

操作说明书

机种名称

LEC 串行通信资料

型号 / 系列

LEC□6 Series

SMC株式会社

1.	安全上的注意	2
2.	概要	4
3.	关于 LEC 串行通信	5
4.	操作例 1 按步数据信息进行动作运行	6
5.	操作例 2 编辑步数据信息	8
6.	操作例 3 读取位置、速度信息	8
7.	操作例 4 直接指定位置、速度等信息的运行、方法	9
8.	通信规格	11
9.	Function 详细	19
10.	存储分配图	27
11.	CRC 计算方法	31



LEC□6 Series/串行通信

1. 安全注意事项

这里所指的注意事项，记载了产品应如何安全正确的使用，以防止对您及他人造成损伤。根据其潜在的危險程度，将有关注意事项分成“注意”，“警告”和“危险”三种标志。不论哪种标志，都是与安全相关的重要内容，故在遵守 ISO 4414^{*1)}、JIS B 8370^{*2)}以及其他安全规则的同时，也必须遵守此项内容。

- *1) ISO 4414: Pneumatic fluid power -- General rules relating to systems
ISO 4413: Hydraulic fluid power -- General rules relating to systems
IEC 60204-1: Safety of machinery -- Electrical equipment of machines (Part 1: General requirements)
ISO 10218-1992: Manipulating industrial robots -- Safety
JIS B 8370: 气压系统通则
JIS B 8361: 油压系统通则
JIS B 9960-1: 机械类的安全性-机械的电气装置(第1部:一般要求事项)
JIS B 8433-1993: 键控工业机器人-安全性等

*2) 劳动安全卫生法等



注意

误操作时，可能造成人员及设备的损伤。



警告

误操作时，可能造成人员的死亡或重伤。



危险

在紧迫的危险状态下，如不回避可能造成人员的死亡或重伤。



警告

1. 请系统的设计者或选型者来判断元件的选型是否合适。

本手册所述产品使用条件多种多样，应由系统设计者或选型者，根据需要作相应的分析和测试后，决定所选元件是否适合该系统。设计者具有保证满足系统所期望的性能和安全的责任。请根据产品的最新资料，检查型号的全部内容，并考虑到元件可能会出现故障的情况，然后进行系统构成工作。

2. 请有充分知识和经验的人进行操作。

本手册刊载的产品，如果操作失误，将造成安全性方面的损失。对设备进行组装、操作和维护等时，应由有足够知识和经验的人进行。

3. 在未确定安全之前，绝对不可对设备进行操作或将其元件拆卸。

- 1) 在设备点检和维修之前，必须确认被驱动物体已进行了防下落或防失控处理。
- 2) 拆卸本产品时，请确认已进行了上述安全处理；切断对该设备供给的动力源，参考产品个别注意事项，方可从进行拆卸作业。
- 3) 设备再启动之前，要确认不会发生误操作等失控的情况。

4. 在下列条件和环境下使用时，请在考虑安全对策的同时，与本公司联系。

- 1) 明确记载规格以外的条件和环境，或室外、阳光直射的情况下使用。
- 2) 使用于原子能、铁路、航空、车辆、医疗器械、食品及饮料机械、娱乐设备、紧急切断回路、压力机用离合器及制动回路、安全机器等。
- 3) 对人身和财产有很大影响，特别是在安全方面有要求的使用。
- 4) 用于互锁回路时，请设置应对故障的机械式的保护机能，进行2重互锁。另外进行定期检查以确认其是否正常动作。



LEC□6 Series/串行通信

1. 安全注意事项



注意

本公司的产品是面向制造业提供的。

这里所述本公司产品是以面向制造业为主，用于和平而提供的。

如果要用于制造业以外的用途时，请与本公司联系，根据需要请交换规格书、合同书。

如果有什么不清楚的地方，请与最近的营业所联系。

保证以及免责事项/适合用途的条件

使用产品时，适用以下「保证以及免责事项」、「适合用途的条件」。

请在确认、允许以下内容的基础上，适用本公司产品。

『保证以及免责事项』

①本公司产品的保证期间为，从使用开始 1 年内，或者从购入开始 1 年半内。^{※3)}

另外产品有设定的使用次数，行走距离，更换零件等，请与最近的营业确认。

②保证期间内由于本公司的责任，产生明显的故障以及损伤时，由本公司提供代替品或者进行必要的零件更换。在此所述的保证，是指对本公司产品各个部件的保证，由于本公司产品故障引发的损害，属于保证外。

③也请参考其他产品个别的保证及免责事项，在理解的基础上使用。

^{※3)} 真空吸盘不适用于使用开始 1 年内的保证期限。

真空吸盘为消耗品，保证期间为购入后 1 年以内。

但是，即使在保证期间内，由于使用产生的磨损或者橡胶材质的劣化等事项属于产品保证适用范围外。

『使用用途的条件』

出口海外时，请必须遵守经济产业省规定的法令(外国汇兑及外国贸易法)、手续。

2. 概要

电缸用控制器（以下简称“控制器”） LEC□6系列中，搭载了RS485传输线路（物理层）的LEC串行通信口。通过该串行通信，可向控制器进行以下操作。

- ① 根据已存储的步数据信息进行动作、运行。
- ② 编辑步数据信息。
- ③ 读取位置、速度的数据信息。
- ④ 直接指示位置、速度、推力等信息进行动作、运行。

本手册是LEC串行通信使用方法的说明资料。

请参照电缸和LEC□6系列的操作说明书一起使用。

3. 关于 LEC 串行通信

它是用RS485传输线路和ModbusRTU协议进行的串行通信。

用这种串行通信，在控制器内可对内部继电器和数据寄存器进行操作。

访问这些内部继电器和数据寄存器可以对控制器内部的数据进行确认、变更并按指令动作。

这个串行通信的存储概略表如下：

名称	地址	内容
内部继电器	X40～X4F Y0～Y1F, Y30	表示并行IO输入输出状态和控制器内状态的内部继电器。与并行IO发出指示一样，通过操作内部继电器即可操作控制器。
步数据寄存器	D0400～D043F	保存步数据信息的存储器。在这里写入数据，可变更步数据的内容。
状态数据寄存器	D9000～D9006	保存运行时的位置、速度等信息。 读取这个数据，即可得到现在的位置和速度信息。
指定数据运行存储器	D9100～D9110	用于直接执行指定位置和速度等信息的动作时使用。 在这里指定位置和速度等信息，然后执行指定的动作。

注意

- 上述范围外的地址里，控制器已预先存入一些固定值。请不要操作该范围外的存储器。
- 上述存储器中，步数据信息（D0400～D043F）是存储在EEPROM（电可擦可编程只读存储器）中。EEPROM的可写入次数只有10万次，请避免频繁写入。
- 上述存储器中，内部继电器、状态数据、指定数据运行的信息，存储在RAM（随机存取存储器）中。如果关掉控制电源，设定值将重置。

以下为通信帧。通过“Function”指定数据的读取、写入等，通过“Data”指示地址和数据。

“Function”详细请参考“9.Function详细”

