

# AnyWireASLINK网络监视功能

## 样本画面说明书

三菱电机株式会社

## 关于样本的使用

在使用样本画面及其说明书等文件之前，请首先同意以下各项。

- (1) 只有正在使用或有意使用本公司产品的用户才能使用。
- (2) 本公司所提供的文件的知识产权归属本公司所有。
- (3) 禁止对本公司提供的文件进行窜改、转载、转让、销售。  
但是，可以将部分或全部内容用于用户制作的机器或系统内的本公司产品上。也可以转载、复制、引用、重新排版于本公司用户制作的规格书、设计书、嵌入式产品的使用说明书中。
- (4) 使用本公司提供的文件或从其抽出的数据所造成的任何损失，本公司不予负责。请用户自行承担 responsibility。
- (5) 请遵守本公司提供的文件中的使用条件。
- (6) 本公司有权利不经通知修改或删除文件。
- (7) 使用本公司提供的文件时, 请务必熟读产品手册及手册中介绍的相关手册。同时请务必充分注意安全事宜，正确使用。

## 目次

目次	3
修订记录	5
1. 概要	6
2. 系统构成	6
3. 关于 GOT	7
3.1 自动选择的系统应用程序	7
3.2 画面设计软件的连接机器设置	7
3.3 画面设计软件的以太网设置	7
4. 关于 AnyWireASLINK 主模块	8
4.1 模块的起始输入输出号	8
5. 画面规格	8
5.1 显示语言	8
5.2 画面切换	8
5.3 画面说明	13
5.3.1 AnyWireASLINK 菜单 (B-30001)	13
5.3.2 AnyWireASLINK 系统图 (B-30002)	14
5.3.3 输入信号监视 (B-30003)	15
5.3.4 输出信号监视 (B-30004)	16
5.3.5 传感器一览 (B-30005)	17
5.3.6 详细信息 (电缆连接类型 (I/O 非绝缘)) (B-30006)	18
5.3.7 详细信息 (电缆连接类型 (I/O 绝缘)) (B-30007)	19
5.3.8 详细信息 (I/O 防水连接器连接类型) (B-30008)	20
5.3.9 详细信息 (传输 • I/O 防水连接器连接类型) (B-30009)	21
5.3.10 详细信息 (光电放大器模块 (CAM, CAS)) (B-30010)	22
5.3.11 详细信息 (接近放大器模块) (B-30011)	24
5.3.12 详细信息 (光纤放大器模块 (CAM, CAS)) (B-30012)	26
5.3.13 详细信息 (气缸开关) (B-30013)	28
5.3.14 详细信息 (光电传感器) (B-30014)	30
5.3.15 详细信息 (光电传感器 (透过型受光)) (B-30015)	32
5.3.16 详细信息 (光电传感器 (透过型投光)) (B-30016)	34
5.3.17 详细信息 (接近开关) (B-30017)	36
5.3.18 详细信息 (压力传感器) (B-30018)	38
5.3.19 详细信息 (ASLINK 终端 (驱动部)) (B-30019)	40
5.3.20 iQSS 菜单 (B-30100)	41
5.3.21 iQSS 备份 (B-30101)	42
5.3.22 iQSS 备份设置 (B-30102)	43
5.3.23 iQSS 备份进展 (B-30103)	44
5.3.24 iQSS 恢复 (B-30104)	45
5.3.25 iQSS 恢复设置 (B-30105)	47
5.3.26 iQSS 恢复进展 (B-30106)	48
5.3.27 报警复位 (W-30001)	49
5.3.28 语言设置 (W-30002)	50

5.3.29	时钟设置(W-30003)	51
5.3.30	参数访问错误(W-30004)	52
5.3.31	数据删除确认对话框(W-30100)	53
5.3.32	通知对话框(W-30101)	54
5.3.33	执行确认对话框(W-30102)	55
5.4	从属模块详细信息画面对照表	56
5.5	使用软元件一览表	57
5.6	注释一览表	63
5.7	软元件数据传送一览表	69
5.8	配方一览表	74
5.9	脚本一览表	76
6.	模板	152
7.	其他	153
7.1	关于用户定义名称的登录	153
7.2	关于系统构成的更改	153
7.3	关于 CPU 的更改	154
7.4	起始输入输出号的更改	155
7.5	iQSS 备份文件夹构成	160

## 修订记录

### 样本画面说明书

修订日期	管理号*	修订内容
2013/10	BCN-P5999-0139	初版

\* 管理号记载在右下方。

### 工程数据

修订日期	工程数据	GT Designer3*	修订内容
2013/10	AnyWireASLINK_V_Ver1_C.GTX	1.100E	初版

\* 制作工程数据时使用的画面设计软件的版本。打开文件时请使用相同版本或更高版本的画面设计软件。

## 1. 概要

本资料是将 GOT2000 和 MELSEC-Q 系列或者 MELSEC-L 系列通过以太网连接时, 对 AnyWireASLINK 主模块 (LJ51AW12AL) 上所连接的从属模块的状态、参数进行监视、更改以及备份/恢复(仅支持 MELSEC-L 系列)的样本画面的说明书。

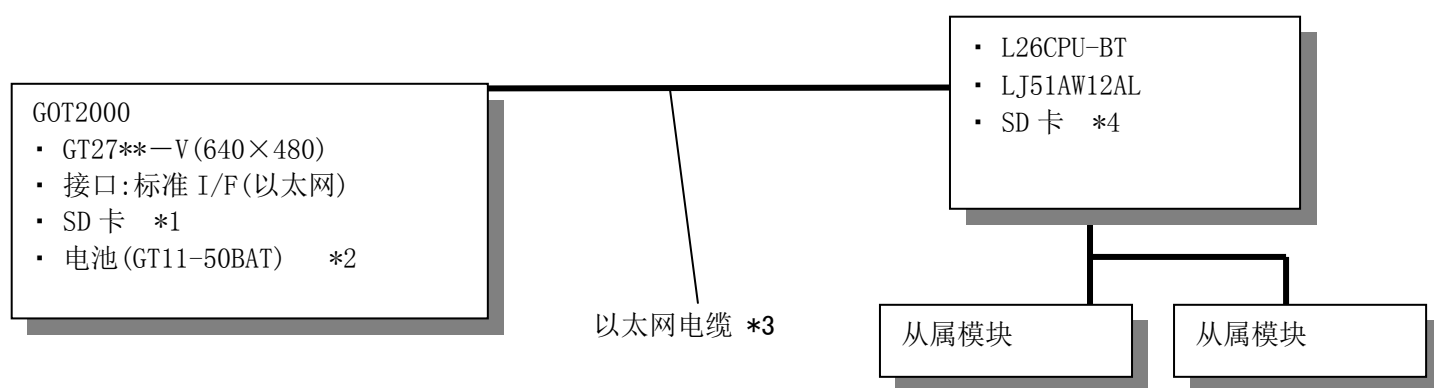
此样本画面是为连接 MELSEC-L 系列而制作的。连接 MELSEC-Q 系列时, 需要修改部分数据。关于具体修改部分, 请参照「7.3 关于 CPU 的更改」。而且 MELSEC-Q 系列不支持从属模块的参数的备份/恢复。

### <注意事项>

通过 iQSS 备份/恢复(可编程控制器⇔传感器)功能对从属模块的参数进行备份/恢复。iQSS 备份/恢复(可编程控制器⇔传感器)功能, 是由 GOT 执行可编程控制器侧的 iQ Sensor Solution 兼容的备份/恢复功能。和 GOT 固有的备份/恢复功能的规格并不一样。

iQ Sensor Solution 兼容的备份/恢复功能, 是将 iQ Sensor Solution 兼容传感器的参数备份到安装在可编程控制器 CPU 上的 SD 卡、或者从 SD 卡恢复的功能。

## 2. 系统构成



\*1: SD 卡, 用于配方功能。

\*2: 电池, 用于时钟数据的停电保持。(GOT中标配电池。)

\*3: 关于电缆的详细内容, 请参照「GOT2000 系列连接手册(三菱电机机器连接篇)」

\*4: SD 卡, 用于 iQSS 备份/恢复(可编程控制器⇔传感器)功能。

### 3. 关于 GOT

#### 3.1 自动选择的系统应用程序

种类	系统应用程序名称	
基本功能	基本系统应用程序	
	标准字体	日语
通讯驱动程序	以太网连接	以太网 (MELSEC), Q17nNC, CRnD-700, 网关
扩展功能	标准字体	中文 (简体)
	轮廓字体	黑体
		英数假名
		日语汉字
	中文 (简体) 汉字	
	软元件数据传送	

#### 3.2 画面设计软件的连接机器设置

详细设置

项 目	设置值	备 注
GOT 网络号	1	
GOT 站号	2	
GOT 以太网设置	参照下表	
GOT 机器通讯用端口号	5001	
重试次数 (次)	3	
启动时间 (秒)	3	
通讯超时时间 (秒)	3	
发送延迟时间 (ms)	0	

GOT 以太网设置

项 目	设置值	备 注
将 GOT 以太网设置反映到 GOT 本体	勾选	
GOT IP 地址	192.168.3.18	
子网掩码	255.255.255.0	
默认网关	0.0.0.0	
周边 S/W 通讯用端口号	5015	
透明用端口号	5014	

#### 3.3 画面设计软件的以太网设置

	本站	网络号	站号	机器	IP 地址	端口号	通讯方式
1	*	1	1	LCPU	192.168.3.39	5006	UDP

4. 关于 AnyWireASLINK 主模块

4.1 模块的起始输入输出号

模块的起始输入输出号设置为 30H。要更改起始输入输出号，请参照「7.4 起始输入输出号的更改」。

5. 画面规格

5.1 显示语言

画面可以显示日语/英语/中文(简体)3 种语言。如下所示各种语言的字符串，登录在注释组号 250~255 的列号 1~3 中。将列号写入语言切换软元件中即可显示与列号相应的语言。

列号	语言
1	中文(简体)
2	日语
3	英语

5.2 画面切换

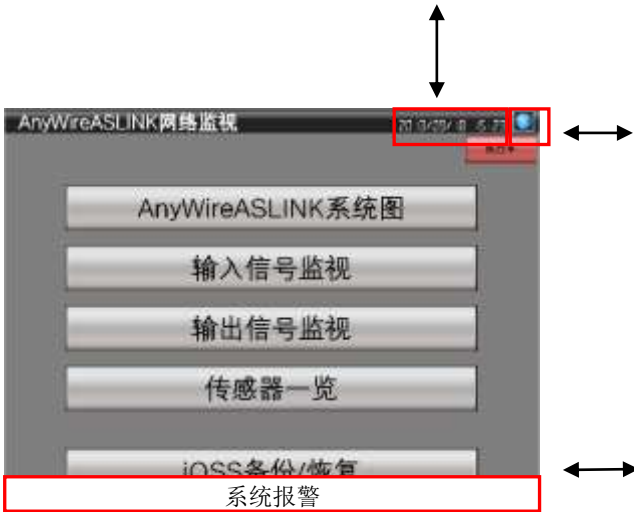
5.2.1 画面切换(公共)



窗口画面 W-30003: 时钟设置



窗口画面 W-30002: 语言设置



基本画面 B-30001: 菜单及全部基本画面



窗口画面 W-30001: 报警复位



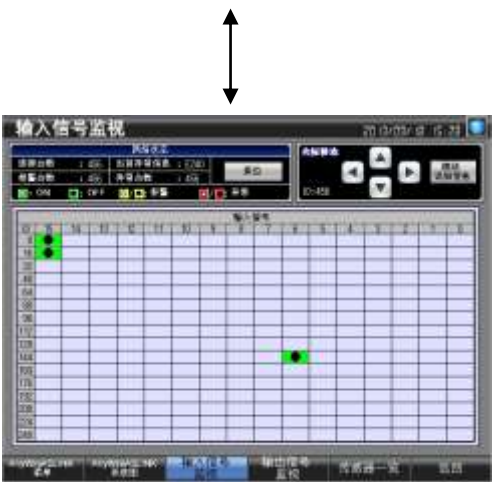
5.2.2 画面切换(个别)



基本画面 B-30001: AnyWireASLINK 菜单



基本画面 B-30002: AnyWireASLINK 系统图



基本画面 B-30003: 输入信号监视

至下一页

接上一页



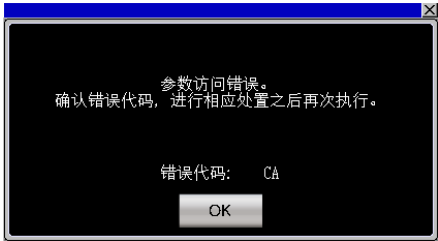
基本画面 B-30004: 输出信号监视



基本画面 B-30005: 传感器一览



基本画面 B-30006~30019: 详细信息  
(画面因从属模块的类型而异)



窗口画面 W-30004: 参数访问错误

至下一页

接上一页



基本画面 B-30100: iQSS 菜单



基本画面 B-30102: iQSS 备份设置



基本画面 B-30101: iQSS 备份



窗口画面 W-30101: 通知对话框  
(B-30101~30106 公共)



基本画面 B-30103: iQSS 备份进展



窗口画面 W-30102: 执行确认对话框  
(B-30101、B-30103、B-30104、B-30106 公共)

至下一页

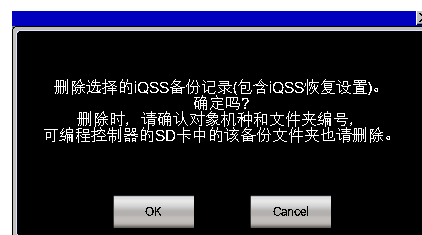
接上一页

IQSS恢复(可编程控制器→传感器)

No.	时间	设置名称	对象名称	对象ID	IO	数据源/目标
1	2013/08/19 10:00	ASLINK4	A	23	03	01-01-00
2	2013/08/19 10:00	ASLINK3	A	22	03	01-01-00
3	2013/08/19 10:00	ASLINK2	A	21	03	01-01-00
4	2013/08/19 10:07	ASLINK1	A	20	03	01-01-00
5	2000/00/00 00:00			00	00	00-00-00
6	2000/00/00 12:00			00	00	00-00-00
7	2000/00/00 06:00			00	00	00-00-00
8	2000/00/00 00:00			00	00	00-00-00
9	2000/00/00 00:00			00	00	00-00-00
10	2000/00/00 00:00			00	00	00-00-00

恢复删除 删除全部记录 设置确认 执行

基本画面 B-30104: iQSS 恢复



窗口画面 W-30100: 数据删除确认对话框 (B-30101、B-30104 公共)

IQSS恢复进展

恢复对象	
设置名称	ASLINK3
执行单位	ID指定
对象名称	14
IO	输入/输出

恢复结果	
开始时间	2013/08/09 10:12
结果	<div style="width: 100%; height: 10px; background-color: green;"></div>
对象计数器数	1
恢复成功	1
恢复失败	0

恢复完成 恢复中止

基本画面 B-30106: iQSS 恢复进展

IQSS恢复设置

设置名称	ASLINK3		
对象名称	AnyWireASLINK		
执行单位	符号	全部站	ID指定 : 14
	地址类型	输入/输出	
文件夹编号	23		
IO	3		
继续时动作设置	继续执行 停止		

返回

基本画面 B-30105: iQSS 恢复设置

5.3 画面说明

5.3.1 AnyWireASLINK 菜单 (B-30001)



概要

AnyWireASLINK 网络监视功能的菜单画面。

详细

1. 切换至 AnyWireASLINK 系统图画面。
2. 切换至输入信号监视画面。
3. 切换至输出信号监视画面。
4. 切换至传感器一览画面。
5. 切换至 iQSS 备份/恢复(可编程控制器⇔传感器)功能的菜单画面(iQSS 菜单画面)。
6. 读取被主模块识别的全部从属模块的参数。请触摸 1 秒以上。
7. 显示当前日期和时间。触摸即显示时钟设置窗口。
8. 显示语言设置窗口。

备注

- 通过软元件数据传送功能读取从属模块的参数。关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 读取参数时无法切换画面。
- GOT 启动时通过工程脚本设置 iQSS 备份的对象机器设置、I/O 编号。关于脚本，请参照「5.9 脚本一览表」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5.3.2 AnyWireASLINK 系统图(B-30002)



概要

AnyWireASLINK 系统图画面。显示从属模块的连接状态，触摸各从属模块可以切换至详细画面。

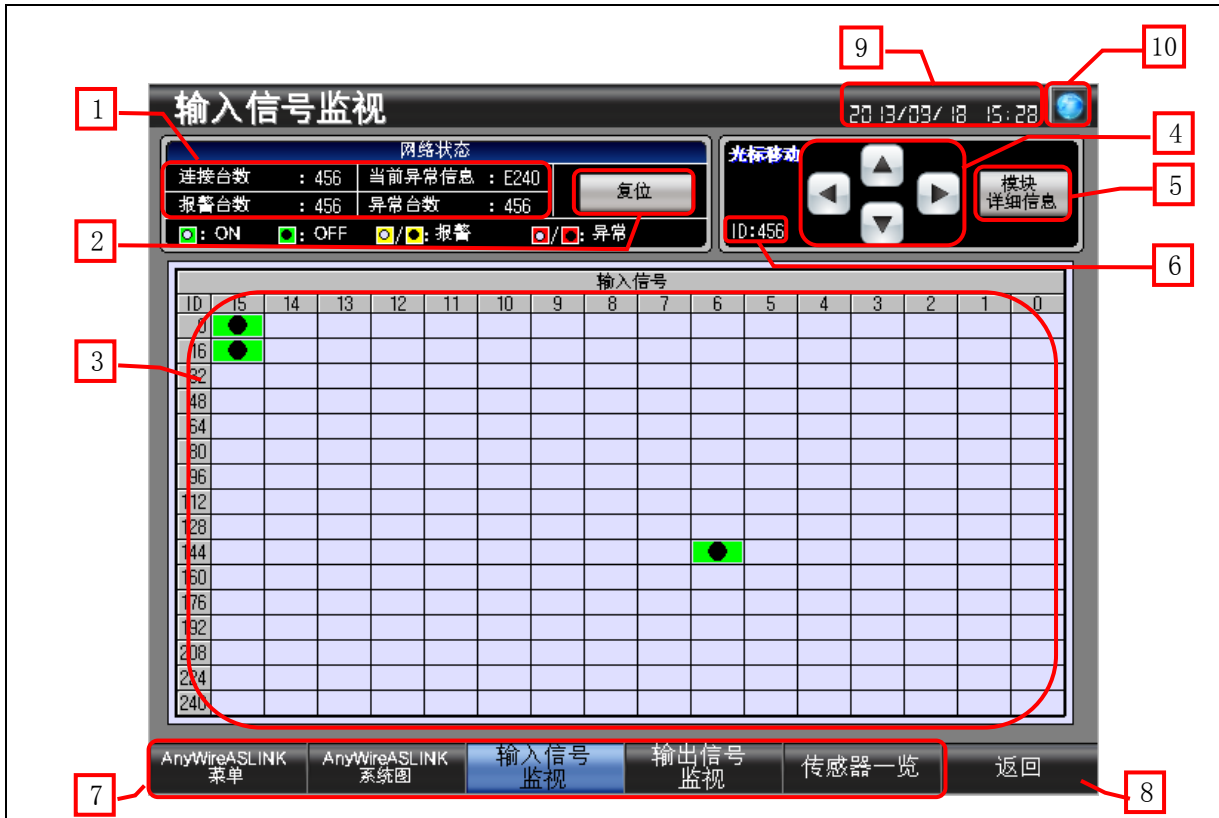
详细

1. 显示连接台数/报警台数/异常台数。
2. 显示从属模块的连接状态。显示模块的 ID/类型/状态(正常/发生报警/发生异常)。触摸显示着模块种类的开关，可以切换至相应从属模块的详细画面。
3. 显示正在发生的报警或者异常的错误代码以及代码所对应的错误内容。有多个报警、异常发生时，显示最新的错误代码。
4. 复位报警及异常。执行复位后，报警台数/异常台数/当前异常信息都清零。
5. 上下滚动显示从属模块。每次切换 32 台。
6. 切换至各画面。蓝色开关为当前显示的画面，所以显示中的画面不被切换。
7. 切换至上次显示画面。
8. 显示当前日期和时间。触摸即显示时钟设置窗口。
9. 显示语言设置窗口。

备注

- 通过脚本实现从属模块的滚动显示。关于脚本，请参照「5.9 脚本一览表」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

### 5.3.3 输入信号监视 (B-30003)



#### 概要

输入信号监视画面。显示所连接着的输入从属模块的信号的状态 (ON/OFF、报警发生中、异常发生中)。不同种类的从属模块所显示的信号点数也会有所不同。

#### 详细

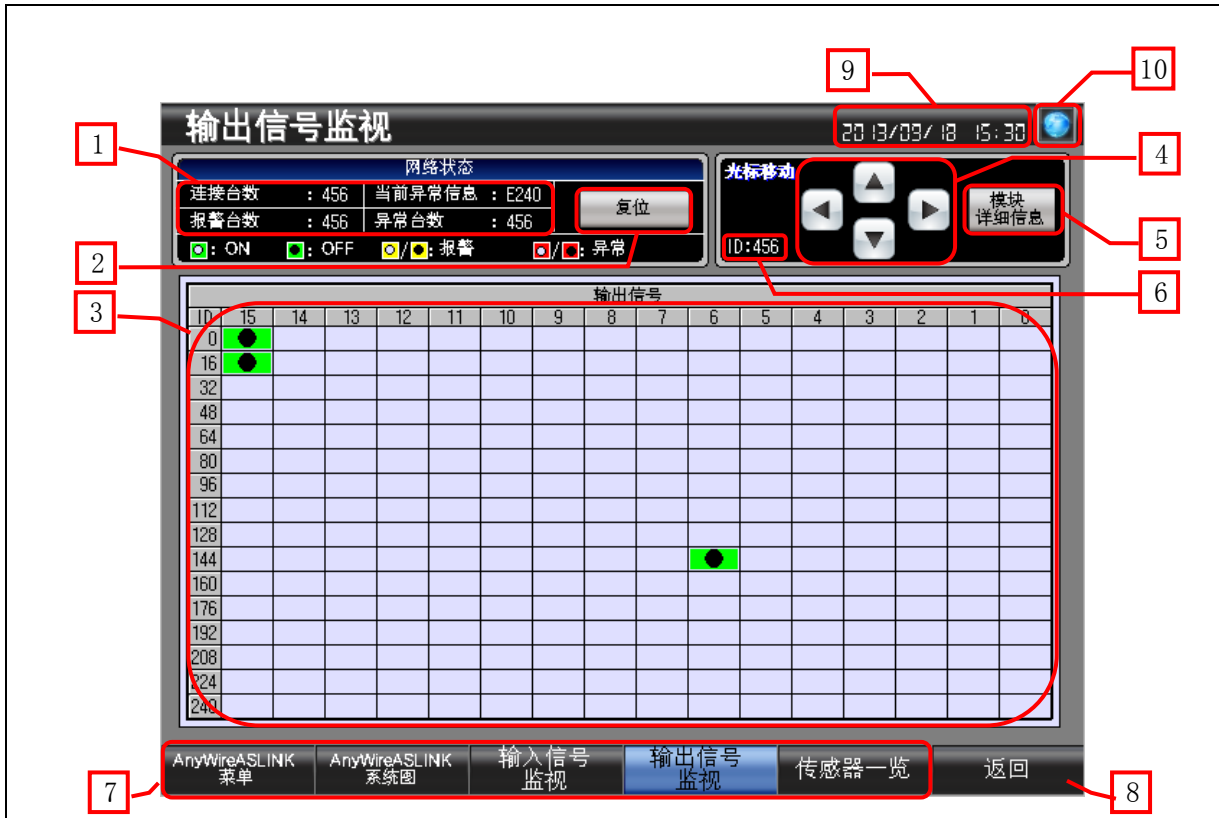
1. 显示连接台数/当前异常信息/报警台数/异常台数。
2. 复位报警及异常。如执行复位，则报警台数/异常台数/当前异常信息清零。
3. 显示输入从属模块的状态。触摸单元格会使光标移动到触摸处。
4. 上下左右移动光标。
5. 切换至光标所在位置的从属模块的详细画面。无法识别光标所在位置的 ID 时，则不会切换至详细画面。
6. 显示光标所在位置的 ID。
7. 切换至各画面。蓝色开关为当前显示的画面，所以显示中的画面不被切换。
8. 切换至上次显示画面。
9. 显示当前日期和时间。触摸即显示时钟设置窗口。
10. 显示语言设置窗口。

#### 备注

- 通过脚本实现光标的移动。关于脚本，请参照「5.9 脚本一览表」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。



### 5.3.4 输出信号监视 (B-30004)



#### 概要

输出信号监视画面。显示所连接着的输出从属模块的信号的状态(ON/OFF、报警发生中、异常发生中)。不同种类的从属模块所显示的信号点数也会有所不同。

#### 详细

1. 显示连接台数/当前异常信息/报警台数/异常台数。
2. 复位报警及异常。如执行复位，则报警台数/异常台数/当前异常信息清零。
3. 显示输出从属模块的状态。触摸单元格会使光标移动到触摸处。
4. 上下左右移动光标。
5. 切换至光标所在位置的从属模块的详细画面。光标在无法识别 ID 的位置时，则不会切换至详细画面。
6. 显示光标所在位置的 ID。
7. 切换至各画面。蓝色开关为当前显示的画面，所以显示中的画面不被切换。
8. 切换至上次显示画面。
9. 显示当前日期和时间。触摸即显示时钟设置窗口。
10. 显示语言设置窗口。

#### 备注

- 通过脚本实现光标的移动。关于脚本，请参照「5.9 脚本一览表」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。



### 5.3.5 传感器一览(B-30005)



#### 概要

显示从属模块的信息。触摸显示着从属模块信息的行，即可切换至相应从属模块的详细画面。

#### 详细

1. 显示正在发生的报警或者异常的错误代码以及代码所对应的错误内容。有多个报警、异常发生时，显示最新的错误代码。
2. 显示从属模块的信息。显示所识别的从属模块的 ID/类型/状态详细信息/用户定义名称。触摸显示着信息的行，即可切换至从属模块的详细画面。
3. 复位报警及异常。如执行复位，则当前异常信息清零。
4. 上下滚动从属模块的信息。每次切换 15 台。
5. 切换至各画面。蓝色开关为当前显示的画面，所以显示中的画面不被切换。
6. 切换至上次显示画面。
7. 显示当前日期和时间。触摸即显示时钟设置窗口。
8. 显示语言设置窗口。

#### 备注

- 通过脚本实现从属模块信息显示的滚动。关于脚本，请参照「5.9 脚本一览表」。
- 用户定义名称，可以显示用户需要显示的任何信息(从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5.3.6 详细信息(电缆连接类型(I/O 非绝缘))(B-30006)



概要

显示、设置电缆连接类型(I/O 非绝缘)的从属模块的详细信息。

详细

1. 显示从属模块的 ID/型号/种类/ I/O 的种类/点数。。
2. 显示从属模块的图像。
3. 显示 I/O 的状态。
4. 显示、更改从属模块的参数。
5. 显示用户定义名称。
6. 显示正在发生的报警或者异常的错误内容及处理方法。有多个报警或者异常发生时，显示最新的错误内容。
7. 滚动显示处理方法。
8. 将 4 中更改过的参数写入到从属模块。执行写入操作后，会自动执行参数的读取处理。
9. 读取从属模块的参数。
10. 切换至各画面。
11. 切换至上次显示画面。
12. 显示当前日期和时间。触摸即显示时钟设置窗口。
13. 显示语言设置窗口。

备注

- 在系统运转中更改参数，有可能导致从属模块动作的变化。请先确认安全后再执行。
- 用户定义名称，可以显示用户需要显示的任何信息(从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

### 5. 3. 7 详细信息 (电缆连接类型 (I/O 绝缘)) (B-30007)



概要
显示电缆连接类型 (I/O 绝缘) 的从属模块的详细信息。
详细
<ol style="list-style-type: none"><li>1. 显示从属模块的 ID/型号/种类/ I/O 的种类/点数。。</li><li>2. 显示从属模块的图像。</li><li>3. 显示 I/O 的状态。</li><li>4. 显示用户定义名称。</li><li>5. 显示正在发生的报警或者异常的错误的内容及处理方法。有多个报警或者异常发生时，显示最新的错误内容。</li><li>6. 滚动显示处理方法。</li><li>7. 因为不存在写入参数，即使触摸也无动作</li><li>8. 因为不存在读取参数，即使触摸也无动作。</li><li>9. 切换至各画面。</li><li>10. 切换至上次显示画面。</li><li>11. 显示当前日期和时间。触摸即显示时钟设置窗口。</li><li>12. 显示语言设置窗口。</li></ol>
备注
<ul style="list-style-type: none"><li>• 用户定义名称，可以显示用户需要显示的任何信息 (从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。</li><li>• 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一一览表」。</li><li>• 画面切换时，将关闭显示中的窗口。</li><li>• 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。</li></ul>

### 5.3.8 详细信息(I/O 防水连接器连接类型) (B-30008)



#### 概要

显示、设置 I/O 防水连接器连接类型的从属模块的详细信息。

#### 详细

1. 显示从属模块的 ID/型号/种类/ I/O 的种类/点数。。
2. 显示从属模块的图像。
3. 显示 I/O 的状态。
4. 显示、更改从属模块的参数。
5. 显示用户定义名称。
6. 显示正在发生的报警或者异常的错误内容及处理方法。有多个报警或者异常发生时，显示最新的错误内容。
7. 滚动显示处理方法。
8. 将 4 中更改过的参数写入到从属模块。执行写入操作后，会自动执行参数的读取处理。
9. 读取从属模块的参数。
10. 切换至各画面。
11. 切换至上次显示画面。
12. 显示当前日期和时间。触摸即显示时钟设置窗口。
13. 显示语言设置窗口。

#### 备注

- 在系统运转中更改参数，有可能导致从属模块动作的变化。请先确认安全后再执行。
- 用户定义名称，可以显示用户需要显示的任何信息(从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5.3.9 详细信息(传输・I/O 防水连接器连接类型)(B-30009)



概要
显示传输・I/O 防水连接器连接类型的从属模块的详细信息。
详细
<div>1. 显示从属模块的 ID/型号/种类/ I/O 的种类/点数。。</div> <div>2. 显示从属模块的图像。</div> <div>3. 显示 I/O 的状态。</div> <div>4. 显示、更改从属模块的参数。</div> <div>5. 显示用户定义名称。</div> <div>6. 显示正在发生的报警或者异常的错误内容及处理方法。有多个报警或者异常发生时，显示最新的错误内容。</div> <div>7. 滚动显示处理方法。</div> <div>8. 将 4 中更改过的参数写入到从属模块。执行写入操作后，会自动执行参数的读取处理。</div> <div>9. 读取从属模块的参数。</div> <div>10. 切换至各画面。</div> <div>11. 切换至上次显示画面。</div> <div>12. 显示当前日期和时间。触摸即显示时钟设置窗口。</div> <div>13. 显示语言设置窗口。</div>
备注
<div>・在系统运转中更改参数，有可能导致从属模块动作的变化。请先确认安全后再执行。</div> <div>・用户定义名称，可以显示用户需要显示的任何信息(从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。</div> <div>・每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。</div> <div>・画面切换时，将关闭显示中的窗口。</div> <div>・系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。</div>

5. 3. 10 详细信息(光电放大器模块(CAM, CAS))(B-30010)



