

三菱电机株式会社 AnyWireASLINK 主模块

QJ51AW12AL (MELSEC-Q)

LJ51AW12AL (MELSEC-L)

GOT 功能样本

AnyWireASLINK 网络监视功能说明书

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) 只有正在使用本公司产品或有意使用本公司产品的用户才能使用。
- (2) 本公司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。
请用户自行承担责任。
- (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以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。

目录

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 画面创建软件的连接机器设置	7
3.3 画面创建软件的以太网设置	7
3.4 画面创建软件的重叠窗口设定	7
4. 关于 AnyWireASLINK 主模块.....	7
4.1 缓冲存储器的起始 I/O 号	7
5. 画面规格	8
5.1 显示语言	8
5.2 画面一览表/切换	8
5.3 画面说明	14
5.3.1 AnyWireASLINK 主模块选择(B-30000).....	14
5.3.2 AnyWireASLINK 菜单(B-30001).....	15
5.3.3 AnyWireASLINK 系统图(B-30002).....	16
5.3.4 输入信号监视(B-30003)	17
5.3.5 输出信号监视(B-30004)	18
5.3.6 传感器一览(B-30005)	19
5.3.7 详细信息(电缆连接类型(I/O 非绝缘))(B-30006).....	20
5.3.8 详细信息(电缆连接类型(I/O 绝缘))(B-30007).....	21
5.3.9 详细信息(I/O 防水连接器连接类型)(B-30008).....	22
5.3.10 详细信息(传输 • I/O 防水连接器连接类型)(B-30009).....	23
5.3.11 详细信息(光电放大器模块(CAM, CAS))(B-30010)	24
5.3.12 详细信息(接近放大器模块)(B-30011)	25
5.3.13 详细信息(光纤放大器模块(CAM, CAS))(B-30012)	26
5.3.14 详细信息(气缸开关)(B-30013)	27
5.3.15 详细信息(光电传感器)(B-30014)	28
5.3.16 详细信息(光电传感器(透过型受光))(B-30015)	29
5.3.17 详细信息(光电传感器(透过型投光))(B-30016)	30
5.3.18 详细信息(接近开关)(B-30017)	31
5.3.19 详细信息(ASLINK 终端(驱动部))(B-30018).....	32
5.3.20 详细信息(光纤放大器模块(投光专用))(B-30019)	33
5.3.21 详细信息(压力传感器(正压 1 点))(B-30020).....	34
5.3.22 详细信息(压力传感器(正压 2 点))(B-30021).....	35
5.3.23 详细信息(压力传感器(正压 16 点))(B-30023).....	36
5.3.24 详细信息(压力传感器(负压 1 点))(B-30024).....	37
5.3.25 详细信息(压力传感器(负压 2 点))(B-30025).....	38
5.3.26 详细信息(压力传感器(负压 16 点))(B-30027).....	39
5.3.27 详细信息(压力传感器(复合压力 1 点))(B-30028).....	40
5.3.28 详细信息(压力传感器(复合压力 2 点))(B-30029).....	41
5.3.29 详细信息(压力传感器(复合压力 16 点))(B-30031).....	42
5.3.30 详细信息(光遮断器)(B-30032)	43
5.3.31 详细信息(映射传感器(梳型主机))(B-30033)	44

5.3.32	详细信息(映射传感器(梳型子机))(B-30034)	45
5.3.33	详细信息(映射传感器(透过型受光))(B-30035)	46
5.3.34	详细信息(映射传感器(透过型投光))(B-30036)	47
5.3.35	详细信息(一体型(e-con 连接器))(B-30037)	48
5.3.36	详细信息(一体型(JST 连接器))(B-30038)	49
5.3.37	详细信息(一体型(MOLEX 连接器))(B-30039)	50
5.3.38	详细信息(多通驱动)(B-30040)	51
5.3.39	详细信息(一体型继电器输出(驱动部))(B-30041)	52
5.3.40	详细信息(一体型继电器输出)(B-30042)	53
5.3.41	详细信息(手持遥控(开关侧))(B-30051)	54
5.3.42	详细信息(手持遥控(LED 侧))(B-30052)	55
5.3.43	详细信息(压力传感器(正压 2 点 有报警))(B-30053)	56
5.3.44	详细信息(压力传感器(正压 3 点 有报警))(B-30054)	57
5.3.45	详细信息(压力传感器(负压 2 点 有报警))(B-30057)	58
5.3.46	详细信息(压力传感器(负压 3 点 有报警))(B-30058)	59
5.3.47	详细信息(压力传感器(复合压力 2 点 有报警))(B-30061)	60
5.3.48	详细信息(压力传感器(复合压力 3 点 有报警))(B-30062)	61
5.3.49	iQSS 菜单(B-30100)	62
5.3.50	iQSS 备份(B-30101)	63
5.3.51	iQSS 备份设置(B-30102)	64
5.3.52	iQSS 备份进展(B-30103)	65
5.3.53	iQSS 恢复(B-30104)	66
5.3.54	iQSS 恢复设置(B-30105)	67
5.3.55	iQSS 恢复进展(B-30106)	68
5.3.56	报警复位(W-30001)	69
5.3.57	语言设置(W-30002)	70
5.3.58	时钟设置(W-30003)	71
5.3.59	参数访问错误(W-30004)	72
5.3.60	参数访问中(W-30005)	73
5.3.61	零点补正执行消息(W-30006)	74
5.3.62	报警位切换消息(W-30007)	75
5.3.63	数据删除确认对话框(W-30100)	76
5.3.64	通知对话框(W-30101)	77
5.3.65	执行确认对话框(W-30102)	78
5.4	从属模块详细信息画面对照表	79
5.5	使用软元件一览表	81
5.6	注释一览表	88
5.7	软元件数据传送一览表	93
5.8	配方一览表	113
5.9	脚本一览表	115
6.	其他	224
6.1	关于用户定义名称的登录	224
6.2	关于系统构成的更改	224
6.3	关于 CPU 的更改	225
6.4	关于起始输入输出号的更改	226
6.5	关于 iQSS 备份文件夹构成	233
6.6	关于主模块的更改	234

修订记录

样本画面说明书

修订日期	管理号*	修订内容
2013/9	BCN-P5999-0139	初版
2015/1	BCN-P5999-0139-2	修改所有的格式 「1. 概要」修改 「2. 系统构成」修改 「4. 关于 AnyWireASLINK 主模块」修改 「5.2.1 画面一览表/切换(公共)」修改 「5.2.2 画面一览表/切换(个别)」修改 「5.3 画面说明」修改 「5.4 从属模块详细信息画面对照表」修改 「5.5 使用软元件一览表」修改 「5.6 注释一览表」修改 「5.7 软元件数据传送一览表」修改 「5.9 脚本一览表」修改 「6. 模板」删除 「6.2 关于系统构成的更改」修改 「6.4 关于起始输入输出号的更改」修改 「6.4.1 [软元件批量更改更改前头输入输出编号时的想法] 追加 「6.4.2 画面中设置的软元件的更改」修改 「6.4.3 [公共设置]中设置的软元件的更改」修改 「6.4.4 [脚本文本]中使用的软元件的更改」修改 「6.4.5 [脚本符号]的更改」追加 「6.4.6 开关的更改」追加 「6.4.7 [缓冲存储器模块号切换]的更改」追加 「6.6 关于主模块的更改」追加

* 管理编号记载在右下方。

工程数据

修订日期	工程数据	GT Designer3*	修订内容
2013/9	AnyWireASLINK_V_Ver1_C.GTX	1.100E	初版
2015/1	AnyWireASLINK_V_Ver2_C.GTX	1.123D	主模块切换对应 对应从属模块追加 模板删除

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

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,最多可连接 4 台 AnyWireASLINK 主模块,对 AnyWireASLINK 主模块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。连接的主模块在 3 台以下时, 需要修改部分数据。关于具体修改部分, 请参照「6.6 关于主模块的台数更改」。

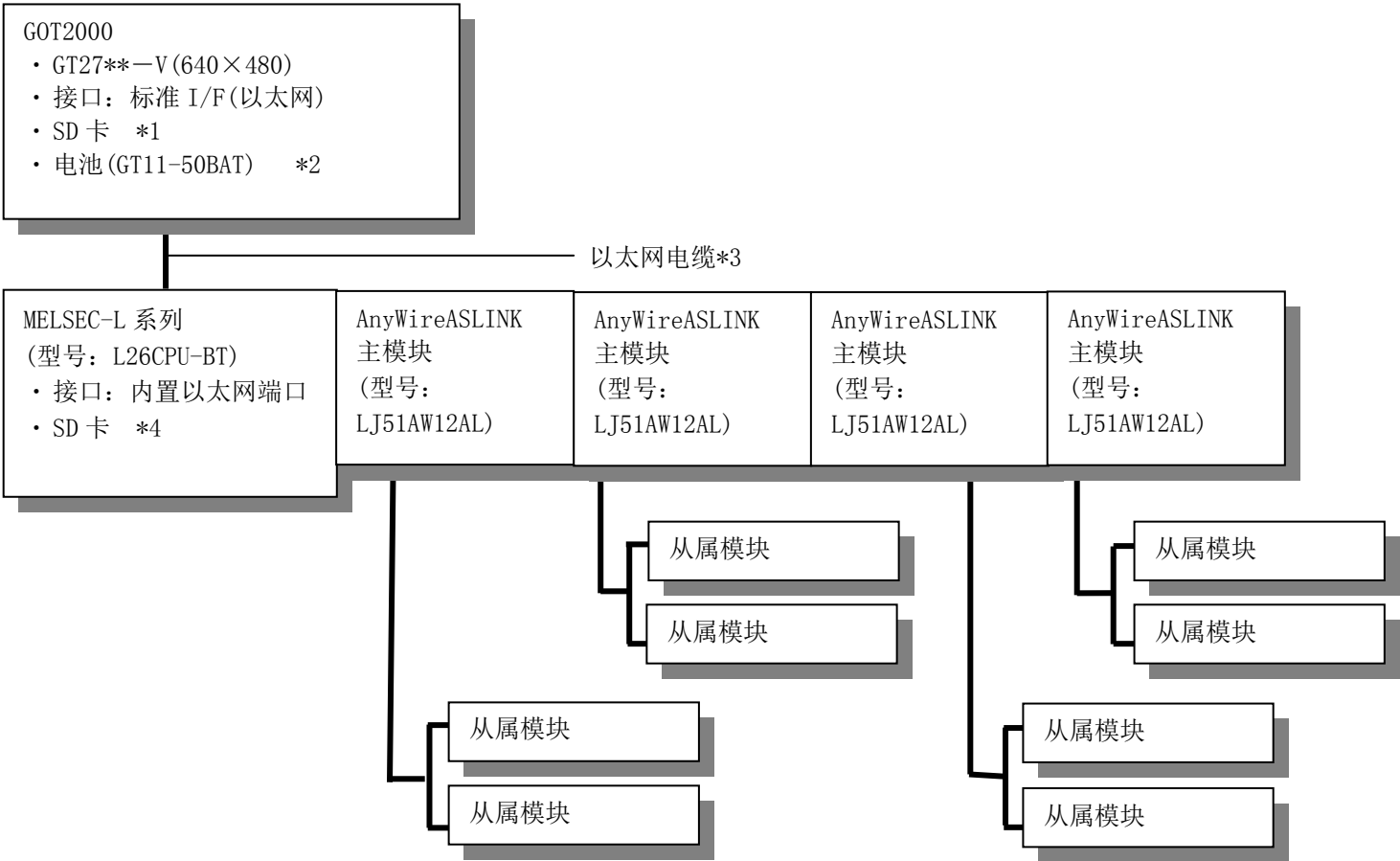
此样本画面是为连接 MELSEC-L 系列而制作的。连接 MELSEC-Q 系列时, 需要修改部分数据。关于具体修改部分, 请参照「6.3 关于 CPU 的更改」。而且 MELSEC-Q 系列不支持从属模块的参数的备份/恢复。

<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	
	轮廓字体	中文(简体)
		英数假名
		日语汉字
	黑体	中文(简体)汉字
		软元件数据传送

3.2 画面创建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 标准以太网设置

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 画面创建软件的以太网设置

	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

3.4 画面创建软件的重叠窗口设定

要关闭基本画面切换时窗口画面,在[画面切换/窗口]的重叠窗口的[详细设置]中把[切换基本画面的同时关闭窗口]设定为有效。

4. 关于 AnyWireASLINK 主模块

4.1 缓冲存储器的起始 I/O 号

设置缓冲存储器的起始 I/O 号为 0、2、4、6。要更改缓冲存储器的起始 I/O 号,请参照「6.4 关于起始输入输出号的更改」。

5. 画面规格

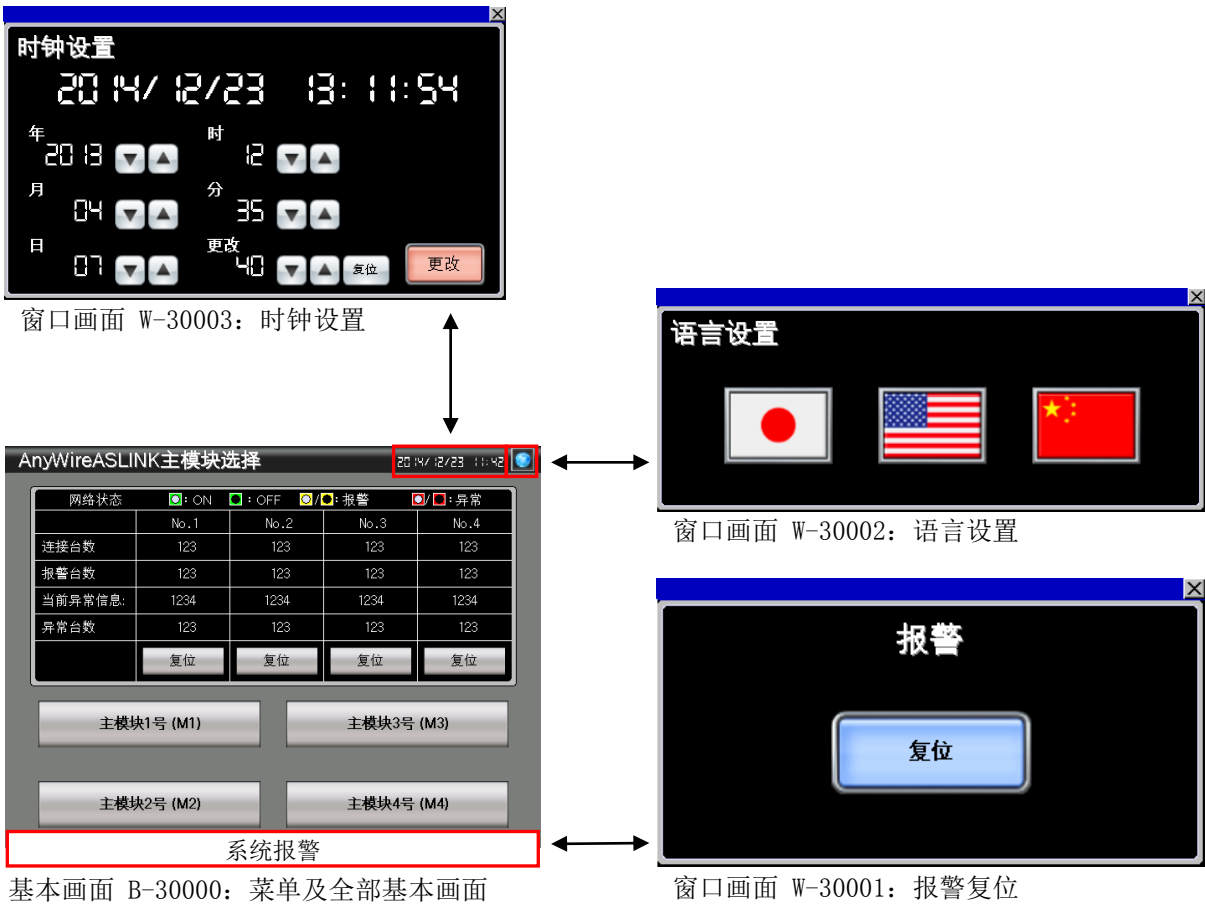
5.1 显示语言

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 495~500 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 画面一览表/切换

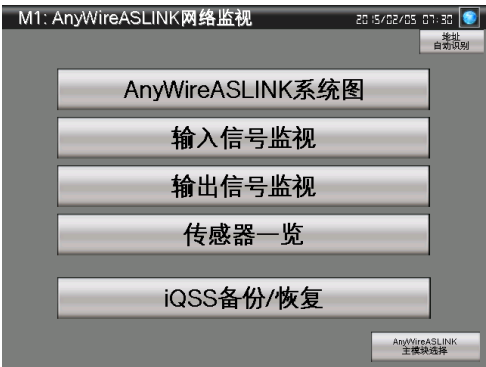
5.2.1 画面一览表/切换(公共)



5.2.2 画面一览表/切换(个别)



基本画面 B-30000: AnyWireASLINK 主模块选择



基本画面 B-30001: AnyWireASLINK 菜单



窗口画面 W-30005: 参数访问中



基本画面 B-30002: AnyWireASLINK 系统图

至下一页

接上一页



基本画面 B-30003: 输入信号监视



基本画面 B-30004: 输出信号监视



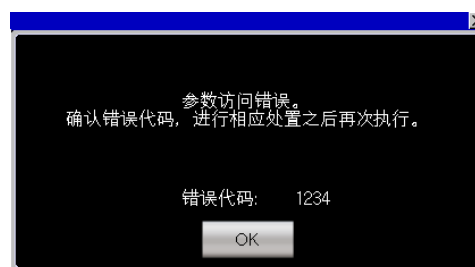
基本画面 B-30005: 传感器一览

至下一页

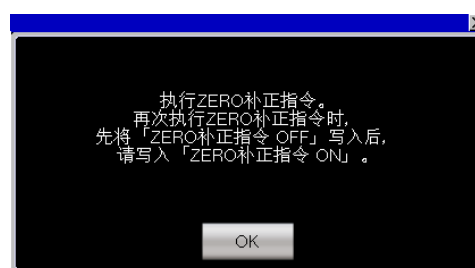
接上一页



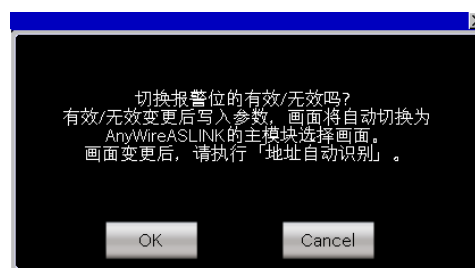
基本画面 B-30006~30062: 详细信息
(画面因从属模块的类型而异)



窗口画面 W-30004: 参数访问错误



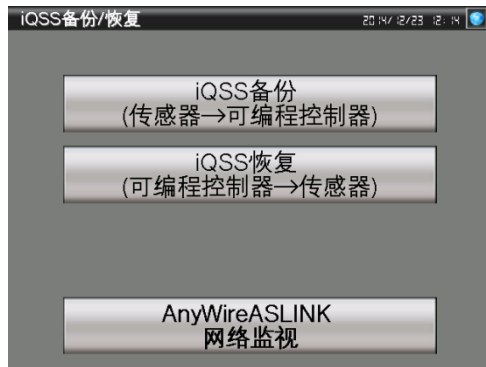
窗口画面 W-30006: 零点校正执行消息
(B-30020~30021、B-30023~30025、
B-30027~30029、B-30031、
B-30053~30054、B-30057~30058、
B-30061~30062 上显示)



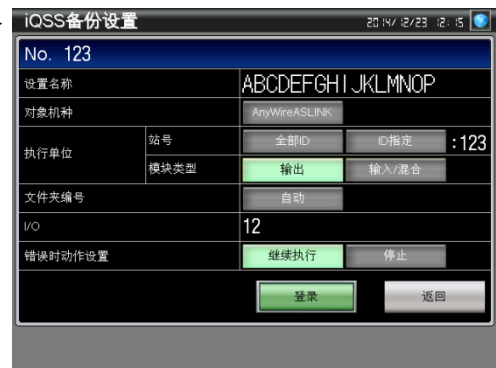
窗口画面 W-30007: 报警位切换消息
(B-30020~30021、B-30023~30025、
B-30027~30029、B-30031、
B-30053~30054、B-30057~30058、
B-30061~30062 上显示)

至下一页

接上一頁



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)



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



基本画面 B-30103: iQSS 备份进展

至下一頁

接上一页

窗口画面 W-30100: iQSS 恢复(可编程控制器→传感器)

No.	时间	设置名称	对象	文件夹编号	I/O	总数	成功	失败
123	12 / 12 / 12	12:12	ABCDEFGH I JKLMNOP	A	12	12	12	12 - 12 - 12
123	12 / 12 / 12	12:12	ABCDEFGH I JKLMNOP	A	12	12	12	12 - 12 - 12
123	12 / 12 / 12	12:12	ABCDEFGH I JKLMNOP	A	12	12	12	12 - 12 - 12
123	12 / 12 / 12	12:12	ABCDEFGH I JKLMNOP	A	12	12	12	12 - 12 - 12
123	12 / 12 / 12	12:12	ABCDEFGH I JKLMNOP	A	12	12	12	12 - 12 - 12
123	12 / 12 / 12	12:12	ABCDEFGH I JKLMNOP	A	12	12	12	12 - 12 - 12
123	12 / 12 / 12	12:12	ABCDEFGH I JKLMNOP	A	12	12	12	12 - 12 - 12
123	12 / 12 / 12	12:12	ABCDEFGH I JKLMNOP	A	12	12	12	12 - 12 - 12
123	12 / 12 / 12	12:12	ABCDEFGH I JKLMNOP	A	12	12	12	12 - 12 - 12
123	12 / 12 / 12	12:12	ABCDEFGH I JKLMNOP	A	12	12	12	12 - 12 - 12

记录删除 删除全部记录 设置确认 执行

删除选择的iQSS备份记录(包含iQSS恢复设置)。
确定吗?
删除时, 请确认对象机种和文件夹编号,
可编程控制器的SD存储卡中的该备份文件夹也请删除。

OK Cancel

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

基本画面 B-30104: iQSS 恢复

窗口画面 W-30106: iQSS 恢复进展

恢复对象	
设置名称	ABCDEFGH I JKLMNOP
执行单位	
对象机种	123
I/O	输出
文件夹编号	12
恢复处理	
开始时间	12 / 12 / 12 12:12
进展	123%
对象机种总数	12
恢复成功	12
恢复失败	12

iQSS 菜单 恢复中止

基本画面 B-30106: iQSS 恢复进展

窗口画面 W-30105: iQSS 恢复设置

No. 123

设置名称	ABCDEFGH I JKLMNOP	
对象机种	全部站号	
执行单位	站号	IC指定
	模块类型	输出
文件夹编号	12	
I/O	12	
错误时动作设置	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 主模块选择 (B-30000)

1

2

3

4

5

AnyWireASLINK主模块选择

2014/12/23 11:42

网络状态

ON

OFF

报警

异常

	No.1	No.2	No.3	No.4
连接台数	123	123	123	123
报警台数	123	123	123	123
当前异常信息:	1234	1234	1234	1234
异常台数	123	123	123	123
	复位	复位	复位	复位

主模块1号 (M1)

主模块3号 (M3)

主模块2号 (M2)

主模块4号 (M4)

概要

选择要监视的 AnyWireASLINK 的主模块画面。

详细

1. 显示 AnyWireASLINK 的各主模块的连接台数/报警台数/当前异常信息/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. 切换至 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

• 切换至 AnyWireASLINK 菜单画面时，决定要连接的主模块。

• 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」。

5.3.2 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 \leftrightarrow 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. 切换至 AnyWireASLINK 主模块选择画面。
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。
- 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」。

5.3.3 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。

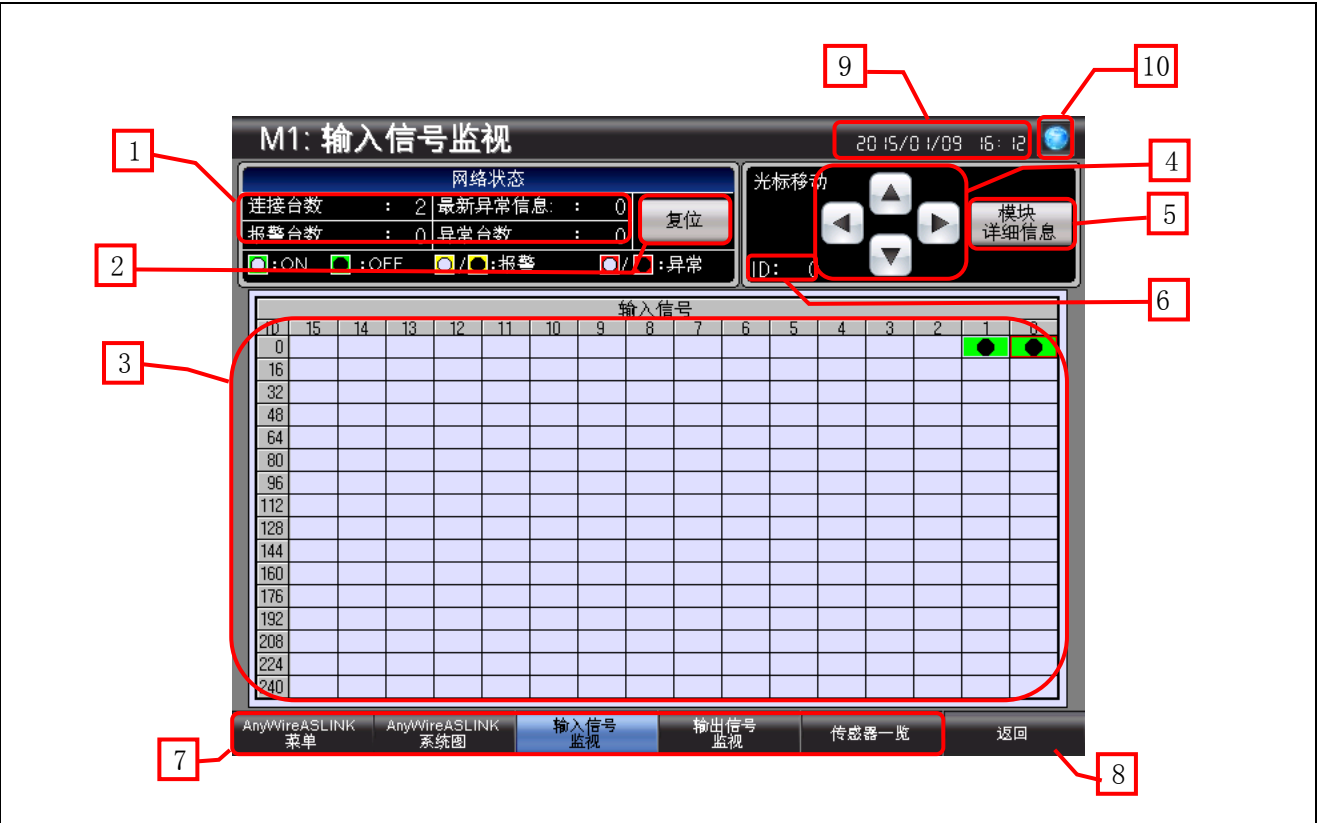
详细

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 软元件数据传送一览表」。

5.3.4 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。

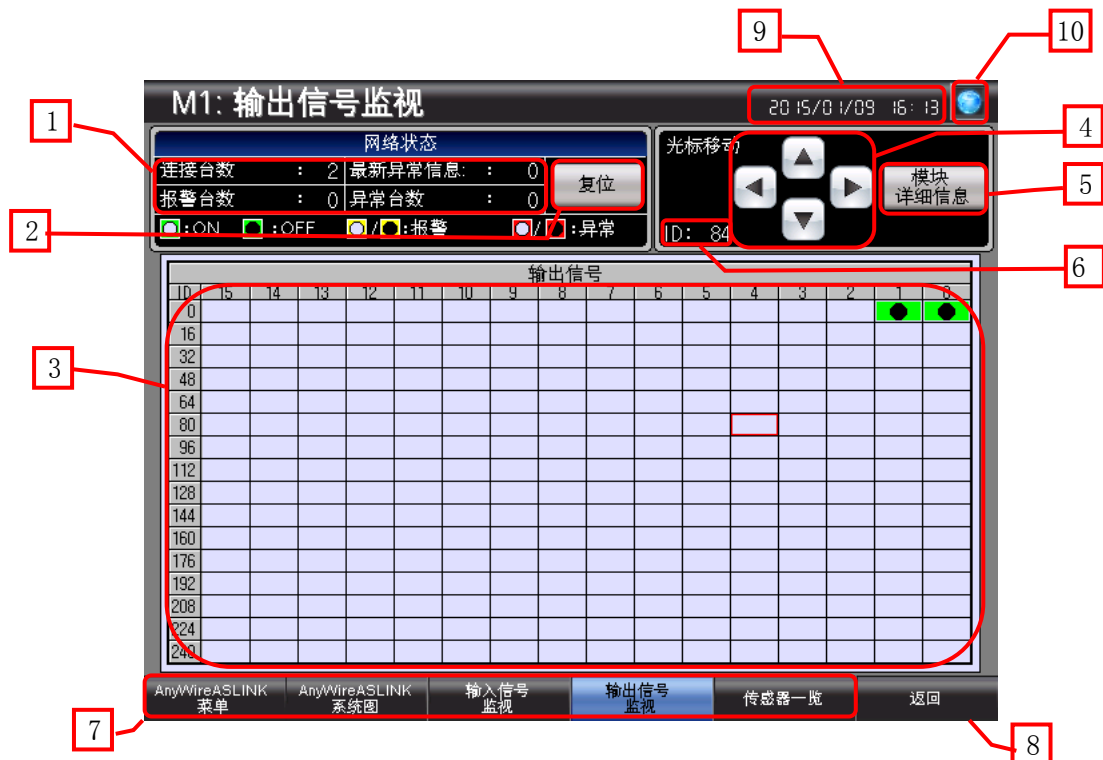
详细

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」。

5.3.5 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」。

5.3.6 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「6.1 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」。

5.3.7 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「6.1 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」。

5.3.8 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

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「6.1 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」。

5.3.9 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「6.1 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」。

5. 3. 10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

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

- ・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「6.1 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」。

5. 3. 11 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「6.1 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」。

5. 3. 12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「6.1 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」。

5. 3. 13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「6.1 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」。

5. 3. 14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、「临界值」、「滞后作用」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「6.1 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」。

5. 3. 15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「6.1 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」。

