = Interface number, transfer rate, parity bit, data bit, stop bit, Timeout

```
define://Variable declaration area
```

```
{
modbusRtu rtu = COM1 · 9600 · n · 8 · 1 · 2000
}
```

```
function:Send
```

```
{
```

{

}

}

```
rtu->Open //Equipment connection
if (rtu->IsWork == true) //Check whether the connection is successful
```

```
rtu->WriteSingleCoil = 1 · true · 1
```

```
Ŭ
```

```
(18) modbustcp creates a modbus Tco interface device
```

Syntax: Reference Example

```
modbustcp thermometerModbus = 192.168.133.182, 502, 1, 6000, 3, 1000, Error
thermometerModbus->Open
//Read Position->[Initial Position, Length]
lbCO2->Text = thermometerModbus->ReadHoldingRegistersToInt16->[0, 1]
```

// Write Position ->[Write Position, Synchronization, Write Value]

thermometerModbus ->WriteSingleRegister = 0, false, 0

(19) importlibrary

```
Create a plug-in DLL program device

Syntax: Reference Example

importlibrary variable name = file path

define://Variable declaration area

{

importlibrary lib = "c:\test\lib\test.dll"

}

main://Program start point

{

lib->Open

lib->RunMethod = Class1 · MainName · ABC

}
```

3.5.5.5 Properties

The function of each data type has a different purpose. When we want to use it more advanced, we will need different commands to get or set the value of the type, which we call the property of the variable type.

When we want to use property, we only need to add "->" after the variable name. The symbol system will automatically bring out the available property command names of the variable type.

As shown in the Figure below

```
For Example:
Get the length of the string variable value
define:
{
string s =empty
}
main
{
s ="ABCDEFG"
Lable1->text = s->Length
}
The screen after execution is as follows
```



3.5.5.6 Command Introduction

You can stop the mouse pointer in the program section, click the right mouse button, and the following prompt screen will appear

V)	工具	m	說明(H)	選項(0)											
		 I≣]			-		A		ଝା	_=						
_J ce		Thing	Ň	/onitor	Acti	on	Group	SFC	Toolbox	Property						
Edito	or C:\	Progra	am Files (×8 ×	Devic	e_Form	n Thing	g_Form ← → ■ 「	Monitor_Fo	rm	Actions_Form					
_			- A U						≞ ◄ ¥ ⊑	4 (🏔 🔛	E t ₁					
Guid	le:				•	X 4) 🖺 🖂 🗗	(2 ₽ #	я -							
1	defi	ine:n	ame//Va	ariable	e declar	ation	area									
3 E	3{	strin		moty												
5 1	-}															
6 7	mair	n:nam	e//Prog	gram st	tart poi	nt										
8 E	٦{		ABCDEEG													
0	. [(Camman	d Index	•		if									
2	-}	١	Variable	Index	•		else	the second state								
3	cle		Compon	ent Inde	• ×		switch	條件判斷 當條件式成1	Z時,則執行陳述	句一,如果有	宣告else時不成立	立就執行陳	述句二			
5							case	條件式可以補	そうちょう 可加	表或者 &&	&表而且, 需注意:	複合式宣告	告 ,&&只能同日	寺存在一種	-	
6 L 7	-}						elsecase	_{範191} . if (條件式)								
8							tor	(唐述句一								
9	erro	oraft	er:name	://Prog	gram exe		UlComponent	}								
1							UIComponent	else {								
2 ^L 3	-}						systemVoice	陳述句二								
							exit	}								
							clickUIName									
							mouseLeaveU	Name								
							mouseEnterUI	Name								
							call									
							callback									
							callproject									
							callprocess									
							writelog									
							messagebox									
							return									
							break									
							getUserGroup									
							getUserName									
							getTMServicel	sWork								
							dialogResult									
I Edi	itor C	Code Ed	itor				triggerSystem	rror								
尋與	置換						////note>									
							////senanote									

You can use it according to the function of the command and the syntax description

3.5.5.7 System reserved characters

- 1 empty represents ""
- 2 hex ReturnLine represents \r\n
- 3 hex LimeReturn represents \n\r
- ④ space represents "_"
- 5 hex Return represents \r
- 6 hex Line represents \n
- ⑦ hex Tab represents \t
- 8 click UIName represents to get the Name of the pressed button
- (9) mouse Enter UIName represents to get the Name of the component with mouseEnter property
- (1) mouse Leave UIName represents to get the Name of the component with mouseLeave property

Reference Example DOC_ReservedWords.prog

3.6 Dashboard Console

3.6.1 Dashboard Advance Confirmation

Deployed Prog Resolution Size



Whether the current computer monitor supports Prog display size

Font display adjustment is 100%

Dashboard Purpose Introduction & Resolution

The Dashboard is used to display the Dashboard project built by the Builder. The size of the Dashboard display depends on the screen resolution, therefore "Preparation" is a very important part, otherwise it will not be able to fully present the designed Dashboard.

The diagram below is the Schematic Diagram. When the resolution is normal, it can be displayed normally. Both of the display resolution of Dashboard project and Dashboard Container are 1920*1080.



The diagram below is the Schematic Diagram. When the resolutions are all different, the display will be incomplete. The display resolution of the Dashboard project is 1920*1080, and the display resolution of the Dashboard Container is 1366*768.



The above two photos show that when the resolution is inconsistent, it will result in some data or components being displayed abnormally.

3.6.2 Dashboard Console Function Explanation

Click the desktop executable file to open the Dashboard Console. The functions above are described as follows.



Dashboard 清單	
	The icon in the upper left corner is the icon pre-displayed
	by Dashboard Console. The system defaults to the icon of
	Dashboard Console. The icon can be replaced.
– ō ×	When fold the Dashboard Console, expand the
	Dashboard Console, close the Dashboard Console is
	closed, and fold the Dashboard Console, the icon of the
	Dashboard Console will be displayed in the lower toolbar
	- # A 当 <u>@</u>] [() () () () () () () () () () () () () () ()
	Press [Alt] on the keyboard to hide the upper window bar
	The name of the project selected in the execution list, and
	a warning will be displayed if it has been executed
	Set Dashboard Container Detailed Parameters

Click "Dashboard Detailed Parameters Setting" can enter the parameter options respectively.

Dashboard Settings			
Set Password Change Bar Color Connection Settings	Upload Image Reset	Delete projects	v Delete
IP 192.168.132.142 Socket 連線等符時間 Projects List Settings	Port 5433	MainD6 嬰更	
自動執行鴻葉	✓ 推加 副除 執行應用程式		制度 堆加 制除

Set Password	Set a Password that can enter the detailed parameter settings of the
	Dashboard
Change Bar Color	Change the color of the Dashboard window, the system default is black
Upload Image	Change the icon in the Dashboard window bar, the system default is the
	Dashboard Console icon
Delete Projects	Delete Projects
IP	Display the IP address where Dashboard Container is currently installed
Port	Display the opened channels of Dashboard Container
Dynamic IP	The checked condition represents that the IP and Port are determined by
	the system. If canceled, then it can be input manually. The default is
	checked.
Socket connection	The Connection time with TMM Service
waiting time	
Auto Update	The version of Dashboard Container can be automatically updated, and the
	default is not checked (when TMM Service has been updated with a new
	version, if Dashboard Container is checked, it will be automatically
	updated)
Automatic Execution	When deploying through the Builder, if Automatic Execution is selected,
List	then the Automatic Executed Prog can be seen in the bottom of the
	drop-down list. If not selected, it can also be added and deleted through the
	list
Execute App	This is the setting of other non-TMM procedures that need to be
	additionally executed when the Dashboard Container is executed

MainDB Change	The operation method is the same as the DB connection method selected by Builder, which is mainly applied to the problem of DB adjustment and redeployment
Save	After setting the parameters, Save must be clicked and restart the Dashboard Container to take effect

3.6.2.1 Dashboard Deployment Teaching

Open TMM Builder to select the project to be deployed, and confirm the deployment of past "IP address", "file" and "automatic execution".



After the deployment is successful, the Dashboard Container will display the projects that have just been deployed.



Assuming an automatic execution option is checked, open "Dashboard Container Detailed Parameter Setting" will see the automatic execution options in the list.

Dashboard Setting			
IP 192.168.132.92	Port 5433		MainDB
☑ 動態IP			變更
Socket 連線等待時間	2000 毫秒	🗹 Auto Update	
自動執行清單		✓ 増加	刪除
PLC_DOME_Final_Version			
執行應用程式		测路 横加	DI PA
		御見 相加	
			儲存
		Version : 4.	03.0101

3.6.2.2 Dashboard Deployment Warning

If the Dashboard Container displays a warning signal when entering in the deployment stage, represents that the project version does not match the current Dashboard Container version.



Tool Version is 4.03.0101 Project is 4.02.0008

Warning

The version of prog and TMM are different, it may cause some unexpected errors. You can click Stop button to terminate or to start automatically after finishing the countdown.



For example, if the new version of the Dashboard Container has an updated Script Language, and the Dashboard language project edited by the old version is used, Complier will cause errors. As shown in the Figure below, procedure adjustments are required.



3.6.2.3 Dashboard Container Connection Interrupted

If the connection between TMM Service and Dashboard Container is interrupted while the Dashboard is running, the warning window shown below will pop up. The Dashboard Container will try to connect to TMM Service every 5 seconds. If the TM Service and Dashboard Container are connected for the first time, the Timeout is 60 Seconds. If the connection is disconnected midway after the connection is executed, then there is no Timeout, and it will continue to try to connect.

Excution Time 3s	
Execution Time 05	Excution Time 3s

3.7 TMflow

3.7.1 TMflow's role

TMflow is a graphical human-machine interface, its purpose is to provide users with a complete, convenient and simple robot motion and logic programming environment. Through the graphical human-machine interface, users can simply manage and set the robot arm parameters, and plan the robot movement and flow logic with a graphical flowchart. The interface design of TMflow also takes into account the operating habits of the touch screen, allowing users to manage multiple Robots with a Windows tablet. TM Robot users and system integrators must read and fully understand the contents of this chapter before using this robot. In addition, when users follow this Manual to perform any operation of the robot, it is necessary to first read and follow the Safety Manual corresponding to the product software and hardware version, and the Hardware Installation Manual of the corresponding hardware version before operation. Refer to the Software Manual TMflow for detailed use.

TMflow can also be linked with the Dashboard. The Dashboard can activate TMflow in passive mode through arguments. When the user operates the Dashboard, the TMflow Client will run in the background and automatically synchronize the Robot to obtain/write the variables, base and other parameters of the Robot. If there is a need to call the TMflow function on a specific page of the Dashboard (for example, modify the TCP parameters in the TMflow program), the developer can bounce TMflow from the background to the main screen through the Builder, and can allow the user to use the existing TMflow UI resource (designated page) on the TMflow Client for programming without having to create a specific UI; when the user set up is complete, click the upper left button, TMflow in passive mode will hide it in the background again. The following is the example of Dashboard using TMflow Client. For detailed functions, refer to the Chapter Appendix A: TMflow Command.



3.8 TMstudio

3.8.1 TMstudio's role

TMstudio is the offline simulation software of TM Robot. The software has three modules: Workstation, TCP Generator and Palletizing Wizard. The Workstation enables users to load their own environment models in the scenario, and can cooperate with the Collision Check node of the TMflow process for collision detection, to complete the Operator development of 3D Random Picking. The TCP Generator assists users to set up the CAD drawing files of the end tool drawn by themselves to generate TCP modules and parameters to be used in TMflow. The Palletizing Wizard provides you with Palletizing Simulation in the scenario created in Workstation, and can generate layout files of TM Palletizing Operator. Refer to the Software Manual TMstudio for detailed operations of the Workstation and the TCP Generator. This chapter will use the Palletizing Wizard to create scenario and generate point files for teaching.

In addition, TMstudio can also be linked with the Dashboard. The Dashboard can activate TMstudio in passive mode through arguments. When users operate Dashboard, if it is required to use specific functions of TMstudio on a specific page of Dashboard, then open the designated tabs and designated files of TMstudio through opening the arguments. After TMstudio in passive mode completes operations such as TCP pages and Palletizing pages, it will bounce and close the window, allowing the users to return to the Dashboard page to complete a continuous operation. Refer to Appendix B: TMstudio Command.

3.8.2 TMstudio Palletizing Wizard and Scene creator, function and limitation The possible palletizing scenarios are listed as below.



The following will create a common left and right Palletizing scenario for example description.



IMPORTANT:

In TMstudio Palletizing Wizard, users can construct various poses to remain for the robot installation, and the output files come with the relationship between the robot base and the pallet base. For the applicability of each point and its motion type in the project, it requires users to adjust by the situations to apply to the custom-designed pallet applications.



Below lists the Workstation page of TMstudio.



Click File and click New to create a new scene. The new scene is named **PalletizingOperator** as shown below.

IMPORTAN

New Scene	×	(
Name	PalletizingOperator	
Path	C:\Techman Robot\TMstudio-V1.12.1800\scen	
	0進走 现/月	

3.8.2.1 Step 1. Import CAD File

Click Import above TMstudio, select CAD, and import the drawn 3D CAD drawing file into TMstudio, as shown below, we import the Palletizing_Stand



3.8.2.2 Step 2. Import Pillar Base

Then click Palletizing under Import, insert the special coordinate system Pillar Base, and select CAD File (the Example is Pillar) as the child component of Pillar Base
Select CAD file	×	Select CAD file	\times
Please select move part of pillar Select CAD file		Please select move part of pillar Select CAD file Pillar	
BACK		BACK NEXT	



NOTE:

In actual running, all objects under Pillar Base will be moved according to the setting item Pillar Height (the column height of each corresponding layer in the simulation).



NOTE:

If the object is not used as a child component of this base in this step, the child-parent relationship can also be changed later through importing the general CAD file.



IMPORTANT:

Only one pillar base can be inserted in this scenario.



IMPORTANT:

Users cannot use the external axis (track) with X or Y direction. Only as the pillar can the Z-direction go with the external axis.

3.8.2.3 Step 3. Set Pillar Base Parameters

After loading Pillar Base and CAD files, the parameters of the Pillar Base can be set. Assuming that the specification of the column is 500mm in length, and it takes 20s to rise from the bottom to the top, the speed is set to 25mm/s and the height is 500mm. The distance can be used as a threshold for calculating the moving height during simulation, and the speed parameter can be used for estimating time during simulation.

Linear Motion Track			×
AXIS	Z	•	
Speed(mm/s)	25		
Distance(mm)	500		
	1		
BACK		Do	ne



IMPORTANT:

The distance can be used as the maximum distance that the child component under Pillar Base can move during simulation.

The speed parameter can be used to estimate the overall running time during the simulation.

After importing, it is as shown in the Figure below



3.8.2.4 Step 4. Import Robot

Click Import Load robot to import the Robot used for Palletizing. If this application uses TM12, TM Operator Platform User Manual 254 then click TM12 to import, and click the List Alignment Assistant above will copy TM12 to the center of the CAD File under Pillar Base

Dialog	×
Move the coordinate origin to the target point	
Origin X-Axis Y-Axis Z-Axis	
Coordinate / Base TM12 Characteristics Points C Arc Center Middle of two points Middle of two arc center Moving item C Coordinate / Base Only C Coordinate / Base Child	
OK	

After the setting is completed, it is shown as below



3.8.2.5 Step 5. Import Tools

Click Attach above TMstudio, select the TCP file, and click Robot, then the Tool and the Robot will connect. Refer to the Instructions Software Manual TMstudio on the TCP Generator operation page for the TCP setting.





IMPORTANT:

The Tool loaded in the scenario will be changed to the default value when the user uses the Palletizing Wizard.

3.8.2.6 Step 6. Insert and align the pallet

In order to place the pallet in the corner of the Palletizing_Stand CAD file, first click New Coordinate to create a base;. Then use the Alignment Assistant to place the base Coord_0 at the corner of Palletizing_Stand.

_		
	Dialog	×
	Move the coordinate origin to the target point	
	Origin X-Axis Y-Axis Z-Axis	
	Characteristics	
	Points	
	C Arc Center	
	C Middle of two points	
	Moving item	
	Coordinate / Base Only	
	C Coordinate / Base Child	
	OK	

Click Import to add the special object "Pallet", set its size, and click OK to generate the pallet. The generated pallet can be set to distance of 0 from Coord_0 through Location to achieve the purpose of Coord_0 overlap



Note	NOTE : Click on the Tree Diagram target on the left, or directly click on the CAD Diagram in TMstudio, and the parameters can be set through right-click.
Note	NOTE: The setting range of the pallet is as follows Length: 1200 mm [setting range: 25~2000] Width: 800 mm [setting range: 25~2000] Height: 144 mm [setting range: 25~2000]
IMPORTANT	IMPORTANT : The pallet size set in the scenario will be changed to the default value when the user uses the Palletizing Wizard.
IMPORTANT	IMPORTANT: Only two pallets can be inserted in the scenario.

3.8.2.7 Step 7. Change the extended direction of pallet







IMPORTANT:

The initial position coordinates of the pallet set in the scenario, will become the extended direction in which the user changes the size of the pallet.

With this method, repeat the above steps to create pallet on the other side.



Users can also right click the CAD file to change the color to their favorite color.



3.8.2.8 Step 8. Set parent-child relationship

The relative parent-child relationship is as follows.



3.8.2.9 Step 9. Start palletizing

After the completion has been confirmed, switch to the Palletizing page to use the self-programmed scenario for palletizing.



NOTE:

Note

According to user-define scene, the TMstudio Palletizing Wizard can achieve the kind of five application as below

Support 5 various modes of stacking/unstacking

- 1. stacking the same cabinets
- 2. stacking the different cabinets: concurrently handle the stacking task of two different cabinets
- 3. unstacking the same cabinets: unstack the cabinets on the pallet
- 4. unstacking different cabinets: unstack different cabinets on different pallets
- 5. replacing the pallet and restacking the same cabinets: unstacking cabinets from pallet A
- to pallet B and the pallets are different specifications

And next, we will introduce how to use Palletizing Wizard to simulate and export point file.

3.8.3 Build your palletizing operator layout in TMstudio

The Palletizing Wizard is a solution for palletizing applications in TMstudio, which allows users to define their preferred palletizing mode. This wizard allows users to verify the feasibility of their palletizing with different tools, palletizing parameters, Box arrangement, etc. It also simulates and detects possible potential errors. The following is a brief introduction to this software.

3.8.3.1 Change Gripper

If a different gripper is required for use, click/tap "Change Tool" to select a suitable gripper.

Connect To	Scene: PalletizingOpe	erator
Select Robot	TM12	•
Change Tool Tool: Adjustable t	flexible gripper w	ith offset
BACK		Next

3.8.3.2 Choose Palletizing Type

Palletizing Wizard supports functions of Palletizing, Un-palletizing and Change Pallet. You can select the corresponding type according to your needs.

Please Select an Application Type	\times
Pallet	
·	
Depallet	
Change Pallet	
BACK Next	

3.8.3.3 Start Palletizing Setting

After selecting the corresponding pallet to be edited, click Edit to edit the pallet.

		Application: Pa	llet 🗵
	Pallet_1: 12	00mm X 800	mm X 144mm
		Layer: 0	
		Boxes: 0	
	[
	Pallet_0: 12	00mm X 800	mm X 144mm
		Layer: 0	
		Boxes: 0	
	Edit	Сору	Paste
,			
	BACK		Next

3.8.3.4 Step 1 - Set the Pallet size and boundary

In Step 1, the user can select the default pallet template to use through the drop-down menu (refer to the following table), or input the length (mm), width (mm) and height (mm) of the pallet to customize the pallet. After setting the size of the pallet, define the maximum range of boxes to be placed on the pallet through the boundary setting. Refer to the following schematic diagram:

The schematic diagram below shows the custom pallet settings.