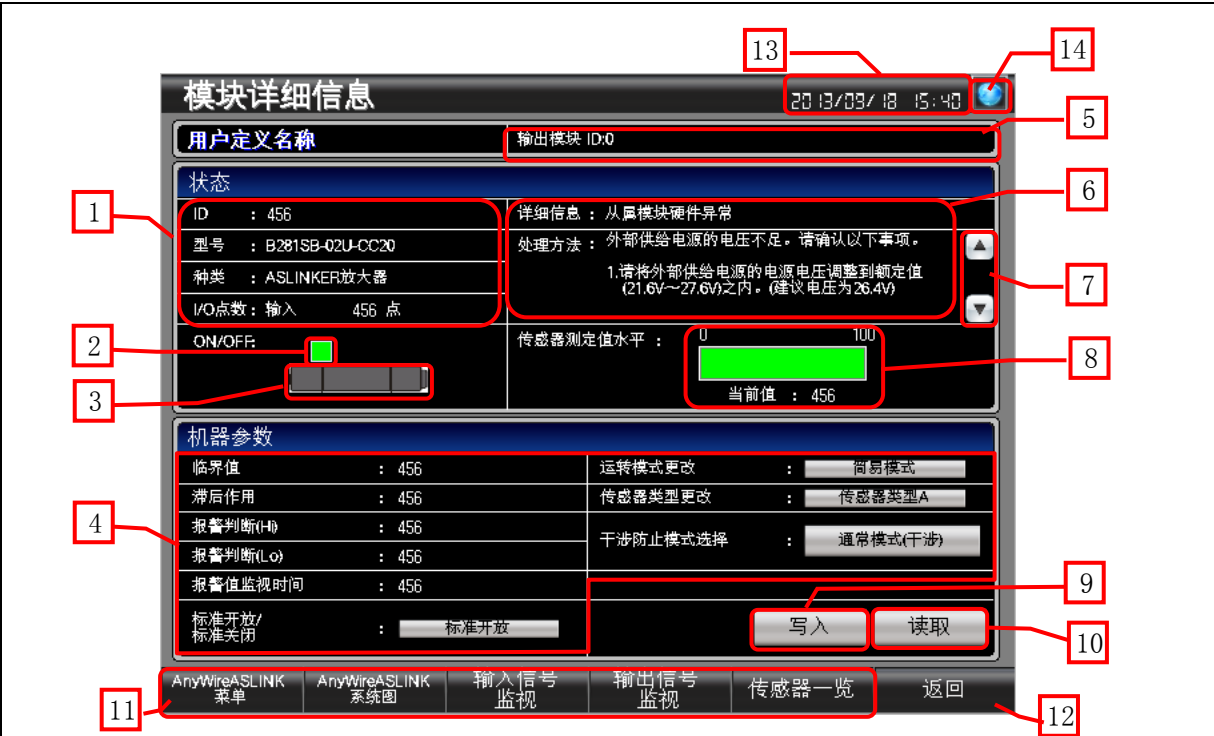
**概要**  
显示、设置光电放大器模块(CAM, CAS)的从属模块的详细信息。

- 详细**
1. 显示从属模块的 ID/型号/种类/ I/O 的种类/点数。。
  2. 显示 I/O 的状态。
  3. 显示从属模块的图像。
  4. 显示、更改从属模块的参数。
  5. 显示用户定义名称。
  6. 显示正在发生的报警或者异常的的错误内容及处理方法。有多个报警或者异常发生时，显示最新的错误内容。
  7. 滚动显示处理方法。
  8. 通过数值显示和液位，显示传感器测定值水平的当前值。
  9. 将 4 中更改过的参数写入到从属模块。执行写入操作后，会自动执行参数的读取处理。
  10. 读取从属模块的参数。
  11. 切换至各画面。
  12. 切换至上次显示画面。
  13. 显示当前日期和时间。触摸即显示时钟设置窗口。
  14. 显示语言设置窗口。

#### 备注

- 在系统运转中更改参数，有可能导致从属模块动作的变化。请先确认安全后再执行。
- 在「传感器测定值水平」的液位以及「临界值」、「报警判断(Hi)」、「报警判断(Lo)」的数值显示中设置了对象脚本。关于脚本，请参照「5.9 脚本一览表」。
- 用户定义名称，可以显示用户需要显示的任何信息(从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5. 3. 11 详细信息(接近放大器模块) (B-30011)



**概要**  
显示、设置接近放大器模块的从属模块的详细信息。

- 详细**
1. 显示从属模块的 ID/型号/种类/ I/O 的种类/点数。。
  2. 显示 I/O 的状态。
  3. 显示从属模块的图像。
  4. 显示、更改从属模块的参数。
  5. 显示用户定义名称。
  6. 显示正在发生的报警或者异常的错误的内容及处理方法。有多个报警或者异常发生时，显示最新的错误内容。
  7. 滚动显示处理方法。
  8. 通过数值显示和液位，显示传感器测定值水平的当前值。
  9. 将 4 中更改过的参数写入到从属模块。执行写入操作后，会自动执行参数的读取处理。
  10. 读取从属模块的参数。
  11. 切换至各画面。
  12. 切换至上次显示画面。
  13. 显示当前日期和时间。触摸即显示时钟设置窗口。
  14. 显示语言设置窗口。



#### 备注

- 在系统运转中更改参数，有可能导致从属模块动作的变化。请先确认安全后再执行。
- 在「传感器测定值水平」的液位以及「临界值」、「报警判断(Hi)」、「报警判断(Lo)」的数值显示中设置了对象脚本。关于脚本，请参照「5.9 脚本一览表」。
- 用户定义名称，可以显示用户需要显示的任何信息(从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5. 3. 12 详细信息(光纤放大器模块(CAM, CAS))(B-30012)



概要

显示、设置光纤放大器模块(CAM, CAS)的从属模块的详细信息。

详细

1. 显示从属模块的 ID/型号/种类/ I/O 的种类/点数。。
2. 显示 I/O 的状态。
3. 显示从属模块的图像。
4. 显示、更改从属模块的参数。
5. 显示用户定义名称。
6. 显示正在发生的报警或者异常的错误的内容及处理方法。有多个报警或者异常发生时，显示最新的错误内容。
7. 滚动显示处理方法。
8. 通过数值显示和液位，显示传感器测定值水平的当前值。
9. 将 4 中更改过的参数写入到从属模块。执行写入操作后，会自动执行参数的读取处理。
10. 读取从属模块的参数。
11. 切换至各画面。
12. 切换至上次显示画面。
13. 显示当前日期和时间。触摸即显示时钟设置窗口。
14. 显示语言设置窗口。

#### 备注

- 在系统运转中更改参数，有可能导致从属模块动作的变化。请先确认安全后再执行。
- 在「传感器测定值水平」的液位以及「临界值」、「报警判断(Hi)」、「报警判断(Lo)」的数值显示中设置了对象脚本。关于脚本，请参照「5.9 脚本一览表」。
- 用户定义名称，可以显示用户需要显示的任何信息(从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5. 3. 13 详细信息(气缸开关) (B-30013)



概要

显示、设置气缸开关的从属模块的详细信息。

详细

1. 显示从属模块的 ID/型号/种类/ I/O 的种类/点数。。
2. 显示从属模块的图像。
3. 显示 I/O 的状态。
4. 显示、更改从属模块的参数。
5. 显示用户定义名称。
6. 显示正在发生的报警或者异常的错误内容及处理方法。有多个报警或者异常发生时，显示最新的错误内容。
7. 滚动显示处理方法。
8. 通过数值显示和液位，显示传感器测定值水平的当前值。
9. 将 4 中更改过的参数写入到从属模块。执行写入操作后，会自动执行参数的读取处理。
10. 读取从属模块的参数。
11. 切换至各画面。
12. 切换至上次显示画面。
13. 显示当前日期和时间。触摸即显示时钟设置窗口。
14. 显示语言设置窗口。

#### 备注

- 在系统运转中更改参数，有可能导致从属模块动作的变化。请先确认安全后再执行。
- 在「传感器测定值水平」的液位以及「OFF→ON 的临界值」、「ON→OFF 的临界值」的数值显示中设置了对象脚本。关于脚本，请参照「5.9 脚本一览表」。
- 用户定义名称，可以显示用户需要显示的任何信息(从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5. 3. 14 详细信息(光电传感器) (B-30014)



**概要**  
显示、设置光电传感器的从属模块的详细信息。

- 详细**
1. 显示从属模块的 ID/型号/种类/ I/O 的种类/点数。。
  2. 显示从属模块的图像。
  3. 显示 I/O 的状态。
  4. 显示、更改从属模块的参数。
  5. 显示用户定义名称。
  6. 显示正在发生的报警或者异常的错误的内容及处理方法。有多个报警或者异常发生时，显示最新的错误内容。
  7. 滚动显示处理方法。
  8. 通过数值显示和液位，显示传感器测定值水平的当前值。
  9. 将 4 中更改过的参数写入到从属模块。执行写入操作后，会自动执行参数的读取处理。
  10. 读取从属模块的参数。
  11. 切换至各画面。
  12. 切换至上次显示画面。
  13. 显示当前日期和时间。触摸即显示时钟设置窗口。
  14. 显示语言设置窗口。

#### 备注

- 在系统运转中更改参数，有可能导致从属模块动作的变化。请先确认安全后再执行。
- 在「传感器测定值水平」的液位以及「临界值」、「报警判断(Hi)」、「报警判断(Lo)」的数值显示中设置了对象脚本。关于脚本，请参照「5.9 脚本一览表」。
- 用户定义名称，可以显示用户需要显示的任何信息(从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5. 3. 15 详细信息(光电传感器(透过型受光)) (B-30015)



概要

显示、设置(光电传感器(透过型受光))的从属模块的详细信息。

详细

1. 显示从属模块的 ID/型号/种类/ I/O 的种类/点数。。
2. 显示从属模块的图像。
3. 显示 I/O 的状态。
4. 显示、更改从属模块的参数。
5. 显示用户定义名称。
6. 显示正在发生的报警或者异常的错误内容及处理方法。有多个报警或者异常发生时，显示最新的错误内容。
7. 滚动显示处理方法。
8. 通过数值显示和液位，显示传感器测定值水平的当前值。
9. 将 4 中更改过的参数写入到从属模块。执行写入操作后，会自动执行参数的读取处理。
10. 读取从属模块的参数。
11. 切换至各画面。
12. 切换至上次显示画面。
13. 显示当前日期和时间。触摸即显示时钟设置窗口。
14. 显示语言设置窗口。



#### 备注

- 在系统运转中更改参数，有可能导致从属模块动作的变化。请先确认安全后再执行。
- 在「传感器测定值水平」的液位以及「临界值」、「报警判断(Hi)」、「报警判断(Lo)」的数值显示中设置了对象脚本。关于脚本，请参照「5.9 脚本一览表」。
- 用户定义名称，可以显示用户需要显示的任何信息(从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5. 3. 16 详细信息(光电传感器(透过型投光)) (B-30016)



概要

显示、设置光电传感器(透过型投光)的从属模块的详细信息。

详细

1. 显示从属模块的 ID/型号/种类/ I/O 的种类/点数。。
2. 显示从属模块的图像。
3. 显示 I/O 的状态。
4. 显示、更改从属模块的参数。
5. 显示用户定义名称。
6. 显示正在发生的报警或者异常的错误内容及处理方法。有多个报警或者异常发生时，显示最新的错误内容。
7. 滚动显示处理方法。
8. 将 4 中更改过的参数写入到从属模块。执行写入操作后，会自动执行参数的读取处理。
9. 读取从属模块的参数。
10. 切换至各画面。
11. 切换至上次显示画面。
12. 显示当前日期和时间。触摸即显示时钟设置窗口。
13. 显示语言设置窗口。

#### 备注

- 在系统运转中更改参数，有可能导致从属模块动作的变化。请先确认安全后再执行。
- 用户定义名称，可以显示用户需要显示的任何信息(从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5. 3. 17 详细信息(接近开关) (B-30017)



**概要**  
显示、设置接近开关的从属模块的详细信息。

- 详细**
1. 显示从属模块的 ID/型号/种类/ I/O 的种类/点数。。
  2. 显示从属模块的图像。
  3. 显示 I/O 的状态。
  4. 显示、更改从属模块的参数。
  5. 显示用户定义名称。
  6. 显示正在发生的报警或者异常的错误的内容及处理方法。有多个报警或者异常发生时，显示最新的错误内容。
  7. 滚动显示处理方法。
  8. 通过数值显示和液位，显示传感器测定值水平的当前值。
  9. 将 4 中更改过的参数写入到从属模块。执行写入操作后，会自动执行参数的读取处理。
  10. 读取从属模块的参数。
  11. 切换至各画面。
  12. 切换至上次显示画面。
  13. 显示当前日期和时间。触摸即显示时钟设置窗口。
  14. 显示语言设置窗口。

#### 备注

- 在系统运转中更改参数，有可能导致从属模块动作的变化。请先确认安全后再执行。
- 在「传感器测定值水平」的液位以及「临界值」、「报警判断(Hi)」、「报警判断(Lo)」的数值显示中设置了对象脚本。关于脚本，请参照「5.9 脚本一览表」。
- 用户定义名称，可以显示用户需要显示的任何信息(从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5. 3. 18 详细信息(压力传感器) (B-30018)



**概要**  
显示、设置压力传感器的从属模块的详细信息。

- 详细**
1. 显示从属模块的 ID/型号/种类/ I/O 的种类/点数。。
  2. 显示从属模块的图像。
  3. 显示 I/O 的状态。
  4. 显示、更改从属模块的参数。
  5. 显示用户定义名称。
  6. 显示正在发生的报警或者异常的错误的内容及处理方法。有多个报警或者异常发生时，显示最新的错误内容。
  7. 滚动显示处理方法。
  8. 通过数值显示和液位，显示传感器测定值水平的当前值。
  9. 将 4 中更改过的参数写入到从属模块。执行写入操作后，会自动执行参数的读取处理。
  10. 读取从属模块的参数。
  11. 切换至各画面。
  12. 切换至上次显示画面。
  13. 显示当前日期和时间。触摸即显示时钟设置窗口。
  14. 显示语言设置窗口。

#### 备注

- 在系统运转中更改参数，有可能导致从属模块动作的变化。请先确认安全后再执行。
- 在「传感器测定值水平」的液位以及「OFF→ON 的临界值」、「ON→OFF 的临界值」的数值显示中设置了对象脚本。关于脚本，请参照「5.9 脚本一览表」。
- 用户定义名称，可以显示用户需要显示的任何信息(从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5. 3. 19 详细信息 (ASLINK 终端 (驱动部)) (B-30019)



概要

显示 ASLINK 终端 (驱动部) 的从属模块的详细信息。

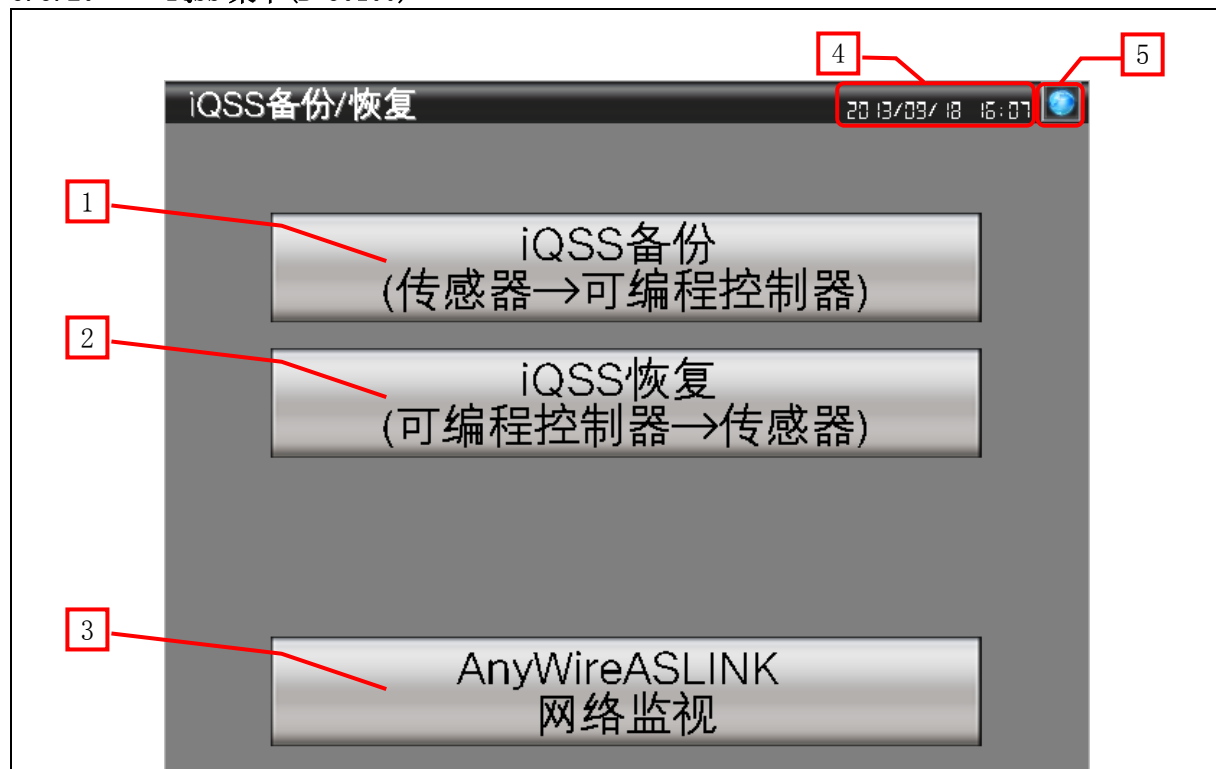
详细

1. 显示从属模块的 ID/型号/种类/ I/O 的种类/点数。。
2. 显示从属模块的图像。
3. 显示 I/O 的状态。
4. 显示用户定义名称。
5. 显示正在发生的报警或者异常的错误的错误内容及处理方法。有多个报警或者异常发生时，显示最新的错误内容。
6. 滚动显示处理方法。
7. 因为不存在写入参数，即使触摸也无动作。
8. 因为不存在读取参数，即使触摸也无动作。
9. 切换至各画面。
10. 切换至上次显示画面。
11. 显示当前日期和时间。触摸即显示时钟设置窗口。
12. 显示语言设置窗口。

备注

- 用户定义名称，可以显示用户需要显示的任何信息 (从属模块的安装地点等)。如需显示自定义内容，请将所需显示的信息登录到 GOT 的注释。详细内容请参照「7.1 关于用户定义名称的登录」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。





#### 概要

iQSS 备份/恢复(可编程控制器 $\leftrightarrow$ 传感器)功能的菜单画面。

#### 详细

1. 切换至 iQSS 备份画面。
2. 切换至 iQSS 恢复画面。
3. 切换至 AnyWireASLINK 网络监视功能的菜单画面 (AnyWireASLINK 菜单画面)。
4. 显示当前日期和时间。触摸即显示时钟设置窗口。
5. 显示语言设置窗口。

#### 备注

- 从配方文件读取 iQSS 备份设置、iQSS 备份记录。关于配方功能，请参照「5.8 配方一览表」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。



### 概要

依据选择状态的 iQSS 备份设置的参数执行 iQSS 备份。或者对 iQSS 备份设置进行编辑、删除。

### 详细

1. 以一览表显示 iQSS 备份设置，每次 10 件，最多 100 件。触摸 iQSS 备份设置即显示光标，成为选择状态。
2. 删除选择状态的 iQSS 备份设置。
3. 上下翻页显示一览表。
4. 切换至 iQSS 备份进展画面的同时，依据选择状态的 iQSS 备份设置的参数执行 iQSS 备份。如果触摸开关时选择状态的 iQSS 备份设置尚未登录，就会显示错误对话框。
5. 切换至 iQSS 备份设置画面。
6. 切换至各画面。蓝色开关为当前显示的画面，所以显示中的画面不被切换。
7. 未使用的基本画面切换开关。
8. 切换至上次显示画面。
9. 显示当前日期和时间。触摸即显示时钟设置窗口。
10. 显示语言设置窗口。

### 备注

- iQSS 备份记录最多 100 件。执行第 101 件的备份时，会有信息提示将要覆盖最旧数据。
- 删除操作，是删除保存在配方文件中的 iQSS 备份设置。关于配方功能，请参照「5.8 配方一览表」。
- 在「执行单位」的字指示灯中设置了对象脚本。关于脚本，请参照「5.9 脚本一览表」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5.3.22 iQSS 备份设置(B-30102)

The screenshot shows the 'iQSS 备份设置' (iQSS Backup Settings) screen. It features a title bar with a clock and a language icon. The main area contains several input fields and buttons. Red boxes and numbers 1 through 11 point to specific elements: 1 points to the 'No.' field (value 123); 2 points to the '设置名称' (Setting Name) field (value ASLINK5); 3 points to the '对象机种' (Object Model) field (value AnyWireASLINK); 4 points to the '执行单位' (Execution Unit) section, specifically the '站号' (Station Number) field (value 5) and the '模块类型' (Module Type) buttons ('全部ID', 'ID指定', '输出', '输入/混合'); 5 points to the '文件夹编号' (Folder Number) field (value 自动); 6 points to the 'I/O' field (value 3); 7 points to the '错误时动作设置' (Action on Error) buttons ('继续执行', '停止'); 8 points to the '登录' (Login) button; 9 points to the '返回' (Return) button; 10 points to the clock in the title bar; and 11 points to the language icon in the title bar.

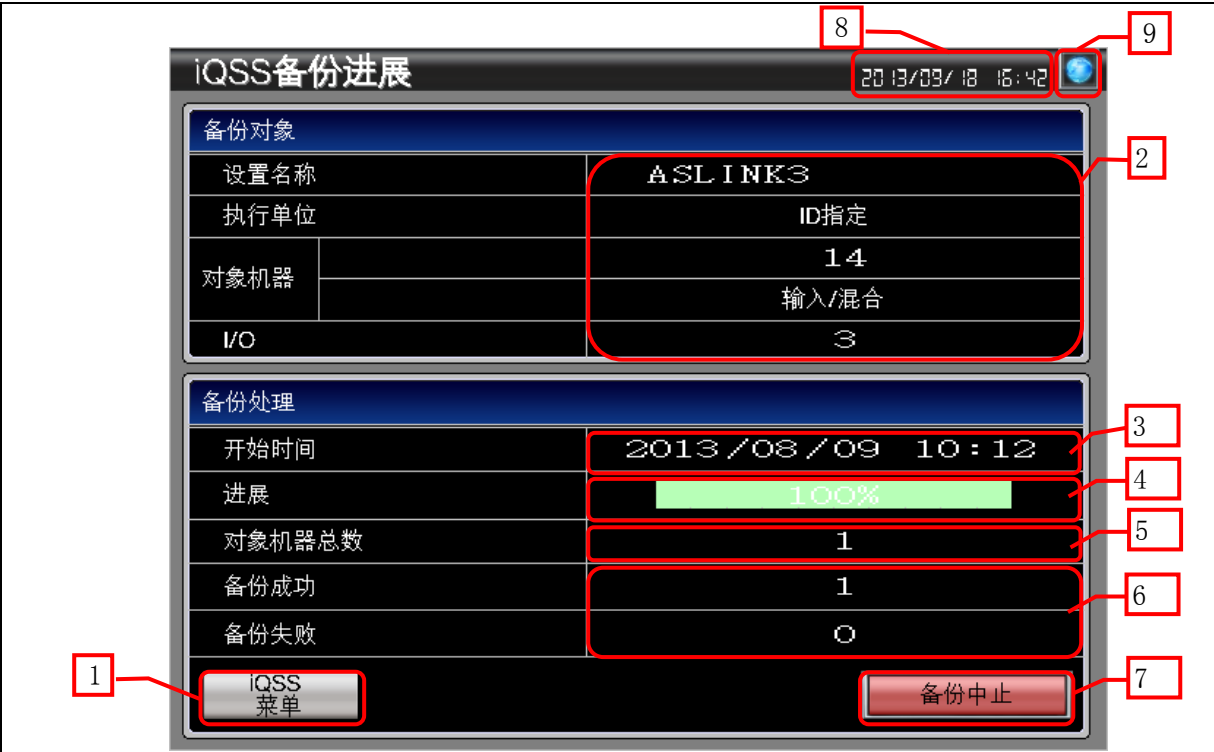
**概要**  
登录 iQSS 备份设置。

- 详细**
1. 显示登录编号。
  2. 输入 iQSS 备份设置的名称。
  3. 显示 iQSS 备份对象机器所连接的网络模块的种类。
  4. 指定 iQSS 备份对象。  
全部 ID...全部 iQSS 兼容传感器  
ID 指定...指定 ID 的 iQSS 兼容传感器  
输出、输入/混合...指定的 iQSS 兼容传感器的模块类型
  5. 用作 iQSS 备份数据的存储文件夹名的文件夹编号是设置成自动分配的。给文件夹编号自动分配 0~99 的值。
  6. 输入 iQSS 备份对象机器所连接的网络模块的 I/O 编号除以 16 得到的值。
  7. 指定 iQSS 备份发生错误时的动作。  
继续执行...对多个 iQSS 兼容传感器执行备份时，即便有部分机器备份失败，也继续执行处理。  
停止...对多个 iQSS 兼容传感器执行备份时，一旦有部分机器备份失败，就停止处理。
  8. 切换至上次显示画面。
  9. 保存设置内容。保存完成时会显示完成对话框。如果设置内容不完备，会显示错误对话框。
  10. 显示当前日期和时间。触摸即显示时钟设置窗口。
  11. 显示语言设置窗口。

**备注**

- 通过画面脚本和配方功能实现 iQSS 备份设置的登录。关于脚本，请参照「5.9 脚本一览表」；关于配方功能，请参照「5.8 配方一览表」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5.3.23 iQSS 备份进展(B-30103)



概要

显示 iQSS 备份的进展状况和结果。

详细

1. 切换至 iQSS 菜单画面。在 iQSS 备份中不显示，在完成 iQSS 备份后显示。
2. 显示 iQSS 备份的设置。
3. 显示开始 iQSS 备份的日期和时间。
4. 用%和条形图显示 1 台 iQSS 备份对象机器的备份进展。1 台 iQSS 备份对象机器的备份完成时显示「100%」，条形图的颜色从黑色变成蓝色。有多台 iQSS 备份对象机器时，逐一显示进展。
5. 显示 iQSS 备份对象机器的总数。
6. 显示 iQSS 备份成功和失败的机器数量。
7. 中止 iQSS 备份。在 iQSS 备份中显示，在完成 iQSS 备份后不显示。
8. 显示当前日期和时间。触摸即显示时钟设置窗口。
9. 显示语言设置窗口。

备注

- 中止 iQSS 备份，不触摸确认窗口的「OK」是不会中止的。
- iQSS 备份失败时，对象机器总数/备份成功/备份失败有可能各自清零。
- 通过画面脚本和配方功能实现 iQSS 备份记录的保存。关于脚本，请参照「5.9 脚本一览表」；关于配方功能，请参照「5.8 配方一览表」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。



### 概要

依据 iQSS 备份记录相对应的 iQSS 恢复设置的参数执行 iQSS 恢复。或者对 iQSS 备份记录相对应的 iQSS 恢复设置进行确认，或者对 iQSS 备份记录进行删除。

### 详细

1. 以一览表显示 iQSS 备份记录，每次 10 件，最多 300 件。触摸 iQSS 备份记录即显示光标，成为选择状态。从上往下按时间降序显示 iQSS 备份记录。
2. 删除选择状态的 iQSS 备份记录。删除 1 件 iQSS 备份记录后，比删除掉的 iQSS 备份记录更旧的记录会逐行上移显示。
3. 删除全部 iQSS 备份记录。
4. 上下翻页显示一览表。
5. 切换至 iQSS 恢复进展画面的同时，依据选择状态的 iQSS 备份记录相对应的恢复设置的参数执行 iQSS 恢复。如果选择了没有显示 iQSS 备份记录的地方执行 iQSS 恢复，就会显示错误对话框。
6. 切换至 iQSS 恢复设置画面。
7. 切换至各画面。蓝色开关为当前显示的画面，所以显示中的画面不被切换。
8. 未使用的基本画面切换开关。
9. 切换至上次显示画面。
10. 显示当前日期和时间。触摸即显示时钟设置窗口。
11. 显示语言设置窗口。

#### 备注

- 执行 iQSS 备份时，只有在可编程控制器上安装的 SD 卡内生成 iQSS 备份数据时才会生成 iQSS 备份记录。
- 无法从异常结束或者中断的 iQSS 备份记录相对应的 iQSS 恢复设置进行恢复。
- 删除操作是删除 GOT 上安装的 SD 卡内保存的 iQSS 备份记录。请注意实际的 iQSS 备份数据是保存在可编程控制器 CPU 上安装的 SD 卡内，无法从 GOT 删除的格式。  
另请注意从 SD 卡删除 iQSS 备份数据后，将无法从相应的 iQSS 备份记录恢复。
- 可编程控制器 CPU 上安装的 SD 卡内的文件夹构成，请参照「7.5 iQSS 备份文件夹构成」。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

概要

确认 iQSS 恢复设置。iQSS 恢复设置的内容，除了文件夹编号之外，和执行 iQSS 备份时的设置内容相同。

详细

1. 显示 iQSS 备份记录的编号。
2. 切换至上次显示画面。
3. 显示 iQSS 备份设置的名称。
4. 显示执行 iQSS 备份时的对象机种。
5. 显示 iQSS 恢复对象 (=执行 iQSS 备份时的 iQSS 备份对象)。
6. 显示执行 iQSS 备份时存储 iQSS 备份数据的文件夹编号。
7. 显示执行 iQSS 备份时 iQSS 备份对象机器所连接的网络模块的 I/O 编号。
8. 显示执行 iQSS 恢复发生错误时的动作。和执行 iQSS 备份时的设置相同。
9. 显示当前日期和时间。触摸即显示时钟设置窗口。
10. 显示语言设置窗口。

备注

- 显示和 iQSS 备份设置画面相同。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。

5.3.26 iQSS 恢复进展(B-30106)

恢复对象	
设置名称	ASLINK3
执行单位	ID指定
对象机器	14
I/O	3

恢复处理	
开始时间	2012/08/09 10:12
进展	100%
对象机器总数	1
恢复成功	1
恢复失败	0

iQSS 菜单      恢复中止

概要

显示 iQSS 恢复的进展。

详细

1. 切换至 iQSS 菜单画面。在 iQSS 恢复中不显示，在完成 iQSS 恢复后显示。
2. 显示 iQSS 恢复的设置。iQSS 恢复设置的内容，和执行 iQSS 备份时的设置内容相同。
3. 显示开始 iQSS 恢复的日期和时间。
4. 用%和条形图显示 1 台 iQSS 恢复对象机器的恢复进展。1 台 iQSS 恢复对象机器的恢复完成时显示「100%」，条形图的颜色从黑色变成蓝色。有多台 iQSS 恢复对象机器时，逐一显示进展。
5. 显示 iQSS 恢复对象机器的总数。
6. 显示 iQSS 恢复成功和失败的机器数量。
7. 中止 iQSS 恢复。在 iQSS 恢复中显示，在完成 iQSS 恢复后不显示。
8. 显示当前日期和时间。触摸即显示时钟设置窗口。
9. 显示语言设置窗口。

备注

- 中止 iQSS 恢复，不触摸确认窗口的「OK」是不会中止的。
- iQSS 恢复发生错误时，对象机器总数/恢复成功/恢复失败有可能各自清零。
- 每秒通过工程脚本监视报警以及异常的件数。一旦两者的件数产生差别，就使用软元件数据传送功能读取最新的参数。关于脚本，请参照「5.9 脚本一览表」；关于软元件数据传送功能，请参照「5.7 软元件数据传送一览表」。
- 画面切换时，将关闭显示中的窗口。
- 系统报警发生时，在画面的下方将显示报警信息。触摸报警信息后，显示报警复位窗口。