5. 3. 16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。

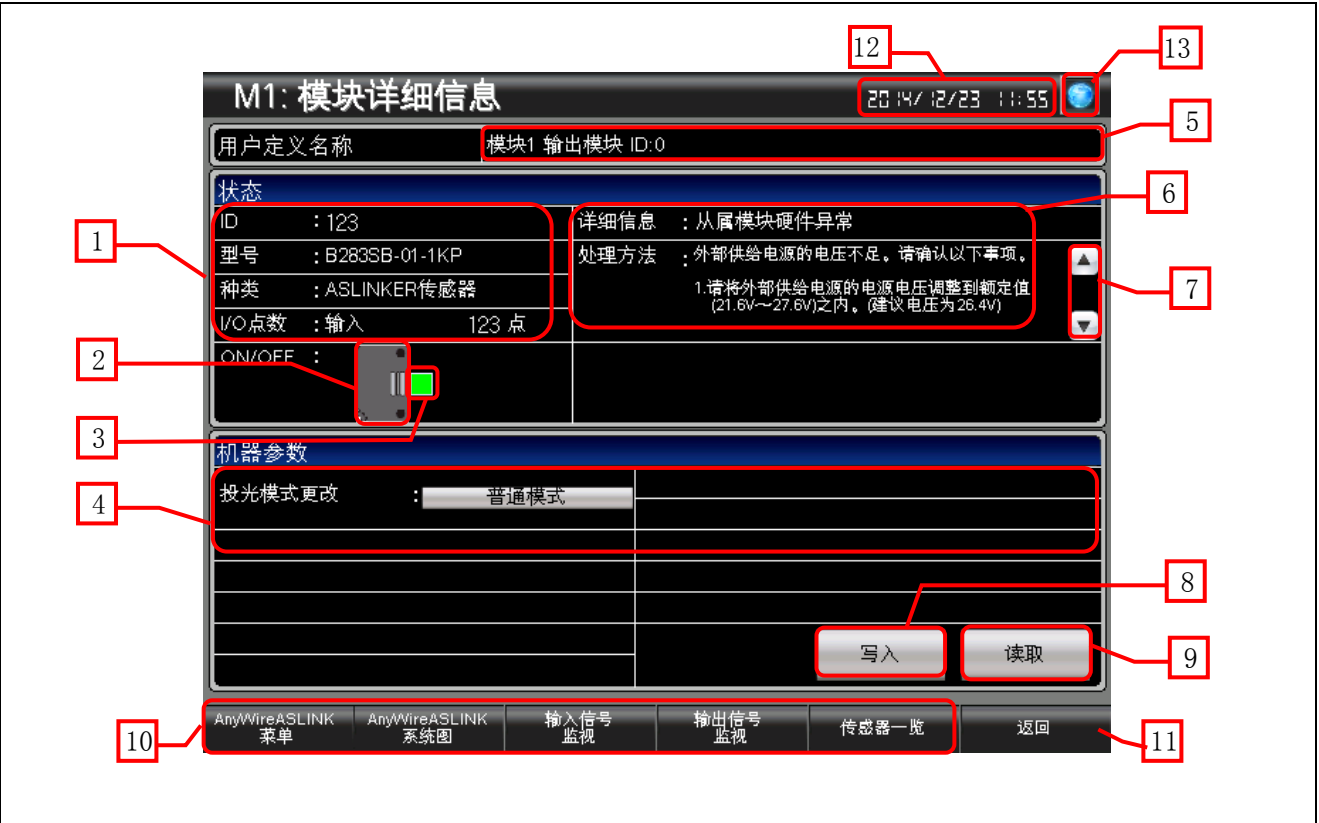
详细

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「6.1 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」。

5. 3. 17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「6.1 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」。

5. 3. 18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、「临界值」、「滞后作用」、「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「6.1 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」。

5.3.19 详细信息 (ASLINK 终端 (驱动部)) (B-30018)



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. 通过数值显示和液位，显示传感器测定值水平的当前值。
8. 因为不存在写入参数，即使触摸也无动作。
9. 因为不存在读取参数，即使触摸也无动作。
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

- 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「6.1 关于用户定义名称的登录」。
- 在「传感器测定值水平」的液位中设置了对象脚本。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」。

5. 3. 20详细信息(光纤放大器模块(投光专用))(B-30019)



概要

显示光纤放大器模块(投光专用)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「6.1 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」。

5. 3. 21详细信息(压力传感器(正压 1点))(B-30020)



概要

显示压力传感器(正压 1点)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。
- 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「6.1 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」。

5. 3. 22详细信息(压力传感器(正压 2点))(B-30021)



概要

显示压力传感器(正压 2点)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。
- 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「6.1 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」。

5. 3. 23详细信息(压力传感器(正压 16 点))(B-30023)



概要

显示压力传感器(正压 16 点)的从属模块的详细信息。

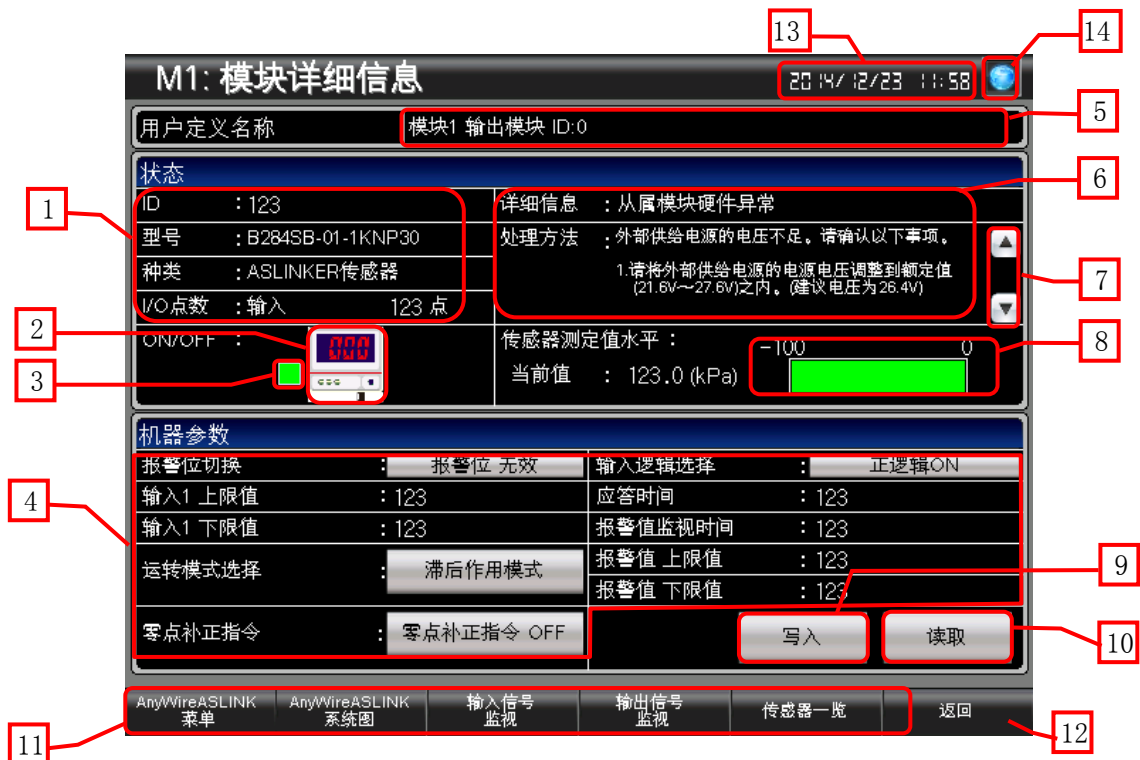
详细

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。
- 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「6.1 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」。

5. 3. 24详细信息(压力传感器(负压 1点))(B-30024)



概要

显示压力传感器(负压 1点)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。
- 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「6.1 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」。

5. 3. 25详细信息(压力传感器(负压 2点))(B-30025)



概要

显示压力传感器(负压 2点)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。
- 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「6.1 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」。

5. 3. 26详细信息(压力传感器(负压 16 点))(B-30027)



概要

显示压力传感器(负压 16 点)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。
- 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「6.1 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」。

5. 3. 27详细信息(压力传感器(复合压力 1 点))(B-30028)



概要

显示压力传感器(复合压力 1 点)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。
- 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「6.1 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」。

5. 3. 28详细信息(压力传感器(复合压力 2 点))(B-30029)



概要

显示压力传感器(复合压力 2 点)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。
- 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「6.1 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」。

5. 3. 29详细信息(压力传感器(复合压力 16 点))(B-30031)



概要

显示压力传感器(复合压力 16 点)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。
- 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 GOTO 的注释。详细内容请参照「6.1 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」。

5. 3. 30详细信息(光遮断器) (B-30032)



概要

显示光遮断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、「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「6.1 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」。

5. 3. 31详细信息(映射传感器(梳型主机))(B-30033)



概要

显示映射传感器(梳型主机)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。
- 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「6.1 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」。

5. 3. 32详细信息(映射传感器(梳型子机))(B-30034)



概要

显示映射传感器(梳型子机)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。
- 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「6.1 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」。

5. 3. 33详细信息(映射传感器(透过型受光))(B-30035)



概要

显示映射传感器(透过型受光)的从属模块的详细信息。

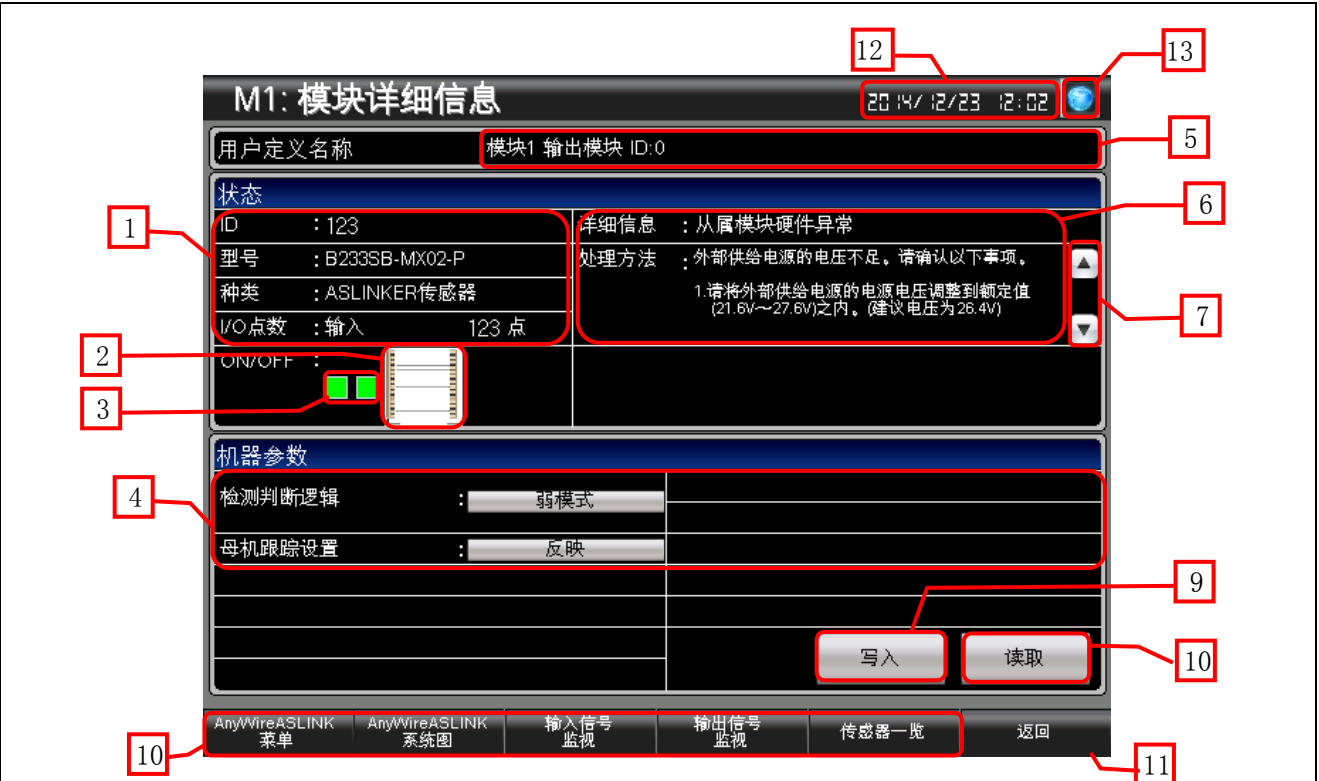
详细

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「报警临界值」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「6.1 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」。

5. 3. 34详细信息(映射传感器(透过型投光))(B-30036)



概要

显示映射传感器(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

- 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「6.1 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」。

5. 3. 35详细信息(一体型(e-con 连接器))(B-30037)



概要

显示一体型(e-con 连接器)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. 显示用户定义名称。
5. 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。
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「6.1 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」。

5. 3. 36详细信息(一体型(JST 连接器))(B-30038)



概要

显示一体型(JST 连接器)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. 显示用户定义名称。
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。
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「6.1 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」。

5. 3. 37详细信息(一体型(MOLEX 连接器))(B-30039)



概要

显示一体型(MOLEX 连接器)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. 显示用户定义名称。
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。
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「6.1 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」。

5. 3. 38详细信息(多通驱动) (B-30040)



概要

显示多通驱动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「6.1 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」。

5.3.39 详细信息(一体型继电器输出(驱动部))(B-30041)



概要

显示一体型继电器输出(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「6.1 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」。

5.3.40 详细信息(一体型继电器输出)(B-30042)



概要

显示一体型继电器输出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「6.1 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」。

5.3.41 详细信息(手持遥控(开关侧))(B-30051)



概要

显示手持遥控(开关侧)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「6.1 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」。

5.3.42 详细信息(手持遥控(LED 侧))(B-30052)



概要

显示手持遥控 (LED 侧) 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「6.1 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」。

5. 3. 43详细信息(压力传感器(正压 2点 有报警))(B-30053)



概要

显示压力传感器(正压 2点 有报警)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。
- 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「6.1 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」。

5. 3. 44详细信息(压力传感器(正压 3点 有报警))(B-30054)



概要

显示压力传感器(正压 3点 有报警)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。
- 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「6.1 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」。

5. 3. 45详细信息(压力传感器(负压 2点 有报警))(B-30057)



概要

显示压力传感器(负压 2点 有报警)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。
- 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「6.1 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」。

5. 3. 46详细信息(压力传感器(负压 3点 有报警))(B-30058)



概要

显示压力传感器(负压 3点 有报警)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。
- 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「6.1 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」。

5. 3. 47详细信息(压力传感器(复合压力 2点 有报警))(B-30061)



概要

显示压力传感器(复合压力 2点 有报警)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。
- 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「6.1 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」。

5. 3. 48详细信息(压力传感器(复合压力 3点 有报警))(B-30062)



概要

显示压力传感器(复合压力 3点 有报警)的从属模块的详细信息。

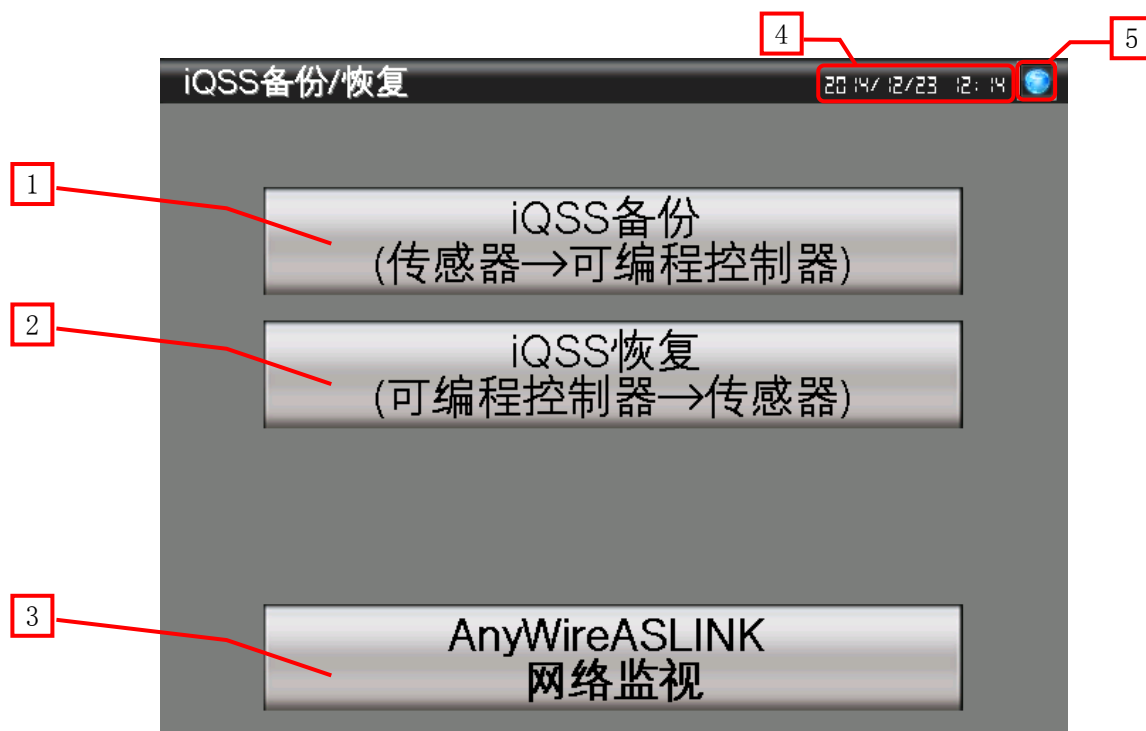
详细

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。
- 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「6.1 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」。

5.3.49iQSS 菜单(B-30100)



概要

iQSS 备份/恢复(可编程控制器↔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」。

5.3. 50iQSS 备份(B-30101)



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。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。每次切换 10 件。向下最大可显示到 No. 100。
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」。

5.3. 51iQSS 备份设置(B-30102)

1

iQSS备份设置

2014/12/23 12:15

10

11

No. 123

设置名称

ABCDEFGHIJKLMN

2

对象机种

AnyWireASLINK

3

执行单位

站号

全部ID

ID指定

:123

4

模块类型

输出

输入/混合

5

文件夹编号

自动

6

I/O

12

7

错误时动作设置

继续执行

停止

8

登录

返回

9

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」。

5.3.52iQSS 备份进展(B-30103)

The screenshot shows the 'iQSS备份进展' (iQSS Backup Progress) window. It is divided into two main sections: '备份对象' (Backup Object) and '备份处理' (Backup Processing). The '备份对象' section contains a table with columns for '设置名称' (Setting Name), '执行单位' (Execution Unit), '对象机器' (Object Machine), and 'I/O'. The '备份处理' section contains a table with columns for '开始时间' (Start Time), '进展' (Progress), '对象机器总数' (Total Object Machines), '备份成功' (Backup Success), and '备份失败' (Backup Failure). At the bottom, there are two buttons: 'iQSS 菜单' (iQSS Menu) and '备份中止' (Backup Stop). Red callout boxes with numbers 1 through 9 point to specific elements: 1 points to the 'iQSS 菜单' button, 2 points to the '设置名称' column, 3 points to the '开始时间' column, 4 points to the '进展' column, 5 points to the '对象机器总数' column, 6 points to the '备份成功' column, 7 points to the '备份中止' button, 8 points to the window title bar, 9 points to the system clock and status bar.

iQSS备份进展			
2014/12/23 12:15			
备份对象			
设置名称	ABCDEFGH I JKLMNOP		
执行单位			
对象机器	123		
	输出		
I/O	12		
备份处理			
开始时间	12 / 12 / 12 12 : 12		
进展	123%		
对象机器总数	12		
备份成功	12		
备份失败	12		
iQSS 菜单		备份中止	

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」。

5.3. 53iQSS 恢复(B-30104)



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。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。每次切换 10 件。向下最大可显示到 No. 200。
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「6.5 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」。

5.3. 54iQSS 恢复设置(B-30105)

iQSS恢复设置

2014/12/23 12:16

1 No. 123

3 设置名称 ABCDEFGHI JKLMNOP

4 对象机种

5 执行单位 站号 全部站号 ID指定 模块类型 输出

6 文件夹编号 12

7 I/O 12

8 错误时动作设置 继续执行 停止

2 返回

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」。

5.3. 55iQSS 恢复进展 (B-30106)



概要

显示 iQSS 恢复的进展。

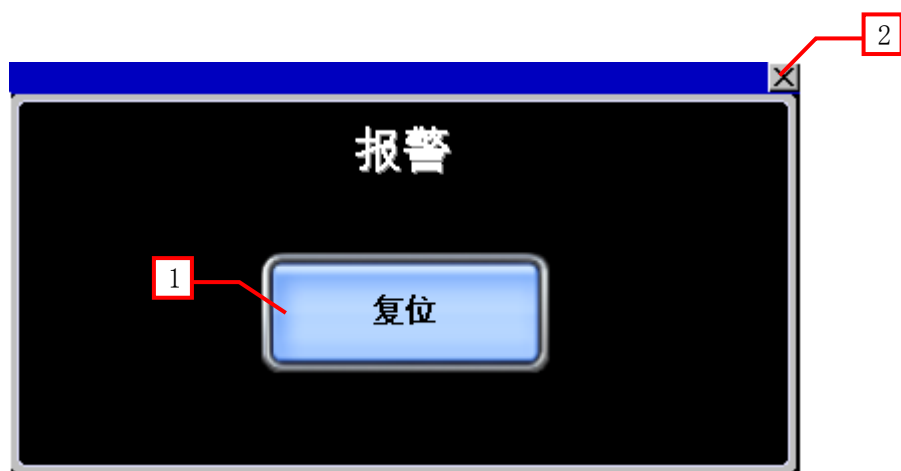
详细

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」。

5. 3. 56报警复位(W-30001)



概要

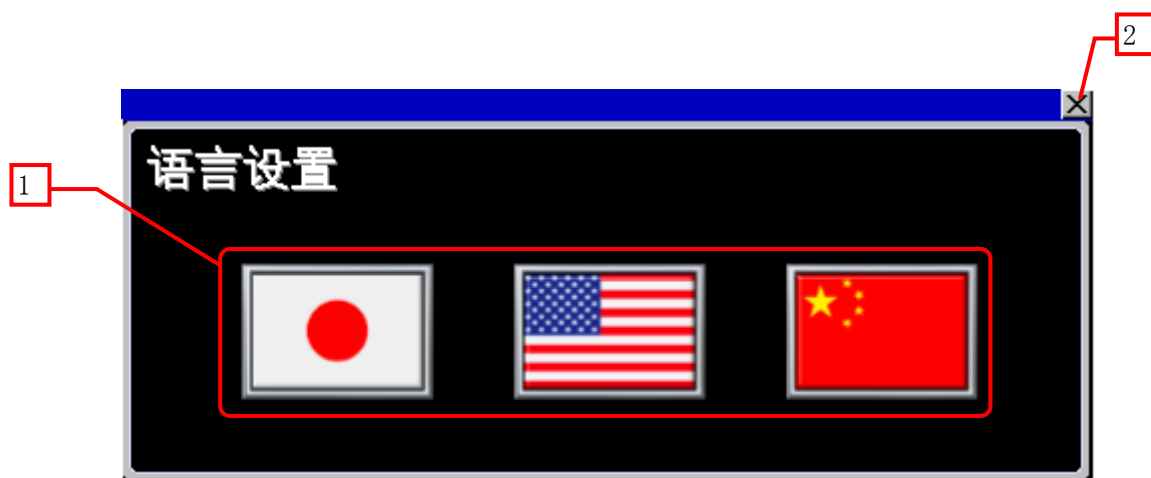
复位系统报警。

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. 57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。

5. 3. 58时钟设置(W-30003)



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。
详细内容，请参照「5.9 脚本一览表」。

5. 3. 59参数访问错误(W-30004)



概要

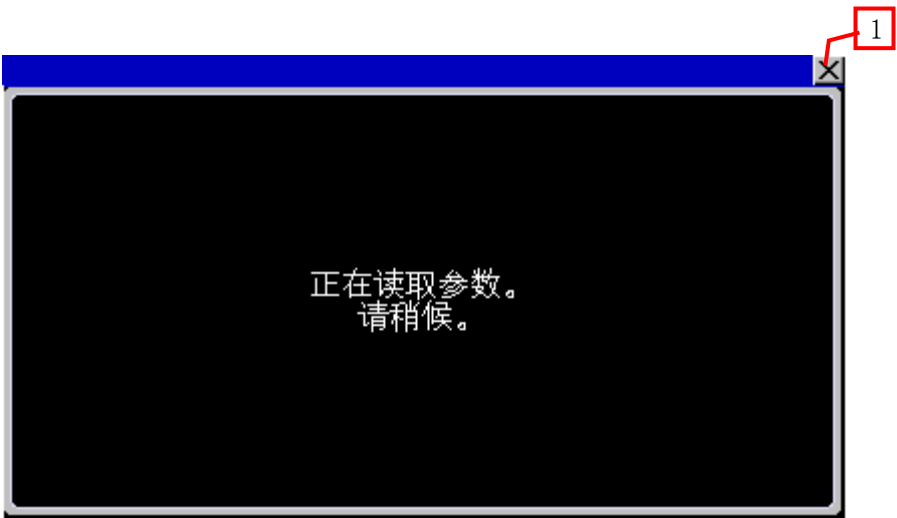
参数访问中发生错误时显示。

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。

备注

5. 3. 60参数访问中(W-30005)



概要

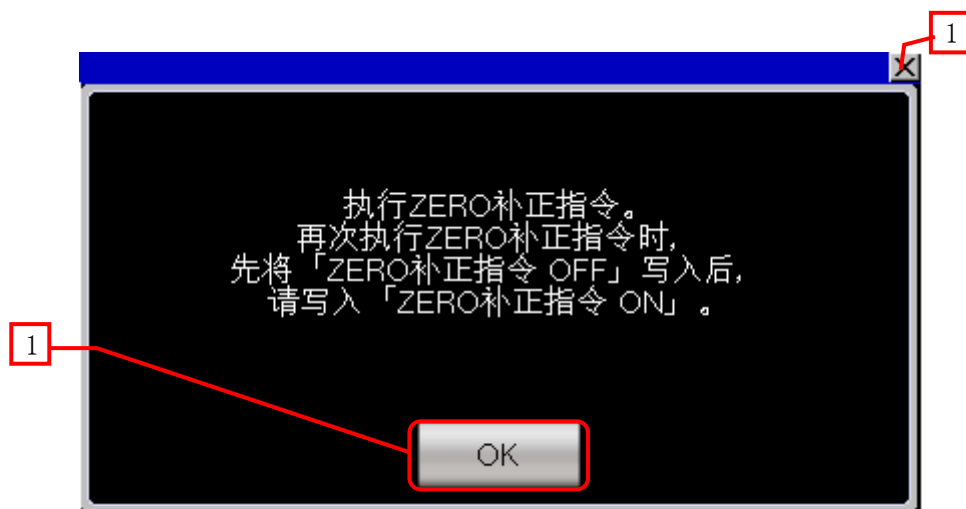
参数访问时显示。

详细

1. 关闭窗口画面。

备注

5. 3. 61 零点补正执行消息 (W-30006)



概要

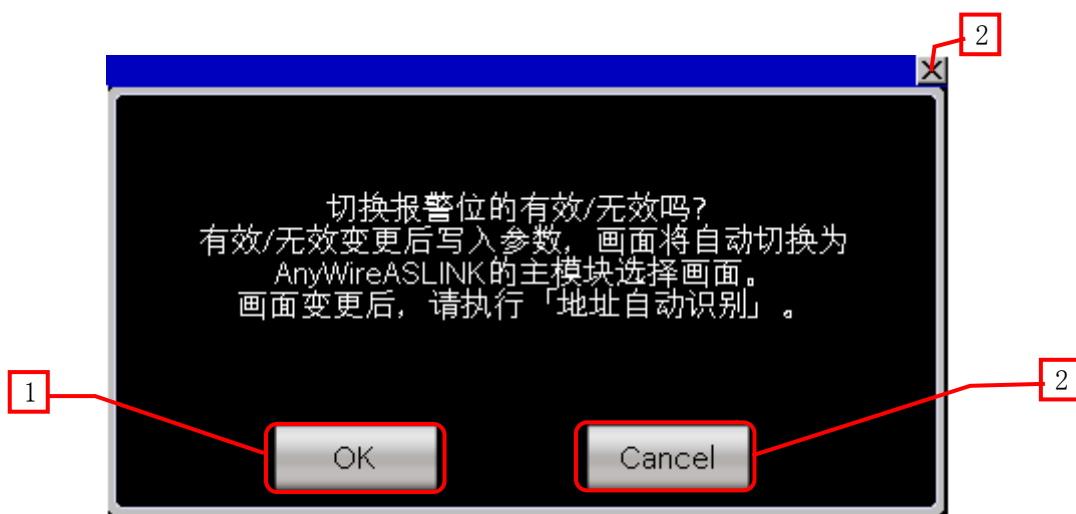
零点补正指令从 OFF 到 ON 时显示。

详细

1. 关闭窗口画面。

备注

5. 3. 62报警位切换消息(W-30007)



概要

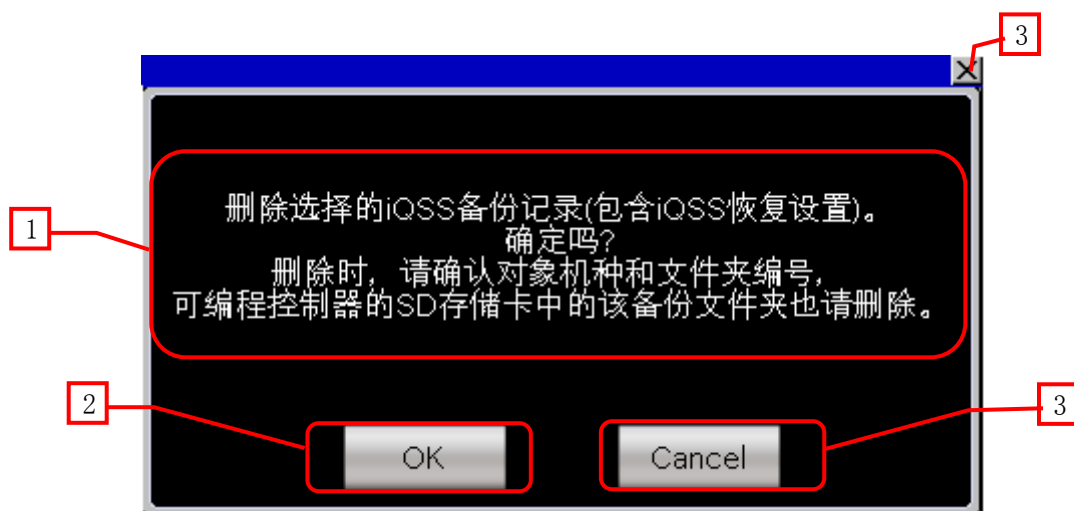
切换报警位的有效/无效。

详细

1. 切换报警位的有效/无效, 并关闭窗口。
2. 关闭窗口画面。

备注

5. 3. 63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。

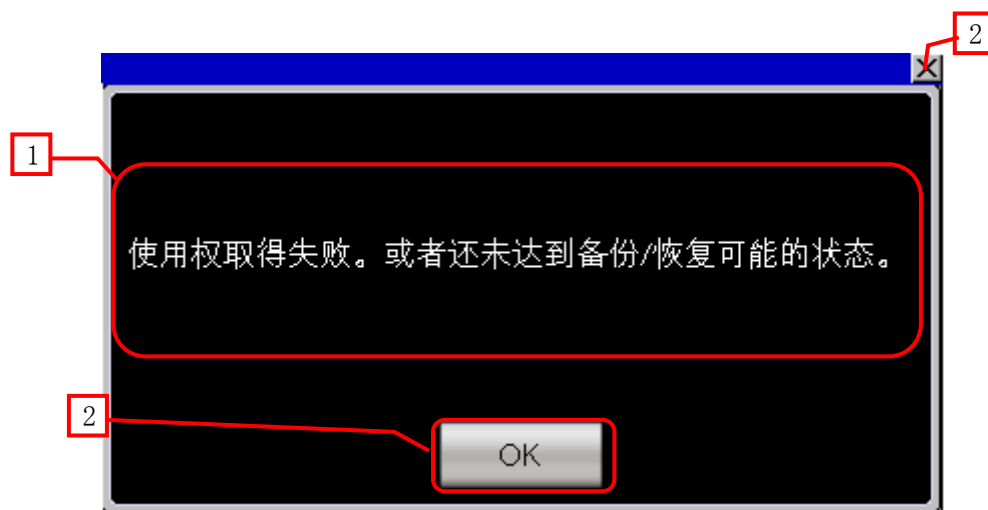
详细

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。

备注

- 显示的信息, 因开启窗口画面的开关而异。

5. 3. 64通知对话框(W-30101)



