Melec

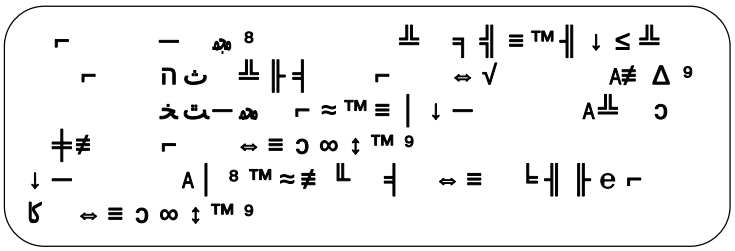
ステッピング＆サーボモータチップコントローラ

C-VX870v1シリーズ

A

(　　)

USER'S MANUAL



MN0482-1

│ ∂ ╘⌐

↓ ─ ₈ A ₉ │ ⁸ ₈ ☻♥ ܦ Ḩ fi ◓ ≶ ɔ ♃ ⁸ ⅔ ╟┘ ◘ ɔ Ẇ ≶ ɔ ♃ ◖ fi ♩ ـ ɔ ꜝ
C-VX870v1 ◦ ˈ ɔ ☼ ₉ ╨ ⇔ ɔ ﷽ ⌐ ئو ⇔ ≡ ™ √ ∞ ɔ √ ╘⌐ ⁸ ث ⌐ ⅝╨⅔™ √ ╡
™ ⌐ ≈ ™ ≡ ⁸ ☻♥ ܦ Ḩ fi ◓ ≶ ɔ ♃ ⁸ №╢ ™ │ ◘ ɔ Ẇ ≶ ɔ ♃ ╨ ئو ⌠ √ ─
╨ ↕ ╡╢ ╨ ⌐ ⁸ ⅔ ╟┘ ث ⌐ ≈ ™ ≡ ⇔ ≡ ™╕ ∆ ⁹
Ẇ ɔ ♪ ◖ fi ♩ ـ ɔ ꜝ ─ ₈ ₉ ≤ ⌐ ⁸ ₈ A ₉ ╨ ɔ ╪≢ ⌐
⇔ ≡ ɔ ∞ ↕ ™ ⁹

↓ ─ ₈ A ₉ │ ⁸ ™ ≈ ≢ ╙ ╡ ⇔ ≡ ╘╢ ╟ ℮ ⌐ ݣ ⇔ ≡ ɔ ∞ ↕ ™ ⁹
⌂ ⅔ ⁸ C-VX870v1 ◦ ˈ ɔ ☼ │ ╨ ≢ ≢ ⅝╢ √ ╘ ⁸ ╨ خ ﬧ ─ ╟ ℮ ⌐ ⇔╕ ∆ ⁹

		1	2	3	4	5	6	7	8	9	10	11	12
C-VX870v1	4	X	Y	Z	A								
C-VX871v1	6	X	Y	Z	A	B	C						
C-VX872v1	8	X1	Y1	Z1	A1	X2	Y2	Z2	A2	−	−		
C-VX873v1	12	X1	Y1	Z1	A1	B1	C1	X2	Y2	Z2	A2	B2	C2
C-VX870Ev1	4	X	Y	Z	A								
C-VX871Ev1	6	X	Y	Z	A	B	C						
C-VX872Ev1	8	X1	Y1	Z1	A1	X2	Y2	Z2	A2	−	−		
C-VX873Ev1	12	X1	Y1	Z1	A1	B1	C1	X2	Y2	Z2	A2	B2	C2

خ ⁸ ≤ ⇔ ≡ X ⌐ ≈ ™ ≡ ─╖ ⇔╕ ∆ ⁹

﷽ ⌐ ∆╢ ڨ ─ ⌐ ≈ ™ ≡

│ ⇔ ™ ≢ ╡ ℮ ↓ ≤ ⅜ ≢ ∆ ⁹
⌠ √ ≢ ئو ↕ ╡ √ ⁸ ۇ ⇔ ⌂ ™ ڨ ╨ ⅝ ↓ ⇔ ⁸ ꞁ ┼─ ╛ ─
⌂ ≥ ─ ╨ ╢ ⅔ ∕ ╡ ⅜№╡ ╕ ∆ ⁹
∕ ─ ╟ ℮ ⌂ ڨ ─ ɔ │ ⁸ ⌂ ╨ ۇ ╘ ⌠ ≡ ™╡ ⌐ ∆╢ ↓ ≤ ⅜≢ ⅝╕ ∆ ⁹
∕ ─ √ ╘ ⁸ ↓ ─ ₈ A ₉ ≢ │ ⌂ ⅜ ۇ ≢ ⅝╢ ⌐ │ ⁸ ڨ ⅜
⇔ ≡ №╡ ╕ ∆ ⁹
∕ ╡ ╠─ │ ⁸ ─ ╟ ℮ ⌂ ◦ fi Ẇ ⁱ ⱱ ɔ ◒ ≤ ◦ ◓ ♫ ⁱ : ɔ ♪ ≢ ⇔ ≡ ™╕ ∆ ⁹

⚠	╡ ™╨ ⌠ √ ⌐ ُ ⁸ ╕ √ │ يخ ╨ ℮ ⅔ ∕ ╡ ─№╢ ڨ ╨ ⇔╕ ∆ ⁹
⚠	╡ ™╨ ⌠ √ ⌐ ⁸ يخ ╨ ℮ ⅔ ∕ ╡ ╛ ⅜ ∆╢ ⅔ ∕ ╡ ⅜№╢ ڨ ╨ ⇔╕ ∆ ⁹

ئو ─ ⌐

Ẽ │ ⁸ ⁸ ⁸ ﺡ ⁸ ⁸ ꞁ ی ⌐ ╥╢
⁸ ⌐ ⅝ ⌂ ⅜ ۇ ↕ ╡╢ ⌂ ≥ ⁸ ⌂ ݣ ⅜ ↕ ╡╢
↑ ⌐ │ ↕ ╡ ≡ ⅔╡ ╕ ∑ ╪ ⁹

Ẽ ﻄﻤ ݣ ث ⌂ ╠┘ ⌐ ⌐ ∆╢ ╡ ™ ⌐ ≈ ™ ≡ │ ⁸ Ẇ ɔ ♪ ◖ fi ♩ ـ ɔ ꜝ ─
╨ ↔ ɔ ∞ ↕ ™ ⁹

Ẽ C-VX870v1 ◦ ˈ ɔ ☼ ─ Ẇ ɔ ♪ ◖ fi ♩ ـ ɔ ꜝ ╨ Windows ≢ ⅔ ئو ™ ⌐ ⌂ ╢ │ ⁸
♦ Ᵽ ▬ ☻ ♪ ꜝ ▬ Ᵽ ╨ ↔ ɔ ∞ ↕ ™ ⁹

Ẽ —

| ⁸ DOS ╕ √ | ⁱ ▪ ⁱ ⁊ ▬ ⊢ OS ⌂ ≥≠ C-VX870v1 ◦ ⁱ ⊃ ☼ ╩ Δ ╢ PORT ث ≤ ◖ ∀ fi ♪ ث ⁸ ⅔ ╟ ┘
≠ ⇔ ≡ ™ ╕ Δ ⁹

1. I/O PORT ث

1-1. PCI Configuration Register

1-2. Local Configuration Register

ה W ⊃ ♪ — ╖ ⇔

1-3. C-VX87xv1 I/O PORT

1-3-1. I/O PORT

1-3-2. MCC PORT

ה DRIVE COMMAND PORT
ה DRIVE DATA1,2 PORT(WRITE)
ה DRIVE DATA1,2 PORT(READ)
ה DRIVE STATUS1--5 PORT

↓ ╗ ╟ — PORT ⌐ ▪ ▬ ☿ ☻ ⇔ ≡ ⁸
♪ ! ▬ Ką ◖ ∀ fi ♪ ⁸ ⅔ ╟ ┘ ○ ▶ fi ⁊
◖ ∀ fi ♪ ╩ ⇔ ╕ Δ ⁹

1-3-3. I/O PORT *

ה PORT(WRITE)
ה PORT(READ)
ה ظ PORT(READ)

↓ ╗ ╟ — PORT ⌐ ▪ ▬ ☿ ☻ ⇔ ≡ ⁸
I/O ╩ ⅄ ⇔ ╕ Δ ⁹

1-3-4. HARD CONFIG PORT

ה HARD CONFIG COMMAND PORT
ה HARD CONFIG DATA1--3 PORT(WRITE)
ה HARD CONFIG DATA1--3 PORT(READ)
ה HARD CONFIG STATUS1--4 PORT(READ)

↓ ╗ ╟ — PORT ⌐ ▪ ▬ ☿ ☻ ⇔ ≡ ⁸
HARD CONFIG ◖ ∀ fi ♪ ╩ ⇔ ╕ Δ ⁹

2. ◖ ∀ fi ♪ ث

— ╜ ⁸ ⅔ ╟ ┘ ╖ ⇔ | ⁸
I/O PORT ┼ — ▪ ▬ ☿ ☻ ⌐ ╟ ∫ ≡ ™ ╕ Δ ⁹
◖ ∀ fi ♪ — ◦ ⊃ ◦ ⁊ ☻ fi ⁊ ♦ ⊃ ⁊ —
╩ ⇔ ╕ Δ ⁹

2-1. ♪ ! ▬ Ką ◖ ∀ fi ♪

ה ظ ث —
ה ♪ ! ▬ Ką R ! ω ⊃ ⁊ —
ה ♪ ! ▬ Ką —
ה ♪ ! ▬ Ką CHANGE —
ה تج ◖ ∀ fi ♪ —
ה ⊏ ⊃ W —
ה ◀ ! ⊃ — ≤ ╖ ⇔
ה ה ♦ ⊃ ⁊ — ╖ ⇔
ה / — ث

2-2. ○ ▶ fi ⁊ ◖ ∀ fi ♪

ה ▪ ♪ ™ ☻ ○ ▶ fi ⁊ —
ה R ⁱ ☻ ○ ▶ fi ⁊ —
ה R ⁱ ☻ تج ○ ▶ fi ⁊ —
ה ○ ▶ fi ⁊ ♦ ⊃ ⁊ — ╖ ⇔
ה ⁱ fi ◓ ○ ▶ fi ⁊ —
ה ○ ▶ fi ♩ ♦ ⊃ ⁊ — ! ℮ ⅷ ה ▬ ⁱ ▪
—

2-3. HARD CONFIG ◖ ∀ fi ♪

ה ☿ fi ⊏ كج —
ה ☻ ⁊ ⊃ ♥ ☻ كج —
ה ☻ ⁊ ⊃ ♩ —
ה — ╖ ⇔

3.

W ⊃ ♪ ◖ fi ♩ ⁼ ⊃ ! — ╩
⇔ ╕ Δ ⁹

3-1. ♪ ! ▬ Ką ث

ה ◖ ∀ fi ♪ ظ
ה ظ ث —
ה ♪ ! ▬ Ką R ! ω ⊃ ⁊
ה ♪ ! ▬ Ką
ה R ⁱ ☻ تج
ה ⊏ ⊃ W
ה ◀ ! ⊃
ה ╖ ⇔
ה / — ث —

3-2. ○ ▶ fi ⁊ ث

ה ◀ fi ◖ ⊃ ♭ R ⁱ ☻ ظ
ה R ⁱ ☻
ה ▪ ♪ ™ ☻ ○ ▶ fi ⁊
ה R ⁱ ☻ ○ ▶ fi ⁊
ה R ⁱ ☻ تج ○ ▶ fi ⁊
ה ◖ fi R ™ ⊃ ⁊
ה ⁱ fi ◓ ○ ▶ fi ⁊
ה ○ ▶ fi ♩ ♦ ⊃ ⁊ — ! ℮ ⅷ ה
▬ ⁱ ▪

3-3. HARD CONFIG ث

ה ☿ fi ⊏ كج
ה ☻ ⁊ ⊃ ♥ ☻ كج
ה ☻ ⁊ ⊃ ♩
ה ╖ ⇔

| № ╕ ╕ ∑ ╪ ⁹ I/O ⁸ | — ,12 6 *

| ∂ ⌐
⌐ Δ ╢ ؤ — ⌐ ≈ ™ ≡
⌐ — ئو

7 7

PAGE

I/O PORT ث

1-1. PCI Configuration Register ---------- 9
1-2. Local Configuration Register ---------- 10
(1) W ↄ ♪ — ╖ ⇔ ---------- 10
1-3. C-VX87xv1 I/O PORT ---------- 11
1-3-1. I/O PORT ---------- 11
(1) MCC07E PORT ---------- 11
(2) I/O PORT ---------- 11
(3) HARD CONFIGURATION PORT ---------- 11
1-3-2. MCC PORT ث ---------- 12
(1) DRIVE COMMAND PORT ---------- 12
(2) DRIVE DATA1,2 PORT(WRITE) ---------- 12
(3) DRIVE DATA1,2 PORT(READ) ---------- 12
(4) DRIVE STATUS1 PORT ---------- 13
(5) DRIVE STATUS2 PORT ---------- 15
(6) DRIVE STATUS3 PORT ---------- 16
(7) DRIVE STATUS4 PORT ---------- 17
(8) DRIVE STATUS5 PORT ---------- 18
1-3-3. I/O PORT ث ---------- 20
(1) PORT(WRITE) ---------- 20
(2) PORT(READ) ---------- 20
(3) ظض PORT(READ) ---------- 20
1-3-4. HARD CONFIGURATION PORT ث ---------- 21
(1) HARD CONFIG COMMAND PORT ---------- 21
(2) HARD CONFIG DATA1,2,3 PORT(WRITE) ---------- 21
(3) HARD CONFIG DATA1,2,3 PORT(READ) ---------- 21
(4) HARD CONFIG STATUS1 PORT ---------- 21
(5) HARD CONFIG STATUS2 PORT ---------- 21
(6) HARD CONFIG STATUS3 PORT ---------- 22
(7) HARD CONFIG STATUS4 PORT ---------- 22

◖ ∇ fi ♪ ث

2-1. ♪ ! ▬ Ką ◖ ∇ fi ♪ ---------- 25
2-1-1. ظض ث — ---------- 25
(1) SPEC INITIALIZE1 ---------- 25
(2) SPEC INITIALIZE2 ---------- 26
(3) SPEC INITIALIZE3 ---------- 28
(4) HARD INITIALIZE1 ---------- 31
(5) HARD INITIALIZE4 ---------- 32
(6) HARD INITIALIZE5 ---------- 33
(7) HARD INITIALIZE6 ---------- 34
(8) HARD INITIALIZE7 ---------- 35
2-1-2. ♪ ! ▬ Ką Ꞧ ! ϖ ↄ ث — ---------- 37
(1) JSPD SET ---------- 37
(2) JOG PULSE SET ---------- 38
(3) FSPD SET ---------- 39
(4) HIGH SPEED SET ---------- 40
(5) LOW SPEED SET ---------- 41
(6) RATE SET ---------- 42
(7) SCAREA SET ---------- 43
(8) DOWN PULSE ADJUST ---------- 44