### 5. 3. 27 报警复位(W-30001)



#### 概要

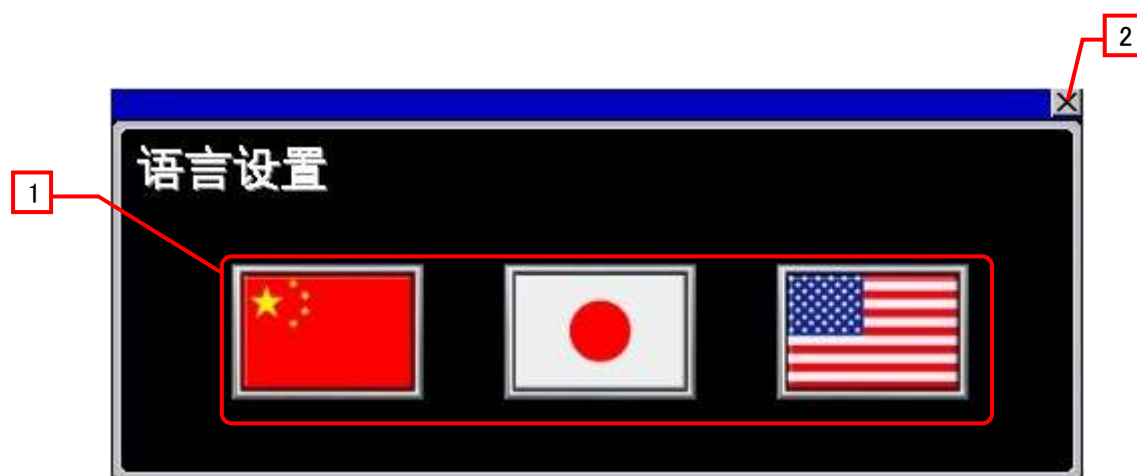
复位系统报警。

#### 详细

1. 复位系统报警，并在 1 秒后关闭窗口画面。
2. 关闭窗口画面。

#### 备注

### 5. 3. 28 语言设置(W-30002)



#### 概要

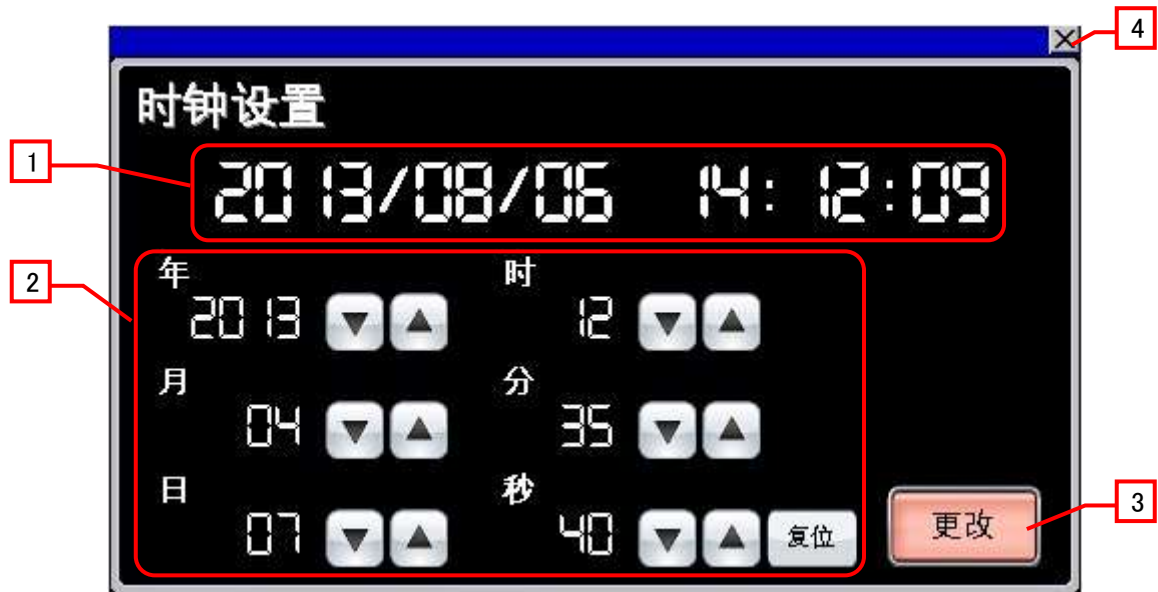
选择 GOT 的显示语言。

#### 详细

1. 切换语言，并关闭窗口画面。
2. 关闭窗口画面。

#### 备注

- 画面显示语言与系统语言同步切换。



#### 概要

更改 GOT 的时钟数据。

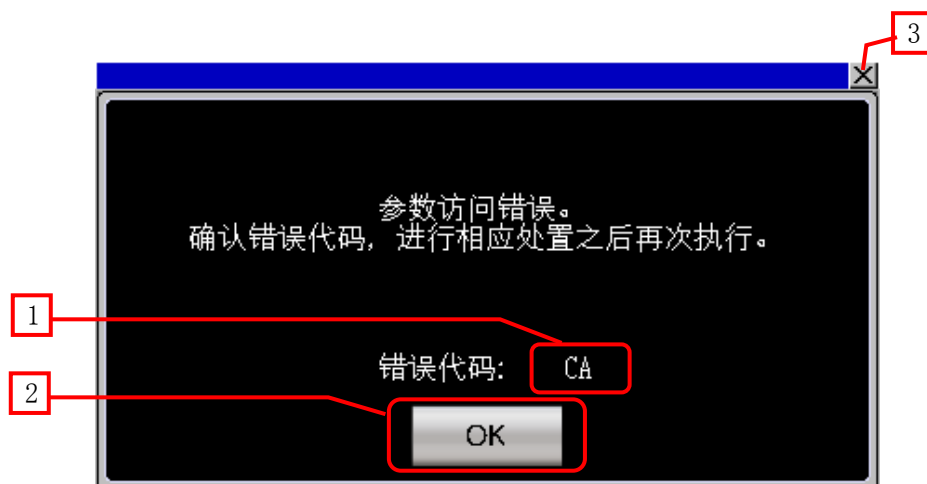
#### 详细

1. 显示当前日期和时间。
2. 通过 ▼ ▲ 开关设置想更改的日期和时间。长按 ▼ ▲ 开关将连续进行增减。复位开关复位秒。
3. 将设置的日期和时间反映到 GOT 的时钟数据中，并在 1 秒后关闭窗口画面。
4. 关闭窗口画面。

#### 备注

- 日期和时间的初始值为窗口画面显示时的日期和时间。
- 更改日期和时间的年/月/日/时/分/秒的数值显示中设置了对象脚本。  
关于脚本的详细内容，请参照「5.9 脚本一览表」。

### 5.3.30 参数访问错误(W-30004)



#### 概要

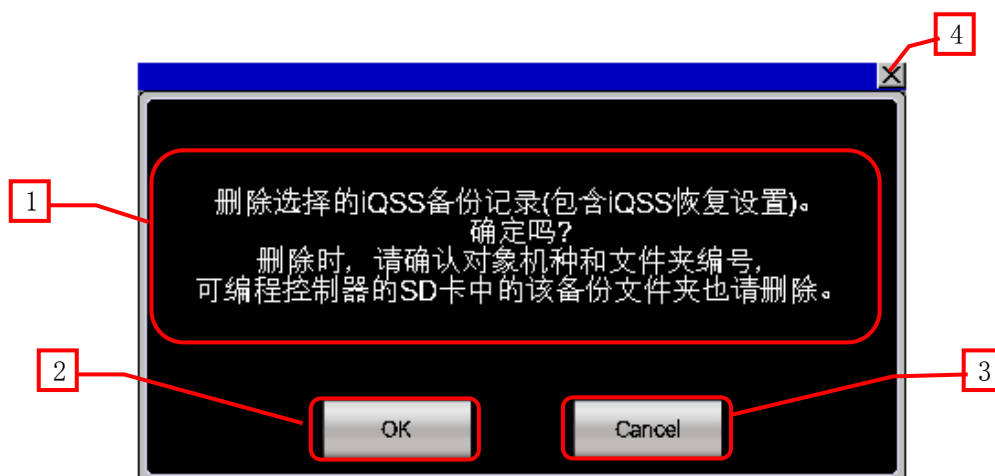
参数访问中发生错误时显示。

#### 详细

1. 显示正在发生的报警或者异常的错误代码以及代码所对应的错误内容。有多个报警、异常发生时, 显示最新的错误代码。
2. 关闭窗口画面。
3. 关闭窗口画面。

#### 备注

### 5.3.31 数据删除确认对话框(W-30100)



#### 概要

确认删除 iQSS 备份设置或者 iQSS 备份记录。

#### 详细

1. 显示信息。
2. 删除对象的数据, 并关闭窗口画面。
3. 关闭窗口画面。
4. 关闭窗口画面。

#### 备注

- 显示的信息, 因开启窗口画面的开关而异。

### 5. 3. 32 通知对话框(W-30101)



#### 概要

通知处理结果。

#### 详细

1. 显示信息。
2. 关闭窗口画面。
3. 关闭窗口画面。

#### 备注

- 信息的内容，因弹出窗口的脚本而异。关于脚本的详细内容，请参照「5.9 脚本一览表」。

### 5. 3. 33 执行确认对话框 (W-30102)



#### 概要

各种动作前进行确认。

#### 详细

1. 显示信息。
2. 执行各种动作，并关闭窗口画面。
3. 关闭窗口画面。
4. 关闭窗口画面。

#### 备注

- 关于信息的内容、动作，因所显示的基本画面而异。
  - iQSS 备份画面……………执行 iQSS 备份
  - iQSS 备份进展画面……………中止 iQSS 备份
  - iQSS 恢复画面……………执行 iQSS 恢复
  - iQSS 恢复进展画面……………中止 iQSS 恢复

## 5.4 从属模块详细信息画面对照表

从属模块 型号编号	型号	大分类	基本画面号
100	B281SB-02U-CC20	ASLINKER	B-30006
101	B281SB-02US-CC20	ASLINKER	B-30006
102	B280SB-02U-C1220	ASLINKER	B-30008
103	B280SB-02US-C1220	ASLINKER	B-30008
104	BL287SB-02F-CC20	ASLINKER	B-30007
105	BL287SB-02FS-CC20	ASLINKER	B-30007
106	B298SB-02U-M12	ASLINKER	B-30009
107	B298SB-02US-M12	ASLINKER	B-30009
400	B281PB-02U-CC20	ASLINKER	B-30006
401	B281PB-02US-CC20	ASLINKER	B-30006
402	B280PB-02U-C1220	ASLINKER	B-30008
403	B280PB-02US-C1220	ASLINKER	B-30008
404	BL287PB-02F-CC20	ASLINKER	B-30007
405	BL287PB-02FS-CC20	ASLINKER	B-30007
406	B298PB-02U-M12	ASLINKER	B-30009
407	B298PB-02US-M12	ASLINKER	B-30009
700	B281XB-02U-CC20	ASLINKER	B-30006
701	B281XB-02US-CC20	ASLINKER	B-30006
702	B280XB-02U-C1220	ASLINKER	B-30008
703	B280XB-02US-C1220	ASLINKER	B-30008
704	BL287XB-02F-CC20	ASLINKER	B-30007
705	BL287XB-02FS-CC20	ASLINKER	B-30007
706	B298XB-02U-M12	ASLINKER	B-30009
707	B298XB-02US-M12	ASLINKER	B-30009
1000	B289SB-01AP-CAM20	ASLINKER 放大器	B-30010
1050	B289SB-01AP-CAS	ASLINKER 放大器	B-30010
1200	B289SB-01AK-CAM20	ASLINKER 放大器	B-30011
1250	B289SB-01AK-CAS	ASLINKER 放大器	B-30011
1400	B289SB-01AF-CAM20	ASLINKER 放大器	B-30012
1450	B289SB-01AF-CAS	ASLINKER 放大器	B-30012
2000	B285SB-01-1K1	ASLINKER 传感器	B-30013
2300	B283SB-01-1KR	ASLINKER 传感器	B-30014
2301	B283SB-01-1KS	ASLINKER 传感器	B-30014
2302	B283SB-01-1KC	ASLINKER 传感器	B-30015
2303	B283SB-01-1KP	ASLINKER 传感器	B-30016
2600	B295SB-01-1K26	ASLINKER 传感器	B-30017
2900	B284SB-01-12	ASLINKER 传感器	B-30018
A000	BL296SB-08F-3	ASLINKER 终端	B-30019
A001	BL296SB-08FS-3	ASLINKER 终端	B-30019
A300	BL296PB-08F-3	ASLINKER 终端	B-30019
A301	BL296PB-08FS-3	ASLINKER 终端	B-30019
A600	BL296XB-08F-3	ASLINKER 终端	B-30019
A601	BL296XB-08FS-3	ASLINKER 终端	B-30019



## 5.5 使用软元件一览表

画面上的开关和指示灯等使用的软元件，有些同时也在脚本等的公共设置中被使用。统一更改此类软元件时，推荐使用[批量更改]。关于[批量更改]的详细内容，请参照「GT Designer3 (GOT2000) 帮助」

### 5.5.1 连接机器的软元件

类型	软元件号	用途
位	X0041	参数访问完成标志
	X0042	参数访问异常
	Y0030	异常标志清除指令
	Y0031	地址自动识别指令
	Y0040	主模块发往从属模块的参数访问请求指令
	Y0041	主模块发往从属模块的参数一起读取指令
	SM1435	iQ Sensor Solution 兼容备份/恢复执行允许位
	SM1436	iQ Sensor Solution 兼容备份请求位
	SM1437	iQ Sensor Solution 兼容备份正常完成位
	SM1438	iQ Sensor Solution 兼容备份异常完成位
	SM1439	iQ Sensor Solution 兼容恢复请求位
	SM1440	iQ Sensor Solution 兼容恢复正常完成位
	SM1441	iQ Sensor Solution 兼容恢复异常完成位
	SM1442	iQ Sensor Solution 兼容备份/恢复中止请求位
字	U03-G0~U03-G15	输入信息区域 (U03-G0, 用作软元件数据传送的基准软元件)
	U03-G1	用作软元件数据传送的基准软元件
	U03-G20	用作软元件数据传送的基准软元件
	U03-G40	用作传感器测定值水平的基准软元件
	U03-G4096 ~ U03-G4111	输出信息区域
	U03-G8192	异常 ID 个数信息
	U03-G8193 ~ U03-G8320	已成 ID 信息存储区域
	U03-G8960	连接台数信息
	U03-G9216	连接 ID 个数信息
	U03-G9217 ~ U03-G9344	连接 ID 信息存储区域
	U03-G9984	报警 ID 个数信息
	U03-G9985 ~ U03-G10112	报警 ID 信息存储区域
	U03-G10256	当前错误代码存储区域
	U03-G10320	参数访问设置
	U03-G10321	参数访问对象 ID 指定
	U03-G10496 ~ U03-G10751	参数存储目标存储器号 (输出)
	U03-G11008 ~ U03-G11263	参数存储目标存储器号 (输入)
	U03-G12288 ~ U03-G18431	参数存储区域
	SD1435	iQ Sensor Solution 兼容备份/恢复使用请求软元件
	SD1436	iQ Sensor Solution 兼容备份/恢复使用权获取状况软元件

类型	软元件号	用途
字	SD1437	iQ Sensor Solution 兼容备份/恢复对象机器/执行单位设置软元件
	SD1438	iQ Sensor Solution 兼容备份/恢复对象文件夹编号设置软元件
	SD1439	iQ Sensor Solution 兼容备份/恢复对象设置软元件(对象模块)
	SD1440	iQ Sensor Solution 兼容备份/恢复对象设置软元件(对象机器 1)
	SD1441	iQ Sensor Solution 兼容备份/恢复对象设置软元件(对象机器 2)
	SD1444	iQ Sensor Solution 兼容备份/恢复动作设置软元件
	SD1446	iQ Sensor Solution 兼容备份/恢复状态软元件
	SD1447	iQ Sensor Solution 兼容备份/恢复执行状况软元件(总对象机器数)
	SD1448	iQ Sensor Solution 兼容备份/恢复执行状况软元件(正常完成机器数)
	SD1449	iQ Sensor Solution 兼容备份/恢复执行状况软元件(异常完成机器数)
	SD1450	iQ Sensor Solution 兼容备份/恢复执行状况软元件(机器单位进展)
	SD1451	iQ Sensor Solution 兼容备份文件夹编软元件
	SD1452	iQ Sensor Solution 兼容备份/恢复模块错误原因
	SD1453	iQ Sensor Solution 兼容备份/恢复对象机器错误原因

### 5.5.2 GOT 内部软元件

类型	软元件号	用途
位	GB40	脚本触发(通常 ON)
	GB41	清零各种位软元件 (通常 OFF)
	GB30000	脚本号 30005 启动触发
	GB30001	脚本号 30003 启动触发
	GB30002	脚本号 30017 启动触发
	GB30003	脚本号 30004 初次启动脚本启动控制标志
	GB30004	脚本号 30006 启动触发
	GB30005	脚本号 30007 启动触发
	GB30006	脚本号 30010 启动触发
	GB30007	脚本号 30023 初次启动脚本启动控制标志
	GB30008	脚本号 30015 启动触发
	GB30009	脚本号 30017 ID 一致标志
	GB30010	脚本号 30001 启动触发
	GB30011	脚本号 30030 启动触发
	GB30012	脚本号 30031 启动触发
	GB30013	脚本号 30032 启动触发
	GB30014	脚本号 30024 启动触发
	GB30015	脚本号 30025 启动触发
	GB30016	OFF 中脚本控制标志
	GB30017	脚本号 30027 初次启动脚本启动控制标志
	GB30018	脚本号 30034 启动触发
	GB30019	脚本号 30038 启动触发
	GB30020	脚本号 30017 混合模块标志
	GB30021	脚本号 30026 控制标志
	GB30024	参数个别读取指示灯位
	GB30025	参数个别写入指示灯位
	GB30026	脚本号 30040 启动触发
	GB30027	脚本号 30041 初次启动脚本启动控制标志
	GB30028	异常发生标志
	GB30029	脚本号 30068 启动触发
	GB30030	脚本号 30069 启动触发
	GB30044	参数同时读取标志
	GB30600	输出判断标志
	GB30601	输入判断标志
	GB54000	脚本号 30046 启动触发
	GB54001	脚本号 30047 启动触发
	GB54002	脚本号 30065 启动触发
	GB54004	iQSS 备份许可标志
	GB54005	脚本号 30056 启动触发
	GB54006	脚本号 30058 启动触发
	GB54008	脚本号 30055 初次启动脚本启动控制标志
	GB54009	脚本号 30043、30045 控制标志
	GB54010	脚本号 30062 启动触发

类型	软元件号	用途
位	GB54011	脚本号 30064 启动触发
	GB54012	脚本号 30058 删除范围设置标志
	GB54013	脚本号 30050 启动触发
	GB54014	B-30100 画面切换开关动作条件
	GB54016	配方号 30001 写入触发
	GB54017	配方号 30001 读取触发
	GB54018	配方号 30002 写入触发
	GB54019	配方号 30002 读取触发
字	GD31000~GD31255	B-30004 从属模块状态显示指示灯
	GD31256~GD31511	B-30003 从属模块状态显示指示灯
	GD31520	B-30003、B-30004 光标显示的移位用软元件
	GD31521	B-30003、B-30004 光标显示的偏置用软元件
	GD31522	B-30003、B-30004 当前光标位置软元件
	GD31523	B-30002 偏置值运算用软元件
	GD31524	B-30002 输入输出检查软元件
	GD31525	B-30002 系统图滚动用偏置
	GD31526	B-30002 系统图部件显示用偏置
	GD31528	B-30005 传感器一览详细显示用偏置
	GD31529	B-30005 传感器一览滚动用偏置
	GD31562	软元件数据传送 1 外部控制软元件
	GD31563	软元件数据传送 1 外部通知软元件
	GD31564	软元件数据传送 2 外部控制软元件
	GD31565	软元件数据传送 2 外部通知软元件
	GD31567	软元件数据传送 2、软元件数据传送 4 的偏置
	GD31568	软元件数据传送 3 外部控制软元件
	GD31569	软元件数据传送 3 外部通知软元件
	GD31570	软元件数据传送 4 外部控制软元件
	GD31571	软元件数据传送 4 外部通知软元件
	GD31600~GD31606	脚本用工作区域
	GD31607	B-30006~30019 从属模块型号编号存储软元件
	GD31608	B-30006~30019 状态状态
	GD31609~GD31616	B-30006~30019 I/O 监视的部件显示
	GD31617~GD31631	B-30005 模块的种类
	GD31633~GD31647	B-30005 状态详细
	GD31649~GD31663	B-30005 用户定义名称
	GD31665~GD31679	B-30005 状态显示指示灯
	GD31681	B-30006~30019 处理方法的显示起始行软元件
	GD31682~GD31989	B-30006~30019 详细信息的 I/O 用偏置
	GD31990~GD31997	B-30006~30019 详细信息的 I/O 的屏蔽用软元件(对象脚本)
	GD32000~GD32007	B-30006~30019 详细信息的 I/O 的字指示灯状态软元件
	GD32008	B-30003、B-30004 光标位置显示用软元件

类型	软元件号	用途
字	GD33020	脚本号 30003 输入/输出区域检查用软元件
	GD35000～GD35127	B-30002 系统图显示用软元件
	GD35128	B-30002 传感器类型 偏置用软元件
	GD35130～GD35642	B-30003、B-30004 模块类型存储用软元件
	GD35643	B-30005 模块信息显示触发软元件
	GD40000～GD40015	输入信息区域
	GD40016～GD40031	输出信息区域
	GD40032	异常 ID 个数
	GD40033～GD40160	连接 ID 信息存储区域
	GD40289	连接台数信息
	GD40290	连接 ID 个数
	GD40291～GD40418	连接 ID 信息存储区域
	GD40547	报警个数
	GD40548～GD40675	报警 ID 信息存储区域
	GD40804～GD41059	参数存储目标存储器号(输出)
	GD41060～GD41315	参数存储目标存储器号(输入)
	GD41316～GD47459	机器参数(批量)
	GD53604～GD53632	机器参数(个别)
	GD54000	B-30101 光标显示软元件
	GD54001	B-30101 数据显示偏置软元件
	GD54002	B-30104 光标显示软元件
	GD54003	B-30104 数据显示偏置软元件
	GD54004	配方动作控制软元件
	GD54005	B-30101 号显示软元件
	GD54006	W-30100～30101 注释显示软元件
	GD54007	B-30101 数据移动用偏置软元件
	GD54008	B-30104 数据移动用偏置软元件
	GD54009	配方外部控制软元件
	GD54010	配方号存储软元件
	GD54011	记录号存储软元件
	GD54012	配方外部通知软元件
	GD54013	配方号通知软元件
	GD54014	记录号通知软元件
	GD54015～GD54034	B-30101、B-30102 备份设置工作区域
	GD54035～GD54039	B-30102 备份设置注释显示软元件
	GD54040～GD54042	B-30103、B-30106 备份/恢复开始时间存储软元件
	GD54043～GD54062	B-30105 备份数据详细显示用区域
	GD54070～GD54074	B-30105 备份数据详细注释显示软元件
	GD54075	脚本号 30042 对象机器判断软元件
	GD54076～GD54085	B-30101 执行单位注释显示软元件
	GD54086	B-30101、B-30102 模块类型选择开关显示标志
	GD54087	AnyWireASLINK 的备份件数
	GD54088	CC-Link 的备份件数

类型	软元件号	用途
字	GD54089	脚本号 30042 机器专用画面设置时的 I/O 编号
	GD54090	配方号 30001 记录号
	GD54091	配方号 30002 记录号
	GD54100～GD56199	备份设置存储区域
	GD56200～GD61599	备份结果存储区域
	GD62000	基本画面切换软元件
	GD62001	重叠窗口 1 画面切换软元件
	GD62004	重叠窗口 2 画面切换软元件
	GD62007	重叠窗口 3 画面切换软元件
	GD62021	语言切换软元件
	GD62022	系统语言切换软元件
	GD62300	时间更改软元件(年)
	GD62301	时间更改软元件(月)
	GD62302	时间更改软元件(日)
	GD62303	时间更改软元件(时)
	GD62304	时间更改软元件(分)
	GD62305	时间更改软元件(秒)
	GS386	工程/画面脚本初次动作控制软元件
	GS513～GSS516	更改时间软元件
	GS650～GSS652	当前时间软元件
	GS654	触摸状态外部通知(X 坐标)
	GS655	触摸状态外部通知(Y 坐标)
	TMP0～TMP28、TMP100 ～ TMP105、TMP110～ TMP111 、 TMP120 ～ TMP126 、 TMP200 ～ TMP206 、 TMP211 ～ TMP213 、 TMP220 ～ TMP223 、 TMP1000 、 TMP1001、 TMP1010、 TMP1020	脚本运算用

## 5.6 注释一览表

注释组号	注释号	使用处
250	No. 1	B-30006～30019
	No. 2	B-30006～30019
	No. 4	B-30006～30019
	No. 8	B-30006～30019
	No. 32	B-30006～30019
	No. 100	B-30006～30019
	No. 101	B-30006～30019
	No. 102	B-30006～30019
	No. 103	B-30006～30019
	No. 104	B-30006～30019
	No. 200	B-30006～30019
	No. 201	B-30006～30019
	No. 202	B-30006～30019
	No. 300	B-30006～30019
	No. 301	B-30006～30019
	No. 302	B-30006～30019
	No. 303	B-30006～30019
	No. 304	B-30006～30019
	No. 305	B-30006～30019
	No. 400	B-30006～30019
	No. 401	B-30006～30019
	No. 500	B-30006～30019
251	请参照「5.4 从属模块详细信息画面对照表」。GOT 为每个从属模块的模块型号编号设置了注释号及用来显示的基本画面号。模块型号编号从 A0000 开始的从属模块设置了从 10000 开始的 GOT 注释号。(GOT 的注释号无法设置成 A0000。)	
252		
253	No. 1～255、512～767	B-30005～30019
254	No. 1	B-30002、B-30005～30019
	No. 2	B-30002、B-30005～30019
	No. 4	B-30002、B-30005～30019
	No. 8	B-30002、B-30005～30019
	No. 32	B-30002、B-30005～30019
	No. 100	B-30002、B-30005～30019、W-30101
	No. 101	B-30002、B-30005～30019、W-30101
	No. 102	B-30002、B-30005～30019、W-30101
	No. 103	B-30002、B-30005～30019、W-30101
	No. 104	B-30002、B-30005～30019、W-30101
	No. 200	B-30002、B-30005～30019、W-30101
	No. 201	B-30002、B-30005～30019、W-30101
	No. 202	B-30002、B-30005～30019、W-30101
	No. 300	B-30002、B-30005～30019、W-30101
	No. 301	B-30002、B-30005～30019、W-30101
	No. 302	B-30002、B-30005～30019、W-30101
	No. 303	B-30002、B-30005～30019、W-30101
	No. 304	B-30002、B-30005～30019、W-30101
	No. 305	B-30002、B-30005～30019、W-30101
	No. 400	B-30002、B-30005～30019、W-30101

注释组号	注释号	使用处
254	No. 401	B-30002、B-30005～30019、W-30101
	No. 500	B-30002、B-30005～30019、W-30101
	No. 18433	W-30101
	No. 18434	W-30101
	No. 18435	W-30101
	No. 18436	W-30101
	No. 18437	W-30101
	No. 18438	W-30101
	No. 18439	W-30101
	No. 18440	W-30101
	No. 18441	W-30101
	No. 32000	W-30101
	No. 32001	W-30101
	No. 32002	W-30101
	No. 32003	W-30101
	No. 32004	W-30101
	No. 32005	W-30101
	No. 32006	W-30101
	No. 32007	W-30101
	No. 32008	W-30101
	No. 32009	W-30101
	No. 32010	W-30101
	No. 32011	W-30101
	No. 32012	W-30101
	No. 32013	W-30101
	No. 32014	W-30101
	No. 32015	W-30101
	No. 32016	W-30101
	No. 32017	W-30101
	No. 32767	B-30002、B-30005～30019
255	No. 20	B-30002
	No. 22	B-30005
	No. 23	B-30005～30019
	No. 24	B-30005～30019
	No. 25	B-30006～30019
	No. 26	B-30002～30019
	No. 27	B-30001、B-30002
	No. 28	B-30002～30019
	No. 29	B-30001～30019
	No. 30	B-30001
	No. 31	B-30001
	No. 32	B-30002～30004
	No. 33	B-30002～30004
	No. 34	B-30002～30004
	No. 35	B-30002～30004
	No. 36	B-30002～30005
	No. 37	B-30002
	No. 38	B-30002～30004、W-30001
	No. 39	B-30002～30004
	No. 40	B-30002～30005、W-30001、W-30003



注释组号	注释号	使用处
255	No. 41	B-30002～30019
	No. 42	B-30002～30019
	No. 43	B-30006～30019
	No. 44	—
	No. 45	B-30003、B-30004
	No. 46	B-30003、B-30004
	No. 47	B-30003、B-30004
	No. 48	B-30003、B-30004
	No. 49	B-30005
	No. 50	B-30005
	No. 51	B-30005～30019
	No. 52	B-30006～30019
	No. 53	B-30006～30019
	No. 54	B-30006～30019
	No. 55	B-30006～30019
	No. 56	B-30006～30019
	No. 57	B-30006～30019
	No. 58	B-30006～30019
	No. 59	B-30006、B-30008、B-30009
	No. 60	B-30006～30019
	No. 61	B-30006～30019
	No. 62	B-30010～30015、B-30017、B-30018
	No. 63	B-30010～30015、B-30017、B-30018
	No. 64	B-30013、B-30018
	No. 65	B-30013、B-30018
	No. 66	B-30013、B-30018
	No. 67	B-30013、B-30018
	No. 68	B-30011、B-30013、B-30017、B-30018
	No. 69	B-30013、B-30017、B-30018
	No. 70	B-30013～30015、B-30017、B-30018
	No. 71	B-30013～30015、B-30017、B-30018
	No. 72	B-30013、B-30017、B-30018
	No. 73	B-30013、B-30017、B-30018
	No. 74	B-30013～30015、B-30017、B-30018
	No. 75	B-30006～30019
	No. 76	B-30006、B-30008、B-30009
	No. 77	B-30006、B-30008、B-30009
	No. 78	W-30004
	No. 79	W-30004
	No. 80	W-30004
	No. 81	W-30003
	No. 82	W-30003
	No. 83	W-30003
	No. 84	W-30003
	No. 85	W-30003
	No. 86	W-30003
	No. 87	W-30003
	No. 88	W-30003
	No. 89	—

注释组号	注释号	使用处
255	No. 90	—
	No. 91	B-30003
	No. 92	B-30004
	No. 93	B-30006～30019
	No. 94	B-30001
	No. 95	B-30002～30019
	No. 96	B-30001、B-30004
	No. 97	B-30001、B-30003
	No. 98	—
	No. 99	B-30100
	No. 100	—
	No. 101	—
	No. 102	—
	No. 103	—
	No. 104	—
	No. 105	—
	No. 106	—
	No. 107	—
	No. 108	—
	No. 109	—
	No. 110	—
	No. 111	—
	No. 112	—
	No. 113	—
	No. 114	—
	No. 115	—
	No. 116	—
	No. 117	B-30010～30012、B-30014、B-30015、B-30017
	No. 118	B-30010～30012、B-30014、B-30015、B-30017
	No. 119	B-30010～30012、B-30014、B-30015、B-30017
	No. 120	B-30010～30012、B-30014、B-30015、B-30017
	No. 121	B-30010、B-30012、B-30014、B-30015
	No. 122	B-30010、B-30012、B-30014、B-30015
	No. 123	B-30010～30012、B-30014、B-30015
	No. 124	B-30014、B-30015
	No. 125	B-30014、B-30016
	No. 126	B-30010～30012、B-30014、B-30015
	No. 127	B-30010～30012、B-30014～30016
	No. 128	W-30002
	No. 129	B-30010、B-30012
	No. 130	B-30010、B-30012
	No. 131	B-30010、B-30012
	No. 132	B-30010、B-30012
	No. 133	B-30010～30012
	No. 134	B-30010～30012
	No. 135	B-30010～30012
	No. 136	B-30010、B-30012
	No. 137	B-30010、B-30012
	No. 138	B-30011

注释组号	注释号	使用处
255	No. 139	B-30011
	No. 140	B-30011
	No. 141	B-30011
	No. 142	B-30011
	No. 143	B-30011
	No. 144	B-30014、B-30016
	No. 145	B-30006～30019
	No. 201	B-30000、B-30100
	No. 202	B-30101
	No. 203	B-30102
	No. 204	B-30103
	No. 205	B-30104
	No. 206	B-30105
	No. 207	B-30106
	No. 208	B-30100
	No. 209	B-30100
	No. 210	B-30101、B-30102、B-30104、B-30105
	No. 211	B-30102、B-30103、B-30105、B-30106
	No. 212	B-30101～30106
	No. 213	B-30103、B-30106
	No. 214	B-30101
	No. 215	B-30101
	No. 216	B-30101
	No. 217	B-30101
	No. 218	B-30101、B-30104
	No. 219	B-30101、B-30103、B-30104、B-30106
	No. 220	B-30101、B-30104
	No. 221	B-30101、B-30104
	No. 222	B-30101、B-30102、B-30104、B-30105
	No. 223	B-30102、B-30105
	No. 224	B-30102、B-30105
	No. 225	B-30101、B-30103、B-30106
	No. 226	B-30101、B-30103、B-30106
	No. 227	B-30103、B-30106
	No. 228	B-30102、B-30105
	No. 229	B-30105
	No. 230	B-30102、B-30105
	No. 231	B-30102
	No. 232	B-30103、B-30105
	No. 233	B-30103、B-30105
	No. 234	B-30102、B-30105
	No. 235	B-30102、B-30105
	No. 236	B-30102
	No. 237	B-30103
	No. 238	B-30103
	No. 239	B-30103、B-30106
	No. 240	B-30103、B-30106
	No. 241	B-30103、B-30106
	No. 242	B-30103、B-30105