ID	Function	Data	CRC Check
1byte	1byte	N byte	2byte

ID是指定通信端控制器的ID。控制器出厂时设定为1。

CRC 校验是相对于通信数据16bit CRC 校验编码。受信方确认收到的数据和CRC校验编码，如果是正常的通信即开始处理。详细请参考“8. 通信规格”。

4. 操作例1 按步数据信息进行动作、运行

本部分内容列举了和运行并行IO一样通过操作内部继电器，来运行指定步数据的例子。

关于通信规格、Function和内部继电器的详细，请参考第8部分。

发送数据的例子，是通信控制器ID = 1时的情况。而且，CRC编码是根据发送数据进行逻辑运算后得到的值。校验码根据发送数据的变化而变化。

另外，下面各操作示例因接入电源后的操作过程状态而变化。请注意实际有可能发生与操作例不同的情况。

4.1 动作准备

4.1.1 指示串行通信有效，在Y30中写入1。

【发送数据例】

01 05 00 30 FF 00 8C 35

【CRC计算例】

35 8C

4.1.2 伺服ON，在SVON (Y19)中写入1。

【发送数据例】

01 05 00 19 FF 00 5D FD

【CRC计算例】

FD 5D

4.1.3 确认SVRE (X49) 变为1。

【发送数据例】 读出 (X40-X4F)。

01 02 00 40 00 10 78 12

【CRC计算例】

12 78

【返回数据例】

01 02 02 00 02 38 79

↑ X49是第2位的值 (00000010)。

4.2 原点复归

4.2.1 在SETUP (Y1C) 中写入1，即开始归零动作。

【发送数据例】

01 05 00 1C FF 00 4D FC

【CRC计算例】

FC 4D

4.2.2 假如归零动作结束，SETON (X4A)变成1。请确认SETON (X4A)是否变成1。

【发送数据例】 读出 (X40-X4F)。

01 02 00 40 00 10 78 12

【CRC计算例】

12 78

【返回数据例】

01 02 02 00 0E 38 7C

↑ X4A是第3位 (00001110)。

4.2.3 动作结束，将SETUP (Y1C) 置0。

【发送数据例】

01 05 00 1C 00 00 0C 0C

【CRC计算例】

0C 0C

4.3 动作（运行）

在 Y10～Y17 写入步数据号，选定运行的步程序。

【发送数据例】 选择步数据 No. 2。

01 0F 00 10 00 08 01 02 BE 97

【CRC 计算例】

97 BE

【发送数据例】 在 DRIVE (Y1A) 中写入 1，开始动作。

01 05 00 1A FF 00 AD FD

【CRC 计算例】

FD AD

4.4 动作结束

读取 (X40～X4F)，如果 INP (X4B) 为 1，可确认动作结束（已到达定位范围内）。

【发送数据例】

01 02 00 40 00 10 78 12

【CRC 计算例】

12 78

【返回数据例】

01 02 02 00 0E 38 7C

↑ X4B 是第 4 位 (00001110)。

【发送数据例】 将 DRIVE (Y1A) 置 0。

01 05 00 1A 00 00 EC 0D

【CRC 计算例】

0D ED

5. 操作例 2 编辑步数据信息

步数据信息存储在 D0400~D043F 中，编辑步数据时，请对这些地址进行写入。

【发送数据例】 将 STEP No. 1 的“位置”写为 150.00 (mm) 。

【CRC 计算例】

01 10 04 12 00 02 04 00 00 3A 98 52 B0

B0 52

↑ (1)

↑ (2)

(1) 写入地址: $0400h + 10h \times 1$ 【STEP No.】 + $02h = 0412h$

(2) 写入数据: $150.00 \times 100 = 15000 = 3A98h$

6. 操作例 3 读取位置、速度信息

当前的位置和速度信息存储在 D9000~D9006，读取这些地址，可确认这些信息。

【发送数据例】 读取位置数据 (D9000)。

【CRC 计算例】

01 03 90 00 00 02 E9 0B

0B E9

【返回数据例】

01 03 04 00 00 3A 98 E9 39

3 A98h = 15000 → 150.00mm

7. 操作例 4 直接指定位置、速度等信息的动作、运行

7.1 动作准备

7.1.1 指示串行通信有效，在Y30中置“1”。

【发送数据例】

01 05 00 30 FF 00 8C 35

【CRC计算例】

35 8C

7.1.2 在SVON (Y19)中写入1, 伺服ON。

【发送数据例】

01 05 00 19 FF 00 5D FD

【CRC计算例】

FD 5D

7.1.3 确认SVRE (X49) 变为1。

【发送数据例】 读出 (X40-X4F)

01 02 00 40 00 10 78 12

【CRC计算例】

12 78

【返回数据例】

01 02 02 00 02 38 79

↑ X49是第2位的值 (00000010)

7.2 原点复归

7.2.1 在SETUP (Y1C)中写入1, 即开始归零动作。

【发送数据例】

01 05 00 1C FF 00 4D FC

【CRC计算例】

FC4D

7.2.2 假如归零动作结束, SETON (X4A) 变成1。请确认SETON (X4A) 已经变成1。

【发送数据例】 读出 (X40-X4F)。

01 02 00 40 00 10 78 12

【CRC计算例】

12 78

【返回数据例】

01 02 02 00 0E 38 7C

↑ X4A是第3位 (00001110)

7.2.3 动作结束, 将ETUP (Y1C) 置0。

【发送数据例】

01 05 00 1C 00 00 0C 0C

【CRC计算例】

0C0C

7.3 运行

在D9102~D9110中写入步数据后，在D9100中写入1，按照D9102~D9110的内容进行动作。（各地址详细，请参考“10.2 直接指定位置、速度等信息的运行”）

以下是移动到 300mm 位置的示例。

7.3.1 写入步数据

【发送数据例】

【CRC计算例】

在 D9102~D9111 中写入表示动作内容的步数据。 48 72

01	10	91	02	00	10	20	00	01	01	F4	00	00	75	30	13	88	13	88	00	00	00	00	00	14
							(1)	(2)		(3)			(4)	(5)	(6)	(7)	(8)							
00	64	00	00	00	00	00	00	00	00	00	00	00	64	72	48									
(9)		(10)					(11)			(12)														

上述发送数据示例的相关内容如下。

- | | | |
|------|---------------|----------|
| (1) | 动作方法 | : 绝对坐标 |
| (2) | 速度 | : 500 |
| (3) | 位置 | : 300.00 |
| (4) | 加速度 | : 5000 |
| (5) | 减速度 | : 5000 |
| (6) | 推压力 | : 0 |
| (7) | 阈值（范围值） | : 0 |
| (8) | 推压速度 | : 20 |
| (9) | 定位最大推力 | : 100 |
| (10) | 区域输出端 1(区域 1) | : 0.00 |
| (11) | 区域输出端 2(区域 2) | : 0.00 |
| (12) | 定位幅 | : 1.00 |

7.3.2 开始运行

在 D9100 中写入 0100h，动作开始。

【发送数据例】

【CRC计算例】

01	10	91	00	00	01	02	01	00	27	09	09	27
----	----	----	----	----	----	----	----	----	----	----	----	----

7.3.3 动作结束确认

读取X40~X4F，如果INP（X4B）为0，可确认动作结束（已经到达定位范围内）

【发送数据例】

【CRC计算例】

01	02	00	40	00	10	78	12	12	78
----	----	----	----	----	----	----	----	----	----

【返回数据例】

01	02	02	00	0E	38	7C
↑ X4B是第4位。（00001110）						

8. 通信规格

和上位计算机的串行通信是通过Modbus 协议进行数据交换的单主站为多从站的方式进行。上位机是主站，控制器是从站。

从主站（上位机）发出询问，从站(控制器)接收询问回复反馈。（从站不能发出询问）
（但是，用广播的方式发出询问，从站将不会有反馈）

項目	方式・条件
接口	依照RS485
通信方式	半双工通信
调用方式	同步调用
通信模式	Modbus Protocol RTU
波特率 (bps)	可根据控制器的参数设定选择下述波特率 9600 / 19200 / 38400 / 57600 / 115200 / 230400 (初期值38400)
数据位	8bit
停止位	1bit
校验位	无

注意

使用示教盒（LEC-T1）时，请将通信速度（波特率）设定在 115200(bps) 以下。
如果设定成比 115200(bps) 更大的值，示教器与控制器将不能通信。

8.1 通信帧

通信帧如下：

Address	Function	Data	CRC Check
1Byte	1Byte	N Byte	2Byte

8.1.1 Address

设定从站地址。

当该值与基本参数内的“控制器 ID”一致时，通信数据判断为本站的数据。

但是，当该值为“0”时，是广播指定（全局地址）

（此时，没有反馈）

“Address”的值和意思如下所述：。

0 (00)	:	广播指定（从站发出的反馈无）
1 (01h) ~ 255 (FFh)	:	控制器 ID

8.1.2 Function

表示本控制器中可使用的Function编码和功能。

（各编码详细，请参考“9. Function详细”）

通信代码 (Hex)	名称	功能	广播指定
01h	读取输出信号 (Y)	Y 接点的读取	不可
02h	读取输出信号 (X)	X 接点的读取	不可
03h	读取数据 (D)	参数及各种数据的读取	不可读取X、Y 接点
05h	强制信号输出 (Y)	Y 接点的 1 点写入	可 (*1)
08h	回送校验	根据回送校验测试通信	不可
0Fh	输出信号全部写入	Y 接点全部写入	可 (*1)
10h	写入数据 (D)	参数及各种数据的写入	不能写入Y接点

(*1)广播时，电机控制器无反馈。

8.1.3 Data

对应各Function编码的数据，最大 256Byte。

8.1.4 CRC Check

相对于通信数据 16bit CRC 校验编码。

因为电机控制器的通信规格是“Modbus Protocol 互换”，CRC 校验的方法也是以 Modbus Protocol 的 RTU 模型为依据。（RTU 模型的 CRC 校验是 CRC-16 多项式数据=A001h）