PAGE

2-1-3. ♪ ! ▬ Ķ ─ ---- 45
(1) JOG ---- 45
(2) JOG ---- 45
(3) SCAN ---- 46
(4) SCAN ---- 46
(5) INC INDEX ---- 47
(6) ABS INDEX ---- 48
2-1-4. ORIGIN ♪ ! ▬ Ķ ─ ≤ ---- 49
(1) ORIGIN SPEC SET ---- 50
(2) ORIGIN SCAN ---- 52
(3) ORIGIN CONSTANT SCAN ---- 52
2-1-5. ش ♪ ! ▬ Ķ ─ ---- 53
(1) CP SPEC SET ---- 54
2-1-6. ♪ ! ▬ Ķ ─ ≤ ---- 55
(1) LONG POSITION SET ---- 59
(2) SHORT POSITION SET ---- 60
(3) MAIN XY STRAIGHT CP ---- 61
(4) SUB STRAIGHT CP ---- 62
(5) MAIN STRAIGHT CP ---- 63
2-1-7. هي ♪ ! ▬ Ķ ─ ≤ ---- 64
(1) CIRCULAR XPOSITION SET ---- 68
(2) CIRCULAR YPOSITION SET ---- 69
(3) CIRCULAR PULSE SET ---- 70
(4) MAIN XY CIRCULAR CP ---- 71
(5) SUB CIRCULAR CP ---- 72
(6) MAIN CIRCULAR CP ---- 73
2-1-8. UP/DOWN/CONST ♪ ! ▬ Ķ CHANGE ─ ≤ ---- 75
(1) UDC SPEC SET ---- 75
(2) UP DRIVE ---- 76
(3) DOWN DRIVE ---- 76
(4) CONST DRIVE ---- 76
2-1-9. SPEED CHANGE ─ ≤ ---- 77
(1) SPEED CHANGE SPEC SET ---- 77
(2) SPEED CHANGE ---- 78
2-1-10. RATE CHANGE ─ ≤ ---- 79
(1) RATE CHANGE ---- 79
2-1-11. INDEX CHANGE ─ ≤ ---- 80
(1) INDEX CHANGE SPEC SET ---- 80
(2) INC INDEX CHANGE ---- 81
(3) ABS INDEX CHANGE ---- 82
(4) PLS INDEX CHANGE ---- 83
2-1-12. تج ◖ ℣ fi ♪ ─ ---- 84
(1) SLOW STOP ---- 84
(2) FAST STOP ---- 84
2-1-13. ◘ ↄ W ─ ---- 84
(1) SIGNAL OUT ---- 85
2-1-14. ◄ ! ↄ ─ ≤ ╗ ⇔ ---- 86
(1) ERROR STATUS MASK ---- 86
(2) ERRINT STATUS MASK ---- 87
(3) ERROR STATUS CLR ---- 88
(4) ERROR STATUS READ ---- 89
2-1-15. ╡ ╗ ─ ≤ ╗ ⇔ ---- 91
(1) INT FACTOR MASK ---- 91
(2) INT FACTOR CLR ---- 92
(3) INT FACTOR READ ---- 93
2-1-16. ה ♦ ↄ ─ ╗ ⇔ ---- 94
(1) MCC SPEED READ ---- 94
(2) MCC SET DATA READ ---- 95

PAGE

2-1-17. ／ー ---- 97
(1) NO OPERATION ---- 97
(2) MCC CHIP RESET ---- 97
2-2. ○ ▶ fi ◖ fi ♪ ---- 98
2-2-1. ▪ ♪ ☻ ○ ▶ fi ー ---- 98
(1) ADDRESS COUNTER INITIALIZE1 ---- 98
(2) ADDRESS COUNTER INITIALIZE2 ---- 101
(3) ADDRESS COUNTER PRESET ---- 103
(4) ADDRESS COUNTER MAX COUNT SET ---- 104
(5) ADRINT COMPARE REGISTER1,2,3 SET ---- 105
(6) ADRINT COMP1 ADD DATA SET ---- 106
2-2-2. R ☻ ○ ▶ fi ー ---- 107
(1) PULSE COUNTER INITIALIZE1 ---- 107
(2) PULSE COUNTER INITIALIZE2 ---- 110
(3) PULSE COUNTER PRESET ---- 112
(4) PULSE COUNTER MAX COUNT SET ---- 113
(5) CNTINT COMPARE REGISTER1,2,3 SET ---- 114
(6) CNTINT COMP1 ADD DATA SET ---- 115
2-2-3. R ☻ ○ ▶ fi ー ---- 116
(1) DFL COUNTER INITIALIZE1 ---- 116
(2) DFL COUNTER INITIALIZE2 ---- 119
(3) DFL COUNTER INITIALIZE3 ---- 121
(4) DFL COUNTER PRESET ---- 122
(5) DFLINT COMPARE REGISTER1,2,3 SET ---- 123
(6) DFLINT COMP1 ADD DATA SET ---- 124
2-2-4. ○ ▶ fi ー ▪ ー ---- 125
(1) COUNT LATCH SPEC SET ---- 125
2-2-5. ○ ▶ fi ♦ ー ⇔ ---- 127
(1) ADDRESS COUNTER READ ---- 127
(2) PULSE COUNTER READ ---- 127
(3) DFL COUNTER READ ---- 127
2-2-6. ○ ▶ fi ♦ ー ♦ ー ⇔ ---- 128
(1) ADDRESS LATCH DATA READ ---- 128
(2) PULSE LATCH DATA READ ---- 128
(3) DFL LATCH DATA READ ---- 128
2-3. HARD CONFIG ◖ fi ♪ ---- 129
2-3-1. ー ---- 129
(1) MAN MASK ---- 129
(2) SENSOR SIGNAL SELECT ---- 130
(3) SIGNAL OUT SELECT ---- 132
(4) SIGNAL OUT TIMER SET ---- 134
(5) SIGNAL OUT LATCH STATUS CLR ---- 135
2-3-2. ☻ ー ≤ ---- 136
(1) PAUSE SET SPEC ---- 136
(2) PAUSE CLR SPEC ---- 138
(3) PAUSE ---- 140
2-3-3. ♦ ー ⇔ ---- 141
(1) HARD CONFIG SET DATA READ ---- 141
2-3-4. ／ー ---- 141
(1) HARD CONFIG RESET ---- 141

3-1. ♪ ! ▬Ką ث ---------- 142
3-1-1. ◖ ∀ fi ♪ ﻭ ---------- 142
3-1-2. ﻁﻄﻲ ث ---------- 144
(1) Ꞧ i ☻ ث ---------- 144
(2) ◘ ↄ W ---------- 145
(3) ﻁﻄﻲ ﻛﺦ — ♦ ☺ ﺍ i ɦ ▫ i ﺍ ---------- 146
(4) ﻁﻄﻲ ﻛﺦ — ╡ ﻉ ---------- 146
3-1-3. ♪ ! ▬Ką Ꞧ ! ϖ ↄ ﺍ ---------- 147
(1) 1 Ꞧ i ☻ ---------- 147
(2) JOG Ꞧ ! ϖ ↄ ﺍ ---------- 148
(3) Ꞧ ! ϖ ↄ ﺍ ---------- 149
(4) ♪ ! ▬Ką ---------- 151
(5) S ♪ ! ▬Ką ---------- 152
(6) ／—ﺙ— ♪ ! ▬Ką ---------- 154
3-1-4. ♪ ! ▬Ką ---------- 156
(1) JOG ♪ ! ▬Ką ---------- 156
(2) SCAN ♪ ! ▬Ką ---------- 156
(3) INDEX ♪ ! ▬Ką ---------- 157
3-1-5. ORIGIN ♪ ! ▬Ką ---------- 158
(1) ORIGIN ♪ ! ▬Ką ---------- 158
3-1-6. ♪ ! ▬Ką ---------- 159
(1) 2 ♪ ! ▬Ką ---------- 159
(2) ش ♪ ! ▬Ką ---------- 160
(3) ♪ ! ▬Ką ---------- 162
(4) ﻫﻲ ♪ ! ▬Ką ---------- 163
(5) ﻱ ---------- 166
3-1-7. ♪ ! ▬Ką CHANGE ---------- 167
(1) UP/DOWN/CONST ♪ ! ▬Ką CHANGE ---------- 167
(2) SPEED CHANGE ---------- 169
(3) RATE CHANGE ---------- 170
(4) INDEX CHANGE ---------- 171
3-1-8. MANUAL ♪ ! ▬Ką ---------- 174
3-1-9. Ꞧ i ☻ ﺗﺞ ---------- 176
(1) ﺗﺞ ---------- 176
(2) ﺗﺞ ---------- 176
(3) LIMIT ﺗﺞ ---------- 176
3-1-10. ◀ ! ↄ ---------- 177
3-1-11. ╡ ╥ ---------- 179
3-1-12. ╥ ⇔ ---------- 180
(1) ☻ ♥ ↄ ﺍ ☻ ╥ ⇔ ---------- 180
(2) ♦ ↄ ﺍ ╥ ⇔ ---------- 180
(3) ﺟﺞ — ♪ ! ▬Ką ╥ ⇔ ---------- 180
(4) ◀ ! ↄ ☻ ♥ ↄ ﺍ ☻ ╥ ⇔ ---------- 180
(5) ╡ ╥ ╥ ⇔ ---------- 180
(6) ○ ▶ fi ♩ ♦ ↄ ﺍ ╥ ⇔ ---------- 180
(7) ○ ▶ fi ♩ ♦ ↄ ﺍ — ! ﺑﺑﺑ ♦ ↄ ﺍ ╥ ⇔ ---------- 180
3-2. ○ ▶ fi ﺍ ث ---------- 181
3-2-1. ◀ fi ◖ ↄ ﺏ Ꞧ i ☻ ﻁﻄﻲ ---------- 181
3-2-2. Ꞧ i ☻ ---------- 182
3-2-3. ▪ ♪ ☻ ○ ▶ fi ﺍ ---------- 184
3-2-4. Ꞧ i ☻ ○ ▶ fi ﺍ ---------- 185
3-2-5. Ꞧ i ☻ ﺗﺞ ○ ▶ fi ﺍ ---------- 186
3-2-6. ◖ fi Ꞧ ↄ ﺍ ---------- 188
3-2-7. i fi ◓ ○ ▶ fi ﺍ ---------- 190
3-2-8. ○ ▶ fi ♩ ♦ ↄ ﺍ — ! ﺑﺑﺑ ◓ ﻫ i ▪ ---------- 191

PAGE

3-3. HARD CONFIG ث ---------- 192
3-3-1. ظص ث ---------- 192
(1) ☿ fi ◘ ---------- 192
(2) ☻ ♥ ↄ ♃ ☻ ---------- 193
3-3-2. ☻ ♃ ↄ ♩ ---------- 194
3-3-3. ╗ ⇔ ---------- 196
(1) ☻ ♥ ↄ ♃ ☻ ╗ ⇔ ---------- 196
(2) ♦ ↄ ♃ ╗ ⇔ ---------- 196

⛸
4-1. ♃ ▬ ⊢ fi ◓ ---------- 197
(1) PCI ₱ ☻ ---------- 197
(2) ⁱ ☿ ₽ ♩ ---------- 198
(3) ◖ ∀ fi ♪ ─ ---------- 198
(4) ♪ ! ▬ Ķ ─ ---------- 199
(5) CHANGE ♪ ! ▬ Ķ ─ ---------- 200
(6) تج ---------- 201
(7) ◘ ↄ W̌ ---------- 202
4-2. پ ث ---------- 203
(1) ---------- 203
(2) ♪ ! ▬ Ķ Ꞧ ! ɷ ↄ ♃ ---------- 207
4-3. ♪ ! ▬ Ķ ◖ ∀ fi ♪ پ ---------- 208
(1) ◖ ∀ fi ♪ ---------- 208
(2) ◖ ∀ fi ♪ ---------- 210
4-4. HARD CONFIG ◖ ∀ fi ♪ پ ---------- 212
4-5. W̌ ↄ ♪ ث پ ---------- 213

♩ ! Ķ ⁱ ₀ ⁺ ↄ ♥ ◘ fi ◓ ---------- 215

≠ ↕ ╬ √ ذ ⌂

１．I/O PORT 仕様

1-1. PCI Configuration Register

C-VX870v1 シリーズを DOS や、リアルタイム OS でお使いになる場合は、PCI Configuration Register で C-VX870v1 シリーズに割り当てられた I/O 空間のベースアドレスを取得してください。

31		16 15	0	Offset
Device ID ***1**		Vendor ID(**H'152E**)		H'00
Status		Command		H'04
Base Class(H'0E)	Sub Class (H'80)	Prog.I/F (H'00)	Revision ID *2	H'08
BIST	Header Type(H'00)	Latency Timer	Cache Line Size	H'0C
Base Address Register0： **未使用**				H'10
Base Address Register1： **Local Configuration Register Base Address**				H'14
Base Address Register2： **C-VX87xv1 I/O PORT Base Address**				H'18
Reserved				H'1C
				H'20
				H'24
Cardbus CIS Pointer				H'28
Subsystem ID ***1**		Subsystem Vendor ID (**H'152E**)		H'2C
Expansion ROM Base Address：未使用				H'30
Reserved				H'34
Reserved				H'38
Max_Lat	Min_Gnt	Interrupt pin(H'01)	Interrupt Line	H'3C

***1** Device ID, Subsystem ID

製品	Device ID, Subsystem ID
C-VX870v1	H'1100
C-VX871v1	H'1110
C-VX872v1	H'1120
C-VX873v1	H'1130
C-VX870Ev1	H'1140
C-VX871Ev1	H'1150
C-VX872Ev1	H'11D0
C-VX873Ev1	H'11E0

*2 Revision ID

製品	Revision ID
C-VX870v1,C-VX871v1	H'01
C-VX872v1,C-VX873v1	
C-VX870Ev1,C-VX871Ev1	
C-VX872Ev1,C-VX873Ev1	

● Local Configuration Register Base Address (Base Address1)
ボード番号読み出しビットを含む内部動作定義用のレジスタ群に割り当てられた I/O 空間のベースアドレスです。
Local Configuration Register の I/O 空間は 128 BYTE です。

● C-VX87xv1 I/O PORT Base Address (Base Address2)
各軸 MCC07E, 汎用 I/O, HARD CONFIGURATION 部に割り当てられた I/O 空間のベースアドレスです。
C-VX87xv1 I/O PORT の I/O 空間は 256 BYTE です。

1-2. Local Configuration Register

Local Configuration Register [illegible]

[illegible] Local Configuration Register [illegible]
Local Configuration Register [illegible] READ [illegible]
WRITE [illegible]

> ⚠ WRITE [illegible]
> Local Configuration Register [illegible] WRITE [illegible]