注释组号	注释号	使用处
255	No. 243	B-30103、B-30105
	No. 244	B-30103、B-30105
	No. 245	B-30101～30103、B-30105、B-30106
	No. 246	B-30103
	No. 247	B-30103
	No. 248	B-30103
	No. 249	B-30104
	No. 250	B-30104
	No. 251	B-30104
	No. 252	B-30106
	No. 253	B-30106
	No. 254	B-30106
	No. 255	B-30106
	No. 256	B-30106
	No. 257	W-30004、W-30100～30102
	No. 258	W-30100、W-30102
	No. 259	—
	No. 260	—
	No. 261	—
	No. 262	—
	No. 263	—
	No. 264	—
	No. 265	—
	No. 266	—
	No. 267	—
	No. 268	W-30102
	No. 269	W-30102
	No. 270	W-30102
	No. 271	W-30102
	No. 272	W-30100
	No. 273	B-30104
	No. 274	B-30104
	No. 275	B-30101、B-30104
	No. 276	B-30101、B-30104
	No. 277	B-30101、B-30104
	No. 278	B-30104
	No. 279	B-30101、B-30104
	No. 280	W-30102
	No. 281	W-30102
	No. 282	W-30100
	No. 283	B-30102、B-30103、B-30105、B-30106
	No. 284	B-30102、B-30103、B-30105、B-30106
	No. 285	—
	No. 286	B-30103、B-30106
	No. 287	B-30101
	No. 288	B-30101
	No. 289	W-30100

## 5.7 软元件数据传送一览表

ID: 201 传送 1

项 目		设 置
软元件数据传送触发	触发类型	上升沿
	外部控制软元件	GD31562
	触发软元件	GD31562. b0
	传送源目标互换标志	GD31562. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD31563
	软元件数据传送处理中通知信号	GD31563. b0
	软元件数据传送错误通知信号	GD31563. b15
软元件	块数	10
块 1	软元件类型	有符号 BIN16
	点数	16
	传送源软元件	U03-G0
	传送目标软元件	GD40000
	偏置	无
块 2	软元件类型	有符号 BIN16
	点数	16
	传送源软元件	U03-G4096
	传送目标软元件	GD40016
	偏置	无
块 3	软元件类型	有符号 BIN16
	点数	129
	传送源软元件	U03-G8192
	传送目标软元件	GD40032
	偏置	无
块 4	软元件类型	有符号 BIN16
	点数	1
	传送源软元件	U03-G8960
	传送目标软元件	GD40289
	偏置	无

项 目		设 置
块 5	软元件类型	有符号 BIN16
	点数	129
	传送源软元件	U03-G9216
	传送目标软元件	GD40290
	偏置	无
块 6	软元件类型	有符号 BIN16
	点数	129
	传送源软元件	U03-G9984
	传送目标软元件	GD40547
	偏置	无
块 7	软元件类型	有符号 BIN16
	点数	256
	传送源软元件	U03-G10496
	传送目标软元件	GD40804
	偏置	无
块 8	软元件类型	有符号 BIN16
	点数	256
	传送源软元件	U03-G11008
	传送目标软元件	GD41060
	偏置	无
块 9	软元件类型	位
	点数	1
	传送源软元件	GB41
	传送目标软元件	GD31562. b0
	偏置	无
块 10	软元件类型	位
	点数	1
	传送源软元件	GB41
	传送目标软元件	GB30016
	偏置	无

## ID: 202 传送 2

项 目		设 置
软元件数据传送触发	触发类型	上升沿
	外部控制软元件	GD31564
	触发软元件	GD31564. b0
	传送源目标互换标志	GD31564. b1
外部通知情報	<input checked="" type="checkbox"/> 外部通知软元件	GD31565
	软元件数据传送处理中通知信号	GD31565. b0
	软元件数据传送错误通知信号	GD31565. b15
软元件	块数	4
块 1	软元件类型	有符号 BIN16
	点数	1
	传送源软元件	U03-G0
	传送目标软元件	GD53604
	偏置	传送源 GD31567
块 2	软元件类型	有符号 BIN16
	点数	28
	传送源软元件	U03-G20
	传送目标软元件	GD53605
	偏置	传送源 GD31567
块 3	软元件类型	位
	点数	1
	传送源软元件	GB41
	传送目标软元件	GD31564. b0
	偏置	无
块 4	软元件类型	位
	点数	1
	传送源软元件	GB41
	传送目标软元件	GB30021
	偏置	无

## ID: 203 传送 3

项 目		设 置
软元件数据传送触发	触发类型	上升沿
	外部控制软元件	GD31568
	触发软元件	GD31568. b0
	传送源目标互换标志	GD31568. b1
外部通知情報	<input checked="" type="checkbox"/> 外部通知软元件	GD31569
	软元件数据传送处理中通知信号	GD31569. b0
	软元件数据传送错误通知信号	GD31569. b15
软元件	块数	2
块 1	软元件类型	有符号 BIN16
	点数	6144
	传送源软元件	U03-G12288
	传送目标软元件	GD41316
	偏置	无
块 2	软元件类型	位
	点数	1
	传送源软元件	GB41
	传送目标软元件	GD31568. b0
	偏置	无

## ID: 204 传送 4

项 目		设 置
软元件数据传送触发	触发类型	上升沿
	外部控制软元件	GD31570
	触发软元件	GD31570. b0
	传送源目标互换标志	GD31570. b1
外部通知情報	<input checked="" type="checkbox"/> 外部通知软元件	GD31571
	软元件数据传送处理中通知信号	GD31571. b0
	软元件数据传送错误通知信号	GD31571. b15
软元件	块数	3



项 目		设 置
块 1	软元件类型	有符号 BIN16
	点数	19
	传送源软元件	GD53605
	传送目标软元件	U03-G1
	偏置	传送目标 GD31567
块 2	软元件类型	位
	点数	1
	传送源软元件	GB40
	传送目标软元件	Y0040
	偏置	无
块 3	软元件类型	位
	点数	1
	传送源软元件	GB41
	传送目标软元件	GD31570. b0
	偏置	无

5.8 配方一览表

5.8.1 公共设置

外部控制信息	
外部控制软元件	GD54009
配方号存储软元件	GD54010
记录号存储软元件	GD54011
外部通知信息	
外部通知软元件	GD54012
配方号通知软元件	GD54013
记录号通知软元件	GD54014

5.8.2 个别设置

配方号 30001 配方 1

项 目		设 置
配方文件	-	使用配方文件（执行写入・读取）
	驱动器名	A:标准 SD 卡
	文件夹名	Package1
	文件名	ARP30001.G2P
触发软元件	写入触发软元件 1	GB54016
	读取触发软元件 1	GB54017
	记录号软元件	GD54090
块数		1
记录数		1
块 1	软元件	GD54100
	软元件类型	有符号 BIN16
	点数	2100

配方号 30002 配方 2

項 目		設 定
配方文件	—	使用配方文件（执行写入・读取）
	驱动器名	A:标准 SD 卡
	文件夹名	Package1
	文件名	ARP30002. G2P
触发软元件	写入触发软元件 1	GB54018
	读取触发软元件 1	GB54019
	记录号软元件	GD54091
块数		2
记录数		1
块 1	软元件	GD56200
	软元件类型	有符号 BIN16
	点数	5400
块 2	软元件	GD54087
	软元件类型	有符号 BIN16
	点数	2

## 5.9 脚本一览表

项 目	设 置
工程脚本	有
画面脚本	有： B-30001～30019、 B-30100～30106
对象脚本	有： B-30006～30019、 B-30101、 W-30003
脚本符号	有
对象脚本符号	有

### 5.9.1 工程脚本

脚本号	30020	脚本名	Script30020
注释	初次启动控制		
数据类型	有符号 BIN16	触发类型	上升沿 GB40
<pre>//Actions at Screen Startup [w:GS386] = 1; //Inhibit Initial Start of Scripts set([b:Y41]); //Execute Parameter Batch Read</pre>			
脚本号	30029	脚本名	Script30029
注释	报警、错误信息取得		
数据类型	有符号 BIN16	触发类型	周期 1 秒
<pre>//Observe Error and Alarm if(([w:GD40032] != [w:U03-G8192])    ([w:GD40547] != [w:U03-G9984])) {     set([b:GD31568.b0]); }</pre>			
脚本号	30033	脚本名	Script30033
注释	状态读取		
数据类型	有符号 BIN16	触发类型	下降沿 GD31568.b0
<pre>//Read Statuses set([b:GD31562.b0]);</pre>			
脚本号	30002	脚本名	Script30002
注释	清除软元件数据传送标志		
数据类型	有符号 BIN16	触发类型	ON 中周期 3 秒 GD31562.b0
<pre>//Clear Flag for Device Data Transfer  rst([b:GD31562.b0]);</pre>			
脚本号	30011	脚本名	Script30011
注释	清除软元件数据传送标志		
数据类型	有符号 BIN16	触发类型	ON 中周期 3 秒 GD31564.b0
<pre>//Clear Flag for Device Data Transfer  rst([b:GD31564.b0]);</pre>			
脚本号	30042	脚本名	Script30042
注释	画面启动时的处理		
数据类型	有符号 BIN16	触发类型	上升沿 GB40
<pre>//Initialize iQSS Backup/Restore Sample Screen [w:GS386] = 1; //Inhibit Initial Start of Scripts [w:GD54005] = 1; //Set 1 to the first number of iQSS backup  //Determine Target Device</pre>			

```

[w:GD54075] = 1;          //0: Not Specified, 1: ASLINK, 2: CC-Link
[w:GD54089] = 3;          //To use as a dedicated screen, enter I/O No. here

//When initially displaying iQSS Backup Screen or iQSS Restoration Screen,
//Display Cursor in No.1
set([b:GD54000.b0]);
set([b:GD54002.b0]);

//When starting screen, to read iQSS backup setting and iQSS backup history,
//start script.
set([b:GB54010]);

//Specify action conditions of screen switching switch on the menu screen.
set([b:GB54014]);

```

### 5.9.2 画面脚本

#### 基本画面 30001

脚本号	30035	脚本名	Script30035
注释	软元件数据传送 3 开始		
数据类型	有符号 BIN16	触发类型	ON 中 Y41
<pre> //Transfer parameters to internal devices using Device Data Transfer. if([b:X41] == ON) {     set([b:GD31568.b0]);     rst([b:Y41]); } </pre>			
脚本号	30041	脚本名	Script30041
注释	网络图、传感器一览显示控制		
数据类型	有符号 BIN16	触发类型	ON 中 GB40
<pre> //Control not to display the Network Map and Sensor List at the initial startup. if([b:GB30027] == OFF) {     [w:TMP0000] = 0xFFFF;     fmov([w:TMP0000], [w:GD35000], 128);     fmov([w:TMP0000], [w:GD31617], 48);     set([b:GB30027]); } </pre>			
脚本号	30074	脚本名	Script30074
注释	传感器自动识别		
数据类型	有符号 BIN16	触发类型	ON 中 Y0031
<pre> //Automatically Detect Sensor if([b:X41] == ON) {     //If the parameter batch read flag is on, reset the auto detect flag and start the device     data transfer.     if([b:GB30044] == ON)     {         rst([b:Y31]);         set([b:GD31568.b0]);         rst([b:GB30044]);     } } else{     //Set the flag when starting the parameter batch read. </pre>			

```

        if([b:GB30044] == OFF)
        {
            set([b:GB30044]);
        }
    }
}

```

## 基本画面 30002

脚本号	30023	脚本名	Script30023
注释	B-30002、30005 画面启动时处理		
数据类型	有符号 BIN16	触发类型	ON 中 GB40
<pre> //Read Statuses or Parameters if([b:GB30007] == OFF) {     if(([w:U03-G8192] != 0)    ([w:U03-G9984] != 0))     {         set([b:GD31568.b0]);     }else{         set([b:GD31562.b0]);     }     set([b:GB30007]); } </pre>			
脚本号	30019	脚本名	Script30019
注释	脚本号 30005 启动		
数据类型	有符号 BIN16	触发类型	OFF 中 GD31563.b0
<pre> //Display Objects if([b:GB30016] == OFF) {     set([b:GB30000]);     set([b:GB30016]); } </pre>			
脚本号	30005	脚本名	Script30005
注释	B-30002 部件显示		
数据类型	有符号 BIN16	触发类型	ON 中 GB30000
<pre> //Display Objects [w:GD31526] = 0; [w:TMP0001] = 0; [w:TMP0004] = 0; [w:TMP0001] = [w:GD31525]; [w:TMP0009] = 0;  //Make Settings to Switch Parts Display According To Language if([w:GD62021] &gt; 0) {     [w:TMP0009] = [w:GD62021] - 1; }  while([w:GD31526] &lt; 32) {     //Check Connection Status     if(([w:GD40290] != 0) &amp;&amp; ([w:TMP0001] &lt; [w:GD40290]))     {         [w:GD31600] = [w:GD40291[w:TMP0001]]; //Connection ID     } } </pre>			

Combined

```
[w:TMP0001] = [w:TMP0001] + 1;

if(([w:GD31600] & 0x0200) == 0)
{
    set([b:GB30600]);          //Output Bit
    [w:GD31601] = ([w:GD31600] & 0x0200) >> 9;
} else {
    set([b:GB30601]);          //Input Bit
    [w:GD31601] = ([w:GD31600] & 0x0200) >> 9;
}

//Device Parameter
[w:TMP0004] = [w:GD31600] & 0x00FF;      //Offset to Obtain Device Parameter
if([b:GB30600] == ON)
{
    //Output
    [w:TMP0005] = [w:GD40804[w:TMP0004]];
} else {
    //Input
    [w:TMP0005] = [w:GD41060[w:TMP0004]];
}
[w:GD31604] = [w:TMP0005] - 12288;

//I/O Points Pattern
[w:TMP0006] = [w:GD31604] + 43; //Parameter Offset
[w:TMP0007] = [w:GD41316[w:TMP0006]];
[w:GD31605] = ([w:TMP0007] & 0x00C0) >> 6;      //0: Input, 1: Output, 2:

[w:GD31606] = ([w:TMP0007] & 0x003F) + 1;      //Points

//Module Model Number
[w:TMP0008] = [w:GD31604] + 44; //Parameter Offset
[w:GD31607] = [w:GD41316[w:TMP0008]];

//Alarm Information
if([w:GD40547] != 0)
{
    [w:TMP0004] = 0;
    while([w:TMP0004] < [w:GD40547])
    {
        if([w:GD40548[w:TMP0004]] == [w:GD31600])
        {
            [w:GD31602] = 1;          //Alarm Occurrence
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
    }
}

//Error Information
if([w:GD40032] != 0)
{
    [w:TMP0004] = 0;
```

```

while([w:TMP0004] < [w:GD40032])
{
    if([w:GD40033[w:TMP0004]] == [w:GD31600])
    {
        [w:GD31603] = 1;          //Error Occurrence
        break;
    }
    [w:TMP0004] = [w:TMP0004] + 1;
}

//Display
if([w:GD31603] == 1)
{
    //Error Occurrence
    switch([w:GD31605])
    {
        case 0: [w:GD35000[w:GD31526]] = 30030 + [w:TMP0009];
//Input
                                break;

        case 1: [w:GD35000[w:GD31526]] = 30021 + [w:TMP0009];
//Output
                                break;

        case 2: [w:GD35000[w:GD31526]] = 30039 + [w:TMP0009];
//Combined
                                break;
    }

    [w:GD35032[w:GD31526]] = 0;
    [w:GD35064[w:GD31526]] = [w:GD31600] & 0x00FF; //ID
    [w:GD35096[w:GD31526]] = [w:GD31607]; //Module Model No.
} else {
    if([w:GD31602] == 1)
    {
        //Alarm Occurrence
        switch([w:GD31605])
        {
            case 0: [w:GD35000[w:GD31526]] = 30027 +
[w:TMP0009]; //Input
                                break;

            case 1: [w:GD35000[w:GD31526]] = 30018 +
[w:TMP0009]; //Output
                                break;

            case 2: [w:GD35000[w:GD31526]] = 30036 +
[w:TMP0009]; //Combined
                                break;
        }
    }
}

```



<pre> [w:GD35032[w:GD31526]] = 0; [w:GD35064[w:GD31526]] = [w:GD31600] &amp; 0x00FF; //ID [w:GD35096[w:GD31526]] = [w:GD31607]; //Module Model No.     }else{         //正常         switch([w:GD31605])         {             case 0: [w:GD35000[w:GD31526]] = 30024 + [w:TMP0009]; //Input                                 break;             case 1: [w:GD35000[w:GD31526]] = 30015 + [w:TMP0009]; //Output                                 break;             case 2: [w:GD35000[w:GD31526]] = 30033 + [w:TMP0009]; //Combined                                 break;         }         [w:GD35032[w:GD31526]] = 0;         [w:GD35064[w:GD31526]] = [w:GD31600] &amp; 0x00FF; //ID         [w:GD35096[w:GD31526]] = [w:GD31607]; //Module Model No.     } }  [w:GD31526] = [w:GD31526] + 1; rst([b:GB30600]); //Output Bit rst([b:GB30601]); //Input Bit  //Work Area Clear [w:TMP0004] = 0; fmov([w:TMP0004], [w:GD31600], 5); }else{     //If data does not exist, do not display.     [w:GD35000[w:GD31526]] = 0xFFFF;     [w:GD35032[w:GD31526]] = 0xFFFF;     [w:GD35064[w:GD31526]] = 0xFFFF;     [w:GD35096[w:GD31526]] = 0xFFFF;      [w:GD31526] = [w:GD31526] + 1;     rst([b:GB30600]); //Output Bit     rst([b:GB30601]); //Input Bit } }  //Trigger Reset rst([b:GB30000]); </pre>			
脚本号	30021	脚本名	Script30021
注释	清除各种标志		
数据类型	有符号 BIN16	触发类型	关闭画面时
<pre> //Clear Flags rst([b:GB30006]); rst([b:GB30007]); </pre>			

rst([b:GB30003]);			
[w:TMP0004] = 0; fmov([w:TMP0004],[w:GD31000],512); //Initialize Object Values(B-30003, 4) fmov([w:TMP0004],[w:GD31520],3); //Clear Cursor Position Information(B-30003, 4) fmov([w:TMP0004],[w:GD31600],8); //Work Area Clear [w:GD32008] = 0; //Cursor Position Information Clear(B-30003, 4)  //Close Overlap Window [w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0;			
脚本号	30006	脚本名	Script30006
注释	上滚动		
数据类型	有符号 BIN16	触发类型	上升沿 GB30004
//Scroll List Up if([w:GD31525] != 0) { [w:GD31525] = [w:GD31525] - 32; set([b:GB30000]); } 			
脚本号	30007	脚本名	Script30007
注释	下滚动		
数据类型	有符号 BIN16	触发类型	上升沿 GB30005
//Scroll List Down if((((w:GD31525] + 32) < [w:U03-G8960]) && ([w:U03-G8960] > 32)) { [w:GD31525] = [w:GD31525] + 32; set([b:GB30000]); } 			
脚本号	30010	脚本名	Script30010
注释	B-30002 详细画面显示		
数据类型	无符号 BIN16	触发类型	ON 中 GB30006
//Determine detail screen to display. [w:TMP0000] = [w:GD31523] & 0x00FF;  if((([w:GD31524] >= 30015) && ([w:GD31524] <= 30023)) { //Output [w:GD31567] = [w:GD40804[w:TMP0000]]; //Offset [w:TMP0001] = [w:TMP0000]; }else{ //Input [w:GD31567] = [w:GD41060[w:TMP0000]]; //Offset [w:TMP0001] = [w:TMP0000] + 0x0200; }  [w:U03-G10320] = 0; //Read Individual Parameter [w:U03-G10321] = [w:TMP0001]; //Read Target ID  //Select base screen to display. switch([w:GD35096[w:GD35128]])			

```

{
    case 0x0100:    [w:GD62000] = 30006;
                    break;

    case 0x0101:    [w:GD62000] = 30006;
                    break;

    case 0x0102:    [w:GD62000] = 30008;
                    break;

    case 0x0103:    [w:GD62000] = 30008;
                    break;

    case 0x0104:    [w:GD62000] = 30007;
                    break;

    case 0x0105:    [w:GD62000] = 30007;
                    break;

    case 0x0106:    [w:GD62000] = 30009;
                    break;

    case 0x0107:    [w:GD62000] = 30009;
                    break;

    case 0x0400:    [w:GD62000] = 30006;
                    break;

    case 0x0401:    [w:GD62000] = 30006;
                    break;

    case 0x0402:    [w:GD62000] = 30008;
                    break;

    case 0x0403:    [w:GD62000] = 30008;
                    break;

    case 0x0404:    [w:GD62000] = 30007;
                    break;

    case 0x0405:    [w:GD62000] = 30007;
                    break;

    case 0x0406:    [w:GD62000] = 30009;
                    break;

    case 0x0407:    [w:GD62000] = 30009;
                    break;

    case 0x0700:    [w:GD62000] = 30006;
                    break;

    case 0x0701:    [w:GD62000] = 30006;
                    break;

```

```

case 0x0702:    [w:GD62000] = 30008;
                break;

case 0x0703:    [w:GD62000] = 30008;
                break;

case 0x0704:    [w:GD62000] = 30007;
                break;

case 0x0705:    [w:GD62000] = 30007;
                break;

case 0x0706:    [w:GD62000] = 30009;
                break;

case 0x0707:    [w:GD62000] = 30009;
                break;

case 0x1000:    [w:GD62000] = 30010;
                break;

case 0x1050:    [w:GD62000] = 30010;
                break;

case 0x1200:    [w:GD62000] = 30011;
                break;

case 0x1250:    [w:GD62000] = 30011;
                break;

case 0x1400:    [w:GD62000] = 30012;
                break;

case 0x1450:    [w:GD62000] = 30012;
                break;

case 0x2000:    [w:GD62000] = 30013;
                break;

case 0x2300:    [w:GD62000] = 30014;
                break;

case 0x2301:    [w:GD62000] = 30014;
                break;

case 0x2302:    [w:GD62000] = 30015;
                break;

case 0x2303:    [w:GD62000] = 30016;
                break;

case 0x2600:    [w:GD62000] = 30017;
                break;

```

```

        case 0x2900:    [w:GD62000] = 30018;
                        break;

        case 0xA000:    [w:GD62000] = 30019;
                        break;

        case 0xA001:    [w:GD62000] = 30019;
                        break;

        case 0xA300:    [w:GD62000] = 30019;
                        break;

        case 0xA301:    [w:GD62000] = 30019;
                        break;

        case 0xA600:    [w:GD62000] = 30019;
                        break;

        case 0xA601:    [w:GD62000] = 30019;
                        break;

        default:        break;
    }

    rst([b:GB30006]);

```

### 基本画面 30003

脚本号	30004	脚本名	Script30004
注释	B-30003 画面初始化		
数据类型	有符号 BIN16	触发类型	ON 中 GB40
<pre> if([b:GB30003] == OFF) {      fmov([w:TMP0000], [w:GD31600], 10);     [w:TMP0110] = 0;          //Initialize Temporary Area for Comparison     [w:TMP0111] = 0;          //Initialize Temporary Area for Comparison     set([b:GD31562.b0]);      //Obtain Parameters and Display Parts     set([b:GB30003]);         //Turn on the flag not to work when displaying the screen for the second time or later. } </pre>			
脚本号	30008	脚本名	Script30008
注释	启动脚本号 30003		
数据类型	有符号 BIN16	触发类型	OFF 中 GD31563.b0
<pre> //After reading the status, start Script No.30003. if([b:GB30016] == OFF) {      set([b:GB30001]);     set([b:GB30016]); } </pre>			
脚本号	30003	脚本名	Script30003
注释	B-30003 部件显示		

数据类型	无符号 BIN16	触发类型	ON 中 GB30001
<pre> //Display Parts [w:TMP0020] = 0;  while(([w:TMP0020] &lt; [w:GD40289]) &amp;&amp; ([w:GD40289] &gt; 0)) {     //Obtain Connection ID     [w:GD31600] = [w:GD40291[w:TMP0020]];     [w:TMP0020] = [w:TMP0020] + 1;      if(([w:GD31600] &amp; 0x0200) == 0)     {         set([b:GB30600]);          //Output Bit     }else{         set([b:GB30601]);          //Input Bit     }      //Alarm Information     if([w:GD40547] != 0)     {         [w:TMP0004] = 0;         while([w:TMP0004] &lt; [w:GD40547])         {             if([w:GD40548[w:TMP0004]] == [w:GD31600])             {                 [w:GD31602] = 1;      //Alarm Occurrence                 break;             }             [w:TMP0004] = [w:TMP0004] + 1;         }     }      //Error Information     if([w:GD40032] != 0)     {         [w:TMP0004] = 0;         while([w:TMP0004] &lt; [w:GD40032])         {             if([w:GD40033[w:TMP0004]] == [w:GD31600])             {                 [w:GD31603] = 1;      //Error Occurrence                 break;             }             [w:TMP0004] = [w:TMP0004] + 1;         }     }      //Device Parameter     [w:TMP0004] = [w:GD31600] &amp; 0x00FF;      //Offset to Obtain Device Parameter      if([b:GB30600] == ON)     {         //Output         [w:TMP0005] = [w:GD40804[w:TMP0004]];     } </pre>			

```

        [w:TMP0012] = [w:TMP0004];
    }else{
        //Input
        [w:TMP0005] = [w:GD41060[w:TMP0004]];
        [w:TMP0012] = [w:TMP0004] + 0x0100;
    }
    [w:GD31604] = [w:TMP0005] - 0x3000;

    //I/O Points Pattern
    [w:TMP0006] = [w:GD31604] + 43; //Parameter Offset
    [w:TMP0007] = [w:GD41316[w:TMP0006]];
    [w:GD31605] = ([w:TMP0007] & 0x00C0) >> 6; //0: Input, 1: Output, 2: Combined
    [w:GD31606] = ([w:TMP0007] & 0x003F) + 1; //Points

    //Module Model Number
    [w:TMP0006] = [w:GD31604] + 44;
    [w:GD35130[w:TMP0012]] = [w:GD41316[w:TMP0006]];

    //ON/OFF Information
    [w:TMP0008] = [w:GD31600] & 0x00FF;
    if([w:GD31605] == 0) //Input Case
    {
        [w:TMP0004] = 0;

        while([w:TMP0004] < [w:GD31606])
        {
            [w:TMP0009] = [w:TMP0008] / 16;
            [w:TMP0010] = [w:TMP0008] % 16;

            [w:GD33020] = 0x0001;
            [w:GD33020] = [w:GD33020] << [w:TMP0010];

            if([w:GD31603] == 1) //Check If Error Occurred
            {
                [w:GD31256[w:TMP0008]] = 3; //Error Occurrence & Input
            }else{
                if([w:GD31602] == 1) //Check If Alarms Occurred
                {
                    [w:GD31256[w:TMP0008]] = 2; //Alarm Occurrence
                }else{
                    [w:GD31256[w:TMP0008]] = 1; //Normal
                }
            }

            [w:TMP0008] = [w:TMP0008] + 1;
            [w:TMP0004] = [w:TMP0004] + 1;
        }
    }else{
        if([w:GD31605] == 1) //Output Case
        {
            [w:TMP0004] = 0;

```

```

while([w:TMP0004] < [w:GD31606])
{
    [w:TMP0009] = [w:TMP0008] / 16;
    [w:TMP0010] = [w:TMP0008] % 16;

    [w:GD33020] = 0x0001;
    [w:GD33020] = [w:GD33020] << [w:TMP0010];

    if([w:GD31603] == 1)    //Check If Error Occurred
    {
        [w:GD31000[w:TMP0008]] = 3;    //Error Occurrence
    }
    else{
        if([w:GD31602] == 1)    //Check If Alarms Occurred
        {
            [w:GD31000[w:TMP0008]] = 2;    //Alarm Occurrence & Output ON
        }
        else{
            [w:GD31000[w:TMP0008]] = 1;    //Normal Occurrence & Output ON
        }
    }

    [w:TMP0008] = [w:TMP0008] + 1;
    [w:TMP0004] = [w:TMP0004] + 1;
}
else{
    if([w:GD31605] == 2)    //Combined Case
    {
        [w:TMP0004] = 0;
        [w:TMP0011] = [w:GD31606] / 2;    //If combined, points are
        reduced by half.

        while([w:TMP0004] < [w:TMP0011])
        {
            [w:TMP0009] = [w:TMP0008] / 16;
            [w:TMP0010] = [w:TMP0008] % 16;

            [w:GD33020] = 0x0001;
            [w:GD33020] = [w:GD33020] << [w:TMP0010];

            //Output
            if([w:GD31603] == 1)    //Check If Error Occurred
            {
                [w:GD31000[w:TMP0008]] = 3;    //Error Occurrence & Output ON
            }
            else{
                if([w:GD31602] == 1)    //Check If Alarms Occurred
                {
                    [w:GD31000[w:TMP0008]] = 2;    //Alarm Occurrence & Output ON
                }
                else{
                    [w:GD31000[w:TMP0008]] = 1;    //Normal Occurred
                }
            }
        }
    }
}

```



```

    }

    //Input
    if([w:GD31603] == 1)    //Check If Error Occurred
    {
        [w:GD31256[w:TMP0008]] = 3;    //Error
    }

    }else{
        if([w:GD31602] == 1)    //Check If Alarms
        {
            [w:GD31256[w:TMP0008]] = 2;
        }

        }else{
            [w:GD31256[w:TMP0008]] = 1;
        }
    }

    [w:TMP0008] = [w:TMP0008] + 1;
    [w:TMP0004] = [w:TMP0004] + 1;
}

}

}

rst([b:GB30600]);
rst([b:GB30601]);

//Work Area Clear
[w:TMP0004] = 0;
fmov([w:TMP0004], [w:GD31600], 5);
}

//Display Cursor
if([w:GD62000] == 30004)
{
    //Output
    [w:GD31520] = 0;    //Left Right
    [w:GD31521] = 0;    //Up Down
    [w:GD31522] = [w:GD31520] + [w:GD31521];
    [w:GD32008] = [w:GD31522];    //Cursor Position Display

    if([w:GD31000[w:GD31522]] < 4)
    {
        [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;
    }
}

}else{
    //Input
    [w:GD31520] = 256;    //Left Right
    [w:GD31521] = 0;    //Up Down
    [w:GD31522] = [w:GD31520] + [w:GD31521];
    [w:GD32008] = [w:GD31522] - 256;    //Cursor Position Display
}

```

<pre> if([w:GD31000[w:GD31522]] &lt; 4) {     [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4; } } </pre>			
脚本号	30021	脚本名	Script30021
注释	清除各种标志		
数据类型	有符号 BIN16	触发类型	关闭画面时
<pre> //Clear Flags rst([b:GB30006]); rst([b:GB30007]); rst([b:GB30003]);  [w:TMP0004] = 0; fmov([w:TMP0004], [w:GD31000], 512);    //Initialize Object Values(B-30003, 4) fmov([w:TMP0004], [w:GD31520], 3);      //Clear          Cursor          Position Information(B-30003, 4) fmov([w:TMP0004], [w:GD31600], 8);      //Work Area Clear [w:GD32008] = 0;          //Cursor Position Information Clear(B-30003, 4)  //Close Overlap Window [w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0; </pre>			
脚本号	30017	脚本名	Script30017
注释	B-30003 详细画面显示		
数据类型	无符号 BIN16	触发类型	ON 中 GB30002
<pre> //Display Detail Information Screen of Cursor Position  //Obtain Temporary ID if([w:GD31522] &lt; 256) {     [w:TMP0000] = [w:GD31522];    //Output ID }else{     [w:TMP0000] = [w:GD31522] - 256;    //Input ID }  if([w:GD31000[w:GD31522]] &gt; 4) //Judge whether a module exists in the cursor position. {     if([w:GD31522] &lt; 256)     {         //Output         [w:TMP0002] = [w:TMP0000];         while([w:TMP0002] &gt;= 0)         {             if([w:GD31000[w:TMP0002]] != 0)             {                 //Output                 [w:TMP0001] = 0;                 while([w:TMP0001] &lt; [w:GD40290])    //Repeat the Number of Connection ID Counts </pre>			

```

{
    if([w:GD40291[w:TMP0001]] == [w:TMP0002])
        //Judge ID
        {
            set([b:GB30009]); //ID Match Flag
            break;
        }else{
            [w:TMP0001] = [w:TMP0001] + 1;
        }
    }

    //Combined Output Side Case
    if([b:GB30009] == OFF)
    {
        [w:TMP0001] = 0;
        while([w:TMP0001] < [w:GD40290]) //Repeat
the Number of Connection ID Counts
        {
            if([w:GD40291[w:TMP0001]] == ([w:TMP0002] +
0x0200)) //Judge ID
            {
                //Judge whether the matched ID is
that of a combined module or not.
                [w:TMP0010] =
[w:GD41060[w:TMP0002]]; //Parameter Top Address
                [w:TMP0011] = [w:TMP0010] - 0x3000;
                [w:TMP0012] =
[w:GD41359[w:TMP0011]];
                if((((w:TMP0012] & 0x00C0) >> 6) ==
2)
                {
                    set([b:GB30009]);
                    set([b:GB30020]);
                    break;
                }
                [w:TMP0001] = [w:TMP0001] + 1;
            }else{
                [w:TMP0001] = [w:TMP0001] + 1;
            }
        }

        if([b:GB30009] == ON) //If matched, exit loop.
        {
            break;
        }
    }else{
        //If No Connected Devices
        break;
    }
}