概要

通知处理结果。

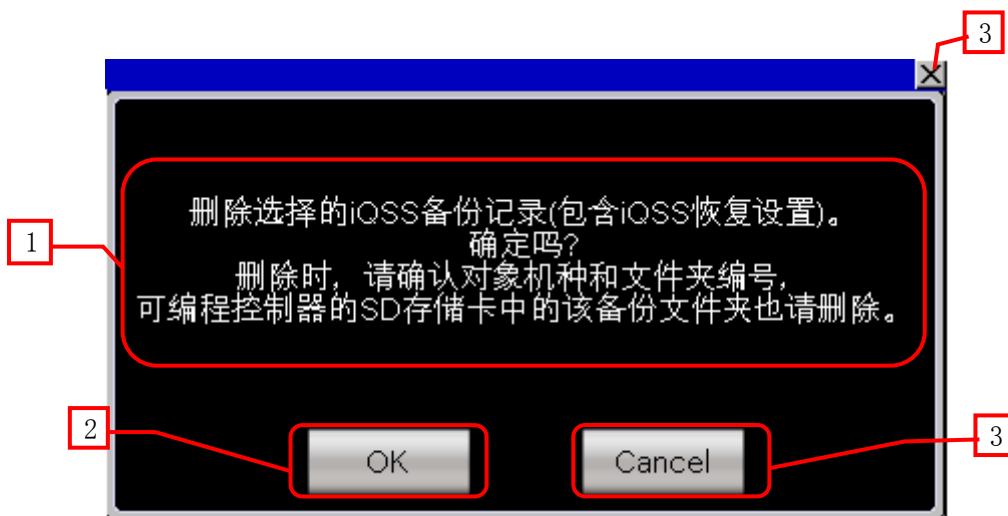
详细

1. 显示信息。
2. 关闭窗口画面。

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. 65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。

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
1401	B289SB-01AF-CAM20P	ASLINKER 放大器	B-30019
1450	B289SB-01AF-CAS	ASLINKER 放大器	B-30012
1451	B289SB-01AF-CASP	ASLINKER 放大器	B-30019
1500	B289SB-01V1-CPM20	ASLINKER 放大器	B-30043
1503	B289SB-J1V1-CPM20	ASLINKER 放大器	B-30046
1553	B289SB-J1V1-CPS	ASLINKER 放大器	B-30050
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
2601	B295SB-01-1K25	ASLINKER 传感器	B-30017
2602	B295SB-01-1K27	ASLINKER 传感器	B-30017
2603	B295SB-01-1K24	ASLINKER 传感器	B-30017
2900	B284SB-01-12	ASLINKER 传感器	B-30020
2903	B284SB-02-PP30	ASLINKER 传感器	B-30021
2909	B284SB-J1-PP30	ASLINKER 传感器	B-30023
2930	B284SB-01-NP30	ASLINKER 传感器	B-30024
2933	B284SB-02-NP30	ASLINKER 传感器	B-30025
2939	B284SB-J1-NP30	ASLINKER 传感器	B-30027

从属模块 型号编号	型号	大分类	基本画面号
2960	B284SB-01-LP30	ASLINKER 传感器	B-30028
2963	B284SB-02-LP30	ASLINKER 传感器	B-30029
2969	B284SB-J1-LP30	ASLINKER 传感器	B-30031
3100	B297SB-01-1K40	ASLINKER 传感器	B-30032
3300	B232SB-MX100-STP	ASLINKER 传感器	B-30033
3301	B232SB-SX100-STP	ASLINKER 传感器	B-30034
3302	B233SB-02-M-C	ASLINKER 传感器	B-30035
3303	B233SB-02-S-C	ASLINKER 传感器	B-30035
3304	B233SB-02-M-P	ASLINKER 传感器	B-30036
3305	B233SB-02-S-P	ASLINKER 传感器	B-30036
10000	BL296SB-08F-20	ASLINKER 终端	B-30018
10001	BL296SB-08FS-20	ASLINKER 终端	B-30018
10002	BL296SB-08F-4-20	ASLINKER 终端	B-30037
10003	BL296SB-08FS-4-20	ASLINKER 终端	B-30037
10004	BL296SB-08F-9-20	ASLINKER 终端	B-30038
10005	BL296SB-08FS-9-20	ASLINKER 终端	B-30038
10006	BL296SB-08F-10-20	ASLINKER 终端	B-30039
10007	BL296SB-08FS-10-20	ASLINKER 终端	B-30039
10010	B262-HR16-IN	ASLINKER 终端	B-30051
10300	BL296PB-08F-20	ASLINKER 终端	B-30018
10301	BL296PB-08FS-20	ASLINKER 终端	B-30018
10302	BL296PB-08F-4-20	ASLINKER 终端	B-30037
10303	BL296PB-08FS-4-20	ASLINKER 终端	B-30037
10304	BL296PB-08F-9-20	ASLINKER 终端	B-30038
10305	BL296PB-08FS-9-20	ASLINKER 终端	B-30038
10306	BL296PB-08F-10-20	ASLINKER 终端	B-30039
10307	BL296PB-08FS-10-20	ASLINKER 终端	B-30039
10308	BL264PB-16F-T5	ASLINKER 终端	B-30040
10309	BL264PB-16FS-T5	ASLINKER 终端	B-30040
10310	B262-HR16-OUT	ASLINKER 终端	B-30052
10320	BL296PB-08RSN	ASLINKER 终端	B-30041
10321	BL296PB-08RS	ASLINKER 终端	B-30042
10322	BL296PB-08RSS	ASLINKER 终端	B-30042
10323	BL296PB-08RSS1	ASLINKER 终端	B-30042
10324	BL296PB-08RSS2	ASLINKER 终端	B-30042
10325	BL296PB-08RSS3	ASLINKER 终端	B-30042
10326	BL296PB-08RSS4	ASLINKER 终端	B-30042
10600	BL296XB-08F-20	ASLINKER 终端	B-30018
10601	BL296XB-08FS-20	ASLINKER 终端	B-30018
10602	BL296XB-08F-4-20	ASLINKER 终端	B-30037
10603	BL296XB-08FS-4-20	ASLINKER 终端	B-30037
10604	BL296XB-08F-9-20	ASLINKER 终端	B-30038
10605	BL296XB-08FS-9-20	ASLINKER 终端	B-30038
10606	BL296XB-08F-10-20	ASLINKER 终端	B-30039
10607	BL296XB-08FS-10-20	ASLINKER 终端	B-30039

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	软元件编号	用途
位	X0011、X0031、X0051、X0071	参数访问完成标志
	X0012、X0032、X0052、X0072	参数访问异常
	Y0000、Y0020、Y0040、Y0060	异常标志清除指令
	Y0001、Y0021、Y0041、Y0061	传感器自动识别指令
	Y0010、Y0030、Y0050、Y0070	主模块发往从属模块的参数访问请求指令
	Y0011、Y0031、Y0051、Y0071	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	U00-G0~U00-G15	主模块 1 号 输入信息区域(U00-G0，用作软元件数据传送的基准软元件)
	U00-G1	主模块 1 号 用作软元件数据传送的基准软元件
	U00-G20	主模块 1 号 用作软元件数据传送的基准软元件
	U00-G40	主模块 1 号 用作传感器测定值水平的基准软元件
	U00-G4096~U00-G4111	主模块 1 号 输出信息区域
	U00-G8192	主模块 1 号 异常 ID 个数信息
	U00-G8193~U00-G8320	主模块 1 号 异常 ID 信息存储区域
	U00-G8960	主模块 1 号 连接台数信息
	U00-G9216	主模块 1 号 连接 ID 个数信息
	U00-G9217~U00-G9344	主模块 1 号 连接 ID 信息存储区域
	U00-G9984	主模块 1 号 报警 ID 个数信息
	U00-G9985~U00-G10112	主模块 1 号 报警 ID 信息存储区域
	U00-G10256	主模块 1 号 最新错误代码存储区域
	U00-G10320	主模块 1 号 参数访问设置
	U00-G10321	主模块 1 号 参数访问对象 ID 指定
	U00-G10496~U00-G10751	主模块 1 号 参数存储目标存储器号(输出)
	U00-G11008~U00-G11263	主模块 1 号 参数存储目标存储器号(输入)
	U00-G12288~U00-G18431	主模块 1 号 参数存储区域
	U02-G1	主模块 2 号 用作软元件数据传送的基准软元件
	U02-G20	主模块 2 号 用作软元件数据传送的基准软元件

类型	软元件编号	用途
字	U02-G40	主模块 2 号 用作传感器测定值水平的基准软元件
	U02-G4096～ U02-G4111	主模块 2 号 输出信息区域
	U02-G8192	主模块 2 号 异常 ID 个数信息
	U02-G8193～ U02-G8320	主模块 2 号 异常 ID 信息存储区域
	U02-G8960	主模块 2 号 连接台数信息
	U02-G9216	主模块 2 号 连接 ID 个数信息
	U02-G9217～ U02-G9344	主模块 2 号 连接 ID 信息存储区域
	U02-G9984	主模块 2 号 报警 ID 个数信息
	U02-G9985～ U02-G10112	主模块 2 号 报警 ID 信息存储区域
	U02-G10256	主模块 2 号 当前错误代码存储区域
	U02-G10320	主模块 2 号 参数访问设置
	U02-G10321	主模块 2 号 参数访问对象 ID 指定
	U02-G10496～ U02-G10751	主模块 2 号 参数存储目标存储器号(输出)
	U02-G11008～ U02-G11263	主模块 2 号 参数存储目标存储器号(输入)
	U02-G12288～ U02-G18431	主模块 2 号 参数存储区域
	U04-G0	主模块 3 号 用作软元件数据传送的基准软元件
	U04-G1	主模块 3 号 用作软元件数据传送的基准软元件
	U04-G20	主模块 3 号 用作软元件数据传送的基准软元件
	U04-G40	主模块 3 号 用作传感器测定值水平的基准软元件
	U04-G4096～ U04-G4111	主模块 3 号 输出信息区域
	U04-G8192	主模块 3 号 异常 ID 个数信息
	U04-G8193～ U04-G8320	主模块 3 号 异常 ID 信息存储区域
	U04-G8960	主模块 3 号 连接台数信息
	U04-G9216	主模块 3 号 连接 ID 个数信息
	U04-G9217～ U04-G9344	主模块 3 号 连接 ID 信息存储区域
	U04-G9984	主模块 3 号 报警 ID 个数信息
	U04-G9985～ U04-G10112	主模块 3 号 报警 ID 信息存储区域
	U04-G10256	主模块 3 号 当前错误代码存储区域
	U04-G10320	主模块 3 号 参数访问设置
	U04-G10321	主模块 3 号 参数访问对象 ID 指定
	U04-G10496～ U04-G10751	主模块 3 号 参数存储目标存储器号(输出)
	U04-G11008～ U04-G11263	主模块 3 号 参数存储目标存储器号(输入)
	U04-G12288～ U04-G18431	主模块 3 号 参数存储区域
	U06-G0	主模块 4 号 用作软元件数据传送的基准软元件
	U06-G1	主模块 4 号 用作软元件数据传送的基准软元件
	U06-G20	主模块 4 号 用作软元件数据传送的基准软元件
	U06-G40	主模块 4 号 用作传感器测定值水平的基准软元件
	U06-G4096～ U06-G4111	主模块 4 号 输出信息区域
	U06-G8192	主模块 4 号 异常 ID 个数信息

类型	软元件编号	用途
字	U06-G8193～ U06-G8320	主模块 4 号 异常 ID 信息存储区域
	U06-G8960	主模块 4 号 连接台数信息
	U06-G9216	主模块 4 号 连接 ID 个数信息
	U06-G9217～ U06-G9344	主模块 4 号 连接 ID 信息存储区域
	U06-G9984	主模块 4 号 报警 ID 个数信息
	U06-G9985～ U06-G10112	主模块 4 号 报警 ID 信息存储区域
	U06-G10256	主模块 4 号 当前错误代码存储区域
	U06-G10320	主模块 4 号 参数访问设置
	U06-G10321	主模块 4 号 参数访问对象 ID 指定
	U06-G10496～ U06-G10751	主模块 4 号 参数存储目标存储器号(输出)
	U06-G11008～ U06-G11263	主模块 4 号 参数存储目标存储器号(输入)
	U06-G12288～ U06-G18431	主模块 4 号 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
	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 兼容备份文件夹编号软元件
	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	软元件编号	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 30013 启动触发
	GB30011	脚本号 30018 启动触发
	GB30012	脚本号 30022 启动触发
	GB30013	脚本号 30036 启动触发

类型	软元件编号	用途
位	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 输入输出混合模块标志
	GB30021	脚本号 30026 初次启动控制标志
	GB30024	参数个别读取标志
	GB30025	参数个别写入标志
	GB30026	脚本号 30040 启动触发
	GB30027	B-30000 初次启动控制标志
	GB30028	ON/OFF 状态检查脚本启动触发
	GB30029	脚本号 30068 启动触发
	GB30030	脚本号 30069 启动触发
	GB30032	脚本号 30086 启动触发
	GB30033	脚本号 30087 启动触发
	GB30034	压力传感器 报警位切换标志
	GB30035	脚本初次动作确认标志
	GB30044	参数一起读取标志
	GB30049	脚本号 30079 启动触发
	GB30050	脚本号 30080 启动触发
	GB30051	脚本号 30033 启动触发
	GB30052	脚本号 30081 启动触发
	GB30053	B-30001 画面切换开关动作条件
	GB30600	输出判断标志
	GB30601	输入判断标志
	GB54000	脚本号 30024 启动触发
	GB54001	脚本号 30025 启动触发
	GB54002	脚本号 30030 启动触发
	GB54004	iQSS 备份许可标志
	GB54005	脚本号 30034 启动触发
	GB54006	脚本号 30036 启动触发
	GB54008	脚本号 30033 初次启动脚本启动控制标志
	GB54009	脚本号 30005、30008 控制标志
	GB54010	脚本号 30040 启动触发
	GB54011	脚本号 30042 启动触发
	GB54012	脚本号 30036 删除范围设置标志
	GB54013	脚本号 30028 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	GD10100~GD10131	系统图用软元件(部件显示)
	GD10132~GD10163	系统图用软元件(ID (注释))
	GD10164~GD10195	系统图用软元件(ID(数值显示))
	GD10196~GD10227	系统图用软元件(模块类型)
	GD10228	系统图传感器类型 偏置用软元件
	GD10230	系统图 偏置值运算用软元件
	GD10231	输入输出检查软元件
	GD10232	系统图切换用偏置
	GD10233	系统图用偏置

类型	软元件编号	用途
字	GD10300	I/O 监视的光标位置显示
	GD10301	光标显示的移位用软元件
	GD10302	光标显示的偏置用软元件
	GD10303	I/O 监视的当前光标位置
	GD10400	传感器一览选择位置偏置
	GD10401	传感器一览滚动偏置
	GD10402~GD10416	传感器一览 模块的种类
	GD10417~GD10431	传感器一览 状态详细
	GD10432~GD10446	传感器一览 用户自定义标签
	GD10447~GD10461	传感器一览 选择位置显示（指示灯）
	GD10462~GD10476	传感器一览 ID
	GD10477~GD10491	传感器一览 模块型号
	GD10492	传感器一览 ID、类型的显示条件(位)
	GD10493	传感器一览 偏置值运算用软元件
	GD10494	传感器一览用偏置
	GD10500	模块详细信息 处理方法的显示起始行软元件
	GD10501	模块详细信息的模块型号编号保存软元件
	GD10502	模块详细信息的详细信息
	GD10503~GD10518	模块详细信息 I/O 监视的部件显示
	GD10519~GD10534	模块详细信息 I/O 的偏置
	GD10535~GD10550	模块详细信息 I/O 的屏蔽用软元件(对象脚本)
	GD10551~GD10566	模块详细信息 I/O 的字指示灯状态软元件
	GD10572	模块详细信息 压力传感器 报警位判断软元件
	GD10600	软元件数据传送 偏置
	GD10601	全部画面公共偏置(个别信息)
	GD10602	全部画面公共偏置(批量信息)
	GD11000	软元件数据传送 1-1 外部控制软元件
	GD11001	软元件数据传送 1-1 外部通知软元件
	GD11002	软元件数据传送 1-2 外部控制软元件
	GD11003	软元件数据传送 1-2 外部通知软元件
	GD11004	软元件数据传送 1-3 外部控制软元件
	GD11005	软元件数据传送 1-3 外部通知软元件
	GD11006	软元件数据传送 1-4 外部控制软元件
	GD11007	软元件数据传送 1-4 外部通知软元件
	GD11008	软元件数据传送 2-1 外部控制软元件
	GD11009	软元件数据传送 2-1 外部通知软元件
	GD11010	软元件数据传送 2-2 外部控制软元件
	GD11011	软元件数据传送 2-2 外部通知软元件
	GD11012	软元件数据传送 2-3 外部控制软元件
	GD11013	软元件数据传送 2-3 外部通知软元件
	GD11014	软元件数据传送 2-4 外部控制软元件
	GD11015	软元件数据传送 2-4 外部通知软元件
	GD11016	软元件数据传送 3-1 外部控制软元件
	GD11017	软元件数据传送 3-1 外部通知软元件
	GD11018	软元件数据传送 3-2 外部控制软元件
	GD11019	软元件数据传送 3-2 外部通知软元件
	GD11020	软元件数据传送 3-3 外部控制软元件
	GD11021	软元件数据传送 3-3 外部通知软元件
	GD11022	软元件数据传送 3-4 外部控制软元件
	GD11023	软元件数据传送 3-4 外部通知软元件
	GD11024	软元件数据传送 4-1 外部控制软元件
	GD11025	软元件数据传送 4-1 外部通知软元件
	GD11026	软元件数据传送 4-2 外部控制软元件

类型	软元件编号	用途
字	GD11027	软元件数据传送 4-2 外部通知软元件
	GD11028	软元件数据传送 4-3 外部控制软元件
	GD11029	软元件数据传送 4-3 外部通知软元件
	GD11030	软元件数据传送 4-4 外部控制软元件
	GD11031	软元件数据传送 4-4 外部通知软元件
	GD11500~GD11528	机器参数(个别)
	GD12000~GD12511	I/O 监视的部件显示
	GD16000~GD16511	I/O 监视 模块的种类
	GD20000~GD20931	模块的个别信息(模块 1)
	GD21000~GD21931	模块的个别信息(模块 2)
	GD22000~GD22931	模块的个别信息(模块 3)
	GD23000~GD23931	模块的个别信息(模块 4)
	GD24000~GD30143	机器参数(批量)(模块 1)
	GD31000~GD37143	机器参数(批量)(模块 2)
	GD38000~GD44143	机器参数(批量)(模块 3)
	GD45000~GD51143	机器参数(批量)(模块 4)
	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-30004~30006 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
	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
	GD62031、GD62041	系统信息

类型	软元件编号	用途
字	GD63990	时间更改软元件(年)
	GD63991	时间更改软元件(月)
	GD63992	时间更改软元件(日)
	GD63993	时间更改软元件(时)
	GD63994	时间更改软元件(分)
	GD63995	时间更改软元件(秒)
	GD64000	缓冲存储器模块号切换软元件
	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～TMP30、 TMP100～TMP101、 TMP110～TMP111、 TMP120～TMP128、 TMP200～TMP202、 TMP950～TMP996 TMP1000、	脚本运算用

5.6 注释一览表

注释组号	注释号	使用处
495	No. 1~2、4、8、32、 100~104、200~202、 300~305、400~401、500	B-30006~30062
496	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。)	
497		
498	No. 1~255、513~767、 1001~1255、1513~1767、 2001~2255、2513~2767、 3001~3255、3513~3767	B-30005~30021、B-30023~30025、 B-30027~30029、B-30031~30042、 B-30051~30054、B-30057~30058、 B-30061~30062
499	No. 1~2、4、8、32、	B-30002、B-30005~30021、 B-30023~30025、B-30027~30029、 B-30031~30042、B-30051~30054、 B-30057~30058、B-30061~30062
	100~104、200~202、 300~305、400~401、500	B-30002、B-30005~30021、 B-30023~30025、B-30027~30029、 B-30031~30042、B-30051~30054、 B-30057~30058、B-30061~30062、 W-30101
	No. 18433~18443、 32000~32017	W-30101
	No. 32767	B-30002、B-30005~30021、 B-30023~30025、B-30027~30029、 B-30031~30042、B-30051~30054、 B-30057~30058、B-30061~30062
500	No. 20	B-30002
	No. 22	B-30005
	No. 23~24	B-30005~30021、B-30023~30025、 B-30027~30029、B-30031~30042、 B-30051~30054、B-30057~30058、 B-30061~30062
	No. 25	B-30006~30021、B-30023~30025、 B-30027~30029、B-30031~30042、 B-30051~30054、B-30057~30058、 B-30061~30062
	No. 26	B-30002~30021、B-30023~30025、 B-30027~30029、B-30031~30042、 B-30051~30054、B-30057~30058、 B-30061~30062
	No. 27	B-30001、B-30002
	No. 28	B-30002~30021、B-30023~30025、 B-30027~30029、B-30031~30042、 B-30051~30054、B-30057~30058、 B-30061~30062
	No. 29	B-30001~30021、B-30023~30025、 B-30027~30029、B-30031~30042、 B-30051~30054、B-30057~30058、 B-30061~30062
	No. 30	B-30001
	No. 32~35	B-30000、B-30002~30004
	No. 36	B-30000、B-30002~30005
	No. 37	B-30002

注释组号	注释号	使用处
500	No. 38	B-30000、B-30002～30004、W-30001
	No. 39	B-30000、B-30002～30004
	No. 40	B-30000、B-30002～30005、 W-30001、W-30003
	No. 41	B-30002～30021、B-30023～30025、 B-30027～30029、B-30031～30042、 B-30051～30054、B-30057～30058、 B-30061～30062
	No. 42	B-30002～30021、B-30023～30025、 B-30027～30029、B-30031～30042、 B-30051～30054、B-30057～30058、 B-30061～30062
	No. 43	B-30006～30021、B-30023～30025、 B-30027～30029、B-30031～30042、 B-30051～30054、B-30057～30058、 B-30061～30062
	No. 45～46	B-30000、B-30003～B-30004
	No. 47～48	B-30003～30004
	No. 49～50	B-30005
	No. 51	B-30005～30021、B-30023～30025、 B-30027～30029、B-30031～30042、 B-30051～30054、B-30057～30058、 B-30061～30062
	No. 52～58	B-30006～30021、B-30023～30025、 B-30027～30029、B-30031～30042、 B-30051～30054、B-30057～30058、 B-30061～30062
	No. 59	B-30006、B-30008～30009、B-30013
	No. 60～61	B-30006～30052
	No. 62～63	B-30010～30015、B-30017～30018、 B-30020～30021、B-30023～30025、 B-30027～30029、B-30031～30035、 B-30053～30054、B-30057～30058、 B-30061～30062
	No. 68	B-30011、B-30013、B-30017
	No. 69	B-30010、B-30012、B-30014～30015、B-30032
	No. 70～71	B-30013～30015
	No. 75	B-30006～30021、B-30023～30025、 B-30027～30029、B-30031～30042、 B-30051～30054、B-30057～30058、 B-30061～30062
	No. 76～77	B-30006、B-30008、B-30009
	No. 78～79	W-30004
	No. 81～88	W-30003
	No. 91	B-30003
	No. 92	B-30004
	No. 93	B-30006～30021、B-30023～30025、 B-30027～30029、B-30031～30042、 B-30051～30054、B-30057～30058、 B-30061～30062
	No. 94	B-30001

注释组号	注释号	使用处
500	No. 95	B-30002~30021、B-30023~30025、 B-30027~30029、B-30031~30042、 B-30051~30054、B-30057~30058、 B-30061~30062
	No. 96	B-30001、B-30004
	No. 97	B-30001、B-30003
	No. 99	B-30100
	No. 117	B-30010~30015、B-30017
	No. 118	B-30010~30015、B-30017、B-30035
	No. 119	B-30010~30015、B-30017、B-30032
	No. 120	B-30010~30012、B-30014~30015、 B-30017、B-30032
	No. 121	B-30010、B-30012、B-30014~30015、 B-30020~30021、B-30024~30025、 B-30028~30029、B-30032、 B-30053~30054、B-30057~30058、 B-30061~30062
	No. 123	B-30010~30012、B-30014~30015、 B-30017、B-30032
	No. 124	B-30014~30015
	No. 125	B-30014、B-30016
	No. 126	B-30010~30015、B-30017、B-30032
	No. 127	B-30014~30016
	No. 128	W-30002
	No. 129~132	B-30010、B-30012、B-30032
	No. 133~135	B-30010~30012
	No. 136~137	B-30010、B-30012
	No. 138~140	B-30011、B-30013、B-30017
	No. 141~143	B-30011
	No. 144	B-30014、B-30016
	No. 145	B-30006~30021、B-30023~30025、 B-30027~30029、B-30031~30042、 B-30051~30054、B-30057~30058、 B-30061~30062
	No. 146~149	B-30000
	No. 152	W-30005
	No. 153~156	B-30001~30021、B-30023~30025、 B-30027~30029、B-30031~30042、 B-30051~30054、B-30057~30058、 B-30061~30062
	No. 165~166	B-30020~30021、B-30024~30025、 B-30028~30029、B-30053~30054、 B-30057~30058、B-30061~30062
	No. 170	B-30020~30021、B-30024~30025、 B-30028~30029、B-30053~30054、 B-30057~30058、B-30061~30062
	No. 171~172	B-30020、B-30024、B-30028、B-30053、B-30057、 B-30061
	No. 173	B-30020~30021、B-30024~30025、 B-30028~30029、B-30053~30054、 B-30057~30058、B-30061~30062
	No. 174~175	B-30020、B-30024、B-30028、B-30053、B-30057、 B-30061

注释组号	注释号	使用处
500	No. 176~178	B-30020、B-30021、B-30024、B-30025、 B-30028、B-30029、B-30053~30054、 B-30057~30058、B-30061~30062
	No. 179~180	B-30020~30021、B-30023~30025、 B-30027~30029、B-30031、B-30053~30054、 B-30057~30058、B-30061~30062
	No. 181~182	B-30021、B-30025、B-30029、B-30054、 B-30058、B-30062
	No. 189~192	B-30033、B-30034
	No. 193~195	B-30034
	No. 203~206	B-30035
	No. 207~214	B-30021、B-30025、B-30029、B-30054、 B-30058、B-30062
	No. 216	B-30000
	No. 217~220	B-30035
	No. 221	B-30035、B-30036
	No. 222	B-30035
	No. 223~224	B-30035
	No. 225	B-30035、B-30036
	No. 226~235	B-30035
	No. 237~240	B-30036
	No. 241	B-30020~30021、B-30023~30025、 B-30027~30029、B-30031、B-30053~30054、 B-30057~30058、B-30061~30062
	No. 242	W-30006
	No. 243	B-30020~30021、B-30023~30025、 B-30027~30029、B-30031、B-30053~30054、 B-30057~30058、B-30061~30062
	No. 244	B-30020~30021、B-30023~30025、 B-30027~30029、B-30031、B-30053~30054、 B-30057~30058、B-30061~30062
	No. 245	B-30020~30021、B-30023~30025、 B-30027~30029、B-30031、B-30053~30054、 B-30057~30058、B-30061~30062
	No. 246	W-30007
	No. 247	B-30001
	No. 1001	B-30001、B-30100
	No. 1002	B-30101
	No. 1003	B-30102
	No. 1004	B-30103
	No. 1005	B-30104
	No. 1006	B-30105
	No. 1007	B-30106
	No. 1008~1009	B-30100
	No. 1010	B-30101~30102、B-30104、B-30105
	No. 1011	B-30102~30103、B-30105、B-30106
	No. 1012	B-30101~30106
	No. 1013	B-30103、B-30106
	No. 1014~1017	B-30101
	No. 1018	B-30101、B-30104
	No. 1019	B-30101、B-30103、B-30104、B-30106
	No. 1020~1021	B-30101、B-30104
	No. 1022	B-30101、B-30102、B-30104、B-30105
	No. 1023~1024	B-30102、B-30105

注释组号	注释号	使用处
500	No. 1025～1026	B-30101、B-30103、B-30106
	No. 1027	B-30103、B-30106
	No. 1028	B-30102、B-30105
	No. 1029	B-30105
	No. 1030	B-30102、B-30105
	No. 1031	B-30102
	No. 1032～1033	B-30103、B-30106
	No. 1034～1035	B-30102、B-30105
	No. 1036	B-30102
	No. 1037～1038	B-30103
	No. 1039～1044	B-30103、B-30106
	No. 1045	B-30101～30103、B-30105～30106
	No. 1046～1048	B-30103
	No. 1049～1051	B-30104
	No. 1052～1056	B-30106
	No. 1057	W-30004、W-30006～30007、W-30100～30102
	No. 1058	W-30007、W-30100、W-30102
	No. 1068	W-30006
	No. 1069～1071	W-30102
	No. 1072	W-30100
	No. 1073～1074	B-30104
	No. 1075～1077	B-30101、B-30104
	No. 1078	B-30104
	No. 1079	B-30101、B-30104
	No. 1080～1081	W-30102
	No. 1082	W-30100
	No. 1083～1084	B-3010～30103、B-30105～30106
	No. 1086	B-30103、B-30106
	No. 1087～1088	B-30101
	No. 1089	W-30100

5.7 软元件数据传送一览表

ID: 201 传送 1-1

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	GD11000
	触发软元件	GD11000. b0
	传送源目标互换标志	GD11000. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11001
	软元件数据传送处理中通知信号	GD11001. b0
	软元件数据传送错误通知信号	GD11001. 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	U00-G0
	传送目标软元件	GD20000
	偏置	无
块 2	软元件类型	有符号 BIN16
	点数	16
	传送源软元件	U00-G4096
	传送目标软元件	GD20016
	偏置	无
块 3	软元件类型	有符号 BIN16
	点数	129
	传送源软元件	U00-G8192
	传送目标软元件	GD20032
	偏置	无
块 4	软元件类型	有符号 BIN16
	点数	1
	传送源软元件	U00-G8960
	传送目标软元件	GD20161
	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	U00-G9216
	传送目标软元件	GD20162
	偏置	无
块 6	软元件类型	有符号 BIN16
	点数	129
	传送源软元件	U00-G9984
	传送目标软元件	GD20291
	偏置	无
块 7	软元件类型	有符号 BIN16
	点数	256
	传送源软元件	U00-G10496
	传送目标软元件	GD20420
	偏置	无
块 8	软元件类型	有符号 BIN16
	点数	256
	传送源软元件	U00-G11008
	传送目标软元件	GD20676
	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	GD11000. 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 1-2

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	GD11002
	触发软元件	GD11002. b0
	传送源目标互换标志	GD11002. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11003
	软元件数据传送处理中通知信号	GD11003. b0
	软元件数据传送错误通知信号	GD11003. b15
软元件	块数	5
块 1	软元件类型	有符号 BIN16
	点数	1
	传送源软元件	U00-G0
	传送目标软元件	GD11500
	偏置	传送源 GD10600
块 2	软元件类型	有符号 BIN16
	点数	28
	传送源软元件	U00-G20
	传送目标软元件	GD11501
	偏置	传送源 GD10600
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	GD11002. 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 1-3