(1) [illegible]

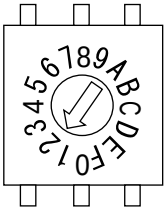
[illegible] Local Configuration Register [illegible] (S1) [illegible]
Offset Address:54h [illegible]

Local Configuration Register

31							16 | 15		0	Offset
UNUSED	D^{23}	UNUSED	D^{20}	UNUSED	D^{17}	UNUSED	D^{14}	UNUSED		54h

[illegible] 2^0
[illegible] 2^1
[illegible] 2^2
[illegible] 2^3

[illegible]



S1

Local Configuration Register

31							16 | 15		0	Offset
UNUSED	0	UNUSED	0	UNUSED	1	UNUSED	0	UNUSED		54h

[illegible] $2^0 = 0$
[illegible] $2^1 = 1$
[illegible] $2^2 = 0$
[illegible] $2^3 = 0$

2h

* C-VX870v1 [illegible]

1-3. C-VX87xv1 I/O PORT

1-3-1. I/O PORT

C-VX870v1 ◦ ⁱ ↄ ☼ — W↗ ↄ ♪ ◖ fi ♩ ˷ ↄ ! — I/O PORT ⇔ ╕ Δ [9]
MCC07E PORT │ ⇔ ≡ ⌐№╡ ╕ Δ [9]
╕ √ [8] I/O PORT ⅔ ╟┘ HARD CONFIGURATION PORT │ [8] W↗ ↄ ♪ ت ⌐ ≈ — — PORT ⅜№╡ ╕ Δ [9]

Offset	4	6	8	12
	C-VX870v1 C-VX870Ev1	C-VX871v1 C-VX871Ev1	C-VX872v1 C-VX872Ev1	C-VX873v1 C-VX873Ev1
H'00 H'0E	X MCC07E	X MCC07E	X1 MCC07E	X1 MCC07E
H'10 H'1E	Y MCC07E	Y MCC07E	Y1 MCC07E	Y1 MCC07E
H'20 H'2E	Z MCC07E	Z MCC07E	Z1 MCC07E	Z1 MCC07E
H'30 H'3E	A MCC07E	A MCC07E	A1 MCC07E	A1 MCC07E
H'40 H'4E	ئو	B MCC07E	ئو	B1 MCC07E
H'50 H'5E	ئو	C MCC07E	ئو	C1 MCC07E
H'60 H'6E	ئو	ئو	X2 MCC07E	X2 MCC07E
H'70 H'7E	ئو	ئو	Y2 MCC07E	Y2 MCC07E
H'80 H'8E	ئو	ئو	Z2 MCC07E	Z2 MCC07E
H'90 H'9E	ئو	ئو	A2 MCC07E	A2 MCC07E
H'A0 H'AE	ئو	ئو	ئو	B2 MCC07E
H'B0 H'BE	ئو	ئو	ئو	C2 MCC07E
H'C0 H'CE	ئو	ئو	ئو	ئو
H'D0 H'DE	ئو	ئو	ئو	ئو
H'E0 H'EE	I/O	ئو	I/O	ئو
H'F0 H'FE	HARD CONFIGURATION	HARD CONFIGURATION	HARD CONFIGURATION	HARD CONFIGURATION

(1) MCC07E PORT

Offset	WRITE PORT	READ PORT
H'x0	DRIVE DATA1	DRIVE DATA1
H'x2	DRIVE DATA2	DRIVE DATA2
H'x4	ئو	ئو
H'x6	DRIVE COMMAND	DRIVE STATUS1
H'x8	ئو	DRIVE STATUS2
H'xA	ئو	DRIVE STATUS3
H'xC	ئو	DRIVE STATUS4
H'xE	ئو	DRIVE STATUS5

(2) I/O PORT

Offset	C-VX870v1 C-VX870Ev1		C-VX872v1 C-VX872Ev1	
	WRITE PORT	READ PORT	WRITE PORT	READ PORT
H'E0			1	1
H'E2	ئو	طص	ئو	1 طص
H'E4	ئو	ئو	ئو	ئو
H'E6	ئو	ئو	ئو	ئو
H'E8	ئو	ئو	2	2
H'EA	ئو	ئو	ئو	2 طص
H'EC	ئو	ئو	ئو	ئو
H'EE	ئو	ئو	ئو	ئو

ה I/O PORT │ [8] C-VX871v1, C-VX871Ev1, C-VX873v1 ⌐ │№╡ ╕ ∑ ╪ [9]

(3) HARD CONFIGURATION PORT

Offset	WRITE PORT	READ PORT
H'F0	**HARD CONFIG DATA1**	**HARD CONFIG DATA1**
H'F2	**HARD CONFIG DATA2**	**HARD CONFIG DATA2**
H'F4	**HARD CONFIG DATA3**	**HARD CONFIG DATA3**
H'F6	**HARD CONFIG COMMAND**	**HARD CONFIG STATUS1**
H'F8	ئو	**HARD CONFIG STATUS2**
H'FA	ئو	**HARD CONFIG STATUS3**
H'FC	ئو	**HARD CONFIG STATUS4**
H'FE	ئو	ئو

1-3-2. MCC PORT ث

(1) DRIVE COMMAND PORT

DRIVE COMMAND ⊥ ⅝ ╗ PORT ≠ Δ ⁹ ↓ ─ PORT ⌐ DRIVE COMMAND ⊥ ⅝ ╗ ≤ ⁸ ♦ ⊃ ㉿ ─
╕ √ │ ♪ ! ▬Ķ ─ ⊥ ™ ╕ Δ ⁹
⅝ ╗ DRIVE COMMAND │ ⠿ ₀ 8 ⅊ ⁸ ♪ ─ ╖ ≠ Δ ⁹ ث ₀ 8 ⅊ ⁸ ♪ │ ⇔ ╕ Δ ⁹
DRIVE COMMAND ⌐ │ ⁸ ◖ ∀ fi ♪ H'00 H'7F ≤ ◖ ∀ fi ♪ H'80 H'FF ⅜№╡ ╕ Δ ⁹

ù ◖ ∀ fi ♪ ─ ⅝ ╗
◖ ∀ fi ♪ │ DRIVE STATUS1 PORT ─ BUSY=0 ⁸ ERROR=0 ─ ⌐ ⅝ ╗ ↓ ≤ ⅜≠ ⅝╕ Δ ⁹
◖ ∀ fi ♪ ─ ₀ ⁸ ♪ ! ▬Ķ ─ 2 ◖ ∀ fi ♪ ⌐ ≈ ™ ≡ │ ⁸ ج ─ BUSY=0 ⁸ ERROR=0 ─ ⌐
⅝ ╗ ↓ ≤ ⅜≠ ⅝╕ Δ ⁹
≠ ─ │ ⠿ خ ─ ╡≠ Δ ⁹

C-VX870v1, C-VX870Ev1

MCC07E	X
	Y
MCC07E	Z
	A

C-VX871v1, C-VX871Ev1

MCC07E	X
	Y
MCC07E	Z
	A
MCC07E	B
	C

C-VX872v1, C-VX872Ev1

MCC07E	X1
	Y1
MCC07E	Z1
	A1
MCC07E	X2
	Y2
MCC07E	Z2
	A2

C-VX873v1, C-VX873Ev1

MCC07E	X1
	Y1
MCC07E	Z1
	A1
MCC07E	B1
	C1
MCC07E	X2
	Y2
MCC07E	Z2
	A2
MCC07E	B2
	C2

◖ ∀ fi ♪ │ ◖ ∀ fi ♪ ⌠ ≠ ◖ ∀ fi ♪ ⊥ ⌠ Δ ╟ ↓ ≤ ⅜≠ ⅝╕ Δ ⁹
◖ ∀ fi ♪ ⊥ ⌠ Δ ╟ ⁸ DRIVE STATUS1 PORT ─ COMREG FL=0 ⁸ ERROR=0 ─ ⌐ ◖ ∀ fi ♪ ⊥
⅝ ╗ ↓ ≤ ⅜≠ ⅝╕ Δ ⁹
2 ◖ ∀ fi ♪ ⊥ ⌠ Δ ╟ ⁸ ج ─ COMREG FL=0 ⁸ ERROR=0 ─ ⌐ ⅝ ╗ ↓ ≤ ⅜≠ ⅝╕ Δ ⁹

ù ◖ ∀ fi ♪ ─ ⅝ ╗
◖ ∀ fi ♪ ─ ⅝ ╗ │ ♪ ! ▬Ķ CHANGE ◖ ∀ fi ♪ ⊥ ⅝ ⁸ ≠ Δ ⁹
◖ ∀ fi ♪ ─ ♪ ! ▬Ķ CHANGE ◖ ∀ fi ♪ │ ⁸ DRIVE STATUS5 PORT ─ ḥ ! ◓ ⊥ ⇔ ≡ ⅝ ╗ ╕ Δ ⁹
ה ☻ Ḥ ⊃ ♪ ─ ♪ ! ▬Ķ CHANGE ◖ ∀ fi ♪ │ SPEED CSET=0 ─ ⌐ ⅝ ╗ ↓ ≤ ⅜≠ ⅝╕ Δ ⁹
ה ☻ Ḥ ⊃ ♪ ─ ♪ ! ▬Ķ CHANGE ◖ ∀ fi ♪ │ SPEED CBUSY=0 ─ ⌐ ⅝ ╗ ↓ ≤ ⅜≠ ⅝╕ Δ ⁹
ה INDEX CHANGE ◖ ∀ fi ♪ │ INDEX CSET=0 ─ ⌐ ⅝ ╗ ↓ ≤ ⅜≠ ⅝╕ Δ ⁹
ה INDEX CHANGE ◖ ∀ fi ♪ │ INDEX CBUSY=0 ─ ⌐ ⅝ ╗ ↓ ≤ ⅜≠ ⅝╕ Δ ⁹

(2) DRIVE DATA1,2 PORT(WRITE)

DRIVE COMMAND ─ ♦ ⊃ ㉿ ⁸ ╕ √ │ ⇔ √ ♪ ! ▬Ķ ─ ⅊ ♦ ⊃ ㉿ ⊥ ⅝ ╗ PORT ≠ Δ ⁹
↓ ─ PORT ┼ ─ ⅝ ╗ │ ≠ Δ ⁹

(3) DRIVE DATA1,2 PORT(READ)

♦ ⊃ ㉿ ⊥ ╗ Δ PORT ≠ Δ ⁹ ╗ ⇔ │ ≠ Δ ⁹
READ ◖ ∀ fi ♪ ⊥ DRIVE COMMAND PORT ⌐ ⅝ ╗ ≤ ⁸ ♦ ⊃ ㉿ ⊥ DRIVE DATA1,2 PORT ⌐
⅊ ⇔ ╕ Δ ⁹
DRIVE DATA1,2 PORT ⌐ ⅊ ⇔ √ ♦ ⊃ ㉿ │ ─ READ ◖ ∀ fi ♪ ─ ⅝ ╗ ╕ ≠ ⅊ ⇔ ╕ Δ ⁹
⇔ ™ ♦ ⊃ ㉿ ⊥ ╗ Δ │ ⁸ READ ◖ ∀ fi ♪ ⊥ ⇔ ≡ ⅛ ╟ ╗ ⇔ ╕ Δ ⁹