发送时，计算 CRC 数据附加于发送帧内发送。

接收时，计算接收数据的 CRC，与附加接收帧内的 CRC 数据比较，CRC 不一致的时候，无应答（无反馈）。

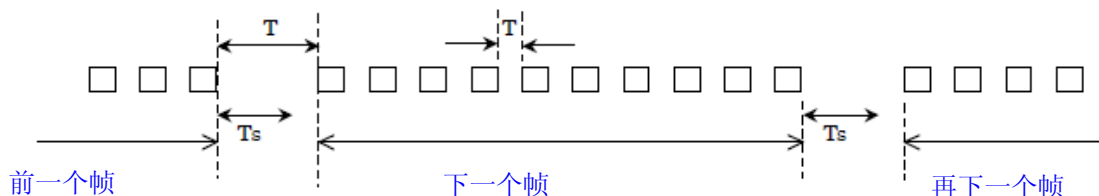
8.2 帧间隔 (Silent INT)

Modbus 协议的 RTU 模型中，利用时间（无通信时间）进行通信帧的定界符判定。

即，通信帧的字符间间隙 (T)，与帧间隔 (Ts) 相比较：

$T \geq T_s$ 时：判断为定界符（通信帧的最后）

$T < T_s$ 时：判断为通信帧的中途



其中，帧间隔 (T_s) 是在“电机参数 (Motor)”内的帧间隔 (Silent INT) 设定。但是，其单位是基本间隔 ($=3.5\text{Char}$) 的倍数，请注意。1Char 是 10bit。

电机参数内的“帧间隔 (Silent INT)”初始值是“1 倍”。

8.3 通信延迟时间

收到询问（从 MASTER 发出的发送帧）后，到反馈发送开始之前，可设定延迟时间。这个时间由电机参数内的“最小通信延迟时间 (Min Resp delay)”来设定。

电机参数 (Motor) 内的“最小通信延迟时间 (Min Resp delay)”的初始值为“5ms”。

8.4 应答时间

MASTER 询问发送后，到该控制器发送反馈结束之间的时间 T_x 如下。MASTER 再次发送询问时，从前一次发信开始等待 T_x 时间以上，然后请再次发送。（ T_x 中不包含询问的通信时间）

$$T_x = T_s + T_0 + T_d + (10 \times Br / Kbr) \quad [\text{ms}]$$

T_s : 帧间隔 (Silent INT)

($1 / \text{波特率 bps} \times (\text{帧间隔 (Silent INT) 设定值} \times 3.5\text{Char} \times 10\text{bit})$)

T_0 : 内部处理时间 (4ms) $\times 1.5$ (安全系数)

T_d : 通信延迟时间

Br : 反馈信息的字节数

Kbr : 波特率 [kbps]

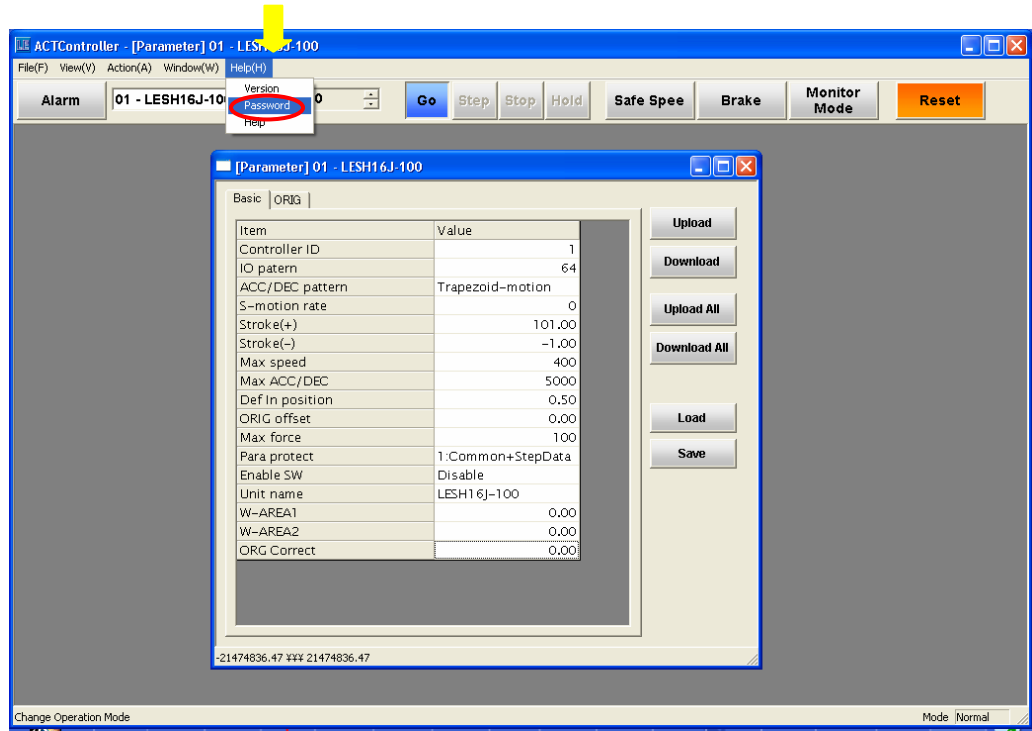
8.6 参数修改方法（例）

以下为“波特率 (Comm speed)”“帧间隔 (Silent INT)”“最小延迟时间 (Min Resp delay)” 的修改方法和顺序。请按照①～⑨的顺序进行。

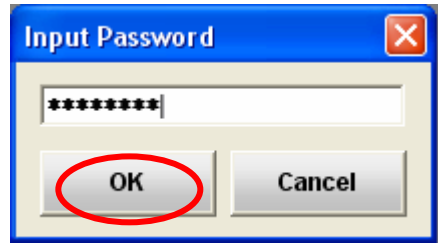
(例)

波特率 (Comm speed)	38400	⇒	57600
帧间隔 (Silent INT)	1		10
最小延迟时间 (Min Resp delay)	5		50

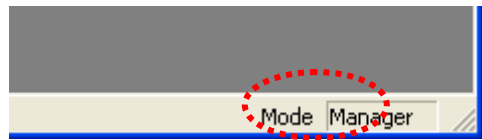
① 启动控制器设定软件（ACT Controller，点击 Help（下图红色部分），选择 Password。