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	GD11004
	触发软元件	GD11004. b0
	传送源目标互换标志	GD11004. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11005
	软元件数据传送处理中通知信号	GD11005. b0
	软元件数据传送错误通知信号	GD11005. b15
软元件	块数	3
块 1	软元件类型	有符号 BIN16
	点数	6144
	传送源软元件	U00-G12288
	传送目标软元件	GD24000
	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	GD11004. b0
	偏置	无
块 3	软元件类型	位
	点数	1
	传送源软元件	GB40
	传送目标软元件	GB30051
	偏置	无

ID: 204 传送 1-4

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	GD11006
	触发软元件	GD11006. b0
	传送源目标互换标志	GD11006. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11007
	软元件数据传送处理中通知信号	GD11007. b0
	软元件数据传送错误通知信号	GD11007. b15
软元件	块数	3
块 1	软元件类型	有符号 BIN16
	点数	19
	传送源软元件	GD11501
	传送目标软元件	U00-G1
	偏置	传送目标 GD10600
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	Y0010
	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	GD11006. b0
	偏置	无

ID: 205 传送 2-1

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	GD11008
	触发软元件	GD11008. b0
	传送源目标互换标志	GD11008. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11009
	软元件数据传送处理中通知信号	GD11009. b0
	软元件数据传送错误通知信号	GD11009. 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	U02-G0
	传送目标软元件	GD21000
	偏置	无
块 2	软元件类型	有符号 BIN16
	点数	16
	传送源软元件	U02-G4096
	传送目标软元件	GD21016
	偏置	无
块 3	软元件类型	有符号 BIN16
	点数	129
	传送源软元件	U02-G8192
	传送目标软元件	GD21032
	偏置	无
块 4	软元件类型	有符号 BIN16
	点数	1
	传送源软元件	U02-G8960
	传送目标软元件	GD21161
	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	U02-G9216
	传送目标软元件	GD21162
	偏置	无
块 6	软元件类型	有符号 BIN16
	点数	129
	传送源软元件	U02-G9984
	传送目标软元件	GD21291
	偏置	无
块 7	软元件类型	有符号 BIN16
	点数	256
	传送源软元件	U02-G10496
	传送目标软元件	GD21420
	偏置	无
块 8	软元件类型	有符号 BIN16
	点数	256
	传送源软元件	U02-G11008
	传送目标软元件	GD21676
	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	GD11008. 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

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	GD11010
	触发软元件	GD11010. b0
	传送源目标互换标志	GD11010. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11011
	软元件数据传送处理中通知信号	GD11011. b0
	软元件数据传送错误通知信号	GD11011. b15
软元件	块数	5
块 1	软元件类型	有符号 BIN16
	点数	1
	传送源软元件	U02-G0
	传送目标软元件	GD11500
	偏置	传送源 GD10600
块 2	软元件类型	有符号 BIN16
	点数	28
	传送源软元件	U02-G20
	传送目标软元件	GD11501
	偏置	传送源 GD10600
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	GD11010. 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: 207 传送 2-3

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	GD11012
	触发软元件	GD11012. b0
	传送源目标互换标志	GD11012. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11013
	软元件数据传送处理中通知信号	GD11013. b0
	软元件数据传送错误通知信号	GD11013. b15
软元件	块数	3
块 1	软元件类型	有符号 BIN16
	点数	6144
	传送源软元件	U02-G12288
	传送目标软元件	GD31000
	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	GD11012. b0
	偏置	无
块 3	软元件类型	位
	点数	1
	传送源软元件	GB40
	传送目标软元件	GB30051
	偏置	无

ID: 208 传送 2-4

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	GD11014
	触发软元件	GD11014. b0
	传送源目标互换标志	GD11014. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11015
	软元件数据传送处理中通知信号	GD11015. b0
	软元件数据传送错误通知信号	GD11015. b15
软元件	块数	3
块 1	软元件类型	有符号 BIN16
	点数	19
	传送源软元件	GD11501
	传送目标软元件	U02-G1
	偏置	传送目标 GD10600
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	Y0030
	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	GD11014. b0
	偏置	无

ID: 209 传送 3-1

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	GD11016
	触发软元件	GD11016. b0
	传送源目标互换标志	GD11016. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11017
	软元件数据传送处理中通知信号	GD11017. b0
	软元件数据传送错误通知信号	GD11017. 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	U04-G0
	传送目标软元件	GD22000
	偏置	无
块 2	软元件类型	有符号 BIN16
	点数	16
	传送源软元件	U04-G4096
	传送目标软元件	GD22016
	偏置	无
块 3	软元件类型	有符号 BIN16
	点数	129
	传送源软元件	U04-G8192
	传送目标软元件	GD22032
	偏置	无
块 4	软元件类型	有符号 BIN16
	点数	1
	传送源软元件	U04-G8960
	传送目标软元件	GD22161
	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	U04-G9216
	传送目标软元件	GD22162
	偏置	无
块 6	软元件类型	有符号 BIN16
	点数	129
	传送源软元件	U04-G9984
	传送目标软元件	GD22291
	偏置	无
块 7	软元件类型	有符号 BIN16
	点数	256
	传送源软元件	U04-G10496
	传送目标软元件	GD22420
	偏置	无
块 8	软元件类型	有符号 BIN16
	点数	256
	传送源软元件	U04-G11008
	传送目标软元件	GD22676
	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	GD11016. 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: 210 传送 3-2

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	GD11018
	触发软元件	GD11018. b0
	传送源目标互换标志	GD11018. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11019
	软元件数据传送处理中通知信号	GD11019. b0
	软元件数据传送错误通知信号	GD11019. b15
软元件	块数	5
块 1	软元件类型	有符号 BIN16
	点数	1
	传送源软元件	U04-G0
	传送目标软元件	GD11500
	偏置	传送源 GD10600
块 2	软元件类型	有符号 BIN16
	点数	28
	传送源软元件	U04-G20
	传送目标软元件	GD11501
	偏置	传送源 GD10600
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	GD11018. 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: 211 传送 3-3

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	GD11020
	触发软元件	GD11020. b0
	传送源目标互换标志	GD11020. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11021
	软元件数据传送处理中通知信号	GD11021. b0
	软元件数据传送错误通知信号	GD11021. b15
软元件	块数	3
块 1	软元件类型	有符号 BIN16
	点数	6144
	传送源软元件	U04-G12288
	传送目标软元件	GD38000
	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	GD11020. b0
	偏置	无
块 3	软元件类型	位
	点数	1
	传送源软元件	GB40
	传送目标软元件	GB30051
	偏置	无

ID: 212 传送 3-4

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	GD11022
	触发软元件	GD11022. b0
	传送源目标互换标志	GD11022. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11023
	软元件数据传送处理中通知信号	GD11023. b0
	软元件数据传送错误通知信号	GD11023. b15
软元件	块数	3
块 1	软元件类型	有符号 BIN16
	点数	19
	传送源软元件	GD11501
	传送目标软元件	U04-G1
	偏置	传送目标 GD10600
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	Y0050
	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	GD11022. b0
	偏置	无

ID: 213 传送 4-1

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	GD11024
	触发软元件	GD11024. b0
	传送源目标互换标志	GD11024. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11025
	软元件数据传送处理中通知信号	GD11025. b0
	软元件数据传送错误通知信号	GD11025. 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	U06-G0
	传送目标软元件	GD23000
	偏置	无
块 2	软元件类型	有符号 BIN16
	点数	16
	传送源软元件	U06-G4096
	传送目标软元件	GD23016
	偏置	无
块 3	软元件类型	有符号 BIN16
	点数	129
	传送源软元件	U06-G8192
	传送目标软元件	GD23032
	偏置	无
块 4	软元件类型	有符号 BIN16
	点数	1
	传送源软元件	U06-G8960
	传送目标软元件	GD23161
	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	U06-G9216
	传送目标软元件	GD23162
	偏置	无
块 6	软元件类型	有符号 BIN16
	点数	129
	传送源软元件	U06-G9984
	传送目标软元件	GD23291
	偏置	无
块 7	软元件类型	有符号 BIN16
	点数	256
	传送源软元件	U06-G10496
	传送目标软元件	GD23420
	偏置	无
块 8	软元件类型	有符号 BIN16
	点数	256
	传送源软元件	U06-G11008
	传送目标软元件	GD23676
	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	GD11024. 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: 214 传送 4-2

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	GD11026
	触发软元件	GD11026. b0
	传送源目标互换标志	GD11026. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11027
	软元件数据传送处理中通知信号	GD11027. b0
	软元件数据传送错误通知信号	GD11027. b15
软元件	块数	5
块 1	软元件类型	有符号 BIN16
	点数	1
	传送源软元件	U06-G0
	传送目标软元件	GD11500
	偏置	传送源 GD10600
块 2	软元件类型	有符号 BIN16
	点数	28
	传送源软元件	U06-G20
	传送目标软元件	GD11501
	偏置	传送源 GD10600
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	GD11026. 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: 215 传送 4-3

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	GD11028
	触发软元件	GD11028. b0
	传送源目标互换标志	GD11028. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11029
	软元件数据传送处理中通知信号	GD11029. b0
	软元件数据传送错误通知信号	GD11029. b15
软元件	块数	3
块 1	软元件类型	有符号 BIN16
	点数	6144
	传送源软元件	U06-G12288
	传送目标软元件	GD45000
	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	GD11028. b0
	偏置	无
块 3	软元件类型	位
	点数	1
	传送源软元件	GB40
	传送目标软元件	GB30051
	偏置	无

ID: 216 传送 4-4

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	GD11030
	触发软元件	GD11030. b0
	传送源目标互换标志	GD11030. b1
外部通知信息	<input checked="" type="checkbox"/> 外部通知软元件	GD11031
	软元件数据传送处理中通知信号	GD11031. b0
	软元件数据传送错误通知信号	GD11031. b15
软元件	块数	3
块 1	软元件类型	有符号 BIN16
	点数	19
	传送源软元件	GD11501
	传送目标软元件	U06-G1
	偏置	传送目标 GD10600
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	Y0070
	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	GD11030. 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	Pacage1
	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

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	Pacage1
	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-30001～30021、 B-30023～30025、 B-30027～30029、 B-30031～30042、 B-30051～30054、 B-30057～30058、 B-30061～30062、 B-30100～30106
对象脚本	有： B-30006～30021、 B-30023～30025、 B-30027～30029、 B-30031～30042、 B-30051～30054、 B-30057～30058、 B-30061～30062、 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	脚本号	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</pre>			
脚本号	30033	脚本号	Script30033
注释	状态读取		
数据类型	有符号 BIN16	触发类型	ON 中 GB30051
<pre>//Read Statuses switch([w:GD64000]) { case Master_1: set([b:GD11000.b0]); break; case Master_2: set([b:GD11008.b0]); break; case Master_3: set([b:GD11016.b0]); break; case Master_4: set([b:GD11024.b0]); break; default: break; } rst([b:GB30051]);</pre>			
脚本号	30042	脚本号	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 [w:GD54075] = 1; //0: Not Specified, 1: ASLINK, 2: CC-Link //[w:GD54089] = 0; //To use as a dedicated screen, enter I/O No. here</pre>			

```
//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]);
```

脚本号	30029	脚本号	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

```
//Observe Error and Alarm

switch([w:GD64000])
{
    case Master_1: if(([w:GD20032] != [w:U00-G8192]) || ([w:GD20291] != [w:U00-G9984]))
        {
            set([b:GD11004.b0]);
        }
        break;

    case Master_2: if(([w:GD21032] != [w:U02-G8192]) || ([w:GD21291] != [w:U02-G9984]))
        {
            set([b:GD11012.b0]);
        }
        break;

    case Master_3: if(([w:GD22032] != [w:U04-G8192]) || ([w:GD22291] != [w:U04-G9984]))
        {
            set([b:GD11020.b0]);
        }
        break;

    case Master_4: if(([w:GD23032] != [w:U06-G8192]) || ([w:GD23291] != [w:U06-G9984]))
        {
            set([b:GD11028.b0]);
        }
        break;

    default:      break;
}
```

5.9.2 画面脚本

基本画面 30001

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

```
//Control not to display the Network Map and Sensor List at the initial startup.
if([b:GB30027] == OFF)
{
    [w:TMP0000] = 0xFFFF;
```

<pre> fmov([w:TMP0000],[w:GD10100],128); fmov([w:TMP0000],[w:GD10402],48); set([b:GB30027]); } </pre>			
脚本号	30085	脚本名	Script30085
注释	传感器自动识别开始		
数据类型	有符号 BIN16	触发类型	ON 中 GB30052
<pre> //Perform the sensor auto recognition. switch([w:GD64000]) { case Master_1: set([b:Y0001]); break; case Master_2: set([b:Y0021]); break; case Master_3: set([b:Y0041]); break; case Master_4: set([b:Y0061]); break; default: break; } rst([b:GB30052]); </pre>			
脚本号	30072	脚本名	Script30072
注释	主模块 1 号 传感器自动识别		
数据类型	有符号 BIN16	触发类型	ON 中 Y0001
<pre> //Master Module No.1 //Perform the sensor auto recognition. if([b:X0011] == ON) { //When the parameter batch read flag turns on, //reset the auto recognition flag, start device data transfer. if([b:GB30044] == ON) { rst([b:Y0001]); set([b:GD11004.b0]); [w:GD62007] = 0; rst([b:GB30044]); rst([b:GB30053]); //Switch Action Conditions } } else{ //When the parameter batch read start, //turn on the flag. if([b:GB30044] == OFF) { set([b:GB30044]); } } </pre>			
脚本号	30074	脚本名	Script30074
注释	主模块 2 号 传感器自动识别		
数据类型	有符号 BIN16	触发类型	ON 中 Y0021

```
//Master Module No.2
//Perform the sensor auto recognition.
if([b:X0031] == ON)
{
    //When the parameter batch read flag turns on,
    //reset the auto recognition flag, start device data transfer.
    if([b:GB30044] == ON)
    {
        rst([b:Y0021]);
        set([b:GD11012.b0]);
        [w:GD62007] = 0;
        rst([b:GB30044]);
        rst([b:GB30053]); //Switch Action Conditions
    }
}
else{
    //When the parameter batch read start,
    //turn on the flag.
    if([b:GB30044] == OFF)
    {
        set([b:GB30044]);
    }
}
```

脚本号	30077	脚本名	Script30077
注释	主模块 3 号 传感器自动识别		
数据类型	有符号 BIN16	触发类型	ON 中 Y0041

```
//Master Module No.3
//Perform the sensor auto recognition.
if([b:X0051] == ON)
{
    //When the parameter batch read flag turns on,
    //reset the auto recognition flag, start device data transfer.
    if([b:GB30044] == ON)
    {
        rst([b:Y0041]);
        set([b:GD11020.b0]);
        [w:GD62007] = 0;
        rst([b:GB30044]);
        rst([b:GB30053]); //Switch Action Conditions
    }
}
else{
    //When the parameter batch read start,
    //turn on the flag.
    if([b:GB30044] == OFF)
    {
        set([b:GB30044]);
    }
}
```

脚本号	30080	脚本名	Script30080
注释	主模块 4 号 传感器自动识别		
数据类型	有符号 BIN16	触发类型	ON 中 Y0061

```
//Master Module No.4
//Perform the sensor auto recognition.
if([b:X0071] == ON)
{
    //When the parameter batch read flag turns on,
    //reset the auto recognition flag, start device data transfer.
    if([b:GB30044] == ON)
```

<pre> { rst([b:Y0061]); set([b:GD11028.b0]); [w:GD62007] = 0; rst([b:GB30044]); rst([b:GB30053]); //Switch Action Conditions } } else{ //When the parameter batch read start, //turn on the flag. if([b:GB30044] == OFF) { set([b:GB30044]); } } </pre>			
脚本号	30035	脚本名	Script30035
注释	主模块 1 号 软元件数据传送开始		
数据类型	有符号 BIN16	触发类型	ON 中 Y0011
<pre> //Master Module No.1 //Transfer parameters to internal devices using Device Data Transfer. if([b:X0011] == 1) { set([b:GD11004.b0]); rst([b:Y0011]); [w:GD62007] = 0; rst([b:GB30053]); //Switch Action Conditions } </pre>			
脚本号	30073	脚本名	Script30073
注释	主模块 2 号 软元件数据传送开始		
数据类型	有符号 BIN16	触发类型	ON 中 Y0031
<pre> //Master Module No.2 //Transfer parameters to internal devices using Device Data Transfer. if([b:X0031] == ON) { set([b:GD11012.b0]); rst([b:Y0031]); [w:GD62007] = 0; rst([b:GB30053]); //Switch Action Conditions } </pre>			
脚本号	30076	脚本名	Script30076
注释	主模块 3 号 软元件数据传送开始		
数据类型	有符号 BIN16	触发类型	ON 中 Y0051
<pre> //Master Module No.3 //Transfer parameters to internal devices using Device Data Transfer. if([b:X0051] == ON) { set([b:GD11020.b0]); rst([b:Y0051]); [w:GD62007] = 0; rst([b:GB30053]); //Switch Action Conditions } </pre>			

脚本号	30079	脚本名	Script30079
注释	主模块 4 号 软元件数据传送开始		
数据类型	有符号 BIN16	触发类型	ON 中 Y0071
<pre>//Master Module No.4 //Transfer parameters to internal devices using Device Data Transfer. if([b:X0071] == ON) { set([b:GD11028.b0]); rst([b:Y0071]); [w:GD62007] = 0; rst([b:GB30053]); //Switch Action Conditions }</pre>			

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:U00-G8192] != 0) ([w:U00-G9984] != 0)) { switch([w:GD64000]) { case Master_1: set([b:GD11004.b0]); break; case Master_2: set([b:GD11012.b0]); break; case Master_3: set([b:GD11020.b0]); break; case Master_4: set([b:GD11028.b0]); break; default: break; } }else{ switch([w:GD64000]) { case Master_1: set([b:GD11000.b0]); break; case Master_2: set([b:GD11008.b0]); break; case Master_3: set([b:GD11016.b0]); break; case Master_4: set([b:GD11024.b0]); break; default: break; } } }</pre>			

<pre> } 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	通常
<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:GD10233] = 0; [w:TMP0004] = 0; [w:TMP0002] = [w:GD10232]; [w:TMP0009] = 0; fmov([w:TMP0004], [w:TMP0010], 7); //Make Settings to Switch Parts Display According To Language if([w:GD62021] > 0) { [w:TMP0009] = [w:GD62021] - 1; } while([w:GD10233] < 32) { //Check Connection Status if(([w:GD20162[w:GD10601]] != 0) && ([w:TMP0002] < [w:GD20162[w:GD10601]])) { [w:TMP0003] = [w:GD10601] + [w:TMP0002]; [w:TMP0010] = [w:GD20163[w:TMP0003]]; //Connection ID [w:TMP0002] = [w:TMP0002] + 1; if(([w:TMP0010] & 0x0200) == 0) { set([b:GB30600]); //Output Bit [w:TMP0011] = ([w:TMP0010] & 0x0200) >> 9; }else{ set([b:GB30601]); //Input Bit [w:TMP0011] = ([w:TMP0010] & 0x0200) >> 9; } //Device Parameter [w:TMP0004] = ([w:TMP0010] & 0x00FF) + [w:GD10601]; //Offset to Obtain Device Parameter if([b:GB30600] == ON) { //Output [w:TMP0005] = [w:GD20420[w:TMP0004]]; }else{ //Input </pre>			

```

[w:TMP0005] = [w:GD20676[w:TMP0004]];
}
[w:TMP0014] = ([w:TMP0005] - 12288) + [w:GD10602];

//I/O Points Pattern
//[w:TMP0006] = [w:TMP0014] + 43; //Parameter Offset
[w:TMP0007] = [w:GD24043[w:TMP0014]];
[w:TMP0015] = ([w:TMP0007] & 0x00C0) >> 6; //0: Input, 1: Output, 2: Combined
[w:TMP0016] = ([w:TMP0007] & 0x003F) + 1; //Points

//Module Model Number
//[w:TMP0008] = [w:TMP0014] + 44; //Parameter Offset
[w:TMP0017] = [w:GD24044[w:TMP0014]];

//Alarm Information
if([w:GD20291[w:GD10601]] != 0)
{
    [w:TMP0004] = 0;
    [w:TMP0018] = [w:TMP0004] + [w:GD10601];
    while([w:TMP0004] < [w:GD20291[w:GD10601]])
    {
        if([w:GD20292[w:TMP0018]] == [w:TMP0010])
        {
            [w:TMP0012] = 1; //Alarm Occurrence
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
        [w:TMP0018] = [w:TMP0018] + 1;
    }
}

//Error Information
if([w:GD20032[w:GD10601]] != 0)
{
    [w:TMP0004] = 0;
    [w:TMP0018] = [w:TMP0004] + [w:GD10601];
    while([w:TMP0004] < [w:GD20032[w:GD10601]])
    {
        if([w:GD20033[w:TMP0018]] == [w:TMP0010])
        {
            [w:TMP0013] = 1; //Error Occurrence
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
        [w:TMP0018] = [w:TMP0018] + 1;
    }
}

//Display
if([w:TMP0013] == 1)
{
    //Error Occurrence
    switch([w:TMP0015])
    {

```

```

        case 0: [w:GD10100[w:GD10233]] = 30030 + [w:TMP0009]; //Input
            break;

        case 1: [w:GD10100[w:GD10233]] = 30021 + [w:TMP0009]; //Output
            break;

        case 2: [w:GD10100[w:GD10233]] = 30039 + [w:TMP0009]; //Combined
            break;
    }

    [w:GD10132[w:GD10233]] = 0;
    [w:GD10164[w:GD10233]] = [w:TMP0010] & 0x00FF; //ID
    [w:GD10196[w:GD10233]] = [w:TMP0017]; ///Module Model No.
} else {
    if([w:TMP0012] == 1)
    {
        //Alarm Occurrence
        switch([w:TMP0015])
        {
            case 0: [w:GD10100[w:GD10233]] = 30027 + [w:TMP0009]; //Input
                break;

            case 1: [w:GD10100[w:GD10233]] = 30018 + [w:TMP0009]; //Output
                break;

            case 2: [w:GD10100[w:GD10233]] = 30036 + [w:TMP0009]; //Combined
                break;

        }
        [w:GD10132[w:GD10233]] = 0;
        [w:GD10164[w:GD10233]] = [w:TMP0010] & 0x00FF; //ID
        [w:GD10196[w:GD10233]] = [w:TMP0017]; //Module Model No.
    } else {
        //Normal
        switch([w:TMP0015])
        {
            case 0: [w:GD10100[w:GD10233]] = 30024 + [w:TMP0009]; //Input
                break;

            case 1: [w:GD10100[w:GD10233]] = 30015 + [w:TMP0009]; //Output
                break;

            case 2: [w:GD10100[w:GD10233]] = 30033 + [w:TMP0009]; //Combined
                break;

        }
        [w:GD10132[w:GD10233]] = 0;
        [w:GD10164[w:GD10233]] = [w:TMP0010] & 0x00FF; //ID
        [w:GD10196[w:GD10233]] = [w:TMP0017]; //Module Model No.
    }
}

[w:GD10233] = [w:GD10233] + 1;
rst([b:GB30600]); //Output Bit
rst([b:GB30601]); //Input Bit

```

```

//Work Area Clear
[w:TMP0004] = 0;
fmov([w:TMP0004], [w:TMP0010], 5);
}else{
//If data does not exist, do not display.
[w:GD10100[w:GD10233]] = 0xFFFF;
[w:GD10132[w:GD10233]] = 0xFFFF;
[w:GD10164[w:GD10233]] = 0xFFFF;
[w:GD10196[w:GD10233]] = 0xFFFF;

[w:GD10233] = [w:GD10233] + 1;
rst([b:GB30600]); ///Output Bit
rst([b:GB30601]); //Input Bit
}
}

```

```

//Trigger Reset
rst([b:GB30000]);

```

```

//Initial Action Check Flag
set([b:GB30035]);

```

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

```

//Clear Flags
rst([b:GB30006]);
rst([b:GB30007]);
rst([b:GB30003]);
rst([b:GB30018]);
rst([b:GB30035]);

[w:TMP0004] = 0;
fmov([w:TMP0004], [w:GD12000], 512); //Initialize Object Values(B-30003, 4)
fmov([w:TMP0004], [w:GD10301], 3); //Clear Cursor Position Information(B-30003, 4)
fmov([w:TMP0004], [w:GD10432], 15); //Work Area Clear
[w:GD10300] = 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:GD10232] != 0)
{
[w:GD10232] = [w:GD10232] - 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:GD10232] + 32) < [w:U00-G8960]) && ([w:U00-G8960] > 32))

```

<pre> { [w:GD10232] = [w:GD10232] + 32; set([b:GB30000]); } </pre>			
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
<pre> //Determine detail screen to display. [w:TMP0000] = [w:GD10230] & 0x00FF; [w:TMP0002] = [w:TMP0000] + [w:GD10601]; if(([w:GD10231] >= 30015) && ([w:GD10231] <= 30023)) { //Output [w:GD10600] = [w:GD20420[w:TMP0002]]; //Offset [w:TMP0001] = [w:TMP0000]; }else{ //Input [w:GD10600] = [w:GD20676[w:TMP0002]]; //Offset [w:TMP0001] = [w:TMP0000] + 0x0200; } [w:U00-G10320] = 0; //Read Individual Parameter [w:U00-G10321] = [w:TMP0001]; //Read Target ID //Select base screen to display. switch([w:GD10196[w:GD10228]] & 0xF000) { case 0x0000: [w:TMP0003] = 0; break; case 0x1000: [w:TMP0003] = 1; break; case 0x2000: [w:TMP0003] = 2; break; case 0x3000: [w:TMP0003] = 3; break; case 0xA000: [w:TMP0003] = 10; break; } if([w:TMP0003] == 0) { switch([w:GD10196[w:GD10228]]) { case 0x0100: [w:GD62000] = 30006; break; case 0x0101: [w:GD62000] = 30006; break; } } </pre>			

```
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;
}
}

if([w:TMP0003] == 1)
{
    switch([w:GD10196[w:GD10228]])
    {
        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 0x1401: [w:GD62000] = 30019;
            break;

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

        case 0x1451: [w:GD62000] = 30019;
            break;

        case 0x1500: [w:GD62000] = 30043;
            break;

        // case 0x1503: [w:GD62000] = 30046;
        //         break;

        case 0x1553: [w:GD62000] = 30050;
            break;
    }
}

if([w:TMP0003] == 2)
{
    switch([w:GD10196[w:GD10228]])
    {

```

```
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 0x2601: [w:GD62000] = 30017;
    break;

case 0x2602: [w:GD62000] = 30017;
    break;

case 0x2603: [w:GD62000] = 30017;
    break;

case 0x2900: [w:GD62000] = 30020;
    break;

case 0x2903: [w:GD62000] = 30021;
    break;

case 0x2909: [w:GD62000] = 30023;
    break;

case 0x290C: [w:GD62000] = 30053;
    break;

case 0x290F: [w:GD62000] = 30054;
    break;

case 0x2930: [w:GD62000] = 30024;
    break;

case 0x2933: [w:GD62000] = 30025;
    break;

case 0x2939: [w:GD62000] = 30027;
    break;

case 0x293C: [w:GD62000] = 30057;
    break;

case 0x293F: [w:GD62000] = 30058;
```



```

        break;

    case 0x2960: [w:GD62000] = 30028;
        break;

    case 0x2963: [w:GD62000] = 30029;
        break;

    case 0x2969: [w:GD62000] = 30031;
        break;

    case 0x293C: [w:GD62000] = 30061;
        break;

    case 0x293F: [w:GD62000] = 30062;
        break;
    }
}

if([w:TMP0003] == 3)
{
    switch([w:GD10196[w:GD10228]])
    {

        case 0x3100: [w:GD62000] = 30032;
            break;

        case 0x3300: [w:GD62000] = 30033;
            break;

        case 0x3301: [w:GD62000] = 30034;
            break;

        case 0x3302: [w:GD62000] = 30035;
            break;

        case 0x3303: [w:GD62000] = 30035;
            break;

        case 0x3304: [w:GD62000] = 30036;
            break;

        case 0x3305: [w:GD62000] = 30036;
            break;
    }
}

if([w:TMP0003] == 10)
{
    switch([w:GD10196[w:GD10228]])
    {

        case 0xA000: [w:GD62000] = 30018;
            break;

        case 0xA001: [w:GD62000] = 30018;

```

```
        break;

    case 0xA002: [w:GD62000] = 30037;
        break;

    case 0xA003: [w:GD62000] = 30037;
        break;

    case 0xA004: [w:GD62000] = 30038;
        break;

    case 0xA005: [w:GD62000] = 30038;
        break;

    case 0xA006: [w:GD62000] = 30039;
        break;

    case 0xA007: [w:GD62000] = 30039;
        break;

    case 0xA010: [w:GD62000] = 30051;
        break;

    case 0xA300: [w:GD62000] = 30018;
        break;

    case 0xA301: [w:GD62000] = 30018;
        break;

    case 0xA302: [w:GD62000] = 30037;
        break;

    case 0xA303: [w:GD62000] = 30037;
        break;

    case 0xA304: [w:GD62000] = 30038;
        break;

    case 0xA305: [w:GD62000] = 30038;
        break;

    case 0xA306: [w:GD62000] = 30039;
        break;

    case 0xA307: [w:GD62000] = 30039;
        break;

    case 0xA308: [w:GD62000] = 30040;
        break;

    case 0xA309: [w:GD62000] = 30040;
        break;

    case 0xA310: [w:GD62000] = 30052;
        break;
```

<pre> case 0xA320: [w:GD62000] = 30041; break; case 0xA321: [w:GD62000] = 30042; break; case 0xA322: [w:GD62000] = 30042; break; case 0xA323: [w:GD62000] = 30042; break; case 0xA324: [w:GD62000] = 30042; break; case 0xA325: [w:GD62000] = 30042; break; case 0xA326: [w:GD62000] = 30042; break; case 0xA600: [w:GD62000] = 30018; break; case 0xA601: [w:GD62000] = 30018; break; case 0xA602: [w:GD62000] = 30037; break; case 0xA603: [w:GD62000] = 30037; break; case 0xA604: [w:GD62000] = 30038; break; case 0xA605: [w:GD62000] = 30038; break; case 0xA606: [w:GD62000] = 30039; break; case 0xA607: [w:GD62000] = 30039; break; } } rst([b:GB30006]); </pre>			
脚本号	30083	脚本名	Script30083
注释	报警复位		
数据类型	有符号 BIN16	触发类型	上升沿 GB30049
<pre> //Turn ON the alarm reset signal. switch([w:GD64000]) { case Master_1: set([b:Y0000]); </pre>			

<pre> break; case Master_2: set([b:Y0020]); break; case Master_3: set([b:Y0040]); break; case Master_4: set([b:Y0060]); break; default: break; } set([b:GB30050]); rst([b:GB30049]); </pre>			
脚本号	30084	脚本名	Script30084
注释	报警清除触发复位		
数据类型	有符号 BIN16	触发类型	ON 中周期 1 秒 GB30050
<pre> //Turn OFF the alarm reset signal. switch([w:GD64000]) { case Master_1: rst([b:Y0000]); break; case Master_2: rst([b:Y0020]); break; case Master_3: rst([b:Y0040]); break; case Master_4: rst([b:Y0060]); break; default: break; } rst([b:GB30050]); </pre>			