(4) DRIVE STATUS1 PORT

♪ ! ▬K ◖ fi ♩ ⌄ ⌐ i — — ╩ Δ ╢ PORT ≠ Δ 9 ╥ ⇔ │ ≠ Δ 9

D15	D14	D13	D12	D11	D10	D9	D8
COMREG FL	COMREG EP	PAUSE	MAN	EXT PULSE	CONST	DOWN	UP

D7	D6	D5	D4	D3	D2	D1	D0
FSEND	SSEND	LSEND	ERROR	DRVEND	DRIVE	STBY	BUSY

D0 : BUSY

1 ◖ ∀ fi ♪ ⌐ 8 ╕ √ │ ♪ ! ▬K ⌐ ╩ ⇔ ╕ Δ 9

0 ◖ ∀ fi ♪ ⌐ ⌐ — ╩ ⇔ ╕ Δ 9

ה 2 ◖ ∀ fi ♪ ⌐ │ 2 ≤ ╨ BUSY=1 ≤ ⌂ ╡ ╕ Δ 9

ה MAN=1 ╕ √ │ EXT PULSE=1 — │ BUSY=1 ≤ ⌂ ╡ ╕ Δ 9

D1 : STBY

1 R i ☻ — ⌐ ⅜ ⌐ ⇔ √ ╩ ⇔ ╕ Δ 9

ה SPEC INITIALIZE3 ◖ ∀ fi ♪ — STBY ⌐ — ≠ STBY ╩ ▪ i ▪ ⇔ R i ☻ ╩ ⇔ ╕ Δ 9

ה STBY=1 ⌐ ⌐ ⌐ ⌐ ╩ ⌐ ⇔ √ 8 R i ☻ │ ╥ ∏ ♪ ! ▬K ╩ ⌐ ⇔ 8 STBY ╩ ▪ i ▪ ⇔ ╕ Δ 9

D2 : DRIVE

1 R i ☻ ⌐ — ╩ ⇔ ╕ Δ 9

0 R i ☻ ⌐ ⌐ — ╩ ⇔ ╕ Δ 9

D3 : DRVEND

1 R i ☻ ╩ ق e ◖ ∀ fi ♪ ⅜ ⌐ ⇔ √ ╩ ⇔ ╕ Δ 9

ה ⌐ ⌐ — ⌐ — ⌐ ! ◀ ⌐ ⌐ ╟ 8 R i ☻ ╩ ╥ ∏ ♪ ! ▬K ╩ ⌐ ⇔ √ ≠ ╨

DRVEND=1 ≤ ⇔ ╕ Δ 9

ה — ◖ ∀ fi ♪ — ≠ ▪ i ▪ ⇔ ╕ Δ 9

ה MANUAL SCAN ♪ ! ▬K — ≠ ╨ ▪ i ▪ ⇔ ╕ Δ 9

D4 : ERROR

1 ◀ ! ⌐ ⅜ ⇔ √ ╩ ⇔ ╕ Δ 9

ה ERROR │ ERROR STATUS — OR( ) ≠ Δ 9

Δ ╢ ERROR STATUS │ ERROR STATUS MASK ◖ ∀ fi ♪ ≠ ⌐ ⌐ ∀ ☻ ▪ Δ ╢ ↓ ≤ ⅜≠ ⅝ ╕ Δ 9

ERROR STATUS │ ERROR STATUS READ ◖ ∀ fi ♪ ≠ ╥ Δ ↓ ≤ ⅜≠ ⅝ ╕ Δ 9

ה ERROR=1 — │ COMREG FL=1 8 COMREG=1 ≤ ⌂ ╡ ◖ ∀ fi ♪ — ⅝ ╥ ⅜ ≤ ⌂ ╡ ╕ Δ 9

ה ERROR │ ERROR STATUS CLR ◖ ∀ fi ♪ ≠ ▪ i ▪ ⇔ ╕ Δ 9 م ⇔ ERROR STATUS — ⌐ ⌐ ⅜ ⌐ ⇔ ≡

™ ╢ │ ▪ i ▪ ↕ ╢ ╕ ∑ ╪ 9

ה ♪ ! ▬K ≠ ◀ ! ⌐ ⅜ ⇔ √ 8 ◀ ! ⌐ ⅜ ERROR=1 ≤ ⌂ ╡ ╕ Δ 9

D5 : LSEND

1 LIMIT ⌐ ⌐ ╩ ⇔ √ ╩ ⇔ ╕ Δ 9

ה — R i ☻ ╩ ق e ◖ ∀ fi ♪ — ≠ ▪ i ▪ ⇔ ╕ Δ 9

ה MANUAL SCAN DRIVE — ≠ ╨ ▪ i ▪ ⇔ ╕ Δ 9

ה 2 ♪ ! ▬K ≠ LIMIT ⌐ ⌐ ╩ ⇔ √ 8 ⌐ ⅜ LSEND=1 ≤ ⌂ ╡ ╕ Δ 9

D6 : SSEND

1 ⌐ ⌐ ╩ ⇔ √ ╩ ⇔ ╕ Δ 9

ה — R i ☻ ╩ ق e ◖ ∀ fi ♪ — ≠ ▪ i ▪ ⇔ ╕ Δ 9

ה MANUAL SCAN DRIVE — ≠ ╨ ▪ i ▪ ⇔ ╕ Δ 9

ה 2 ♪ ! ▬K ≠ ⌐ ⌐ ╩ ⇔ √ 8 ⌐ ⅜ SSEND=1 ≤ ⌂ ╡ ╕ Δ 9

D7 : FSEND

D8 : UP

D9 : DOWN

D10: CONST

D11: EXT PULSE

D12: MAN

D13: PAUSE

D14: COMREG EP

D15: COMREG FL

COMREG FL	COMREG EP	
0	0	(EMPTY)
0	1	
1	0	(FULL)
1	1	RESET ERROR=1

(5) DRIVE STATUS2 PORT

طمي كخ — ╩ Δ ╢ PORT ≠ Δ ⁹ ╖ ⇔ │ ≠ Δ ⁹

D15	D14	D13	D12	D11	D10	D9	D8
DEND BUSY	DALM	DEND/PO	DRST	0	NORG	ZORG	ORG

D7	D6	D5	D4	D3	D2	D1	D0
0	0	ORG SIGNAL	PULSE MASK	CCWLM	CWLM	FSSTOP	0

D1 : FSSTOP

1 FSSTOP طمي كخ ⁸ ╕ √ │ $\overline{\text{FSSTOP}}$ طمي كخ ⅜ ▪ ◒ ♥ ▫ Ϥ ϩ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D2 : CWLM

1 CWLM طمي كخ ⅜ ▪ ◒ ♥ ▫ Ϥ ϩ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D3 : CCWLM

1 CCWLM طمي كخ ⅜ ▪ ◒ ♥ ▫ Ϥ ϩ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D4 : PULSE MASK

1 PULSE OUTPUT MASK = 1 ⌐ ⇔ ≡ ™╢

ה SPEC INITIALIZE1 ◖ ∀ fi ♪ ≠ ⁸ PULSE OUTPUT MASK = 1 ⌐ ⇔ ≡ ™╢ ≠ Δ ⁹

D5 : ORG SIGNAL

1 ORG كخ ⅜ ▪ ◒ ♥ ▫ Ϥ ϩ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ה ORIGIN SPEC SET ◖ ∀ fi ♪ — ORG SIGNAL TYPE ≠ ⇔ ≡ ™╢ كخ — ≠ Δ ⁹

D8 : ORG

1 $\overline{\text{ORG}}$ طمي كخ ⅜ ▪ ◒ ♥ ▫ Ϥ ϩ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D9 : ZORG

1 ÑZORG طمي كخ ⅜ ▪ ◒ ♥ ▫ Ϥ ϩ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D10: NORG

1 $\overline{\text{NORG}}$ طمي كخ ⅜ ▪ ◒ ♥ ▫ Ϥ ϩ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D12: DRST

1 $\overline{\text{DRST}}$ كخ ⅜ ▪ ◒ ♥ ▫ Ϥ ϩ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D13: DEND/PO

1 $\overline{\text{DEND/PO}}$ طمي كخ ⅜ ▪ ◒ ♥ ▫ Ϥ ϩ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D14: DALM

1 DALM طمي كخ ⅜ ▪ ◒ ♥ ▫ Ϥ ϩ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ה SPEC INITIALIZE3 ◖ ∀ fi ♪ ≠ DALM طمي كخ ╩ تجـ ⌐ Δ ╢ ↓ ≤ ⅜≠ ⅝ ╕ Δ ⁹
DALM كخ ⌐ │ ⁸ $\overline{\text{INn0}}$(Xn), $\overline{\text{INn1}}$(Yn), $\overline{\text{INn2}}$(Zn), $\overline{\text{INn3}}$(An)⅜ ╡ ↑ ╟╢ ≡ ™╕ Δ ⁹
⌂ ⅔ ⁸ DALM │ ⁸ طمي $\overline{\text{INnx}}$ كخ ╩ √ ⌂ ™ ≠ │ ⇔ ╕ ∑ ╪ ⁹
ה $\overline{\text{INnx}}$ كخ ╩ DALM ≤ ⇔ √ │ ⁸ طمي كخ ╩ ╡ ╩ ╡ ↓ ≤ ⅜≠ ⅝ ╕ Δ ⁹

D15: DEND BUSY

1 ⌈R ⁱ ☻ ⁸ $\overline{\text{DEND}}$/$\overline{\text{PO}}$ طمي كخ — ▪ ◒ ♥ ▫ Ϥ ϩ ⁱ └ — ╩ ⇔ ╕ Δ ⁹

ה SERVO SPEC SET ◖ ∀ fi ♪ ≠ $\overline{\text{DEND}}$/$\overline{\text{PO}}$ طمي كخ ╩<◘ ⊃ W̄ >⌐ ⇔ ≡ ™╢ ⌐ ≠ Δ ⁹

(6) DRIVE STATUS3 PORT

[illegible] PORT [illegible]

D15	D14	D13	D12	D11	D10	D9	D8
(ث)	(ث)	(ث)	(ث)	0	0	FSSTOP	(ث)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	OUT B	OUT A	0	0	0	INT

D0 : INT

1 [illegible]

[illegible] INT [illegible] OR( )[illegible]

[illegible] INT [illegible] INT FACTOR READ [illegible]

[illegible] INT [illegible] INT=0 [illegible]

D4 : OUT A

D5 : OUT B

1 [illegible]

[illegible] OUT A,B [illegible] HARD INITIALIZE1 [illegible]

[illegible] 15 [illegible]

OUT A,B [illegible]

D9 : FSSTOP

1 FSSTOP [illegible] FSSTOP [illegible]

[illegible] STATUS2 PORT [illegible] D1 [illegible]

(7) DRIVE STATUS4 PORT

○ ► fi ㄹ — ◖ fi Ꞁ ㄷ ㄹ ≤ ○ ㄷ ₱ ㄷ ㄴ ㄷ ╝ Δ ╢ PORT ≠ Δ ⁹ ╖ ⇔ │ ≠ Δ ⁹

D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	DFL OVF	DFLINT COMP3	DFLINT COMP2	DFLINT COMP1

D7	D6	D5	D4	D3	D2	D1	D0
PULSE OVF	CNTINT COMP3	CNTINT COMP2	CNTINT COMP1	ADDRESS OVF	ADRINT COMP3	ADRINT COMP2	ADRINT COMP1

D0 : ADRINT COMP1

1 ▪ ♪ ☻ ○ ► fi ㄹ — ㄱ ⅜ COMPARE REGISTER1 — ㄹ ≤ ㅂ ⇔ √ ↓ ≤ ╝ ⇔ ╕ Δ ⁹

D1 : ADRINT COMP2

1 ▪ ♪ ☻ ○ ► fi ㄹ — ㄱ ⅜ COMPARE REGISTER2 — ㄹ ≤ ㅂ ⇔ √ ↓ ≤ ╝ ⇔ ╕ Δ ⁹

D2 : ADRINT COMP3

1 ▪ ♪ ☻ ○ ► fi ㄹ — ㄱ ⅜ COMPARE REGISTER3 — ㄹ ≤ ㅂ ⇔ √ ↓ ≤ ╝ ⇔ ╕ Δ ⁹

ㅁ ADRINT COMP1,2,3 — ㄹ ⁸ ⅔ ╟ ┘ ▬ ⁱ ▪ ㄹ │ ADDRESS COUNTER INITIALIZE1,2 ◖ ∀ fi ♪ ≠ ⇔ ╕ Δ ⁹

D3 : ADDRESS OVF

1 ▪ ♪ ☻ ○ ► fi ㄹ — ㄱ ⅜ ○ ㄷ ₱ ㄷ ㄴ ㄷ ⇔ √ ↓ ≤ ╝ ⇔ ╕ Δ ⁹

ㅁ ADDRESS OVF │ ADDRESS COUNTER PRESET ◖ ∀ fi ♪ — ≠ ▬ ⁱ ▪ ⇔ ╕ Δ ⁹

D4 : CNTINT COMP1

1 Ꞁ ⁱ ☻ ○ ► fi ㄹ — ㄱ ⅜ COMPARE REGISTER1 — ㄹ ≤ ㅂ ⇔ √ ↓ ≤ ╝ ⇔ ╕ Δ ⁹

D5 : CNTINT COMP2

1 Ꞁ ⁱ ☻ ○ ► fi ㄹ — ㄱ ⅜ COMPARE REGISTER2 — ㄹ ≤ ㅂ ⇔ √ ↓ ≤ ╝ ⇔ ╕ Δ ⁹

D6 : CNTINT COMP3

1 Ꞁ ⁱ ☻ ○ ► fi ㄹ — ㄱ ⅜ COMPARE REGISTER3 — ㄹ ≤ ㅂ ⇔ √ ↓ ≤ ╝ ⇔ ╕ Δ ⁹

ㅁ CNTINT COMP1,2,3 — ㄹ ⁸ ⅔ ╟ ┘ ▬ ⁱ ▪ ㄹ │ PULSE COUNTER INITIALIZE1,2 ◖ ∀ fi ♪ ≠ ⇔ ╕ Δ ⁹

D7 : PULSE OVF

1 Ꞁ ⁱ ☻ ○ ► fi ㄹ — ㄱ ⅜ ○ ㄷ ₱ ㄷ ㄴ ㄷ ⇔ √ ↓ ≤ ╝ ⇔ ╕ Δ ⁹

ㅁ PULSE OVF │ PULSE COUNTER PRESET ◖ ∀ fi ♪ — ≠ ▬ ⁱ ▪ ⇔ ╕ Δ ⁹

D8 : DFLINT COMP1

1 Ꞁ ⁱ ☻ ㅈ ○ ► fi ㄹ — ㄱ ⅜ COMPARE REGISTER1 — ㄹ ≤ ㅂ ⇔ √ ↓ ≤ ╝ ⇔ ╕ Δ ⁹

D9 : DFLINT COMP2

1 Ꞁ ⁱ ☻ ㅈ ○ ► fi ㄹ — ㄱ ⅜ COMPARE REGISTER2 — ㄹ ≤ ㅂ ⇔ √ ↓ ≤ ╝ ⇔ ╕ Δ ⁹

D10: DFLINT COMP3

1 Ꞁ ⁱ ☻ ㅈ ○ ► fi ㄹ — ㄱ ⅜ COMPARE REGISTER3 — ㄹ ≤ ㅂ ⇔ √ ↓ ≤ ╝ ⇔ ╕ Δ ⁹

ㅁ DFLINT COMP1,2,3 — ㄹ ⁸ ⅔ ╟ ┘ ▬ ⁱ ▪ ㄹ │ DFL COUNTER INITIALIZE1,2 ◖ ∀ fi ♪ ≠ ⇔ ╕ Δ ⁹

ㅁ DFLINT ○ ► fi ㄹ │ ⁸ 16 ⅟ ㄷ ♩ — Ł ㄷ ♪ ○ ► fi ㄹ ≤ ⇔ ≡ ≠ ⅝ ╕ Δ ⁹

Ł ㄷ ♪ ○ ► fi ㄹ — COMPARE REGISTER ㄹ ╝ ╢ ↓ ≤ ⅜ ≠ ⅝ ╕ Δ ⁹

D11: DFL OVF

1 Ꞁ ⁱ ☻ ㅈ ○ ► fi ㄹ — ㄱ ⅜ ○ ㄷ ₱ ㄷ ㄴ ㄷ ⇔ √ ↓ ≤ ╝ ⇔ ╕ Δ ⁹

ㅁ DFL OVF │ DFL COUNTER PRESET ◖ ∀ fi ♪ — ≠ ▬ ⁱ ▪ ⇔ ╕ Δ ⁹

(8) DRIVE STATUS5 PORT

طمي كخ ≤ ♪ ! ▬Ķ CHANGE خ — — ╩ Δ ╢ PORT ≠ Δ ⁹ ╥ ⇔ │ ≠ Δ ⁹

D15	D14	D13	D12	D11	D10	D9	D8
INDEX CSET	INDEX CBUSY	SPEED CSET	SPEED CBUSY	RATE CSET	CPP MASK	CPPOUT	CPPIN

D7	D6	D5	D4	D3	D2	D1	D0
EB1 *1	EA1 *1	EB *1	EA0 *1	CCWMS	CWMS	SS1	SS0

D0 : SS0
1 ☿ fi ◘ طمي SS0 ⅜ ▪ ◒ ♥ ▫ Ķ˝ ȚĮ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D1 : SS1
1 ☿ fi ◘ طمي SS1 ⅜ ▪ ◒ ♥ ▫ Ķ˝ ȚĮ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ה ☿ fi ◘ طمي SS0,SS1 │ ☿ fi ◘ ⌐ ╟╫ ╡ SENSORn0,n1 طمي كخ ⁸ SIGNAL INn0,n1 طمي كخ ⁸
╕ √ │ ش — ☻ ♥ ◘ ح ⁸ كخ (OUT A,B)╩ ╡ ↑ ↕ ╢ ↓ ≤ ⅜≠ ⅝╕ Δ ⁹
م ⇔ ⁸ MANUAL ⁸ ح ♪ │ ☿ fi ◘ ⌐ ╟╫ ╡ ↑ ↕ │ ≤ ⌂ ╡ ⁸ J3 ◖ ⅛ ◒ ⁸ — SS0,SS1 طمي
كخ ⅜ SEL A SEL D طمي كخ ⌐ ╟╫ MANUAL SCAN ♪ ! ▬Ķ — SS0,SS1 ⌐ ↕ ╢ ╕ Δ ⁹