Length: 1000 mm Width: 800 mm Height: 144 mm

	Choose/	Set Pallet	\times
Choose a Template: Customize Length(mm): Width(mm): Height(mm): Save As	 ▼ 1000 800 144 Delete 	Width(mn)	Length(mm) Height(mm)
Boundary (L)Pallet Left(mm): (R)Pallet Right(mm): (T)Pallet Top(mm): (B)Pallet Bottom(mm): Preview	0 0 0		
ВАСК			Next



NOTE:

The user can "Save New File" for the pallet parameters; or "Delete" the self defined pallet parameters

Template	Size
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)

3.8.3.5 Step 2 - Set the box size and label direction

In Step 2, the user can set the box length (mm), width (mm), height (mm), weight (kg) and label direction. In addition, the Z-direction spacing (dX/dY/dZ) can also be enabled

The following schematic diagram is based on Length: 200 mm Width: 370 mm Height: 250 mm

dx: 10 mm

dy: 10 mm

Setting Box Size and	d Label Direction	Setting Box Size an	d Label Direction
Customize Image: Customize L1-Length(mm): 200 L2-Width(mm): 370 Height(mm): 250 Weight(kg): 5 Save As Delete Clearances Setting dX(mm): 10 dY(mm): 10		Customize Image: Customize L1-Length(mm): 200 L2-Width(mm): 370 Height(mm): 250 Weight(kg): 5 Save As Delete Clearances Setting dX(mm): 10 dY(mm): 10	
☐ Enable Z Direction Clearances		 ✓ Enable 2 Direction Clearances dZ(mm): 1 *This value will apply to **** of user-defined layer of step3 	
BACK	Next	BACK	Next



NOTE:

The user can "Save New File" for the box parameters; or "Delete" the self defined pallet parameters



NOTE:

Enabling the Z-direction clearance to allow users to set tasks between layers, such as inserting partitions. When you activate the Z Direction clearance, you can insert the spacing between the layers in step 3.

3.8.3.6 Step 3 - Create layers and placement method

In step 3, the user can set the number of boxes and the arrangement method. The following will demonstrate how to create layers and how to arrange the boxes.

The following will take 4 layers as the example.

Click "Add Box" To lay out the position of boxes. Note that the dimensions of the box

and pallet are based on the settings in the previous step.

	Create	Pa	llet La	yers	\mathbf{X}
🗲 📑			Ŧ		
Layer No.	Style		Item	Location	
0	Pallet				
1	Box				
2	Box				
4	1.0mm				
5	Box				
6	1.0mm				
<u>/</u>	Box 1.0mm				
0	1.01111				
			<		>
	Deeet	1	,		
	Reset			🗌 Snap	\wedge \neq \rightarrow
					\bigcirc
	PACK			Λ	lovt
	BACK			N N	iext

Note

NOTE:

If you have checked and set the Z direction clearance in step 2, click , will insert your selected layer into the previously set clearance value.

Refer to the table below for more detailed information about the function icons. Users can display the function description corresponding to the icon through the lower right corner (?).

lcon	Function	lcon	Function			
*	Increase Box Layer	+	Add Box			
	Increase the clearance					
	in Z-axis direction	in the				
	Сору	Easter]	Change Box Label			
	Paste	6 6	Auto Layout			
	Clear	↑	Move Box			
Ī	Delete	Ī	Delete Box			

The user clicks the right mouse button on the box or presses the box on the screen to move the box, delete or change the label.



The user can use the Auto Layout button **a**, to select the type of Auto Layout according to their requirements.

Auto Layout						×
Layers]					
Odd layer						
14	16	12	10	14	12	9
Even layer						
			APPLY			

	NOTE: When odd layer and even layer are will be placed 180 degrees different f	, INVERT can be selected, then the even layer rom the odd layer.
ote	When odd layer and even layer are even layer will be placed 90 degrees	different from the odd layer.
	When odd layer and even layer are	, , or , Mirror can be selected, then

Use the mouse to click

-

or press the box and click to move the box, the user can move

these boxes according to preference



Option	Function Description
X, Y (mm)	The position of the box relative to the origin of the pallet. Click Apply to employ.
Box jog (mm)	Set the box jog, default is 10mm.
Rotate all	Rotate all boxes 90 degrees
boxes	
Center all	Place all boxes in Center Position (Center Position is defined by the Pallet
boxes	Boundary)
	Move the box in the direction of the arrow through the defined box jog distance
~	 If a single box is selected, click/tap this arrow to move the selected box
	according to the defined box jog distance
	• If no box is selected, click/tap this arrow to move all boxes according to the
	defined box jog distance
	Move the box according to the direction of the arrow so that it is close to the
N	boundary or the boxes previously placed according to the boundary setting (dx,
	dy)
	 If a single box is selected, then click/tap this arrow to move only the
	selected box.
	• If no box is selected, then click/tap this arrow to move only the selected
	box.

Snap can align the boundary or boxes to quickly complete your layout. You can enable this function by checking the Snap checkbox below. Refer to the following for the related functions and examples.

Option	Function Description
$\land \rightarrow$	After placement, it will be aligned upward, and then aligned to the right
\frown	After clicked, it will change to
\checkmark	After clicked, it will change to
	After clicked, it will change to
\rightarrow	After clicked, it will change to
	Switch left and right, such as $\land \rightarrow$, after clicked, it will change to $\rightarrow \land$
	Turn ON/OFF Snap Function





3.8.3.7 Step 4 - Create the Layer Placement Sequence

In **Step 4**, the user can create Palletizing Sequence according to the Palletizing set in Step 3. The following steps are used as an example of creating a Palletizing Sequence for the first time, demonstrate with "Automatic Sequence"

Create Palletizing Sequence								
				2 3				
Layer No.	Contents	Current He	ight(mm)	Current Weig	ht(kg)			
1	8 Boxes	394.00		40.00				
3	8 Boxes	645.00		80.50				
5	8 Boxes	896.00		121.00				
7	8 Boxes	1147.00		161.50				
					\bigcirc			
	BACK			Done				

"Automatic Sequence" allows the program to automatically set the sequence of box palletization. The numbers represent the sequence of box palletization.



For more detailed information about the function icons, see the table below.

The user can click 💓 to display the button corresponding functions.

lcon	Function	lcon	Function
	Copy Box Sequence	2 3	Automatic Sequence
	Affix Box Sequence	1	Increase Box Sequence
ſ	Previous Step		

3.8.3.8 Copy Box

After completing the pallet setting, it will return to the previous page, and the user can program another pallet one by one.



Or use copy and paste to complete the settings of pallets on both side, click Next to enter the

next step.	
Pallet X	
Choose a Template:	
EPAL_EUR(1200.0-800.0-144.0)	
Length(mm):	
Width(mm):	
Height(mm):	
(L)Pallet Left(mm):	
(R)Pallet Right(mm):	
(T)Pallet Top(mm):	
(B)Pallet Bottom(mm): -0	
Preview	
Rotate All Boxes 180	
ОК	

Note	

NOTE:

When copying the Pallet_0 box to Pallet_1, if it is rotated, in addition to the box placement position, the placement vector will also be rotated.



NOTE:

When copying the pallet, because the "Actual Boundary Size" is copied, there will not be problem of the box exceeding the boundary after copy and paste.

3.8.3.9 Simulation Page - Parameter Setting

	Palleti	zing Simula	tion 🔀
Project Speed	15%		Parameters Setting
Palletizing	Progress		
Pallet_1			Pallet_0
0/3	2 Tota	0/64	0/32
0%		0%	0%
Simu-Speed			Cycle Time
	Validate	Play	Stop
В	ACK		Export

Click **Parameter Setting can set** the Parameters of Palletizing. As shown in the figure below, the direction parameters can be input as parameters during simulation. Refer to the detailed description of related parameters below. Click Apply after setting is complete

Param	eters Sett	ing					×	Param	eters Sett	ing					×
Pallet	_0		Ŧ	·				Pallet	_1		•	·			
Robot	t Initial P	ose(R	obot		Contr	oller		Robo	t Initial P	ose(R	obot		Contr	oller	
x	355	Y	-155	z	720	mm		x	355	Y	-155	z	720	mm	
RX	180	RY	0	RZ	0	degree		RX	180	RY	0	RZ	0	degree	
Box P	osition							Box F	osition			,			
x	1000	Y	0	z	200	mm		x	1000	Y	0	z	200	mm	
RX	180	RY	0	RZ	0	degree		RX	180	RY	0	RZ	0	degree	
Elevat	Elevated height 200 mm Motion Setting				1	Eleva	ed heigh	t	200	mm	Mot	tion Setting			
Via Po	int Offse	t	x -	400	Y -40	0 z 0	mm	Via Po	int Offset	t	x 🔤	400	Y 40	0 z 0	mm
Place	ment Veo	tor-			Pillar			Place	ment Vec	tor			Pillar		
Box A	pproachi	ng Ve	ctor X, Y		Pillar H	eight of Each		Box A	pproachii	ng Ve	ctor X, Y		Pillar H	leight of Eacl	ı
					Floor	Distance Compe	nsation	I					Floor	Distance Cor	npensation
Box A	pproachi	ng Ve	ctor Z					Box A	pproachii	ng Ve	ctor Z				
30	0 mm				Dillor			30	0 mm				Pillar		
Elevat	ted heigh	nt befo	ore placir	na	0,0,0,0	0,0,224,224	mm	Eleva	ted heigh	t befo	ore placin	ng	0,0,0,	0,0,224,224	mm
20	mm							20) mm				<u> </u>		
,												_			
Safety	(5					Safet	·		5				
DE	FAULT			APF	PLY			DE	FAULT			APF	PLY		

Parameter	Description
Controller	Use the controller to teach the corresponding posture the robot before stacking or lifting the pillar

	Parameter	Description			
		up and down.			
Ro	bot Initial Pose (Robot Base)	The posture of the robot before stacking or lifting the pillar up and down.			
	Motion Setting	Set the motion of Elevated Height to Via Point Offset and Via Point Offset to Placement Vector to PTP or Line . (PTP recommended) (If the robot is not installed upright, the box may be tilted during the task. At this time, the motion path can be modified to Line to prevent the box from tilting.)			
	Box Position (A)	The position of the Robot gripping the box			
Get th	ne elevated height of the box (B)	After the robot grips the box, the elevated height along the Z-Axis			
	Relay point offset (C)	Get the corresponding position of the elevated height of the box			
Placement	The box approaching vector X, Y, Z (D)	The box approaching vector X, Y, Z Axes			
Vector	Lowered height of placing the box (E)	Height before placing the box			
Pillar	Pillar Height of Each Layer (Z-direction height is regarded as the Layer)	Set the Pillar Height corresponding to the left and right layers (This parameter is the absolute height of the Pillar, if set to 100, 300, it represents that when the Robot palletizes the first layer of boxes, the Pillar's absolute height is 100mm. When palletizing the second layer, the Pillar's absolute height is 300mm)			
	Plane Distance Compensation	If the user has requirements, the Plane Distance Compensation can be clicked. At this time, the robot will try to place the robot box with different column heights according to the parameters on the current page.			
	Safe Distance (mm)	If it is detected that the distance between the Robot and other objects is below this threshold, it will be considered as a collision.			



3.8.3.10 Verification

Through simulation, the potential errors that may occur during Palletizing/Un-palletizing/Change Pallet (such as point position cannot be reached, collisions in space, etc.), and estimate the total time required for the process. Click Verification. After the Verification is complete, click Play to simulate.



NOTE:

If a collision occurs during the simulation, the user can simulate the state before the collision through the Play button to facilitate the adjustment of Palletizing parameters

3.8.3.11 Simulation

Click Play to start the simulation.



NOTE:



- If the dongle is not connected, the "Export" function is not be available. The
 Palletizing simulation cycle time does not include the I/O waiting time and the time of
 the gripper gripping the box. The box gripping method for Palletizing simulation is for
 the TCP overlap with the center of the box.
- Motion simulated on TMstudio Palletizing Wizard is a combination of LINE motion (e.g.: picking/ placing) and PTP motion (e.g.: moving to via Point) with 100% of the Project speed, precise positioning but no Blending, while the actual motion is defined by the TMflow Project.

3.8.3.12 Export

After confirming that all the settings are correct, click "Export" to export the files required for Palletizing; click **"Export"** To output the files required by the Palletizing, and save the files to the following path:

Create on the USB device named TMROBOT: TM_Export\TMPLTZOP\TextFiles.

Export File	×		
File Name	Layout		
Infor	mation		
Box Size(LxHxW)	Pallet_0:200.0x250.0x370.0 Pallet_1:200.0x250.0x370.0		
Box Weight	Pallet_0:5.0		
Pallet Size(LxHxW)	Pallet_1:5.0 Pallet_0:1000.0x144.0x800.0		
Total Quantity	Pallet_1:1000.0x144.0x800.0		
Export Path:	🗌 Auto Save Project		
C:\Techman Robot\TMstudio-V	1.12.1800\TM_Export\TMstudio\TextF		
CANCEL	EXPORT		

3.8.3.13 Point File Format Description

The point file generated by TMstudio can be used with TMflow-related commands to read the document to complete the pick and place of the box. The corresponding function of the relevant document can be referred to as follows.

Name	Content		
Part_No	Part Number (Can be input when TMstudio file save/export point)		
Total Quantity	Total Quantity of Boxes		
date	The date when this point file was generated		
pallet_version	Version of this document format		
	control = 1; Palletizing		
pallet_contorl	control = 2; Un- Palletizing		
	control = 3; Change Pallet		
	BoxN_Info, box information corresponding to the Nth box		
	BoxN_Info [0]; L1-Length (mm)		
hox info	BoxN_Info [1]; L2-Width (mm)		
	BoxN_Info [2]; Height (mm)		
	BoxN_Info [3]; Weight (Kg)		
	BoxN_Info [4]; Label Direction (Top:0; Left:1; Bottom:2; Right: 3)		
box Elevated Height	boxN_Elevated_Height: Elevated Height corresponding to taking out		
	the Nth box (mm)		
Pallet_base_xyzabc	The relative relationship between the robot base and the pallet base		
Pollot info	Pallet parameters (name Pallet, will change with the name of the		
	scenario setting)		

TM Operator Platform User Manual

r	
	Pallet_info [0]; Length (mm)
	Pallet_info [1]; Width(mm)
	Pallet_info [2]; Height (mm)
	Pallet_info [3]; Weight (Kg) [This is a reserved item and cannot be
	set from TMstudio]
	Corresponding tasks of array corresponding to each layer of Pallet
Pallot lavor info	Pallet_layer_info = 1; this layer is the box layer
	Pallet_layer_info = 2: this layer is the Z-Axis gap
	Pallet_layer_info = 3; complete
	The quantity of boxes corresponding each layer
Pollot lover we num	Pallet_layer_wp_num [0]; Total quantity of boxes on the first layer
	Pallet_layer_wp_num [1]; Total quantity of boxes on the second layer
	Pallet_layer_wp_num [N]; Total quantity of boxes on the Nth layer
	Cumulative total quantity of boxes
	Pallet_layer_wp_num [0]; Total quantity of boxes on the first layer
Dellet lever up euro	Pallet_layer_wp_num [1]; Total quantity of boxes on the first to
Pallet_layer_wp_sum	second layer
	Pallet_layer_wp_num [N]; Total quantity of boxes from the first to Nth
	layer
	Column height corresponding to each layer of Pallet corresponding
Pallet_Pillar_Height	to the array (mm)
	[The height of the inserted Z-Axis will be regarded as a layer]
	Move in X, Y, Z direction relative to Elevated_Height position
Pallet Via Paint Offect	Pallet_Via_Point_Offset [0]; Relative movement toward X-Axis (mm)
	Pallet_Via_Point_Offset [1]; Relative movement toward Y-Axis (mm)
	Pallet_Via_Point_Offset [2]; Relative movement toward Z-Axis (mm)
	Relative to the position of Pallet_Elevated_Height_Before_Placing,
	move to the position from the position reached by
	Pallet_Via_Point_Offset
Pallet_Approach_XY_Z	Pallet_Via_Point_Offset[0], [1] ; Relative movement toward X,
	Y-Axis (mm)
	Pallet_Approach_XY_Z [2]; Reverse placing vector from the Z-Axis
	(mm)
Pallet_Elevated_Height_Before_Placing	Elevated Height before Placing box (mm)
	Pallet_point_wp_N: The position of the Nth box on the pallet
	Pallet_point_wp [0]; The relative relationship between the box and
	the X-Aaxis of the pallet
	Pallet_point_wp [1]; The relative relationship between the box and
	the Y-Axis of the pallet
	Pallet_point_wp [2]; The relative relationship between the box and
Pallet_point_wp	the Z-Axis of the pallet
	Pallet_point_wp [3]; The relative relationship between the box and
	the RX-Axis of the pallet
	Pallet_point_wp [4]; The relative relationship between the box and
	the RY-Axis of the pallet
	Pallet_point_wp [6]; The relative relationship between the box and
	the RZ-Axis of the pallet
Pallet_set_wp	Pallet_set_wp_N: Setting of the Nth box on the pallet

TM Operator Platform User Manual





NOTE:

All parameters in the Notepad documents can refer to the contents and icons of 3.8.3.9Simulation Page - Parameter Setting.

3.8.4 Set the project as Palletizing Wizard default project

If you want your hardware as the default project on the Palletizing Wizard page, you can modify the information in ConfigPallet.txt to your Palletizing project in the document in the TMstudio folder. (This document can be imported and exported via the button in the upper right corner of TMstudio).

📷 avuur-55.uii			20	
🔮 ConfigFoot.xml			202	20/1/8
🖭 ConfiaHead.xml			202	20/1/8
ConfigPallet.txt			202	20/6/3
S Freelmann dll			201	15/0/9
🗟 Freelma 🗐 Con	figPallet.t	xt - 記事本		
🗟 freetyp 檔案(F)	編輯(E)	格式(O)	檢視(V)	說明
🗟 gl2ps.d TM Pal	letizin	g OP.xm	ıl	
lterLog.				
🗋 JointLir				

If there is no pallet in the scenario or there is no scenario, then switch to Palletizing Wizard and click Wizard. The following window will pop up. Click OK to open the default scenario.



IMPORTANT

IMPORTANT:

When the Palletizing Wizard project file is opened in the way of default scenario, Palletizing Wizard will read.

All setting parameters in XML, such as pallet size, box size, and placement vector as the default values, hence, the developers can determine their own hardware default values.



IMPORTANT:

When the Palletizing Wizard project file is opened under the default scenario, the boxes on the pallet and their placement order will be cleared.

4. Common Techniques

In this chapter, several common technique (CTQ) of building the Operator UI using TM Operator Suite are introduced.

4.1 CTQ1 : Create a Operator UI

To build the Operator UI, a component named **TabControl** is introduced:

TabControl for Page Switching



Use TabControl for pages.

Select different TabPage: TabPageX to jump to specific page as below.

TabControl Example: Page Switching

- Eccel Sharloog Lenghan M	● X Tabley: Tabley: 2: 10 1		Properties Tablage Tablage6 Tablage Tablage29 Tablage Tablage3
TM®perator Palletizing Operator Example - Powered by TM Operator F	Aucher Top, Left Deurgins Sold Brane Solder Brane Solgen Songel Aufschaften Brane Tuffshaften Brane Korennalty	Select Palletizing Type and Go to Layout Editor	Tablege Tablege, NO Tablege Tablege, NO Tablege Tablege, NA Tablege Tablege, NA Tablege Tablege, NG Tablege Tablege, NO Tablege Tablege, NO Tableg
Select Project to Work	Bale Advanced Betting Advanced Betting Belleville State (State State Sta	A A B	TextBox: TextBox: En1 TextBox: TextBox: Rev TextBox: CodBox; Text TextBox: CodBox; Rev TextBox: CodBox; Text TextBox: CodBox; Text TextBox: CodBox; Text TextBox: CodBox; Text TextBox: CodBox; TextBox TextBox: CodBox; TextBox TextBox: CodBox; TextBox TextBox: CodBox; TextBox TextBox: CodBox; TextBox TextBox: CodBox; TextBox; TextBox TextBox: CodBox; TextBox; Tex
Project Information Part No. Part No. Part No. Soc Size(LWWW) Box Size(LWWW)	finicidar Consultar Lett 4 Bar UTZ, SRT Stat Esthapt Paulo 7 Trub Austor	Description	ForColor Left Size Text Top Yuible
and Constant	Defau de rendeur e de a de la Catacit aut de reper Catacit, de specifiel et gravell materia e titel datace TabPage1	Elizer Gobilian	Ascher Define the contributer of voll maintain a fixed for Page 6

TabControl in component.

In function Next in Code, TabControl is switched to different page and not in sequence:





IMPORTANT: Page Management in TabControl

Create TabPage at Beginning

The name of each TabPage(e.g., TabPage10) is generated automatically and could NOT be modified after the TabPage is created. Plan empty TabPage and create them at beginning of designing phase will be easier for page management.

Add/Remove Tab in TabControl

Click **TabControl Tasks** to add/remove tab if needed. <u>It's **NOT** necessary to modify the</u> **TabControl** in this project.