基本画面 30003

脚本号.	30004	スクリプト名	Script30004
注释	B-30003, 30004 画面初始化		
数据类型	有符号 BIN16	触发类型	ON 中 GB40
<pre> f([b:GB30003] == OFF) { fmov([w:TMP0000], [w:TMP0020], 10); [w:TMP0110] = 0; //Initialize Temporary Area for Comparison [w:TMP0111] = 0; //Initialize Temporary Area for Comparison //Obtain Parameters and Display Parts switch([w:GD64000]) { </pre>			

<pre> case Master_1: set([b:GD11000.b0]); break; case Master_2: set([b:GD11008.b0]); break; case Master_3: set([b:GD11016.b0]); break; case Master_4: set([b:GD11024.b0]); break; default: break; } 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	通常
<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, 30004 部件显示		
数据类型	无符号 BIN16	触发类型	ON 中 GB30001
<pre> //Display Parts [w:TMP0015] = 0; [w:TMP0008] = 0; fmov([w:TMP0008], [w:TMP0020], 6); while(([w:TMP0015] < [w:GD20161[w:GD10601]]) && ([w:GD20161[w:GD10601]] > 0)) { //Obtain Connection ID [w:TMP0000] = [w:GD10601] + [w:TMP0015]; [w:TMP0020] = [w:GD20163[w:TMP0000]]; [w:TMP0015] = [w:TMP0015] + 1; if(([w:TMP0020] & 0x0200) == 0) { set([b:GB30600]); //Output Bit }else{ set([b:GB30601]); //Input Bit } //Alarm Information if([w:GD20291[w:GD10601]] != 0) { [w:TMP0004] = 0; [w:TMP0001] = [w:GD10601] + [w:TMP0004]; while([w:TMP0004] < [w:GD20291[w:GD10601]]) { </pre>			

```

        if([w:GD20292[w:TMP0001]] == [w:TMP0020])
        {
            [w:TMP0022] = 1; //Alarm Occurrence
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
        [w:TMP0001] = [w:TMP0001] + 1;
    }
}

//Error Information
if([w:GD20032[w:GD10601]] != 0)
{
    [w:TMP0004] = 0;
    [w:TMP0001] = [w:GD10601] + [w:TMP0004];
    while([w:TMP0004] < [w:GD20032[w:GD10601]])
    {
        if([w:GD20033[w:TMP0001]] == [w:TMP0020])
        {
            [w:TMP0023] = 1; //Error Occurrence
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
        [w:TMP0001] = [w:TMP0001] + 1;
    }
}

//Device Parameter
[w:TMP0004] = [w:TMP0020] & 0x00FF; //Offset to Obtain Device Parameter
[w:TMP0002] = [w:GD10601] + [w:TMP0004];
if([b:GB30600] == ON)
{
    //Output
    [w:TMP0005] = [w:GD20420[w:TMP0002]];
    //[w:TMP0012] = [w:TMP0004] + [w:GD10601];
    [w:TMP0012] = [w:TMP0004];
}else{
    //Input
    [w:TMP0005] = [w:GD20676[w:TMP0002]];
    //[w:TMP0012] = ([w:TMP0004] + 0x0100) + [w:GD10601];
    [w:TMP0012] = [w:TMP0004] + 0x0100;
}

[w:TMP0024] = ([w:TMP0005] - 0x3000) + [w:GD10602];

//I/O Points Pattern
//[w:TMP0006] = [w:TMP0024] + 43; //Parameter Offset
[w:TMP0007] = [w:GD24043[w:TMP0024]];
[w:TMP0025] = ([w:TMP0007] & 0x00C0) >> 6; //0: Input, 1: Output, 2: Combined
[w:TMP0026] = ([w:TMP0007] & 0x003F) + 1; //Points

//Module Model Number
//[w:TMP0006] = [w:TMP0024] + 44;
[w:GD16000[w:TMP0012]] = [w:GD24044[w:TMP0024]];

//ON/OFF Information

```

```

[w:TMP0008] = [w:TMP0020] & 0x00FF;
if([w:TMP0025] == 0)    //Input Case
{
    [w:TMP0004] = 0;

    while([w:TMP0004] < [w:TMP0026])
    {
        [w:TMP0009] = [w:TMP0008] / 16;
        [w:TMP0010] = [w:TMP0008] % 16;

        if([w:TMP0023] == 1)    //Check If Error Occurred
        {
            [w:GD12256[w:TMP0008]] = 3; //Error Occurrence & Input ON
        }else{
            if([w:TMP0022] == 1)    //Check If Alarms Occurred
            {
                [w:GD12256[w:TMP0008]] = 2; //Alarm Occurrence & Input ON
            }else{
                [w:GD12256[w:TMP0008]] = 1; //Normal
            }
        }

        [w:TMP0008] = [w:TMP0008] + 1;
        [w:TMP0004] = [w:TMP0004] + 1;
    }
}else{
    if([w:TMP0025] == 1)    //Output Case
    {
        [w:TMP0004] = 0;

        while([w:TMP0004] < [w:TMP0026])
        {
            [w:TMP0009] = [w:TMP0008] / 16;
            [w:TMP0010] = [w:TMP0008] % 16;

            if([w:TMP0023] == 1)    //Check If Error Occurred
            {
                [w:GD12000[w:TMP0008]] = 3; //Error Occurrence & Input ON
            }else{
                if([w:TMP0022] == 1)    //Check If Alarms Occurred
                {
                    [w:GD12000[w:TMP0008]] = 2; //Alarm Occurrence & Input ON
                }else{
                    [w:GD12000[w:TMP0008]] = 1; //Normal
                }
            }

            [w:TMP0008] = [w:TMP0008] + 1;
            [w:TMP0004] = [w:TMP0004] + 1;
        }
    }
}else{
    if([w:TMP0025] == 2)    //Combined Case
    {
        [w:TMP0004] = 0;
        [w:TMP0011] = [w:TMP0026] / 2;    //If combined, points are reduced by half.
        while([w:TMP0004] < [w:TMP0011])

```

```

{
    [w:TMP0009] = [w:TMP0008] / 16;
    [w:TMP0010] = [w:TMP0008] % 16;

    //Output
    if([w:TMP0023] == 1)    //Check If Error Occurred
    {
        [w:GD12000[w:TMP0008]] = 3; //Error Occurrence & Input ON
    }else{
        if([w:TMP0022] == 1)    //Check If Alarms Occurred
        {
            [w:GD12000[w:TMP0008]] = 2; //Alarm Occurrence & Input ON
        }else{
            [w:GD12000[w:TMP0008]] = 1; //Normal
        }
    }

    //Input
    if([w:TMP0023] == 1)    //Check If Error Occurred
    {
        [w:GD12256[w:TMP0008]] = 3; //Error Occurrence & Input ON
    }else{
        if([w:TMP0022] == 1)    //Check If Alarms Occurred
        {
            [w:GD12256[w:TMP0008]] = 2; //Alarm Occurrence & Input ON
        }else{
            [w:GD12256[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:TMP0020], 5);
}

//Display Cursor
if([w:GD62000] == 30004)
{
    //Output
    [w:GD10301] = 0; //Left Right
    [w:GD10302] = 0; //Up Down
    [w:GD10303] = [w:GD10301] + [w:GD10302];
    [w:GD10300] = [w:GD10303]; //Cursor Position Display
    //[w:TMP0030] = [w:GD10303] + [w:GD10601];

    if([w:GD12000[w:GD10303]] < 4)

```



```

{
    [w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
}
}else{
    //Input
    [w:GD10301] = 256; //Left Right
    [w:GD10302] = 0; //Up Down
    [w:GD10303] = [w:GD10301] + [w:GD10302];
    [w:GD10300] = [w:GD10303] - 256; //Cursor Position Display
    //[w:TMP0030] = [w:GD10303] + [w:GD10601];

    if([w:GD12000[w:GD10303]] < 4)
    {
        [w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
    }
}

```

rst([b:GB30001]);

//Initial Action Check Flag

set([b:GB30035]);

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

//Clear Flags

rst([b:GB30006]);

rst([b:GB30007]);

rst([b:GB30003]);

rst([b:GB30018]);

rst([b:GB30035]);

[w:TMP0004] = 0;

fmov([w:TMP0004], [w:GD12000], 512); //Initialize Object Values(B-30003, 4)

fmov([w:TMP0004], [w:GD10301], 3); //Clear Cursor Position Information(B-30003, 4)

fmov([w:TMP0004], [w:GD10432], 15); //Work Area Clear

[w:GD10300] = 0; //Cursor Position Information Clear(B-30003, 4)

//Close Overlap Window

[w:GD62001] = 0;

[w:GD62004] = 0;

[w:GD62007] = 0;

脚本号.	30017	脚本名	Script30017
注释	B-30003, 30004 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:GD10303] < 256)

{

 [w:TMP0000] = [w:GD10303]; //Output ID

}else{

 [w:TMP0000] = [w:GD10303] - 256; //Input ID

}

if([w:GD12000[w:GD10303]] > 4) //Judge whether a module exists in the cursor position.

{

```

if([w:GD10303] < 256)
{
    //Output
    [w:TMP0002] = [w:TMP0000];
    while([w:TMP0002] >= 0)
    {
        if([w:GD12000[w:TMP0002]] != 0)
        {
            //Output
            [w:TMP0001] = 0;
            [w:TMP0005] = [w:GD10601];
            while([w:TMP0001] < [w:GD20162[w:GD10601]]) //Repeat the Number of Connection ID
Counts
            {
                if([w:GD20163[w:TMP0005]] == [w:TMP0002]) //Judge ID
                {
                    set([b:GB30009]); //ID Match Flag
                    break;
                }else{
                    [w:TMP0001] = [w:TMP0001] + 1;
                    [w:TMP0005] = [w:TMP0005] + 1;
                }
            }

            //Combined Output Side Case
            if([b:GB30009] == OFF)
            {
                [w:TMP0001] = 0;
                [w:TMP0005] = [w:GD10601];
                while([w:TMP0001] < [w:GD20162[w:GD10601]]) //Repeat the Number of Connection ID
Counts
                {
                    if([w:GD20163[w:TMP0005]] == ([w:TMP0002] + 0x0200)) //Judge ID
                    {

                        //Judge whether the matched ID is that of a combined module or not.
                        [w:TMP0006] = [w:GD10601] + [w:TMP0002];
                        [w:TMP0010] = [w:GD20676[w:TMP0006]]; //Parameter Top Address
                        [w:TMP0011] = ([w:TMP0010] - 0x3000) + [w:GD10602];

                        [w:TMP0012] = [w:GD24043[w:TMP0011]];
                        if((((w:TMP0012] & 0x00C0) >> 6) == 2)
                        {
                            set([b:GB30009]); //ID Match Flag
                            set([b:GB30020]); //Combined Output Flag
                            break;
                        }
                        [w:TMP0001] = [w:TMP0001] + 1;
                        [w:TMP0005] = [w:TMP0005] + 1;
                    }else{
                        [w:TMP0001] = [w:TMP0001] + 1;
                        [w:TMP0005] = [w:TMP0005] + 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:TMP0001] = 0;
        [w:TMP0005] = [w:GD10601];
        while([w:TMP0001] < [w:GD20162[w:GD10601]]) //Repeat the Number of Connection ID Counts
        {
            if([w:GD20163[w:TMP0005]] == ([w:TMP0002] + 0x0200)) //Judge ID
            {
                set([b:GB30009]); //ID Match Flag
                break;
            }else{
                [w:TMP0001] = [w:TMP0001] + 1;
                [w:TMP0005] = [w:TMP0005] + 1;
            }
        }

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

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

//If matched, display screen.
if([b:GB30009] == ON)
{
    if([w:GD10303] < 256)
    {
        [w:TMP0007] = [w:GD10601] + [w:TMP0002];

        if([b:GB30020] == OFF)
        {
            //Output
            [w:GD10600] = [w:GD20420[w:TMP0007]]; //Offset
            [w:TMP0003] = [w:TMP0002];
            //[w:TMP0004] = [w:TMP0002] + [w:GD10601];
        }
    }
}

```

```

        [w:TMP0004] = [w:TMP0002];
    }else{
        //If combined, refer to the input side.
        [w:GD10600] = [w:GD20676[w:TMP0007]]; //Offset
        [w:TMP0003] = [w:TMP0002] + 0x0200;
        //[w:TMP0004] = ([w:TMP0002] + 0x0100) + [w:GD10601];
        [w:TMP0004] = [w:TMP0002] + 0x0100;
    }
}
}else{
    //Input
    [w:TMP0007] = [w:GD10601] + [w:TMP0002];
    [w:GD10600] = [w:GD20676[w:TMP0007]]; //Offset
    [w:TMP0003] = [w:TMP0002] + 0x0200;
    //[w:TMP0004] = ([w:TMP0002] + 0x0100) + [w:GD10601];
    [w:TMP0004] = [w:TMP0002] + 0x0100;
}

[w:U00-G10320] = 0; //Read Individual Parameter
[w:U00-G10321] = [w:TMP0003]; //Read Target ID

//Select base screen to display.
switch([w:GD16000[w:TMP0004]] & 0xF000)
{
    case 0x0000: [w:TMP0008] = 0;
                break;

    case 0x1000: [w:TMP0008] = 1;
                break;

    case 0x2000: [w:TMP0008] = 2;
                break;

    case 0x3000: [w:TMP0008] = 3;
                break;

    case 0xA000: [w:TMP0008] = 10;
                break;

}

if([w:TMP0008] == 0)
{
    switch([w:GD16000[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;
    }
}

if([w:TMP0008] == 1)
{
    switch([w:GD16000[w:TMP0004]])
    {
        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 0x1401: [w:GD62000] = 30019;
            break;

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

        case 0x1451: [w:GD62000] = 30019;
            break;

        case 0x1500: [w:GD62000] = 30043;
            break;

        // case 0x1503: [w:GD62000] = 30046;
        //         break;

        case 0x1553: [w:GD62000] = 30050;
            break;
    }
}

if([w:TMP0008] == 2)
{
    switch([w:GD16000[w:TMP0004]])
    {

        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 0x2601: [w:GD62000] = 30017;
    break;

case 0x2602: [w:GD62000] = 30017;
    break;

case 0x2603: [w:GD62000] = 30017;
    break;

case 0x2900: [w:GD62000] = 30020;
    break;

case 0x2903: [w:GD62000] = 30021;
    break;

case 0x2909: [w:GD62000] = 30023;
    break;

case 0x290C: [w:GD62000] = 30053;
    break;

case 0x290F: [w:GD62000] = 30054;
    break;

case 0x2930: [w:GD62000] = 30024;
    break;

case 0x2933: [w:GD62000] = 30025;
    break;

case 0x2939: [w:GD62000] = 30027;
    break;

case 0x293C: [w:GD62000] = 30057;
    break;

case 0x293F: [w:GD62000] = 30058;
    break;

case 0x2960: [w:GD62000] = 30028;
    break;

case 0x2963: [w:GD62000] = 30029;
    break;
```

```

        case 0x2969: [w:GD62000] = 30031;
            break;

        case 0x293C: [w:GD62000] = 30061;
            break;

        case 0x293F: [w:GD62000] = 30062;
            break;
    }
}

if([w:TMP0008] == 3)
{
    switch([w:GD16000[w:TMP0004]])
    {

        case 0x3100: [w:GD62000] = 30032;
            break;

        case 0x3300: [w:GD62000] = 30033;
            break;

        case 0x3301: [w:GD62000] = 30034;
            break;

        case 0x3302: [w:GD62000] = 30035;
            break;

        case 0x3303: [w:GD62000] = 30035;
            break;

        case 0x3304: [w:GD62000] = 30036;
            break;

        case 0x3305: [w:GD62000] = 30036;
            break;
    }
}

if([w:TMP0008] == 10)
{
    switch([w:GD16000[w:TMP0004]])
    {

        case 0xA000: [w:GD62000] = 30018;
            break;

        case 0xA001: [w:GD62000] = 30018;
            break;

        case 0xA002: [w:GD62000] = 30037;
            break;

        case 0xA003: [w:GD62000] = 30037;
            break;
    }
}

```



```
case 0xA004: [w:GD62000] = 30038;
    break;

case 0xA005: [w:GD62000] = 30038;
    break;

case 0xA006: [w:GD62000] = 30039;
    break;

case 0xA007: [w:GD62000] = 30039;
    break;

case 0xA010: [w:GD62000] = 30051;
    break;

case 0xA300: [w:GD62000] = 30018;
    break;

case 0xA301: [w:GD62000] = 30018;
    break;

case 0xA302: [w:GD62000] = 30037;
    break;

case 0xA303: [w:GD62000] = 30037;
    break;

case 0xA304: [w:GD62000] = 30038;
    break;

case 0xA305: [w:GD62000] = 30038;
    break;

case 0xA306: [w:GD62000] = 30039;
    break;

case 0xA307: [w:GD62000] = 30039;
    break;

case 0xA308: [w:GD62000] = 30040;
    break;

case 0xA309: [w:GD62000] = 30040;
    break;

case 0xA310: [w:GD62000] = 30052;
    break;

case 0xA320: [w:GD62000] = 30041;
    break;

case 0xA321: [w:GD62000] = 30042;
    break;

case 0xA322: [w:GD62000] = 30042;
```

```

        break;

    case 0xA323: [w:GD62000] = 30042;
        break;

    case 0xA324: [w:GD62000] = 30042;
        break;

    case 0xA325: [w:GD62000] = 30042;
        break;

    case 0xA326: [w:GD62000] = 30042;
        break;

    case 0xA600: [w:GD62000] = 30018;
        break;

    case 0xA601: [w:GD62000] = 30018;
        break;

    case 0xA602: [w:GD62000] = 30037;
        break;

    case 0xA603: [w:GD62000] = 30037;
        break;

    case 0xA604: [w:GD62000] = 30038;
        break;

    case 0xA605: [w:GD62000] = 30038;
        break;

    case 0xA606: [w:GD62000] = 30039;
        break;

    case 0xA607: [w:GD62000] = 30039;
        break;
    }
}
//Flag Reset
rst([b:GB30009]);
rst([b:GB30020]);
}

rst([b:GB30002]);

```

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

```
//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:GD11000.b0] == OFF) && ([b:GB30001] == OFF))
{
//Save Current Cursor Position
[w:TMP0122] = [w:GD10303];
//[w:TMP0127] = [w:TMP0122] + [w:GD10601];

//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:GD10301] = (16 - ([w:TMP0124] + 1)) + 256;

//Y Coordinate
[w:TMP0125] = [w:TMP0121] - Input_Y;
[w:TMP0126] = [w:TMP0125] / Object_Y;
[w:GD10302] = [w:TMP0126] * 16;

//Offset
[w:GD10303] = [w:GD10301] + [w:GD10302];
//[w:TMP0128] = [w:GD10303] + [w:GD10601];

//If the offset position does not change, displayed parts do not change.
if([w:TMP0122] != [w:GD10303])
{
//Cursor Display
if([w:GD12000[w:GD10303]] < 4)
{
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
}

//Cursor Delete
if([w:GD12000[w:TMP0122]] < 4)
{
[w:GD12000[w:TMP0122]] = 0;
}else{
[w:GD12000[w:TMP0122]] = [w:GD12000[w:TMP0122]] - 4;
}
}
}

[w:GD10300] = [w:GD10303] - 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

```

//Control Parts Display

//Calculate Cursor Position
[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position
//[w:TMP0021] = [w:TMP0020] + [w:GD10601];

if([w:GD10301] == 271)

```

```

{
    [w:GD10301] = 256;
}else{
    [w:GD10301] = [w:GD10301] + 1;
}

[w:TMP0000] = [w:GD10301];

[w:GD10303] = [w:TMP0000] + [w:GD10302]; //Offset
//[w:TMP0022] = [w:GD10303] + [w:GD10601];

if([w:GD12000[w:TMP0020]] >= 4)
{
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;
}else{
    [w:GD12000[w:TMP0020]] = 0;
}
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;

[w:GD10300] = [w:GD10303] - 256; //Cursor Position Display

```

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

```

//Control Parts Display

//Calculate Cursor Position
[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position
//[w:TMP0021] = [w:TMP0020] + [w:GD10601];

if([w:GD10301] == 256)
{
    [w:GD10301] = 271;
}else{
    [w:GD10301] = [w:GD10301] -1;
}

[w:TMP0000] = [w:GD10301];

[w:GD10303] = [w:TMP0000] + [w:GD10302]; //Offset
//[w:TMP0022] = [w:GD10303] + [w:GD10601];

if([w:GD12000[w:TMP0020]] >= 4)
{
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;
}else{
    [w:GD12000[w:TMP0020]] = 0;
}
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;

[w:GD10300] = [w:GD10303] - 256; //Cursor Position Display

```

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

```

//Control Parts Display

//Calculate Cursor Position

```

```
[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position
```

```
//[w:TMP0021] = [w:TMP0020] + [w:GD10601];
```

```
if([w:GD10302] == 240)
```

```
{
```

```
    [w:GD10302] = 0;
```

```
}else{
```

```
    [w:GD10302] = [w:GD10302] + 16;
```

```
}
```

```
[w:GD10303] = [w:GD10301] + [w:GD10302]; //Offset
```

```
//[w:TMP0022] = [w:GD10303] + [w:GD10601];
```

```
if([w:GD12000[w:TMP0020]] >= 4)
```

```
{
```

```
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;
```

```
}else{
```

```
    [w:GD12000[w:TMP0020]] = 0;
```

```
}
```

```
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
```

```
[w:GD10300] = [w:GD10303] - 256; //Cursor Position Display
```

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
------	-----------	------	-------------

```
//Control Parts Display
```

```
//Calculate Cursor Position
```

```
[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position
```

```
//[w:TMP0021] = [w:TMP0020] + [w:GD10601];
```

```
if([w:GD10302] == 0)
```

```
{
```

```
    [w:GD10302] = 240;
```

```
}else{
```

```
    [w:GD10302] = [w:GD10302] - 16;
```

```
}
```

```
[w:GD10303] = [w:GD10301] + [w:GD10302]; //Offset
```

```
//[w:TMP0022] = [w:GD10303] + [w:GD10601];
```

```
if([w:GD12000[w:TMP0020]] >= 4)
```

```
{
```

```
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;
```

```
}else{
```

```
    [w:GD12000[w:TMP0020]] = 0;
```

```
}
```

```
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
```

```
[w:GD10300] = [w:GD10303] - 256; //Reduce Cursor Position Display Value.
```

脚本号.	30083	脚本名	Script30083
------	-------	-----	-------------

注释	报警复位
----	------

数据类型	有符号 BIN16	触发类型	上升沿 GB30049
------	-----------	------	-------------

```
//Turn ON the alarm reset signal.
```

```
switch([w:GD64000])
```

```

{
    case Master_1: set([b:Y0000]);
        break;

    case Master_2: set([b:Y0020]);
        break;

    case Master_3: set([b:Y0040]);
        break;

    case Master_4: set([b:Y0060]);
        break;

    default:      break;
}
set([b:GB30050]);

rst([b:GB30049]);

```

脚本号.	30084	脚本名	Script30084
注释	报警清除触发复位		
数据类型	有符号 BIN16	触发类型	ON 中周期 1 秒 GB30050

```
//Turn OFF the alarm reset signal.
```

```

switch([w:GD64000])
{
    case Master_1: rst([b:Y0000]);
        break;

    case Master_2: rst([b:Y0020]);
        break;

    case Master_3: rst([b:Y0040]);
        break;

    case Master_4: rst([b:Y0060]);
        break;

    default:      break;
}

rst([b:GB30050]);

```

*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

```

if([b:GB30003] == OFF)
{

    fmov([w:TMP0000], [w:TMP0020], 10);
    [w:TMP0110] = 0; //Initialize Temporary Area for Comparison
    [w:TMP0111] = 0; //Initialize Temporary Area for Comparison
}

```

```
//Obtain Parameters and Display Parts
switch([w:GD64000])
{
    case Master_1: set([b:GD11000.b0]);
        break;

    case Master_2: set([b:GD11008.b0]);
        break;

    case Master_3: set([b:GD11016.b0]);
        break;

    case Master_4: set([b:GD11024.b0]);
        break;

    default:      break;
}

set([b:GB30003]); //Turn on the flag not to work when displaying the screen for the second time
or later.
}
```

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	通常

```
//After reading the status, start Script No.30003.
if([b:GB30016] == OFF)
{
    set([b:GB30001]);
    set([b:GB30016]);
}
```

脚本号.	30003	脚本名	Script30003
注释	B-30003, 30004 部件显示		
数据类型	无符号 BIN16	触发类型	ON 中 GB30001

```
//Display Parts
[w:TMP0015] = 0;
[w:TMP0008] = 0;
fmov([w:TMP0008], [w:TMP0020], 6);

while(([w:TMP0015] < [w:GD20161[w:GD10601]]) && ([w:GD20161[w:GD10601]] > 0))
{
    //Obtain Connection ID
    [w:TMP0000] = [w:GD10601] + [w:TMP0015];
    [w:TMP0020] = [w:GD20163[w:TMP0000]];
    [w:TMP0015] = [w:TMP0015] + 1;

    if(([w:TMP0020] & 0x0200) == 0)
    {
        set([b:GB30600]); //Output Bit
    }else{
        set([b:GB30601]); //Input Bit
    }

    //Alarm Information
    if([w:GD20291[w:GD10601]] != 0)
    {
        [w:TMP0004] = 0;
    }
}
```

```

[w:TMP0001] = [w:GD10601] + [w:TMP0004];
while([w:TMP0004] < [w:GD20291[w:GD10601]])
{
    if([w:GD20292[w:TMP0001]] == [w:TMP0020])
    {
        [w:TMP0022] = 1; //Alarm Occurrence
        break;
    }
    [w:TMP0004] = [w:TMP0004] + 1;
    [w:TMP0001] = [w:TMP0001] + 1;
}
}

//Error Information
if([w:GD20032[w:GD10601]] != 0)
{
    [w:TMP0004] = 0;
    [w:TMP0001] = [w:GD10601] + [w:TMP0004];
    while([w:TMP0004] < [w:GD20032[w:GD10601]])
    {
        if([w:GD20033[w:TMP0001]] == [w:TMP0020])
        {
            [w:TMP0023] = 1; //Error Occurrence
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
        [w:TMP0001] = [w:TMP0001] + 1;
    }
}

//Device Parameter
[w:TMP0004] = [w:TMP0020] & 0x00FF; //Offset to Obtain Device Parameter
[w:TMP0002] = [w:GD10601] + [w:TMP0004];
if([b:GB30600] == ON)
{
    //Output
    [w:TMP0005] = [w:GD20420[w:TMP0002]];
    //[w:TMP0012] = [w:TMP0004] + [w:GD10601];
    [w:TMP0012] = [w:TMP0004];
}else{
    //Input
    [w:TMP0005] = [w:GD20676[w:TMP0002]];
    //[w:TMP0012] = ([w:TMP0004] + 0x0100) + [w:GD10601];
    [w:TMP0012] = [w:TMP0004] + 0x0100;
}

[w:TMP0024] = ([w:TMP0005] - 0x3000) + [w:GD10602];

//I/O Points Pattern
//[w:TMP0006] = [w:TMP0024] + 43; //Parameter Offset
[w:TMP0007] = [w:GD24043[w:TMP0024]];
[w:TMP0025] = ([w:TMP0007] & 0x00C0) >> 6; //0: Input, 1: Output, 2: Combined
[w:TMP0026] = ([w:TMP0007] & 0x003F) + 1; //Points

//Module Model Number
//[w:TMP0006] = [w:TMP0024] + 44;

```



```

[w:GD16000[w:TMP0012]] = [w:GD24044[w:TMP0024]];

//ON/OFF Information
[w:TMP0008] = [w:TMP0020] & 0x00FF;
if([w:TMP0025] == 0) //Input Case
{
    [w:TMP0004] = 0;

    while([w:TMP0004] < [w:TMP0026])
    {
        [w:TMP0009] = [w:TMP0008] / 16;
        [w:TMP0010] = [w:TMP0008] % 16;

        if([w:TMP0023] == 1) //Check If Error Occurred
        {
            [w:GD12256[w:TMP0008]] = 3; //Error Occurrence & Input ON
        }else{
            if([w:TMP0022] == 1) //Check If Alarms Occurred
            {
                [w:GD12256[w:TMP0008]] = 2; //Alarm Occurrence & Input ON
            }else{
                [w:GD12256[w:TMP0008]] = 1; //Normal
            }
        }

        [w:TMP0008] = [w:TMP0008] + 1;
        [w:TMP0004] = [w:TMP0004] + 1;
    }
}else{
    if([w:TMP0025] == 1) //Output Case
    {
        [w:TMP0004] = 0;

        while([w:TMP0004] < [w:TMP0026])
        {
            [w:TMP0009] = [w:TMP0008] / 16;
            [w:TMP0010] = [w:TMP0008] % 16;

            if([w:TMP0023] == 1) //Check If Error Occurred
            {
                [w:GD12000[w:TMP0008]] = 3; //Error Occurrence & Input ON
            }else{
                if([w:TMP0022] == 1) //Check If Alarms Occurred
                {
                    [w:GD12000[w:TMP0008]] = 2; //Alarm Occurrence & Input ON
                }else{
                    [w:GD12000[w:TMP0008]] = 1; //Normal
                }
            }

            [w:TMP0008] = [w:TMP0008] + 1;
            [w:TMP0004] = [w:TMP0004] + 1;
        }
    }
}else{
    if([w:TMP0025] == 2) //Combined Case
    {

```

```

[w:TMP0004] = 0;
[w:TMP0011] = [w:TMP0026] / 2; //If combined, points are reduced by half.
while([w:TMP0004] < [w:TMP0011])
{
    [w:TMP0009] = [w:TMP0008] / 16;
    [w:TMP0010] = [w:TMP0008] % 16;

    //Output
    if([w:TMP0023] == 1) //Check If Error Occurred
    {
        [w:GD12000[w:TMP0008]] = 3; //Error Occurrence & Input ON
    }else{
        if([w:TMP0022] == 1) //Check If Alarms Occurred
        {
            [w:GD12000[w:TMP0008]] = 2; //Alarm Occurrence & Input ON
        }else{
            [w:GD12000[w:TMP0008]] = 1; //Normal
        }
    }

    //Input
    if([w:TMP0023] == 1) //Check If Error Occurred
    {
        [w:GD12256[w:TMP0008]] = 3; //Error Occurrence & Input ON
    }else{
        if([w:TMP0022] == 1) //Check If Alarms Occurred
        {
            [w:GD12256[w:TMP0008]] = 2; //Alarm Occurrence & Input ON
        }else{
            [w:GD12256[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:TMP0020], 5);

}

//Display Cursor
if([w:GD62000] == 30004)
{
    //Output
    [w:GD10301] = 0; //Left Right
    [w:GD10302] = 0; //Up Down
    [w:GD10303] = [w:GD10301] + [w:GD10302];
    [w:GD10300] = [w:GD10303]; //Cursor Position Display

```

```

//[w:TMP0030] = [w:GD10303] + [w:GD10601];

if([w:GD12000[w:GD10303]] < 4)
{
    [w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
}
}else{
    //Input
    [w:GD10301] = 256; //Left Right
    [w:GD10302] = 0; //Up Down
    [w:GD10303] = [w:GD10301] + [w:GD10302];
    [w:GD10300] = [w:GD10303] - 256; //Cursor Position Display
    //[w:TMP0030] = [w:GD10303] + [w:GD10601];

    if([w:GD12000[w:GD10303]] < 4)
    {
        [w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
    }
}

rst([b:GB30001]);