D2 : CWMS
1 CWMS طمي كخ ⅜ ▪ ◒ ♥ ▫ Ķ˝ ȚĮ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D3 : CCWMS
1 CCWMS طمي كخ ⅜ ▪ ◒ ♥ ▫ Ķ˝ ȚĮ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ה CWMS,CCWMS │ MANUAL ⁸ ح ♪ ⁸ SEL A SEL D طمي كخ ≠ ↕ ╢ √ — ╥ ≠ Δ ⁹

D4 : EA0 *1
1 ÑEA طمي كخ ⅜ ▪ ◒ ♥ ▫ Ķ˝ ȚĮ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D5 : EB0 *1
1 ÑEB طمي كخ ⅜ ▪ ◒ ♥ ▫ Ķ˝ ȚĮ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ה Xn ,Zn ,Bn — ⁸ EA0,EB0 ⌐ ÑEA,ÑEB طمي ╩ ⇔ ╕ Δ ⁹

D6 : EA1 *1
1 ÑEA طمي كخ ⅜ ▪ ◒ ♥ ▫ Ķ˝ ȚĮ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D7 : EB1 *1
1 ÑEB طمي كخ ⅜ ▪ ◒ ♥ ▫ Ķ˝ ȚĮ ⁱ ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ה Yn ,An ,Cn — ⁸ EA1,EB1 ⌐ ÑEA,ÑEB طمي ╩ ⇔ ╕ Δ ⁹

*1 ◄ fi ◖ ح ♭ طمي ╩ ◘ W ح ♩ ⇔ ≡ ™ ⌂ ™ ⅜№╡ ╕ Δ ⁹
⇔ ح │ ⁸ 4-5. ⁸ W ح ♪ ث پ ⁹ ╩ ↔ ح ∞ ↕ ™ ⁹

D8 : CPPIN
1 CPPIN كخ — — طمي ⅜ Ł ▬˝ ȚĮ ⁱ طمي ⇔ —

D9 : CPPOUT
1 CPPOUT كخ — — ⅜ Ł ▬˝ ȚĮ ⁱ ⇔ —

D10: CPP MASK
1 CPPIN طمي — ☻ ◒ ⅜ ☻ ◒ ⇔ ≡ ™ ╢
0 STATUS1 PORT — ERROR = 1 ⅜ 0 ≠ ◒ ⁱ ▪ ⇔ ╕ Δ

ה ◘ Ķ ♪ ! ▬Ķ — CPPIN ☻ ◒ ⅜ ⅛ Δ ╢ ≤ ⁸ CPP MASK = 1 ⌐ ⌂ ╡ ╕ Δ ⁹
CPPIN طمي │ ⁸ X, Y — CPP MASK = 1 — OR ≠ ☻ ◒ ⇔ ╕ Δ ⁹

D11 : RATE CSET

1 RATE CHANGE ⅜ ⇔ ╕ Δ 9

0 RATE CHANGE ⌂ ⇔ ╕ Δ 9

ה CHANGE ♪ ! K CHANGE ⇔ ╕ Δ 9

RATE CHANGE ♪ SPEED CBUSY=0 ⇔ ≡ ⅛ ⇔ ≡ ∞ ™ 9

D12 : SPEED CBUSY

1 ♪ ! K CHANGE ⇔ ╕ Δ 9

0 ♪ ! K CHANGE ⌂ ⇔ ╕ Δ 9

ה ♪ ! K CHANGE ♪ SPEED CBUSY=0 ⇔ ≡ ⅛ ⇔ ≡ ∞ ™ 9

D13 : SPEED CSET

1 ♪ ! K CHANGE ⅜ ⇔ ╕ Δ 9

0 ♪ ! K CHANGE ⌂ ⇔ ╕ Δ 9

ה CHANGE CHANGE ⇔ ╕ Δ 9

♪ ! K CHANGE ♪ SPEED CSET=0 ⇔ ≡ ⅛ ⇔ ≡ ∞ ™ 9

D14 : INDEX CBUSY

1 INDEX CHANGE ♪ ⇔ ╕ Δ 9

0 INDEX CHANGE ♪ ⌂ ⇔ ╕ Δ 9

ה INDEX CHANGE ♪ INDEX CBUSY=0 ⇔ ≡ ⅛ ⇔ ≡ ∞ ™ 9

D15 : INDEX CSET

1 INDEX CHANGE ⅜ ⇔ ╕ Δ 9

0 INDEX CHANGE ⌂ ⇔ ╕ Δ 9

ה CHANGE INDEX CHANGE ⇔ ╕ Δ 9

INDEX CHANGE ♪ INDEX CSET=0 ⇔ ≡ ⅛ ⇔ ≡ ∞ ™ 9

o ♪ ! K CHANGE ♪ x

ה ♪ UDC SPEC SET 8 SPEED CHANGE SPEC SET

ה ♪ UP DRIVE 8 DOWN DRIVE 8 CONST DRIVE 8 SPEED CHANGE 8 RATE CHANGE

o ♪ ! K CHANGE x

ה UP/DOWN/CONST DRIVE ⇔ √ SS0, SS1

1-3-3. I/O PORT ث

(1) PORT(WRITE)

كخ — ♦ ⊃ ⅈ ⅀ ⅝ ╖ PORT ≠ Δ ⁹ ↓ — PORT ┼ — ⅝ ╖ │ ≠ Δ ⁹

D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	0	0	0	0

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	$\overline{OUTn3}$	$\overline{OUTn2}$	$\overline{OUTn1}$	$\overline{OUTn0}$

D0 : $\overline{OUTn0}$
D1 : $\overline{OUTn1}$
D2 : $\overline{OUTn2}$
D3 : $\overline{OUTn3}$

1 كخ — ⅀ ▪ ♥ ▫ Ϟ ⌐ ⇔ ╕ Δ ⁹
0 كخ — ⅀ ⅛ ⅜ ▪ ♥ ▫ Ϟ ⌐ ⇔ ╕ Δ ⁹ (ثم)

(2) PORT(READ)

— كخ — ⅀ ╖ ⇔ Δ ╢ PORT ≠ Δ ⁹ ↓ — PORT — ╖ ⇔ │ ≠ Δ ⁹

D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	0	0	0	0

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	$\overline{OUTn3}$	$\overline{OUTn2}$	$\overline{OUTn1}$	$\overline{OUTn0}$

D0 : $\overline{OUTn0}$
D1 : $\overline{OUTn1}$
D2 : $\overline{OUTn2}$
D3 : $\overline{OUTn3}$

1 كخ — ⅜ ▪ ♥ ▫ Ϟ ≠ № ╢ ↓ ≥ ⅀ ⇔ ╕ Δ ⁹
0 كخ — ⅜ ⅛ ⅜ ▪ ♥ ▫ Ϟ ≠ № ╢ ↓ ≥ ⅀ ⇔ ╕ Δ ⁹

(3) طص PORT(READ)

— طص كخ — ⅀ ╖ ⇔ Δ ╢ PORT ≠ Δ ⁹ ↓ — PORT — ╖ ⇔ │ ≠ Δ ⁹

D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	0	0	0	0

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	$\overline{INn3}$	$\overline{INn2}$	$\overline{INn1}$	$\overline{INn0}$

D0 : $\overline{INn0}$
D1 : $\overline{INn1}$
D2 : $\overline{INn2}$
D3 : $\overline{INn3}$

1 كخ — طص ⅜ ▪ ♥ ▫ Ϟ ≠ № ╢ ↓ ≥ ⅀ ⇔ ╕ Δ ⁹
0 كخ — طص ⅜ ⅛ ⅜ ▪ ♥ ▫ Ϟ ≠ № ╢ ↓ ≥ ⅀ ⇔ ╕ Δ ⁹

1-3-4. HARD CONFIGURATION PORT 仕様

(1) HARD CONFIG COMMAND PORT

HARD CONFIG COMMAND を書き込む PORT です。この PORT に HARD CONFIG COMMAND を書き込むと、データの設定または機能の実行を行います。
書き込む HARD CONFIG COMMAND は下位 8 ビットのみ有効です。上位 8 ビットは無視します。
HARD CONFIG COMMAND の書き込みは常時可能です。

(2) HARD CONFIG DATA1,2,3 PORT(WRITE)

HARD CONFIG COMMAND の設定データ、または機能操作を行うデータを書き込む PORT です。
この PORT への書き込みは常時可能です。

(3) HARD CONFIG DATA1,2,3 PORT(READ)

HARD CONFIG の設定データを読み出す PORT です。読み出しは常時可能です。
HARD CONFIG SET DATA READ コマンドを HARD CONFIG COMMAND PORT に書き込むと、該当データを HARD CONFIG DATA1,2,3 PORT にセットします。
HARD CONFIG DATA1,2 3 PORT にセットしたデータは、次の HARD CONFIG SET DATA READ コマンドの書き込みまで保持します。

(4) HARD CONFIG STATUS1 PORT

MAN RDY 出力信号の状態を読み出す PORT です。読み出しは常時可能です。

D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	0	0	0	(不定)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	$\overline{\text{MAN RDY}}$	0	0	0	(不定)

D4 : $\overline{\text{MAN RDY}}$
1：$\overline{\text{MAN}}$ 入力信号受付可能の状態を示します。(初期値)
0：$\overline{\text{MAN}}$ 入力信号受付不可の状態を示します。

・ $\overline{\text{MAN RDY}}$ の状態は MAN MASK コマンドで操作することができます。

(5) HARD CONFIG STATUS2 PORT

各軸の PAUSE 状態を一括で表示する PORT です。読み出しは常時可能です。

D15	D14	D13	D12	D11	D10	D9	D8	
0	0	C2 PAUSE	B2 PAUSE	A2 PAUSE	Z2 PAUSE	Y2 PAUSE	X2 PAUSE	C-VX872v1, C-VX872Ev1 C-VX873v1, C-VX873Ev1
		0	0	0	0	0	0	C-VX870v1, C-VX870Ev1 C-VX871v1, C-VX871Ev1

D7	D6	D5	D4	D3	D2	D1	D0	
0	0	C1 PAUSE	B1 PAUSE	A1 PAUSE	Z1 PAUSE	Y1 PAUSE	X1 PAUSE	C-VX872v1, C-VX872Ev1 C-VX873v1, C-VX873Ev1
		C PAUSE	B PAUSE	A PAUSE	Z PAUSE	Y PAUSE	X PAUSE	C-VX870v1, C-VX870Ev1 C-VX871v1, C-VX871Ev1

C-VX870v1, C-VX870Ev1
C-VX872v1, C-VX872Ev1
C-VX871v1, C-VX871Ev1
C-VX873v1, C-VX873Ev1

D0-D5 : X-C PAUSE(X1-C1 PAUSE)
D8-D13 : X2-C2 PAUSE
1：該当軸が待機指令中であることを示します。

・ 各軸 DRIVE STATUS1 PORT の PAUSE と同じです。

(6) HARD CONFIG STATUS3 PORT

各軸の STBY 状態を一括で表示する PORT です。読み出しは常時可能です。

D15	D14	D13	D12	D11	D10	D9	D8	
0	0	C2 STBY (C2 OUT A)	B2 STBY (B2 OUT A)	A2 STBY (A2 OUT A)	Z2 STBY (Z2 OUT A)	Y2 STBY (Y2 OUT A)	X2 STBY (X2 OUT A)	C-VX872v1, C-VX872Ev1 C-VX873v1, C-VX873Ev1
		0	0	0	0	0	0	C-VX870v1, C-VX870Ev1 C-VX871v1, C-VX871Ev1

D7	D6	D5	D4	D3	D2	D1	D0	
0	0	C1 STBY (C1 OUT A)	B1 STBY (B1 OUT A)	A1 STBY (A1 OUT A)	Z1 STBY (Z1 OUT A)	Y1 STBY (Y1 OUT A)	X1 STBY (X1 OUT A)	C-VX872v1, C-VX872Ev1 C-VX873v1, C-VX873Ev1
		C STBY (C OUT A)	B STBY (B OUT A)	A STBY (A OUT A)	Z STBY (Z OUT A)	Y STBY (Y OUT A)	X STBY (X OUT A)	C-VX870v1, C-VX870Ev1 C-VX871v1, C-VX871Ev1

C-VX870v1, C-VX870Ev1
C-VX872v1, C-VX872Ev1
C-VX871v1, C-VX871Ev1
C-VX873v1, C-VX873Ev1

D0-D5 : X-C STBY(X1-C1 STBY)
D8-D13 : X2-C2 STBY
　1：該当軸のパルス出力の準備が完了した状態を示します。

・ 各軸 DRIVE STATUS3 PORT の OUT A と同じです。
・ STBY 状態を表示するために、全軸の OUT A 信号にそれぞれ STBY フラグを出力する必要があります。

(7) HARD CONFIG STATUS4 PORT

J3 コネクタの $\overline{\text{SIGNAL INn0,n1}}$ 入力信号と $\overline{\text{SIGNAL\_OUTn0,n1}}$ 出力信号の状態を表示する PORT です。
読み出しは常時可能です。

D15	D14	D13	D12	D11	D10	D9	D8	
0	0	SIGNAL IN21	SIGNAL IN20	SIGNAL OUT21 LATCH	SIGNAL OUT20 LATCH	SIGNAL OUT21	SIGNAL OUT20	C-VX872v1, C-VX872Ev1 C-VX873v1, C-VX873Ev1
		0	0	0	0	0	0	C-VX870v1, C-VX870Ev1 C-VX871v1, C-VX871Ev1

D7	D6	D5	D4	D3	D2	D1	D0	
0	0	SIGNAL IN11	SIGNAL IN10	SIGNAL OUT11 LATCH	SIGNAL OUT10 LATCH	SIGNAL OUT11	SIGNAL OUT10	C-VX872v1, C-VX872Ev1 C-VX873v1, C-VX873Ev1
		SIGNAL IN1	SIGNAL IN0	SIGNAL OUT1 LATCH	SIGNAL OUT0 LATCH	SIGNAL OUT1	SIGNAL OUT0	C-VX870v1, C-VX870Ev1 C-VX871v1, C-VX871Ev1