It's not recommended that add/remove TabPage during designing phase of **UI**. A negative example of removing TabPage that causes missing of **TabPage7**:



TabPage System

Page Planning before Implementation

It's recommended to plan the operation flow at beginning since the sequences of pages in **Dashboard** are complicated.

4.2 CTQ2 : Login, Logout and Access Control of Robot

To control the robot, perform "Get Control" on the robot first is necessary. There are some examples.

<u>Login</u>

Use following code to login and check the connection on target robot.



Get Control

Make sure that the target robot is free and no one control it.

```
function:btn_GetControl
{
    OperatorActiveX1->GetControl = true
}
```

Logout

Logout if you want to release the control on target robot.

```
function:btn_LogOut
```

```
OperatorActiveX1->LogOut = true
```

Robot Status Monitoring

The robot status could be monitored with timer function.

First, a timer is created with 100ms interval:

```
timer _tmr1 = CallTMR1,100
//...
function:CallTMR1
{
    call SubWindows_UpdateRBStatus
}
function:Main
{
    _tmr1->Open //Start timer
}
```

Connects checkbox component with the status:

```
function:SubWindows_UpdateRBStatus
{
    CheckBox7->Checked = OperatorActiveX1->IsRobotLink
    CheckBox6->Checked = OperatorActiveX1->IsControlled
    CheckBox5->Checked = OperatorActiveX1->IsProjectRunning
}
```

As a result, a panel is created to monitor the status:

Call TMFlow-S		
Close TMflow	Login	GetControl
9		
Show/Hide TMFlow	Logoff	Test area
	Call TMFlow-S Close TMflow 99 Show/Hide TMFlow	Call TMFlow-S Close TMflow G Show/Hide TMFlow Logoff

An example of a robot status monitor

4.3 CTQ3 : Embedded Tutorial Video

Using embedded tutorial video in UI can instruct users how to do, which is better than just text and graphics. Following steps will show how to embed video in UI.

Step 1. Drag and drop "AxwindowsMediaPlayer" button form toolbox.



Location of a component of media player

Step 2. Press the "..." button of URL layer, and choose a media file(*.GIF/*.MP4).



Properties of the media player

Step 3. If you want to remove the menu bar under the video, type "none" behind "uiMode".

Bui Vier D	ilder w(V) To evices	Dol(T) Help(H)	Options Tags	s(O) Monitors	Actions	Groups	SFC	, Ø Management		Service Engine Statu	s TMMSe	rvice1	-	Display	×
Toologies States	box hop Floor vivice Control and Control Pointery abel utiton abel utiton abel utiton abel onbobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combobox abel combox abel combobox abel combobox abel combobox abel combox abel	av n MediaPhyer er	μ x	Editor Ope	rator 0814.3	Le	t's cre	eate a n	ew work	piece.	A A	Properties AstWindowMediaPls AstWindowMediaPls Description Anchor Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description Description	Top, Left None True XwindowsMedi True True True True Cone Viewource(VID) 0 2255, 255, 2 PMingLit, 9, 94 105, 105, 11 304 269, 215 e mode (the mode can one, Invisible)	p Player6 iaPlayer6 30%Step3.gi 555 35	A X

How to Hide the Menu of the Media Player

Step 4. Use following code to replay the media automatically:

AxWindowsMediaPlayer1->EnableLoop = true

4.4 CTQ4 : Call External Executive File

When calling specific software externally, additional parameters could be added. For example:

[-Page \"Palletizing Wizard\"] : jump to specific page (Palletizing Wizard) in TMstudio.

[-File \"TM Palletizing OP1.xml\"] : open an existed TMstudio project (TM Palletizing OP1.xml)

TMstudioPath = "C:\Program Files (x86)\Techman\Operator UI Builder\Builder\Projects\TMstudio" callApplication TMstudioPath,"TMstudio.exe","-S -Page \"Palletizing Wizard\" -File \"TM Palletizing OP1.xml\""

4.5 CTQ5 : Call TMflow Automatically When Windows Begins

Using embedded tutorial video in UI can instruct users how to do, which is better than just text and graphics. Following steps will show how to embed video in UI. TMflow client should run in background all the time to synchronize data between robot and UI. So it's important to run it when Windows begins. There are steps to achieve this.

Step 1. Create a file with following code:

start "" "C:\Techman Robot\TM flow\TMflow.exe" "-S"

Step 2. Save it as "RunTMflow.bat"



RunTMflow.bat

Step 3. Windows Start button > right click > "Run"

Step 4. Enter shell:startup

🖅 Run		×
	shell:startup	~
	Shell startup	

Step 5. Copy RunTMflow.bat to StartUp folder





Step 6. Reboot Windows

Step 7. TMflow client should pop up automatically next time logging in Windows.



IMPORTANT:

TMflow client is a necessary software and should be executed in background. Once it's started, it will not be killed even if the button ([]) on left-up corner is pressed. It's hidden in background in most time, waiting for commands from Operator UI.

4.6 CTQ6 : Call TMflow and Send Index Accordingly

It is possible to call TMflow in background if it's not started yet or it had been shut down before. Use code below to call it with client mode and set the language to en-US:

- -S : Client Mode
- -L : Language Mode
en-US : language code(see table below for supported languages)

function:btn_CallTMflow

callApplication "C:\Techman Robot\TM flow","TMflow.exe","-S" //with last configured language
 //or
 callApplication "C:\Techman Robot\TM flow","TMflow.exe","-S -L en-US" //with en-US language
 OperatorActiveX1->ShowTMflow = false
}

Name	Code
Chinese (China)	zh-CN
Chinese (Taiwan)	zh-TW
Dutch (Netherlands)	nl-NL
English (United States)	en-US
French (France)	fr-FR
German (Germany)	de-DE
Hungarian (Hungary)	hu-HU
Italian (Italy)	it-IT
Japanese (Japan)	Ja-JP
Korean (Korea)	ko-KR
Portuguese (Portugal)	pt-PT
Romanian (Romania)	ro-RO
Spanish (Castilian)	es-ES
Thai (Thailand)	th-TH
Vietnamese (Viet Nam)	vi-VN
Danish (Denmark)	da-DK
Czech (Czech Republic)	cs-CZ
Turkish (Turkey)	tr-TR
Polish (Poland)	pl-PL

Supported languages of TMflow

4.7 CTQ7 : Read/Write Variable, Base and Point

This section describes how to access variables.



IMPORTANT:

Before accessing variables (both global/local variables) on robot, the variables should be created on robot first.

Read Variables with Operator Function

Here is an example to get the value of global g_TMOperator_TMROBOT_TMPLTZOP_Right_full with

Operator function on robot and save it to variable in UI.



Access Variables with Tags

Instead of using Operator function, we could use tag to access variables. Here is an example to show the speed of robot on UI with a Tag through TCP/IP Modbus address:

Step 1. Create a Tag

Tag Settings —					Dashboard Pict	ture
Tag Name	RB_Speed				Within Range	Out of Range
Service Engine	ServiceEngine1	- Capture Mode	Service Loop Time			
Tag Type	Get Modbus	 Data Type 	int	~		
Category Name	Default	Set	Keep Last Value		Browse	Browse
Description					Clear	Clear
Save Mode					0 < Val	lue < 0
Get Modbus —						
Get Modbus — Get Modbus		Filter	Convert	Calculat	ion and Statistics	Action After Value Char
Get Modbus	TM_Robot	Filter	Convert	Calculat	ion and Statistics	Action After Value Char
Get Modbus	TM_Robot Coil (0x01)	Filter	Convert	Calculat	ion and Statistics	Action After Value Char
Get Modbus	TM_Robot Coil (0x01) 7101	Filter	Convert	Calculat	ion and Statistics	Action After Value Char
Get Modbus — Get Modbus Device Name Function Code Start Address Length	TM_Robot Coil (0x01) 7101	Filter	Convert Convert Format Decimal	Calculat	ion and Statistics	Action After Value Char

Creating a Tag



NOTE:

About the definition of Modbus address, related manual could be found on: (TM Robot HMI) Menu > Setting > Connection > Modbus Slave > Code Table

Step 2. Link a label with the Tag



Configuring the Tag in a label component

As a result, the value of the label will be synchronized with the speed of robot automatically when the UI is running.



NOTE: Use Tags or Operator Function?

There are two ways to get variables from TM Robot on dashboard: (1) Tags (2) Operator Function. Here is the table of comparison.

	Applied Scenario	Comment
Tags	Use a label to show single	No coding needed, but
	variable and update it	limited to 100 Points in
	automatically.	Builder.
	E.g., show the "speed (%)"	
	of TMflow project on a	
	label of the UI.	
Operator	The variable needs to be	For more complicated
Function	calculated.	calculation, and coding is
	E.g., show the cycle time =	needed.
	"total time" divided by	
	"process boxes" on a label	

Access Base and Point

Bases and Points are accessible variables that could be accessed through Operator function.

Step 1. Get Base List

Here is an example to get the list of Base and save it to a list variable _List_temp1 in UI.

_List_tmp1->AddValue = OperatorActiveX1->GetBaseList

Step 2. Change the Value of Base

Here is an example to change the value of Base.

_strTemp1 = "0,0,10,0,0,0" OperatorActiveX1->ChangeBaseTeachValue = "Base_A",_strTemp1 ComboBox_CommandReturnValue->Text = OperatorActiveX1->GetBaseTeachValue->["Base_A"]

Step 3. Change the Value of Point

Similar to the operation on Base, Point could be accessed also:

```
_strTemp1 = "300,0,300,0,0,0"
OperatorActiveX1->ChangePointTeachValue = "P1", strTemp1
ComboBox_CommandReturnValue->Text = OperatorActiveX1->GetPointTeachValue->["P1"]
```



IMPORTANT:

An error will occur if the value of Point could not be reached by robot, i.e., "0,0,0,0,0,0". Try another Point that is reachable.

4.8 CTQ8 : Open Designated Subflow

To protect the main structure of flow program, we suggest system maker to call subflow for user it if the user want to do modification. Sensor checking after gripping for example. Here is an example to open a main flow or subflow.



As a result, a subflow page on TMflow Client will be shown. The user could add his customized flow here.



Calling a Subflow

4.9 CTQ9 : Control I/O on robot

Note

Here is an example to manually control I/O from Operator UI directly.



NOTE:

There is a parameter named "Device Type" in I/O manipulation of Operator command. "Device Type = 64" means accessing I/O on the flange.

"Device Type = 16" means accessing I/O in the control box.



Meaning of Different Device Types

4.10 CTQ10 : Open, Copy and Save the TMflow Project

It's easy to manipulate TMflow project through Operator command. Here is an example to open a

designated project and save it as new project or overwrite it.



An example to show the list of current TMflow project with a component ComboBox.

ComboBox_CommandReturnValue->**RemoveItem** = "all" //Initialize list ComboBox_CommandReturnValue->**Items** = OperatorActiveX1->GetProjectList

Returned Value:	
Collision_TM5	~
Collision_TM5	
TMflow_Example1	
TMflow_Example1_New	

The Project List on UI

4.11 CTQ11 : Open the Specified View of TMflow

TMflow view pages could be leveraged with show page command:



As a result, I/O page is shown as below.

Ŀ-			2	№ 0 mn	n/s 1	00 % 👒	BC91	<i>.</i>	DEMOS	i	īl.
Display Board Flow IO	Simulator Status Actio	oner Force Sensor						TMflo	ow_Exa	mple1	.prog
	Control Box						14 14				
	End Module)									
	Camera Module Light X							~			
	I/	O Page o	n TN	lflow	Clier	nt					

4.12 CTQ12 : Import TCP and Text File

External TCP and text file could be imported to robot through UI. There are two examples to import TCP file from UI to robot.

Example 1 : Import TCP File

An TCP file (TCP.ZIP) could be generated from other TM robot or TMstudio. In this example, a TCP file is stored on a USB stick then transferred to shared folder through a customized software tool as below.



Description of Importing TCP File.



IMPORTANT:

TM robot system will recognize the USB stick labelled **TMROBOT** only.



IMPORTANT:

A customized tool to copy file on different paths is needed in this section. In this example, this tool is developed by visual studio - C#. The UI maker should build this tool himself since there are no direct tool in Builder.

Follow steps below to finish the example:

Step 1 Create a shared folder.

1. [On the PC where UI will be deployed] Go to Control Panel, and select network & Sharing.

* Action Center Administrative Tools Image: AutoPlay Backup and Restore ** Action Center Image: Center Credential Manager Date and Time ** Default Programs Image: Desktop Gadgets Image: Desktop Gadgets Image: Desktop Gadgets Image: Desktop Gadgets ** Display Image: Desktop Gadgets Image:			ngs	Adjust your computer's settir
Itil Cocker Drive Encryption Imagement	🚯 Backup and Restore	E AutoPlay	🖏 Administrative Tools	Action Center
Default Programs Posktop Gadgets Device Manager Reproducts and Printers Display Esse of Access Center Folder Options Fonts HomeGroup HomeGroup HomeGroup Esse of Access Center Folder Options Estel HD Graphics Internet Options Excloseral Manager Indexing Options Estel Intel® HD Graphics Internet Options Excloseral Manager Estel Intel® Location and Other Sensors Mouse Network and Sharing Center En Notification Area Icons Performance Information and Tools Personalization	Date and Time	Credential Manager	Color Management	BitLocker Drive Encryption
Ibiplay Searce of Access Center Folder Options Fonts HomeGroup 100 IDS Camera Manager Indexing Options 100 Intel® HD Graphics Internet Options Image: Keyboard 100 Location and Other Sensors Mouse Network and Sharing Center Image: Notification Area Icons Image: Personalization	Devices and Printers	Device Manager	📑 Desktop Gadgets	👦 Default Programs
MemeGroup INDE Camera Manager Indexing Options Intel® HD Graphics Internet Options Image: Keyboard Image: Location and Other Sensors Image: Mouse Network and Sharing Center Image: Notification Area Icons Image: Personalization	A Fonts	Folder Options	Ease of Access Center	🜉 Display
Internet Options Explored Explored Mouse Network and Sharing Center Image: Notification Area Icons Performance Information and Tools Personalization	Intel® HD Graphics	Indexing Options	IDS Camera Manager	🕹 HomeGroup
🖞 Network and Sharing Center 🛛 🚍 Notification Area Icons 🔲 Performance Information and Tools 🖉 Personalization	3 Mouse	E Location and Other Sensors	E Keyboard	Internet Options
	Personalization	Performance Information and Tools	Rotification Area Icons	Network and Sharing Center
🚍 Phone and Modern 🛛 🎯 Power Options 🔂 Programs and Features 🖉 Realtek HD Audio Manager	🗃 Realtek HD Audio Manager	Programs and Features	Power Options	Phone and Modem
P Recovery 🧬 Region and Language 🕫 RemoteApp and Desktop Connections 🛐 RTX Properties	RTX Properties	RemoteApp and Desktop Connections	🔗 Region and Language	Recovery
🛛 Sound 🚯 Speech Recognition 🔞 Sync Center 🛛 🙀 System	🕎 System	Sync Center	Speech Recognition	Sound
上 Taskbar and Start Menu 📧 Troubleshooting 🍭 User Accounts 📑 Windows CardSpace	📑 Windows CardSpace	Se User Accounts	Troubleshooting	Taskbar and Start Menu
Windows Firewall Windows Under			Windows Update	Windows Firewall



2. Go to advance sharing settings





3. Scroll down, and select Turn on sharing so anyone with network access can read and write files in the Public folders and Turn off password protected sharing.

Change sharing ontions for different network profiles	
Vindows extenting options for onrecent interesting promose Vindows where the second profile for each network you use. You can choose specific options for	Home or Work (
ach protile.	When network discovery is on this computer can see other network computers and devices and is
lome or Work	visible to other network computers. <u>What is network discovery?</u>
ublic (current profile)	Turn on network discovery
Network discovery	Urn off network discovery
When network discovery is on, this computer can see other network computers and devices and is visible to other network computers. What is network discovery?	File and printer sharing
Turn on network discovery	When file and printer sharing is on, files and printers that you have shared from this computer can be accessed by people on the network.
Turn off network discovery	Turn on file and printer sharing
File and printer sharing	Turn off file and printer sharing
When file and printer sharing is on, files and printers that you have shared from this computer can	Public folder sharing
be accessed by people on the network.	When Public folder sharing is on, people on the network, including homegroup members, can access files in the Public folders. What are the Public folders?
Turn off file and printer sharing	Turn on chating so anyone with network access can read and write files in the Public folde
Public folder sharing	 I um off Public tolder sharing (people logged on to this computer can stall access these folders)
When Public folder sharing is on, people on the network, including homegroup members, can access files in the Public folders. What are the Public folders?	Media streaming
Turn on sharing so anyone with network access can read and write files in the Public folders Turn off Public folder sharing (neople logged on to this computer can still access these	When media streaming is on, people and devices on the network can access pictures, music, and videos on this computer. This computer can also find media on the network.
folders)	Media streaming is off.
Media streaming	Choose media streaming options
When media streaming is on, people and devices on the network can access pictures, music, and videos on this computer. This computer can also find media on the network.	File sharing connections Windows 7 uses 128-bit encryption to help protect file sharing connections. Some devices don't
Choose media streaming options	support 128-bit encryption and must use 40- or 56-bit encryption.
File sharing connections	(e) Use 128-bit encryption to help protect file sharing connections (recommended) Enable file sharing for devices that use 40- or 56-bit encryption
Windows 7 uses 128-bit encryption to help protect file sharing connections. Some devices don't support 128-bit encryption and must use 40- or 56-bit encryption.	Password protected sharing
Use 128-bit encryption to help protect file sharing connections (recommended)	When password protected sharing is on, only people who have a user account and password on t
Enable file sharing for devices that use 40- or 56-bit encryption	computer can access shared files, printers attached to this computer, and the Public folders. To gi other people access, you must turn off password protected sharing.
Password protected sharing	Turn on password protected sharing
When password protected sharing is on, only people who have a user account and password on this	Turn off password protected sharing
computer can access shared files, printers attached to this computer, and the Public folders. To give other people access, you must turn off password protected sharing.	HomeGroup connections
Turn on password protected sharing	Typically, Windows manages the connections to other homegroup computers. But if you have the
 Turn off password protected sharing 	same user accounts and passwords on all of your computers, you can have HomeGroup use your account instead. <u>Help me decide</u>
	Allow Windows to manage homegroup connections (recommended) Use user accounts and passwords to connect to other computers
	Public (current profile)
Save changes Cancel	Save changes Cance

Network and Sharing-3

4. Create a folder named "file" under C:\, and right click on the file properties.

Computer Local Disk (C:)		✓ +→ Search Local Disk (C;)				
rganize 🔻 🛛 😭 Open 🛛 Include in library	✓ Share with ▼ New folder			·= •		
F avorites	Name	Date modified	Туре	Size		
📃 Desktop	🐌 file	2017/4/13 下午 03:	File folder			
Downloads	🌗 inetpub	2016/5/12 上午 07:	File folder			
🔚 Recent Places	🌗 Intel	2016/8/4 上午 10:35	File folder			
	LocalService	2020/2/25 上午11:	File folder			
🗧 Libraries	🐌 PerfLogs	2016/5/12 上午 07:	File folder			
Documents	퉬 Program Files	2019/8/30 下午 04:	File folder			
J Music	li Users	2019/3/15下午05:	File folder			
E Pictures	iii Windows	2020/9/8 上午 11:00	File folder			
Videos						
Computer						
Local Disk (C:) New Volume (D:)						
Network						
DESKTOP-5HKG8PF						
	工在 03.30					

Network and Sharing-4

5. Click over the "Sharing" tab and select "share..."

General	Sharing	Security	Previous Versions	Customize
Netwo	ork File and	Folder Sh	aring	
	file Not Sł	nared		
Netwo	ork Path:			
S	hare			
Advar	nced Shari	na		
Set c advar	ustom per nced shari	nissions, cr ng options.	eate multiple shares,	and set other
	Advance	d Sharing		
		tion		
Passv Peopl	vord Protect	a user acco ers shared	ount and password fo with everyone.	or this computer
Passw Peop can a To ch	word Protect le without a loccess fold mange this	a user acci ers shared setting, use	ount and password fo with everyone. e the <u>Network and S</u>	or this computer haring Center.
Passw Peop can a To ch	word Protection without a cocess fold mange this	a user acco ers shared setting, use	ount and password fo with everyone. a the <u>Network and S</u>	or this computer

Network and Sharing-5

6. Click "add" and add "Everyone" to the list

Choose people to share with Type a name and then click Add, or click the an	row to find someone.
	► Add
Name	Permission Level
& Administrator	Read/Write 🔫
Administrators	Owner
Le Everyone	Read/Write 🔻

Network and Sharing-6

7. Click over the "Security" tab and select "add...".

Object name: C:\file		
Group or user names:		
& Authenticated Users		
SYSTEM .		
& Administrators (TM000222)	Administrators)	
& Users (TM000222\Users)		
		_
	Add	Pamaua
Permissions for Authenticated	Add	Remove
Permissions for Authenticated Users	Add Allow	Remove Deny
Permissions for Authenticated Users Full control	Add Allow	Deny
Permissions for Authenticated Users Full control Modify	Add	Deny
Permissions for Authenticated Users Full control Modify Read & execute	Add	Deny
Permissions for Authenticated Users Full control Modify Read & execute List folder contents	Add	Deny
Permissions for Authenticated Users Full control Modify Read & execute List folder contents Read	Add Allow	Deny

Network and Sharing-7

8. Add "Everyone" to the list.

elect Users or Groups	<u>δ</u> Σ
Select this object type:	
Users, Groups, or Built-in security principals	Object Types
From this location:	
TM000222	Locations
Enter the object names to select (examples)	Check Names
I	

Network and Sharing-8

9. Go back to the "Security" tab, click "Everyone" of the list, and select all the permissions.

Permissions for file		23
Security		
Object name: C:\file		
Group or user names:		
Authenticated Users		
Administrators (TM000222	2\Administrators)	
Everyone		
Users (TMUUU222\Users)		
	Add	Remove
Permissions for Everyone	Allow	Deny
Full control		
Modify		
Read & execute		
List folder contents		
Read		
Learn about access control and	d permissions	
202		

Network and Sharing-9

10. Add the folder "TM_Export" and "TM_DeburringOperator" into the existing shared file.



Network and Sharing-10

<u>Step 2</u> Build a Customized Tool to Copy File.