```

```

        [w:TMP0002] = [w:TMP0002] - 1;
    }

} else {
    //Input
    [w:TMP0002] = [w:TMP0000];
    while([w:TMP0002] >= 0)
    {
        [w:GD65535] = [w:TMP0002];
        [w:TMP0001] = 0;
        while([w:TMP0001] < [w:GD40290])//Repeat the Number of Connection ID
Counts
        {
            if([w:GD40291[w:TMP0001]] == ([w:TMP0002] + 0x0200))
//Judge ID
            {
                set([b:GB30009]); //ID Match Flag
                break;
            } else {
                [w:TMP0001] = [w:TMP0001] + 1;
            }
        }

        if([b:GB30009] == 0N) //If matched, exit loop.
        {
            break;
        }

        [w:TMP0002] = [w:TMP0002] - 1;
    }
}

//If matched, display screen.
if([b:GB30009] == 0N)
{
    if([w:GD31522] < 256)
    {
        if([b:GB30020] == OFF)
        {
            //Output
            [w:GD31567] = [w:GD40804[w:TMP0002]]; //Offset
            [w:TMP0003] = [w:TMP0002];
            [w:TMP0004] = [w:TMP0002];
        } else {
            //If combined, refer to the input side.
            [w:GD31567] = [w:GD41060[w:TMP0002]]; //Offset
            [w:TMP0003] = [w:TMP0002] + 0x0200;
            [w:TMP0004] = [w:TMP0002] + 0x0100;
        }
    }
} else {

```

```

//Input
[w:GD31567] = [w:GD41060[w:TMP0002]]; //Offset
[w:TMP0003] = [w:TMP0002] + 0x0200;
[w:TMP0004] = [w:TMP0002] + 0x0100;
}

[w:U03-G10320] = 0; //Read Individual Parameter
[w:U03-G10321] = [w:TMP0003]; //Read Target ID

//Select base screen to display.
switch([w:GD35130[w:TMP0004]])
{
    case 0x0100: [w:GD62000] = 30006;
                  break;

    case 0x0101: [w:GD62000] = 30006;
                  break;

    case 0x0102: [w:GD62000] = 30008;
                  break;

    case 0x0103: [w:GD62000] = 30008;
                  break;

    case 0x0104: [w:GD62000] = 30007;
                  break;

    case 0x0105: [w:GD62000] = 30007;
                  break;

    case 0x0106: [w:GD62000] = 30009;
                  break;

    case 0x0107: [w:GD62000] = 30009;
                  break;

    case 0x0400: [w:GD62000] = 30006;
                  break;

    case 0x0401: [w:GD62000] = 30006;
                  break;

    case 0x0402: [w:GD62000] = 30008;
                  break;

    case 0x0403: [w:GD62000] = 30008;
                  break;

    case 0x0404: [w:GD62000] = 30007;
                  break;

    case 0x0405: [w:GD62000] = 30007;
                  break;
}

```

```

case 0x0406:    [w:GD62000] = 30009;
                break;

case 0x0407:    [w:GD62000] = 30009;
                break;

case 0x0700:    [w:GD62000] = 30006;
                break;

case 0x0701:    [w:GD62000] = 30006;
                break;

case 0x0702:    [w:GD62000] = 30008;
                break;

case 0x0703:    [w:GD62000] = 30008;
                break;

case 0x0704:    [w:GD62000] = 30007;
                break;

case 0x0705:    [w:GD62000] = 30007;
                break;

case 0x0706:    [w:GD62000] = 30009;
                break;

case 0x0707:    [w:GD62000] = 30009;
                break;

case 0x1000:    [w:GD62000] = 30010;
                break;

case 0x1050:    [w:GD62000] = 30010;
                break;

case 0x1200:    [w:GD62000] = 30011;
                break;

case 0x1250:    [w:GD62000] = 30011;
                break;

case 0x1400:    [w:GD62000] = 30012;
                break;

case 0x1450:    [w:GD62000] = 30012;
                break;

case 0x2000:    [w:GD62000] = 30013;
                break;

case 0x2300:    [w:GD62000] = 30014;
                break;

```

<pre>                                 case 0x2301:    [w:GD62000] = 30014;   break;                                  case 0x2302:    [w:GD62000] = 30015;   break;                                  case 0x2303:    [w:GD62000] = 30016;   break;                                  case 0x2600:    [w:GD62000] = 30017;   break;                                  case 0x2900:    [w:GD62000] = 30018;   break;                                  case 0xA000:    [w:GD62000] = 30019;   break;                                  case 0xA001:    [w:GD62000] = 30019;   break;                                  case 0xA300:    [w:GD62000] = 30019;   break;                                  case 0xA301:    [w:GD62000] = 30019;   break;                                  case 0xA600:    [w:GD62000] = 30019;   break;                                  case 0xA601:    [w:GD62000] = 30019;   break;                                  default:        break;                                 }                                 }                                  //Flag Reset                                 rst([b:GB30009]);                                 rst([b:GB30020]);                                 }                                  rst([b:GB30002]); </pre>			
脚本号	30012	脚本名	Script30012
注释	坐标计算(输入) *1		
数据类型	有符号 BIN16	触发类型	通常
<pre> //Determine the cursor position from the touched coordinate.  if(((([w:TMP0120] != [w:GS654])    ([w:TMP0121] != [w:GS655])) &amp;&amp; (([w:GS654] &gt; Input_X) &amp;&amp; ([w:GS655] &gt; Input_Y))     &amp;&amp; (([w:GS654] &lt; Frame_X) &amp;&amp; ([w:GS655] &lt; Frame_Y))) {     [w:TMP0120] = [w:GS654];     [w:TMP0121] = [w:GS655]; </pre>			

```

if(([b:GD31562.b0] == OFF) && ([b:GB30001] == OFF))
{
//Save Current Cursor Position
[w:TMP0122] = [w:GD31522];

//X Coordinate
//Change values to subtract depending on whether input or output.
//Input
[w:TMP0123] = [w:TMP0120] - Input_X;
[w:TMP0124] = [w:TMP0123] / Object_X;
[w:GD31520] = (16 - ([w:TMP0124] + 1)) + 256;

//Y Coordinate
[w:TMP0125] = [w:TMP0121] - Input_Y;
[w:TMP0126] = [w:TMP0125] / Object_Y;
[w:GD31521] = [w:TMP0126] * 16;

//Offset
[w:GD31522] = [w:GD31520] + [w:GD31521];

//If the offset position does not change, displayed parts do not change.
if([w:TMP0122] != [w:GD31522])
{
//Cursor Display
if([w:GD31000[w:GD31522]] < 4)
{
[w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;
}

//Cursor Delete
if([w:GD31000[w:TMP0122]] < 4)
{
[w:GD31000[w:TMP0122]] = 0;
}else{
[w:GD31000[w:TMP0122]] = [w:GD31000[w:TMP0122]] - 4;
}
}
}

[w:GD32008] = [w:GD31522] - 256; //Cursor Position Display
}

```

脚本号	30013	脚本名	Script30013
注释	光标显示(向左移动)(输入)		
数据类型	有符号 BIN16	触发类型	上升沿 GB30010
<pre> //Control Parts Display  //Calculate Cursor Position [w:TMP0020] = [w:GD31522]; //Evacuate Previous Cursor Position  if([w:GD31520] == 271) { [w:GD31520] = 256; </pre>			



<pre> }else{     [w:GD31520] = [w:GD31520] + 1; }  [w:TMP0000] = [w:GD31520];  [w:GD31522] = [w:TMP0000] + [w:GD31521];          //Offset  if([w:GD31000[w:TMP0020]] &gt;= 4) {     [w:GD31000[w:TMP0020]] = [w:GD31000[w:TMP0020]] - 4; }else{     [w:GD31000[w:TMP0020]] = 0; } [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;  [w:GD32008] = [w:GD31522] - 256;          //Cursor Position Display </pre>			
脚本号	30018	脚本名	Script30018
注释	光标显示(向右移动)(输入)		
数据类型	有符号 BIN16	触发类型	上升沿 GB30011
<pre> //Control Parts Display  //Calculate Cursor Position [w:TMP0020] = [w:GD31522];          //Evacuate Previous Cursor Position  if([w:GD31520] == 256) {     [w:GD31520] = 271; }else{     [w:GD31520] = [w:GD31520] -1; }  [w:TMP0000] = [w:GD31520];  [w:GD31522] = [w:TMP0000] + [w:GD31521];          //Offset  if([w:GD31000[w:TMP0020]] &gt;= 4) {     [w:GD31000[w:TMP0020]] = [w:GD31000[w:TMP0020]] - 4; }else{     [w:GD31000[w:TMP0020]] = 0; } [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;  [w:GD32008] = [w:GD31522] - 256;          //Cursor Position Display </pre>			
脚本号	30022	脚本名	Script30022
注释	光标显示(向下移动)(输入)		
数据类型	有符号 BIN16	触发类型	上升沿 GB30012
<pre> //Control Parts Display  //Calculate Cursor Position [w:TMP0020] = [w:GD31522];          //Evacuate Previous Cursor Position </pre>			

<pre> if([w:GD31521] == 240) {     [w:GD31521] = 0; }else{     [w:GD31521] = [w:GD31521] + 16; }  [w:GD31522] = [w:GD31520] + [w:GD31521];          //Offset  if([w:GD31000[w:TMP0020]] &gt;= 4) {     [w:GD31000[w:TMP0020]] = [w:GD31000[w:TMP0020]] - 4; }else{     [w:GD31000[w:TMP0020]] = 0; } [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;  [w:GD32008] = [w:GD31522] - 256;          //Cursor Position Display </pre>			
脚本号	30036	脚本名	Script30036
注释	光标显示(向上移动)(输入)		
数据类型	有符号 BIN16	触发类型	上升沿 GB30013
<pre> //Control Parts Display  //Calculate Cursor Position [w:TMP0020] = [w:GD31522];          //Evacuate Previous Cursor Position  if([w:GD31521] == 0) {     [w:GD31521] = 240; }else{     [w:GD31521] = [w:GD31521] - 16; }  [w:GD31522] = [w:GD31520] + [w:GD31521];          //Offset  if([w:GD31000[w:TMP0020]] &gt;= 4) {     [w:GD31000[w:TMP0020]] = [w:GD31000[w:TMP0020]] - 4; }else{     [w:GD31000[w:TMP0020]] = 0; } [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;  [w:GD32008] = [w:GD31522] - 256;          //Cursor Position Display </pre>			

\*1: 使用了[脚本符号]。关于[脚本符号]，请参照「5.9.4 脚本符号」。

## 基本画面 30004

脚本号	30004	脚本名	Script30004
注释	B-30003 画面初始化		
数据类型	有符号 BIN16	触发类型	ON 中 GB40
<pre> if([b:GB30003] == OFF) {     fmov([w:TMP0000], [w:GD31600], 10);     [w:TMP0110] = 0;          //Initialize Temporary Area for Comparison     [w:TMP0111] = 0;          //Initialize Temporary Area for Comparison     set([b:GD31562.b0]);      //Obtain Parameters and Display Parts     set([b:GB30003]);         //Turn on the flag not to work when displaying the screen for the second time or later. } </pre>			
脚本号	30008	脚本名	Script30008
注释	启动脚本号 30003		
数据类型	有符号 BIN16	触发类型	OFF 中 GD31563.b0
<pre> //After reading the status, start Script No.30003. if([b:GB30016] == OFF) {     set([b:GB30001]);     set([b:GB30016]); } </pre>			
脚本号	30003	脚本名	Script30003
注释	B-30003 部件显示		
数据类型	无符号 BIN16	触发类型	ON 中 GB30001
<pre> //Display Parts [w:TMP0020] = 0;  while(([w:TMP0020] &lt; [w:GD40289]) &amp;&amp; ([w:GD40289] &gt; 0)) {     //Obtain Connection ID     [w:GD31600] = [w:GD40291[w:TMP0020]];     [w:TMP0020] = [w:TMP0020] + 1;      if(([w:GD31600] &amp; 0x0200) == 0)     {         set([b:GB30600]);          //Output Bit     }else{         set([b:GB30601]);          //Input Bit     }      //Alarm Information     if([w:GD40547] != 0)     {         [w:TMP0004] = 0;         while([w:TMP0004] &lt; [w:GD40547])         {             if([w:GD40548[w:TMP0004]] == [w:GD31600])             {                 [w:GD31602] = 1;          //Alarm Occurrence                 break;             }         }     } } </pre>			

```

    }
    [w:TMP0004] = [w:TMP0004] + 1;
}

//Error Information
if([w:GD40032] != 0)
{
    [w:TMP0004] = 0;
    while([w:TMP0004] < [w:GD40032])
    {
        if([w:GD40033[w:TMP0004]] == [w:GD31600])
        {
            [w:GD31603] = 1;          //Error Occurrence
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
    }
}

//Device Parameter
[w:TMP0004] = [w:GD31600] & 0x00FF;    //Offset to Obtain Device Parameter

if([b:GB30600] == ON)
{
    //Output
    [w:TMP0005] = [w:GD40804[w:TMP0004]];
    [w:TMP0012] = [w:TMP0004];
} else {
    //Input
    [w:TMP0005] = [w:GD41060[w:TMP0004]];
    [w:TMP0012] = [w:TMP0004] + 0x0100;
}

[w:GD31604] = [w:TMP0005] - 0x3000;

//I/O Points Pattern
[w:TMP0006] = [w:GD31604] + 43; //Parameter Offset
[w:TMP0007] = [w:GD41316[w:TMP0006]];
[w:GD31605] = ([w:TMP0007] & 0x00C0) >> 6;    //0: Input, 1: Output, 2: Combined
[w:GD31606] = ([w:TMP0007] & 0x003F) + 1;    //Points

//Module Model Number
[w:TMP0006] = [w:GD31604] + 44;
[w:GD35130[w:TMP0012]] = [w:GD41316[w:TMP0006]];

//ON/OFF Information
[w:TMP0008] = [w:GD31600] & 0x00FF;
if([w:GD31605] == 0)    //Input Case
{
    [w:TMP0004] = 0;

    while([w:TMP0004] < [w:GD31606])
    {
        [w:TMP0009] = [w:TMP0008] / 16;
    }
}

```

```

[w:TMP0010] = [w:TMP0008] % 16;

[w:GD33020] = 0x0001;
[w:GD33020] = [w:GD33020] << [w:TMP0010];

if([w:GD31603] == 1)    //Check If Error Occurred
{
    [w:GD31256[w:TMP0008]] = 3;    //Error Occurrence & Input
ON
}
else{
    if([w:GD31602] == 1)    //Check If Alarms Occurred
    {
        [w:GD31256[w:TMP0008]] = 2;    //Alarm Occurrence
& Input ON
    }
    else{
        [w:GD31256[w:TMP0008]] = 1;    //Normal
    }
}

[w:TMP0008] = [w:TMP0008] + 1;
[w:TMP0004] = [w:TMP0004] + 1;
}
}
else{
    if([w:GD31605] == 1)    //Output Case
    {
        [w:TMP0004] = 0;

        while([w:TMP0004] < [w:GD31606])
        {
            [w:TMP0009] = [w:TMP0008] / 16;
            [w:TMP0010] = [w:TMP0008] % 16;

            [w:GD33020] = 0x0001;
            [w:GD33020] = [w:GD33020] << [w:TMP0010];

            if([w:GD31603] == 1)    //Check If Error Occurred
            {
                [w:GD31000[w:TMP0008]] = 3;    //Error Occurrence
& Output ON
            }
            else{
                if([w:GD31602] == 1)    //Check If Alarms Occurred
                {
                    [w:GD31000[w:TMP0008]] = 2;    //Alarm
Occurrence & Output ON
                }
                else{
                    [w:GD31000[w:TMP0008]] = 1;    //Normal
                }
            }

            [w:TMP0008] = [w:TMP0008] + 1;
            [w:TMP0004] = [w:TMP0004] + 1;
        }
    }
}
else{
    if([w:GD31605] == 2)    //Combined Case

```

```

{
    [w:TMP0004] = 0;
    [w:TMP0011] = [w:GD31606] / 2; //If combined, points are
reduced by half.
    while([w:TMP0004] < [w:TMP0011])
    {
        [w:TMP0009] = [w:TMP0008] / 16;
        [w:TMP0010] = [w:TMP0008] % 16;

        [w:GD33020] = 0x0001;
        [w:GD33020] = [w:GD33020] << [w:TMP0010];

        //Output
        if([w:GD31603] == 1) //Check If Error Occurred
        {
            [w:GD31000[w:TMP0008]] = 3; //Error
Occurrence & Output ON
        }else{
            if([w:GD31602] == 1) //Check If Alarms
Occurred
            {
                [w:GD31000[w:TMP0008]] = 2;
                //Alarm Occurrence & Output ON
            }else{
                [w:GD31000[w:TMP0008]] = 1;
                //Normal
            }
        }

        //Input
        if([w:GD31603] == 1) //Check If Error Occurred
        {
            [w:GD31256[w:TMP0008]] = 3; //Error
Occurrence & Input ON
        }else{
            if([w:GD31602] == 1) //Check If Alarms
Occurred
            {
                [w:GD31256[w:TMP0008]] = 2;
                //Alarm Occurrence & Input ON
            }else{
                [w:GD31256[w:TMP0008]] = 1;
                //Normal
            }
        }

        [w:TMP0008] = [w:TMP0008] + 1;
        [w:TMP0004] = [w:TMP0004] + 1;
    }
}
}
rst([b:GB30600]);
rst([b:GB30601]);

```

```

//Work Area Clear
[w:TMP0004] = 0;
fmov([w:TMP0004], [w:GD31600], 5);
}

//Display Cursor
if([w:GD62000] == 30004)
{
    //Output
    [w:GD31520] = 0;          //Left Right
    [w:GD31521] = 0;          //Up Down
    [w:GD31522] = [w:GD31520] + [w:GD31521];
    [w:GD32008] = [w:GD31522];    //Cursor Position Display

    if([w:GD31000[w:GD31522]] < 4)
    {
        [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;
    }
} else {
    //Input
    [w:GD31520] = 256;        //Left Right
    [w:GD31521] = 0;          //Up Down
    [w:GD31522] = [w:GD31520] + [w:GD31521];
    [w:GD32008] = [w:GD31522] - 256;    //Cursor Position Display

    if([w:GD31000[w:GD31522]] < 4)
    {
        [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;
    }
}

rst([b:GB30001]);

```

脚本号	30001	脚本名	Script30001
注释	光标显示(向左移动)		
数据类型	有符号 BIN16	触发类型	上升沿 GB30010

```

//Control Parts Display

//Calculate Cursor Position
[w:TMP0020] = [w:GD31522];    //Previous Cursor Position

if([w:GD31520] == 15)
{
    [w:GD31520] = 0;
} else {
    [w:GD31520] = [w:GD31520] + 1;
}

[w:TMP0000] = [w:GD31520];

[w:GD31522] = [w:TMP0000] + [w:GD31521];    //Offset

if([w:GD31000[w:TMP0020]] >= 4)

```

<pre> {     [w:GD31000[w:TMP0020]] = [w:GD31000[w:TMP0020]] - 4; }else{     [w:GD31000[w:TMP0020]] = 0; } [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;  [w:GD32008] = [w:GD31522];      //Cursor Position Display </pre>			
脚本号	30030	脚本名	Script30030
注释	光标显示(向右移动)		
数据类型	有符号 BIN16	触发类型	上升沿 GB30011
<pre> //Control Parts Display  //Calculate Cursor Position [w:TMP0020] = [w:GD31522];      //Evacuate Previous Cursor Position  if([w:GD31520] == 0) {     [w:GD31520] = 15; }else{     [w:GD31520] = [w:GD31520] -1; }  [w:TMP0000] = [w:GD31520];  [w:GD31522] = [w:TMP0000] + [w:GD31521];      //Offset  if([w:GD31000[w:TMP0020]] &gt;= 4) {     [w:GD31000[w:TMP0020]] = [w:GD31000[w:TMP0020]] - 4; }else{     [w:GD31000[w:TMP0020]] = 0; } [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4; </pre>			
脚本号	30031	脚本名	Script30031
注释	光标显示(向下移动)		
数据类型	有符号 BIN16	触发类型	上升沿 GB30012
<pre> //Control Parts Display  //Calculate Cursor Position [w:TMP0020] = [w:GD31522];      //Evacuate Previous Cursor Position  if([w:GD31521] == 240) {     [w:GD31521] = 0; }else{     [w:GD31521] = [w:GD31521] + 16; }  [w:GD31522] = [w:GD31520] + [w:GD31521];      //Offset  if([w:GD31000[w:TMP0020]] &gt;= 4) { </pre>			



<pre> [w:GD31000[w:TMP0020]] = [w:GD31000[w:TMP0020]] - 4; }else{ [w:GD31000[w:TMP0020]] = 0; } [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;  [w:GD32008] = [w:GD31522];      //Cursor Position Display </pre>			
脚本号	30032	脚本名	Script30032
注释	光标显示(向上移动)		
数据类型	有符号 BIN16	触发类型	上升沿 GB30013
<pre> //Control Parts Display  //Calculate Cursor Position [w:TMP0020] = [w:GD31522];      //Evacuate Previous Cursor Position  if([w:GD31521] == 0) { [w:GD31521] = 240; }else{ [w:GD31521] = [w:GD31521] - 16; }  [w:GD31522] = [w:GD31520] + [w:GD31521];      //Offset  if([w:GD31000[w:TMP0020]] &gt;= 4) { [w:GD31000[w:TMP0020]] = [w:GD31000[w:TMP0020]] - 4; }else{ [w:GD31000[w:TMP0020]] = 0; } [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4; </pre>			
脚本号	30021	脚本名	Script30021
注释	清除各种标志		
数据类型	有符号 BIN16	触发类型	关闭画面时
<pre> //Clear Flags rst([b:GB30006]); rst([b:GB30007]); rst([b:GB30003]);  [w:TMP0004] = 0; fmov([w:TMP0004], [w:GD31000], 512);      //Initialize Object Values(B-30003, 4) fmov([w:TMP0004], [w:GD31520], 3);      //Clear Cursor Position Information(B-30003, 4) fmov([w:TMP0004], [w:GD31600], 8);      //Work Area Clear [w:GD32008] = 0;      //Cursor Position Information Clear(B-30003, 4)  //Close Overlap Window [w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0; </pre>			
脚本号	30017	脚本名	Script30017
注释	B-30003 详细画面显示		
数据类型	无符号 BIN16	触发类型	ON 中 GB30002

```

//Display Detail Information Screen of Cursor Position

//Obtain Temporary ID
if([w:GD31522] < 256)
{
    [w:TMP0000] = [w:GD31522];      //Output ID
}else{
    [w:TMP0000] = [w:GD31522] - 256;      //Input ID
}

if([w:GD31000[w:GD31522]] > 4) //Judge whether a module exists in the cursor position.
{
    if([w:GD31522] < 256)
    {
        //Output
        [w:TMP0002] = [w:TMP0000];
        while([w:TMP0002] >= 0)
        {
            if([w:GD31000[w:TMP0002]] != 0)
            {
                //Output
                [w:TMP0001] = 0;
                while([w:TMP0001] < [w:GD40290])      //Repeat the Number
of Connection ID Counts
                {
                    if([w:GD40291[w:TMP0001]] == [w:TMP0002])
                    {
                        set([b:GB30009]);      //ID Match Flag
                        break;
                    }else{
                        [w:TMP0001] = [w:TMP0001] + 1;
                    }
                }

                //Combined Output Side Case
                if([b:GB30009] == OFF)
                {
                    [w:TMP0001] = 0;
                    while([w:TMP0001] < [w:GD40290])      //Repeat
the Number of Connection ID Counts
                    {
                        if([w:GD40291[w:TMP0001]] == ([w:TMP0002] +
0x0200))      //Judge ID
                        {
                            //Judge whether the matched ID is
that of a combined module or not.
                            [w:TMP0010] =
[w:GD41060[w:TMP0002]]; //Parameter Top Address
                            [w:TMP0011] = [w:TMP0010] - 0x3000;
                            [w:TMP0012] =
[w:GD41359[w:TMP0011]];

```

```

2) if(((w:TMP0012] & 0x00C0) >> 6) ==
    {
        set([b:GB30009]);
        set([b:GB30020]);
        break;
    }
    [w:TMP0001] = [w:TMP0001] + 1;
} else {
    [w:TMP0001] = [w:TMP0001] + 1;
}
}

if([b:GB30009] == 0N) //If matched, exit loop.
{
    break;
}
} else {
    //If No Connected Devices
    break;
}

[w:TMP0002] = [w:TMP0002] - 1;
}

} else {
    //Input
    [w:TMP0002] = [w:TMP0000];
    while([w:TMP0002] >= 0)
    {
        [w:GD65535] = [w:TMP0002];
        [w:TMP0001] = 0;
        while([w:TMP0001] < [w:GD40290]) //Repeat the Number of Connection ID
        {
            if([w:GD40291[w:TMP0001]] == ([w:TMP0002] + 0x0200))
            {
                set([b:GB30009]); //ID Match Flag
                break;
            } else {
                [w:TMP0001] = [w:TMP0001] + 1;
            }
        }

        if([b:GB30009] == 0N) //If matched, exit loop.
        {
            break;
        }
    }
}

```

Counts

```

        [w:TMP0002] = [w:TMP0002] - 1;
    }
}

//If matched, display screen.
if([b:GB30009] == ON)
{
    if([w:GD31522] < 256)
    {
        if([b:GB30020] == OFF)
        {
            //Output
            [w:GD31567] = [w:GD40804[w:TMP0002]]; //Offset
            [w:TMP0003] = [w:TMP0002];
            [w:TMP0004] = [w:TMP0002];
        }else{
            //If combined, refer to the input side.
            [w:GD31567] = [w:GD41060[w:TMP0002]]; //Offset
            [w:TMP0003] = [w:TMP0002] + 0x0200;
            [w:TMP0004] = [w:TMP0002] + 0x0100;
        }
    }else{
        //Input
        [w:GD31567] = [w:GD41060[w:TMP0002]]; //Offset
        [w:TMP0003] = [w:TMP0002] + 0x0200;
        [w:TMP0004] = [w:TMP0002] + 0x0100;
    }

    [w:U03-G10320] = 0; //Read Individual Parameter
    [w:U03-G10321] = [w:TMP0003]; //Read Target ID

    //Select base screen to display.
    switch([w:GD35130[w:TMP0004]])
    {
        case 0x0100: [w:GD62000] = 30006;
                     break;

        case 0x0101: [w:GD62000] = 30006;
                     break;

        case 0x0102: [w:GD62000] = 30008;
                     break;

        case 0x0103: [w:GD62000] = 30008;
                     break;

        case 0x0104: [w:GD62000] = 30007;
                     break;

        case 0x0105: [w:GD62000] = 30007;
                     break;
    }
}

```

```

case 0x0106:    [w:GD62000] = 30009;
                break;

case 0x0107:    [w:GD62000] = 30009;
                break;

case 0x0400:    [w:GD62000] = 30006;
                break;

case 0x0401:    [w:GD62000] = 30006;
                break;

case 0x0402:    [w:GD62000] = 30008;
                break;

case 0x0403:    [w:GD62000] = 30008;
                break;

case 0x0404:    [w:GD62000] = 30007;
                break;

case 0x0405:    [w:GD62000] = 30007;
                break;

case 0x0406:    [w:GD62000] = 30009;
                break;

case 0x0407:    [w:GD62000] = 30009;
                break;

case 0x0700:    [w:GD62000] = 30006;
                break;

case 0x0701:    [w:GD62000] = 30006;
                break;

case 0x0702:    [w:GD62000] = 30008;
                break;

case 0x0703:    [w:GD62000] = 30008;
                break;

case 0x0704:    [w:GD62000] = 30007;
                break;

case 0x0705:    [w:GD62000] = 30007;
                break;

case 0x0706:    [w:GD62000] = 30009;
                break;

case 0x0707:    [w:GD62000] = 30009;
                break;

```

```

case 0x1000:    [w:GD62000] = 30010;
                break;

case 0x1050:    [w:GD62000] = 30010;
                break;

case 0x1200:    [w:GD62000] = 30011;
                break;

case 0x1250:    [w:GD62000] = 30011;
                break;

case 0x1400:    [w:GD62000] = 30012;
                break;

case 0x1450:    [w:GD62000] = 30012;
                break;

case 0x2000:    [w:GD62000] = 30013;
                break;

case 0x2300:    [w:GD62000] = 30014;
                break;

case 0x2301:    [w:GD62000] = 30014;
                break;

case 0x2302:    [w:GD62000] = 30015;
                break;

case 0x2303:    [w:GD62000] = 30016;
                break;

case 0x2600:    [w:GD62000] = 30017;
                break;

case 0x2900:    [w:GD62000] = 30018;
                break;

case 0xA000:    [w:GD62000] = 30019;
                break;

case 0xA001:    [w:GD62000] = 30019;
                break;

case 0xA300:    [w:GD62000] = 30019;
                break;

case 0xA301:    [w:GD62000] = 30019;
                break;

case 0xA600:    [w:GD62000] = 30019;
                break;

```

```

        case 0xA601:      [w:GD62000] = 30019;
                                break;

        default:          break;

    }

}

//Flag Reset
rst([b:GB30009]);
rst([b:GB30020]);
}

```

```
rst([b:GB30002]);
```

脚本号	30009	脚本名	Script30009
注释	坐标计算(输出) *1		
数据类型	有符号 BIN16	触发类型	通常

```
//Determine the cursor position from the touched coordinate.
```

```

if((([w:TMP0120] != [w:GS654]) || ([w:TMP0121] != [w:GS655])) && (([w:GS654] > Input_X) &&
([w:GS655] > Input_Y))
    && (([w:GS654] < Frame_X) && ([w:GS655] < Frame_Y)))
{
    [w:TMP0120] = [w:GS654];
    [w:TMP0121] = [w:GS655];

    if(([b:GD31562.b0] == OFF) && ([b:GB30001] == OFF))
    {
        //Save Current Cursor Position
        [w:TMP0122] = [w:GD31522];

        //X Coordinate
        //Change values to subtract depending on whether input or output.
        //Output
        [w:TMP0123] = [w:TMP0120] - Input_X;
        [w:TMP0124] = [w:TMP0123] / Object_X;
        [w:GD31520] = 16 - ([w:TMP0124] + 1);

        //Y Coordinate
        [w:TMP0125] = [w:TMP0121] - Input_Y;
        [w:TMP0126] = [w:TMP0125] / Object_Y;
        [w:GD31521] = [w:TMP0126] * 16;

        //Offset
        [w:GD31522] = [w:GD31520] + [w:GD31521];

        //If the offset position does not change, displayed parts do not change.
        if([w:TMP0122] != [w:GD31522])
        {
            //Cursor Display
            if([w:GD31000[w:GD31522]] < 4)
            {
                [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;
            }
        }
    }
}

```

```

        //Cursor Delete
        if([w:GD31000[w:TMP0122]] < 4)
        {
            [w:GD31000[w:TMP0122]] = 0;
        }else{
            [w:GD31000[w:TMP0122]] = [w:GD31000[w:TMP0122]] - 4;
        }
    }

    [w:GD32008] = [w:GD31522];      //Cursor Position Display
}