//Initial Action Check Flag
set([b:GB30035]);

```

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:GD10303]; //Evacuate Previous Cursor Position

if([w:GD10301] == 15)
{
    [w:GD10301] = 0;
}else{
    [w:GD10301] = [w:GD10301] + 1;
}

[w:TMP0000] = [w:GD10301];

[w:GD10303] = [w:TMP0000] + [w:GD10302]; //Offset

if([w:GD12000[w:TMP0020]] >= 4)
{
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;
}else{
    [w:GD12000[w:TMP0020]] = 0;
}

[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;

[w:GD10300] = [w:GD10303]; //Cursor Position Display

```

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

```
//Control Parts Display

//Calculate Cursor Position
[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position

if([w:GD10301] == 0)
{
    [w:GD10301] = 15;
}else{
    [w:GD10301] = [w:GD10301] -1;
}

[w:TMP0000] = [w:GD10301];

[w:GD10303] = [w:TMP0000] + [w:GD10302]; //Offset

if([w:GD12000[w:TMP0020]] >= 4)
{
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;
}else{
    [w:GD12000[w:TMP0020]] = 0;
}
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
```

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

```
//Control Parts Display

//Calculate Cursor Position
[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position

if([w:GD10302] == 240)
{
    [w:GD10302] = 0;
}else{
    [w:GD10302] = [w:GD10302] + 16;
}

[w:GD10303] = [w:GD10301] + [w:GD10302]; //Offset

if([w:GD12000[w:TMP0020]] >= 4)
{
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;
}else{
    [w:GD12000[w:TMP0020]] = 0;
}
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;

[w:GD10300] = [w:GD10303]; //Cursor Position Display
```

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

```
//Control Parts Display

//Calculate Cursor Position
[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position
```

```

if([w:GD10302] == 0)
{
    [w:GD10302] = 240;
}else{
    [w:GD10302] = [w:GD10302] - 16;
}

[w:GD10303] = [w:GD10301] + [w:GD10302];    //Offset

if([w:GD12000[w:TMP0020]] >= 4)
{
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;
}else{
    [w:GD12000[w:TMP0020]] = 0;
}
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;

```

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

```

//Clear Flags
rst([b:GB30006]);
rst([b:GB30007]);
rst([b:GB30003]);
rst([b:GB30018]);
rst([b:GB30035]);

[w:TMP0004] = 0;
fmov([w:TMP0004],[w:GD12000],512); //Initialize Object Values(B-30003, 4)
fmov([w:TMP0004],[w:GD10301],3);    //Clear Cursor Position Information(B-30003,4)
fmov([w:TMP0004],[w:GD10432],15); //Work Area Clear
[w:GD10300] = 0; //Cursor Position Information Clear(B-30003,4)

//Close Overlap Window
[w:GD62001] = 0;
[w:GD62004] = 0;
[w:GD62007] = 0;

```

脚本号.	30017	脚本名	Script30017
注释	B-30003, 30004 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:GD10303] < 256)
{
    [w:TMP0000] = [w:GD10303]; //Output ID
}else{
    [w:TMP0000] = [w:GD10303] - 256; //Input ID
}

if([w:GD12000[w:GD10303]] > 4)    //Judge whether a module exists in the cursor position.
{
    if([w:GD10303] < 256)
    {
        //Output
        [w:TMP0002] = [w:TMP0000];
    }
}

```

```

while([w:TMP0002] >= 0)
{
    if([w:GD12000[w:TMP0002]] != 0)
    {
        //Output
        [w:TMP0001] = 0;
        [w:TMP0005] = [w:GD10601];
        while([w:TMP0001] < [w:GD20162[w:GD10601]]) //Repeat the Number of Connection ID
Counts
        {
            if([w:GD20163[w:TMP0005]] == [w:TMP0002]) //Judge ID
            {
                set([b:GB30009]); //ID Match Flag
                break;
            }else{
                [w:TMP0001] = [w:TMP0001] + 1;
                [w:TMP0005] = [w:TMP0005] + 1;
            }
        }

        //Combined Output Side Case
        if([b:GB30009] == OFF)
        {
            [w:TMP0001] = 0;
            [w:TMP0005] = [w:GD10601];
            while([w:TMP0001] < [w:GD20162[w:GD10601]]) //Repeat the Number of Connection ID
Counts
            {
                if([w:GD20163[w:TMP0005]] == ([w:TMP0002] + 0x0200)) //Judge ID
                {

                    //Judge whether the matched ID is that of a combined module or not.
                    [w:TMP0006] = [w:GD10601] + [w:TMP0002];
                    [w:TMP0010] = [w:GD20676[w:TMP0006]]; //Parameter Top Address
                    [w:TMP0011] = ([w:TMP0010] - 0x3000) + [w:GD10602];

                    [w:TMP0012] = [w:GD24043[w:TMP0011]];
                    if((((w:TMP0012] & 0x00C0) >> 6) == 2)
                    {
                        set([b:GB30009]); //ID Match Flag
                        set([b:GB30020]); //Combined Output Flag
                        break;
                    }
                    [w:TMP0001] = [w:TMP0001] + 1;
                    [w:TMP0005] = [w:TMP0005] + 1;
                }else{
                    [w:TMP0001] = [w:TMP0001] + 1;
                    [w:TMP0005] = [w:TMP0005] + 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:TMP0001] = 0;
        [w:TMP0005] = [w:GD10601];
        while([w:TMP0001] < [w:GD20162[w:GD10601]]) //Repeat the Number of Connection ID Counts
        {
            if([w:GD20163[w:TMP0005]] == ([w:TMP0002] + 0x0200)) //Judge ID
            {
                set([b:GB30009]); //ID Match Flag
                break;
            }else{
                [w:TMP0001] = [w:TMP0001] + 1;
                [w:TMP0005] = [w:TMP0005] + 1;
            }
        }

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

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

//If matched, display screen.
if([b:GB30009] == ON)
{
    if([w:GD10303] < 256)
    {
        [w:TMP0007] = [w:GD10601] + [w:TMP0002];

        if([b:GB30020] == OFF)
        {
            //Output
            [w:GD10600] = [w:GD20420[w:TMP0007]]; //Offset
            [w:TMP0003] = [w:TMP0002];
            // [w:TMP0004] = [w:TMP0002] + [w:GD10601];
            [w:TMP0004] = [w:TMP0002];
        }else{
            //If combined, refer to the input side.
            [w:GD10600] = [w:GD20676[w:TMP0007]]; //Offset

```

```

[w:TMP0003] = [w:TMP0002] + 0x0200;
//[w:TMP0004] = ([w:TMP0002] + 0x0100) + [w:GD10601];
[w:TMP0004] = [w:TMP0002] + 0x0100;
}
}else{
//Input
[w:TMP0007] = [w:GD10601] + [w:TMP0002];
[w:GD10600] = [w:GD20676[w:TMP0007]]; //Offset
[w:TMP0003] = [w:TMP0002] + 0x0200;
//[w:TMP0004] = ([w:TMP0002] + 0x0100) + [w:GD10601];
[w:TMP0004] = [w:TMP0002] + 0x0100;
}

[w:U00-G10320] = 0; //Read Individual Parameter
[w:U00-G10321] = [w:TMP0003]; //Read Target ID

//Select base screen to display.
switch([w:GD16000[w:TMP0004]] & 0xF000)
{
    case 0x0000: [w:TMP0008] = 0;
                break;

    case 0x1000: [w:TMP0008] = 1;
                break;

    case 0x2000: [w:TMP0008] = 2;
                break;

    case 0x3000: [w:TMP0008] = 3;
                break;

    case 0xA000: [w:TMP0008] = 10;
                break;

}

if([w:TMP0008] == 0)
{
    switch([w:GD16000[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;
}

```

```

}

if([w:TMP0008] == 1)
{
    switch([w:GD16000[w:TMP0004]])
    {
        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 0x1401: [w:GD62000] = 30019;
            break;

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

        case 0x1451: [w:GD62000] = 30019;
            break;

        case 0x1500: [w:GD62000] = 30043;
            break;

        // case 0x1503: [w:GD62000] = 30046;
        //         break;

        case 0x1553: [w:GD62000] = 30050;
            break;
    }
}

if([w:TMP0008] == 2)
{
    switch([w:GD16000[w:TMP0004]])
    {

        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 0x2601: [w:GD62000] = 30017;
        break;

    case 0x2602: [w:GD62000] = 30017;
        break;

    case 0x2603: [w:GD62000] = 30017;
        break;

    case 0x2900: [w:GD62000] = 30020;
        break;

    case 0x2903: [w:GD62000] = 30021;
        break;

    case 0x2909: [w:GD62000] = 30023;
        break;

    case 0x290C: [w:GD62000] = 30053;
        break;

    case 0x290F: [w:GD62000] = 30054;
        break;

    case 0x2930: [w:GD62000] = 30024;
        break;

    case 0x2933: [w:GD62000] = 30025;
        break;

    case 0x2939: [w:GD62000] = 30027;
        break;

    case 0x293C: [w:GD62000] = 30057;
        break;

    case 0x293F: [w:GD62000] = 30058;
        break;

    case 0x2960: [w:GD62000] = 30028;
        break;

    case 0x2963: [w:GD62000] = 30029;
        break;

    case 0x2969: [w:GD62000] = 30031;
        break;

```

```

        case 0x293C: [w:GD62000] = 30061;
            break;

        case 0x293F: [w:GD62000] = 30062;
            break;
    }
}

if([w:TMP0008] == 3)
{
    switch([w:GD16000[w:TMP0004]])
    {

        case 0x3100: [w:GD62000] = 30032;
            break;

        case 0x3300: [w:GD62000] = 30033;
            break;

        case 0x3301: [w:GD62000] = 30034;
            break;

        case 0x3302: [w:GD62000] = 30035;
            break;

        case 0x3303: [w:GD62000] = 30035;
            break;

        case 0x3304: [w:GD62000] = 30036;
            break;

        case 0x3305: [w:GD62000] = 30036;
            break;
    }
}

if([w:TMP0008] == 10)
{
    switch([w:GD16000[w:TMP0004]])
    {

        case 0xA000: [w:GD62000] = 30018;
            break;

        case 0xA001: [w:GD62000] = 30018;
            break;

        case 0xA002: [w:GD62000] = 30037;
            break;

        case 0xA003: [w:GD62000] = 30037;
            break;

        case 0xA004: [w:GD62000] = 30038;
            break;
    }
}

```

```
case 0xA005: [w:GD62000] = 30038;
    break;

case 0xA006: [w:GD62000] = 30039;
    break;

case 0xA007: [w:GD62000] = 30039;
    break;

case 0xA010: [w:GD62000] = 30051;
    break;

case 0xA300: [w:GD62000] = 30018;
    break;

case 0xA301: [w:GD62000] = 30018;
    break;

case 0xA302: [w:GD62000] = 30037;
    break;

case 0xA303: [w:GD62000] = 30037;
    break;

case 0xA304: [w:GD62000] = 30038;
    break;

case 0xA305: [w:GD62000] = 30038;
    break;

case 0xA306: [w:GD62000] = 30039;
    break;

case 0xA307: [w:GD62000] = 30039;
    break;

case 0xA308: [w:GD62000] = 30040;
    break;

case 0xA309: [w:GD62000] = 30040;
    break;

case 0xA310: [w:GD62000] = 30052;
    break;

case 0xA320: [w:GD62000] = 30041;
    break;

case 0xA321: [w:GD62000] = 30042;
    break;

case 0xA322: [w:GD62000] = 30042;
    break;

case 0xA323: [w:GD62000] = 30042;
    break;
```

```

        case 0xA324: [w:GD62000] = 30042;
                        break;

        case 0xA325: [w:GD62000] = 30042;
                        break;

        case 0xA326: [w:GD62000] = 30042;
                        break;

        case 0xA600: [w:GD62000] = 30018;
                        break;

        case 0xA601: [w:GD62000] = 30018;
                        break;

        case 0xA602: [w:GD62000] = 30037;
                        break;

        case 0xA603: [w:GD62000] = 30037;
                        break;

        case 0xA604: [w:GD62000] = 30038;
                        break;

        case 0xA605: [w:GD62000] = 30038;
                        break;

        case 0xA606: [w:GD62000] = 30039;
                        break;

        case 0xA607: [w:GD62000] = 30039;
                        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:GB30001] == OFF)

```

```

{
    //Save Current Cursor Position
    [w:TMP0122] = [w:GD10303];

    //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:GD10301] = 16 - ([w:TMP0124] + 1);

    //Y Coordinate
    [w:TMP0125] = [w:TMP0121] - Input_Y;
    [w:TMP0126] = [w:TMP0125] / Object_Y;
    [w:GD10302] = [w:TMP0126] * 16;

    //Offset
    [w:GD10303] = [w:GD10301] + [w:GD10302];

    //If the offset position does not change, displayed parts do not change.
    if([w:TMP0122] != [w:GD10303])
    {
        //Cursor Display
        if([w:GD12000[w:GD10303]] < 4)
        {
            [w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
        }

        //Cursor Delete
        if([w:GD12000[w:TMP0122]] < 4)
        {
            [w:GD12000[w:TMP0122]] = 0;
        }else{
            [w:GD12000[w:TMP0122]] = [w:GD12000[w:TMP0122]] - 4;
        }
    }
}

[w:GD10300] = [w:GD10303]; //Cursor Position Display
}

```

脚本号.	30083	脚本名	Script30083
注释	报警复位		
数据类型	有符号 BIN16	触发类型	上升沿 GB30049

```

//Turn ON the alarm reset signal.

```

```

switch([w:GD64000])
{
    case Master_1: set([b:Y0000]);
                    break;

    case Master_2: set([b:Y0020]);
                    break;

    case Master_3: set([b:Y0040]);
                    break;
}

```

```

    case Master_4: set([b:Y0060]);
        break;

```

```

    default:      break;

```

```

}

```

```

set([b:GB30050]);

```

```

rst([b:GB30049]);

```

脚本号.	30084	脚本名	Script30084
注释	报警清除触发复位		
数据类型	有符号 BIN16	触发类型	ON 中周期 1 秒 GB30050

```

//Turn OFF the alarm reset signal.

```

```

switch([w:GD64000])
{

```

```

{

```

```

    case Master_1: rst([b:Y0000]);
        break;

```

```

    case Master_2: rst([b:Y0020]);
        break;

```

```

    case Master_3: rst([b:Y0040]);
        break;

```

```

    case Master_4: rst([b:Y0060]);
        break;

```

```

    default:      break;

```

```

}

```

```

rst([b:GB30050]);

```

*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

```

//Read Statuses or Parameters

```

```

if([b:GB30007] == OFF)
{

```

```

{

```

```

    if(([w:U00-G8192] != 0) || ([w:U00-G9984] != 0))
    {

```

```

    {

```

```

        switch([w:GD64000])
        {

```

```

        {

```

```

            case Master_1: set([b:GD11004.b0]);
                break;

```

```

            case Master_2: set([b:GD11012.b0]);
                break;

```

```

            case Master_3: set([b:GD11020.b0]);
                break;

```



```

        case Master_4: set([b:GD11028.b0]);
                        break;

        default:       break;
    }
} else {
    switch([w:GD64000])
    {
        case Master_1: set([b:GD11000.b0]);
                        break;

        case Master_2: set([b:GD11008.b0]);
                        break;

        case Master_3: set([b:GD11016.b0]);
                        break;

        case Master_4: set([b:GD11024.b0]);
                        break;

        default:       break;
    }
}

set([b:GB30007]);
}

```

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	通常

//After reading the status, start Script No.30015.

```

if([b:GB30016] == OFF)
{
    set([b:GB30008]);
    set([b:GB30016]);
}

```

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

```

//Display Objects
[w:GD10494] = 0;
[w:TMP0004] = 0;
[w:TMP0006] = 1;
[w:TMP0100] = [w:GD10401];
[w:TMP0101] = [w:GD10401] + [w:GD10601];
fmov([w:TMP0004], [w:TMP0010], 5);

while([w:GD10494] < 15)
{
    //Check Connection Status
    if(([w:GD20162[w:GD10601]] != 0) && ([w:TMP0100] < [w:GD20162[w:GD10601]]))
    {
        [w:TMP0010] = [w:GD20163[w:TMP0101]]; //Connection ID
        [w:TMP0100] = [w:TMP0100] + 1;
        [w:TMP0101] = [w:TMP0101] + 1;

        if(([w:TMP0010] & 0x0200) == 0)

```

```

{
    set([b:GB30600]); //Output Bit
    [w:TMP0011] = ([w:TMP0010] & 0x0200) >> 9;
} else {
    set([b:GB30601]); //Input Bit
    [w:TMP0011] = ([w:TMP0010] & 0x0200) >> 9;
}

//Offset
[w:TMP0000] = ([w:TMP0010] & 0x00FF) + [w:GD10601];

if(([w:TMP0010] & 0x0200) == 0)
{
    //Output
    [w:TMP0001] = [w:GD20420[w:TMP0000]];
} else {
    //Input
    [w:TMP0001] = [w:GD20676[w:TMP0000]];
}
[w:TMP0014] = ([w:TMP0001] - 12288) + [w:GD10602];

//Module Type
//[w:TMP0002] = [w:TMP0014] + [w:GD10602];
[w:GD10402[w:GD10494]] = ([w:GD24043[w:TMP0014]] & 0x00C0) >> 6;

//Module Model No.
//[w:TMP0005] = [w:TMP0014] + [w:GD10601];
[w:GD10477[w:GD10494]] = [w:GD24044[w:TMP0014]];

//Alarm Information
if([w:GD20291[w:GD10601]] != 0)
{
    [w:TMP0004] = 0;
    [w:TMP0005] = [w:GD10601];
    while([w:TMP0004] < [w:GD20291[w:GD10601]])
    {
        if([w:GD20292[w:TMP0005]] == [w:TMP0010])
        {
            [w:TMP0012] = 1; //Alarm Occurrence
            if((([w:U00-G10256] != 304) && (([w:U00-G10256] < 200) || ([w:U00-G10256] > 202))))
            {
                [w:GD10417[w:GD10494]] = [w:U00-G10256];
            } else {
                [w:TMP0003] = ([w:TMP0001] - 0x3000) + [w:GD10602];
                [w:GD10417[w:GD10494]] = [w:GD24039[w:TMP0003]];
            }
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
        [w:TMP0005] = [w:TMP0005] + 1;
    }
}

//Error Information
if([w:GD20032[w:GD10601]] != 0)
{

```

```

[w:TMP0004] = 0;
[w:TMP0005] = [w:GD10601];
while([w:TMP0004] < [w:GD20032[w:GD10601]])
{
    if([w:GD20033[w:TMP0005]] == [w:TMP0010])
    {
        [w:TMP0013] = 1; //Error Occurrence
        if((([w:U00-G10256] == 304) || (([w:U00-G10256] >= 200) && ([w:U00-G10256] <= 202))))
        {
            [w:GD10417[w:GD10494]] = [w:U00-G10256];
        } else {
            [w:TMP0003] = ([w:TMP0001] - 0x3000) + [w:GD10602];
            [w:GD10417[w:GD10494]] = [w:GD24039[w:TMP0003]];
        }
        break;
    }
    [w:TMP0004] = [w:TMP0004] + 1;
    [w:TMP0005] = [w:TMP0005] + 1;
}

//Display
if([w:TMP0013] == 1)
{
    [w:GD10447[w:GD10494]] = 3; //Error Occurrence
    [w:GD10132[w:GD10494]] = [w:TMP0011]; //I/O
    [w:GD10462[w:GD10494]] = [w:TMP0010] & 0x00FF; //ID
    [w:GD10432[w:GD10494]] = [w:TMP0010]; //User-defined name
    [w:GD10492] = [w:GD10492] | ([w:TMP0006] << [w:GD10494]); //Display Conditions
} else {
    if([w:TMP0012] == 1)
    {
        [w:GD10447[w:GD10494]] = 2; //Alarm Occurrence
        [w:GD10132[w:GD10494]] = [w:TMP0011]; //I/O
        [w:GD10462[w:GD10494]] = [w:TMP0010] & 0x00FF; //ID
        [w:GD10432[w:GD10494]] = [w:TMP0010]; //User-defined name
        [w:GD10492] = [w:GD10492] | ([w:TMP0006] << [w:GD10494]); //Display Conditions
    } else {
        [w:GD10447[w:GD10494]] = 1; //Normal
        [w:GD10132[w:GD10494]] = [w:TMP0011]; //I/O
        [w:GD10462[w:GD10494]] = [w:TMP0010] & 0x00FF; //ID
        [w:GD10432[w:GD10494]] = [w:TMP0010]; //User-defined name
        [w:GD10417[w:GD10494]] = 0; //Status
        [w:GD10492] = [w:GD10492] | ([w:TMP0006] << [w:GD10494]); //Display Conditions
    }
}

[w:GD10494] = [w:GD10494] + 1;
rst([b:GB30600]); //Output Bit
rst([b:GB30601]); //Input Bit

//Work Area Clear
[w:TMP0004] = 0;
fmov([w:TMP0004], [w:TMP0010], 5);

```

```

}else{
    //If data does not exist, do not display.
    [w:GD10447[w:GD10494]] = 0xFFFF;
    [w:GD10132[w:GD10494]] = 0xFFFF;
    [w:GD10462[w:GD10494]] = 0xFFFF;
    [w:GD10477[w:GD10494]] = 0xFFFF;
    [w:GD10402[w:GD10494]] = 0xFFFF;
    [w:GD10432[w:GD10494]] = 0x7FFE - [w:GD10601];
    [w:GD10417[w:GD10494]] = 24219;
    [w:GD10492] = [w:GD10492] ^ ([w:TMP0006] << [w:GD10494]); //Display Conditions
    [w:GD10494] = [w:GD10494] + 1;
    rst([b:GB30600]); //Output Bit
    rst([b:GB30601]); //Input Bit
}
}

```

```

//Trigger Reset
rst([b:GB30008]);

```

```

//Initial Action Check Flag
set([b:GB30035]);

```

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

```

//Clear Flags
rst([b:GB30006]);
rst([b:GB30007]);
rst([b:GB30003]);
rst([b:GB30018]);
rst([b:GB30035]);

```

```

[w:TMP0004] = 0;
fmov([w:TMP0004], [w:GD12000], 512); //Initialize Object Values(B-30003, 4)
fmov([w:TMP0004], [w:GD10301], 3); //Clear Cursor Position Information(B-30003, 4)
fmov([w:TMP0004], [w:GD10432], 15); //Work Area Clear
[w:GD10300] = 0; //Cursor Position Information Clear(B-30003, 4)

```

```

//Close Overlap Window
[w:GD62001] = 0;
[w:GD62004] = 0;
[w:GD62007] = 0;

```

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

```

//Scroll List Up
if([w:GD10401] > 0)
{
    [w:GD10401] = [w:GD10401] - 15;
    set([b:GB30008]);
}

```

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

```

//Scroll List Down
if(((w:GD10401] + 15) < [w:U00-G8960]) && ([w:U00-G8960] > 15))
{

```

[w:GD10401] = [w:GD10401] + 15; set([b:GB30008]); }			
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:GD10493] & 0x00FF) + [w:GD10601]; if((([w:GD10493] & 0x0200) >> 8) == 0) { //Output [w:GD10600] = [w:GD20420[w:TMP0000]]; //Offset }else{ //Input [w:GD10600] = [w:GD20676[w:TMP0000]]; //Offset } [w:U00-G10320] = 0; //Read Individual Parameter [w:U00-G10321] = [w:GD10493]; //Read Target ID //Select base screen to display. switch([w:GD10477[w:GD10400]] & 0xF000) { case 0x0000: [w:TMP0003] = 0; break; case 0x1000: [w:TMP0003] = 1; break; case 0x2000: [w:TMP0003] = 2; break; case 0x3000: [w:TMP0003] = 3; break; case 0x9000: [w:TMP0003] = 9; break; case 0xA000: [w:TMP0003] = 10; break; } if([w:TMP0003] == 0) { switch([w:GD10477[w:GD10400]]) { case 0x0100: [w:GD62000] = 30006; break; case 0x0101: [w:GD62000] = 30006; break; case 0x0102: [w:GD62000] = 30008; break;</pre>			

```
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;
}
}

if([w:TMP0003] == 1)
{
    switch([w:GD10477[w:GD10400]])
    {
        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 0x1401: [w:GD62000] = 30019;
            break;

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

        case 0x1451: [w:GD62000] = 30019;
            break;

        case 0x1500: [w:GD62000] = 30043;
            break;

        // case 0x1503: [w:GD62000] = 30046;
        //         break;

        case 0x1553: [w:GD62000] = 30050;
            break;
    }
}

if([w:TMP0003] == 2)
{
    switch([w:GD10477[w:GD10400]])
    {

        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 0x2601: [w:GD62000] = 30017;
    break;

case 0x2602: [w:GD62000] = 30017;
    break;

case 0x2603: [w:GD62000] = 30017;
    break;

case 0x2900: [w:GD62000] = 30020;
    break;

case 0x2903: [w:GD62000] = 30021;
    break;

case 0x2909: [w:GD62000] = 30023;
    break;

case 0x290C: [w:GD62000] = 30053;
    break;

case 0x290F: [w:GD62000] = 30054;
    break;

case 0x2930: [w:GD62000] = 30024;
    break;

case 0x2933: [w:GD62000] = 30025;
    break;

case 0x2939: [w:GD62000] = 30027;
    break;

case 0x293C: [w:GD62000] = 30057;
    break;

case 0x293F: [w:GD62000] = 30058;
    break;

case 0x2960: [w:GD62000] = 30028;
```



```

        break;

    case 0x2963: [w:GD62000] = 30029;
        break;

    case 0x2969: [w:GD62000] = 30031;
        break;

    case 0x293C: [w:GD62000] = 30061;
        break;

    case 0x293F: [w:GD62000] = 30062;
        break;
    }
}

if([w:TMP0003] == 3)
{
    switch([w:GD10477[w:GD10400]])
    {

        case 0x3100: [w:GD62000] = 30032;
            break;

        case 0x3300: [w:GD62000] = 30033;
            break;

        case 0x3301: [w:GD62000] = 30034;
            break;

        case 0x3302: [w:GD62000] = 30035;
            break;

        case 0x3303: [w:GD62000] = 30035;
            break;

        case 0x3304: [w:GD62000] = 30036;
            break;

        case 0x3305: [w:GD62000] = 30036;
            break;
    }
}

if([w:TMP0003] == 10)
{
    switch([w:GD10477[w:GD10400]])
    {

        case 0xA000: [w:GD62000] = 30018;
            break;

        case 0xA001: [w:GD62000] = 30018;
            break;

        case 0xA002: [w:GD62000] = 30037;

```

```
        break;

    case 0xA003: [w:GD62000] = 30037;
        break;

    case 0xA004: [w:GD62000] = 30038;
        break;

    case 0xA005: [w:GD62000] = 30038;
        break;

    case 0xA006: [w:GD62000] = 30039;
        break;

    case 0xA007: [w:GD62000] = 30039;
        break;

    case 0xA010: [w:GD62000] = 30051;
        break;

    case 0xA300: [w:GD62000] = 30018;
        break;

    case 0xA301: [w:GD62000] = 30018;
        break;

    case 0xA302: [w:GD62000] = 30037;
        break;

    case 0xA303: [w:GD62000] = 30037;
        break;

    case 0xA304: [w:GD62000] = 30038;
        break;

    case 0xA305: [w:GD62000] = 30038;
        break;

    case 0xA306: [w:GD62000] = 30039;
        break;

    case 0xA307: [w:GD62000] = 30039;
        break;

    case 0xA308: [w:GD62000] = 30040;
        break;

    case 0xA309: [w:GD62000] = 30040;
        break;

    case 0xA310: [w:GD62000] = 30052;
        break;

    case 0xA320: [w:GD62000] = 30041;
        break;
```

```

        case 0xA321: [w:GD62000] = 30042;
            break;

        case 0xA322: [w:GD62000] = 30042;
            break;

        case 0xA323: [w:GD62000] = 30042;
            break;

        case 0xA324: [w:GD62000] = 30042;
            break;

        case 0xA325: [w:GD62000] = 30042;
            break;

        case 0xA326: [w:GD62000] = 30042;
            break;

        case 0xA600: [w:GD62000] = 30018;
            break;

        case 0xA601: [w:GD62000] = 30018;
            break;

        case 0xA602: [w:GD62000] = 30037;
            break;

        case 0xA603: [w:GD62000] = 30037;
            break;

        case 0xA604: [w:GD62000] = 30038;
            break;

        case 0xA605: [w:GD62000] = 30038;
            break;

        case 0xA606: [w:GD62000] = 30039;
            break;

        case 0xA607: [w:GD62000] = 30039;
            break;
    }
}

```

```
rst([b:GB30018]);
```

脚本号.	30083	脚本名	Script30083
注释	报警复位		
数据类型	有符号 BIN16	触发类型	上升沿 GB30049

```
//Turn ON the alarm reset signal.
```

```

switch([w:GD64000])
{
    case Master_1: set([b:Y0000]);
        break;
}

```

<pre> case Master_2: set([b:Y0020]); break; case Master_3: set([b:Y0040]); break; case Master_4: set([b:Y0060]); break; default: break; } set([b:GB30050]); rst([b:GB30049]); </pre>			
脚本号.	30084	脚本名	Script30084
注释	报警清除触发复位		
数据类型	有符号 BIN16	触发类型	ON 中周期 1 秒 GB30050
<pre> //Turn OFF the alarm reset signal. switch([w:GD64000]) { case Master_1: rst([b:Y0000]); break; case Master_2: rst([b:Y0020]); break; case Master_3: rst([b:Y0040]); break; case Master_4: rst([b:Y0060]); break; default: break; } rst([b:GB30050]); </pre>			

基本画面 30006～30021、基本画面 30023～30025、基本画面 30027～30029、基本画面 30031～30042、基本画面 30051～30054、基本画面 30057～30058、基本画面 30061～30062

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:GD20032[w:GD10601]] != 0) { [w:TMP0000] = 0; [w:TMP0001] = [w:GD10601] + [w:TMP0000]; while([w:TMP0000] < [w:GD20032[w:GD10601]]) { if([w:GD20033[w:TMP0001]] == [w:U00-G10321]) { </pre>			

```

        set([b:GB30028]); //Error Occurrence Flag
        break;
    }
    [w:TMP0000] = [w:TMP0000] + 1;
    [w:TMP0001] = [w:TMP0001] + 1;

}

}

if([b:GB30028] == ON)
{
    //If errors occurred, read the most recently read settings.
    switch([w:GD64000])
    {
        case Master_1: set([b:GD11002.b0]);
            break;

        case Master_2: set([b:GD11010.b0]);
            break;

        case Master_3: set([b:GD11018.b0]);
            break;

        case Master_4: set([b:GD11026.b0]);
            break;

        default:      break;

    }
}
else{
    //If errors are not occurred, write the latest information.
    switch([w:GD64000])
    {
        case Master_1: set([b:Y0010]);
            break;

        case Master_2: set([b:Y0030]);
            break;

        case Master_3: set([b:Y0050]);
            break;

        case Master_4: set([b:Y0070]);
            break;

        default:      break;