A customized tool is needed to move the file from source path to shard folder. Follow the specification below to create the tool in an IDE (Visual Studio – C# for example).

Parameters(String)	Input/Output	Description				
Disk_Name	Input	The name of USB stick.				
		Example: "TMROBOT"				
Source	Input	Path for the source file in USB stick or local path.				
		Example:				
		"TM_Export\RCB100-190906_BC180918\TCP.zip				
Туре	Input	Type of Target (file or folder).				
		Example: " <mark>File</mark> "				
Target	Input	Path for the target file (shared folder).				
		Example: "\\A11090031NB\file\TCP.zip"				
Result.txt	Output	A file to store the result.				
		The result is stored in file as below:				
		-1 = Error, result file is locked.				
		0 = Copy finished.				
		1 = Input parameters error.				
		2 = Source disk could not be found.				
		3 = Source file is not existed.				
		4 = Source path could not be found.				
		5 = Copy failed.				
		6 = Target path is not existed.				

The Specification of the Customized Tool

As a result, an exe file : Copy_File.exe is created.

> Copy_File	
Name	^
Copy_File.exe	
Result.txt	

The Customized Tool named Copy_File.exe

Move the folder Copy_File to C:\Project\Copy_File.

^ Name ^ ^	le
Copy_File.exe	
Result.txt	

The Copy_File Folder

<u>Step 3</u> Build the UI in Builder.

1. Build the UI layout as below.



The Example UI for Importing TCP

2. Build codes in Builder to create the function Move_File:



```
function: Move File
//Move File from Source to Target Path
{
 //Result.txt will be deleted before running Copy_File.exe
 Parameter = Disk_Name + " " + Source + " " + Type + " " + Target// Input parameters to Copy_File.exe, separated by blanks callApplication EXE_Path,EXE_Name,Parameter //Run Copy_File.exe with input parameters
 sleep 500
 for (i=0;<5;+1)//for timeout counter
    if (fe->IsExisted == true)
     file_Connection = fe->ReadLinesTolist
      switch (file_Connection->GetValue->[0])//read Result.txt
        case == -1
        TextBox_CopyFile_Result->Text = "Error, Result.txt is locked"
        break
        case == 0
        TextBox_CopyFile_Result->Text = "File copied...done"
        break
        case == 1
        TextBox CopyFile Result->Text = "Input parameters error" //
        break
        case == 2
        TextBox_CopyFile_Result->Text = "Source disk could not be found"
        break
        case == 3
        TextBox_CopyFile_Result->Text = "Source file is not existed"
       break
        case == 4
        TextBox_CopyFile_Result->Text = "Source path could not be found"
        break
        case == 5
        TextBox_CopyFile_Result->Text = "Copy failed"
        break
        case == 6
        TextBox_CopyFile_Result->Text = "Target path is not existed"
        break
      break
    else
      sleep 500
    }
 }
ļ
```

3. Create the function *btn_ImportTCP*:

```
function:btn_ImportTCP
{
    call Move_File
    OperatorActiveX1->ImportTCP = "\\A11090031NB\file", "RCB100-190906_BC180918", "TCP1"
    // \\A11090031NB\file = shared folder
    // RCB100-190906_BC180918 = Name of the target TM robot
    // TCP1 = Name of the target TCP, should be fixed when it created
    ComboBox_TCP->Removeltem = "all" //Initialize list
    ComboBox_TCP->Items = OperatorActiveX1->GetTCPList //Add TCP list
}
```

When accessing Operator Command : *ImportTCP* or *ImportTextFile*, the naming rule of path should be followed:

...->ImportTCP = "Source Path", "Robot_Name", "TCP_Name"

For example, if an command is:

IMPORTANT:

IMPORTAN1

OperatorActiveX1->ImportTCP = "\\A11090031NB\file", "RCB100-190906_BC180918", "TCP1"

Then it will load the TCP.zip from path: "\A11090031NB\file\TM_Export\ RCB100-190906_BC180918\TCP.zip

Be noted that it will always find the folder TM_Export (prefixed name) under the source path since it complies the rule of TMflow.



IMPORTANT:

The TCP name ("TCP1" in this example) could not be changed after TCP.zip is created.

4. Create other functions.



Step 4 Run the Example UI

1. Export the TCP file on the USB stick labelled "TMROBOT".



Source TCP file on the USB Stick

The path should like this:

X:\TM_Export\RCB100-190906_BC180918\TCP.zip

- 2. Insert the USB stick on the PC of example UI.
- 3. Run the TMflow Client.
- 4. Run the UI.

The "TCP1" should be created on "TCP list" after the button "Import TCP" is pressed.

Automatic Actioner		_	×
	Copy Result File copieddone TCP List TCP1 ~ NOTOOL TCP1	Import TCP	
	Error Log	Exit	

The Result of Executing the Example UI

5. Check TM robot side, a new TCP file "TCP1" is now added.

Example 2 : Import TextFile

Similar to Example1 : Import TCP File, the codes could be extended:



IMPORTANT:

When accessing Operator Command : *ImportTCP* or *ImportTextFile*, the naming rule of path should be followed:

...->ImportTextFile = "Source Path", "Robot_Name", "TextFile_Name.txt"



For example, if an command is: OperatorActiveX1->ImportTextFile = "\\A11090031NB\file","RCB100-190906_BC180918","Text1.txt"

Then it will load the Text1.txt from path: \\A11090031NB\file\TM_Export\RCB100-190906_BC180918\TextFiles Be noted that it will always find the folder TM_Export and TextFiles (prefixed name) under the source path since it complies the rule of TMflow.

4.13 CTQ13 : File Storage

What is an Application Project File?

A text-based Application Project File (APF) is introduced in this section, to carry necessary information such as TCP name, robot points, TMflow project name and variables during setting on UI. There are some characteristics about an APF.

1. An APF could be manipulated as a common file in Windows as below.

Delete	8/3/2020 6:49 PM	File folder			
📕 Template	8/3/2020 7:30 PM	File folder			
Test0729	8/3/2020 6:36 PM	File	9 KB		
Test0729 - Notepad				- 0	×
ProjectInfo,_,Test0729, TCPName,EPickShort,_,_	, _,_,_,_,Part1,_,_,_,_, _,_,_,_,Part1,_,_,_,_,_,	,_,_,_,_,_,_,@	0,_		^
PickPoint1,0.446807861, PickPoint2,	4.09790039,0.28153038,2	.0435698,-6.	389675,76.4398	346,_,_	,_,_,_
PickPoint3,_,_,_,_,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,	179.57814,3.	634315,-72.004	364,_,_	,_,_,_
PlacePoint2,_,_,_,_,_ PlacePoint3,_,_,_,_,_, 2DWIStatus false	·_·_·_·_·_·_·_·_·_·_·	07,_ 08,_ @9			
	,,_,_,_,_,_,_,@10,_ ,_,_,_,_,_,_,@10,_ @11.	,_,@,,_			
,,_,_,_,_,_,_,_,_,_,_,_,_	,_,_,_,_,_,_,@12,_ ,_,_,_,_,_,_,@13,_				
,,_,_,_,_,_,_,_,_,_,_,_,_	,_,_,_,_,_,_,@14,_				~
<					>
	N	Vindows (CRLF)	Ln 1, Col 1	100%	

An APF in Designated Folder

- 2. It's a text-based file.
- 3. Each element in APF could be defined and manipulated.
- 4. Each element in APF is separated by ",".

Example : Create an Simple UI to Save and Load Data on an APF

This is an example of utilizing an APF in an UI, these are the steps.

Step 1. Create a Folder and a Template for APF

Create a folder and a template file named Template1 in: C:\Project\APF\Template

Content of Template1:

Name	
Template1	
Template1 - Notepad -	- 🗆
File Edit Format View Help	
ProjectInfo,_,Test0729,_,_,_,Part1,_,_,_,_,_,_,_,_,_,@0,_	
TCPName,EPickShort,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_	
TCPName,EPickShort,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_	346
TCPName,EPickShort,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_	346,_,_,
TCPName, EPickShort,, _, _, _, _, _, _, _, _, _, _, _, _	346,_,_, 364
TCPName,EPickShort,,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,	346,_,_, 364,_,_,
TCPName, EPickShort,, _, _, _, _, _, _, _, _, _, _, _, _	346,_,_, 364,_,_,
<pre>TCPName,EPickShort,,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,</pre>	346,_,_,
<pre>TCPName,EPickShort,,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,</pre>	346,_,_,
TCPName,EPickShort,,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,	346,_,_,

Content of Template1

For better understanding the structure of Template1, it is translated to an excel file as below (it's NOT necessary step).

2D Array in UI														
	2nd Index (Index_Y) of Array:													
		0	1	2	3	4	5	6	7	8	9	10	 18	19
	0	ProjectInfo	_	ExampleProject1	_			_	Part1	_			@0	
	1	TCPName	EPickShort		_		_	_	_	_			@1	
1st	2	_	_		_			_	_	_			@2	
Ind	3	PickPoint1	-2.22772217	0.036315918	3.890132	3.652691	-1.77018	93.32517	_	_			@3	
lex	4	PickPoint2	_		_			_	_	_			@4	
(In	5	PickPoint3	_		_			_	_	_			@5	
dex	6	PlacePoint1	-144.976776	-790.5937	127.6212	179.0668	2.518495	-93.45637	_	_			<u>@6</u>	
×	7	PlacePoint2	_	-	_			_	_	_			@7	
ġ,	8	PlacePoint3	_	_	_			_	_	_			<u>@8</u>	
An	9	2DVJStatus	FALSE	_	_			_	_				@9	
:Ye	10	_	_	_	_			_	_				@10	
		_	_	_	_				_					
	199	_	_	_	_		-	_	_	_			@199	

Structure of Template1

Step 2. Create Variables

Create necessary variables as below.



NOTE:

Due to the limitation of the compiler, the declaration of string variable could not be ended by "\". For example:

<pre>string _strTemp1= "C:\Project\APF\Template\ "</pre>	Compile result : No Error
<pre>string _strTemp1= "C:\Project\APF\Template\"</pre>	Compile result : Error

A workaround is to use a blank value in the declaration and use "Replace" function in script to remove it later. For example:

_strTemp1= "C:\Project\APF\Template\ "

_strTemp1= _strTemp1->Text->Replace[" ",""]

Step 3. Build the Layout of the UI

Create the UI in UI editor. Components : Button, ComboBox, Label, TextBox are used in this example as below.



Step 4. Implement Functions

Build necessary functions for initialize.



Build a function to update the file list.



In this function ReadFile, a 2D array is initialized with size 200x20 to load the data from APF.



In ReadArray and WriteValue, the element in 2D array could be accessed.

```
function:ReadArray
//Read Selected Array
// (Input)Int_ArrayIndex_X=XYZ; (Input)Int_ArrayIndex_Y=ZYX; ReadArray; Output = TextBox_Result->Text
{
    Int_ArrayIndex_X = TextBox_Row->Text->Format[toint]
    Int_ArrayIndex_Y = TextBox_Col->Text->Format[toint]
    TextBox_Result->Text = 2DArray_FileDataTable->GetValue->[Int_ArrayIndex_X,Int_ArrayIndex_Y]
    TextBox_SelectRow->Text = 2DArray_FileDataTable->GetValue->[Int_ArrayIndex_X]
//Get project name
    TextBox_Project->Text = 2DArray_FileDataTable->GetValue->[0,2]

function:WriteValue
// (Input)Int_ArrayIndex_X=XYZ; (Input)Int_ArrayIndex_Y=ZYX; WriteValue;
{
//Write value to selected content
Int_ArrayIndex_X = TextBox_Row->Text->Format[toint]
Int_ArrayIndex_Y = TextBox_Row->Text->Format[toint]
Int_ArrayIndex_Y = TextBox_Col->Text->Format[toint]
2DArray_FileDataTable->ChangeValue = Int_ArrayIndex_X,Int_ArrayIndex_Y,TextBox_Result->Text
}
```

In SaveFile, the data in 2D array is saved to a new file.

```
function:SaveFile
// Save 2D Array to File
{
    file2->ChangeDefinition = Input_FilePath + Input_FileName
    file2->DeleteFile
    for (j=0;<200;+1)
    {
        _strTemp1 = 2DArray_FileDataTable->GetValue->[j]
        file2->WriteLinesByAppend = _strTemp1
    }
}
```

This is the searching function for the array.

```
function:Find2DArray
// Find string command in target 2DArray and return result
// (Input)TargetStr_Find2DArray=XYZ; Find2DArray; Output=ResultList_Find2DArray
{
  Button_Find->Text = "Finding..."
ResultList_Find2DArray->DelValue = "all"
  //TargetStr_Find2DArray = TextBox_TargetStr->Text
  for (i=0;<200;+1)
    _strTemp1 = 2DArray_FileDataTable->GetValue->[i,0]
    if (_strTemp1 == TargetStr_Find2DArray)
      ResultList Find2DArray->AddValue = 2DArray FileDataTable->GetValue->[i]
      Label i->Text = i
      break
    if (i == 199)
      ResultList_Find2DArray->AddValue = "No Data"
Label_i->Text = 0
    1
  Button_Find->Text = "Finding Done"
  TextBox_ResultList->Text = ResultList_Find2DArray->GetValue
}
```

A function to copy the template to create a new APF.

```
function:CopyAndSaveTemplateProject
{
    // Open and copy template file
    Input_FilePath = Location_ApplicationProject_Template
    Input_FileName = "Template1"
    // Save to new file
    Input_FilePath = Location_ApplicationProject
    Input_FilePath = TextBox_Project->Text
    2DArray_FileDataTable->ChangeValue = 0,2,Input_FileName
    call SaveFile
}
```

Some functions to define button events.

function:btn_ReadProject
<pre>{ Input_FilePath = Location_ApplicationProject Input_FileName = ComboBox1->Text call ReadFile call ReadArray</pre>
<i>function:</i> btn_WriteProject
{ call WriteValue call ReadArray }
function:btn_CreateNewProject
call CopyAndSaveTemplateProject
function:btn_Find
{ TargetStr_Find2DArray = TextBox_TargetStr->Text call Find2DArray }

Step 5. Run the UI

As the result, an interactive panel is shown with data manipulation to an APF.



The UI to Handle File

4.14 CTQ14 : Virtual Keyboard

There is a built-in keyboard in TMflow folder could be leveraged. A tool named TMKeyboard.exe in TMflow client folder (default path: C:\Techman Robot\TM flow\Tools\keyboard\TMKeyboard.exe) could be used.

String Location_KeyBoard = "C:\Techman Robot\TM flow\Tools\keyboard" //... function:btn_KeyBoard { callApplication Location_KeyBoard,"TMKeyboard.exe" }



Virtual Keyboard

4.15 CTQ15 : Error Handling

There is a predefined function *dashboarderror* that will be executed automatically when UI (dashboard) error occurred. Use following code to show the error message:.

dashboarderror: //Program execution point after system error TextBox Err->Text = this->GetDashboardErrorMessage return }

Besides *dashboarderror*, the component: *OperatorActiveX* has a unique error event handler. For example, a function *ErrOperator* is created and linked to the error event of *OperatorActiveX1* as below.

	 Properties 		Ψ×
	OperatorActiveX:OperatorActive>	(1	-
	8		
	 Identification 		
^^	Anchor	Top, Left	
OperatorActiveX	Description		
	Dock	None	
	IsIndex	True	
	Name	OperatorActiveX1	
	 Accessibility 		
	Role		
	 Advanced Setting 		
	CMDTimeout	6000	
	ErrorCodeShowInt	False	
	TMFlowIP	127.0.0.1	
	TMFlowPort	3849	
	 Appearance 		
Light On	Left	799	
	Size	50, 50	
Light Off	Тор	44	
	Visible	True	
	- Event		
	HasErrorEvent	call ErrOperator	

Error Event in an OperatorActiveX Component

function:ErrOperator
<pre> TextBox_Err->Text = OperatorActiveX1->GetOperatorErrorCode TextBox2->Text = OperatorActiveX1->GetOperatorErrorCode</pre>
TextBox3->Text = OperatorActiveX1->GetOperatorErrorCommand //Get the last command that triggered error return
}

Example: Showing the Error Message of the Operator Command

With function *ErrOperator*, there will be an error code *F00000E2* with the last command *LogOut* if we

triggered *Logoff* continuously. Where *F00000E2* means "already logout" on error code table.

	TMFlow Error: F00000E2
Login	LogOut
Logoff	

Error Code after Logout



IMPORTANT:

Full table of error codes could be found in Software Manual TMflow 1.80.

Normal Procedure to Leave UI

There are positive and negative examples to leave UI.





IMPORTANT:

Do not use operation command in predefined function <u>*close*</u>. Use a self-defined function to exit UI instead.

4.16 CTQ16 : About Dashboard Console

After the trail run of UI in builder, the UI could be transfer to the dashboard console on target PC. The relation is described as below.



The Position of the Dashboard Console

4.16.1 Deploy UI to Dashboard Console

These are the final steps before running the UI :

1. [On target PC] Run Dashboard Console 🚺



- 2. [On builder PC] (Back to Builder) press Deploy icon.
- 3. Enter IP of target machine.
- 4. Press automatic claw attached file and auto run.
- 5. Deploy.

E builder			- 0 ×
View(1) Tool(T) Help(H) Options(D)	2		
Devices Designer Tags Monitors Actions	Groups Management	Service Engine Status ServiceEngine1	• Date
Teebes 9 × 10	ter Paletong, Sergida, VI X	Properties	+ ×
Device Controls		Hsin/Washow Msin/Washow	
Data Controls	an Worker Stabilization 1	20 14 IM	
Common Controls ProductProcessCS.		Accessibility Eds	
StationControl		Delaymalloor	
C AlgatorCeter RationImage	TM peratorPalletizing Operator Example - Pour	wered by Detgeoutingsteed Challon Parent	For
		Couver	Default
	3	RuiDulyour	Tree
		Designed and the second s	- Control
	M Dunborud,D	D × Dates	ching Type
	4	Rechter	ControlText
	Destruction 127.001 Port: 5433	5000 775 228	1201, 443
	Project Namer - Company from the international state LL R	Led.	
	Attached File Attached Remove Automatic daw et	tached file	
		Contribut	100
	Resource (or annument prog	Control Constitution	Tree
	'M_DeshBoard2.jpg	Pendlostedhje	Sinable
	1 indefinejog	E Hussarder	Tree
	Lobot pro	O Monoralia	0,0
		Minimiz-Dox	Tree
		Maannellas	0,0
	S Ato fue	Devisiteiter	Tree
		Deploy	a, a
		Telder	
	and the second se	Waday Laborat	False
		Technolog	AV MARK
		3 Define the background voltor of	rele Couteral
	tay Cole May		
and the second se			

Setting on Dashboard Console

4.16.2 Lock the Toolbar on the Dashboard Console

Now there should be an UI running on the dashboard console (on target PC). To lock the toolbar, some steps should be performed.