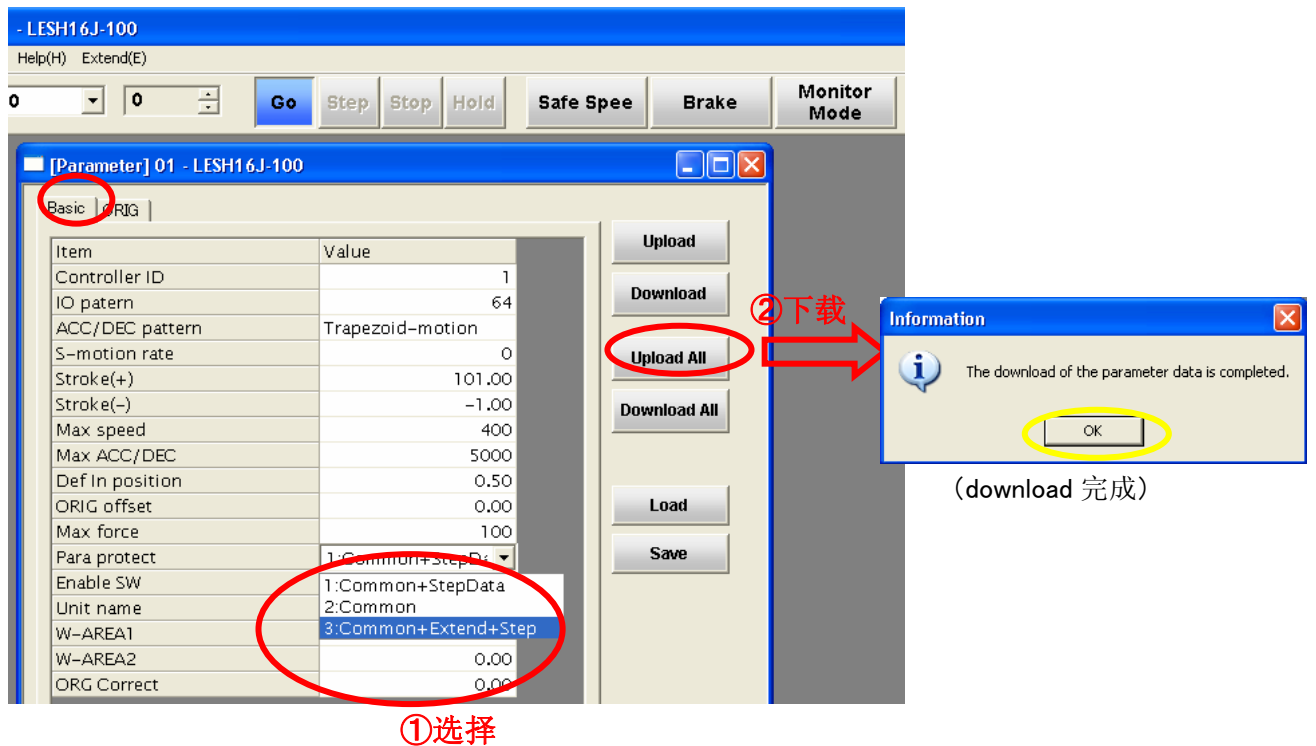
② 在密码输入框内，输入 Password 点击“OK”即可。



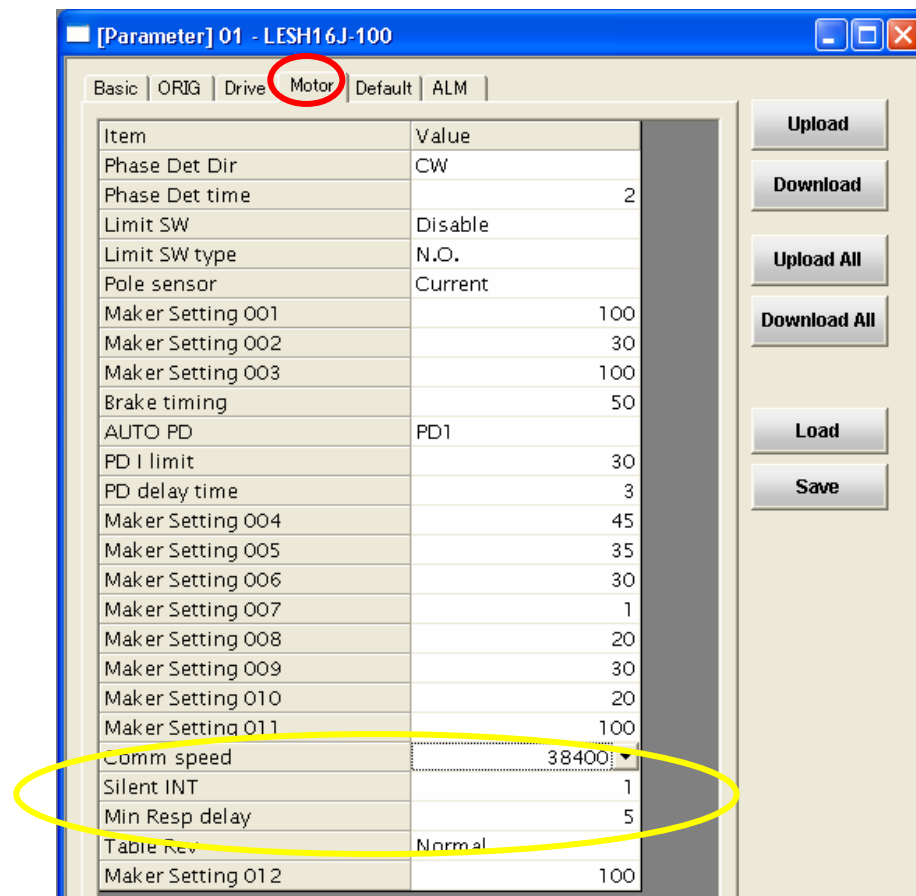
输入成功，画面右下方显示“Manager”。



- ③ 基本参数(Basic)的参数保护中追加了“3:Common+Extend+Step”，选择“3:Common+Extend+Step”之后，请点击 [Download]。