    }
}

[w:GD10500] = 1; //Specify Start Line of Processing Method
set([b:GB30017]);
}

```

脚本号.	30014	脚本名	Script30014
注释	主模块 1 号 参数访问请求指令 OFF		
数据类型	有符号 BIN16	触发类型	ON 中 Y0010
<pre>//Master Module No.1 //Processing after Reading Parameters Individually if([w:U00-G10320] == 0) { set([b:GB30024]); //Read Flag }else{ set([b:GB30025]); //Write Flag } if([b:X0012] == ON) { [w:GD62007] = 30004; rst([b:GB30024]); rst([b:GB30025]); rst([b:Y0010]); }else{ if([b:X0011] == ON) { rst([b:GB30024]); rst([b:GB30025]); //If the alarm bit of the pressure sensor has been switched, //then return to the [AnyWireASLINK Master Module SEL] screen. if([b:GB30034] == ON) { [w:GD62000] = 30000; }else{ set([b:GB30026]); } rst([b:Y0010]); } } }</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	通常
<pre>if([b:GB30021] == OFF) { [w:TMP1000] = 0; //Display contents according to the module model number. switch([w:GD11525] & 0xF000) { case 0x0000: [w:TMP1000] = 0; break; case 0x1000: [w:TMP1000] = 1; break; case 0x2000: [w:TMP1000] = 2; break; case 0x3000: [w:TMP1000] = 3;</pre>			

```

        break;

    case 0x9000: [w:TMP1000] = 9;
        break;

    case 0xA000: [w:TMP1000] = 10;
        break;

    default:      [w:TMP1000] = 65535;
        break;
}

if([w:TMP1000] == 0)
{
    switch([w:GD11525])
    {
        case 0x0100: [w:GD10501] = 10000;
            break;
        case 0x0101: [w:GD10501] = 10001;
            break;
        case 0x0102: [w:GD10501] = 10002;
            break;
        case 0x0103: [w:GD10501] = 10003;
            break;
        case 0x0104: [w:GD10501] = 10004;
            break;
        case 0x0105: [w:GD10501] = 10005;
            break;
        case 0x0106: [w:GD10501] = 10006;
            break;
        case 0x0107: [w:GD10501] = 10007;
            break;
        case 0x0400: [w:GD10501] = 10008;
            break;
        case 0x0401: [w:GD10501] = 10009;
            break;
        case 0x0402: [w:GD10501] = 10010;
            break;
        case 0x0403: [w:GD10501] = 10011;
            break;
        case 0x0404: [w:GD10501] = 10012;
            break;
        case 0x0405: [w:GD10501] = 10013;
            break;
        case 0x0406: [w:GD10501] = 10014;
            break;
        case 0x0407: [w:GD10501] = 10015;
            break;
        case 0x0700: [w:GD10501] = 10016;
            break;
        case 0x0701: [w:GD10501] = 10017;
            break;
        case 0x0702: [w:GD10501] = 10018;
            break;
    }
}

```

```

        case 0x0703: [w:GD10501] = 10019;
            break;
        case 0x0704: [w:GD10501] = 10020;
            break;
        case 0x0705: [w:GD10501] = 10021;
            break;
        case 0x0706: [w:GD10501] = 10022;
            break;
        case 0x0707: [w:GD10501] = 10023;
            break;
        default:      break;
    }
}

if([w:TMP1000] == 1)
{
    switch([w:GD11525])
    {
        case 0x1000: [w:GD10501] = 11000;
            break;
        case 0x1050: [w:GD10501] = 11001;
            break;
        case 0x1200: [w:GD10501] = 11002;
            break;
        case 0x1250: [w:GD10501] = 11003;
            break;
        case 0x1400: [w:GD10501] = 11004;
            break;
        case 0x1401: [w:GD10501] = 11005;
            break;
        case 0x1450: [w:GD10501] = 11006;
            break;
        case 0x1451: [w:GD10501] = 11007;
            break;
        case 0x1500: [w:GD10501] = 11008;
            break;
        case 0x1501: [w:GD10501] = 11009;
            break;
        case 0x1502: [w:GD10501] = 11010;
            break;
        case 0x1503: [w:GD10501] = 11011;
            break;
        case 0x1504: [w:GD10501] = 11012;
            break;
        case 0x1550: [w:GD10501] = 11013;
            break;
        case 0x1551: [w:GD10501] = 11014;
            break;
        case 0x1552: [w:GD10501] = 11015;
            break;
        case 0x1553: [w:GD10501] = 11016;
            break;
        case 0x1554: [w:GD10501] = 11017;
            break;
        default:      break;
    }
}

```



```

}

if([w:TMP1000] == 2)
{
    switch([w:GD11525])
    {
        case 0x2000: [w:GD10501] = 12000;
            break;
        case 0x2300: [w:GD10501] = 12001;
            break;
        case 0x2301: [w:GD10501] = 12002;
            break;
        case 0x2302: [w:GD10501] = 12003;
            break;
        case 0x2303: [w:GD10501] = 12004;
            break;
        case 0x2600: [w:GD10501] = 12005;
            break;
        case 0x2601: [w:GD10501] = 12006;
            break;
        case 0x2602: [w:GD10501] = 12007;
            break;
        case 0x2603: [w:GD10501] = 12008;
            break;
        case 0x2900: [w:GD10501] = 12009;
            break;
        case 0x2901: [w:GD10501] = 12010;
            break;
        case 0x2902: [w:GD10501] = 12011;
            break;
        case 0x2903: [w:GD10501] = 12012;
            break;
        case 0x2904: [w:GD10501] = 12013;
            break;
        case 0x2905: [w:GD10501] = 12014;
            break;
        case 0x2906: [w:GD10501] = 12015;
            break;
        case 0x2907: [w:GD10501] = 12016;
            break;
        case 0x2908: [w:GD10501] = 12017;
            break;
        case 0x2909: [w:GD10501] = 12018;
            break;
        case 0x290A: [w:GD10501] = 12019;
            break;
        case 0x290B: [w:GD10501] = 12020;
            break;
        case 0x290C: [w:GD10501] = 12021;
            break;
        case 0x290D: [w:GD10501] = 12022;
            break;
        case 0x290E: [w:GD10501] = 12023;
            break;
        case 0x290F: [w:GD10501] = 12024;
            break;
    }
}

```

```
case 0x2910: [w:GD10501] = 12025;
    break;
case 0x2911: [w:GD10501] = 12026;
    break;
case 0x2930: [w:GD10501] = 12027;
    break;
case 0x2931: [w:GD10501] = 12028;
    break;
case 0x2932: [w:GD10501] = 12029;
    break;
case 0x2933: [w:GD10501] = 12030;
    break;
case 0x2934: [w:GD10501] = 12031;
    break;
case 0x2935: [w:GD10501] = 12032;
    break;
case 0x2936: [w:GD10501] = 12033;
    break;
case 0x2937: [w:GD10501] = 12034;
    break;
case 0x2938: [w:GD10501] = 12035;
    break;
case 0x2939: [w:GD10501] = 12036;
    break;
case 0x293A: [w:GD10501] = 12037;
    break;
case 0x293B: [w:GD10501] = 12038;
    break;
case 0x293C: [w:GD10501] = 12039;
    break;
case 0x293D: [w:GD10501] = 12040;
    break;
case 0x293E: [w:GD10501] = 12041;
    break;
case 0x293F: [w:GD10501] = 12042;
    break;
case 0x2940: [w:GD10501] = 12043;
    break;
case 0x2941: [w:GD10501] = 12044;
    break;
case 0x2960: [w:GD10501] = 12045;
    break;
case 0x2961: [w:GD10501] = 12046;
    break;
case 0x2962: [w:GD10501] = 12047;
    break;
case 0x2963: [w:GD10501] = 12048;
    break;
case 0x2964: [w:GD10501] = 12049;
    break;
case 0x2965: [w:GD10501] = 12050;
    break;
case 0x2966: [w:GD10501] = 12051;
    break;
case 0x2967: [w:GD10501] = 12052;
    break;
```

```

        case 0x2968: [w:GD10501] = 12053;
            break;
        case 0x2969: [w:GD10501] = 12054;
            break;
        case 0x296A: [w:GD10501] = 12055;
            break;
        case 0x296B: [w:GD10501] = 12056;
            break;
        case 0x296C: [w:GD10501] = 12057;
            break;
        case 0x296D: [w:GD10501] = 12058;
            break;
        case 0x296E: [w:GD10501] = 12059;
            break;
        case 0x296F: [w:GD10501] = 12060;
            break;
        case 0x2970: [w:GD10501] = 12061;
            break;
        case 0x2971: [w:GD10501] = 12062;
            break;
        default:      break;
    }
}

```

```

if([w:TMP1000] == 3)
{
    switch([w:GD11525])
    {
        case 0x3100: [w:GD10501] = 13000;
            break;
        case 0x3101: [w:GD10501] = 13001;
            break;
        case 0x3300: [w:GD10501] = 13002;
            break;
        case 0x3301: [w:GD10501] = 13003;
            break;
        case 0x3302: [w:GD10501] = 13004;
            break;
        case 0x3303: [w:GD10501] = 13005;
            break;
        case 0x3304: [w:GD10501] = 13006;
            break;
        case 0x3305: [w:GD10501] = 13007;
            break;
        default:      break;
    }
}

```

```

if([w:TMP1000] == 9)
{
    switch([w:GD11525])
    {
        case 0x9000: [w:GD10501] = 14000;
            break;
        case 0x9001: [w:GD10501] = 14001;

```

```

        break;
    case 0x9002: [w:GD10501] = 14002;
        break;
    case 0x9003: [w:GD10501] = 14003;
        break;
    case 0x9004: [w:GD10501] = 14004;
        break;
    case 0x9005: [w:GD10501] = 14005;
        break;
    case 0x9006: [w:GD10501] = 14006;
        break;
    case 0x9007: [w:GD10501] = 14007;
        break;
    case 0x9008: [w:GD10501] = 14008;
        break;
    case 0x9009: [w:GD10501] = 14009;
        break;
    case 0x900A: [w:GD10501] = 14010;
        break;
    case 0x900B: [w:GD10501] = 14011;
        break;
    case 0x900C: [w:GD10501] = 14012;
        break;
    case 0x900D: [w:GD10501] = 14013;
        break;
    case 0x900F: [w:GD10501] = 14014;
        break;
    case 0x9100: [w:GD10501] = 14015;
        break;
    case 0x9101: [w:GD10501] = 14016;
        break;
    default:      break;
}
}

if([w:TMP1000] == 10)
{
    switch([w:GD11525])
    {
        case 0xA000: [w:GD10501] = 15000;
            break;
        case 0xA001: [w:GD10501] = 15001;
            break;
        case 0xA002: [w:GD10501] = 15002;
            break;
        case 0xA003: [w:GD10501] = 15003;
            break;
        case 0xA004: [w:GD10501] = 15004;
            break;
        case 0xA005: [w:GD10501] = 15005;
            break;
        case 0xA006: [w:GD10501] = 15006;
            break;
        case 0xA007: [w:GD10501] = 15007;
            break;
        case 0xA008: [w:GD10501] = 15008;

```

```

        break;
    case 0xA009: [w:GD10501] = 15009;
        break;
    case 0xA010: [w:GD10501] = 15010;
        break;
    case 0xA011: [w:GD10501] = 15011;
        break;
    case 0xA300: [w:GD10501] = 15012;
        break;
    case 0xA301: [w:GD10501] = 15013;
        break;
    case 0xA302: [w:GD10501] = 15014;
        break;
    case 0xA303: [w:GD10501] = 15015;
        break;
    case 0xA304: [w:GD10501] = 15016;
        break;
    case 0xA305: [w:GD10501] = 15017;
        break;
    case 0xA306: [w:GD10501] = 15018;
        break;
    case 0xA307: [w:GD10501] = 15019;
        break;
    case 0xA308: [w:GD10501] = 15020;
        break;
    case 0xA309: [w:GD10501] = 15021;
        break;
    case 0xA30A: [w:GD10501] = 15022;
        break;
    case 0xA30B: [w:GD10501] = 15023;
        break;
    case 0xA310: [w:GD10501] = 15024;
        break;
    case 0xA320: [w:GD10501] = 15025;
        break;
    case 0xA321: [w:GD10501] = 15026;
        break;
    case 0xA322: [w:GD10501] = 15027;
        break;
    case 0xA323: [w:GD10501] = 15028;
        break;
    case 0xA324: [w:GD10501] = 15029;
        break;
    case 0xA325: [w:GD10501] = 15030;
        break;
    case 0xA326: [w:GD10501] = 15031;
        break;
    case 0xA600: [w:GD10501] = 15032;
        break;
    case 0xA601: [w:GD10501] = 15033;
        break;
    case 0xA602: [w:GD10501] = 15034;
        break;
    case 0xA603: [w:GD10501] = 15035;
        break;
    case 0xA604: [w:GD10501] = 15036;

```

<pre> break; case 0xA605: [w:GD10501] = 15037; break; case 0xA606: [w:GD10501] = 15038; break; case 0xA607: [w:GD10501] = 15039; break; case 0xA610: [w:GD10501] = 15040; break; default: break; } } 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]); rst([b:GB30034]); rst([b:GB30035]); [w:TMP0004] = 0; fmov([w:TMP0004], [w:GD12000], 512); //Initialize Object Values(B-30002) //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:TMP0020], 7); //Alarm Information if([w:GD20291[w:GD10601]] != 0) { [w:TMP0004] = 0; [w:TMP0001] = [w:GD10601] + [w:TMP0004]; while([w:TMP0004] < [w:GD20291[w:GD10601]]) </pre>			

```

{
    if([w:GD20292[w:TMP0001]] == [w:GD11500])
    {
        [w:TMP0022] = 1; //Alarm Occurrence
        if(([w:U00-G10256] != 305) && (([w:U00-G10256] != 304) && (([w:U00-G10256] < 200) ||
([w:U00-G10256] > 202))))
        {
            [w:GD10502] = [w:U00-G10256];
        }else{
            [w:GD10502] = [w:GD11520];
        }
        break;
    }
    [w:TMP0004] = [w:TMP0004] + 1;
    [w:TMP0001] = [w:TMP0001] + 1;
}
}

//Error Information
if([w:GD20032[w:GD10601]] != 0)
{
    [w:TMP0004] = 0;
    [w:TMP0001] = [w:GD10601] + [w:TMP0004];
    while([w:TMP0004] < [w:GD20032[w:GD10601]])
    {
        if([w:GD20033[w:TMP0001]] == [w:GD11500])
        {
            [w:TMP0023] = 1; //Error Occurrence
            if(([w:U00-G10256] != 305) && (([w:U00-G10256] == 304) || (([w:U00-G10256] >= 200) &&
([w:U00-G10256] <= 202))))
            {
                [w:GD10502] = [w:U00-G10256];
            }else{
                [w:GD10502] = [w:GD11520];
            }
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
        [w:TMP0001] = [w:TMP0001] + 1;
    }
}

//Device Parameter
//I/O Points Pattern
[w:TMP0025] = ([w:GD11524] & 0x00C0) >> 6; //0: Input, 1: Output, 2: Combined
[w:TMP0026] = ([w:GD11524] & 0x003F) + 1; //Points

//ON/OFF Information
[w:TMP0008] = 0;
[w:TMP0000] = [w:GD11500] & 0x00FF;
if([w:TMP0025] == 0) //Input Case
{
    [w:TMP0004] = 0;

    while([w:TMP0004] < [w:TMP0026])

```

```

{
    [w:TMP0009] = [w:TMP0000] / 16;
    [w:TMP0010] = [w:TMP0000] % 16;

    [w:GD10519[w:TMP0008]] = [w:TMP0009];
    [w:TMP0012] = 0x0001;
    [w:GD10535[w:TMP0008]] = [w:TMP0012] << [w:TMP010];

    if([w:TMP0023] == 1) //Check If Error Occurred
    {
        [w:GD10503[w:TMP0008]] = 3; //Error Occurrence
    }else{
        if([w:TMP0022] == 1) //Check If Alarms Occurred
        {
            [w:GD10503[w:TMP0008]] = 2; //Alarm Occurrence
        }else{
            [w:GD10503[w:TMP0008]] = 1; //Normal
            [w:GD10502] = 0;
        }
    }

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

}else{
    if([w:TMP0025] == 1) //Output Case
    {
        [w:TMP0004] = 0;

        while([w:TMP0004] < [w:TMP0026])
        {
            [w:TMP0009] = [w:TMP0000] / 16;
            [w:TMP0010] = [w:TMP0000] % 16;

            [w:GD10519[w:TMP0008]] = [w:TMP0009] + 4096;
            [w:TMP0012] = 0x0001;
            [w:GD10535[w:TMP0008]] = [w:TMP0012] << [w:TMP010];

            if([w:TMP0023] == 1) //Check If Error Occurred
            {
                [w:GD10503[w:TMP0008]] = 3; //Error Occurrence
            }else{
                if([w:TMP0022] == 1) //Check If Alarms Occurred
                {
                    [w:GD10503[w:TMP0008]] = 2; //Alarm Occurrence
                }else{
                    [w:GD10503[w:TMP0008]] = 1; //Normal
                    [w:GD10502] = 0;
                }
            }
        }

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

```



```

    }

} else {
    if([w:TMP0025] == 2)    //Combined Case
    {
        [w:TMP0004] = 0;
        [w:TMP0011] = [w:TMP0026] / 2;    //If combined, points are reduced by half.
        while([w:TMP0004] < [w:TMP0011])
        {
            [w:TMP0009] = [w:TMP0000] / 16;
            [w:TMP0010] = [w:TMP0000] % 16;

            //出力
            [w:GD10519[w:TMP0008]] = [w:TMP0009] + 4096;
            [w:TMP0012] = 0x0001;
            [w:GD10535[w:TMP0008]] = [w:TMP0012] << [w:TMP010];

            if([w:TMP0023] == 1)    //Check If Error Occurred
            {
                [w:GD10503[w:TMP0008]] = 3; //Error Occurrence
            } else {
                if([w:TMP0022] == 1)    //Check If Error Occurred
                {
                    [w:GD10503[w:TMP0008]] = 2; //Alarm Occurrence
                } else {
                    [w:GD10503[w:TMP0008]] = 1; //Normal
                    [w:GD10502] = 0;
                }
            }
        }

        //Input
        [w:GD10520[w:TMP0008]] = [w:TMP0009];
        [w:TMP0012] = 0x0001;
        [w:GD10536[w:TMP0008]] = [w:TMP0012] << [w:TMP010];

        if([w:TMP0023] == 1)    //Check If Error Occurred
        {
            [w:GD10504[w:TMP0008]] = 3; //Error Occurrence
        } else {
            if([w:TMP0022] == 1)    //Check If Alarms Occurred
            {
                [w:GD10504[w:TMP0008]] = 2; //Alarm Occurrence
            } else {
                [w:GD10504[w:TMP0008]] = 1; //Normal
                [w:GD10502] = 0;
            }
        }
    }

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

```

rst([b:GB30019]);			
//Initial Action Check Flag set([b:GB30035]);			
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	通常
<pre>//Re-read Parameters if([b:GB30016] == OFF) { switch([w:GD64000]) { case Master_1: set([b:GD11002.b0]); break; case Master_2: set([b:GD11010.b0]); break; case Master_3: set([b:GD11018.b0]); break; case Master_4: set([b:GD11026.b0]); break; default: break; } 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([b:Y0010] == OFF && [b:Y0030] == OFF && [b:Y0050] == OFF && [b:Y0070] == OFF) { if([w:U00-G10320] == 0) { switch([w:GD64000]) { case Master_1: set([b:GD11002.b0]); break; case Master_2: set([b:GD11010.b0]); break; case Master_3: set([b:GD11018.b0]); break; case Master_4: set([b:GD11026.b0]); break; default: break; } } } else{ [w:U00-G10320] = 0; switch([w:GD64000])</pre>			

```

{
    case Master_1: set([b:Y0010]);
        break;

    case Master_2: set([b:Y0030]);
        break;

    case Master_3: set([b:Y0050]);
        break;

    case Master_4: set([b:Y0070]);
        break;

    default:      break;

}
}

```

```

rst([b:GB30026]);
}

```

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

```

//Read the detail information of slave module.

```

```

//Error Information
if([w:GD20032[w:GD10601]] != 0)
{
    [w:TMP0000] = 0;
    [w:TMP0001] = [w:GD10601] + [w:TMP0000];

    while([w:TMP0000] < [w:GD20032[w:GD10601]])
    {
        if([w:GD20033[w:TMP0001]] == [w:U00-G10321])
        {
            set([b:GB30028]); //Error Occurrence Flag
            break;
        }
        [w:TMP0000] = [w:TMP0000] + 1;
        [w:TMP0001] = [w:TMP0001] + 1;
    }
}

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:U00-G10320] = 0;
    [w:U00-G10321] = [w:GD11500];

    switch([w:GD64000])
    {

```

```

        case Master_1: set([b:Y0010]);
                        break;

        case Master_2: set([b:Y0030]);
                        break;

        case Master_3: set([b:Y0050]);
                        break;

        case Master_4: set([b:Y0070]);
                        break;

        default:       break;
    }
}

```

```
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:GD20032[w:GD10601]] != 0)
```

```

{
    [w:TMP0000] = 0;
    [w:TMP0001] = [w:GD10601] + [w:TMP0000];
    while([w:TMP0000] < [w:GD20032[w:GD10601]])
    {
        if([w:GD20033[w:TMP0001]] == [w:U00-G10321])
        {
            set([b:GB30028]); //Error Occurrence Flag
            break;
        }
        [w:TMP0000] = [w:TMP0000] + 1;
        [w:TMP0001] = [w:TMP0001] + 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:U00-G10320] = 1;
    [w:U00-G10321] = [w:GD11500];

    if((([w:GD11525] >= 0x2900) && ([w:GD11525] <= 0x2971)))
    {
        //Judge whether the alarm bit of the pressure sensor has been switched.
        if([w:GD10572] == [w:GD11501])
        {
            rst([b:GB30034]);
        }else{

```

```

        set([b:GB30034]);
    }
}

switch([w:GD64000])
{
    case Master_1: set([b:GD11006.b0]);
                    break;

    case Master_2: set([b:GD11014.b0]);
                    break;

    case Master_3: set([b:GD11022.b0]);
                    break;

    case Master_4: set([b:GD11030.b0]);
                    break;

    default:       break;
}
}

```

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

脚本号.	30075	脚本名	Script30075
注释	主模块 2 号 参数访问请求指令 OFF		
数据类型	有符号 BIN16	触发类型	ON 中 Y0030

```

//Master Module No.2
//Processing after Reading Parameters Individually
if([w:U02-G10320] == 0)
{
    set([b:GB30024]); //Read Flag
}else{
    set([b:GB30025]); //Write Flag
}

if([b:X0032] == 0N)
{
    [w:GD62007] = 30004;
    rst([b:GB30024]);
    rst([b:GB30025]);
    rst([b:Y0030]);
}else{
    if([b:X0031] == 0N)
    {
        rst([b:GB30024]);
        rst([b:GB30025]);
        set([b:GB30026]);
        rst([b:Y0030]);
    }
}
}

```

脚本号.	30078	脚本名	Script30078
注释	主模块 3 号 参数访问请求指令 OFF		
数据类型	有符号 BIN16	触发类型	ON 中 Y0050

```

//Master Module No.3
//Processing after Reading Parameters Individually
if([w:U04-G10320] == 0)

```

```

{
    set([b:GB30024]); //Read Flag
}else{
    set([b:GB30025]); //Write Flag
}

if([b:X0052] == 0N)
{
    [w:GD62007] = 30004;
    rst([b:GB30024]);
    rst([b:GB30025]);
    rst([b:Y0050]);
}else{
    if([b:X0051] == 0N)
    {
        rst([b:GB30024]);
        rst([b:GB30025]);
        set([b:GB30026]);
        rst([b:Y0050]);
    }
}
}

```

脚本号.	30081	脚本名	Script30081
注释	主模块 4 号 参数访问请求指令 OFF		
数据类型	有符号 BIN16	触发类型	ON 中 Y0070

```

//Master Module No.4
//Processing after Reading Parameters Individually.
if([w:U06-G10320] == 0)
{
    set([b:GB30024]); //Read Flag
}else{
    set([b:GB30025]); //Write Flag
}

if([b:X0072] == 0N)
{
    [w:GD62007] = 30004;
    rst([b:GB30024]);
    rst([b:GB30025]);
    rst([b:Y0070]);
}else{
    if([b:X0071] == 0N)
    {
        rst([b:GB30024]);
        rst([b:GB30025]);
        set([b:GB30026]);
        rst([b:Y0070]);
    }
}
}

```

基本画面 30020～30029、基本画面 30031、基本画面 30053～30054、基本画面 30057～30058、
基本画面 30061～30062

脚本号.	30086	脚本名	Script30086
注释	压力传感器 初始值保存		
数据类型	有符号 BIN16	触发类型	ON 中 GB30032
<pre>//Store the initial value of the alarm bit switching of the pressure sensor. [w:GD10572] = [w:GD11501]; rst([b:GB30032]);</pre>			

基本画面 30100

脚本号.	30062	脚本名	Script30062
注释	B-30100 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	Script30063
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 //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) {</pre>			

<pre>//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; [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.	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 GD54012.b5
<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.	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) { [w:GD54006] = 1080; }else{ [w:GD54006] = 1068; } }else{ //CC-Link if([w:GD54088] == 100) { [w:GD54006] = 1081; }else{ [w:GD54006] = 1068; } } [w:GD62007] = 30102; }else{ [w:GD54006] = 32007; [w:GD62007] = 30101; } }else{ [w:GD54006] = 32017; [w:GD62007] = 30101; } rst([b:GB54013]);</pre>			
脚本号.	30052	脚本名	Script30052
注释	B-30101 iQSS 备份前处理 2		
数据类型	有符号 BIN16	触发类型	ON 中 GB54002
<pre>//Transfer Specified Contents to PLC Devices //Setting Name bmov([w:GD54102[w:TMP0000]], [w:GD54017], 12);</pre>			

```
//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]];

//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) && ([w:SD1446] == 0x0001))
{
    set([b:SM1436]);
    [w:GD62000] = 30103;
}else{
    [w:GD54006] = 32006;
    [w:GD62007] = 30101;
}

rst([b:GB54002]);
```

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

```
//Clear Flags

rst([b:GB54008]);
rst([b:GB54009]);

//Close Overlap Window
[w:GD62001] = 0;
[w:GD62004] = 0;
[w:GD62007] = 0;
```

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

```
//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) && ([w:GD54032] >= 0)) || (([w:GD54031] >=
3) && ([w:GD54032] >= 0) && ([w:GD54033] >= 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

        }
    }
    }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] = 1026; //ID or Station

```

```

        [w:GD54036] = 1032; //All IDs or All Stations

```

```

        [w:GD54037] = 1033; //ID Specification or Station Specification

```

<pre> [w:GD54038] = 1085; //Not Specified or Station Sub [w:GD54039] = 0; //Not Specified or Station Sub Specified break; //CC-Link case 2: [w:GD54035] = 1025; //ID or Station [w:GD54036] = 1042; ///All IDs or All Stations [w:GD54037] = 1043; //ID Specification or Station Specification [w:GD54038] = 1027; //Not Specified or Station Sub [w:GD54039] = 1044; //Not Specified or Station Sub Specified break; //その他 default: [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] & 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) </pre>			

```

{
    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 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 専用画面
    [w:TMP0001] = 99;
}else{
    //汎用画面
    [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 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 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;

    [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
}

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]);
}

```

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)
{
    rst([b:GB54017]); //Recipe No.30001 Read Trigger
}else{
    rst([b:GB54019]); //Recipe No.30001 Read Trigger
}

```

rst([b:GD54012. b5]);

脚本号.	30037	脚本名	Script30037
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

```

//Clear Flags
rst([b:GB30006]);
rst([b:GB30007]);
rst([b:GB30003]);

rst([b:GB30016]);
rst([b:GB30017]);
rst([b:GB30021]);

```

rst([b:GB30028]);			
rst([b:GB30034]);			
rst([b:GB30035]);			
[w:TMP0004] = 0;			
fmov([w:TMP0004], [w:GD12000], 512); //Initialize Object Values(B-30002)			
//Close Overlap Window			
[w:GD62001] = 0;			
[w:GD62004] = 0;			
[w:GD62007] = 0;			
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
//Clear Flags			
rst([b:GB54008]);			
rst([b:GB54009]);			
//Close Overlap Window			
[w:GD62001] = 0;			
[w:GD62004] = 0;			
[w:GD62007] = 0;			
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
//Abort iQSS Backup or iQSS Restoration			
set([b:SM1442]); //iQSS Backup/Restoration Cancel Request			
rst([b:GB54002]);			

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
//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]);			
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
//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) && ([w:GD56223[w:TMP0000]] == [w:GD56224[w:TMP0000]])))
    {
        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
<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.	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) { //It's impossible to restore from the iQSS backup which does not end normally. if(([w:GD56223[w:TMP0000]] != 0) && ([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]] & 0x00FF) << 8) + ([w:GD56201[w:TMP0000]] & 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) && ([w:SD1446] == 0x0001)) { set([b:SM1439]); [w:GD62000] = 30106; } } }else{</pre>			