- 1. Click "setting".
- 2. Enter password (enter a new password in the first time).

⁰²⁴⁵³⁵ perator Palletizing (Deperator Example - Powered by TM Operator Platform	
	Select Project to Work	
	Projec Part No. Box Size(LxVxrr) Box Size(LxHxW) Box Weight Box Weight Pallet Size(LxWxH) Pallet Size(LxHxW) Total Quantity Total Quantity Ready to Work Project Date Project Date	

Setting on the Dashboard Console-1

- 3. Click "lock".
- 4. Save.
- 5. Now the toolbar of dashboard console should be locked (Windows toolbar will be hidden also).

8 #	- 0 ×
Dashbourd List	
Dashboard Manager	5
Dashboard Settings	
Lock 3 Reset Password	
Change Bar Color 🔟 Upload Image Reset Delete projects 🗸 Delete	
Connection Settings	
IP 1277.0.1 Port 5433 Dynamic IP Man/D6 Change	
Societ communication connect wait time 2000 ms Auto Update	
Projects List Settings	
Auto Run List V Add Delete Run Applications Browse Add Delete	
AAA Palletizing_Template_V1 C\\Techman Robott;TM flow\tTMflow.exe -5	
Version: 4041102 Cancel Saw 4	

Setting on the Dashboard Console-2

4.16.3 Lock the Hot Key on Dashboard Console

It's also important that to lock the hot key on the keyboard to prevent users to enter Windows directly. Use the 3rd party keyboard configuration tool (for example: SharpKeys, which is open software) to lock the keys as below.

From:		To:	
App: Calculator (E0 21)		Tum Key Off (00_00)	
App: My Computer (E0_6B)		Tum Key Off (00_00)	
Function: F10 (00_44)		Tum Key Off (00_00)	
Media: Play/Pause (E0_22)		Tum Key Off (00_00)	
Special: Application (E0_5D)		Tum Key Off (00_00)	
Special: Home (E0_47)		Tum Key Off (00_00)	
Special: Left Alt (00_38)		Tum Key Off (00_00)	
Special: Left Windows (E0_5E) Special: Right Alt (E0_38) Special: Right Windows (E0_5C) Web: Home (E0_32)		Tum Key Off (00_00)	
		Tum Key Off (00_00)	
		Tum Key Off (00_00)	
		Tum Key Off (00_00)	
444 1240	Tulies Dates 41	Write to Passister	Char
Add Edit	Delste All	Write to Registry	Close

Screenshot of Locking Keyboard on SharpKeys

5. Advanced techniques

This chapter introduces several advanced techniques of building the Operator UI using TM Operator Suite. These techniques are presented in an example of simple Pick and Place application.

5.1 General description of the example

Usually, an operator contains 2 phases: (1) programming phase and, (2) running phase. An example of planning the relations ship between the page of "Robot status" and the pages of "Programming UI" is shown below.

In this case, we will use the vision function to classify objects a, b and according to the different results to place the object to the corresponding area. The diagrams below present the conceptual automation process diagram and the UI design diagram of the example of the pick and place Robot Operator.



Conceptual automation process diagram of the pick and place robot operator



UI design diagram of the example of the pick and place Robot Operator

5.2 ATQ 1: Protection of TMflow Project (Component)

In order to avoid the end user directly open the original programming project, we can use the component editor to protect the project. For "Component Editor" programming and its rule, please refer to Software Manual TMflow.

Step 1. Flow programming

In first, according to this application, we finishing the TMflow project by Component Editor, and save it as the component.



Pick and place programming on TMflow



IMPORTANT:

The naming rule of **TM Component Editor** goes by Application_Provider_Model_Version_Function

IMPORTANT:

The component's icon supports PNG image files only, and the suggested maximum resolution is 73 x 73 pixels to avoid distortion or blur.





Component Icon Resolution

Page	Description
Main	Pick&Place logic programming
GRIP	Gripper gripping logic programming
RELEASE	Gripper placement logic programming
Get_Value	Using varSync command to send value of variable to dashboard

IMPORTANT:



- When using a TCP/Global Variable, the prefix of the TCP/Global Variable has to go with the first two names (Application_Provider_) of the project name in TM Component Editor.
- Inappropriate changes of project names in TM Component may result in missing of the embedded Global Variables.

The relevant logic programming of main flow instructions are as follows.



Pick and place main flow instructions

Area	Description
Α	According to the value of global variables to take a different trajectory.
В	Vision function to classify the objects a, b.
С	Reach above the object position then reach grip position.
D	Call grip subflow to grip object.
E	Judge the object type to determine the placement area.
F	Reach above the place position then reach the "a" object placement area.
G	Reach above the place position then reach the "b" object placement area.
Н	Call release subflow to release object.
I	According to the object type to determine the back trajectory.
J	Back trajectory.

Descriptions of Each Block in the Flow



NOTE:

In order to make Grip/Release replaceable, these two subflow targets are parameterizable.

Step 2. Enable Component

When the component is completed, In order to show this component in flow left side, you can enable this component on robot setting page.



Enable Pick and Place Component

Step 3. With component is enabled, create the new project and drag this component into this project.


Drag Component into project

Step 4. In order to add the flexibility to allow users to replace the gripper by themselves. We can add two empty subflow into this project, User_Grip, and User_release.



Add subflow on project for grip and release

Step 5. Finally, if you want to give the component for the user to use, you can export this component from system page of import/export page.

\equiv \leftarrow .	🧷 👫 0 mm/s 5 % 🤫 🖅 🔋 🧧 📔 🗈
Import Export Select files \TM000223	Selected files
тср Т	Component TMOPerator_TMROBOT_TMPICKPLACE_V001_Mai
Command	
앱 Component	
Operation Space	
Var Global Variable	
Network Service	
Text file	
IODD Files	
⊟ ^{ti} Ethernet Slave	
Backup File	Free Space: 25332 MB Export

Export Component to USB



IMPORTANT: If the project package to the component, the user can run it, but cannot to parse and modify its content.

5.3 ATQ 2: How to establish Accessibility management system on builder

After finishing the TMflow project, you can start creating a User Interface; with Builder of Operator Platform, it is easy and intuitive to do so. In order to create a multi-page system, we use tab control as the container. In this case, on page 1, we planning the user can select edit parameters or start to run directly, but edit the operator's parameters ability we only open to the engineer role can use, so in this section, we will Introduce how to establish a rights management system on Dashboard.



Pick and Place Operator Start Page



NOTE:

In order to run or edit projects, we have added commands to open the specified project under these buttons.

Step 1. Usually, the Operator might require a login entrance with different accounts of different authorities, in this case, we could use the "Role Property" function of Builder to create different accounts. For example, we have "Abner" as an engineer account while "Andy" is an operator account. According to different role, UI appearance or functionality could be different. (e.g. define by the property "Visible" or "Enable").

in Window	SubWindow1			
	200 00100001			
Role	ePropertyForm			
	Group Name Eng	ineer 🗸	+	
05	Username]	
•	Enable True	· · ·	Ac	dd Data
p	Visible True	× ×		
_	a 11			
	Group Name	Usemame	Enable	Visible
			_	

Set role property with account

Step 2. When the project is running, the dashboard will pop-up the login system windows for users enter the identity information.

🔤 Builder			
檢視(V) 工具(T) 說明(H)	選項(O)		
Designer	n 🖹 📊 💷	A. 🔅	□ × Service Engine 狀態
: 🔤 使用增助编 Toolbox			
Device Con			🖡 🔒 🎹 🔏 📲 🕒 <table-cell-rows> rr 📍 🕮 📇 🖾 💵</table-cell-rows>
Data Contro Common Co	Image Folder		
<pointer></pointer>	Group Name: Engineer	~	
ab Button GroupBox PictureBox	User Name : Abner		Robot Error message :
Panel	Password :		voor Gipper TCP Setting Controller Grippe
Ei ComboBox Di TextBox TabContro dataGridVi	Cancel	Login	use the Pick and Place (
 RadioButto CheckBox 			
WebBrows			
A HScrollBar		Can minow	
VScrollBar			
		Connect	
		Login	Start to use TM Op
	UI Editor	Code Editor	

Login by identity information

5.4 ATQ 3: Call vision job

On the second page, the user can call up the programmed vision job to reprogramming to identify the object a, b. Therefore, we call the Smart-Pick vision job through the command to programming the object a, b.

In order to let the user can re-teach the vision job quickly, so we used the Smart-Pick job in TMflow. The user can re-teach the object features step by step and without using the calibration board.



Call vision job page on dashboard



5.5 ATQ 4: Choose executing gripper on dashboard

For the user, sometimes they don't want to use the operator's official gripper, therefore, the maker must add the flexibility that the user can change gripper in dashboard by themselves. In this section, we will introduce how to design this flexibility that change grippe in the dashboard.

5.5.1 Choose Gripper by parameterize of subflow in TMflow

On the third page, the user can decide which gripper will be used (Use default or their own gripper) as below.



Select gripper to use

Step 1. If the user uses the built-in gripper and clicks the Next button, the Grip/Release subflow target in the original component shall be the Grip/Release subflow in the original component.

If the user wants to use other grippers, the Grip/Release subflow target in the original component shall be the new project subflow.





IMPORTANT:

If the TMflow project is packaged as a component, the parameters of the original project will be prefixed. For Component Editor programming and its rule, please refer to Software Manual TMflow.

5.5.2 Programming custom gripper logic

The operator command can jump to the specified subflow of the specified project of TMflow for programming, please refer to Chapter 4 CTQ11.



5.5.3 Change TCP parameters

Assuming that a custom gripper is used, you can modify the TCP value for fine-tuning, or directly call the TCP teaching page of TMflow to restart the teaching.



TCP setting page on dashboard



<i>function:</i> Get_TCP_Value
TCPvalue = OperatorActiveX1->GetTCPTeachCoordinates->["TMOperator_TMROBOT_PickGripper"]
TextBox1->Text = TCPvalue ->GetValue->[0]
TextBox2->Text = TCPvalue ->GetValue->[1]
TextBox3->Text = TCPvalue ->GetValue->[2]
TextBox4->Text = TCPvalue ->GetValue->[3]
TextBox5->Text = TCPvalue ->GetValue->[4]
TextBox6->Text = TCPvalue ->GetValue->[5]
TCPvalue = OperatorActiveX1->GetTCPTeachMass->["TMOperator_TMROBOT_PickGripper"]
TextBox7->Text = tcpmass->GetValue->[0]
TCPinertia = OperatorActiveX1->GetTCPTeachInertia->["TMOperator_TMROBOT_PickGripper"]
TextBox8->Text = [CPinettia->GetValue->[0]
lextBox9-7fext = 1CPinettia-SGetValue->[1]
lexibox10->7ext = 1CPinettia->GetValue->[2]
Torcenter = OperatorActiveX1->Get/CF/eacromassCenter->[TMOperator_TMROBOT_PickGripper]
TextBox11->7ext = 1CCCenter->GetValue>[0]
TextBox12>7ext = TCCenter-SetValue>[1]
TextBox10>-7ext = TCCenter-CetValue>[2]
TextBox16-7Ext = TCPconter-CetValue-[3]
TextBox16>Text = TCPcenter->CotValue>[4]
function:Set_TCP_Value
l //TextBox6->Text="TMOnerator_TMROBOT_PickGripper"
OperatorActiveX1->ChangeTCPTeachValue=TextBox6->TextTextBox12->TextTextBox13->TextTextBox14->TextTextBox14->TextTextBox15->TextTextTextTextTextTextTextTextTextText
Bayla-STayl TextBoyl2-STayl
OperatorActiveX1->ChangeTCPTeachMass = TextBox6->TextTextBox19->Text
OperatorActiveX1->ChangeTCPTeachInertia =TextBox6->Text.TextBox21->Text.TextBox22->Text.TextBox23->Text
OperatorActiveX1->ChangeTCPTeachMassCenter =
TextBox6->Text, TextBox25->Text, TextBox26->Text, TextBox27->Text, TextBox28->Text, TextBox29->Text, TextBox30->Text



NOTE:

You can put more complex parameters in advanced like moment of inertia.

5.6 ATQ 5: Move the robot and record the point.

And next, in order to let the user can overwrite these points from TMflow project, the maker needs to establish the robot controller on the dashboard for the user to move the robot until the robot moving the appropriate position.

Point Name
Above_pick
pick
Above_place_A
place_A
Above_place_B
place_B

Point Name in TMflow

Therefore, we will introduce the function as below in this section.

- 1. How to establish robot controller on dashboard
- 2. How to overwrite the exist point.



Move the robot and record the point page on dashboard

5.6.1 How to establish robot controller on dashboard

The figure below is an example of the robot controller we built. The radio button represent the axis of robot movement (X, Y, Z, RX, RY, RZ). Click + or -, the robot will according to the selected base, TCP and speed to move forward or backward, the following we will introduce how to establish this robot controller on the dashboard.



Robot controller on dashboard

Step 1. In order to obtain a robot's current coordinate, we need to create a thread first. As follows, we create a Thread named "Get_Robot_Coordinate", and the thread will execute the function named "Get_Coordinate_Function" to continuously obtain the current coordinate of the robot.

define: //Variable declaration area
{
thread Get_Robot_Coordinate = Get_Coordinate_Function,500,0,0
list lstCoordinates = string
}
close: //Program execution point before the project leaves
{
Get_Robot_Coordinate->Close
}

```
function: Get Coordinate Function
   if (OperatorActiveX1->IsControlled == true)
      lstCoordinates = OperatorActiveX1->GetCoordinatesByCurrentBase
if (lstCoordinates->Length == 6)
      ł
        LabelX->Text = lstCoordinates->GetValue->[0]
lblControllerX->Text = lstCoordinates->GetValue->[0]
LabelY->Text = lsgetCoordinates->GetValue->[1]
lblControllerY->Text = lstCoordinates->GetValue->[1]
LabelZ->Text = lstCoordinates->GetValue->[2]
         lblControllerZ->Text = lstCoordinates->GetValue->[2]
         LabelRX->Text = lstCoordinates->GetValue->[3]
         IblControllerRX->Text = IstCoordinates->GetV
                                                                                 ue->[3]
         LabelRY->Text = IstCoordinates->GetValue->[4]
         lblControllerRY->Text = lstCoordinates->GetValu
LabelRZ->Text = lstCoordinates->GetValue->[5]
                                                                               a<mark>lue</mark>->[4]
         IblControllerRZ->Text = IstCoordinates->GetValue->[5]
     }
  }
}
```

In order to achieve the button must hold press when the robot will be moved, we send the movement distance to 9999 for a specific axis according to the radio button selected and continue to send the "StartBeacon" function (Keep the robot to move). When the button is released, we stop sending "StartBeacon", and send "StopBeacon" (force interruption robot to move) to stop the robot movement.

Event	
ClickEvent	
MouæDown	call Start_Beacon
MouseEnter	
MouseLeave	
MouseUp	call Stop_Beacon
Target	

Button event to hold a press

```
function:RadioButton_Controller
 OperatorActiveX1->ChangeCurrentBase = &cbxBase->Text
OperatorActiveX1->ChangeCurrentTool = &cbxTool->Text
 targetPos = lstCoordinates->GetValue
if (Add_or_Not == true)
   Move_Distance = 9999
 else
   Move_Distance = -9999
 }
if (&radioBtnX->Checked == true)
   targetPos->ChangeValue = 0,Move_Distance
 if (&radioBtnY->Checked == true)
   targetPos->ChangeValue = 1,Move_Distance
 if (&radioBtnZ->Checked == true)
   targetPos->ChangeValue = 2,Move_Distance
  if (&radioBtnRX->Checked == true)
   targetPos->ChangeValue = 3,Move_Distance
  if (&radioBtnRY->Checked == true)
   targetPos->ChangeValue = 4,Move Distance
  if (&radioBtnRZ->Checked == true)
   targetPos->ChangeValue = 5,Move_Distance
 }
```

```
function:Start_Beacon
{
    while (Start_Add == true)
    {
        OperatorActiveX1->StartBeacon = true
    }
}
function:Stop_Beacon
{
        Start_Add = false
        OperatorActiveX1->StopBeacon = true
}
```

Step 3. When you click + button (Execute Add_Move), the robot will move in a positive direction, otherwise, when you click the "-" button (Reduce_Move), the robot will move in the negative direction.



5.6.2 How to overwrite the exist point

After obtaining these parameters, we need to obtain the robot's current coordinate and pose,

and write the current coordinate and pose to Above_pick to complete the overwrite point.



5.7 ATQ 6: Execute grip/release function on dashboard

To teach the gripping position of the object, it is necessary to actually grip the object. This section will introduce how to execute Grip/Release on the dashboard.



Execute grip and Release page on dashboard

5.7.1 Create action to play robot

If your gripper is controlled by IO, you can directly control the IO through the Operator command, but if your gripper cannot be directly controlled by IO, you can run the project to achieve grip/release. In order to play a project, first, we set up the Modbus for running the robot.

B	Builder						- 0	×
V	iew(V) Tool(T) Help(H) Options(O)							
	Devices Designer Tags Monitors	Actions Groups	FC Management	Se	rvice Engine Status TM	Service1	Display	
A	Toolbox P	× NEW vvv Action	s Form X			Properties		φ×
ğ	Shop Floor Controls	E9				RadioButton:RadioButton1		
w.	Device Controls	Action Setting				9== \$J. 1 ==		
	<pointer></pointer>	Action setting				Identification		
	X RobotControl1	Action Name Play	Servic	ce Engine TMService1	~	Anchor	Top. Left	
	RobotControl2			THE PLAN		Description		
	DeviceActiveX	Action Type Set Modbus	~ Categ	ory Name IMR_Robot	Set Category	Dock	None	
	TagsActiveX					Islndex	True	
		Description				Name	RadioButton1	
		- Set Modbus				TagDataAutoChange	False	
		Set moubus				TagName		
		Device Name	Robot			Accessibility		
						Kult		
		Group Name				Checked	False	
						 Annearance 	1 (110)	
		Function Code	WriteSingleCoil(0x05)	~		BackColor	Control	
						Datatype	string_Type	
		Start Address	7104			Enabled	Irue	
						Font	PMingLiU, 9pt	
		Convert Type	Binary Convert to byte[]	~		ForeColor	ControlText	
			4	1		Left	130	
		Value	η .]		Size	104, 24	
						Text	RadioButton1	
						Top	49	
						V isible	Irue	
	Data Controls Common Controls					Name Code configuration number, us unique in each project	ed for programming, must be	
	Compile/Runtime Error Messages 搜尋與置換							
Upd	ate completed							

Create action for robot play



NOTE:

Before using Modbus to play the robot, please make sure that Modbus under the connection page of the robot setting page is turned on.

5.7.2 Execute grip/release

We modify the value of global variables and run the project, let the robot only run the subflow of grip/release.



Grip/Release flow instructions





NOTE:

Global variables can be modified through operator command before the robot is running.

5.8 Dashboard Page show robot information

Here is the page for Running UI. There are some examples to show necessary information on Running UI:.



Dashboard page for running UI

Although the operator command can be used to obtain the variable value, but for the information that needs to be constantly refreshed, we recommend using tag.

5.8.1 TMflow sync value to dashboard

We can actively send a value of the variable to the dashboard through the SET node of VarSync function of TMflow.



Instructions of Sync variable between TMflow and dashboard

\leftarrow	Expression Editor Setting	
	+= 🔻	
	Add	
Int	var_Test = VarSync("g_A_Count")	
	ок	

SET node setting with the TMflow send the value of the variable to dashboard



IMPORTANT:

The waitfor node is to put the current thread to sleep to avoid the system busy.

5.8.2 Use tag to receive value

The parameters obtained by the VarSync function can be received through the Tag function.