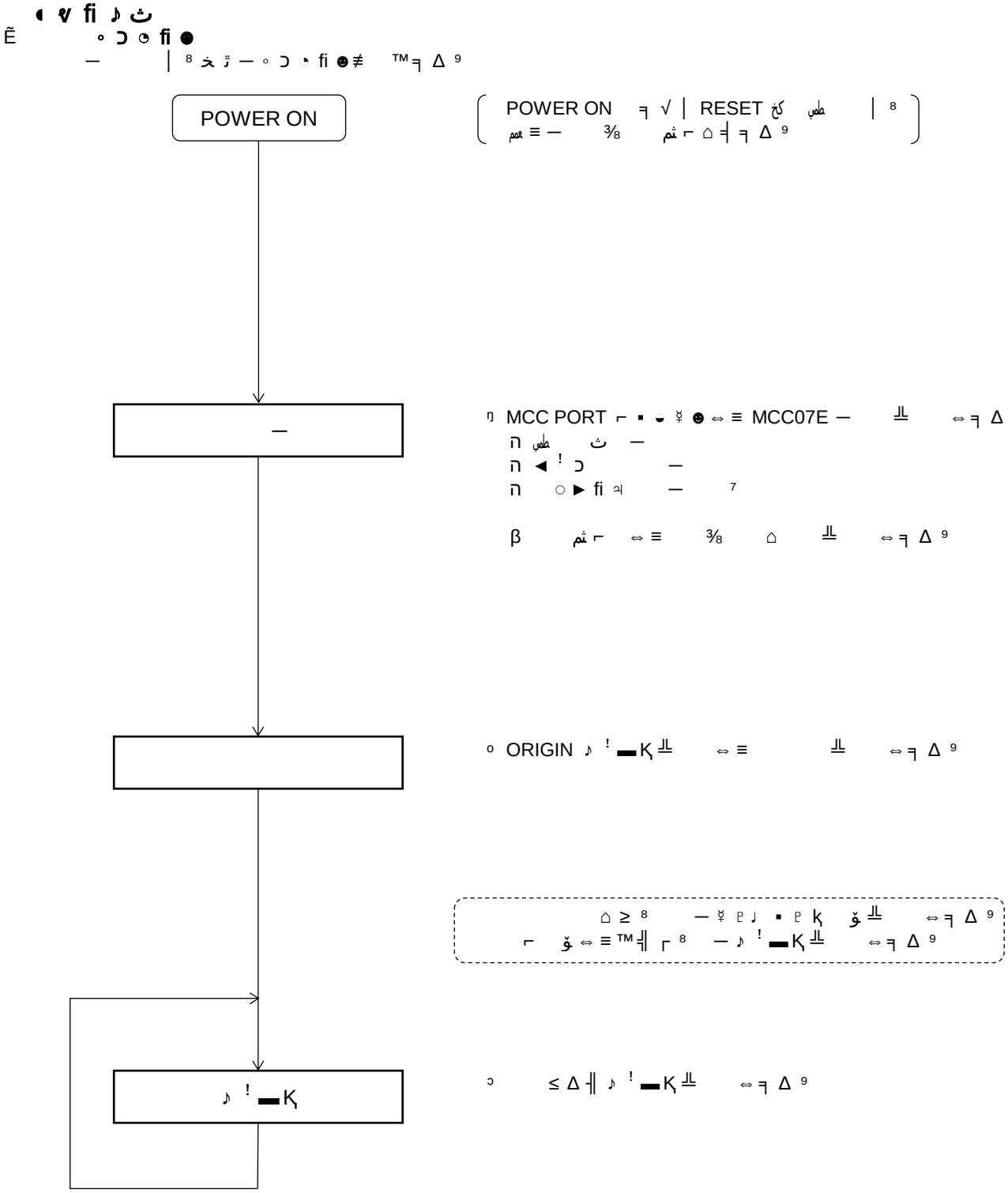
D0-D1 : SIGNAL OUT x(SIGNAL OUT1x)
D8-D9 : $\overline{\text{SIGNAL OUT2x}}$
　1：$\overline{\text{SIGNAL OUTnx}}$ 出力信号がアクティブレベルであることを示します。

D2-D3 : SIGNAL OUT x LATCH(SIGNAL OUT1x LATCH)
D10-D11 : $\overline{\text{SIGNAL OUT2x}}$ LATCH
　1：$\overline{\text{SIGNAL OUTnx}}$ 出力信号のアクティブエッジを検出したことを示します。

・ SIGNAL OUTnx LATCH は SIGNAL OUT LATCH STATUS CLR コマンドの実行でクリアします。

D4-D5 : SIGNAL IN x(SIGNAL IN1x)
D12-D13 : $\overline{\text{SIGNAL IN2x}}$
　1：$\overline{\text{SIGNAL INnx}}$ 入力信号がアクティブレベルであることを示します。

・ MANUAL モード時は SIGNAL INnx は 0 となります。


POWER ON
POWER ON
RESET
MCC PORT
MCC07E
ORIGIN

STATUS1 PORT
ERROR = 0 ?
N *1
Y
N
STATUS1 PORT
BUSY = 0 ?
Y
DATA1,2 PORT
WRITE
N
STATUS1 PORT
BUSY = 0 ?
Y
STATUS1 PORT
ERROR = 0 ?
N *1
Y

2-1. ♪ ! ▬Ķ ◖ ∀ fi ♪

2-1-1. ظص ث —

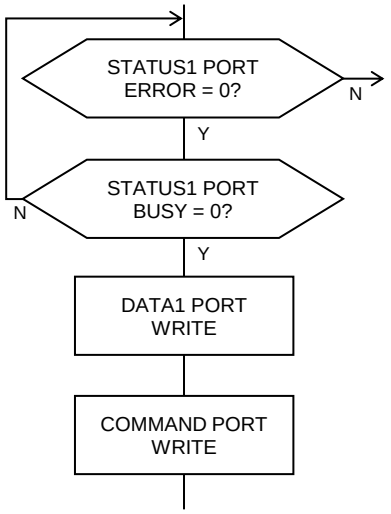
(1) SPEC INITIALIZE1

♪ ! ▬Ķ Ꞁ i ☻ — ث ⇓ ⇔ ╕ Δ 9

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'01

SPEC INITIALIZE1 COMMAND



ŋ DRIVE STATUS1 PORT — ERROR ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ⇓ ⇔ ╕ Δ 9

o DRIVE STATUS1 PORT — BUSY ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ⇓ ⇔ ╕ Δ 9

ɔ DRIVE DATA1 PORT ⌐ ♦ ↄ ⅁ ⇓ ⅝ ╗ ╕ Δ 9

ˆ DRIVE COMMAND PORT ⌐ COMMAND ⇓ ⅝ ╗ ╕ Δ 9

DRIVE DATA1 PORT — ♦ ↄ ⅁

D15	D14	D13	D12	D11	D10	D9	D8

D7	D6	D5	D4	D3	D2	D1	D0
			MANUAL DRIVE MODE		PULSE OUTPUT MASK	PULSE OUTPUT TYPE1	PULSE OUTPUT TYPE0

ÙΨ ⅈ ⅊ ⅋ ♩ — ثم | H'00 (▪ fi ♭ ↄ ! ▬ fi سخ) ≠ Δ 9

D0 PULSE OUTPUT TYPE0

D1 PULSE OUTPUT TYPE1

CWP, CCWP كخ — ♪ ! ▬Ķ Ꞁ i ☻ ⇓ ⇔ ╕ Δ 9

TYPE1	TYPE0	Ꞁ i ☻	CWP كخ	CCWP كخ
0	0		— Ꞁ i ☻	— Ꞁ i ☻
0	1		Ꞁ i ☻	
1	0	2 ه — يى كخ	A	B
1	1	4 ه — يى كخ	A	B

ה — 8 CCWP كخ = HIGH ≠ +(CW) , CCWP كخ = LOW ≠ -(CCW) ⇓ ⇔ ╕ Δ 9

D2 PULSE OUTPUT MASK

CWP, CCWP كخ — ♪ ! ▬Ķ Ꞁ i ☻ ⇓ 8 ∀ ☻ ▾ Δ ╢ ∀ ☻ ▾ ⇔ ⌂ ™ 9 ⇓ ⇔ ╕ Δ 9

0 ♪ ! ▬Ķ Ꞁ i ☻ ⇓ ∀ ☻ ▾ ⇔ ⌂ ™ Ꞁ i ☻ ⇓ Δ ╢

1 ♪ ! ▬Ķ Ꞁ i ☻ ⇓ ∀ ☻ ▾ Δ ╢ Ꞁ i ☻ ⇓ ⇔ ⌂ ™

ה 8 ∀ ☻ ▾ Δ ╢ 9 ⇓ ⇔ √ | 8 CWP, CCWP كخ — ⇓OFF" ⅄ i ⌐ ⇔ ╕ Δ 9

▪ ♪" ☻ ○ ▶ fi ⅁ | 8 ○ ▶ fi ♩ Ꞁ i ☻ — ○ ▶ fi ♩ ⇓ تجـ ⇔ 8 ○ ▶ fi ♩ Ꞁ i ☻ — ⇓ تجـ ⇔ ╕ Δ 9

Ꞁ i ☻ ○ ▶ fi ⅁ ≤ Ꞁ i ☻ ئجـ ○ ▶ fi ⅁ | Ꞁ i ☻ ⇓ ○ ▶ fi ♩ Δ ╢ ↓ ≤ ⅜ ≠ ⅝ ╕ ∑ ╪ 9

▪ ♪" ☻ ○ ▶ fi ⅁ ⅜ تجـ Δ ╢ √ ⅃ 8 ABS INDEX ♪ ! ▬Ķ ⇓ Δ ╢ ≤ تجـ ≠ ⅝ ╕ ∑ ╪ 9

/ —ث— | 8 ∀ ☻ ▾ ⇔ ⌂ ™ 9 ⇓ ⇔ √ ≤ ≠ Δ 9

ה 8 ∀ ☻ ▾ Δ ╢ 9 ⌐ Δ ╢ ≤ 8 DRIVE STATUS2 PORT —PULSE MASK = 1 ⌐ ⌂ ⅎ ╕ Δ 9

Ꞁ i ☻ ⇓ ∀ ☻ ▾ ⇔ √ ♪ ! ▬Ķ — | 8 ⅁ ▬ ∀ ≤ ⇔ ≡ ئو ≠ ⅝ ╕ Δ 9

D4 MANUAL DRIVE MODE

MANUAL ♪ ! ▬Ķ — ♪ ! ▬Ķ ⇓ ⇔ ╕ Δ 9

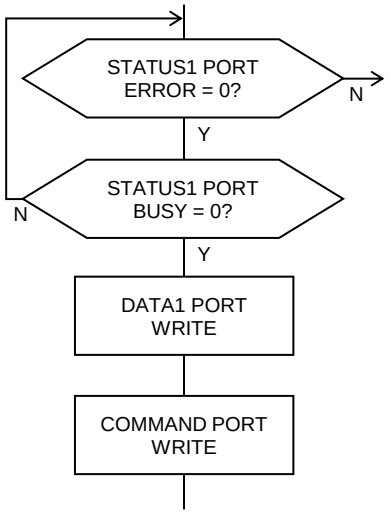
0 SCAN ♪ ! ▬Ķ

1 JOG ♪ ! ▬Ķ

(2) SPEC INITIALIZE2

CWLM, CCWLM كخ — طمي ⁸ RDYINT — ث ⁸ ☿ fi ◘ كخ SS0, SS1 — طمي ╩ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'02 SPEC INITIALIZE2 COMMAND



ŋ DRIVE STATUS1 PORT — ERROR ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

o DRIVE STATUS1 PORT — BUSY ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ɔ DRIVE DATA1 PORT ⌐ ♦ ↄ ㅋ ╩ ⅝ ╖ ╕ Δ ⁹

ˆ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

DRIVE DATA1 PORT — ♦ ↄ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8
				SS1 TYPE1	SS1 TYPE0	SS0 TYPE1	SS0 TYPE0

D7	D6	D5	D4	D3	D2	D1	D0
		RDYINT TYPE1	RDYINT TYPE0	CCWLM TYPE1	CCWLM TYPE0	CWLM TYPE1	CWLM TYPE0

ùr ˡ ☿ ᵖ ♩ — ثم | H'F00 (▪ fi ♭ ↄ ! ▬ fi سخ) ≠ Δ ⁹

D0 CWLM TYPE0
D1 CWLM TYPE1

CWLM كخ طمي — ▪ ◒ ♥ ▫ Ꝗ ˢ ᶻ ˡ — طمي ╩ ⇔ ╕ Δ ⁹

TYPE1	TYPE0	CWLM كخ — طمي
0	0	— LIMIT تجـ كخ ≤ ⇔ ≡ ئو Δ ╢
0	1	— LIMIT تجـ كخ ≤ ⇔ ≡ ئو Δ ╢
1	0	تجـ كخ ≤ ⇔ ≡ ئو Δ ╢
1	1	طمي ≤ ⇔ ≡ ئو Δ ╢

D2 CCWLM TYPE0
D3 CCWLM TYPE1

CCWLM كخ طمي — ▪ ◒ ♥ ▫ Ꝗ ˢ ᶻ ˡ — طمي ╩ ⇔ ╕ Δ ⁹

TYPE1	TYPE0	CCWLM كخ — طمي
0	0	— LIMIT تجـ كخ ≤ ⇔ ≡ ئو Δ ╢
0	1	— LIMIT تجـ كخ ≤ ⇔ ≡ ئو Δ ╢
1	0	تجـ كخ ≤ ⇔ ≡ ئو Δ ╢
1	1	طمي ≤ ⇔ ≡ ئو Δ ╢

o ڤ x

LIMIT تجـ ╩ تجـ ≠ ئو Δ ╢ ⁸ LIMIT تجـ خ ⅜ ▪ ◒ ♥ ▫ Ꝗ ≠ ╚

LIMIT ⌐ ♪ ! ▬ Ꝗ ╩ Δ ╢ ≤ ⁸ Ꝓ ˡ ☻ ╩ ⇔ ≡ تجـ ⇔ ╕ Δ ⁹

o x

LIMIT تجـ ╩ تجـ ≠ ئو Δ ╢ │ LSEND=1 ≠ ERROR ≤ ⌂ ╢ ╟ e ⌐

ERROR STATUS MASK ◖ ∀ fi ♪ ≠ ⇔ ≡ ↕ ∞ ↕ ™ ⁹

↓ ─ ⌐ ╟ ╡ ⁸ ERROR ≤ ⌂ ∫ √ ≤ ⅝ │ ─ ♪ ! ▬ Ķ ╩ ⇔ ╕ Δ ⁹

ERROR │ ERROR STATUS CLR ◖ ∀ fi ♪ ≠ ERROR ╩ ▪ ⁱ ▪ ⇔ √ ⁸

STATUS2 PORT ╩ ╖ ⇔ ≡ ⁸ ≥ ∟ ╟ ─ LIMIT(CWLM/CCWLM) ⌐ طص ⇔ ≡ ™ ╢ ⅛ ╩

⇔ ⁸ LIMIT ┼ ─ ORIGIN ♪ ! ▬ Ķ ≠ ╩ ⇔ ≡ ↕ ™ ⁹

D4 RDYINT TYPE0

D5 RDYINT TYPE1

◖ ∀ fi ♪ ﻭ ─ ╡ ╖ RDYINT ─ ث ╩ ⇔ ╕ Δ ⁹

TYPE1	TYPE0	RDYINT─ ث
0	0	STATUS1PORT─DRVEND = 0 ǵ 1 ─◄ ☺ ≠ ⁸ RDYINT = 1 ⌐ Δ ╢
0	1	
1	0	STATUS1 PORT─DRIVE = 1 ǵ 0 ─◄ ☺ ≠ ⁸ RDYINT = 1 ⌐ Δ ╢
1	1	⇔ ⌂ ™(RDYINT = 0)