<pre> [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] = 1026; //ID or Station [w:GD54071] = 1032; //All IDs or All Stations [w:GD54072] = 1033; //ID Specification or Station Specification [w:GD54073] = 1085; //Not Specified or Station Sub [w:GD54074] = 0; //Not Specified or Station Sub Specified break; //CC-Link case 2: [w:GD54070] = 1025; //ID or Station [w:GD54071] = 1042; //All IDs or All Stations [w:GD54072] = 1043; //ID Specification or Station Specification </pre>			

```

[w:GD54073] = 1027; //Not Specified or Station Sub
[w:GD54074] = 1044; //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
<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.	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
<pre> //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 </pre>			
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
<pre> //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]; //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{ </pre>			

<pre> [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 30006～30021、基本画面 30023～30025、基本画面 30027～30029、基本画面 30031～30042、基本画面 30051～30054、基本画面 30057～30058、基本画面 30061～30062

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:GD10551] = \$\$ & [w:GD10535]; </pre>			

基本画面 30013

对象	液位 *1		
脚本用户 ID	1		
数据类型	有符号 BIN16	触发类型	通常
<pre> //Display ON/OFF Threshold Value screen_draw(0); redraw_object(); [w:TMP0200] = (object_width / 100) * [w:GD11501]; [w:TMP0201] = (object_width / 100) * [w:GD11503]; d_line([w:TMP0200],0,[w:TMP0200],object_height,0,1,224); //OFF / ON Threshold Value d_line([w:TMP0201],0,[w:TMP0201],object_height,0,1,3); //Alarm judgment (Hi) </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 30010～30012、基本画面 30014～30015、基本画面 30017～30018、基本画面 30032

对象	液位 *1		
脚本用户 ID	1		
数据类型	有符号 BIN16	触发类型	通常
//Display Threshold Value, Alarm Judgment(Hi), (Lo) screen_draw(0); redraw_object(); [w:TMP0200] = (object_width / 100) * [w:GD11501]; [w:TMP0201] = (object_width / 100) * [w:GD11503]; [w:TMP0202] = (object_width / 100) * [w:GD11504]; 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)			

*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 30032

对象	数值入力 *1		
脚本用户 ID	1		
数据类型	无符号 BIN16	触发类型	入力確定時
//When input is determined, redraw the screen. redraw_screen();			

*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 30035

对象	液位 *1		
脚本用户 ID	1		
数据类型	有符号 BIN16	触发类型	通常
//Display Alarm threshold Value screen_draw(0); redraw_object(); [w:TMP0202] = (object_width / 100) * [w:GD11504]; d_line([w:TMP0202],0,[w:TMP0202],object_height,0,1,3); //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 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
//Script to Display Comments if([w:GD54122[w:GD54001]] != 0) { [w:GD54077] = (([w:GD54122[w:GD54001]] * 10) + \$\$) + 1021; }else{ [w:GD54077] = 0; } 			

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 30003

对象	数值表示		
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 & Month from Clock Data [w:TMP950] = [w:GS650] & 0xF000; //Obtain Tenths Digit of "Last 2-Digits of Year" from Clock Data for Setting [w:TMP960] = [w:TMP950] >> 12; //Decimal Alignment [w:TMP968] = [w:TMP960] * 10; //BCD->BIN [w:TMP951] = [w:GS650] & 0x0F00; //Obtain Ones Digit of "Last 2-Digits of Year" from Clock Data for Setting [w:TMP961] = [w:TMP951] >> 8; //BCD->BIN [w:TMP973] = 2000 + [w:TMP968] + [w:TMP961]; //Set Year to TMP973 as BIN [w:GD63990] = [w:TMP973]; //Set Year [w:TMP952] = [w:GS650] & 0x00F0; //Obtain Tenths Digit of Month from Clock Data for Setting [w:TMP962] = [w:TMP952] >> 4; //Decimal Alignment [w:TMP969] = [w:TMP962] * 10; //BCD->BIN [w:TMP953] = [w:GS650] & 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] & 0xF000; //Obtain Tenths Digit of "Last 2-Digits of Day" from Clock Data for Setting [w:TMP963] = [w:TMP954] >> 12; //Decimal Alignment [w:TMP970] = [w:TMP963] * 10; //BCD->BIN [w:TMP955] = [w:GS651] & 0x0F00; //Obtain Ones Digit of "Last 2-Digits of Day" from Clock Data for Setting [w:TMP964] = [w:TMP955] >> 8; //BCD->BIN [w:TMP975] = [w:TMP970] + [w:TMP964]; //Set Day to TMP975 as BIN [w:GD63992] = [w:TMP975]; //Set Day [w:TMP956] = [w:GS651] & 0x00F0; //Obtain Tenths Digit of Hour from Clock Data for Setting [w:TMP965] = [w:TMP956] >> 4; //Decimal Alignment [w:TMP971] = [w:TMP965] * 10; //BCD->BIN [w:TMP957] = [w:GS651] & 0x000F; //Obtain Ones Digit of Hour from Clock Data for Setting [w:TMP976] = [w:TMP971] + [w:TMP957]; //Set Year to TMP976 as BIN [w:GD63993] = [w:TMP976]; //Set Hour [w:TMP958] = [w:GS652] & 0xF000; //Obtain Tenths Digit of "Last 2-Digits of Minute" from Clock Data for Setting [w:TMP966] = [w:TMP958] >> 12; //Decimal Alignment [w:TMP972] = [w:TMP966] * 10; //BCD->BIN [w:TMP959] = [w:GS652] & 0x0F00; //Obtain Ones Digit of "Last 2-Digits of Minute" from Clock Data for Setting [w:TMP967] = [w:TMP959] >> 8; //BCD->BIN [w:TMP977] = [w:TMP972] + [w:TMP967]; //Set Minute to TMP977 as BIN [w:GD63994] = [w:TMP977]; //Set Minute [w:TMP993] = [w:GS652] & 0x00F0; //Obtain Tenths Digit of Second from Clock Data for Setting [w:TMP995] = [w:TMP993] >> 4; //Decimal Alignment [w:TMP996] = [w:TMP995] * 10; //BCD->BIN [w:TMP994] = [w:GS652] & 0x000F; //Obtain Ones Digit of Second from Clock Data for Setting [w:TMP978] = [w:TMP996] + [w:TMP994]; //Set Second to TMP978 as BIN [w:GD63995] = [w:TMP978]; //Set Second</pre>			

对象	数值表示		
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	数值表示		
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	数值表示		
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	数值表示		
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	数值表示		
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	44	输入值区域的左上角 X 坐标
Input_Y	165	输入值区域的左上角 Y 坐标
Frame_X	620	I/O 显示区域的右下角 X 坐标
Frame_Y	437	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
Master_1	0	主模块 1 的 I/O 号
Master_2	2	主模块 2 的 I/O 号
Master_3	4	主模块 3 的 I/O 号
Master_4	6	主模块 4 的 I/O 号

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

6.1 关于用户定义名称的登录

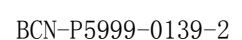
- 输出模块：请把注释号设置为模块的 ID+1 的号。(注释号无法设置为 0，因此将 ID 加 1 后的值作为注释号。)

- 输出模块 ID14~17 的
注释号

输入模块 ID14~17 的
注释号

- [illegible]

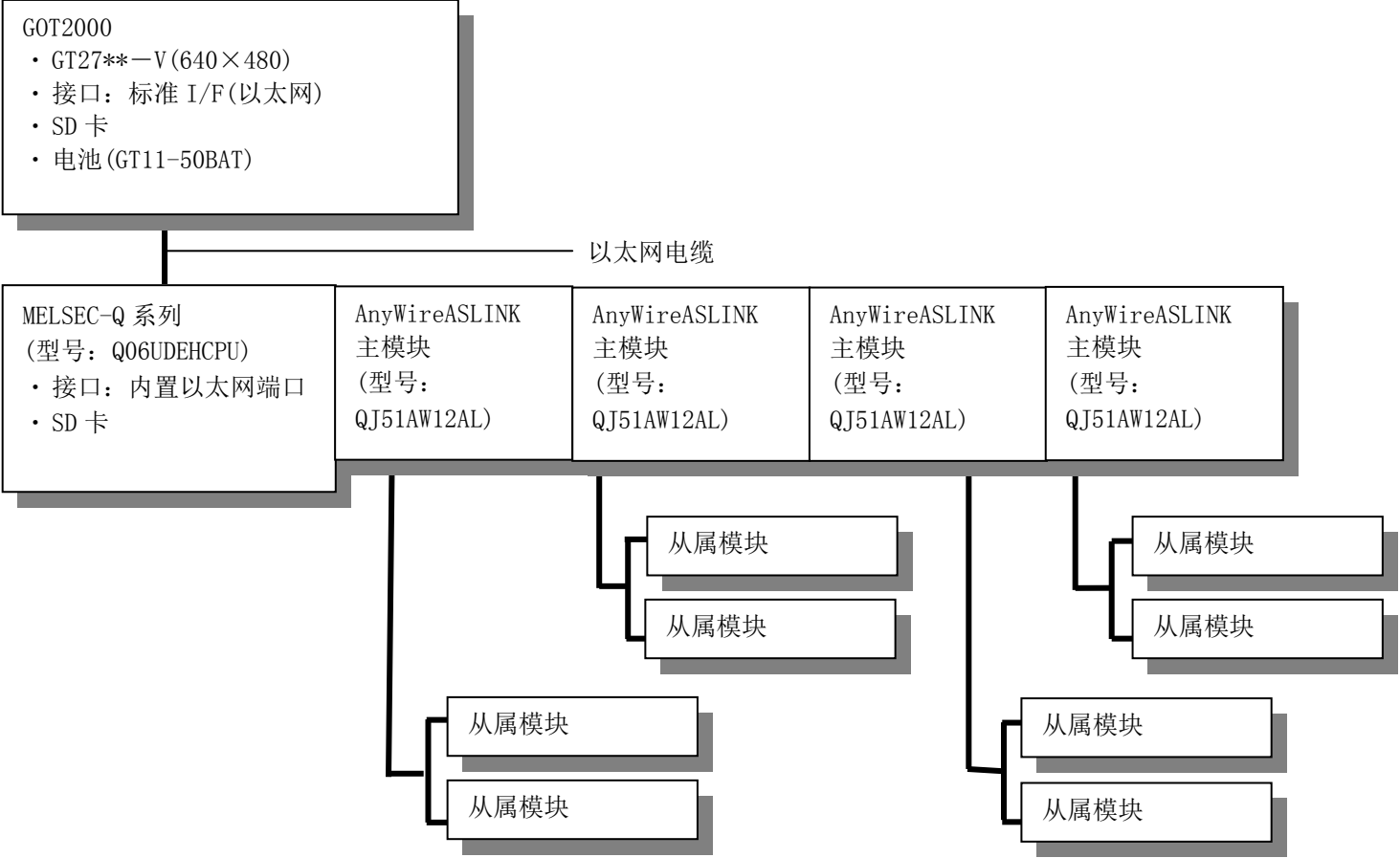
在 GOT 启动状态下对从属模块进行添加、删除、更改 ID 后，请返回基本画面 B-30001 并触摸画面右上角的「地址自动识别」开关 1 秒以上，或者重新启动 GOT。



6.3 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。

6.3.1 系统构成



6.3.2 画面创建软件的以太网设置

	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

6.4 关于起始输入输出号的更改

更改模块的起始输入输出号时，请按照以下步骤进行更改。
(例：起始输入输出号从 0H 更改到 80H 时)

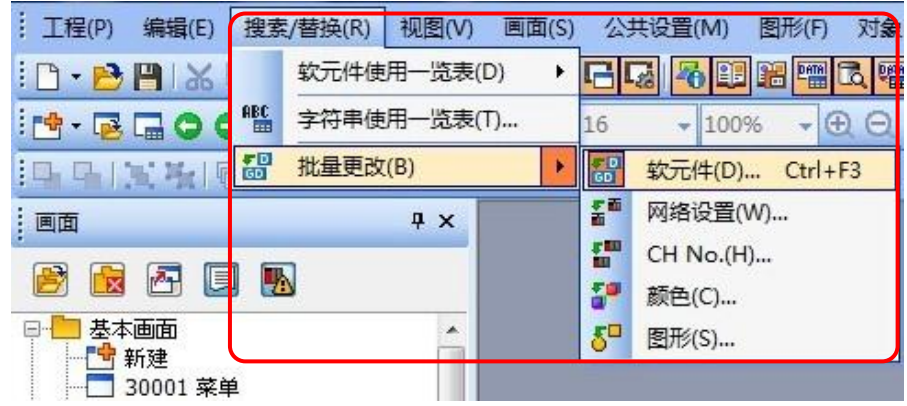
6.4.1 软元件批量更改更改前头输入输出编号时的想法

(1) 缓冲存储器的前头输入输出编号
从[更改前]Un-G0 更改为[更改后]Um-G0 时，将 m 设置为更改处的前头输入输出编号 16 等分的值。前头输入输出编号从 0H 更改为 80H 时，将 U00-G0 更改 U08-G0。
对于[点数]，将指定为设置显示的最小地址到最大地址的点数。从 U00-G0~U00-G99 更改时，将点数设置为 100。

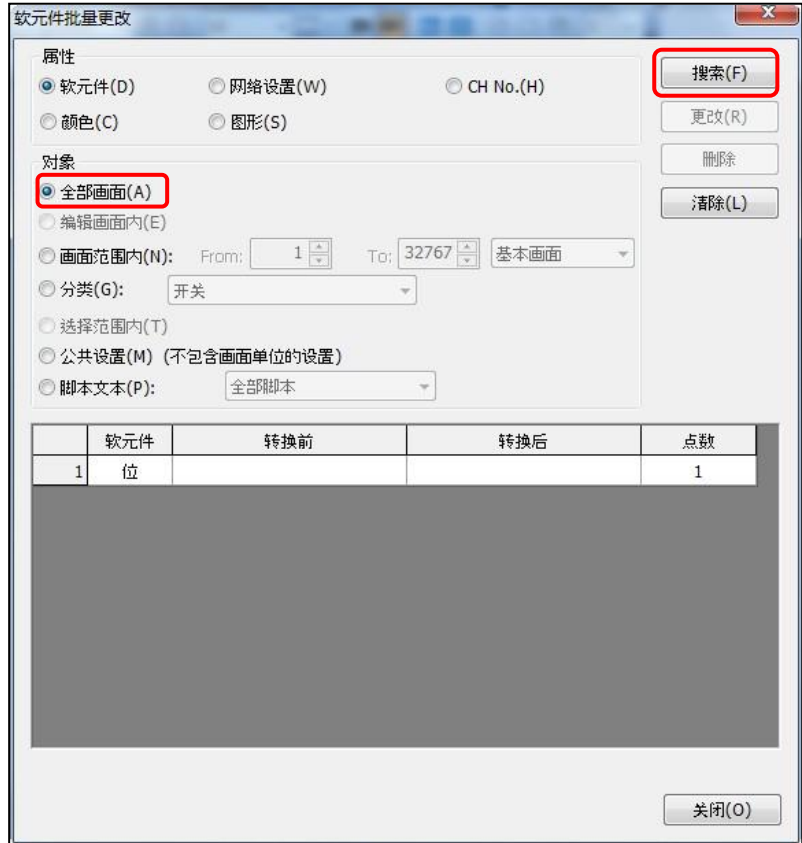
(2) 输入输出信号的前头输入输出编号
从[更改前]Xn0 更改为[更改后]Xm0 时，将 m 设置为更改处的前头输入输出编号 16 等分的值。前头输入输出编号从 0H 更改 80H 时，将 U00-G0 更改为 U08-G0。从 X00~X1F 更改时，将[点数]设置为 32。AnyWireASLINK 管理模块一个卡槽占据 32 点的输入输出信号 (Xn0~X(n+1)F、Yn0~Y(n+1)F)。请注意更改时不要更改不必要的点。

6.4.2 画面中设置的软元件的更改

(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]U00-G0、[转换后]U08-G0、[点数]10257，点击[更改]。U00-G0~U00-G10256 将更改到 U08-G0~U08-G10256。

	软元件	更改前	更改后	点数
1132	字	SD1449	SD1449	1
1133	字	SD1450	SD1450	1
1134	字	U00-G0	U08-G0	10257
1135	字	U00-G1	U00-G1	1
1136	字	U00-G2	U00-G2	1
1137	字	U00-G3	U00-G3	1
1138	字	U00-G4	U00-G4	1
1139	字	U00-G5	U00-G5	1
1140	字	U00-G6	U00-G6	1
1141	字	U00-G7	U00-G7	1

- 输入输出信号的起始输入输出号的更改

更改输出信号(Y 软元件)：设置[转换前]Y0000、[转换后]Y0080、[点数]18，点击[更改]。Y0000~Y0011 将更改到 Y0080~Y0091。

	软元件	更改前	更改后	点数
1102	字	GD63995	GD63995	1
1103	字	GD64000	GD64000	1
1104	位	Y0000	Y0080	18
1105	位	Y0001	Y0001	1
1106	位	Y0010	Y0010	1
1107	位	Y0011	Y0011	1
1108	位	Y0020	Y0020	1
1109	位	Y0021	Y0021	1
1110	位	Y0030	Y0030	1
1111	位	Y0031	Y0031	1

6.4.3 [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 6.4.2 同样的步骤执行。

选择[公共设置]后更改的软元件：U00-G0~U00-G12288、Y0010。

- 缓冲存储器的起始输入输出号的更改

设置[转换前]U00-G0、[转换后]U08-G0、[点数]12289，点击[更改]。U00-G0~U00-G12288 将更改到 U08-G0~U08-G12288。

	软元件	更改前	更改后	点数
114	字	GD64000	GD64000	1
115	位	Y0010	Y0010	1
116	位	Y0030	Y0030	1
117	位	Y0050	Y0050	1
118	位	Y0070	Y0070	1
119	字	U00-G0	U08-G0	11289
120	字	U00-G1	U00-G1	1
121	字	U00-G20	U00-G20	1
122	字	U00-G4096	U00-G4096	1
123	字	U00-G8192	U00-G8192	1

- 输入输出信号的起始输入输出号的更改

设置[转换前]Y0010、[转换后]Y0090、[点数]1，点击[更改]。Y0010 将更改到 Y0090。

	软元件	更改前	更改后	点数
113	字	GD62041	GD62041	1
114	字	GD64000	GD64000	1
115	位	Y0010	Y0090	1
116	位	Y0030	Y0030	1
117	位	Y0050	Y0050	1
118	位	Y0070	Y0070	1
119	字	U00-G0	U08-G0	11289
120	字	U00-G1	U00-G1	1
121	字	U00-G20	U00-G20	1
122	字	U00-G4096	U00-G4096	1

6.4.4 [脚本文本]中使用的软元件的更改

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 6.4.2 同样的步骤执行。选择[脚本文本]后更改的软元件：U00-G8192~U00-G10321、X0011~X0012、Y0000~Y0011

- 缓冲存储器的起始输入输出号的更改

设置[转换前] U00-G8192、[转换后] U08-G8192、[点数] 2130，点击[更改]。U00-G8192~U00-G10321 将更改到 U08-G8192~U08-G10321。

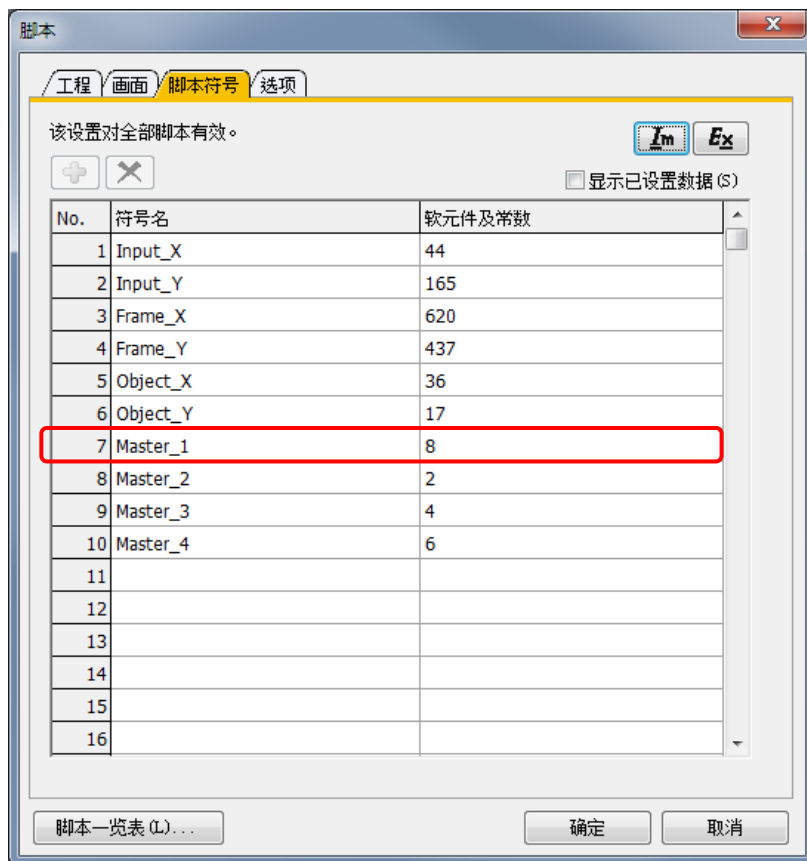
	软元件	更改前	更改后	点数
319	字	SD1452	SD1452	1
320	字	SD1453	SD1453	1
321	字	U00-G8192	U08-G8192	2130
322	字	U00-G8960	U00-G8960	1
323	字	U00-G9984	U00-G9984	1
324	字	U00-G10256	U00-G10256	1
325	字	U00-G10320	U00-G10320	1
326	字	U00-G10321	U00-G10321	1
327	字	U02-G8192	U02-G8192	1
328	字	U02-G9984	U02-G9984	1

- 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]X0011、[转换后]X0091、[点数]2，点击 [更改]。X0011~X0012 将更改到 X0091~X0092。如要更改输出信号(Y 软元件)，请设置[转换前]Y0000、[转换后]Y0080、[点数]18，点击[更改]。Y0000~Y0011 将更改到 Y0080~Y0091。

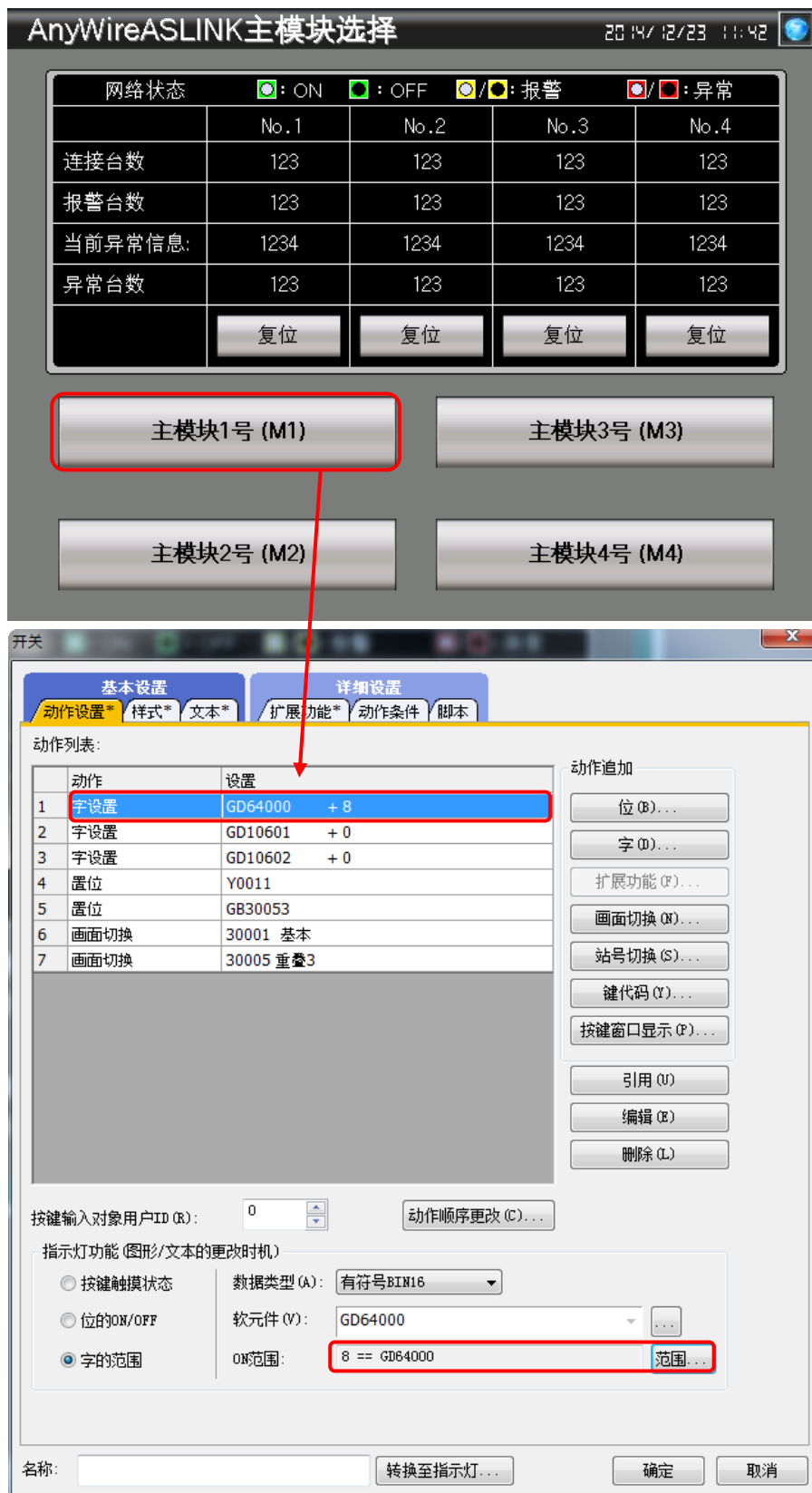
	软元件	更改前	更改后	点数
278	字	GS655	GS655	1
279	位	X0011	X0091	2
280	位	Y0000	Y0080	18
281	位	X0031	X0031	1
282	位	X0032	X0032	1
283	位	X0051	X0051	1
284	位	X0052	X0052	1
285	位	X0071	X0071	1
286	位	X0072	X0072	1
287	位	Y0000	Y0000	1

6.4.5 [脚本符号]的更改



更改[脚本符号]。将[符号名]为「Master_1」的[软元件及常数]从 0 更改到 8。

6.4.6 开关的更改



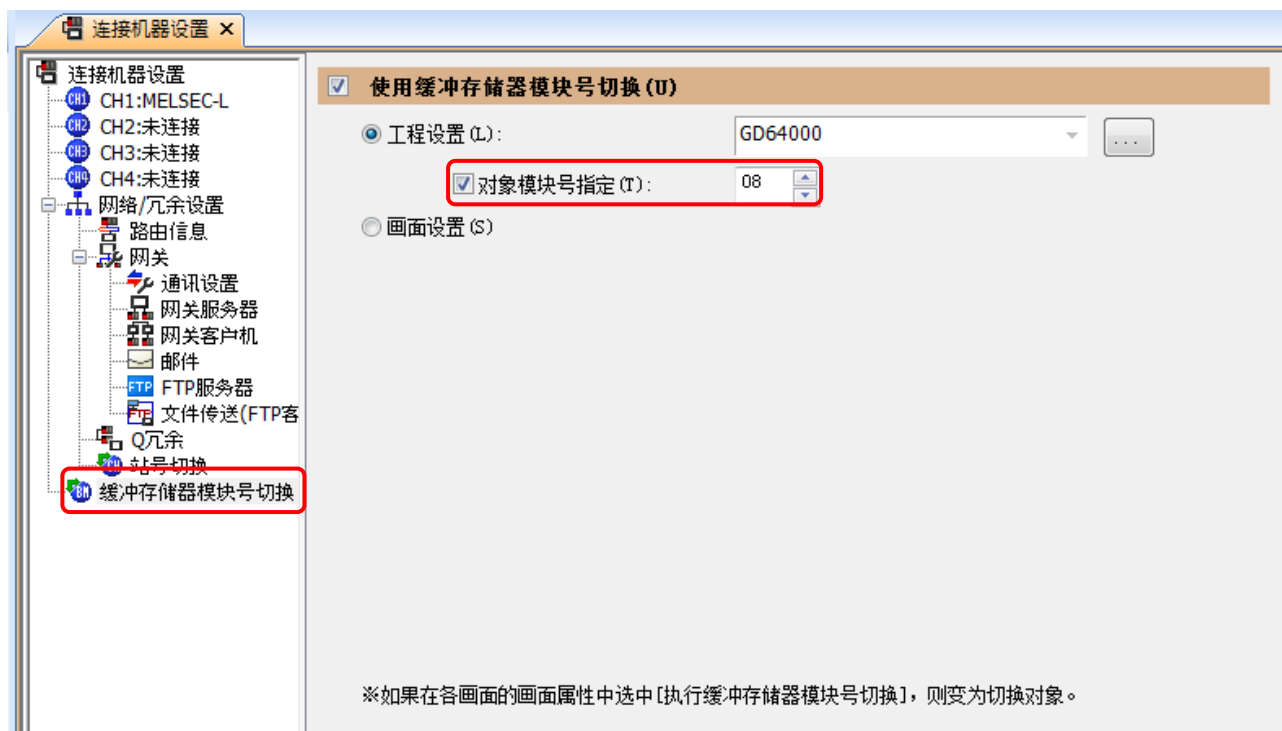
更改基本画面 B-30000 的「主模块 1 号 (M1)」的设置。将 GD64000 的值从 0 更改到 8。相应地将[指示灯功能 (图形/文本的更改时机)]的[ON 范围]更改为「8==GD64000」。

6.4.7 [缓冲存储器模块号切换]的更改

(1) 选择[公共设置]-[连接机器设置]菜单



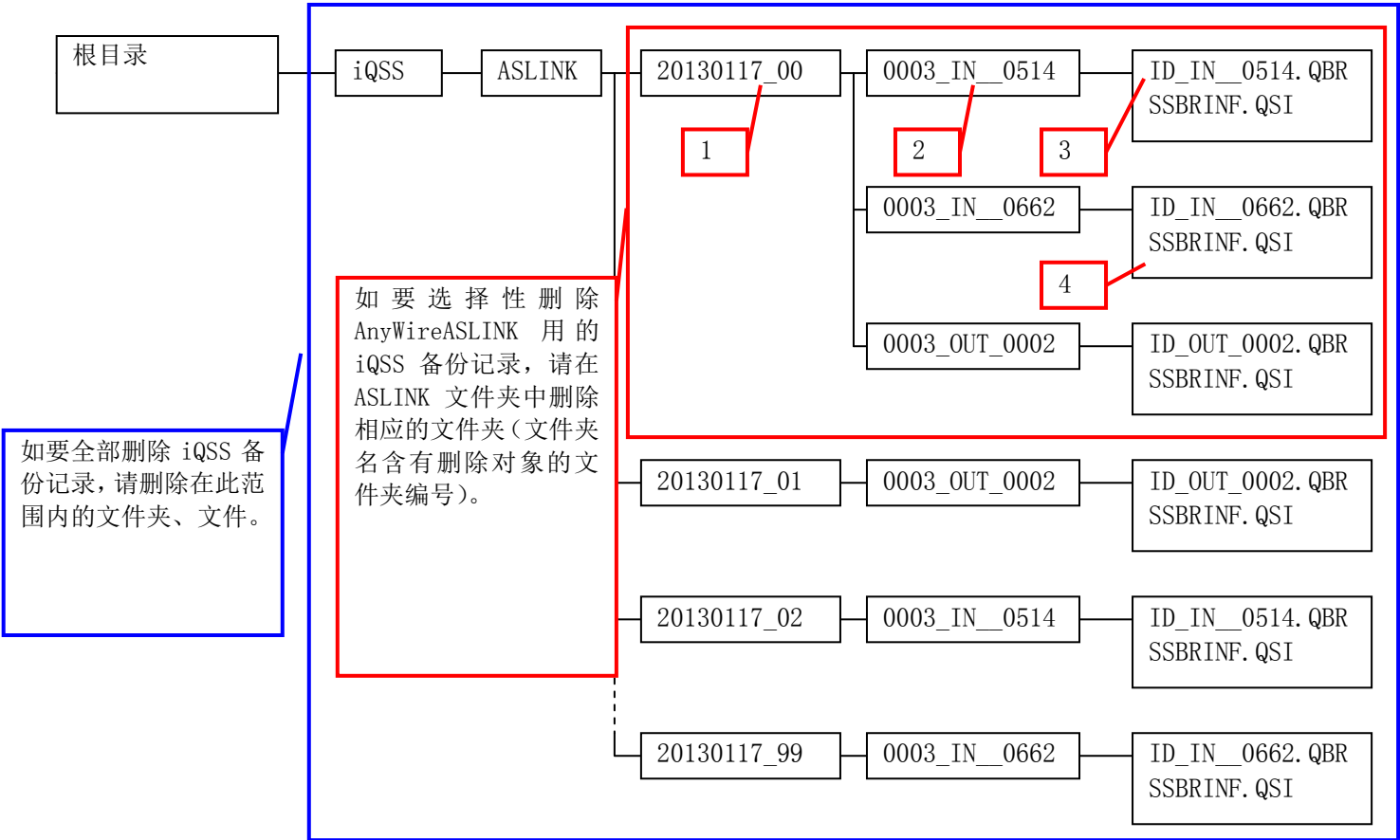
(2) 选择[缓冲存储器模块号切换]，将[对象模块号指定]从 00 更改为 08。



6.5 关于 iQSS 备份文件夹构成

6.5.1 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「6.5.2 文件夹名、文件名详细」。



6.5.2 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)
—— 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)
—— 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
—— I/O 编号(4 位(16 进制))
3. ID _ IN _ 0514.QBR ... 备份文件名
—— 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。

6.6 关于主模块的更改

减少主模块的台数时，请按照以下步骤修改基本画面 B-30000 和脚本。下例记载了删除主模块的 I/O 4 号的步骤。

6.6.1 基本画面的更改



更改基本画面 B-30000。删除「网络状态」中「No. 4」的项目以及「主模块 4 号 (M4)」的开关。

6.6.2 脚本的更改

删除基本画面 B-30001 的脚本 No. 30079、No. 30080 以及基本画面 B-30006～30051 的脚本 No. 30081。