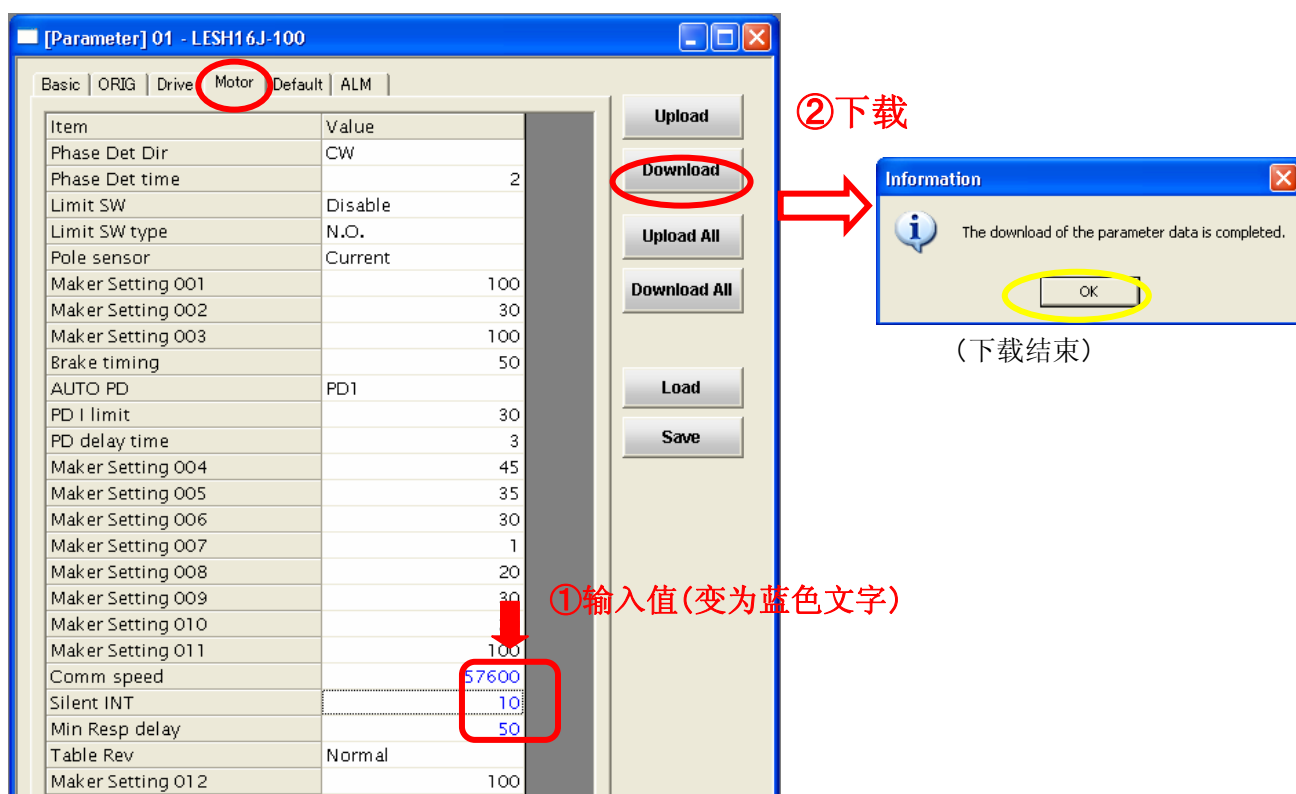


- ④ 扩充参数将会显示，请选择电机参数（Motor）。
（“Comm speed”、“Silent INT”、“Min Resp delay”可变更。）



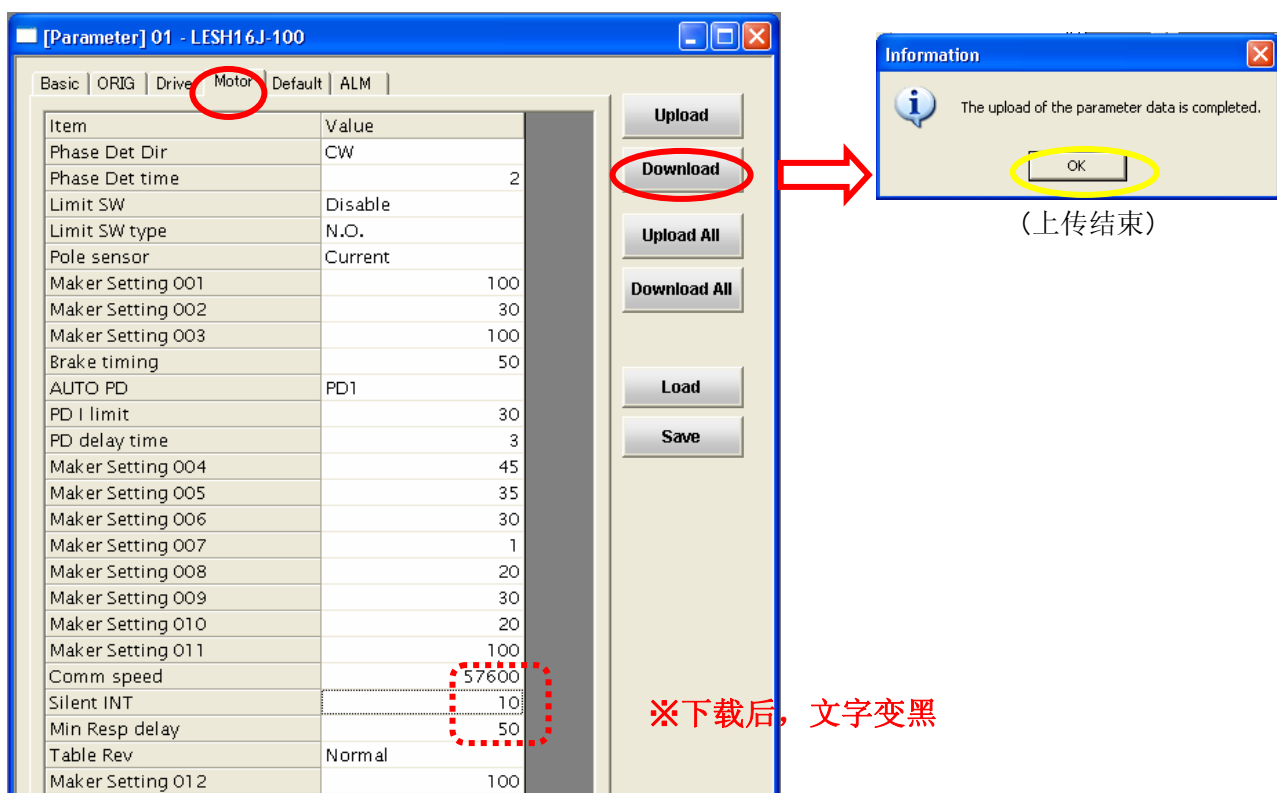
⑤ 修改“Comm speed”、“Silent INT”、“Min Resp delay”的值，点击“Download”，该值即存入至控制器。

(※修改的值文字为蓝色，下载后修改过的值文字为黑色。)

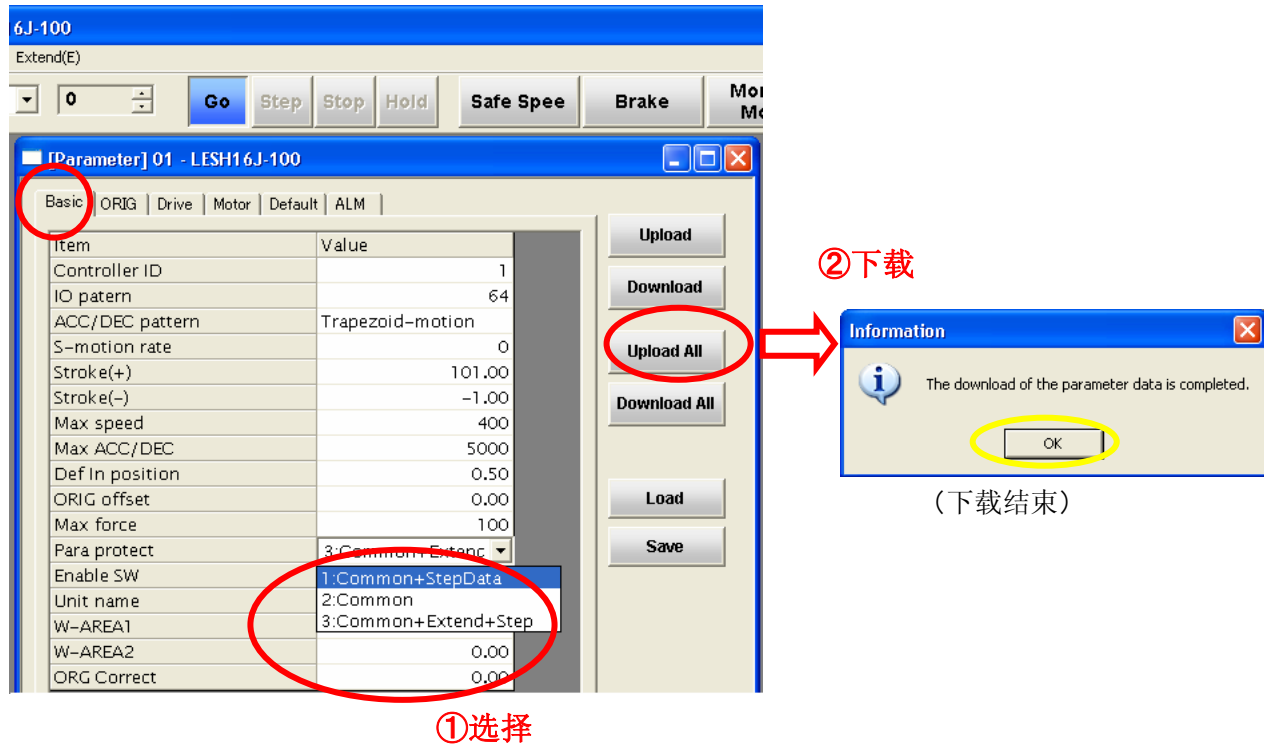


⑥ [Download] 后，请点击 [Upload] 确认。

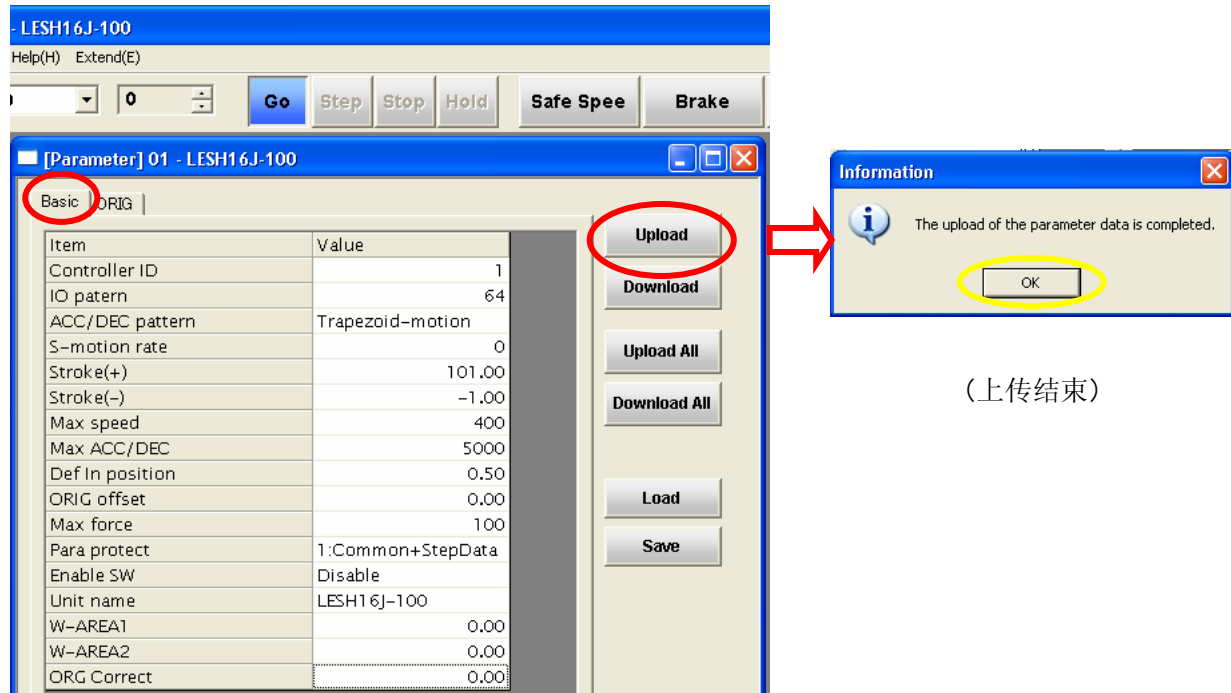
请确认设定值已经修改。



⑦ 如果设定值已修改，选择基本参数(Basic)的参数保护 [1:Common+StepDate] 选项，点击“Download”还原。



⑧ [download] 后，请点击 [upload] 确认。
基本参数的参数保护“1: Common+StepDate”已经修改，参数修改结束。



警告

- 未在操作方法内提示的参数，请勿修改。
- 请不要点击“Download All”。如果其他参数变更了，导致不可变更的参数也变更了，将可能造成电缸及控制器发生起火、误动作、破损等现象。