RDYINT ─ ▪ ⁱ ▪ ﺳ RDYINT = 0 ⌐ ⇔ ╕ Δ

ה STATUS1 PORT ─ ⁱ ↄ ♪ ﻭ ≠ ▪ ⁱ ▪

ה BUSY = 0 ǵ 1 ╕ √ │ ﻭ ◖ ∀ fi ♪ ─LOAD ≤ ⌐ ▪ ⁱ ▪

D8 SS0 TYPE0

D9 SS0 TYPE1

SS0 كخ طص ─ ▪ ▪ ♥ ▫ Ķ‟ ʒ ⁱ ─طص ╩ ⇔ ╕ Δ ⁹

D10 SS1 TYPE0

D11 SS1 TYPE1

SS1 كخ طص ─ ▪ ▪ ♥ ▫ Ķ‟ ʒ ⁱ ─طص ╩ ⇔ ╕ Δ ⁹

TYPE1	TYPE0	SS0,SS1 كخ ─طص
0	0	UP/DOWN/CONST ♪ ! ▬ Ķ CHANGE طص خ ≤ ⇔ ≡ ئو Δ ╢
0	1	╢ Δ ئو ≡ ⇔ ≤ كخ تجـ
1	0	╢ Δ ئو ≡ ⇔ ≤ كخ تجـ
1	1	╢ Δ ئو ≡ ⇔ ≤ طص

ה SS0, SS1 كخ ─ UP/DOWN/CONST ♪ ! ▬ Ķ CHANGE طص خ

SS0, SS1 كخ ╩ لأ Δ ╢ ↓ ≤ ≠ ⁸ UP/DOWN/CONST ─ ♪ ! ▬ Ķ CHANGE ⅜ ≠ ⅝ ╕ Δ ⁹

ה SS0, SS1 كخ ┼ ─ │ ⁸ SENSORnx كخ ⁸ OUTA,OUTB كخ ⁸ SIGNAL INnx كخ ─ جـ ⅛ ╟

HARD CONFIG ─ SENSOR SIGNAL SELECT ◖ ∀ fi ♪ ⌐ ╟ ╢ ⅜ ≠ Δ ⁹

SS0 TYPE = "00" ⁸

SS1 TYPE = "00"─

SS1	SS0	♪ ! ▬ Ķ CHANGE لأ
0	0	CONST DRIVE
0	1	UP DRIVE
1	0	DOWN DRIVE
1	1	CONST DRIVE

SS0 TYPE = "00" ⁸

SS1 TYPE = "00خ "─

SS1	SS0	♪ ! ▬ Ķ CHANGE لأ
*	0	│ №╡ ╕ ∑ ╪
*	1	UP DRIVE

SS0 TYPE = "00خ " ⁸

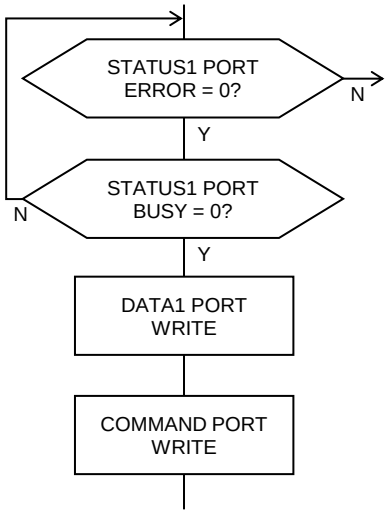
SS1 TYPE = "00"─

SS1	SS0	♪ ! ▬ Ķ CHANGE لأ
0	*	│ №╡ ╕ ∑ ╪
1	*	DOWN DRIVE

(3) SPEC INITIALIZE3

$\overline{\text{DRST}}$信号の出力機能、$\overline{\text{DEND}}$/$\overline{\text{PO}}$, DALM信号の入力機能、STBY解除条件を設定します。

< 実行シーケンス >　　COMMAND : H'03　　SPEC INITIALIZE3 COMMAND



① DRIVE STATUS1 PORT の ERROR フラグが "0" であることを確認します。

② DRIVE STATUS1 PORT の BUSY フラグが "0" であることを確認します。

③ DRIVE DATA1 PORT に設定データを書き込みます。

④ DRIVE COMMAND PORT に COMMAND を書き込みます。

DRIVE DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8
–	–	–	DOWN PULSE MASK	–	STBY TYPE2	STBY TYPE1	STBY TYPE0

D7	D6	D5	D4	D3	D2	D1	D0
–	–	DALM TYPE1	DALM TYPE0	DEND/PO TYPE1	DEND/PO TYPE0	DRST TYPE1	DRST TYPE0

● リセット後の初期値は H'003F (アンダーライン側) です。

D0 ：DRST TYPE0
D1 ：DRST TYPE1

$\overline{\text{DRST}}$ / $\overline{\text{M.F}}$ 信号の出力機能を選択します。

TYPE1	TYPE0	$\overline{\text{DRST}}$ / $\overline{\text{M.F}}$ 信号の出力機能	サーボ対応
0	0	停止時に10 ms 間アクティブレベルを出力する	<サーボ対応>
0	1	設定禁止	–
1	0	設定禁止	–
1	1	汎用出力として使用する	–

・「00」を選択した場合は、即時停止による DRST 機能が有効になります。
SIGNAL OUT コマンドによる $\overline{\text{DRST}}$ 信号からの出力は、設定された出力レベルを出力します。

・「11」を選択した場合は、即時停止による DRST 機能が無効になります。
SIGNAL OUT コマンドによる $\overline{\text{DRST}}$ 信号からの出力は、設定された出力レベルを出力します。

・$\overline{\text{DRST}}$/$\overline{\text{MF}}$ 信号をサーボモータドライバに接続しない場合、 DRST TYPE = "00" ＜サーボ対応＞には設定しないでください。

D2 ： DEND/PO TYPE0
D3 ： DEND/PO TYPE1
$\overline{\text{DEND}}$ / $\overline{\text{PO}}$ 信号の入力機能を選択します。

TYPE1	TYPE0	$\overline{\text{DEND}}$ / $\overline{\text{PO}}$信号の入力機能	サーボ対応
0	0	$\overline{\text{DEND}}$ / $\overline{\text{PO}}$ のアクティブを検出するまでドライブを終了しない	<サーボ対応>
0	1	設定禁止	ー
1	0	設定禁止	ー
1	1	汎用入力として使用する	ー

・ 00を設定した場合は、$\overline{\text{DEND}}$ 信号の〈サーボ対応〉を実行します。

● $\overline{\text{DEND}}$ 信号のサーボ対応
・ ドライブパルス出力が終了しても、
$\overline{\text{DEND}}$ 信号のアクティブレベルを検出するまで、DRIVE STATUS2 PORT のDEND BUSY = 1 にします。
$\overline{\text{DEND}}$ BUSY = 1 の間は、DRVE STATUS1 PORT のBUSY = 1 のままです。
$\overline{\text{DEND}}$ 信号のアクティブレベルを検出すると、DEND BUSY = 0 にします。

・ 即時停止指令を検出すると、$\overline{\text{DEND}}$ 信号の<サーボ対応>を中止して、DEND BUSY = 0 にします。

D4 ： DALM TYPE0
D5 ： DALM TYPE1
DALM 信号のアクティブレベル検出時の入力機能を選択します。

TYPE1	TYPE0	DALM 信号の入力機能	サーボ対応
0	0	機能はありません(汎用入力)	ー
0	1	減速停止信号として使用する	ー
1	0	即時停止信号として使用する	ー
1	1	汎用入力として使用する	ー

・ MCC07E への DALM 信号入力は、汎用入力信号 $\overline{\text{INn0}}$--$\overline{\text{INn3}}$ によって機能します。
$\overline{\text{INn0}}$--$\overline{\text{INn3}}$ 信号の状態は、汎用入力 PORT からも確認できます。
汎用入力信号 $\overline{\text{INn0}}$--$\overline{\text{INn3}}$ を停止信号として利用するとき、DALM TYPE を停止機能に設定します。

汎用入力信号	C-VX870v1 C-VX870Ev1
$\overline{\text{IN0}}$	X 軸
$\overline{\text{IN1}}$	Y 軸
$\overline{\text{IN2}}$	Z 軸
$\overline{\text{IN3}}$	A 軸

汎用入力信号	C-VX872v1 C-VX872Ev1
$\overline{\text{IN10}}$	X1 軸
$\overline{\text{IN11}}$	Y1 軸
$\overline{\text{IN12}}$	Z1 軸
$\overline{\text{IN13}}$	A1 軸
$\overline{\text{IN20}}$	X2 軸
$\overline{\text{IN21}}$	Y2 軸
$\overline{\text{IN22}}$	Z2 軸
$\overline{\text{IN23}}$	A2 軸

D8 ： STBY TYPE0
D9 ： STBY TYPE1
D10 ： STBY TYPE2
STATUS1 PORT の STBY フラグを "0" にする STBY 解除条件を選択します。
STBY = 1 の状態から、STBY = 0 になるとドライブパルス出力を開始します。
同期スタート機能により PAUSE を操作することができます。

TYPE2	TYPE1	TYPE0	STBY 解除条件<レベル検出>
0	0	0	PAUSE = 0 でSTBY = 0 にする
0	0	1	設定禁止
0	1	0	設定禁止
0	1	1	設定禁止
1	0	0	設定禁止
1	0	1	設定禁止
1	1	0	設定禁止
1	1	1	設定禁止

D12 DOWN PULSE MASK

INDEX ♪ ! ▬ Ķ — تج ⅟ 8 ∀ ☻ ▾ Δ ╫ ∀ ☻ ▾ ⇔ ⌂ ™ 9 ⅟ ⇔ ┐ Δ 9

0 ∀ ☻ ▾ ⇔ ⌂ ™ تج ه ┼ ∀ ♬ ▪ i — تج ≠ لأ تج Δ ╫

1 ∀ ☻ ▾ Δ ╫ تج ه ≠ تج Δ ╫ Ꞧ i ☻ "0"

ה 8 ∀ ☻ ▾ ⇔ ⌂ ™ 9 ⅟ ⇔ √ | 8 8 ♪ ! ▬ Ķ 9 ⅔ ╫ ┘ 8 ♪ ! ▬ Ķ 9 — INDEX ♪ ! ▬ Ķ ج ⌐ 8

Ꞧ i ☻ ⅟ ⇔ ≡ ▪ ♪ ‟ ☻ ≠ تج ⇔ ┐ Δ 9

ה 8 ∀ ☻ ▾ Δ ╫ 9 ⅟ ⇔ √ | 8 تج | لأ ⇔ ┐ ∑ ╪ 9

INDEX ♪ ! ▬ Ķ — ▪ ♪ ‟ ☻ ⌐ Δ ╫ ≤ 8 Ꞧ i ☻ ⌂ ⇔ ≠ تج ⇔ ┐ Δ 9

♪ ! ▬ Ķ ⅟ 8 ♪ ! ▬ Ķ 9 ⌐ ⇔ ≡ ™ ╫ | 8 8 ♪ ! ▬ Ķ 9 — ≠ ♪ ! ▬ Ķ ⅟ ۆ

⇔ ┐ Δ 9 ↓ — — ۆ | 8 HSPD x RESOL ≠ Δ 9

♪ ! ▬ Ķ ⅟ 8 ♪ ! ▬ Ķ 9 ⌐ ⇔ ≡ ™ ╫ | 8 8 پ ♪ ! ▬ Ķ 9 — ≠ ♪ ! ▬ Ķ ⅟ ۆ

⇔ ┐ Δ 9 ↓ — — ۆ | 8 HSPD x RESOL ≠ Δ 9

S INDEX ♪ ! ▬ Ķ — ث ⅟ ⌐ ⌂ ╪ ┐ Δ 9

♪ ! ▬ Ķ ج ⌐ تج خ ⅟ ⇔ √ | 8 ۆ ┐ ≠ ⇔ ≡ ⅛ ╫ ♪ ! ▬ Ķ ⅟ ۆ ⇔ ┐ Δ 9

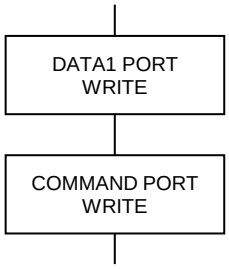
S چ — تج خ — ث ⅟ ≠ Δ 9

ه ⇔ 8 چ ⌐ ▪ ♪ ‟ ☻ ⌐ ⇔ √ | 8 ▪ ♪ ‟ ☻ ≠ تج ⇔ ┐ Δ 9

(4) HARD INITIALIZE1

☻ ♥ ⊂ ㉻ ☻ ㉾ (OUT A,B ㉾) ⌐ Δ ╢ ╩ ⇔ ╕ Δ ⁹
↓ ─ ◗ ∀ fi ♪ ─ │ ≠ Δ ⁹

< ◦ ⊃ ◔ fi ☻ > COMMAND : H'F1 HRAD INITIALIZE1 COMMAND



ŋ DRIVE DATA1 PORT ⌐ ♦ ⊃ ㉻ ╩ ⅝ ╥ ╕ Δ ⁹

º DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╥ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ⊃ ㉻

D15	D14	D13	D12	D11	D10	D9	D8
1	1	1	0	1	1	1	0

D7	D6	D5	D4	D3	D2	D1	D0
OUT B TYPE3	OUT B TYPE2	OUT B TYPE1	OUT B TYPE0	OUT A TYPE3	OUT A TYPE2	OUT A TYPE1	OUT A TYPE0

ℳ ⁱ ☿ ℙ ♩ ─ ㉷ │ H'EE21 (▪ fi ♭ ⊃ ! ▬ fi ㉿) ≠ Δ ⁹

D3--D0 OUT A TYPE3--0 ㉷ CNTINT
D7--D4 OUT B TYPE3--0 ㉷ DFLINT
OUT A,B ㉾ ⌐ Δ ╢ ☻ ㉻ ♥ ☻ ╩ ⇔ ╕ Δ ⁹

TYPE3	TYPE2	TYPE1	TYPE0	OUT A,B ㉾ ⌐ Δ ╢	
0	0	0	0	ADRINT	─ ADRINT ╥ ╡ ㉻ fi ► ○
0	0	0	1	CNTINT	─ CNTINT ╥ ╡ ㉻ fi ► ○
0	0	1	0	DFLINT	─ DFLINT ╥ ╡ ㉻ fi ► ○
0	0	1	1	RDYINT	─ RDYINT ╥ ╡ ㉻ fi ♪ ◗ ∀
0	1	0	0	STBY	STATUS1 ─ STBY ɧ ! ◓
0	1	0	1	nDRIVE	STATUS1 ─ DRIVE ɧ ! ◓ ─
0	1	1	0	nSPEED CBUSY	STATUS5 ─ SPEED CBUSY ɧ ! ◓ ─
0	1	1	1	nINDEX CBUSY	STATUS5 ─ INDEX CBUSY ɧ ! ◓ ─
1	0	0	0	UP	STATUS1 ─ UP ɧ ! ◓
1	0	0	1	DOWN	STATUS1 ─ DOWN ɧ ! ◓
1	0	1	0	CONST	STATUS1 ─ CONSY ɧ ! ◓
1	0	1	1	EXT PULSE	STATUS1 ─ EXT PULSE ɧ ! ◓
1	1	0	0	nPULSE MASK	STATUS2 ─ PULSE MASK ɧ ! ◓ ─
1	1	0	1	ORG SIGNAL	STATUS2 ─ ORG SIGNAL ɧ ! ◓
1	1	1	0		Δ ╢ ئو ≡ ⇔ ≤
1	1	1	1	PULSE OVF	STATUS4 ─ PULSE OVF ɧ ! ◓