```

\*1: 使用了[脚本符号]。关于[脚本符号]，请参照「5.9.4 脚本符号」。



## 基本画面 30005

脚本号	30023	脚本名	Script30023
注释	B-30002, 30005 画面启动时处理		
数据类型	有符号 BIN16	触发类型	ON 中 GB40
<pre>//Read Statuses or Parameters if([b:GB30007] == OFF) {     if(([w:U03-G8192] != 0)    ([w:U03-G9984] != 0))     {         set([b:GD31568.b0]);     }else{         set([b:GD31562.b0]);     }     set([b:GB30007]); } }</pre>			
脚本号	30016	脚本名	Script30016
注释	脚本号 30015 启动		
数据类型	有符号 BIN16	触发类型	OFF 中 GD31563.b0
<pre>//After reading the status, start Script No.30015. if([b:GB30016] == OFF) {     set([b:GB30008]);     set([b:GB30016]); } }</pre>			
脚本号	30015	脚本名	Script30015
注释	B-30005 型号等		
数据类型	有符号 BIN16	触发类型	ON 中 GB30008
<pre>//Display Objects [w:GD31526] = 0; [w:TMP0004] = 0; [w:TMP0006] = 1; [w:TMP0100] = [w:GD31529];  while([w:GD31526] &lt; 15) {     //Check Connection Status     if(([w:GD40290] != 0) &amp;&amp; ([w:TMP0100] &lt; [w:GD40290]))     {         [w:GD31600] = [w:GD40291[w:TMP0100]]; //Connection ID         [w:TMP0100] = [w:TMP0100] + 1;          if(([w:GD31600] &amp; 0x0200) == 0)         {             set([b:GB30600]); //Output Bit             [w:GD31601] = ([w:GD31600] &amp; 0x0200) &gt;&gt; 9;         }else{             set([b:GB30601]); //Input Bit             [w:GD31601] = ([w:GD31600] &amp; 0x0200) &gt;&gt; 9;         }          //Offset         [w:TMP0000] = [w:GD31600] &amp; 0x00FF;     } }</pre>			

```

if(([w:GD31600] & 0x0200) == 0)
{
    //Output
    [w:TMP0001] = [w:GD40804[w:TMP0000]];
} else {
    //Input
    [w:TMP0001] = [w:GD41060[w:TMP0000]];
}
[w:GD31604] = [w:TMP0001] - 12288;

//Module Type
[w:TMP0002] = [w:GD31604] + 43;
[w:GD31617[w:GD31526]] = ([w:GD41316[w:TMP0002]] & 0x00C0) >> 6;

//Module Model No.
[w:TMP0005] = [w:GD31604] + 44;
[w:GD35096[w:GD31526]] = [w:GD41316[w:TMP0005]];

//Alarm Information
if([w:GD40547] != 0)
{
    [w:TMP0004] = 0;
    while([w:TMP0004] < [w:GD40547])
    {
        if([w:GD40548[w:TMP0004]] == [w:GD31600])
        {
            [w:GD31602] = 1;           //Alarm Occurrence
            if(([w:U03-G10256] != 304) && ([w:U03-G10256] < 200)
|| ([w:U03-G10256] > 202)))
            {
                [w:GD31633[w:GD31526]] = [w:U03-G10256];
            } else {
                [w:TMP0003] = ([w:TMP0001] - 0x3000) + 39;
                [w:GD31633[w:GD31526]] =
[w:GD41316[w:TMP0003]];
            }
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
    }
}

//Error Information
if([w:GD40032] != 0)
{
    [w:TMP0004] = 0;
    while([w:TMP0004] < [w:GD40032])
    {
        if([w:GD40033[w:TMP0004]] == [w:GD31600])
        {
            [w:GD31603] = 1;           //Error Occurrence
            if(([w:U03-G10256] == 304) || ([w:U03-G10256] >=
200) && ([w:U03-G10256] <= 202)))

```

```

        {
            [w:GD31633[w:GD31526]] = [w:U03-G10256];
        }else{
            [w:TMP0003] = ([w:TMP0001] - 0x3000) + 39;
            [w:GD31633[w:GD31526]] =
[w:GD41316[w:TMP0003]];

        }
        break;
    }
    [w:TMP0004] = [w:TMP0004] + 1;
}
}

//Display
if([w:GD31603] == 1)
{
    [w:GD31665[w:GD31526]] = 3;    //Error Occurrence
    [w:GD35032[w:GD31526]] = [w:GD31601];    //I/O
    [w:GD35064[w:GD31526]] = [w:GD31600] & 0x00FF;    //ID
    [w:GD31649[w:GD31526]] = [w:GD31600];
    [w:GD35643] = [w:GD35643] | ([w:TMP0006] << [w:GD31526]);
//Display Conditions

}else{
    if([w:GD31602] == 1)
    {
        [w:GD31665[w:GD31526]] = 2;    //Alarm Occurrence
        [w:GD35032[w:GD31526]] = [w:GD31601];    //I/O
        [w:GD35064[w:GD31526]] = [w:GD31600] & 0x00FF;    //ID
        [w:GD31649[w:GD31526]] = [w:GD31600];
        [w:GD35643] = [w:GD35643] | ([w:TMP0006] << [w:GD31526]);
//Display Conditions
    }else{
        [w:GD31665[w:GD31526]] = 1;    //Normal
        [w:GD35032[w:GD31526]] = [w:GD31601];    //I/O
        [w:GD35064[w:GD31526]] = [w:GD31600] & 0x00FF;    //ID
        [w:GD31649[w:GD31526]] = [w:GD31600];
        [w:GD31633[w:GD31526]] = 0;    //ステータス
        [w:GD35643] = [w:GD35643] | ([w:TMP0006] << [w:GD31526]);
//Display Conditions
    }
}

[w:GD31526] = [w:GD31526] + 1;
rst([b:GB30600]);    //Output Bit
rst([b:GB30601]);    //Input Bit

//Work Area Clear
[w:TMP0004] = 0;
fmov([w:TMP0004], [w:GD31600], 5);

}else{
    //If data does not exist, do not display.
    [w:GD31665[w:GD31526]] = 0xFFFF;

```

<pre> [w:GD35032[w:GD31526]] = 0xFFFF; [w:GD35064[w:GD31526]] = 0xFFFF; [w:GD35096[w:GD31526]] = 0xFFFF; [w:GD31617[w:GD31526]] = 0xFFFF; [w:GD31649[w:GD31526]] = 0x7FFE; [w:GD31633[w:GD31526]] = 24219; [w:GD35643] = [w:GD35643] ^ ([w:TMP0006] &lt;&lt; [w:GD31526]);          //Display Conditions [w:GD31526] = [w:GD31526] + 1; rst([b:GB30600]);          //Output Bit rst([b:GB30601]);          //Input Bit     } }  //Trigger Reset rst([b:GB30008]); </pre>			
脚本号	30021	脚本名	Script30021
注释	清除各种标志		
数据类型	有符号 BIN16	触发类型	关闭画面时
<pre> //Clear Flags rst([b:GB30006]); rst([b:GB30007]); rst([b:GB30003]);  [w:TMP0004] = 0; fmov([w:TMP0004], [w:GD31000], 512);          //Initialize Object Values(B-30003, 4) fmov([w:TMP0004], [w:GD31520], 3);          //Clear          Cursor          Position Information(B-30003, 4) fmov([w:TMP0004], [w:GD31600], 8);          //Work Area Clear [w:GD32008] = 0;          //Cursor Position Information Clear(B-30003, 4)  //Close Overlap Window [w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0; </pre>			
脚本号	30024	脚本名	Script30024
注释	上一页		
数据类型	有符号 BIN16	触发类型	上升沿 GB30014
<pre> //Scroll List Up if([w:GD31529] &gt; 0) {     [w:GD31529] = [w:GD31529] - 15;     set([b:GB30008]); } </pre>			
脚本号	30025	脚本名	Script30025
注释	下一页		
数据类型	有符号 BIN16	触发类型	上升沿 GB30015
<pre> //Scroll List Down if((((w:GD31529] + 15) &lt; [w:U03-G8960]) &amp;&amp; ([w:U03-G8960] &gt; 15)) {     [w:GD31529] = [w:GD31529] + 15;     set([b:GB30008]); } </pre>			

脚本号	30034	脚本名	Script30034
注释	B-30005 详细画面显示		
数据类型	无符号 BIN16	触发类型	ON 中 GB30018
<pre>//Determine detail screen to display. [w:TMP0000] = [w:GD31523] &amp; 0x00FF;  if((([w:GD31523] &amp; 0x0200) &gt;&gt; 8) == 0) {     //Output     [w:GD31567] = [w:GD40804[w:TMP0000]];    //Offset }else{     //Input     [w:GD31567] = [w:GD41060[w:TMP0000]];    //Offset }  [w:U03-G10320] = 0;    //Read Individual Parameter [w:U03-G10321] = [w:GD31523];    //Read Target ID  //Select base screen to display. switch([w:GD35096[w:GD31528]]) {     case 0x0100:    [w:GD62000] = 30006;                     break;      case 0x0101:    [w:GD62000] = 30006;                     break;      case 0x0102:    [w:GD62000] = 30008;                     break;      case 0x0103:    [w:GD62000] = 30008;                     break;      case 0x0104:    [w:GD62000] = 30007;                     break;      case 0x0105:    [w:GD62000] = 30007;                     break;      case 0x0106:    [w:GD62000] = 30009;                     break;      case 0x0107:    [w:GD62000] = 30009;                     break;      case 0x0400:    [w:GD62000] = 30006;                     break;      case 0x0401:    [w:GD62000] = 30006;                     break;      case 0x0402:    [w:GD62000] = 30008;                     break;</pre>			

```

case 0x0403:    [w:GD62000] = 30008;
                break;

case 0x0404:    [w:GD62000] = 30007;
                break;

case 0x0405:    [w:GD62000] = 30007;
                break;

case 0x0406:    [w:GD62000] = 30009;
                break;

case 0x0407:    [w:GD62000] = 30009;
                break;

case 0x0700:    [w:GD62000] = 30006;
                break;

case 0x0701:    [w:GD62000] = 30006;
                break;

case 0x0702:    [w:GD62000] = 30008;
                break;

case 0x0703:    [w:GD62000] = 30008;
                break;

case 0x0704:    [w:GD62000] = 30007;
                break;

case 0x0705:    [w:GD62000] = 30007;
                break;

case 0x0706:    [w:GD62000] = 30009;
                break;

case 0x0707:    [w:GD62000] = 30009;
                break;

case 0x1000:    [w:GD62000] = 30010;
                break;

case 0x1050:    [w:GD62000] = 30010;
                break;

case 0x1200:    [w:GD62000] = 30011;
                break;

case 0x1250:    [w:GD62000] = 30011;
                break;

case 0x1400:    [w:GD62000] = 30012;
                break;

```

```

    case 0x1450:    [w:GD62000] = 30012;
                                   break;

    case 0x2000:    [w:GD62000] = 30013;
                                   break;

    case 0x2300:    [w:GD62000] = 30014;
                                   break;

    case 0x2301:    [w:GD62000] = 30014;
                                   break;

    case 0x2302:    [w:GD62000] = 30015;
                                   break;

    case 0x2303:    [w:GD62000] = 30016;
                                   break;

    case 0x2600:    [w:GD62000] = 30017;
                                   break;

    case 0x2900:    [w:GD62000] = 30018;
                                   break;

    case 0xA000:    [w:GD62000] = 30019;
                                   break;

    case 0xA001:    [w:GD62000] = 30019;
                                   break;

    case 0xA300:    [w:GD62000] = 30019;
                                   break;

    case 0xA301:    [w:GD62000] = 30019;
                                   break;

    case 0xA600:    [w:GD62000] = 30019;
                                   break;

    case 0xA601:    [w:GD62000] = 30019;
                                   break;

    default:        break;
}

rst([b:GB30018]);

```

# 基本画面 30006～30019

脚本号	30027	脚本名	Script30027
注释	参数显示		
数据类型	有符号 BIN16	触发类型	ON 中 GB40
<pre> if([b:GB30017] == OFF) {     //Error Information     if([w:GD40032] != 0)     {         [w:TMP0000] = 0;         while([w:TMP0000] &lt; [w:GD40032])         {             if([w:GD40033[w:TMP0000]] == [w:U03-G10321])             {                 set([b:GB30028]);        //Error Occurrence Flag                 break;             }             [w:TMP0000] = [w:TMP0000] + 1;         }     }      if([b:GB30028] == ON)     {         //If errors occurred, read the most recently read settings.         set([b:GD31564.b0]);     }else{         //If errors are not occurred, write the latest information.         set([b:Y40]);     }     [w:GD31681] = 1;        //Specify Start Line of Processing Method     set([b:GB30017]); } </pre>			
脚本号	30014	脚本名	Script30014
注释	参数访问请求指令 OFF		
数据类型	有符号 BIN16	触发类型	ON 中 Y40
<pre> //Processing after Reading Parameters Individually if([w:U03-G10320] == 0) {     set([b:GB30024]);        //Read Flag }else{     set([b:GB30025]);        //Write Flag }  if([b:X42] == ON) {     [w:GD62007] = 30004;     rst([b:GB30024]);     rst([b:GB30025]);     rst([b:Y40]); }else{     if([b:X41] == ON)     { </pre>			



<pre> rst([b:GB30024]); rst([b:GB30025]); set([b:GB30026]); rst([b:Y40]); } } </pre>			
脚本号	30026	脚本名	Script30026
注释	模块型号编号修正		
数据类型	有符号 BIN16	触发类型	OFF 中 GD31565.b0
<pre> if([b:GB30021] == OFF) {     //Convert Module Model Number Into Decimal Number     [w:TMP0000] = [w:GD53629] &amp; 0x000F;    //Ones Digit     [w:TMP0001] = (([w:GD53629] &amp; 0x00F0) &gt;&gt; 4) * 10;    //Tenths Digit     [w:TMP0002] = (([w:GD53629] &amp; 0x0F00) &gt;&gt; 8) * 100;    //Hundreds Digit     [w:TMP0003] = (([w:GD53629] &amp; 0xF000) &gt;&gt; 12) * 1000;    //Thousands Digit      [w:GD31607] = [w:TMP0000] + [w:TMP0001] + [w:TMP0002] + [w:TMP0003];      set([b:GB30019]);     set([b:GB30021]); } </pre>			
脚本号	30037	脚本名	Script30037
注释	清除各种标志(详细画面)		
数据类型	有符号 BIN16	触发类型	关闭画面时
<pre> //Clear Flags rst([b:GB30006]); rst([b:GB30007]); rst([b:GB30003]);  rst([b:GB30016]); rst([b:GB30017]); rst([b:GB30021]);  rst([b:GB30028]);  [w:TMP0004] = 0; fmov([w:TMP0004], [w:GD31000], 512);    //Initialize Object Values(B-30002) fmov([w:TMP0004], [w:GD31600], 17);    //Work Area Clear  //Close Overlap Window [w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0; </pre>			
脚本号	30038	脚本名	Script30038
注释	显示模块详细信息		
数据类型	有符号 BIN16	触发类型	ON 中 GB30019
<pre> //Display Detail Information  //Work Area Clear [w:TMP0004] = 0; fmov([w:TMP0004], [w:GD31600], 7); </pre>			

```

//Alarm Information
if([w:GD40547] != 0)
{
    [w:TMP0004] = 0;
    while([w:TMP0004] < [w:GD40547])
    {
        if([w:GD40548[w:TMP0004]] == [w:GD53604])
        {
            [w:GD31602] = 1;          //Alarm Occurrence
            if(([w:U03-G10256] != 305) && (([w:U03-G10256] != 304) &&
            (([w:U03-G10256] < 200) || ([w:U03-G10256] > 202))))
            {
                [w:GD31608] = [w:U03-G10256];
            }else{
                [w:GD31608] = [w:GD53624];
            }
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
    }
}

//Error Information
if([w:GD40032] != 0)
{
    [w:TMP0004] = 0;
    while([w:TMP0004] < [w:GD40032])
    {
        if([w:GD40033[w:TMP0004]] == [w:GD53604])
        {
            [w:GD31603] = 1;          //Error Occurrence
            if(([w:U03-G10256] != 305) && (([w:U03-G10256] == 304) ||
            (([w:U03-G10256] >= 200) && ([w:U03-G10256] <= 202))))
            {
                [w:GD31608] = [w:U03-G10256];
            }else{
                [w:GD31608] = [w:GD53624];
            }
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
    }
}

//Device Parameter
//I/O Points Pattern
[w:GD31605] = ([w:GD53628] & 0x00C0) >> 6;      //0: Input, 1: Output, 2: Combined
[w:GD31606] = ([w:GD53628] & 0x003F) + 1;      //Points

//ON/OFF Information
[w:TMP0008] = 0;
[w:TMP0000] = [w:GD53604] & 0x00FF;
if([w:GD31605] == 0)    //Input Case
{

```

```

[w:TMP0004] = 0;

while([w:TMP0004] < [w:GD31606])
{
    [w:TMP0009] = [w:TMP0000] / 16;
    [w:TMP0010] = [w:TMP0000] % 16;

    [w:GD31682[w:TMP0008]] = [w:TMP0009];
    [w:TMP0012] = 0x0001;
    [w:GD31990[w:TMP0008]] = [w:TMP0012] << [w:TMP010];

    if([w:GD31603] == 1)    //Check If Error Occurred
    {
        [w:GD31609[w:TMP0008]] = 3;    //Error Occurrence
    }else{
        if([w:GD31602] == 1)    //Check If Alarms Occurred
        {
            [w:GD31609[w:TMP0008]] = 2;    //Alarm Occurrence
        }else{
            [w:GD31609[w:TMP0008]] = 1;    //Normal
            [w:GD31608] = 0;
        }
    }

    [w:TMP0000] = [w:TMP0000] + 1;
    [w:TMP0008] = [w:TMP0008] + 1;
    [w:TMP0004] = [w:TMP0004] + 1;
}

}else{
    if([w:GD31605] == 1)    //Output Case
    {
        [w:TMP0004] = 0;

        while([w:TMP0004] < [w:GD31606])
        {
            [w:TMP0009] = [w:TMP0000] / 16;
            [w:TMP0010] = [w:TMP0000] % 16;

            [w:GD31682[w:TMP0008]] = [w:TMP0009] + 4096;
            [w:TMP0012] = 0x0001;
            [w:GD31990[w:TMP0008]] = [w:TMP0012] << [w:TMP010];

            if([w:GD31603] == 1)    //Check If Error Occurred
            {
                [w:GD31609[w:TMP0008]] = 3;    //Error Occurrence
            }else{
                if([w:GD31602] == 1)    //Check If Alarms Occurred
                {
                    [w:GD31609[w:TMP0008]] = 2;    //Alarm Occurrence
                }else{
                    [w:GD31609[w:TMP0008]] = 1;    //Normal
                    [w:GD31608] = 0;
                }
            }
        }
    }
}

```

```

    }

    [w:TMP0000] = [w:TMP0000] + 1;
    [w:TMP0008] = [w:TMP0008] + 1;
    [w:TMP0004] = [w:TMP0004] + 1;
}

}else{
    if([w:GD31605] == 2)    //Combined Case
    {
        [w:TMP0004] = 0;
        [w:TMP0011] = [w:GD31606] / 2; //If combined, points are reduced by
half.
        while([w:TMP0004] < [w:TMP0011])
        {
            [w:TMP0009] = [w:TMP0000] / 16;
            [w:TMP0010] = [w:TMP0000] % 16;

            //Output
            [w:GD31682[w:TMP0008]] = [w:TMP0009] + 4096;
            [w:TMP0012] = 0x0001;
            [w:GD31990[w:TMP0008]] = [w:TMP0012] << [w:TMP010];

            if([w:GD31603] == 1)    //Check If Error Occurred
            {
                [w:GD31609[w:TMP0008]] = 3;    //Error Occurrence
            }else{
                if([w:GD31602] == 1)    //Check If Error Occurred
                {
                    [w:GD31609[w:TMP0008]] = 2;    //Alarm
Occurrence

                }else{
                    [w:GD31609[w:TMP0008]] = 1;    //Normal
                    [w:GD31608] = 0;
                }
            }

            //Input
            [w:GD31683[w:TMP0008]] = [w:TMP0009];
            [w:TMP0012] = 0x0001;
            [w:GD31991[w:TMP0008]] = [w:TMP0012] << [w:TMP010];

            if([w:GD31603] == 1)    //Check If Error Occurred
            {
                [w:GD31610[w:TMP0008]] = 3;    //Error Occurrence
            }else{
                if([w:GD31602] == 1)    //Check If Alarms Occurred
                {
                    [w:GD31610[w:TMP0008]] = 2;    //Alarm
Occurrence

                }else{
                    [w:GD31610[w:TMP0008]] = 1;    //Normal
                    [w:GD31608] = 0;
                }
            }
        }
    }
}

```

<pre> }  [w:TMP0000] = [w:TMP0000] + 1; [w:TMP0008] = [w:TMP0008] + 1; [w:TMP0004] = [w:TMP0004] + 1; } } } } }  rst([b:GB30019]); </pre>			
脚本号	30039	脚本名	Script30039
注释	参数再读取		
数据类型	有符号 BIN16	触发类型	OFF 中 GD31563. b0
<pre> //Re-read Parameters if([b:GB30016] == OFF) {     set([b:GD31564. b0]);     set([b:GB30016]); } </pre>			
脚本号	30040	脚本名	Script30040
注释	参数个别读取		
数据类型	有符号 BIN16	触发类型	ON 中 GB30026
<pre> //Processing after Reading Parameters if([w:U03-G10320] == 0) {     set([b:GD31564. b0]); }else{     [w:U03-G10320] = 0;     set([b:Y40]); }  rst([b:GB30026]); </pre>			
脚本号	30068	脚本名	Script30068
注释	机器参数读取		
数据类型	有符号 BIN16	触发类型	ON 中 GB30029
<pre> //Read the detail information of slave module.  //Error Information if([w:GD40032] != 0) {     [w:TMP0000] = 0;     while([w:TMP0000] &lt; [w:GD40032])     {         if([w:GD40033[w:TMP0000]] == [w:U03-G10321])         {             set([b:GB30028]);    //Error Occurrence Flag             break;         }         [w:TMP0000] = [w:TMP0000] + 1;     } } </pre>			

```

if([b:GB30028] == ON)
{
    //If errors occurred, display error message.
    [w:GD62007] = 30004;
    rst([b:GB30024]);
}else{
    //If errors are not occurred, write the latest information.
    [w:U03-G10320] = 0;
    [w:U03-G10321] = [w:GD53604];
    set([b:Y0040]);
}

```

```
rst([b:GB30029]);
```

脚本号	30069	脚本名	Script30069
注释	机器参数写入		
数据类型	有符号 BIN16	触发类型	ON 中 GB30030

```
//Write the detail information of slave module.
```

```

//Error Information
if([w:GD40032] != 0)
{
    [w:TMP0000] = 0;
    while([w:TMP0000] < [w:GD40032])
    {
        if([w:GD40033[w:TMP0000]] == [w:U03-G10321])
        {
            set([b:GB30028]);          //Error Occurrence Flag
            break;
        }
        [w:TMP0000] = [w:TMP0000] + 1;
    }
}

```

```

if([b:GB30028] == ON)
{
    //If errors occurred, display error message.
    [w:GD62007] = 30004;
    rst([b:GB30025]);
}else{
    //If errors are not occurred, write the latest information.
    [w:U03-G10320] = 1;
    [w:U03-G10321] = [w:GD53604];
    set([b:GD31570.b0]);
}

```

```
rst([b:GB30030]);
```

## 基本画面 30100

脚本号	30062	脚本名	Script30062
注释	B-30100 iQSS 备份设置一览读取		
数据类型	有符号 BIN16	触发类型	ON 中 GB54010
<pre>//Start from Project Script //Read the iQSS Backup Setting list and the iQSS Backup History list from Recipe.  //Check if the A drive is accessible. if([b:GS251.b0] == ON) {     set([b:GB54016]);          //Recipe No.30001 Write Trigger     [w:GD54090] = 1;          //Recipe No.30001 Record No.      [w:GD54004] = 1;          //Next Data Acquisition Flag }else{     rst([b:GB54014]);          //Screen Switching Switch Action Conditions OFF     [w:GD54006] = 32015;     [w:GD62007] = 30101; } rst([b:GB54010]);</pre>			
脚本号	30063	脚本名	Script30063
注释	B-30100 配方写入触发 OFF		
数据类型	有符号 BIN16	触发类型	ON 中 GD54012.b4
<pre>//Turn OFF Write Trigger of Recipe  //If the flag was on, read the next data. if([w:GD54004] == 1) {     set([b:GB54011]); }else{     rst([b:GB54014]);          //Screen Switching Switch Action Conditions OFF }  if([w:GD54013] == 30001) {     rst([b:GB54016]);          //Recipe No.30001 Write Trigger }else{     rst([b:GB54018]);          //Recipe No.30002 Write Trigger }  rst([b:GD54012.b4]);</pre>			
脚本号	30064	脚本名	Script30064
注释	B-30100 iQSS 备份记录一览读取		
数据类型	有符号 BIN16	触发类型	ON 中 GB54011
<pre>//Read iQSS Backup History  [w:GD54091] = 1;          //Recipe No.30002 Record No. set([b:GB54018]);          //Recipe No.30002 Write Trigger  [w:GD54004] = 0;          //Flag Clear  rst([b:GB54011]);</pre>			

## 基本画面 30101

脚本号	30046	脚本名	Script30046
注释	B-30101 iQSS 备份设置传送		
数据类型	有符号 BIN16	触发类型	ON 中 GB54000
<pre>//Transfer iQSS parameters from the iQSS Backup Setting list to the editor area. [w:TMP0000] = [w:GD54007] * 21; [w:GD54100[w:TMP0000]] = [w:GD54007] + 1; bmov([w:GD54100[w:TMP0000]], [w:GD54015], 20); [w:GD54086] = [w:GD54120[w:TMP0000]];  //If target model is specified, assign target model and I/O No. if([w:GD54075] != 0) {     [w:GD54016] = [w:GD54075];    //Target Model     [w:GD54030] = [w:GD54089];    //I/O No.      //Execute only when target model is AnyWireASLINK.     if([w:GD54075] == 1)     {         set([b:GD54086.b0]);    //Flag to Select Output, Input/Combined     } }  //Folder Numbers Fixed [w:GD54029] = -2;    //Folder Numbers Automatic Acquisition  [w:GD62000] = 30102;    //Base Screen Switching  rst([b:GB54000]);</pre>			
脚本号	30057	脚本名	Script30057
注释	B-30101 数据删除		
数据类型	有符号 BIN16	触发类型	ON 中 GB54006
<pre>//Delete the specified data from the iQSS Backup Setting list.  //Check if the data is selected if([w:GD54000] != 0) {     //Offset     [w:TMP0000] = [w:GD54007] * 21;      //If no data exists in the selected place, do not delete     if([w:GD54101[w:TMP0000]] != 0)     {         //Clear Data         [w:TMP0002] = 0;         fmov([w:TMP0002], [w:GD54100[w:TMP0000]], 21);          [w:GD54090] = 1;    //Recipe No.30001 Record No.         set([b:GB54017]);    //Recipe No.30001 Read Trigger     }else{         [w:GD54006] = 32012;</pre>			



<pre> [w:GD62007] = 30101;     } } else {     [w:GD54006] = 32012;     [w:GD62007] = 30101; }  rst([b:GB54006]); </pre>			
脚本号	30051	脚本名	Script30051
注释	检查 iQSS 备份/恢复使用权取得信息		
数据类型	有符号 BIN16	触发类型	周期 1 秒
<pre> //Periodically check the status of obtaining the right to use. if([w:SD1435] == [w:SD1436]) {     set([b:GB54004]);          //iQSS Backup Permission Bit ON } else {     rst([b:GB54004]);          //iQSS Backup Permission Bit OFF } </pre>			
脚本号	30048	脚本名	Script30048
注释	配方触发 OFF		
数据类型	有符号 BIN16	触发类型	ON 中 GD54012. b5
<pre> //Turn OFF Read Trigger of Recipe  if([w:GD54013] == 30001) {     rst([b:GB54017]);          //Recipe No.30001 Read Trigger } else {     rst([b:GB54019]);          //Recipe No.30002 Read Trigger }  rst([b:GD54012. b5]); </pre>			
脚本号	30050	脚本名	Script30050
注释	B-30101 iQSS 备份预处理 1		
数据类型	有符号 BIN16	触发类型	ON 中 GB54013
<pre> //iQSS Backup Preprocessing  //Offset [w:TMP0000] = [w:GD54007] * 21;  //Check if the A drive is accessible. if([b:GS251. b0] == ON) {      //If module type is 0, no settings are entered and backup will not be executed.     if([w:GD54101[w:TMP0000]] != 0)     {         //Check the number of backed up cases.         //When exceeding 100 cases, display a message before making backup.         if([w:GD54101[w:TMP0000]] == 1)         {             //AnyWireASLINK             if([w:GD54087] == 100) </pre>			

```

        {
            [w:GD54006] = 280;
        }else{
            [w:GD54006] = 268;
        }
    }else{
        //CC-Link
        if([w:GD54088] == 100)
        {
            [w:GD54006] = 281;
        }else{
            [w:GD54006] = 268;
        }
    }

    [w:GD62007] = 30102;

} else{
    [w:GD54006] = 32007;
    [w:GD62007] = 30101;
}
} else{
    [w:GD54006] = 32017;
    [w:GD62007] = 30101;
}

```

rst([b:GB54013]);

脚本号	30052	脚本名	Script30052
注释	B-30101 iQSS 备份预处理 2		
数据类型	有符号 BIN16	触发类型	ON 中 GB54002

//Transfer Specified Contents to PLC Devices

//Setting Name

bmov([w:GD54102[w:TMP0000]], [w:GD54017], 12);

//SD1437: iQSS Backup/Restoration Target Model/Execution Unit Setting

[w:SD1437] = (([w:GD54116[w:TMP0000]] & 0x00FF) << 8) + ([w:GD54101[w:TMP0000]] & 0x00FF);

//Execution Unit Display

[w:GD54016] = [w:GD54101[w:TMP0000]];

//SD1438: iQSS Backup/Restoration Target Folder No.

[w:SD1438] = [w:GD54114[w:TMP0000]];

//SD1439: iQSS Backup/Restoration Target Setting (Target Module I/O No.)

[w:SD1439] = [w:GD54115[w:TMP0000]];

//SD1440: iQSS Backup/Restoration Target Setting (Target Device 1 ASLINK: ID No., CC-Link: Station No.)

[w:SD1440] = [w:GD54117[w:TMP0000]];

//SD1441: iQSS Backup/Restoration Target Setting (Target Device 2 ASLINK: Not Used, CC-Link: Sub Station No.)