9. Function 详细

9.1 输出信号读取 (01h) ※不可广播

进行Y接点的读取。

(例) 从ID=3控制器读取Y10(h)～Y1F(h) 16bit

●询问

			Example
1	Address		03
2	Function		01
3	DATA	Read start number (H)	00
4		Read start number (L)	10
5		Read points (H)	00
6		Read points (L)	10
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

●正常反馈

			Example
1	Address		03
2	Function		01
3	DATA	Data bytes	02
4		Data 1	00
5		Data 2	12
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

bit7

Y17	Y16	Y15	Y14	Y13	Y12	Y11	Y10
-----	-----	-----	-----	-----	-----	-----	-----

bit0

Y1F	Y1E	Y1D	Y1C	Y1B	Y1A	Y19	Y18
-----	-----	-----	-----	-----	-----	-----	-----

上面数据1～3的示例为Y1F～Y10=0001 0010 0000 0000的情况。

●异常时反馈

			Example
1	Address		03
2	Function (*1)		81
3	Error code (*2)		01
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

(*1) 异常时反馈的Function编码是将询问Function编码MSB (最高位) 做为1的值。

(*2) 报警编码请参考“9.8报警编码”。

9.2 输入信号读取(02h) ※不可广播

进行 X 接点的读取。（输出信号读取 (01h) 和格式相同。）

（例）从ID=3的控制器读取X40～X4F(h) 16bit

●询问

			Example
1	Address		03
2	Function		02
3	DATA	Read start number (H)	00
4		Read start number (L)	40
5		Read points (H)	00
6		Read points (L)	10
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

●正常反馈

			Example
1	Address		03
2	Function		02
3	DATA	Data bytes	02
4		Data 1	04
5		Data 2	8E
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

bit7

bit0

X47	X46	X45	X44	X43	X42	X41	X40
X4F	X4E	X4D	X4C	X4B	X4A	X49	X48

上述数据1～3示例为X4F～X40=1000 1110 0000 0100的情况。

●异常时反馈

			例
1	Address		03
2	Function		82
3	报警编码		01
7	CRC16 (L)		计算值
8	CRC16 (H)		计算值

9.3 数据读取(03h) ※不可广播

进行参数及各种数据的读取。

(例) 从ID=3的控制器读取D0400~D0403 (2Byte=1word)

●询问

			Example
1	Address		03
2	Function		03
3	DATA	Read start number (H)	04
4		Read start number (L)	00
5		Read words (H)	00
6		Read words (L)	04
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

●正常反馈

			Example	
1	Address		03	
2	Function		03	
3	DATA	Data bytes	08	
4		word1 (H)	00	D0400
5		word1 (L)	01	D0400
6		word2 (H)	00	D0401
7		word2 (L)	28	D0401
8		word3 (H)	00	D0402
9		word3 (L)	00	D0402
10		word4 (H)	06	D0403
11		word4 (L)	40	D0403
12	CRC16 (L)		Calculated value	
13	CRC16 (H)		Calculated value	

Operation method (For 1h)
Speed (For 28h)
Position (For 640h)

●异常反馈

			Example
1	Address		03
2	Function		83
3	Error code		01
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

9.4 强制信号输出（05h）※可广播

写入Y接点1点。

（例）ID=3控制器Y19 (SV0N) 开启。

●询问

			Example
1	Address		03
2	Function		05
3	DATA	Contact number (H)	00
4		Contact number (L)	19
5		Contact state (H)	FF
6		Contact state (L)	00
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

} ON : FF00h
OFF: 0000h

●正常反馈

			Example
1	Address		03
2	Function		05
3	DATA	Contact number (H)	00
4		Contact number (L)	19
5		Contact state (H)	FF
6		Contact state (L)	00
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

●异常时反馈

			Example
1	Address		03
2	Function		85
3	Error code		01
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

9.5 回送校验(08h) ※不可广播

根据回送校验进行通信检测。

(例) 对ID=3控制器进行回送校验。

●询问

			Example
1	Address		03
2	Function		08
3	DATA	Test code (H)	00
4		Test code (L)	00
5		Data (H)	12
6		Data (L)	34
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

Specify 0000h

Arbitrary

●正常反馈

			Example
1	Address		03
2	Function		08
3	DATA	Test code (H)	00
4		Test code (L)	00
5		Data (H)	12
6		Data (L)	34
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

Received data is returned

●异常时反馈

			Example
1	Address		03
2	Function		88
3	Error code		01
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

9.6 输出信号全体写入(0Fh) ※可广播

Y接点进行全部写入。（写入数据的格式与输出信号读出(01h)的读取数据格式相同。）
（例）在ID=3控制器Y10~Y0F中进行写入。

●询问

			Example
1	Address		03
2	Function		0F
2	DATA	Write start number (H)	00
3		Write start number (L)	10
4		Write points (H)	00
5		Write points (L)	10
6		Number of data	02
7		Set data 1	0F
8		Set data 2	12
9	CRC16 (L)		Calculated value
10	CRC16 (H)		Calculated value

bit7

bit0

Y17	Y16	Y15	Y14	Y13	Y12	Y11	Y10
Y1F	Y1E	Y1D	Y1C	Y1B	Y1A	Y19	Y18

* Set contacts Y16, Y17, Y1D, Y1E, Y1F to "0".

上述数据1~2示例为Y1F~Y10=0001 0010 0000 1111的情况。

●正常反馈

			Example
1	Address		03
2	Function		0F
3	DATA	Write start number (H)	00
4		Write start number (L)	00
5		Write points (H)	00
6		Write points (L)	0E
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

●异常时反馈

			Example
1	Address		03
2	Function		8F
3	Error code		01
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

9.7 数据写入(10h) ※可广播

进行数据写入。

写入数据的格式与数据读出(03h)的读取数据格式相同)

(例) 向ID=3控制器写入D0410~D0413 4word。(2Byte=1word)

●询问

		Example		
1	Address	03		
2	Function	10		
3	DATA	Write start number (H)	04	D0410 Operation method (For 1h) D0411 Speed (For 28h) D0412 Position (For 640h) D0413
4		Write start number (L)	10	
5		Words written (H)	00	
6		Words written (L)	04	
7		Number of data	08	
8		word1 (H)	00	
9		word1 (L)	01	
10		word2 (H)	00	
11		word2 (L)	28	
		word3 (H)	00	
		word3 (L)	00	
12		word4 (H)	06	
13		word4 (L)	40	
14	CRC16 (L)		Calculated value	
15	CRC16 (H)		Calculated value	

●正常反馈


			Example
1	Address		03
2	Function		10
3	DATA	Write start number (H)	04
4		Write start number (L)	10
5		Words written (H)	00
6		Words written (L)	04
7	CRC16 (L)		Calculated value
8	CRC16 (H)		Calculated value

● 异常时反馈

		Example
1	Address	03
2	Function	90
3	Error code	01
7	CRC16 (L)	Calculated value
8	CRC16 (H)	Calculated value

9.8 报警代码

报警代码及其造成原因如下所示。

报警代码	名称	发生原因
01	非法功能	Function编码不正确（未经定义）。
02	地址范围以外	<p>1) 读取或写入开始地址代码是指定范围外的地址代码。</p> <p>2) 通过回送校验，测试编码不是0000h。</p> <ul style="list-style-type: none"> ▪ 写入了没被允许的号码（地址）。 ▪ 在参数保护设定中写入了没有允许的参数。 ▪ 运行电机（并行/串行）写入了没被允许写入的参数。 <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p style="text-align: center;"> 警告</p> <p>本资料中，只能使用允许使用的范围。其它的地址，制造商已经设定了范围值，修改这些设定范围，将有可能造成控制器和电缸的损坏。</p> </div>
03	读写点数超范围	<p>1) 读取或者写入的最终地址代码超出范围。</p> <p>2) 通信帧“Data”值超过256Byte。</p> <p>3) 在Function 05（强制输出信号）中，指定的“端子状态”数据不是FF00h(ON)或 0000h(OFF)</p> <p>4) Function 0F（输出信号全部写入）中，指定的“写入点数”比256大。</p> <p>5) 读出或者写入的指定尺寸为0。</p>