2 Builder		- 0 ×
View(V) Tool(T) Help(H) Options(O)		
Devices Designer Tags Monitors Actions Groups	🔊 . 🔯 . Src Management	Service Engine Status TMService1 Display
Vertify Vectory respin Capacital Color Devices Designer Tags Monitors Actions Toolbox 0 Subo Floor Controls 0 Device controls 0 Device controls 0 Common Controls 0 Device controls 0 Textbox 1 Device controls 0 Device controls 0 Dev	Device Name Robot Variable Period Variable Filter Convert Calculation and Statistica Action After Value Change Variable Period Variable Filter Convert Calculation and Statistica Variable Period Variable Filter Convert Calculation and Statistica Action After Value Change Variable Oedot Variable Filter Convert Calculation and Statistica Action After Value Change Over Name Constant Option Constant Calculation and Statistica Action After Value Change Over Name Obed Variable Obed of Variable Out of Price O	Properties P × MatWatory MatWatory * Accessition Eccessition BekgroutInage BekgroutInage BekgroutInage BekgroutInage BekgroutInage BekgroutInage BekgroutInage Eccessition Carso Defealt MatGologic Eccessition Detectory Control MatGory Control Control Oto Tree M
		BackColor Defare the background color of the Control.

Set the tag to receive TMflow value of the variable

5.8.3 Link value of variable to label

Label can directly link the value of tag.

Identification	
Anchor	Top, Left
Description	
Dock	None
IsIndex	True
Name	Labell
TagDataAutoChange	True
TagName	Get_A_Count
Accessibility	

Link value of variable to label

5.9 ATQ 7: Engineer Mode

There is a specific page (component: SubWindow) of engineering mode on Dashboard. This page will be called if the transparent button is pressed. Engineering mode is a convenient tool for debugging during developing the Dashboard. Here is an example to test TM Operator Platform Command:

ShowPage in Engineering mode.

©perator	Palleti	zing Operato	Pr Example - P	owered by TN	Engineering r		Engineerin	g Mode -	– Editing F	⊃age		
Reference Control Panel	2		9		GoToSubPage Ba	JumpToPage Ne Exit	Er Pallet sing_Template_VI	× 0 1. 17 5. 5	****	<u>]</u> • • • €, ? ⊒,	A ■ ■;	
Basic function InRobotLink? GetControl? IsProjectRunnie Command #0101 ShowFlow #0101	Call TMRow-S Close TMRow w Show/Hide TMRow	Login Logoff Input Parameter	GetControl	22.01.46 TMFlow Ener Water State ShouPage Water State Water State Water State Water State Water State	Light On Light Of Pett Crease NewPage 200 Lines	Cal Thitudo SendTable	Basic function Basic function In RobotLink2 GotControl? In BProjectRunnin	Cell TMFlow-S Close TMflow Show/Hide TMFlow	Login	GetControl	TMFlow Error: ShowPage	Light On
#0101~ Excute	Example: La	Returned Value:	(Click me for update	SettingCartesianA SettingCartesianA SettingCartesian8 SettingProjectSpeed SettingController SettingSpeech	Here Col. Va 0 Read 0	lue Write Value and Save to Project	Command #0101		Input Parameter		FileControl	Light of
Run Point Save Current Point	Run Point1 Run Point2	TMOperatur, TMROBOT_T Shp2_Boxl_Recognition_I Shp3_Boxl_Teach_Orippu vision_Shp3_Boxl_Recogn Label17	MPLTZOP_V002_Mein Jox_Pointon u_Point utton_Box_Pointon	SettingComponent SettingComponent SettingCommand SettingConnectionMoc SettingConnectionMoc SettingConnectionTcpl SettingPosture SettingPosture SettingPosture SettingClobalVariable	Finding in 0 Data		#0101~ Excute	Example: Lab	vel8 Returned Value:	Labelő	Selected Proje	ct Create NewPro ere Col.

Engineer Mode

5.10 ATQ 8: Protection of Builder Project

When the TMflow project is completed, you can start to add the Interactive dashboard for this UI, In order to avoid the user directly open the original programming project, we can use the Encryption function in build to protect the project.

MainWindow	SubWindow1		∔ ≗ ▣ ※ ч	<u>[]</u> , +1 (+ <mark>)</mark> [⁽	? 🗷 A 🖸) B(††
1005e you	F r Gipper TCP S	Robot Error mes	sage :	After Place	Place Point	Rot
u can y perf	use the orm clar	Password: (empty means	no pessword) Cencel	ok 5 b	oints	Cor
G	Gripper					_

Set Password to protect the builder project

Step 1. Click this button on the Builder to pop up the password window for setting the password to log in after opening the file.



Enter password to enter builder project to program



IMPORTANT:

The password will only appear when the builder is using it. If the project is opened from the dashboard console, there is no need to enter the password.

5.11 Deliver files to robot operator IPC

When you all finish your operator, you can deliver files to robot operator IPC, this section will introduce how to deliver files to robot operator IPC.

Step 1. The maker installs the TMflow Client to the path assigned by themselves. <If a newer version of TMflow Client already exists on the robot operator IPC /IPC base pendant, it needs to be removed before installation>

Step 2. If the TMstudio be used in this operator.

Step 2.1. The maker installs the TMstudio to the path assigned by themselves.

<If others version of TMstudio already exists on the robot operator IPC /IPC base pendant, it needs to be removed before installation>

Step 2.2. The maker needs to place the TMstudio environment file to the specified path. (According to the TMstudio specification). (EX. Conveyor CAD, Pillar CAD, TCP File, TCP CAD, etc.) [Can use TMstudio import and export function to achieve]

Step 3. If the Maker has provided the additional software, the maker must place/install it to the path assigned by themselves.

Step 4. The maker needs to install the database to the robot operator IPC/IPC base pendant.

Step 5. Maker installs dashboard console and service engine by themselves.

Step 6. The maker needs to check the TMmanager checkbox of Enable TMmanager on the TMmanager setting page in TMflow and set the corresponding IP of service engine.

Step 7. The maker needs to place the operator UI related files (projects, pictures, etc.) to the path assigned by themselves.(\\ TM Manager \ TM Dashboard \ Project) (The maker can deploy the project through the builder deploy function or using method 4.12 to replace step 6, 8)



IMPORTANT:

The exported file has two types that project file (.prog), or a file (.obj) purely for the dashboard console to run. If the maker does not want the programming project to be opened by the other people who have the builder. The maker only needs to provide the .obj file to the dashboard console without providing the .prog file.

Step 8. The maker needs to import the file of the database. (Use the database built-in function "backup and restore" or using method 5.12 to replace step 6, 8)

<The data in the database needs to be compatible with the version that already existed in the robot operator IPC/IPC base pendant.>

<If old database already existed in the robot operator IPC /IPC base pendant., the maker need to manually delete TMMainDB, TMMLogDB, TMMDemoDB first>

<If another version of dashboard console already exists in the robot operator IPC /IPC base pendant, the maker need to remove it before installing>

Step 9. The maker import TMflow projects/component and other related files (TCP, global variables) to the Robot controller via using USB.

Step 10. The maker needs to set the dashboard console as automatically executing after the computer turned on. (Please refer to Chapter 4 CTQ5)

Step 11. The maker needs to set a dashboard as automatically executing after the Dashboard Console is open. (Dashboard Console setting)



IMPORTANT:

The maker can manually modify the "Dashboardconfig.XML" under the folder of the dashboard console to achieve the step 11-13. The maker can copy this file and paste it to another robot operator IPC and quickly complete the same setting in another robot operator IPC.

Step 12. Setting windows and dashboard console to prevent users from entering windows. (Please refer to Chapter 4 CTQ15)

Step 13. Press the ESC button on your keyboard to hide the frame of the dashboard console. (This status will be recorded.)

Step 14. Re-boot robot operator IPC. (According to the previous settings, when the computer re-boot, the dashboard console and its project will be executed automatically.)

5.12 ATQ 9: User updates Dashboard Project

This function can export the dashboard and database data created by the builder and import it to the dashboard console. This function can not only transfer the data on the maker computer to the robot operator IPC. For end-users, if the maker updates the operator UI version, the user can download the new version file of the dashboard on the website, and using USB to update the dashboard on the dashboard console.

😳 Builder						-	ð	×
檢視(V) 工具(T) 說明(H) 選項(O)								
Modbus Test Modbus Test Designe Data Import / Export	Management		Service Engine 狀態	Servic	eEngine1	•	顯示	
editor Controller Project Impo				→ Press	operties		I	Ψ×
Export Import				Ta	Page:TabPage1			
Evenent Dreiset				^ 🔒	₩			
Export Project					Identification			^
Droject Name					Anchor	Top, Left		
Project Name					Description	None		
Config	Image	Select Images			Isindex	Ттпе		
-		L			Name	TabPage1		
					TagDataAutoChange	False		
					TagName			
				- ×	Accessibility			
					Role			
				- ×	Advanced Setting			
					AutoScroll	False		
					BackgroundImage			
					BackgroundImageLayout	Tile		
					BorderStyle	None		
				- ×	外觀			
					BackColor	64, 64, 64		
					Enabled	Тгие		
					Font	¥erdana, 9pt		
					ForeColor	ControlTe	ct	
					Left	4		
				>	Size	944, 646		
					Text	Robot Connect		
					Тов	22		~
				Ba	ckColor			
				De	tine the background color of	the Control.		

Export builder project

Step 1. Open the Data Import / Export in the Tool and select Project Import / Export.

Buil	der						
檢視	(V) 工具(T)	說明(H) 3	選項(O)				
(Des	signer De	evices T	ags Monit	ors Actions	Groups	Management	Service Engine 狀態
₹_	Editor Cont	roller	Project Import/E	xport ×			
	Export	Import					
·		Devices		ServiceEngine I.Riv	•		
		Devices		TM3630.TMRobot		-	
		Tags		aaa		-	
		Tags		AchievingRate		-	
		Tags		bbb		_	
		Tags		Consumption		-	
		Tags		Current		_	
		Tags		CurrentDeviceMode	1	_	
		Tags		CurrentDeviceNo		_	
		Tags		CurrentOrderNo		_	
		Tags		FailQty		_	
		Tags		HourSpent		_	
		Tags		InspectionResult		_	
		Tags		LastLabel		_	
		Tags		OrderQty		_	
		Tags		PassQty		_	
		Tags		ProducedQty			
		Tags		RemainingHour			
		Tags		StartTime			
		Tags		Temperature		×	

Export SQL information

Step 2. Choose to export and save as a zip. File to the USB.

	– 0 ×
Dashboard 清單 🔽 💽 💽 💽	
Import Project	
Import Project File	
Select Service Engine	
Device Replacing	
Previous Devices Current Devices IP Address	
Check	

Open the page for Import project on the dashboard console

Import Project					
Import Project File		Controller08	24_08242020)145422.zij	p •••
Select Service Engine		ServiceEn	igine1		\sim
Device Replacing					
Previous Devices	Current De	evices	IPA	ddress	
			C	ancel	Import

Import .obj file and SQL information to console

Step 3. Insert the USE to the robot operator IPC. In the dashboard console, select the .zip file exported by the build and select import to import the file into the dashboard console. If importing finish, you can find the new dashboard on the Drop-down menu.



Open the imported dashboard

Appendix A: TMflow Command

The chapter will introduce the command of TMflow when it's as the builder slave.

Before you use these commands, please confirm the TMflow opened in passive mode, and Dashboard has the component named OperatorActiveX.



TMflow startup parameters are as follows: TMflow [/S] [/L Lang]

- /S Assign TMflow to be activated in Slave Mode (If this parameter not sent, it's Normal mode)
- /L Assign language
- Lang Assign catalog and/or file name for new catalog or file

The following example, we use passive mode to open the TMflow and set its language is English: call Application "C:\Techman Robot\TM flow","TMflow.exe","-S -L wn-US "

If your TMflow has these languages, you can use these commands to open the TMflow with the corresponding language.

Language/region	Language/region tag
Arabic (Saudi Arabia)	ar-SA
Bulgarian (Bulgaria)	bg-BG
Chinese (Hong Kong SAR)	zh-HK
	Note: No longer used. See zh-TW.
Chinese (PRC)	zh-CN
Chinese (Taiwan)	zh-TW
Croatian (Croatia)	hr-HR
Czech (Czech Republic)	cs-CZ
Danish (Denmark)	da-DK

Dutch (Netherlands)	nl-NI
English (United States)	
English (United Vingdom)	en GB
Finnish (Finland)	
French (Canada)	fr-CA
French (France)	fr-FR
German (Germany)	de-DE
Greek (Greece)	el-GR
Hebrew (Israel)	he-IL
Hungarian (Hungary)	hu-HU
Italian (Italy)	it-IT
Japanese (Japan)	ja-JP
Korean (Korea)	ko-KR
Latvian (Latvia)	lv-LV
Lithuanian (Lithuania)	lt-LT
Norwegian, Bokmål (Norway)	nb-NO
Polish (Poland)	pl-PL
Portuguese (Brazil)	pt-BR
Portuguese (Portugal)	pt-PT
Romanian (Romania)	ro-RO
Russian (Russia)	ru-RU
Serbian (Latin, Serbia)	sr-Latn-CS
	Note: No longer used. See sr-Latn-RS.
Serbian (Latin, Serbia)	sr-Latn-RS
Slovak (Slovakia)	sk-SK
Slovenian (Slovenia)	sl-Sl
Spanish (Mexico)	es-MX
Spanish (Spain)	es-ES
Swedish (Sweden)	sv-SE
Thai (Thailand)	th-TH
Turkish (Turkey)	tr-TR
Ukrainian (Ukraine)	uk-UA

In passive mode:

- 1. TMflow can receive Operator Command .
- 2. The hamburg button in the upper left corner of TMflow is changed to a hidden button. After clicking it, TMflow will be hidden.



IMPORTANT:

Before using operator commands, please make sure that the TMflow in passive mode and the dashboard has an OperatorActiveX component, otherwise, the Operator command will be unavailable.



IMPORTANT:

The operator command cannot be executed under the close area on the code editor.

A.1 ShowPage

Open and Show a specific Page of client TMflow.

Syntax1

OperatorActiveX->ShowPage = string

Parameter

string page name

page		Payload		
	Connect	"Connect"		
	Display	"ViewDisplay"		
	IO	"ViewIO"		
View	Simulation	"ViewSimulation"		
view	Status	"ViewStatus"		
	Vision Viewer	"ViewVisionViewer"		
	Force Control	"ViewForceControl"		
Run Setting		"RunSetting"		
	Wizard	"SettingWizard"		
	Vision Setting	"SettingVision"		
Setting	TCD Sotting	"SettingTcpTeaching"		
	TCP Setting	"SettingTcpManual"		
	IO Setup	"SettingIOSetupOutputDefaultValue"		

		"SettingIOSetupUserDefined"			
		"SettingSafetyPerformance"			
		"SettingSafetyHuman"			
	0.64	"SettingSafetyIO"			
	Safety	"SettingCartesianA"			
		"SettingCartesianB"			
		"SettingProjectSpeed"			
	Controller	"SettingController"			
	Speech	"SettingSpeech"			
	End Button	"SettingEndButton"			
	Component	"SettingComponent"			
	Operation Space	"SettingOperationSapce"			
	Command	"SettingCommand"			
	Connection: Modbus Slave RTU	"SettingConnectionModbusSlaveTCP"			
	Connection: Modbus Slave TCP	"SettingConnectionModbusSlaveRTU"			
	Connection: TCP/IP Server	"SettingConnectionTcpIp"			
	PosTrue Setting	"SettingPosTrue"			
	TMManger	"SettingTMManager"			
	Global Variable	"SettingGlobalVariable"			
	Text File Manger	"SettingTextFileManager"			
	Language	"SystemLanguage"			
	System Update	"SystemUpdate"			
	Group	"SystemGroup"			
	User Account	"SystemUserAccount"			
System	Network	"SystemNetwork"			
	Import/Export	"SystemImport"			
	Data Time	"SystemDateTime"			
	Administrator	"SystemAdministrator"			
	Network Service	"SystemNetworkService"			
	Backup\Restore	"SystemBackup"			
	Input/Display Devices	"SystemInputDisplayDevices"			
	Auto Remote Mode	"SystemAutoRemoteMode"			

Return

No Return

Note

```
OperatorActiveX->ShowPage = " Connect " // Show Connect Page
```

A.2 ShowFlow

Set the focus of the Project Flow on client TMflow.

Syntax1

OperatorActiveX->ShowFlow = string, string

Parameter

string[0]: Subflow Name (can be empty)

string[1]: Thread Name (can be empty)

Return

No Return

Note

Example 1

```
OperatorActiveX->ShowFlow = empty // Show Main flow
```

Example 2

OperatorActiveX->ShowFlow = "Subflow_1",empty // Show Subflow named Subflow_1

Example 3

```
OperatorActiveX->ShowFlow = empty,"Thread_1" // Show Thread named Thread_1
```

A.3 ShowVisionJob

Open and show a specific Vision job on the current Project on TMflow.

Syntax1

OperatorActiveX->ShowVisionJob = string

Parameter

string Vision Job Name

Return

TM Operator Platform User Manual

```
No Return
```

Note

```
OperatorActiveX->ShowVisionJob = "Vision_1" // Show TMflow and open vision Job named Vision_1
```

A.4 ShowTMflow

Show or Hide the client TMflow.

Syntax1

OperatorActiveX->ShowTMflow = bool

Parameter

bool true: Show TMflow.

false: Hide TMflow

Return

No Return

Note

Example 1

OperatorActiveX->ShowTMflow = true // Show TMflow

Example 2

OperatorActiveX->ShowTMflow = false // Hide TMflow

A.5 ConnectRobot

Connect to a robot through client TMflow.

Syntax1

bool = OperatorActiveX-> ConnectRobot [string]

Parameter

string IP Address. Ex 192.168.1.14

Return

bool true: Connect success

false: Connect fail

Note

```
string IP = "192.168.1.14" // Robot IP
bool Result = false //Define variable for receiving result
Result = OperatorActiveX-> ConnectRobot ->[IP] //Connect robot that 192.168.1.14
```

A.6 LogIn

Login onto the client TMflow.

Syntax1

OperatorActiveX->LogIn = string, string

Parameter

string[0]: User name

string[1]: Password

Return

No Return

Note

```
OperatorActiveX->LogIn = "Abner ", " 12345" //Login with Abner account
```

A.7 LogOut

Logout from the client TMflow.

Syntax1

OperatorActiveX->LogOut = bool

Parameter

bool true

Return

No Return

Note

```
OperatorActiveX->LogOut = true //Logout
```

A.8 GetControl

Get permission and to control a specific robot (already login) through client TMflow.

Syntax1

OperatorActiveX-> GetControl = bool

Parameter

bool true

Return

No Return

Note

OperatorActiveX-> GetControl = true //Get Control

A.9 CreateNewProject

Create a new Project on TMflow.

Syntax1

OperatorActiveX->CreateNewProject = string

Parameter

string New Project Name

Return

No Return

Note

OperatorActiveX->CreateNewProject = "Polishing" //Create new project named Polishing

A.10 OpenProject

Open a specific poject on client TMflow.

Syntax1

```
OperatorActiveX->OpenProject = string, bool
```

Parameter

string Project Name

bool

true: If the project was not closed properly last time, open the project as last saved version

false: If the project was not closed properly last time, open the project as last editing version.

(If there is this no parameter, it would be default as opening the project as the last saved version)

Return

No Return

Note

Example 1

OperatorActiveX->OpenProject = "Polishing" //Open Project named Polishing, If the project was not closed properly last time, open the project as last saved version.

Example 2

OperatorActiveX->OpenProject = "Polishing",false //Open Project named Polishing, If the project was not closed properly last time, open the project as last editing version.

A.11 SaveProject

Save the current Project on TMflow.

Syntax1

OperatorActiveX->SaveProject = string

Parameter

string Project Name

Return

No Return

Note

OperatorActiveX->SaveProject = "Polishing"//Overwrite current project or save as the currently project as a new name.

A.12 GetProjectList

Get the list of Projects from TMflow.

Syntax1

list = OperatorActiveX->GetProjectList

Parameter

No input value

Return

list Project name list •

Note

list Project_List = string //Define variable for receiving result
Project_List->AddValue = OperatorActiveX->GetProjectList //Get project list

A.13 IsProjectCloseNormal

Check if the Project is closed properly.

Syntax1

bool OperatorActiveX-> IsProjectCloseNormal->[string]

Parameter

string Project Name

Return

bool true: normal close

false: not normal close

Note

bool Project_Status = false //Define variable for receiving result

Project_Status = OperatorActiveX-> IsProjectCloseNormal->["Project"] //Get project closed

properly

A.14 IsProjectModified

Check if the current Project on client TMflow being modified without saving yet.

Syntax1

bool = OperatorActiveX-> IsProjectModified

Parameter

No input value

Return

bool true project edited and not save false project not edited or saved

Note

```
bool Modify_Status = false //Define variable for receiving result
```

```
Modify_Status = OperatorActiveX-> IsProjectModified //Check if the current project modified
```

without saving yet

A.15 GetTCPList

Get the list of Tools from TMflow.