[w:SD1441] = [w:GD54118[w:TMP0000]];

<pre>//SD1444: iQSS Backup/Restoration Action Setting [w:SD1444] = [w:GD54119[w:TMP0000]];  //If the right to use the iQSS backup is obtained, start backup. if(([b:GB54004] == ON) &amp;&amp; ([w:SD1446] == 0x0001)) {     set([b:SM1436]);     [w:GD62000] = 30103; } else {     [w:GD54006] = 32006;     [w:GD62007] = 30101; }  rst([b:GB54002]);</pre>			
脚本号	30059	脚本名	Script30059
注释	画面切换时处理		
数据类型	有符号 BIN16	触发类型	关闭画面时
<pre>//Clear Flags  rst([b:GB54008]); rst([b:GB54009]);  //Close Overlap Window [w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0;</pre>			

#### 基本画面 30102

脚本号	30047	脚本名	Script30047
注释	B-30102 iQSS 备份设置保存		
数据类型	有符号 BIN16	触发类型	ON 中 GB54001
<pre>//Save iQSS Backup Setting of Device to Recipe  //Check if the A drive is accessible. if([b:GS251.b0] == ON) {     //Check if target model is specified.     if([w:GD54016] != 0)     {         //Check whether the execution unit is specified.         if([w:GD54031] != 0)         {             //Check whether the correct value is in Station No. and Station Sub.             if(((w:GD54031] == 1)    (([w:GD54031] == 2) &amp;&amp; ([w:GD54032] &gt;= 0))    (([w:GD54031] &gt;= 3) &amp;&amp; ([w:GD54032] &gt;= 0) &amp;&amp; ([w:GD54033] &gt;= 0))))             {                 //If all checks are OK, save to recipe.                 [w:TMP0000] = ([w:GD54015] - 1) * 21;                 bmov([w:GD54015], [w:GD54100[w:TMP0000]], 20);                 [w:GD54120[w:TMP0000]] = [w:GD54086];                 [w:GD54090] = 1; //Recipe No. 30001 Record No.                 set([b:GB54017]); //Recipe No. 30001 Read Trigger             }         }     } }</pre>			

```

        }else{
            [w:GD54006] = 32011;
            [w:GD62007] = 30101;
        }
    }else{
        [w:GD54006] = 32011;
        [w:GD62007] = 30101;
    }
}else{
    [w:GD54006] = 32009;
    [w:GD62007] = 30101;
}
}
}else{
    [w:GD54006] = 32017;
    [w:GD62007] = 30101;
}
}
rst([b:GB54001]);

```

脚本号	30067	脚本名	Script30067
注释	B-30102 配方触发 OFF		
数据类型	有符号 BIN16	触发类型	ON 中 GD54012. b5

//Turn OFF Read Trigger of Recipe

```
rst([b:GB54017]);          //Recipe No. 30001 Read Trigger
```

```

[w:GD54006] = 32016;
[w:GD62007] = 30101;

```

```
rst([b:GD54012. b5]);
```

脚本号	30054	脚本名	Script30054
注释	B-30102 注释显示控制		
数据类型	有符号 BIN16	触发类型	通常

//Switch Comments Between CC-Link and ASLINK

```

switch([w:GD54016])
{
    //ASLINK
    case 1: [w:GD54035] = 226;          //ID or Station
            [w:GD54036] = 232;          //All IDs or All Stations
            [w:GD54037] = 233;          //ID Specification or Station Specification
            [w:GD54038] = 285;          //Not Specified or Station Sub
            [w:GD54039] = 0;            //Not Specified or Station Sub Specified
            break;

    //CC-Link
    case 2: [w:GD54035] = 225; //ID or Station
            [w:GD54036] = 242; //All IDs or All Stations
            [w:GD54037] = 243;          //ID Specification or Station Specification
            [w:GD54038] = 227;          //Not Specified or Station Sub
            [w:GD54039] = 244;          //Not Specified or Station Sub Specified
            break;

    //Other

```

default:	<pre> [w:GD54035] = 0;          //ID or Station [w:GD54036] = 0;          //All IDs or All Stations [w:GD54037] = 0;          //ID Specification or Station Specification [w:GD54038] = 0;          //Not Specified or Station Sub [w:GD54039] = 0;          //Not Specified or Station Sub Specified break; </pre>		
脚本号	30053	脚本名	Script30053
注释	B-30102 输入输出判定		
数据类型	有符号 BIN16	触发类型	上升沿/下降沿 GD54086.b2
<pre> //While using AnyWireASLINK, turn OFF/ON the 9th bit depending on the unit type.  if([b:GD54086.b2] == ON) {     [w:GD54032] = [w:GD54032]   0x0200;    //Input/Combined }else{     [w:GD54032] = [w:GD54032] &amp; 0x00FF;    //Output } </pre>			

### 基本画面 30103

脚本号	30055	脚本名	Script30055
注释	B-30103、30106 画面启动时动作		
数据类型	有符号 BIN16	触发类型	ON 中 GB40
<pre> //Make various settings when starting screen  //Actions at Screen Startup Only if([b:GB54008] == OFF) {     //Backup Execution Date/Time     [w:GD54040] = [w:GS650];     [w:GD54041] = [w:GS651];     [w:GD54042] = [w:GS652];      set([b:GB54008]); } </pre>			
脚本号	30060	脚本名	Script30060
注释	B-30103 正常结束		
数据类型	有符号 BIN16	触发类型	上升沿 SM1437
<pre> //After iQSS backup is completed, save results to Recipe.  //When ASLINK or CC-Link exceeding 100 cases, delete the oldest backup history. //If the screen was created as a dedicated screen, it does not work. if([w:GD54075] == 0) {     if(([w:GD54087] == 100) &amp;&amp; (([w:SD1437] &amp; 0x00FF) == 1))     {         [w:TMP0001] = 0;         [w:TMP0003] = 0;         //Search for ASLINK history from all history.         while([w:TMP0001] &lt;= 199)         {             [w:TMP0002] = [w:TMP0001] * 27; //Offset             if([w:GD56201[w:TMP0002]] == 1) </pre>			

```

        {
            [w:TMP0003] = [w:TMP0003] + 1;
            if([w:TMP0003] == 100)
            {
                //If history reached 100th case, exit loop.
                break;
            }
        }
        [w:TMP0001] = [w:TMP0001] + 1;
    }

    [w:TMP0004] = 0;
    fmov([w:TMP0004], [w:GD56200[w:TMP0002]], 27);

    //Move the data after deleted portion up.
    while([w:TMP0001] < 199)
    {
        [w:TMP0002] = [w:TMP0001] * 27; //Offset
        [w:TMP0005] = ([w:TMP0001] + 1) * 27; //Copy Destination Offset

        bmov([w:GD56200[w:TMP0005]], [w:GD56200[w:TMP0002]], 27);

        [w:TMP0001] = [w:TMP0001] + 1;
    }
} else {
    if(([w:GD54088] == 100) && (([w:SD1437] & 0x00FF) == 2))
    {
        [w:TMP0001] = 0;
        [w:TMP0003] = 0;
        //Search for CC-Link history from all history.
        while([w:TMP0001] <= 199)
        {
            [w:TMP0002] = [w:TMP0001] * 27; //Offset
            if([w:GD56201[w:TMP0002]] == 2)
            {
                [w:TMP0003] = [w:TMP0003] + 1;
                if([w:TMP0003] == 100)
                {
                    //If history reached 100th case, exit loop.
                    break;
                }
            }
            [w:TMP0001] = [w:TMP0001] + 1;
        }

        [w:TMP0004] = 0;
        fmov([w:TMP0004], [w:GD56200[w:TMP0002]], 27);

        //Move the data after deleted portion up.
        while([w:TMP0001] < 199)
        {
            [w:TMP0002] = [w:TMP0001] * 27; //Offset
            [w:TMP0005] = ([w:TMP0001] + 1) * 27; //Copy Destination Offset

```

```

        bmov([w:GD56200[w:TMP0005]], [w:GD56200[w:TMP0002]], 27);

        [w:TMP0001] = [w:TMP0001] + 1;
    }
}

//Offset
if(([w:GD54075] == 1) || ([w:GD54075] == 2))
{
    //ASLINK or CC-Link Dedicated Screen
    [w:TMP0001] = 99;
}else{
    //General Screen
    [w:TMP0001] = 199;
}

//Store the iQSS backup result in descending order.
while([w:TMP0001] > 0)
{
    [w:TMP0000] = [w:TMP0001] * 27;
    [w:TMP0002] = ([w:TMP0001] - 1) * 27;

    bmov([w:GD56200[w:TMP0002]], [w:GD56200[w:TMP0000]], 27);
    [w:GD56200[w:TMP0000]] = [w:TMP0001] + 1;

    [w:TMP0001] = [w:TMP0001] - 1;
}

//No.
[w:GD56200] = 1;

//SD1437: iQSS Backup/Restoration Target Model/Execution Unit Setting
[w:GD56201] = [w:SD1437] & 0x00FF;
[w:GD56216] = ([w:SD1437] & 0xFF00) >> 8;

//Setting Name
bmov([w:GD54017], [w:GD56202], 12);

//SD1451: iQSS Backup Folder No.
[w:GD56214] = [w:SD1451];

//SD1439: iQSS Backup/Restoration Target Setting (Target Module I/O No.)
[w:GD56215] = [w:SD1439];

//SD1440: iQSS Backup/Restoration Target Setting (Target Device 1 ASLINK: ID No., CC-Link:
Station No.)
[w:GD56217] = [w:SD1440];

//SD1441: iQSS Backup/Restoration Target Setting (Target Device 2 ASLINK: Not Used, CC-Link:
Sub Station No.)
[w:GD56218] = [w:SD1441];

```

```

//SD1444: iQSS Backup/Restoration Action Setting
[w:GD56219] = [w:SD1444];

//iQSS Backup Execution Date/Time
[w:GD56220] = [w:GD54040];
[w:GD56221] = [w:GD54041];
[w:GD56222] = [w:GD54042];

//SD1447: iQSS Backup/Restoration Number of Target Devices
[w:GD56223] = [w:SD1447];

//SD1448: iQSS Backup/Restoration Number of Normal Completion Devices
[w:GD56224] = [w:SD1448];

//SD1449: iQSS Backup/Restoration Number of Abnormal Completion Devices
[w:GD56225] = [w:SD1449];

//iQSS Backup Cases
if([w:GD56201] == 1)
{
    //AnyWireASLINK
    if([w:GD54087] >= 100)
    {
        [w:GD54087] = 100;
    }else{
        [w:GD54087] = [w:GD54087] + 1;
    }
}else{
    //CC-LINK
    if([w:GD54088] >= 100)
    {
        [w:GD54088] = 100;
    }else{
        [w:GD54088] = [w:GD54088] + 1;
    }
}

//Save to Recipe
[w:GD54091] = 1;           //Recipe No.30002 Record No.
set([b:GB54019]);         //Recipe No.30002 Read Trigger

//Change messages depending on whether canceled or not.
if([b:SM1442] == ON)
{
    [w:GD54006] = 32002; //Backup Cancel Message
}else{
    [w:GD54006] = 32000; //Backup Successful Message
}

[w:GD62007] = 30101;//Notification Dialog

```

脚本号	30043	脚本名	Script30043
注释	B-30103 异常结束		
数据类型	有符号 BIN16	触发类型	ON 中 SM1438



```

//After iQSS backup is completed, save results to Recipe.

if([b:GB54009] == OFF)
{
    //If no folders were created, do not save history.
    if(([w:SD1447] != 0) && ([w:SD1448] != 0))
    {
        //When ASLINK or CC-Link exceeding 100 cases, delete the oldest backup history.
        //If the screen was created as a dedicated screen, it does not work.
        if([w:GD54075] == 0)
        {
            if(([w:GD54087] == 100) && (([w:SD1437] & 0x00FF) == 1))
            {
                [w:TMP0001] = 0;
                [w:TMP0003] = 0;
                //Search for ASLINK history from all history.
                while([w:TMP0001] <= 199)
                {
                    [w:TMP0002] = [w:TMP0001] * 27; //Offset
                    if([w:GD56201[w:TMP0002]] == 1)
                    {
                        [w:TMP0003] = [w:TMP0003] + 1;
                        if([w:TMP0003] == 100)
                        {
                            //If history reached 100th case, exit
                            break;
                        }
                    }
                    [w:TMP0001] = [w:TMP0001] + 1;
                }

                [w:TMP0004] = 0;
                fmov([w:TMP0004], [w:GD56200[w:TMP0002]], 27);

                //Move the data after deleted portion up.
                while([w:TMP0001] < 199)
                {
                    [w:TMP0002] = [w:TMP0001] * 27; //Offset
                    [w:TMP0005] = ([w:TMP0001] + 1) * 27; //Copy
                    Destination Offset

                    bmov([w:GD56200[w:TMP0005]], [w:GD56200[w:TMP0002]], 27);

                    [w:TMP0001] = [w:TMP0001] + 1;
                }
            }
        }
    }
    }else{
        if(([w:GD54088] == 100) && (([w:SD1437] & 0x00FF) == 2))
        {
            [w:TMP0001] = 0;
            [w:TMP0003] = 0;
            //Search for CC-Link history from all history.

```

```

while([w:TMP0001] <= 199)
{
    [w:TMP0002] = [w:TMP0001] * 27; //Offset
    if([w:GD56201[w:TMP0002]] == 2)
    {
        [w:TMP0003] = [w:TMP0003] + 1;
        if([w:TMP0003] == 100)
        {
            //If history reached 100th
            break;
        }
    }
    [w:TMP0001] = [w:TMP0001] + 1;
}

[w:TMP0004] = 0;
fmov([w:TMP0004], [w:GD56200[w:TMP0002]], 27);

//Move the data after deleted portion up.
while([w:TMP0001] < 199)
{
    [w:TMP0002] = [w:TMP0001] * 27; //Offset
    [w:TMP0005] = ([w:TMP0001] + 1) * 27; //Copy
    bmov([w:GD56200[w:TMP0005]], [w:GD56200[w:TMP0002]], 27);

    [w:TMP0001] = [w:TMP0001] + 1;
}

}

//Offset Calculation
if((([w:GD54075] == 1) || ([w:GD54075] == 2))
{
    //ASLINK or CC-Link Dedicated Screen
    [w:TMP0001] = 99;
} else {
    //General Screen
    [w:TMP0001] = 199;
}

//Store the iQSS backup result in descending order.
while([w:TMP0001] > 0)
{
    [w:TMP0000] = [w:TMP0001] * 27;
    [w:TMP0002] = ([w:TMP0001] - 1) * 27;

    bmov([w:GD56200[w:TMP0002]], [w:GD56200[w:TMP0000]], 27);
    [w:GD56200[w:TMP0000]] = [w:TMP0001] + 1;
}

```

case, exit loop.

Destination Offset

```

        [w:TMP0001] = [w:TMP0001] - 1;
    }

    //No.
    [w:GD56200] = 1;

    //SD1437: iQSS Backup/Restoration Target Model/Execution Unit Setting
    [w:GD56201] = [w:SD1437] & 0x00FF;
    [w:GD56216] = ([w:SD1437] & 0xFF00) >> 8;

    //Setting Name
    bmov([w:GD54017], [w:GD56202], 12);

    //SD1451: iQSS Backup Folder No.
    [w:GD56214] = [w:SD1451];

    //SD1439: iQSS Backup/Restoration Target Setting (Target Module I/O No.)
    [w:GD56215] = [w:SD1439];

    //SD1440: iQSS Backup/Restoration Target Setting (Target Device 1 ASLINK: ID
    No., CC-Link: Station No.)
    [w:GD56217] = [w:SD1440];

    //SD1441: iQSS Backup/Restoration Target Setting (Target Device 2 ASLINK: Not
    Used, CC-Link: Sub Station No.)
    [w:GD56218] = [w:SD1441];

    //SD1444: iQSS Backup/Restoration Action Setting
    [w:GD56219] = [w:SD1444];

    //iQSS Backup Execution Date/Time
    [w:GD56220] = [w:GD54040];
    [w:GD56221] = [w:GD54041];
    [w:GD56222] = [w:GD54042];

    //SD1447: iQSS Backup/Restoration Number of Target Devices
    [w:GD56223] = [w:SD1447];

    //SD1448: iQSS Backup/Restoration Number of Normal Completion Devices
    [w:GD56224] = [w:SD1448];

    //SD1449: iQSS Backup/Restoration Number of Abnormal Completion Devices
    [w:GD56225] = [w:SD1449];

    //iQSS Backup Cases
    if([w:GD56201] == 1)
    {
        //AnyWireASLINK
        if([w:GD54087] >= 100)
        {
            [w:GD54087] = 100;
        }else{
            [w:GD54087] = [w:GD54087] + 1;
        }
    }

```

<pre> }else{     //CC-Link     if([w:GD54088] &gt;= 100)     {         [w:GD54088] = 100;     }else{         [w:GD54088] = [w:GD54088] + 1;     } }  //Save to Recipe [w:GD54091] = 1;          //Recipe No.30002 Record No. set([b:GB54019]);        //Recipe No.30002 Read Trigger }  if([w:SD1452] != 0) {     if([w:SD1452] == 16894)     {         [w:GD54006] = 32001; //Backup Failure Message         [w:GD62007] = 30101; //Notification Dialog     }else{         [w:GD54006] = [w:SD1452]; //Backup Failure Message         [w:GD62007] = 30101; //Notification Dialog     } } }else{     if([w:SD1453] != 0)     {         [w:GD54006] = [w:SD1453]; //Backup Failure Message         [w:GD62007] = 30101; //Notification Dialog     }else{         [w:GD54006] = 32001; //Backup Failure Message         [w:GD62007] = 30101; //Notification Dialog     } }  set([b:GB54009]); } </pre>			
脚本号	30048	脚本名	Script30048
注释	配方触发 OFF		
数据类型	有符号 BIN16	触发类型	ON 中 GD54012. b5
<pre> //Turn OFF Read Trigger of Recipe  if([w:GD54013] == 30001) {     rst([b:GB54017]);          //Recipe No.30001 Read Trigger }else{     rst([b:GB54019]);          //Recipe No.30002 Read Trigger }  rst([b:GD54012. b5]); </pre>			
脚本号	30059	脚本名	Script30059
注释	画面切换时处理		

数据类型	有符号 BIN16	触发类型	关闭画面时
<pre>//Clear Flags  rst([b:GB54008]); rst([b:GB54009]);  //Close Overlap Window [w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0;</pre>			
脚本号	30065	脚本名	Script30065
注释	B-30103、30106 iQSS 备份/恢复中止		
数据类型	有符号 BIN16	触发类型	ON 中 GB54002
<pre>//Abort iQSS Backup or iQSS Restoration set([b:SM1442]);          //iQSS Backup/Restoration Cancel Request  rst([b:GB54002]);</pre>			

#### 基本画面 30104

脚本号	30056	脚本名	Script30056
注释	B-30104 iQSS 恢复设置确认		
数据类型	有符号 BIN16	触发类型	ON 中 GB54005
<pre>//Check iQSS backup contents. [w:TMP0000] = ([w:GD54008] * 27) + [w:GD54003]; bmov([w:GD56200[w:TMP0000]], [w:GD54043], 20);  [w:GD62000] = 30105;  rst([b:GB54005]);</pre>			
脚本号	30058	脚本名	Script30058
注释	B-30104 数据删除		
数据类型	有符号 BIN16	触发类型	ON 中 GB54006
<pre>//After deleting the specified data from the iQSS Backup History list, move the rest of the list up.  //Check whether to delete all or delete specified range cases. if([b:GB54012] == OFF) {     //Check if the data is selected     if([w:GD54002] != 0)     {         //Offset         [w:TMP0000] = ([w:GD54008] * 27) + [w:GD54003];         //Copy Source Offset         [w:TMP0001] = (([w:GD54008] + 1) * 27) + [w:GD54003];          //If no backup data exists in the selected place, do not process         if([w:GD56201[w:TMP0000]] != 0)         {             //Reduce Number of Backup Cases             if(([w:GD56223[w:TMP0000]] != 0) &amp;&amp; ([w:GD56223[w:TMP0000]] == [w:GD56224[w:TMP0000]]))             {</pre>			

```

        if([w:GD56201[w:TMP0000]] == 1)
        {
            //AnyWireASLINK
            if([w:GD54087] > 0)
            {
                [w:GD54087] = [w:GD54087] - 1;
            }else{
                [w:GD54087] = 0;
            }
        }else{
            //CC-LINK
            if([w:GD54088] > 0)
            {
                [w:GD54088] = [w:GD54088] - 1;
            }else{
                [w:GD54088] = 0;
            }
        }
    }

    //Calculate the Number of Processing Counts to Repeat
    [w:TMP0003] = [w:TMP0000] / 27;
    if(([w:GD54075] == 1) || ([w:GD54075] == 2))
    {
        //ASLINK or CC-Link Dedicated Screen
        [w:TMP0002] = 99;
    }else{
        //General Screen
        [w:TMP0002] = 199;
    }

    while([w:TMP0003] <= [w:TMP0002])
    {
        if([w:TMP0003] != [w:TMP0002])
        {
            //Overwrite Data

            bmov([w:GD56200[w:TMP0001]], [w:GD56200[w:TMP0000]], 27);
            //Adjust the No. to the current No. If No. is 0 or
less, fix it to 0.

            if([w:GD56200[w:TMP0000]] > 0)
            {
                [w:GD56200[w:TMP0000]] =
[w:GD56200[w:TMP0000]] - 1;
            }else{
                [w:GD56200[w:TMP0000]] = 0;
            }

            //Add Offset
            [w:TMP0000] = [w:TMP0000] + 27;
            [w:TMP0001] = [w:TMP0001] + 27;

            //Add Count
            [w:TMP0003] = [w:TMP0003] + 1;

```

```

    }else{
        //Set the last data to 0
        [w:TMP0004] = 0;

        if(([w:GD54075] == 1) || ([w:GD54075] == 2))
        {
            //ASLINK or CC-Link Dedicated Screen
            fmov([w:TMP0004], [w:GD58873], 27);
        }else{
            //General Screen
            fmov([w:TMP0004], [w:GD61573], 27);
        }
        [w:TMP0000] = [w:TMP0000] + 27;

        //Add Count
        [w:TMP0003] = [w:TMP0003] + 1;
    }
}

//Save to Recipe
[w:GD54091] = 1;          //Recipe No.30002 Record No.
set([b:GB54019]);        //Recipe No.30002 Read Trigger

}else{
    [w:GD54006] = 32012;
    [w:GD62007] = 30101;
}

}else{
    [w:GD54006] = 32012;
    [w:GD62007] = 30101;
}

}else{

    //All Delete
    [w:TMP0000] = 0;
    fmov([w:TMP0000], [w:GD56200], 5400);

    //Clear Backup Cases
    //AnyWireASLINK
    [w:GD54087] = 0;
    //CC-Link
    [w:GD54088] = 0;

    //Save to Recipe
    [w:GD54091] = 1;          //Recipe No.30002 Record No.
    set([b:GB54019]);        //Recipe No.30002 Read Trigger
}
rst([b:GB54006]);

```

脚本号	30048	脚本名	Script30048
注释	配方触发 OFF		
数据类型	有符号 BIN16	触发类型	ON 中 GD54012. b5

//Turn OFF Read Trigger of Recipe

```
if([w:GD54013] == 30001)
```

<pre> {     rst([b:GB54017]);          //Recipe No.30001 Read Trigger }else{     rst([b:GB54019]);          //Recipe No.30002 Read Trigger }  rst([b:GD54012.b5]); </pre>			
脚本号	30061	脚本名	Script30061
注释	B-30104 恢复前处理		
数据类型	有符号 BIN16	触发类型	ON 中 GB54002
<pre> //Transfer Specified Contents to PLC Devices  //Offset [w:TMP0000] = ([w:GD54008] * 27) + [w:GD54003];  //If target model is 0, no settings are entered and restoration will not be executed. if([w:GD56201[w:TMP0000]] != 0) {     //Do no restore data from the backup that was not ended normally.     if(([w:GD56223[w:TMP0000]]      !=      0)      &amp;&amp;      ([w:GD56223[w:TMP0000]]      == [w:GD56224[w:TMP0000]]))     {         //Setting Name         bmov([w:GD56202[w:TMP0000]], [w:GD54045], 24);          //Target Device Comment Display         [w:GD54016] = [w:GD56201[w:TMP0000]];          //SD1437: iQSS Backup/Restoration Target Model/Execution Unit Setting         [w:SD1437] = (([w:GD56216[w:TMP0000]] &amp; 0x00FF) &lt;&lt; 8) + ([w:GD56201[w:TMP0000]] &amp; 0x00FF);          //SD1438: iQSS Backup/Restoration Target Folder No.         [w:SD1438] = [w:GD56214[w:TMP0000]];          //SD1439: iQSS Backup/Restoration Target Setting (Target Module I/O No.)         [w:SD1439] = [w:GD56215[w:TMP0000]];          //SD1440: iQSS Backup/Restoration Target Setting (Target Device 1 ASLINK: ID No., CC-Link: Station No.)         [w:SD1440] = [w:GD56217[w:TMP0000]];          //SD1441: iQSS Backup/Restoration Target Setting (Target Device 2 ASLINK: Not Used, CC-Link: Sub Station No.)         [w:SD1441] = [w:GD56218[w:TMP0000]];          //SD1444: iQSS Backup/Restoration Action Setting         [w:SD1444] = [w:GD56219[w:TMP0000]];          //If the right to use the iQSS backup is obtained, start backup.         if(([b:GB54004] == ON) &amp;&amp; ([w:SD1446] == 0x0001))         {             set([b:SM1439]);             [w:GD62000] = 30106; </pre>			



<pre>         }else{             [w:GD54006] = 32006;             [w:GD62007] = 30101;         }     }else{         [w:GD54006] = 32013;         [w:GD62007] = 30101;     } } else{     [w:GD54006] = 32008;     [w:GD62007] = 30101; }  rst([b:GB54002]); </pre>			
脚本号	30051	脚本名	Script30051
注释	检查 iQSS 备份/恢复使用权取得信息		
数据类型	有符号 BIN16	触发类型	周期 1 秒
<pre> //Periodically check the status of obtaining the right to use. if([w:SD1435] == [w:SD1436]) {     set([b:GB54004]);          //iQSS Backup Permission Bit ON }else{     rst([b:GB54004]);          //iQSS Backup Permission Bit OFF } </pre>			
脚本号	30059	脚本名	Script30059
注释	画面切换时处理		
数据类型	有符号 BIN16	触发类型	关闭画面时
<pre> //Clear Flags  rst([b:GB54008]); rst([b:GB54009]);  //Close Overlap Window [w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0; </pre>			

#### 基本画面 30105

脚本号	30066	脚本名	Script30066
注释	B-30105 注释显示控制		
数据类型	有符号 BIN16	触发类型	通常
<pre> //Switch Comments Between CC-Link and ASLINK  switch([w:GD54044]) {     //ASLINK     case 1: [w:GD54070] = 226;          //ID or Station             [w:GD54071] = 232;          //All IDs or All Stations             [w:GD54072] = 233;          //ID Specification or Station Specification             [w:GD54073] = 285;          //Not Specified or Station Sub             [w:GD54074] = 0;            //Not Specified or Station Sub Specified             break; </pre>			

```

//CC-Link
case 2: [w:GD54070] = 225;      //ID or Station
        [w:GD54071] = 242;      //All IDs or All Stations
        [w:GD54072] = 243;      //ID Specification or Station Specification
        [w:GD54073] = 227;      //Not Specified or Station Sub
        [w:GD54074] = 244;      //Not Specified or Station Sub Specified
        break;
}

```

#### 基本画面 30106

脚本号	30055	脚本名	Script30055
注释	B-30103、30106 画面启动时动作		
数据类型	有符号 BIN16	触发类型	ON 中 GB40

```
//Make various settings when starting screen
```

```
//Actions at Screen Startup Only
```

```
if([b:GB54008] == OFF)
```

```
{
```

```
    //Backup Execution Date/Time
```

```
    [w:GD54040] = [w:GS650];
```

```
    [w:GD54041] = [w:GS651];
```

```
    [w:GD54042] = [w:GS652];
```

```
    set([b:GB54008]);
```

```
}
```

脚本号	30044	脚本名	Script30044
注释	B-30106 正常结束		
数据类型	有符号 BIN16	触发类型	上升沿 SM1440

```
//iQSS Restoration Normal End Processing
```

```
if([b:SM1442] == ON)
```

```
{
```

```
    [w:GD54006] = 32005;    //iQSS Restoration Cancel
```

```
}else{
```

```
    [w:GD54006] = 32003;    //iQSS Restoration Normal Completion
```

```
}
```

```
[w:GD62007]=30101;    //Notification Dialog Display
```

脚本号	30045	脚本名	Script30045
注释	B-30106 异常结束		
数据类型	有符号 BIN16	触发类型	ON 中 SM1441

```
//iQSS Restoration Abnormal End Processing
```

```
if([b:GB54009] == OFF)
```

```
{
```

```
    if([w:SD1452] != 0)
```

```
    {
```

```
        if([w:SD1452] == 16894)
```

```
        {
```

```
            [w:GD54006] = 32004; //Restoration Failure Message
```

```
            [w:GD62007] = 30101; //Notification Dialog
```

```
        }else{
```

```
            [w:GD54006] = [w:SD1452]; //Restoration Failure Message
```

```
            [w:GD62007] = 30101; //Notification Dialog
```

```
        }
```

```
}
```

<pre> }else{     if([w:SD1453] != 0)     {         [w:GD54006] = [w:SD1453]; //Restoration Failure Message         [w:GD62007] = 30101; //Notification Dialog     }else{         [w:GD54006] = 32004; //Restoration Failure Message         [w:GD62007] = 30101; //Notification Dialog     } }  set([b:GB54009]); } </pre>			
脚本号	30065	脚本名	Script30065
注释	B-30103、30106 iQSS 备份/恢复中止		
数据类型	有符号 BIN16	触发类型	ON 中 GB54002
<pre> //Abort iQSS Backup or iQSS Restoration set([b:SM1442]);          //iQSS Backup/Restoration Cancel Request  rst([b:GB54002]); </pre>			
脚本号	30059	脚本名	Script30059
注释	画面切换时处理		
数据类型	有符号 BIN16	触发类型	关闭画面时
<pre> //Clear Flags  rst([b:GB54008]); rst([b:GB54009]);  //Close Overlap Window [w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0; </pre>			

### 5.9.3 对象脚本

#### 基本画面 30006～30019

对象	字指示灯 *1		
脚本用户 ID	1		
数据类型	无符号 BIN16	触发类型	通常
<pre> //Display ON/OFF Status [w:GD32000] = \$\$ &amp; [w:GD31990]; </pre>			

\*1: 设置在 I/O 的 ON/OFF 的字指示灯。I/O 点数不同的话，软元件也会不同。

#### 基本画面 30013、基本画面 30018

对象	液位 *1		
脚本用户 ID	2		
数据类型	有符号 BIN16	触发类型	通常
<pre> //Display ON/OFF Threshold Value screen_draw(); redraw_object();  [w:TMP0200] = (object_width / 100) * [w:GD53605]; [w:TMP0201] = (object_width / 100) * [w:GD53606]; </pre>			

d_line([w:TMP0200], 0, [w:TMP0200], object_height, 0, 1, 224);	//OFF to ON Threshold Value
d_line([w:TMP0201], 0, [w:TMP0201], object_height, 0, 1, 3);	//ON to OFF Threshold Value

\*1: 使用了[对象脚本符号]。关于[对象脚本符号]，请参照「5.9.5 对象脚本符号」。

#### 基本画面 30010～30012、基本画面 30014、基本画面 30015、基本画面 30017

对象	液位 *1		
脚本用户 ID	2		
数据类型	有符号 BIN16	触发类型	通常
<pre>//Display Threshold Value, Alarm Judgment(Hi), (Lo) screen_draw(); redraw_object();  [w:TMP0200] = (object_width / 100) * [w:GD53605]; [w:TMP0201] = (object_width / 100) * [w:GD53607]; [w:TMP0202] = (object_width / 100) * [w:GD53608];  d_line([w:TMP0200], 0, [w:TMP0200], object_height, 0, 1, 131); //Threshold Value d_line([w:TMP0201], 0, [w:TMP0201], object_height, 0, 1, 224); //Alarm Judgment (Hi) d_line([w:TMP0202], 0, [w:TMP0202], object_height, 0, 1, 3); //Alarm Judgment (Lo)</pre>			

\*1: 使用了[对象脚本符号]。关于[对象脚本符号]，请参照「5.9.5 对象脚本符号」。

#### 基本画面 30010～30015、基本画面 30017、基本画面 30018

对象	数值输入 *1		
脚本用户 ID	3		
数据类型	无符号 BIN16	触发类型	输入确定时
<pre>//When input is determined, redraw the screen. redraw_screen();</pre>			

\*1: 设置在临界值、报警判断(Hi)、报警判断(Lo)、OFF→ON 的临界值、ON→OFF 的临界值。

#### 基本画面 30101

对象	字指示灯		
脚本用户 ID	1		
数据类型	有符号 BIN16	触发类型	通常
<pre>//Script to Display Comments if([w:GD54101[w:GD54001]] != 0) {     [w:GD54076] = (([w:GD54101[w:GD54001]] * 10) + \$\$) + 221; }else{     [w:GD54076] = 0; }</pre>			