10. 存储分配图

这里指电机控制器的存储分配图。

仅能使用有效的寄存器及继电器，其它（包括未定义、不可使用）请绝对不要使用。

10.1 状态数据

地址D9000～D9008中，存储了电缸状态（现在位置、现在速度、现在推力）的信息。

地址	参数名	byte	设定范围	数据类型	内容
D9000	当前位置	4	± 2147483647	整数	表示当前位置 (单位:0.01mm)
D9002	当前速度	2	0～65535		表示当前速度 (单位:mm/sec)
D9003	当前推力	2	0～300		表示当前推力 (单位:%)
D9004	目标位置	4	± 2147483647		表示目标位置 (单位:0.01mm)
D9006	运行数据No.	2	0～63		表示运转完成或运行中的 步数据No
D9008	报警	8	0～255×8个		表示当前报警No。(※1)

※1很多报警同时发生时，最大可保存8个报警代码。

10.2 直接指定位置、速度等信息的动作

读写地址D9100~D9110, 可让电缸直接按照指定的位置、数据运行。依照该运行, 内部继电器(领域D9084)随着运行条件而变化。

【直接指定位置、速度等信息动作的步骤】

- ①将内部继电器 Y30置为“1: 打开串行通信模式”。
- ②在内部继电器Y19(SVON)中写入“1”, 确认内部继电器X49(SVRE)变为“1”。
- ③在内部继电器Y1C(SETUP)中写入“1”, 确认内部继电器X4A(SETON)变为“1”。
- ④将地址从D9102到D9110的数据写入控制器。
- ⑤从地址D9100写入开始运行指示。

地址	运行开始指示	byte	设定内容
D9100	运行数值指定模式	1	1: 按照运行数据(D9102~D9110)开始运行。(运行开始处理后返回0)
	未定义	1	-

地址	假设运行数据	byte	设定范围	单位	数据类型
D9102	动作方法	2	1: 绝对坐标移动 2: 相对坐标移动	- -	整数
D9103	速度	2	1~65535	mm/s	
D9104	位置	4	±2147483647	0.01mm	
D9106	加速度	2	1~65535	mm/s ²	
D9107	减速度	2	1~65535	mm/s ²	
D9108	推压力	2	0~100 (“0”的情况, 变成定位运转)	%	
D9109	阈值(范围值)	2	0~100	%	
D910a	推压速度	2	1~65535	mm/s	
D910b	定位最大推力	2	0~300	%	
D910c	区域输出端 1 (区域 1)	4	±2147483647	0.01mm	
D910e	区域输出端 2 (区域 2)	4	±2147483647	0.01mm	
D9110	定位幅	4	1~2147483647	0.01mm	

 警告
<p>设定范围依各电缸而异。请不要用于设定范围外。</p> <p>关于设定范围, 请参照各电缸操作说明书。</p>

10.3 关于内部FLAG

地址D9084 (X40~X4F) 可确认电机控制器的状态信息。另外, 使用地址D90c1 (Y10~Y1F) 、 D90c2 (Y30~Y3F) 可操作电缸内部继电器。

● 内部继电器 (X接点)

FLAG名称		Read	Write	内容
X40	OUT0	○	×	作为控制器内部处理 (与并行或串行无关), 如左侧所示有输出时ON
X41	OUT1	○	×	
X42	OUT2	○	×	
X43	OUT3	○	×	
X44	OUT4	○	×	
X45	OUT5	○	×	
X46	—	○	×	不可使用
X47				
X48	BUSY	○	×	作为控制器内部处理 (与并行或串行无关), 如左侧所示有输出时ON 但是, 与并行I/O运行不同, ESTOP和ALARM信号变成正逻辑电平。 EMG停止时, 变为ON。 报警发生时, 变为ON。
X49	SVRE	○	×	
X4A	SETON	○	×	
X4B	INP	○	×	
X4C	AREA	○	×	
X4D	WAREA	○	×	
X4E	ESTOP	○	×	
X4F	ALARM	○	×	

● 内部继电器 (Y接点)

FLAG 名称		Read	Write	内容
Y10	IN0	○	○	●Read 时 显示工作在串行运行模式时被指示的状态。 (ON: 1、OFF: 0) ●Write 时 对控制器进行指示。 只在对工作串行运行模式时有效。 (ON: 1、OFF: 0)
Y11	IN1	○	○	
Y12	IN2	○	○	
Y13	IN3	○	○	
Y14	IN4	○	○	
Y15	IN5	○	○	
Y16	—	○	○	不可使用
Y17		○	○	
Y18	HOLD	○	○	●Read 时 显示串行运行模式时被指示的状态。 (ON: 1、OFF: 0) ●Write 时 对控制器进行指示。 只在串行运行模式时有效。 (ON: 1、OFF: 0)
Y19	SVON	○	○	
Y1A	DRIVE	○	○	
Y1B	RESET	○	○	
Y1C	SETUP	○	○	
Y1D	—	○	○	
Y1E		○	○	不可使用
Y1F		○	○	
Y30	输入无效 FLAG (*1)(*2)	○	○	0 : 并行输入运行模式【并行输入输出端子 (CN5) 可正常动作】 1 : 串行输入运行模式【并行输入输出端子 (CN5) 不可用】
Y31 ~Y3F	—	○	×	不可使用 (不能变更)

(*1) 通过 Y30 切换输入运行模式 (并行/串行)。

(*2) 将Y30从0到1指示的时候, 指示前的并行输入状态可以保持。相反, 将Y30从1到0指示的时候, 即可反映并行输入端子的状态。

10.4 步数据

地址	参数名	byte	输入范围	数据类型	内容
D0400 ～ D07FF	步数据 (No.0～63) 32byte×64	2048	—	—	—

(例)地址是步数据 No.0 的情况

地址	参数名	byte	输入范围	数据类型	内容
D0400	动作方法	2	0～255	整数	1:绝对坐标移动(Absolute) 2:相对坐标移动(Relative)
D0401	速度	2	1～65535		设定向目标位置或者推压开始位置的移动速度。 (单位:mm/s)
D0402	位置	4	±21474 83647		设定目标位置或者推压开始位置。(单位:0.01mm)
D0404	加速度	2	1～65535		设定向移动速度加速的加速度。 (单位:mm/s ²)
D0405	减速度	2	1～65535		设定对移动速度减速的减速度。 (单位:mm/s ²)
D0406	推压力	2	0～100		依照设定值选择推压动作或定位动作。 0:定位动作 1～100:推压动作的推压力设定。(单位:1%)
D0407	阈值 (范围值)	2	0～100		推压动作时,发生超出该值的推力, INP 输出变为 ON。 (单位:%)本参数为设定于推压力以下的值。
D0408	推压速度	2	1～65535		(单位:mm/s)推压动作时的移动速度。
D0409	定位最大推力	2	0～300		(单位:%)定位运动时的最大推力。
D040a	区域输出端 1 (区域 1)	4	±21474 83647		AREA 输出是变为 ON 的条件。 区域输出端 1～区域输出端 2 的范围内时, AREA 输出是 ON。
D040c	区域输出端 2 (区域 2)	4	±21474 83647		
D040e	定位幅	4	1～21474 83647		推压运行、定位运行各功能有不同。 ●定位运行: 到位范围(单位:0.01mm) ●推压运行: 推压行程(单位:0.01mm)

11. CRC 计算方法

计算的对象数据为全部信息数据。每 2Byte(16bit)为单位计算一次。

步骤	计算方法
【I】	设置 CRC 寄存器，并给赋值“FFFFH”
【II】	将数据的第一个字节(1 st value、8-bit)的数据与 16 位的低 8 位进行异或运算
【III】	【II】的结果向低位方向移 1 位，最高位补 0。
【IV】	【III】的结果，假如向低位移出的是 1 的时候，需用【III】的值和“A001h”异或运算，假如向低位移出的是 0 的话，则不用。
【V】	上面【III】~【IV】的内容，反复进行直至移位 8 次后结束。
【VI】	【V】的结果与数据的第 2 个字节(2 nd value)的值进行异或运算。
【VII】	【III】~【VI】的内容，反复进行直到最后的数据(3 rd value~最终 value)字节。
【VIII】	【VII】的结果的 2Byte 数据即为 CRC 校验码。