Syntax1

list = OperatorActiveX->GetTCPList

Parameter

No input value

Return

list string TCP name list

Note

list = TCPList //Define variable for receiving result

TCPList->AddValue = OperatorActiveX->GetTCPList //Get TCP list

A.16 GetTCPTeachCoordinates

Get the TCP Teach Value of a specific tool.

Syntax1

array GetTCPTeachCoordinates->[string]

Parameter

string Designated TCP name

Return

array string, string, string, string, string

TCP Coordinate: [X, Y, Z, RX, RY, RZ]

Note

array TCP_Value = string, string, string, string, string, string //Define variable for receiving result

TCP_Value = OperatorActiveX-> GetTCPTeachCoordinates ->["TCP_1"] //Get the Coordinate Value of TCP_1 that its X, Y, Z, RX, RY, RZ.

A.17 ChangeTCPTeachValue

Change the TCP TeachValue of a specific tool.

Syntax1

OperatorActiveX-> ChangeTCPTeachValue= string, array

Parameter

string	TCP Name,
array	float, float, float, float, float, float
	Value of TCP Parameter. X, Y, Z, RX, RY, RZ

Return

No Return

Note

array TCP_Value = float, float, float, float, float, float, float //Define variable for receiving result

TCP_Value = 0,0,0,1,0,10 //Define variable for change value of TCP parameter. X, Y, Z, RX, RY,

RZ

OperatorActiveX->ChangeTCPTeachValue = "TCP_1",TCP_Value // Change TCP value of named TCP_1's X, Y, Z, RX, RY, RZ to 200,100,300,30,60,90

A.18 CreateTCP

Create a new Tool. TM Operator Platform User Manual

Syntax1

```
OperatorActiveX->CreateTCP = string, array
```

Parameter

string	TCP Name
array	string, string, string, string, string, string
	Value of TCP Parameter. X, Y, Z, RX, RY, RZ

Return

No Return

Note

array TCP_Value = string, string, string, string, string, string //Define variable for receiving result TCP_Value = 0, 0, 180, 0, 0, 0 //Define variable for change value of TCP parameter. X, Y, Z, RX,

RY, RZ

OperatorActiveX->CreateTCP = "TCP_2",TCP_Value //Create new TCP named TCP_2, and set it's value X, Y, Z, RX, RY, RZ as 0, 0, 180, 0, 0, 0

A.19 GetCurrentTCP

Get the Name of the current Tool.

Syntax1

string = OperatorActiveX->GetCurrentTCP

Parameter

No input value

Return

string TCP Name

Note

string TCP_Name = empty //Define variable for receiving result
TCP_Name = OperatorActiveX->GetCurrentTCP // Get the Current TCP name

A.20 GetTCPTeachMass
Get the Teach Mass of aTool.

Syntax1

string = OperatorActiveX->GetTCPTeachMass->[string]

Parameter

string TCP Name

Return

string Mass

Note

```
TCP_Name = OperatorActiveX-> GetTCPTeachMass ->[TCP_1] // Get the TCP named TCP_1's
```

mass

A.21 ChangeTCPTeachMass

Change the mass setting of a specific tool.

Syntax1

OperatorActiveX->ChangeTCPTeachMassCenter = string, string

Parameter

string TCP Name

string Value of TCP mass

Return

No Return

Note

```
OperatorActiveX->ChangeTCPTeachMass = "TCP_1","4" //Change TCP_1's mass to 4kg
```

A.22 GetTCPTeachInertia

Get the Inertia value of a Tool.

Syntax1

TM Operator Platform User Manual

array = OperatorActiveX-> GetTCPTeachInertia ->[string]

Parameter

string TCP Name

Return

array string, string, string string[0]: Ixx string[1]: Iyy string[2]: Izz

Note

```
array TCP_Value = string, string, string //Define variable for receiving result
```

```
TCP_Value = OperatorActiveX-> GetTCPTeachInertia ->["TCP_1"] //Get TCP_1's Inertia value
```

A.23 ChangeTCPTeachInertia

Change the Inertia setting of a specific tool.

Syntax1

OperatorActiveX-> ChangeTCPTeachInertia = string, array

Parameter

string TCP Name

array string, string, string

string[0]: Ixx

string[1]: lyy

string[2]: Izz

Return

No Return

Note

```
array TCP_Value = "0","0","0" //Define variable for change value of TCP parameter (Ixx, Iyy,
```

lzz)

```
OperatorActiveX-> ChangeTCPTeachInertia = "TCP_1", TCP_Value //Change TCP_1's Inertia (Ixx, Iyy, Izz) to 0,0,0
```

A.24 ChangeTCPTeachMassCenter

Change the Mass Center setting of a specific tool.

Syntax1

```
OperatorActiveX->ChangeTCPTeachMassCenter = string, array
```

Parameter

string	TCP name	
--------	----------	--

array string, string, string, string, string, string

Value of TCP Parameter. X, Y, Z, RX, RY, RZ

Return

No Return

Note

array TCP_Value = "0","0","0","0","0","0" //Define variable for change value of TCP parameter.

OperatorActiveX->ChangeTCPTeachMassCenter = "TCP_1",TCP_Value //Change TCP_1's Mass

```
Center (X, Y, Z, RX, RY, RZ) to 0,0,0,0,0
```

A.25 GetTCPTeachMassCenter

Get the Mass Center parameters of a Tool.

Syntax1

array = OperatorActiveX->GetTCPTeachMassCenter->[string]

Parameter

string TCP Name

Return

array string, string, string, string, string

Value of TCP Teach Mass Center Parameter. X, Y, Z, RX, RY, RZ

Note

array TCP_Value = string, string, string, string, string, string, string //Define variable for receiving result TCP_Value = OperatorActiveX-> GetTCPTeachMassCenter->["TCP_1"] //Get the TCP Teach

Mass Center

A.26 GetLocalVariableList

Get the list of local Variables of the current Project on TMflow.

Syntax1

list = OperatorActiveX->GetLocalVariableList

Parameter

No input value

Return

list string Local variable name list

Note

list VariableList = string //Define variable for receiving result

```
VariableList- >AddValue = OperatorActiveX->GetLocalVariableList //Get variable name list from cruuent project.
```

A.27 GetGlobalVariableList

Get the list of all global Variables.

Syntax1

list = OperatorActiveX->GetGlobalVariableList

Parameter

No input value

Return

list string Global variable name list

Note

list GlobalVariableList = string //Define variable for receiving result

GlobalVariableList-> >AddValue = OperatorActiveX-> GetGlobalVariableList //Get Global variable

name list

A.28 CreateVariable

Create a new Variable.

Syntax1

OperatorActiveX->CreateVariable = string, string, string, string

Parameter

string[0] local/global

string[1] Variable Type ("int","int[]"...)

string[2] Variable Name (Don't Need Prefix word)

string[3] Variable Value

Return

No Return

Note

OperatorActiveX->CreateVariable = "local","int","Force","20" //In the current project, create an int type variable named var Force and endure its value is 20.

A.29 GetVariableValue

Get the value a Variable.

Syntax1

array OperatorActiveX->GetVariableValue->[array]

Parameter

array string, string, string.... Variable Name (Need Prefix word)

Return

array string, string, string.... Variable Value

Note

```
Example 1
```

string Get_Force = empty //Define variable for receiving result

```
Get_Force = OperatorActiveX->GetVariableValue->["g_Force"] // Get the value of g_Force
```

Example 2

array Get_Force_Speed = empty, empty //Define variable for receiving result

Get_Force_Speed = OperatorActiveX->GetVariableValue->["g_Force", "g_Speed"] // Get the

value of g_Force

A.30 ChangeLocalVariableValue

Change the value of a local Variable.

Syntax1

OperatorActiveX->ChangeLocalVariableValue = string, string

Parameter

string[0]: Variable Name (Need Prefix word)
string[1]: Variable Value

Return

No Return

Note

OperatorActiveX->ChangeLocalVariableValue = "var_Force","20" // Change the exist variable named var_Force to 20.

A.31 ChangeGlobalVariableValue

Change the value of a global Variable.

Syntax1

OperatorActiveX->ChangeLocalVariableValue = string, string

Parameter

string[0]: Variable Name (Need Prefix word)

string[1]: Variable Value

Return

No Return

Note

OperatorActiveX->ChangeGlobalVariableValue ="g_Force","20" // Change the exist golbal

variable named var_Force to 20.

A.32 GetVariableType

Get the Type of a Variable.

Syntax1

```
array = OperatorActiveX->GetVariableType->[array]
```

Parameter

array string, string, string.... Variable Name (Need Prefix word)

Return

array string, string, string.... Variable Type

Note

Example 1

string Get_Var_Type = empty //Define variable for receiving result

```
Get_Var_Type = OperatorActiveX->GetVariableType->["g_Force"] //Get the type of g_Force
```

Example 2

array Get_Force_Speed_Type = empty, empty //Define variable for receiving result

```
Get_Force_Speed_Type = OperatorActiveX->GetVariableType->["g_Force", "g_Speed"]
```

//Get the type of g_Force

A.33 GetPointList

Get the Point List of the current Project on TMflow.

Syntax1

list OperatorActiveX->GetPointList

Parameter

No input value

Return

list string point name list

Note

list Point_List = string //Define variable for receiving result

Point_List->AddValue = OperatorActiveX->GetPointList //Get the point list

A.34 CreatePointByJoints

Create a Point defined by Joint Angles.

Syntax1

OperatorActiveX->CreatePointByJoints = string, string, string, string

Parameter

string[0] Point Name

string[1] Value of Coordinate. Format: "J1,J2,J3,J4,J5,J6"

string[2] Reference Coordinate name

string[3] Tool Name

Return

No Return

Note

OperatorActiveX->CreatePointByJoints = "Above_Point","0,90,0,0,90,0","RobotBase","NOTOOL" //Create a new point named Above_Point recorded on robot base and with NOTOOL, which robot angle is 0,90,0,0,90,0

A.35 GetPointTeachPose

Get the robot pose configurations of a Point.

Syntax1

array = OperatorActiveX-> GetPointTeachPose -> [string]

Parameter

string Point Name

Return

arraystring, string, stringstring[0]: Config1string[1]: Config2string[2]: Config3

Note

array Point_Pose = string, string, string //Define variable for receiving result

Point_Pose = OperatorActiveX-> GetPointTeachPose ->["Above_Point"] //Get point named Above_Point pose.

A.36 ChangePointBase

Change the base of a Point to another base.

Syntax1

OperatorActiveX->ChangePointBase = string, string

Parameter

string[0]: Point Name

string[1]: Base Name

Return

No Return

Note

OperatorActiveX->ChangePointBase = "Above_Point","RobotBase" //Change the Above_Point's base to RobotBase.

A.37 ChangePointTeachPose

Change the robot pose configurations of a Point.

Syntax1

OperatorActiveX-> Changecse = string, string, string, string

Parameter

string[0]: Point Name string[1]: Config0 string[2]: Config1 string[2]: Config2 **Return** No Return

Note TM Operator Platform User Manual OperatorActiveX-> ChangePointTeachPose = "Above_Point","0","2","4" //Change the

Above_Point's pose to 0,2,4

A.38 GetPointTCPName

Get the name of the Tool of a Point.

Syntax1

string = OperatorActiveX->GetPointTCPName->[string]

Parameter

string Point Name

Return

string TCP Name

Note

string TCP_Name = empty //Define variable for receiving result

```
TCP_Name = OperatorActiveX->GetPointTCPName->["Above_Point"] //Get the tool name that
```

Above_Point used

A.39 GetPointBaseName

Get the name of the Base of a Point.

Syntax1

string = OperatorActiveX->GetPointBaseName->[string]

Parameter

string Point Name

Return

string Tcp Name

Note

string Base_Name = empty //Define variable for receiving result

Base_Name = OperatorActiveX->GetPointBaseName->["Above_Point"] //Get the base name that

Above_Point used

A.40 GetBaseList

Get the list of Base from current Project on TMflow.

Syntax1

list = OperatorActiveX->GetBaseList

Parameter

No input value

Return

list string base name list

Note

list Get_Base_Name = string //Define variable for receiving result

```
Get_Base_Name->AddValue = OperatorActiveX->GetBaseList //Get the base name list from
```

cruuent project.

A.41 CreateNewBase

Create a New Base onto the current Project on TMflow.

Syntax1

OperatorActiveX->CreateNewBase = string,array

Parameter

string	Base name
array	float, float, float, float, float, float
float [0]: X	
float [1]: Y	
float [2]: Z	
float [3]: RX	
float [4]: RY	
float [5]: RZ	
Return	

No Return

Note

array Base_Value = float, float, float, float, float, float, float //Define variable for receiving result Base_Value= 0,0,0,0,0,0 //Define the variable for create base. OperatorActiveX->CreateNewBase = "Base_1", Base_Value //Create new Base named Base_1, and set it's value X, Y, Z, RX, RY, RZ as 0,0,0,0,0

A.42 GetCurrentBaseName

Get the Name of the current Base.

Syntax1

string = OperatorActiveX->GetCurrentBaseName

Parameter

No input value

Return

string Coordinate name

Note

String Current_Base_Name = empty //Define variable for receiving result

Current_Base_Name = OperatorActiveX->GetCurrentBaseName //Get the current base name

A.43 GetBaseTeachValue

Get the parameters of the current Base.

Syntax1

array OperatorActiveX-> GetBaseTeachValue ->[string]

Parameter

string Base name

Return

array string, string, string, string, string, string

Value of Coordinate Parameter. X, Y, Z, RX, RY, RZ

Note

Example 1

array Base_Value = string, string, string, string, string, string //Define variable for

receiving result

Base_Value = OperatorActiveX-> GetBaseTeachValue ->["Base_1"] //Get the Cordinate Value of Base_1 that its X, Y, Z, RX, RY, RZ : 200,100,300,30,60,90

Example 2

array Base_Value = string, string, string, string, string, string //Define variable for receiving result

Base_Value = OperatorActiveX-> GetBaseTeachValue ->["Base_1"] //Get the Cordinate Value of Base_1 that its X, Y, Z, RX, RY, RZ : 200,100,300,30,60,90

A.44 ChangeBaseTeachValue

Change the TeachValue of a custom Base of the current Project on TMflow.

Syntax1

OperatorActiveX->ChangeBaseTeachValue = string, array

Parameter

string	Base name	
array	float, float, float, float, float, float	
string[0]	Х	
string[1]	Υ	
string[2]	Z	
string[3]	RX	
string[4]	RY	
string[5]	RZ	

Return

No Return

Note

array Base_Value = float, float, float, float, float, float

Base_Value= 0,0,0,0,0,0

OperatorActiveX->ChangeBaseTeachValue = "Base_1", Base_Value //In current project, chanage Base_1's value of base as 0,0,0,0,0,0

A.45 GetBaseType

Get the type of the a specific Base.

Syntax1

string = OperatorActiveX->GetBaseType->[string]

Parameter

string Base name

Return

string Base type

- "R": Robot Base
- "C": Customer Base
- "V": Vision Base

Note

string Base_Type = empty //Define variable for receiving result

Base_Type = OperatorActiveX->GetBaseType->["Base_1"] //Get Base_1 Type

A.46 CreatePointByCoordinates

Create a Point with given TCP coordinates defined by a given Base.

Syntax1

```
OperatorActiveX->CreatePointByCoordinates = array
```

Parameter

- array string, string, string, string, string
- string[0] Point Name
- string[1] End Tool Coordinate("X,Y,Z,Rx,Ry,Rz")
- string[2] Configs(eg: "024")
- string[3] Base Name
- string[4] Tcp Name

Return

No Return

Note

array New_Point_Parameters = string, string, string, string //Define variable for create point New_Point_Parameters = "P88","200,170,230,0,0,0","024","RobotBase","NOTOOL" //Define point parameters

OperatorActiveX->CreatePointByCoordinates = New_Point_Parameters //Create point

A.47 ChangePointTeachValue

Change point teach value

Syntax1

ChangePointTeachValue ->ChangePointTeachValue = string, array

Parameter

string[0] Point Name

andy noul, noul, noul, noul, noul, nou

- float [0] X axis in End Tool Coordinate
- float [1] Y axis in End Tool Coordinate
- float [2] Z axis in End Tool Coordinate
- float [3] RX axis in End Tool Coordinate
- float [4] RY axis in End Tool Coordinate
- float [5]: RZ axis in End Tool Coordinate

Return

No Return

Note

OperatorActiveX->ChangePointTeachValue = "P1",-466,374,278,175,1,-127 //Point point named P1's coordinate (X, Y, Z, RX, RY, RZ) as -466,374,278,175,1,-127

A.48 ChangePointTool

Re-Recorder Point on Tool

Syntax1

OperatorActiveX->ChangePointTool = string, string, string

Parameter

string[0]: Point Name

string[1]: Tcp Name

string[2]: "0" => keep pose, "1"=> keep path

No Return

Note

```
OperatorActiveX->ChangePointTool = "P1", "TCP_2", "0" //Change P1 TCP to TCP_2 and keep its pose
```

A.49 GetPointTeachValue

Get point position of value on Tool.

Syntax1

```
array = OperatorActiveX->GetPointTeachValue->[string]
```

Parameter

string Coordinate name

Return

- array float, float, float, float, float, float
- float [0]: X axis in End Tool Coordinate
- float [1]: Y axis in End Tool Coordinate
- float [2]: Z axis in End Tool Coordinate
- float [3]: RX axis in End Tool Coordinate
- float [4]: RY axis in End Tool Coordinate
- float [5]: RZ axis in End Tool Coordinate

Note

array Point_Value = float , float, float, float, float, float //Define variable for receiving result Point_Value = OperatorActiveX->GetPointTeachValue->["P1"] //Get P1 teach value

A.50 SetCustomFreeBot

Custom FreeBot Setting.

Syntax1

OperatorActiveX->SetCustomFreeBot = string, array

Parameter

string Mode ; "True" => by base, "False" => by Tool.

array string, string, string, string, string, string

string[0] Free X axis or not. "True" =>Free. "False"=>Locked

string[1] Free Y axis or not. "True" =>Free. "False"=>Locked

string[2] Free Z axis or not. "True"=>Free. "False"=>Locked

string[3] Free RX axis or not. "True" =>Free. "False"=>Locked

string[4] Free RY axis or not. "True" =>Free. "False"=>Locked

string[5] Free RZ axis or not. "True" =>Free. "False"=>Locked

Return

No Return

Note

OperatorActiveX->SetCustomFreeBot = "True","True","True","True","False","False","False "," False" //Set the robot can moving by hand guiding only with X, Y, Z axis

A.51 MoveLineByToolAxis

Jog the robot according to the current Tool base (relative motion).

Syntax1

OperatorActiveX->MoveLineByToolAxis = string, array

Parameter

string:	Speed in percentage (float). e.g. 1.5% =>"0.015"
array	string, string, string, string, string
string[0] X.	
string[1] Y	
string[2] Z	
string[3] RX	
string[4] RY	
string[5] RZ	
Return	
No Retur	n

Note

OperatorActiveX->MoveLineByToolAxis = "0.015","321","551","251","171","0","161" //Move the robot X, Y, Z, RX, RY, RZ to 321, 551 ,251 ,171 ,0, 161 with current tool, and set moving speed is 1.5%. (Need to make up the start/stop beacon function)

A.52 MoveLineByCoordinates

Jog the robot with given coordinates according to the current Base (relative motion)

Syntax1

```
OperatorActiveX->MoveLineByCoordinates= string, array
```

Parameter

string	Speed in percentage (float). e.g. 1.5% =>"0.015"
array	string, string, string, string, string
string[0]	X
string[1] `	Y
string[2]	Ζ
string[3]	RX
string[4]	RY
string[5]	RZ
Return	

No Return

Note

OperatorActiveX->MoveLineByCoordinates = "0.015","321","551","251","171","0","161" //Move the robot X, Y, Z, RX, RY, RZ to 321, 551 ,251 ,171 ,0, 161 with current Base, and set moving speed is 1.5%. (Need to make up the start/stop beacon function)

A.53 MovePTP

Jog the robot with PTP motion.