上述脚本设置在「执行单位」的所有字指示灯。但是软元件因对象而异。

## 窗口画面 30001

对象	数值显示		
脚本用户 ID	1		
数据类型	无符号 BIN16	触发类型	上升沿 GB40
<pre>//Obtain Today's Year &amp; Month from Clock Data [w:TMP950] = [w:GS650] &amp; 0xF000;//Obtain Tenths Digit of "Last 2-Digits of Year" from Clock Data for Setting [w:TMP960] = [w:TMP950] &gt;&gt; 12;//Decimal Alignment [w:TMP968] = [w:TMP960] * 10;//BCD-&gt;BIN [w:TMP951] = [w:GS650] &amp; 0x0F00;//Obtain Ones Digit of "Last 2-Digits of Year" from Clock Data for Setting [w:TMP961] = [w:TMP951] &gt;&gt; 8;//BCD-&gt;BIN [w:TMP973] = 2000 + [w:TMP968] + [w:TMP961];//Set Year to TMP973 as BIN [w:GD63990] = [w:TMP973];//Set Year  [w:TMP952] = [w:GS650] &amp; 0x00F0;//Obtain Tenths Digit of Month from Clock Data for Setting [w:TMP962] = [w:TMP952] &gt;&gt; 4;//Decimal Alignment [w:TMP969] = [w:TMP962] * 10;//BCD-&gt;BIN [w:TMP953] = [w:GS650] &amp; 0x000F;//Obtain Ones Digit of Month from Clock Data for Setting [w:TMP974] = [w:TMP969] + [w:TMP953];//Set Month to TMP974 as BIN [w:GD63991] = [w:TMP974];//Set Month  [w:TMP954] = [w:GS651] &amp; 0xF000;//Obtain Tenths Digit of "Last 2-Digits of Day" from Clock Data for Setting [w:TMP963] = [w:TMP954] &gt;&gt; 12;//Decimal Alignment [w:TMP970] = [w:TMP963] * 10;//BCD-&gt;BIN [w:TMP955] = [w:GS651] &amp; 0x0F00;//Obtain Ones Digit of "Last 2-Digits of Day" from Clock Data for Setting [w:TMP964] = [w:TMP955] &gt;&gt; 8;//BCD-&gt;BIN [w:TMP975] = [w:TMP970] + [w:TMP964];//Set Day to TMP975 as BIN [w:GD63992] = [w:TMP975];//Set Day  [w:TMP956] = [w:GS651] &amp; 0x00F0;//Obtain Tenths Digit of Hour from Clock Data for Setting [w:TMP965] = [w:TMP956] &gt;&gt; 4;//Decimal Alignment [w:TMP971] = [w:TMP965] * 10;//BCD-&gt;BIN [w:TMP957] = [w:GS651] &amp; 0x000F;//Obtain Ones Digit of Hour from Clock Data for Setting [w:TMP976] = [w:TMP971] + [w:TMP957];//Set Hour to TMP976 as BIN [w:GD63993] = [w:TMP976];//Set Hour  [w:TMP958] = [w:GS652] &amp; 0xF000;//Obtain Tenths Digit of "Last 2-Digits of Minute" from Clock Data for Setting [w:TMP966] = [w:TMP958] &gt;&gt; 12;//Decimal Alignment [w:TMP972] = [w:TMP966] * 10;//BCD-&gt;BIN [w:TMP959] = [w:GS652] &amp; 0x0F00;//Obtain Ones Digit of "Last 2-Digits of Minute" from Clock Data for Setting [w:TMP967] = [w:TMP959] &gt;&gt; 8;//BCD-&gt;BIN [w:TMP977] = [w:TMP972] + [w:TMP967];//Set Minute to TMP977 as BIN [w:GD63994] = [w:TMP977];//Set Minute  [w:TMP993] = [w:GS652] &amp; 0x00F0;//Obtain Tenths Digit of Second from Clock Data for Setting [w:TMP995] = [w:TMP993] &gt;&gt; 4;//Decimal Alignment [w:TMP996] = [w:TMP995] * 10;//BCD-&gt;BIN [w:TMP994] = [w:GS652] &amp; 0x000F;//Obtain Ones Digit of Second from Clock Data for Setting</pre>			

[w:TMP978] = [w:TMP996] + [w:TMP994]; //Set Second to TMP978 as BIN [w:GD63995] = [w:TMP978]; //Set Second			
对象	数值显示		
脚本用户 ID	2		
数据类型	无符号 BIN16	触发类型	通常
//BIN -> BCD Conversion			
[w:TMP979] = [w:GD63990] - 2000;                    //Last 2-Digits of Year			
[w:TMP980] = (([w:TMP979] / 10) << 4) + ([w:TMP979] % 10);                    //Year BIN -> BCD			
[w:TMP981] = (([w:GD63991] / 10) << 4) + ([w:GD63991] % 10);                    //Month BIN -> BCD			
[w:TMP982] = (([w:GD63992] / 10) << 4) + ([w:GD63992] % 10);                    //Day BIN -> BCD			
[w:TMP983] = (([w:GD63993] / 10) << 4) + ([w:GD63993] % 10);                    //Hour BIN -> BCD			
[w:TMP984] = (([w:GD63994] / 10) << 4) + ([w:GD63994] % 10);                    //Minute BIN -> BCD			
[w:TMP985] = (([w:GD63995] / 10) << 4) + ([w:GD63995] % 10);                    //Second BIN -> BCD			
对象	数值显示		
脚本用户 ID	3		
数据类型	无符号 BIN16	触发类型	通常
//Year & Month Setting			
[w:GS513] = ([w:TMP980] << 8) + [w:TMP981];                    //Set Year & Month to Change Time Device			
对象	数值显示		
脚本用户 ID	4		
数据类型	无符号 BIN16	触发类型	通常
//Date & Time Setting			
[w:GS514] = ([w:TMP982] << 8) + [w:TMP983];                    //Set Date & Time to Change Time Device			
对象	数值显示		
脚本用户 ID	5		
数据类型	无符号 BIN16	触发类型	通常
//Minute & Second Setting			
[w:GS515] = ([w:TMP984] << 8) + [w:TMP985];                    //Set Minute & Second to Change Time Device			
对象	数值显示		
脚本用户 ID	6		
数据类型	无符号 BIN16	触发类型	通常
//Day of Week Setting			
[w:TMP986] = [w:GD63990];                    //Year (BIN)			
[w:TMP987] = [w:GD63991];                    //Month (BIN)			
[w:TMP988] = [w:GD63992];                    //Day (BIN)			
if(([w:TMP987] == 1)    ([w:TMP987] == 2)) { // Correction Processing to Calculate January and February as 13th/14th Month			
[w:TMP986] = [w:TMP986] - 1; //Subtract 1 from Year			
[w:TMP987] = [w:TMP987] + 12; //Add 12 to Month			
}			
[w:TMP989] = [w:TMP986]/4; //Create Items Required for Zeller's Congruence			
[w:TMP990] = [w:TMP986]/100; //Create Items Required for Zeller's Congruence			
[w:TMP991] = [w:TMP986]/400; //Create Items Required for Zeller's Congruence			
[w:TMP992] = (13*[w:TMP987]+8)/5; //Create Items Required for Zeller's Congruence			
//Calculate Day of Week Using Zeller's Congruence and Set the Day to Change Time Device			
[w:GS516] = ([w:TMP986]+[w:TMP989]-[w:TMP990]+[w:TMP991]+[w:TMP992]+[w:TMP988])%7;			

**5.9.4 脚本符号**

符号名	软元件及常数	备注
Input_X	42	输入值区域的左上角 X 坐标
Input_Y	155	输入值区域的左上角 Y 坐标
Frame_X	618	I/O 显示区域的右下角 X 坐标
Frame_Y	427	I/O 显示区域的右下角 Y 坐标
Object_X	36	字指示灯的宽度
Object_Y	17	字指示灯的高度

**5.9.5 对象脚本符号**

符号名	软元件及常数	备注
object_width	124	液位的宽度
object_height	25	液位的高度

6. 模板

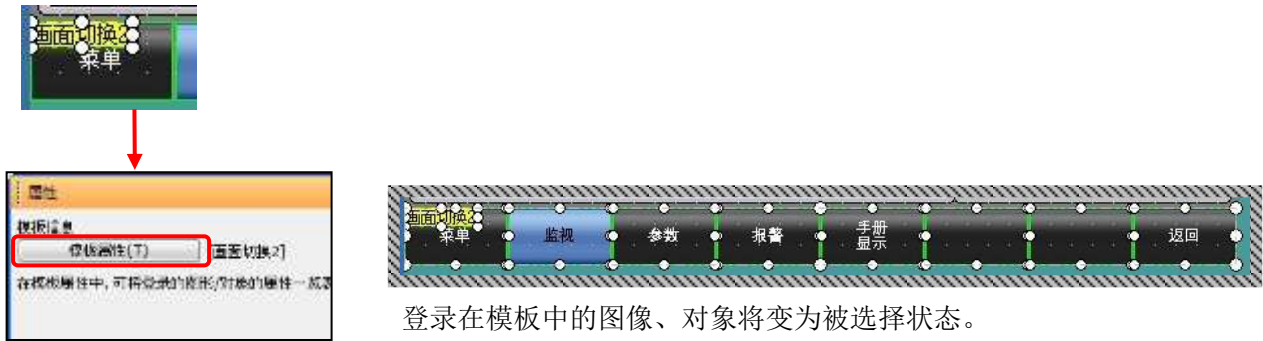
模板是指图形和对象的集合体。因为将相关设置都作为模板的属性一并进行了登录，所以可以简单地统一更改软元件、颜色等。关于更改属性设置值的详细内容，请参照「GT Designer3 (GOT2000) 帮助」。



模板信息仅在画面设计的编辑画面上显示，不会显示在 GOT 的显示画面中。

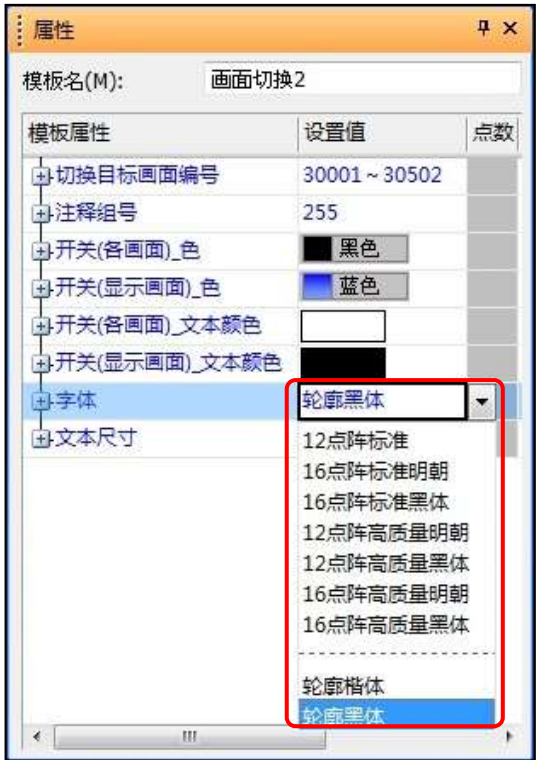
例：更改字体时

- (1) 选择[模板信息]，点击[模板属性] (或双击[模板信息])



登录在模板中的图像、对象将变为被选择状态。

- (2) 点击[字体]的设置值，选择要更改的字体





## 7.1 关于用户定义名称的登录

- 输出模块: 请把注释号设置为模块的 ID+1 的号。(注释号无法设置为 0, 因此将 ID 加 1 后的值作为注释号。)

输出模块 ID14~17 的  
注释号

- 输入模块 ID14~17 的  
注释号

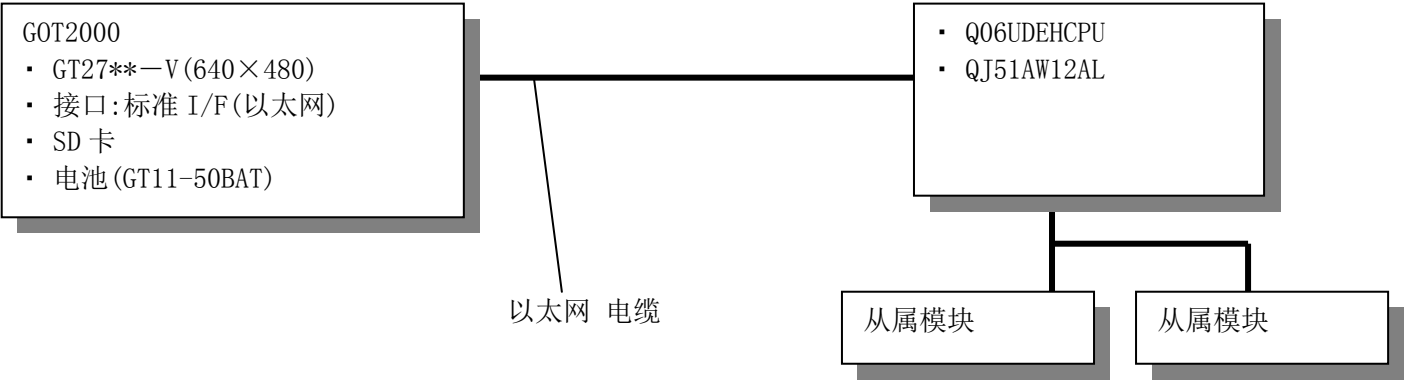
在 GOT 启动状态下对从属模块进行添加、删除、更改 ID 后，请返回基本画面 B-30001 并触摸画面右上角的「传感器自动识别」开关 1 秒以上，或者重新启动 GOT。



7.3 关于 CPU 的更改

使用 MELSEC-Q 系列时，请更改系统构成和以太网设置。

7.3.1 系统构成



7.3.2 画面设计软件的以太网设置

	本站	网络号	站号	机器	IP 地址	端口号	通讯方式
1	*	1	1	QnUD (P) V/QnUDEH	192. 168. 3. 39	5006	UDP

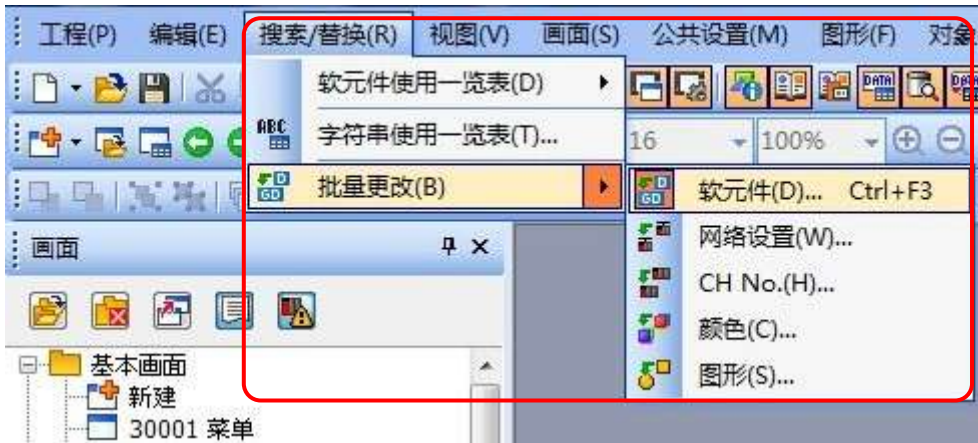
7.4 起始输入输出号的更改

将模块的起始输入输出号更改到 30H 以外时，请按照以下步骤进行更改。  
(例：起始输入输出号从 30H 更改到 50H 时)

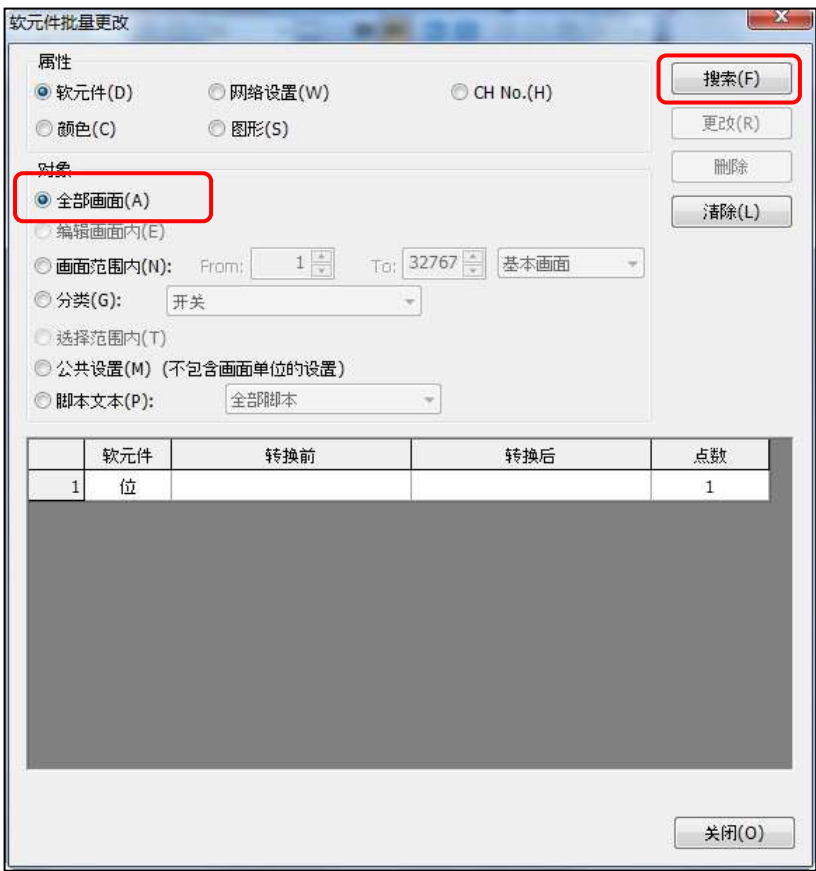
7.4.1 画面中设置的软元件的更改

更改画面中设置的软元件时，将软元件批量更改的对象设置为[全部画面]。  
选择[全部画面]后更改的软元件：U03-G0~U03-G10321、X0041、Y0030~Y0041。

(1) 选择菜单[搜索/替换]-[批量更改]-[软元件]。



(2) 弹出设置对话框，选择对象为[全部画面]，点击[搜索]。



(3) 设置[转换后]的软元件、[点数]，执行批量更改。

- 缓冲存储器的起始输入输出号的更改

设置[转换前]U03-G0、[转换后]U05-G0、[点数]10322，点击[更改]。U03-G0~U03-G10321 将更改到 U05-G0~U05-G10321。



- 输入输出信号的起始输入输出号的更改

更改输入信号(X 软元件)：设置[转换前]X0041、[转换后]X0061、[点数]1，点击[更改]。X0041 将更改到 X0061。更改输出信号(Y 软元件)：设置[转换前]Y0030、[转换后]Y0050、[点数]18，点击[更改]。Y0030~Y0041 将更改到 Y0050~Y0061。



7.4.2 [公共设置]中设置的软元件的更改

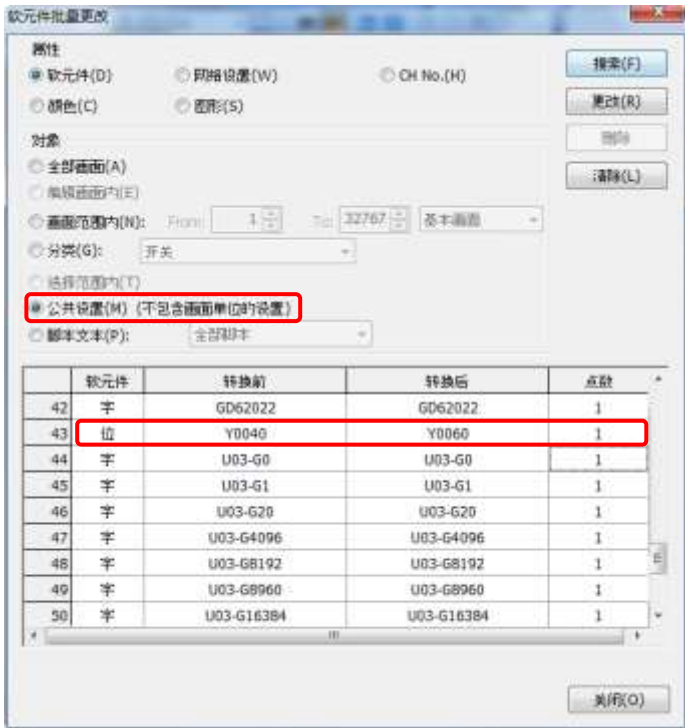
更改[GOT 环境设置]、配方功能等各种功能中设置的软元件时，软元件批量更改的对象设置为[公共设置]。请按照与 7. 4. 1 同样的步骤执行。

选择[公共设置]后更改的软元件：U03-G0~U03-G12288、Y0040。

- 缓冲存储器的起始输入输出号的更改  
设置[转换前]U03-G0、[转换后]U05-G0、[点数]12289，点击[更改]。U03-G0~U03-G12288 将更改到 U05-G0~U05-G12288。



- 输入输出信号的起始输入输出号的更改  
设置[转换前]Y0040、[转换后]Y0060、[点数]1，点击[更改]。Y0040 将更改到 Y0060。

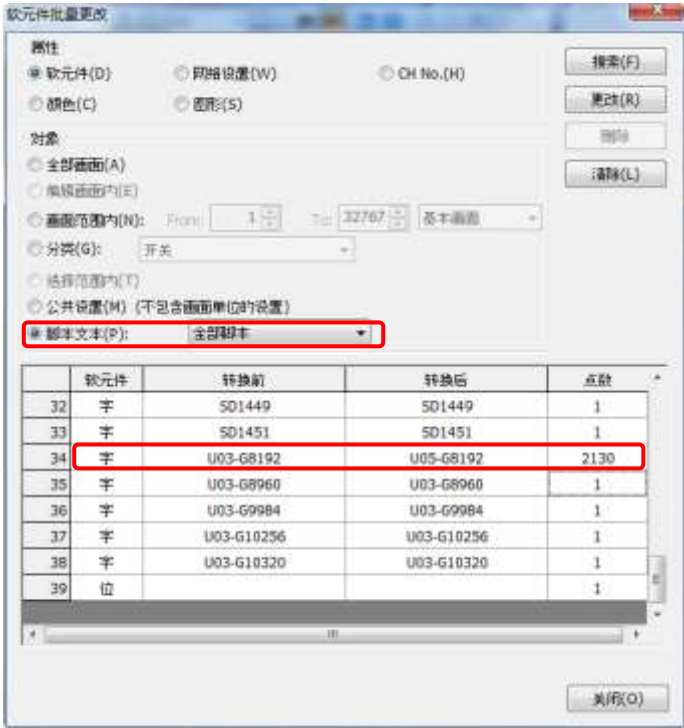


7.4.3 [脚本文本]中设置的软元件的更改

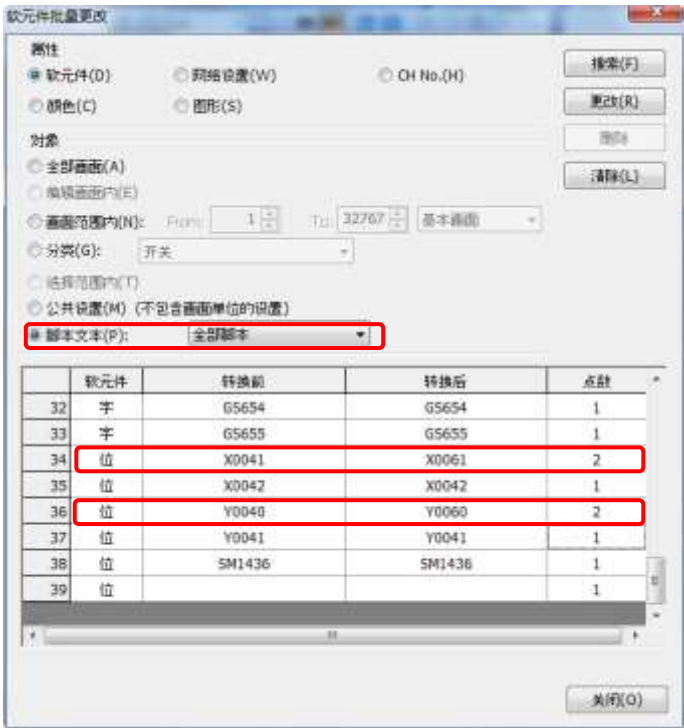
更改[脚本文本]中设置的软元件时，软元件批量更改的对象设置为[脚本文本]。请按照与 7.4.1 同样的步骤执行。

选择[脚本文本]后更改的软元件：U03-G8192~U03-G10321、X0041~X0042、Y0040~Y0041。

- 缓冲存储器的起始输入输出号的更改  
设置[转换前]U03-G8192、[转换后]U05-G8192、[点数]2130，点击[更改]。U03-G8192~U03-G10321 将更改到 U05-G8192~U05-G10321。

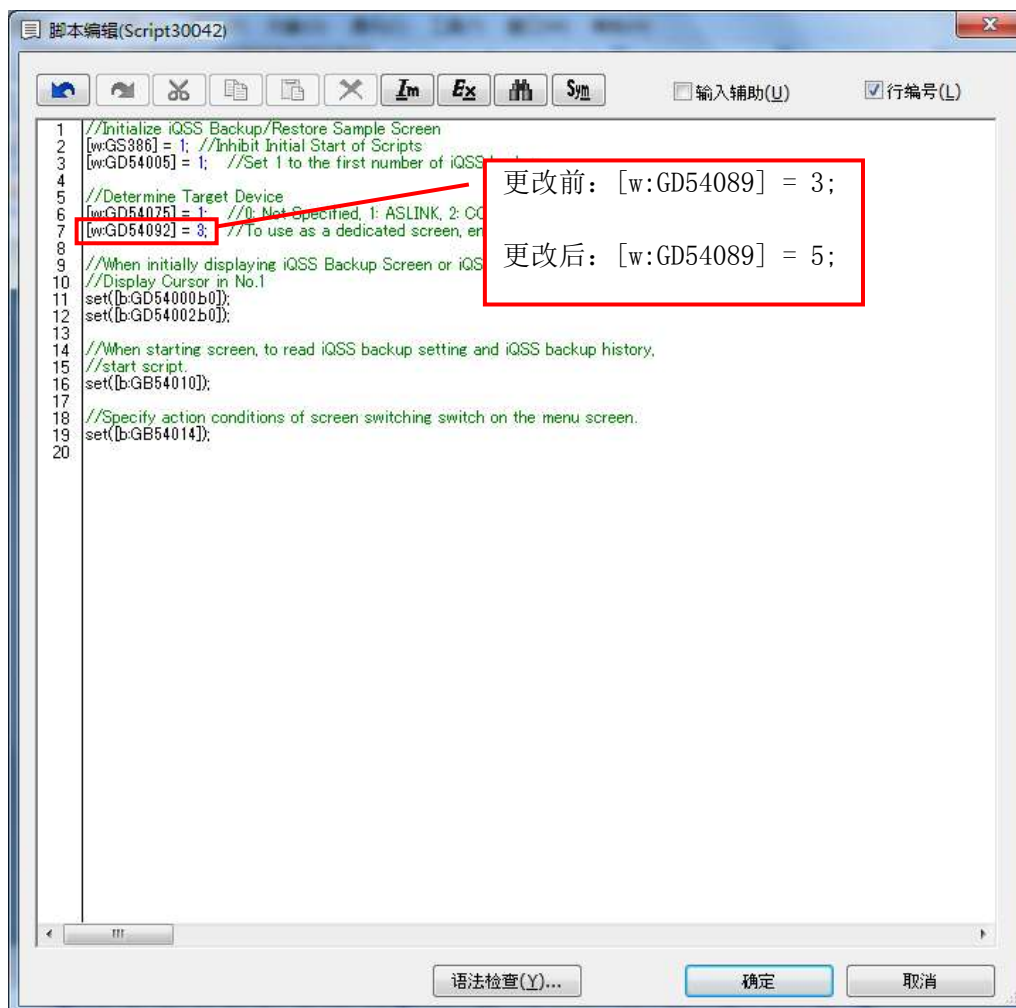


- 输入输出信号的起始输入输出号的更改  
如要更改输入信号(X 软元件)，请设置[转换前]X0041、[转换后]X0061、[点数]2，点击 [更改]。X0041~X0042 将更改到 X0061~X0062。如要更改输出信号(Y 软元件)，请设置[转换前]Y0040、[转换后]Y0060、[点数]2，点击[更改]。Y0040~Y0041 将更改到 Y0060~Y0061。



#### 7.4.4 工程脚本的更改

工程脚本号 30042 中指定了起始输入输出号。请按照以下方法进行修改。

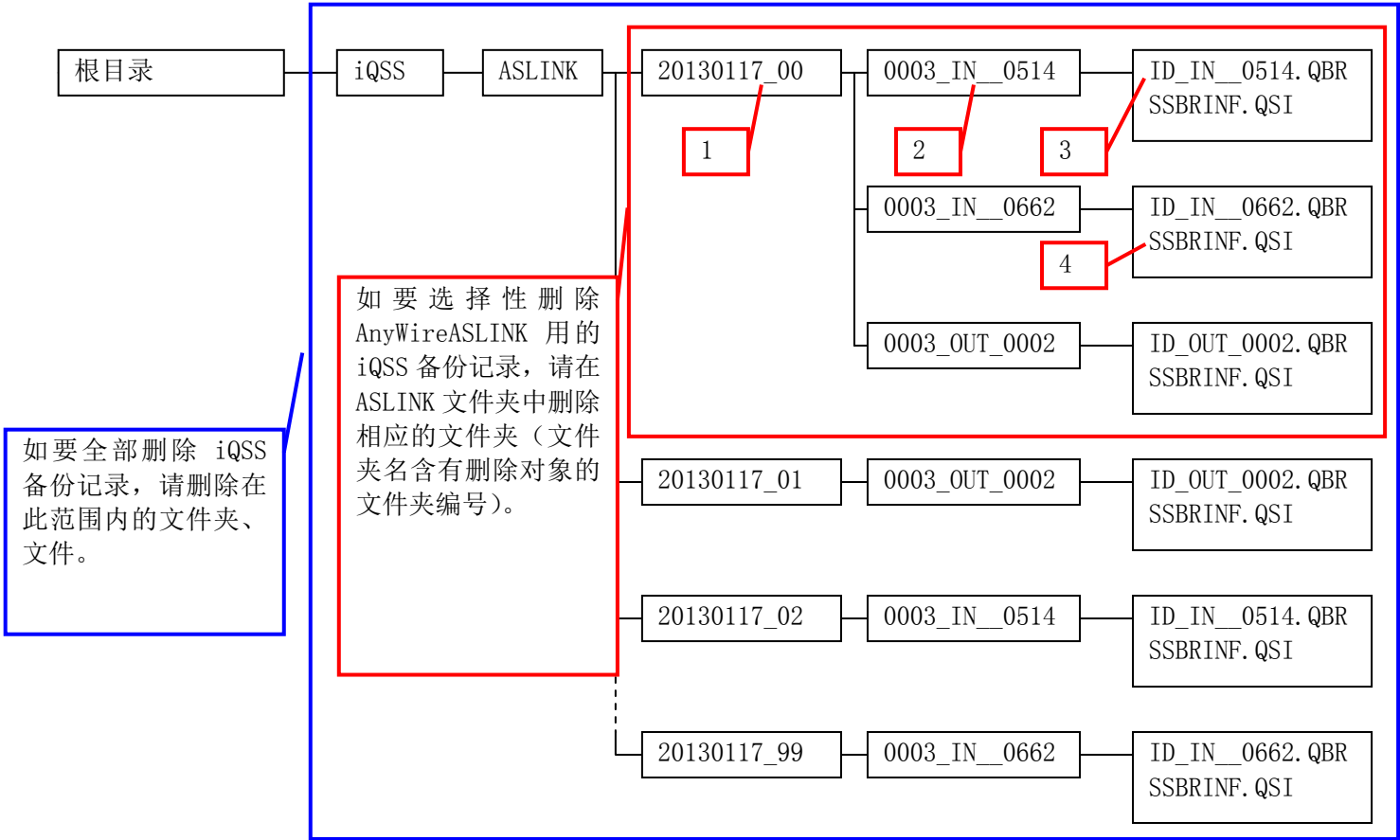




7.5 iQSS 备份文件夹构成

7.5.1 文件夹构成

安装在可编程控制器上的 SD 卡内的 iQSS 备份文件夹构成，如下所示。关于表中的 1~4，请参照「7.5.2 文件夹名、文件名详细」。



7.5.2 文件夹名、文件名详细

1. 20130117\_00 . . . 备份文件夹名(日期、文件夹编号)  
    2013 年份  
    01 月份  
    17 日期  
    00 编号(2 位(00~99(10 进制)))  
    备份日期(YYYYMMDD)
2. 0003\_IN\_0514 . . . 备份文件夹名(I/O 编号、ID)  
    0003 I/O 编号(4 位(16 进制))  
    IN\_ 输入/混合从属模块  
    OUT\_ 输出从属模块  
    0514 ID 编号(4 位(10 进制)) \*1
- 3 ID\_IN\_0514.QBR . . . 备份文件名  
    ID IN\_ 0514 ID 编号(4 位(10 进制)) \*1  
    IN\_ 输入/混合从属模块  
    OUT\_ 输出从属模块
4. SSBRINF.QSI . . . 系统文件

\*1: AnyWireASLINK 的输入/混合从属模块的 ID 号，是按照分配给模块本体的 ID 号+512 后的号进行管理的。因此，输入/混合从属模块用的备份文件夹名的末尾附加了管理号 512~766。