注意

在数据中追加 CRC 时，请注意⑧结果 High Byte「CRC16(H)」和 Low Byte「CRC16(L)」的顺序。

【计算例】

对地址 20，返回校验码通信测试（Function08:测试代码 0000h，测试数据 5AA5h）进行 CRC 校验计算。

Address	20	→	1 st value
Function	08	→	2 nd value
Data	00	→	3 rd value
	00	→	4 th value
	5A	→	5 th value
	A5	→	6 th value (最终 value)

		High Byte「CRC16(H)」								Low Byte「CRC16(L)」								flag	步骤 对应
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
1	FFFFh load	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		【I】
2	1 st value(20h)	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0		【II】
3	No.1 xor No.2	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1		【II】
4	Shift>>1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	【III】
5	No.4 xor a001h	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	0		【IV】
6	Shift>>2	0	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	【V】
7	No.6 xor a001h	1	0	0	1	0	1	1	1	1	1	1	1	1	0	1	0		【V】
8	Shift>>2	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	0	1	【V】
9	No.8 xor a001h	1	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1		【V】
10	shift>>1	0	1	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	【V】

11	No.10 xor a001h	1	1	1	0	0	0	1	0	1	1	1	1	1	1	0		【V】
12	shift>>2	0	0	1	1	1	0	0	0	1	0	1	1	1	1	1	1	【V】
13	No.12 xor a001h	1	0	0	1	1	0	0	0	1	0	1	1	1	1	0		【V】
14	2 nd value(08h)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	【VI】
15	No.13 xor No.15	1	0	0	1	1	0	0	0	1	0	1	1	0	1	1	0	【VI】
16	shift>>2	0	0	1	0	0	1	1	0	0	0	1	0	1	1	0	1	【VI】
17	No.16 xor a001h	1	0	0	0	0	1	1	0	0	0	1	0	1	1	0	0	【VI】
18	shift>>3	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0	1	【VI】
19	No.18 xor a001h	1	0	1	1	0	0	0	0	1	1	0	0	0	1	0	0	【VI】
20	shift>>3	0	0	0	1	0	1	1	0	0	0	0	1	1	0	0	0	【VI】
21	No.20 xor a001h	1	0	1	1	0	1	1	0	0	0	0	1	1	0	0	1	【VI】
22	3 rd value(00h)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	【VII】
23	No.21 xor No.22	1	0	1	1	0	1	1	0	0	0	0	1	1	0	0	1	【VII】
24	shift>>1	0	1	0	1	1	0	1	1	0	0	0	0	1	1	0	0	【VII】
25	No.24 xor a001h	1	1	1	1	1	0	1	1	0	0	0	0	1	1	0	1	【VII】
26	shift>>1	0	1	1	1	1	1	0	1	1	0	0	0	0	1	1	0	【VII】
27	No.26 xor a001h	1	1	0	1	1	1	0	1	1	0	0	0	0	1	1	1	【VII】
28	shift>>1	0	1	1	0	1	1	1	0	1	1	0	0	0	0	1	1	【VII】
29	No.28 xor a001h	1	1	0	0	1	1	1	0	1	1	0	0	0	0	1	0	【VII】
30	shift>>2	0	0	1	1	0	0	1	1	1	0	1	1	0	0	0	0	【VII】
31	No.30 xor a001h	1	0	0	1	0	0	1	1	1	0	1	1	0	0	0	1	【VII】
32	shift>>1	0	1	0	0	1	0	0	1	1	1	0	1	1	0	0	0	【VII】
33	No.32 xor a001h	1	1	1	0	1	0	0	1	1	1	0	1	1	0	0	1	【VII】
34	shift>>1	0	1	1	1	0	1	0	0	1	1	1	0	1	1	0	0	【VII】
35	No.34 xor a001h	1	1	0	1	0	1	0	0	1	1	1	0	1	1	0	1	【VII】
36	shift>>1	0	1	1	0	1	0	1	0	0	1	1	1	0	1	1	0	【VII】
37	No.36 xor a001h	1	1	0	0	1	0	1	0	0	1	1	1	0	1	1	1	【VII】
38	4 th value(00h)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	【VII】
39	No.37 xor No.38	1	1	0	0	1	0	1	0	0	1	1	1	0	1	1	1	【VII】
40	shift>>1	0	1	1	0	0	1	0	1	0	0	1	1	1	0	1	1	【VII】
41	No.40 xor a001h	1	1	0	0	0	1	0	1	0	0	1	1	1	0	1	0	【VII】
42	shift>>2	0	0	1	1	0	0	0	1	0	1	0	0	1	1	1	0	【VII】
43	No.42 xor a001h	1	0	0	1	0	0	0	1	0	1	0	0	1	1	1	1	【VII】
44	shift>>1	0	1	0	0	1	0	0	0	1	0	1	0	0	1	1	1	【VII】
45	No.44 xor a001h	1	1	1	0	1	0	0	0	1	0	1	0	0	1	1	0	【VII】
46	shift >>2	0	0	1	1	1	0	1	0	0	0	1	0	1	0	0	1	【VII】
47	No.46 xor a001h	1	0	0	1	1	0	1	0	0	0	1	0	1	0	0	0	【VII】
48	shift>>2	0	0	1	0	0	1	1	0	1	0	0	0	1	0	1	0	【VII】
49	5 th value(5Ah)	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	【VII】

50	No.48 xor No.49	0	0	1	0	0	1	1	0	1	1	0	1	0	0	0		【Ⅶ】	
51	shift>>5	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1	0	1	【Ⅶ】
52	No.51 xor a001h	1	0	1	0	0	0	0	1	0	0	1	1	0	1	1	1		【Ⅶ】
53	shift>>1	0	1	0	1	0	0	0	0	1	0	0	1	1	0	1	1	1	【Ⅶ】
54	No.53 xor a001h	1	1	1	1	0	0	0	0	1	0	0	1	1	0	1	0		【Ⅶ】
55	shift>>2	0	0	1	1	1	1	0	0	0	0	1	0	0	1	1	0	1	【Ⅶ】
56	No.55 xor a001h	1	0	0	1	1	1	0	0	0	0	1	0	0	1	1	1		【Ⅶ】
57	6 th value(A5h)	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1		【Ⅶ】
58	No.56 xor No.57	1	0	0	1	1	1	0	0	1	0	0	0	0	0	1	0		【Ⅶ】
59	shift>>2	0	0	1	0	0	1	1	1	0	0	1	0	0	0	0	0	1	【Ⅶ】
60	No.59 xor a001h	1	0	0	0	0	1	1	1	0	0	1	0	0	0	0	1		【Ⅶ】
61	shift>>1	0	1	0	0	0	0	1	1	1	0	0	1	0	0	0	0	1	【Ⅶ】
62	No.61 xor a001h	1	1	1	0	0	0	1	1	1	0	0	1	0	0	0	1		【Ⅶ】
63	shift>>1	0	1	1	1	0	0	0	1	1	1	0	0	1	0	0	0	1	【Ⅶ】
64	No.63 xor a001h	1	1	0	1	0	0	0	1	1	1	0	0	1	0	0	1		【Ⅶ】
65	shift>>1	0	1	1	0	1	0	0	0	1	1	1	0	0	1	0	0	1	【Ⅶ】
66	No.65 xor a001h	1	1	0	0	1	0	0	0	1	1	1	0	0	1	0	1		【Ⅶ】
67	shift>>1	0	1	1	0	0	1	0	0	0	1	1	1	0	0	1	0	1	【Ⅶ】
68	No.67 xor a001h	1	1	0	0	0	1	0	0	0	1	1	1	0	0	1	1		【Ⅶ】
69	shift>>1	0	1	1	0	0	0	1	0	0	0	1	1	1	0	0	1	1	【Ⅶ】
70	No.69 xor a001h	1	1	0	0	0	0	1	0	0	0	1	1	1	0	0	0		【Ⅶ】
71	shift>>1	0	1	1	0	0	0	0	1	0	0	0	1	1	1	0	0		【Ⅶ】
	换成16进制	6			1			1			C							【Ⅷ】	

修改记录

SMC株式会社 URL <http://www.smcworld.com>

㉔ 本手册内容如有变更，将不予通知，请知悉。
© 2009 SMC Corporation All Rights Reserved