Syntax1

OperatorActiveX->MovePTP = string, array

Parameter

stringSpeed in percentage (float). e.g.1.5% => 0.015 =>"0.015"arraystring, string, string, string, stringstring[0]: J0 (Joint angle)string[1]: J1 (Joint angle)

string[2]: J2 (Joint angle)

string[3]: J3 (Joint angle)

string[4]: J4 (Joint angle)

string[5]: J5 (Joint angle)

Return

No Return

Note

OperatorActiveX->MovePTP = "0.015","90","0","0","0","0","0","0"/Move the robot J1, J2, J3, J4, J5, J6 to 90, 0, 0, 0, 0, 0, and set moving speed is 1.5%. (Need to make up the start/stop beacon function)

A.54 StartBeacon

Start the Beacon signal for Move command.

Syntax1

OperatorActiveX->StartBeacon = bool

Parameter

bool true

Return

No Return

Note

OperatorActiveX->StartBeacon = true

A.55 StopBeacon

Stop the Beacon signal for Move command.

Syntax1

OperatorActiveX-> StopBeacon = bool

Parameter

bool true

Return

No Return

OperatorActiveX-> StopBeacon = true

A.56 SetPayload

Set Payload value of the robot.

Syntax1

OperatorActiveX->SetPayload = float

Parameter

float Payload

Return

No Return

Note

OperatorActiveX->SetPayload = 2 //Set current payload to 2kg

A.57 ChangeCurrentBase

Change the current Base.

Syntax1

OperatorActiveX->ChangeCurrentBase = string

Parameter

string Base Name

Return

No Return

Note

OperatorActiveX->ChangeCurrentBase = "Base_1" //Change current base to Base_1

A.58 ChangeCurrentTool

Change the current Tool.

Syntax1

```
OperatorActiveX->ChangeCurrentTool = string
```

Parameter

string TCP Name

Return

No Return

Note

```
OperatorActiveX->ChangeCurrentTool = "TCP_1" //Change current tool to TCP_1
```

A.59 ChangeFreeBotSensor

Change the Sensing mode of FreeBot.

Syntax1

OperatorActiveX->ChangeFreeBotSensor = bool, string

Parameter

bool	FreeBot Joint setting		
	true=>	FreeBot use force sensor	
	false=>	FreeBot use joint setting	

string Sensor name

Return

No Return

Note

Example 1

```
OperatorActiveX->ChangeFreeBotSensor = false,"" //Using FreeBot use sensor
```

Example 2

```
OperatorActiveX->ChangeFreeBotSensor = true,"fts_Force" //Using force sensor device
```

named fts_Force.

A.60 GetVPointTeachValue

Read the initial point of a vision job

Syntax1

array = OperatorActiveX->GetVPointTeachValue->[string]

Parameter

string Vison Job Name

Return

array float, float, float, float, float, float float [0]: X. float [1]: Y float [2]: Z float [3]: RX(float [4]: RY(float [5]: RZ(

Note

array Point_Value = float, float, float, float, float, float, float //Define variable for receiving result Point_Value = OperatorActiveX->GetVPointTeachValue->["Vision_P1"] //Get Vision job initial point

A.61 GetVPointBase

Read the Coordinate name of the initial point of a vision job

Syntax1

string = OperatorActiveX->GetVPointBase->[string]

Parameter

string Vison Job Name

Return

string Base Name

Note

string Base_Name = empty //Define variable for receiving result
Base_Name = OperatorActiveX->GetVPointBase->["Vision_P1"] //Get vision job base

A.62 ReadDigitalInput

Read the Digital input of the robot.

Syntax1

OperatorActiveX->ReadDigitalInput = string, string, string

Parameter

string [0] Device type.

Controller Box = "16"

End Module = "64"

Ext. Module ="80"

string [1] Serial number of the device of the same type. 0 => First Device of the same type.

string [2] Digit Input channel number. Number of Digit Input

Return

string Digital input value "True" => High "False"=> Low

Note

string Digitial_intput = empty //Define variable for receiving result

```
Digitial_intput = OperatorActiveX->ReadDigitalInput = "16","0","2" //Read DI0 status from
```

controller box

A.63 ReadDigitalOutput

Read the Digital output of the robot.

Syntax1

string = OperatorActiveX-> ReadDigitalOutput = string, string, string

Parameter

string [0] Device type.

Controller Box = "16"

End Module = "64"

Ext. Module ="80"

string [1] Serial number of the device of the same type. 0 => First Device of the same type.

string [2] Digit Input channel number. Number of Digit Output

Return

string Digital Output value "True" => High "False"=> Low

Note

string Digitial_Output = empty //Define variable for receiving result

```
Digitial_Output = OperatorActiveX-> ReadDigitalOutput = "16","0","0"` //Read DO0 status from controller box
```

A.64 WriteDigitalOutuput

Set the Digital output of the robot.

Syntax1

OperatorActiveX->WriteDigitalOutuput = string, string, string, string

Parameter

string [0]: Device type.

Controller Box = "16" End Module = "64" Ext. Module = "80"

string [1]: Serial number of the device of the same type. 0 => First Device of the same type.

string [2]: Digit Output channel number. Number of Digit Output

string [3]: 1 => High, 0 => Low

Return

No Return

Note

OperatorActiveX->WriteDigitalOutuput ="64","0","0","0" //Write DO0 on End Module to High.

A.65 ReadAnalogInput

Read the Analog Input of the robot.

Syntax1

string = OperatorActiveX->ReadAnalogInput = string, string, string

Parameter

string [0] Device type.

Controller Box = "16"

```
End Module = "64"
```

Ext. Module ="80"

string [1] Serial number of the Device of the same type. 0 => First Device of the same type.

string [2] Analog Input channel number. Number of Analog Input

Return

string Analog input(float). -10.24~10.24 Voltage

Note

string Analog_Input = empty //Define variable for receiving result

```
Analog_Input = OperatorActiveX->ReadAnalogInput = "16","0","0" //Read Al0 value from controller
```

box

A.66 ReadAnalogOutput

Read the Analog output setting of the robot.

Syntax1

string = OperatorActiveX->ReadAnalogOutput = string, string, string

Parameter

```
string [0] Device type.
```

```
Controller Box = "16"
```

End Module = "64"

Ext. Module ="80"

string [1] Serial number of the Device of the same type. 0 => First Device of the same type.

string [2] Analog Output channel number. Number of Analog Output

Return

string Analog input. -10.0~10.0 Voltage

Note

string Analog_Output = empt

OperatorActiveX->ReadAnalogOutput ="16","0","0" //Read Al0 value from controller box

A.67 WriteAnalogOutput

Set the Analog output of the robot.

Syntax1

OperatorActiveX->WriteAnalogOutput = string, string, string, string

Parameter

string[0] Device type.

Controller Box = "16"

End Module = "64"

Ext. Module ="80"

string [1] Serial number of the Device of the same type. 0 => First Device of the same type. string [2] Analog Output channel number. Number of Analog Output

string [3] Value(float). -10.0 ~ 10.0

Return

No Return

Note

OperatorActiveX->WriteAnalogOutput ="64","0","0","5" //Write AO0 on End Module to 5V.

A.68 GetCoordinatesByRobotBase

Read Coordinate of Robot TCP which is relative to Robot Coordinate (RobotBase)

Syntax1

array = OperatorActiveX->GetCoordinatesByRobotBase

Parameter

No input value

Return

array float, float, float, float, float, float

float[0]: X

float[1]: Y

float[2]: Z

float[3]: RX

float[4]: RY

float[5]: RZ

Note

array Current_Coordinate = float, float, float, float, float, float, float, float //Define variable for receiving result Current_Coordinate = OperatorActiveX->GetCoordinatesByRobotBase //Get coordinates by robot

base.

A.69 GetCoordinatesByCurrentBase

Read Coordinate of Robot TCP which is relative to tool Coordinate (Current tool)

Syntax1

array OperatorActiveX->GetCoordinatesByCurrentBase

Parameter

No input value

Return

array float, float, float, float, float, float

float[0]: X

float[1]: Y

float[2]: Z

float[3]: RX

float[4]: RY

float[5]: RZ

Note

array Current_Coordinate = float, float, float, float, float, float, float //Define variable for receiving result Current_Coordinate = OperatorActiveX->GetCoordinatesByCurrentBase //Get coordinates by

current Base.

A.70 GetCurrentJointAngles

Get the current Joint angels of the robot.

Syntax1

array OperatorActiveX->GetCurrentJointAngles

Parameter

No input value

Return

array float, float, float, float, float, float

float[0]: J1

float[1]: J2 float[2]: J3 float[3]: J4 float[4]: J5

float[5]: J6

Note

array Current_Joint = float, float, float, float, float, float, float //Define variable for receiving result Current_Joint = OperatorActiveX->GetCurrentJointAngles //Get current joint angles

A.71 IsRobotLink

Check if the robot is linked or not.

Syntax1

bool = OperatorActiveX->IsRobotLink

Parameter

No input value

Return

string true: Linked. false: Not Linked.

Note

bool Robot_Link =false //Define variable for receiving result
Robot_Link = OperatorActiveX->IsRobotLink //Get robot link status

A.72 IsRobotInError

Check if the robot is in error status.

Syntax1

bool = OperatorActiveX-> IsRobotInError

Parameter

No input value

Return

bool true: Error. false: No Error.

Note

TM Operator Platform User Manual

bool Robot_Error =false //Define variable for receiving result
Robot_Error = OperatorActiveX->IsRobotInError //Get robot error or not

A.73 IsProjectRunning

Check if the robot is currently running with a Project.

Syntax1

bool = OperatorActiveX-> IsProjectRunning

Parameter

No input value

Return

bool true: Project is running now. false: No project running.

Note

bool Project_Running =false //Define variable for receiving result

Project_Running = OperatorActiveX->IsProjectRunning //Get project running or not

A.74 IsProjectEditing

Check if the robot is current under Project Editing.

Syntax1

bool = OperatorActiveX->IsProjectEditing

Parameter

No input value

Return

bool true: Project is Editing now. false: No project Editing.

Note

bool Project_Editing = false //Define variable for receiving result
Project_Editing = OperatorActiveX->IsProjectEditing //Get project edit or not

A.75 IsProjectPaused

Check if the robot status is in Pause or not.

Syntax1

bool OperatorActiveX->IsProjectPaused

Parameter

No input value

Return

bool true: Project is paused now. false: No project paused.

Note

bool Project_Paused =false //Define variable for receiving result
Project_Paused = OperatorActiveX->IsProjectPaused //Get project pause or not

A.76 IsControlled

Check if the robot permission is being accessed.

Syntax1

bool = OperatorActiveX-> IsControlled

Parameter

No input value

Return

bool true: Got Contol. false: Not getting control.

Note

bool Is_Controlled = false //Define variable for receiving result

Is_Controlled = OperatorActiveX->IsControlled // Get robot is controlled or not

A.77 GetCurrentPose

Get robot current pose.

Syntax1

array = OperatorActiveX-> GetCurrentPose

Parameter

No input value

Return

array string, string, string

string [0]: 0 – Right Arm, 1 – Left Arm

string [1]: 2 - Above Elbow, 3 - Below Elbow

string [2]: 4 - Up Wrist, 5 - Down Wrist

Note

array Current_Pose = string, string, string //Define variable for receiving result Current_Pose = OperatorActiveX-> GetCurrentPose //Get robot current pose

A.78 ShowOperationSpace

Open the "OperationSpace" Setting Page of TMflow.

Syntax1

OperatorActiveX->ShowOperatorSpace = bool

Parameter

bool true

Return

No Return

Note

OperatorActiveX->ShowOperationSpace = true //Show operation space page in project page

A.79 ShowModbusDev

Open the "ModbusDev" Setting Page of TMflow.

Syntax1

OperatorActiveX->ShowModbusDev = bool

Parameter

bool true

Return

No Return

Note

OperatorActiveX->ShowModbusDev = true //Show ModbusDev page in project page

A.80 ShowSetIOWhileProjectError

Open the "SetIOWhileProjectError" Setting Page of TMflow.

Syntax1

OperatorActiveX->ShowSetIOWhileProjectError = bool

Parameter

bool true

Return

No Return

Note

OperatorActiveX->ShowSetIOWhileProjectError = true //Show ShowSetIOWhileProjectError page in project page

A.81 ShowSetIOWhileProjectStop

Open the "SetIOWhileProjectStop" Setting Page of TMflow.

Syntax1

OperatorActiveX->ShowSetIOWhileProjectStop = bool

Parameter

bool true

Return

No Return

Note

OperatorActiveX->ShowSetIOWhileProjectStop = true //Show SetIOWhileProjectStop page in

project page

A.82 ShowStopWatch

Open the "Stop Watch" Setting Page of TMflow.

Syntax1

OperatorActiveX->ShowStopWatch = bool

Parameter

bool true

Return

No Return

Note

OperatorActiveX->ShowStopWatch = true //Show Stop Watch page in project page

A.83 ShowFTSensor

Open the "FT Sensor" Setting Page of TMflow.

Syntax1

OperatorActiveX->ShowFTSensor = bool

Parameter

bool true

Return

No Return

Note

OperatorActiveX->ShowFTSensor = true //Show FT Sensor page in project page

A.84 ShowView

Open the "View" (Camera) page of TMflow

Syntax1

OperatorActiveX->ShowView = bool

Parameter

bool true

Return

No Return

Note

OperatorActiveX->ShowView = true //Show View (Camera) page in project page

A.85 ShowSevrialPort

Open the "Serial Port" Setting Page of TMflow.

Syntax1

OperatorActiveX->ShowSerialPort = bool

Parameter

bool true

Return

No Return

Note

OperatorActiveX->ShowSerialPort = true //Show Serial Port page in project page

A.86 ShowPathGenerate

Open "Path Generate" Setting Page of TMflow.

Syntax1

OperatorActiveX->ShowPathGenerate = bool

Parameter

bool true

Return

No Return

Note

OperatorActiveX->ShowPathGenerate =true //Show Path Generate page in project page

A.87 ImportTCP

Import TCP file

Syntax1

OperatorActiveX->ImportTCP = string, string, string

Parameter

string[0]: Device path of file // Share folder string[1]: Robot Name string[2]: TCP Name

Return

No Return

Note

OperatorActiveX->ImportTCP = "\A11080051NB\Folder","TM3630_BC1924004","TCP_1" //Import TCP named TCP_1 from the shared folder, and the TCP.zip location is under the \\A11080051NB\Folder\Export\TM3630_BC1924004

A.88 ImportTextFile

Import Text File

Syntax1

OperatorActiveX->ImportTextFile = string, string, string

Parameter

string[0]: Device path of file string[1]: Robot Name string[2]: Text File Name

Return

No Return

Note

OperatorActiveX->ImportTextFile = "\A11080051NB\Folder","TM3630_BC1924004","Layout.txt" //Import text file named Layout.txt from the shared folder, and the Layout.txt location is under the \A11080051NB\Folder\Export\TM3630 BC1924004\TextFiles

A.89 ShutDown

Shutdown the robot through TMflow.

Syntax1

OperatorActiveX-> ShutDown = string

Parameter

string Target robot IP

Return

No Return

Note

```
OperatorActiveX-> ShutDown = "192.168.132.141" //Shutdown the robot that IP is
```

192.168.132.141

A.90 LeaveTMflow

Leave and close the client TMflow

Syntax1

OperatorActiveX-> LeaveTMflow = bool

Parameter

bool true

Return

No Return

Note

OperatorActiveX-> LeaveTMflow = true //Close flow.exe on client-side

A.91 Iswork

Judgment Operator command effective or not

Syntax1

bool = OperatorActiveX-> Iswork
Parameter

No input value

Return

bool true: effective ; false: Invalid

Note

bool Command_Result =false //Define variable for receiving result Command_Result = OperatorActiveX-> Iswork //Get command status

A.92 GetOperatorErrorCommand

Get the name of the Operator Error command where the last error occurred

Syntax1

string = OperatorActiveX->GetOperatorErrorCommand

Parameter

No input value

Return

string command name

Note

String Error_Command_Name = empty //Define variable for receiving result OperatorActiveX->GetOperatorErrorCommand //Get error command name

A.93 GetOperatorErrorcode

Get the current code of the OperatorActiveX component.

Syntax1

string = OperatorActiveX->GetOperatorErrorcode

Parameter

No input value

Return

string command name

```
Exception = 1
Incorrect type =2
Incorrect number of parameters = 3
Time out =4
No connect = 5
Packet In Abnorma = 6
```

Note

String Error_Code = empty //Define variable for receiving result OperatorActiveX->GetOperatorErrorCode//Get error code Note: Definition of Robot Configuration: [Config1, Config2, Config3]



Appendix B: TMstudio Command

The chapter will introduce the command of TMstudio when it's as the builder slave.

TMstudio startup parameters are as follows: TMflow [/S] [/L Lang]

- /S Assign TMstudio to be activated in Slave Mode (If this parameter not sent, it's Normal mode)
- /L Assign language
- /Page Specified page
- / File Specified file

The example below denotes the Dashboard starts TMstudio in passive mode, switches to the Palletizing Wizard page, and open the specified file.

callApplication "C:\Techman Robot\TMstudio","TMstudio.exe","-S -Page \"Palletizing Wizard\" -File \"TM Palletizing OP.xml\""

Language/region	Language/region tag
English (United States)	en-US
Chinese (Taiwan)	zh-TW
Chinese (PRC)	zh-CN
Japanese (Japan)	ja-JP
German (Germany)	de-DE
Spanish (Spain)	es-ES
French (France)	fr-FR
Italian (Italy)	it-IT
Korean (Korea)	ko-KR
Vietnamese (Viet nam)	vi-VN
Thai (Thailand)	th-TH
Polish (Poland)	pl-PL
Portuguese (Portugal)	pt-PT
Hungarian (Hungary)	hu-HU

Page	File
Workstation	Scene folder
TCP Generator	TCP file
Palletizing Wizard	Pallet project



IMPORTANT:

Naming is available with spaces but with the double quotes for tab launching, the scene file folders to open, the TCP files, the pallet projects, and the xmls.

In passive mode:

- 1. Removed the X button at top right of the TCP Wizard window and the Palletizing Wizard window during normal operation.
- 2. Prompts to close the window while generating TCP (.tcp) files or exporting palletizing files (.txt).
- 3. Displays the tab to be swtiched without showing others while switching tabs.



OMRON AUTOMATION AMERICAS HEADQUARTERS • Chicago, IL USA • 847.843.7900 • 800.556.6766 • automation.omron.com

OMRON CANADA, INC. • HEAD OFFICE Toronto, ON, Canada • 416.286.6465 • 866.986.6766 • automation.omron.com

OMRON ELECTRONICS DE MEXICO • HEAD OFFICE Ciudad de México • 52.55.5901.4300 • 01.800.386.6766 • mela@omron.com

OMRON ELECTRONICS DE MEXICO • SALES OFFICE San Pedro Garza García, N.L. • 81.12.53.7392 • 01.800.386.6766 • mela@omron. com

OMRON ELECTRONICS DE MEXICO • SALES OFFICE Eugenio Garza Sada,León, Gto • 01.800.386.6766 • mela@omron.com

Authorized Distributor:

OMRON ELETRÔNICA DO BRASIL LTDA • HEAD OFFICE São Paulo, SP, Brasil • 55 11 5171-8920 • automation.omron.com

OMRON ARGENTINA • SALES OFFICE Buenos Aires, Argentina • +54.11.4521.8630 • +54.11.4523.8483 mela@omron.com

OTHER OMRON LATIN AMERICA SALES +54.11.4521.8630 • +54.11.4523.8483 • mela@omron.com

Controllers & I/O

Machine Automation Controllers (MAC)
 Motion Controllers

Programmable Logic Controllers (PLC)
 Temperature Controllers
 Remote I/O

Robotics

Industrial Robots
 Mobile Robots

Operator Interfaces

• Human Machine Interface (HMI)

Motion & Drives

- Machine Automation Controllers (MAC)
 Motion Controllers
 Servo Systems
- Frequency Inverters

Vision, Measurement & Identification

Vision Sensors & Systems • Measurement Sensors • Auto Identification
Systems

Sensing

- Photoelectric Sensors Fiber-Optic Sensors Proximity Sensors
- Rotary Encoders
 Ultrasonic Sensors

Safety

Safety Light Curtains
 Safety Laser Scanners
 Programmable Safety Systems

- Safety Mats and Edges
 Safety Door Switches
 Emergency Stop Devices
- Safety Switches & Operator Controls Safety Monitoring/Force-guided Relays

Control Components

- Power Supplies Timers Counters Programmable Relays
- Digital Panel Meters
 Monitoring Products

Switches & Relays

- Limit Switches Pushbutton Switches Electromechanical Relays
- Solid State Relays

Software

Programming & Configuration
 Runtime

© 2021 Omron. All Rights Reserved.