⁸ ⁹ ╩ ⇔ √ │ ⁸ SIGNAL OUT ◗ ∀ fi ♪ ≠ ″ ℤ ⁱ ╩ ⅄ ⇔ ╕ Δ ⁹

ℳ ☻ ♥ ⊂ ㉻ ☻ ㉾ (OUT A,B ㉾) │ ☻ ㉻ ♥ ☻ ≠ I/O ◗ Ł ▬ ㉻ ⅛ ╣ Δ ╢ ↓ ≤ ⅜ ≠ ⅝ ╕ Δ ⁹
╕ √ ⁸ ☿ fi ◘ ⁸ ☻ ㉻ ♩ ⊂ ⌐ ئو Δ ╢ ↓ ≤ ⅜ ≠ ⅝ ╕ Δ ⁹
⇔ ⊃ │ 3-3. HARD CONFIG ث ╩ ↔ ⊃ ∞ ↕ ™ ⁹

(5) HARD INITIALIZE4

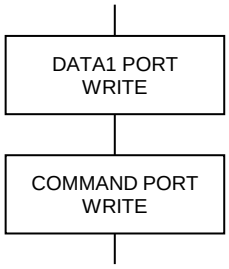
CWLM, CCWLM [illegible] DEND/PO [illegible] DALM [illegible] (INnx [illegible]) [illegible]

[illegible]

< [illegible] >

COMMAND : H'F4

HRAD INITIALIZE4 COMMAND



[illegible] DRIVE DATA1 PORT [illegible]

[illegible] DRIVE COMMAND PORT [illegible] COMMAND [illegible]

DRIVE DATA1 PORT [illegible]

D15	D14	D13	D12	D11	D10	D9	D8
	SERVO FILTER2	SERVO FILTER1	SERVO FILTER0		0	0	0

D7	D6	D5	D4	D3	D2	D1	D0
	LIMIT FILTER2	LIMIT FILTER1	LIMIT FILTER0		0	0	0

[illegible] H'0000 ([illegible])

D6--D4 LIMIT FILTER2--0 CWLM, CCWLM [illegible]

D14--D12 SERVO FILTER2--0 DEND/PO, DALM [illegible]

D2--D0 0 [illegible]

D10--D8 0 [illegible]

[illegible]

FLT2	FLT1	FLT0	[illegible]
0	0	0	0 50 ns
0	0	1	50 εs
0	1	0	100 εs
0	1	1	200 εs
1	0	0	500 εs
1	0	1	1.0 ms
1	1	0	5.0 ms
1	1	1	10.0 ms

(+10, -0 εs)

[illegible]

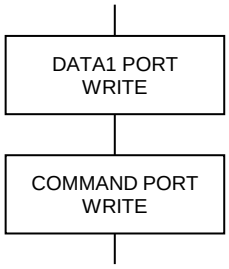
(6) HARD INITIALIZE5

ORG كج ⁸ NORG كج , ÑZORG كج طص — ♦ ☺ ⁹ i ɳ ▫ i ⁹ — ⇔ ⁹ Δ ⁹
↓ — ◖ ∀ fi ♪ — | ≠ Δ ⁹

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'F5

HRAD INITIALIZE5 COMMAND



ɳ DRIVE DATA1 PORT ⌐ ♦ ↄ ⁹ ⅝ ⁹ Δ ⁹

º DRIVE COMMAND PORT ⌐ COMMAND ⁹ ⅝ ⁹ Δ ⁹

DRIVE DATA1 PORT — ♦ ↄ ⁹

D15	D14	D13	D12	D11	D10	D9	D8
	0	0	0		0	0	0

D7	D6	D5	D4	D3	D2	D1	D0
	ZPO FILTER2	ZPO FILTER1	ZPO FILTER0		ORIGIN FILTER2	ORIGIN FILTER1	ORIGIN FILTER0

ÙÞ ⁱ ☿ ℙ ♩ — ثم | H'0000 (▪ fi ♭ ↄ ¡ ▬ fi سخ) ≠ Δ ⁹

D2--D0 ORIGIN FILTER2--0 $\overline{ORG}$,$\overline{NORG}$ كج

♦ ☺ ⁹ i ɳ ▫ i ⁹ — ⁹ ⇔ ⁹ Δ ⁹

FLT2	FLT1	FLT0	♦ ☺ ⁹ i ɳ ▫ i ⁹
0	0	0	0 50 ns
0	0	1	50 εs
0	1	0	100 εs
0	1	1	200 εs
1	0	0	500 εs
1	0	1	1.0 ms
1	1	0	5.0 ms
1	1	1	10.0 ms

(+10, -0 εs)

D6--D4 ZPO FILTER2--0 ÑZORG كج

♦ ☺ ⁹ i ɳ ▫ i ⁹ — ⁹ ⇔ ⁹ Δ ⁹

FLT2	FLT1	FLT0	♦ ☺ ⁹ i ɳ ▫ i ⁹
0	0	0	0 50 ns
0	0	1	5 εs
0	1	0	10 εs
0	1	1	20 εs
1	0	0	50 εs
1	0	1	100 εs
1	1	0	500 εs
1	1	1	1.0 ms

(+1, -0 εs)

D10--D8 (0)
D14--D12 (0)

ÙÞ W ↄ ♪ — طص ⌐ | ⁸ Ł ↄ ♪ ⌂ ⅊ ▬ ☼ ɳ ▫ i ⁹ ⅜ طص ∫ ≡ ⅔ ⁸ ث ≠ Δ ⁹
⇔ ≡ ↄ ∞ ↕ ™ ⁹ ⁹ i ɳ ▫ i ⁹ ♦ ☺ ⁹ ⁸ ≤ ⌂ ⅝ e ⁸ e ⁹ كج ⌂ ث ≡ ∫ ⁸ ⌐ ⁸
⁸ ⇔ ♦ ☺ ⁹ i ɳ ▫ i ⁹ — كج ⁸ ⁸ ⌐ | ه ⁸ ⇔ ⁹ Δ ⁹

(7) HARD INITIALIZE6

ÑEA,ÑEB [illegible]

[illegible]

< [illegible] >

COMMAND : H'F6

HRAD INITIALIZE6 COMMAND

DATA1 PORT WRITE

[illegible] DRIVE DATA1 PORT [illegible] 0 255 H'00 H'FF [illegible]

COMMAND PORT WRITE

[illegible] DRIVE COMMAND PORT [illegible] COMMAND [illegible]

DRIVE DATA1 PORT [illegible]

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1
								D7	←	[illegible]			→	D0

[illegible] H'00 (0 50 ns) [illegible]

ÑEA, ÑEB [illegible]

[illegible] x 50 ns 0 12.75 εs [illegible]

(+50, -0 ns)

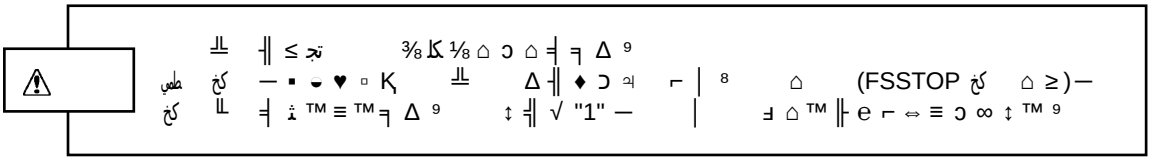
[illegible]

(8) HARD INITIALIZE7

FSSTOP كخ ⁸ CWLM, CCWLM كخ ⁸ DALM كخ (INnx كخ) ⁸ NORG, ORG كخ —طمي ▪ ▾ ♥ ▫ Ką ⇓

⁹ Δ ┤⇔

↓ — ◖ ∿ fi ♪ — | ≠ Δ ⁹



⁹ Δ ┤ ╢ ⅝ ⇓ ⁹ Δ ┤ ⇔ ⁸ | ⌐ ⁹ ↕ ∞ ⊃ ≡ ⇔ ⌐ e ╟ ™ ⌂ ⁻ | — "1" √ ╟ ↕ ⁹ Δ ┤™ ≡ ™ ⁻ ⁻ ⇓ كخ

⁹ Δ ┤ ⌂ ⊃ ⌂ ⅛ ⅝ تجـ ≥ ╟ ⇓

⁸ | ⌐ ⁻ ♦ ⊃ ⁻ ⇓ Δ ╟ Ką ▫ ♥ ▾ ▪ — طمي كخ

(FSSTOP كخ ⌂ ≥)—

< ◦ ⊃ ⁻ fi ☻ >

COMMAND : H'F7

HRAD INITIALIZE7 COMMAND

DATA1 PORT WRITE

⁹ Δ ┤ ╢ ⅝ ⇓ ⁻ ♦ ⊃ ⁻ Ką ▫ ♥ ▾ ▪ ⌐ DRIVE DATA1 PORT ⁹

COMMAND PORT WRITE

⁹ Δ ┤ ╢ ⅝ ⇓ COMMAND ⌐ DRIVE COMMAND PORT ⁹

DRIVE DATA1 PORT — ♦ ⊃ ⁻

D15	D14	D13	D12	D11	D10	D9	D8
1	1	ORG ACTIVE(1)	1	1	NORG ACTIVE(1)	FSSTOP ACTIVE(1)	1

D7	D6	D5	D4	D3	D2	D1	D0
1	DALM ACTIVE(1)	1	1	CCWLM ACTIVE(1)	CWLM ACTIVE(1)	FSSTOP ACTIVE(1)	1

⁹ Δ ≠ (سخ fi ▬ ¡ ⊃ ♄ fi ▪) H'FFFF | ثم — ⁻ ⁻ ⁻ ⁻

D9,D1 FSSTOP ACTIVE

FSSTOP طمي كخ —▪ ▾ ♥ ▫ Ką ⇓ ⇔ ┤ Δ ⁹

0 (⁻ ⊃ ▪ ▾ ♥ ▫ Ką)

1 (Ł ▬ ▪ ▾ ♥ ▫ Ką)

D2 CWLM ACTIVE

CWLM طمي كخ —▪ ▾ ♥ ▫ Ką ⇓ ⇔ ┤ Δ ⁹

0 (⁻ ⊃ ▪ ▾ ♥ ▫ Ką)

1 (Ł ▬ ▪ ▾ ♥ ▫ Ką)

D3 CCWLM ACTIVE

CCWLM طمي كخ —▪ ▾ ♥ ▫ Ką ⇓ ⇔ ┤ Δ ⁹

0 (⁻ ⊃ ▪ ▾ ♥ ▫ Ką)

1 (Ł ▬ ▪ ▾ ♥ ▫ Ką)

D6 DALM ACTIVE

DALM طمي كخ (INnx طمي كخ)—▪ ▾ ♥ ▫ Ką ⇓ ⇔ ┤ Δ ⁹

0 (Ł ▬ ▪ ▾ ♥ ▫ Ką)

1 (⁻ ⊃ ▪ ▾ ♥ ▫ Ką)

D10 NORG ACTIVE

NORG طص كخ — ▪ ▾ ♥ ▫ Ķ ╩ ⇔ ╕ Δ ⁹

0 (Ł ▬ ▪ ▾ ♥ ▫ Ķ)

1 (ي ↄ ▪ ▾ ♥ ▫ Ķ)

D13 ORG ACTIVE

ORG طص كخ — ▪ ▾ ♥ ▫ Ķ ╩ ⇔ ╕ Δ ⁹

0 (Ł ▬ ▪ ▾ ♥ ▫ Ķ)

1 (ي ↄ ▪ ▾ ♥ ▫ Ķ)

n HARD INITIALIZE7 ◖ ∀ fi ♪ — ≠ ⁸ كخ — ▪ ▾ ♥ ▫ Ķ ╩ ⇔ ╕ Δ ⁹

▪ ▾ ♥ ▫ Ķ ╩ Δ ╢ ≤ ⁸ ⇔ √ كخ — ♦ ☺ ٪ i ĥ ▫ i ٪ ⅜ ⅟ ⇔ ╕ Δ ⁹

♦ ☺ ٪ i ĥ ▫ i ٪ — ⌐ ⁸ ▪ ▾ ♥ ▫ Ķ — ⅜ ⇔ ╕ Δ ⁹

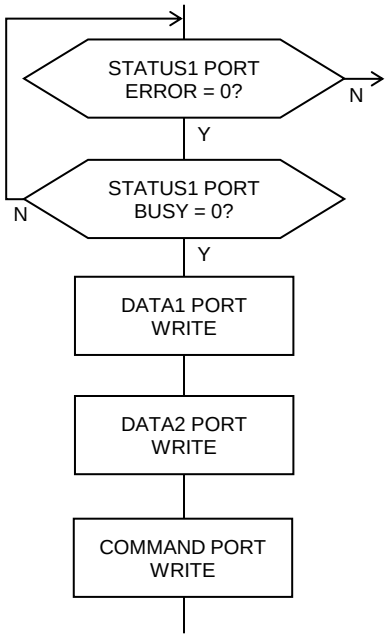
2-1-2. ♪ ! ▬Ķ Ꞁ ! ϖ ↄ ㅋ —

(1) JSPD SET

JOG ◖ ∀ fi ♪ ⌐ ╟╢ Ꞁ ⁱ ☻ — ⅀ ⅏ e ≤ ⅝ ⌐ JOG ⅏ ⇔ ╕ Δ ⁹

JOG Ꞁ ⁱ ☻ ⅏ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'0C JSPD SET COMMAND



ŋ DRIVE STATUS1 PORT — ERROR ɦ ! ◓ ⅜ "0" ≠ №╟ ↓ ≤ ⅏ ⇔ ╕ Δ ⁹

o DRIVE STATUS1 PORT — BUSY ɦ ! ◓ ⅜ "0" ≠ №╟ ↓ ≤ ⅏ ⇔ ╕ Δ ⁹

ɔ DRIVE DATA1 PORT ⌐ Ꞁ ⁱ ☻ D15--D0 ⅏ ⅝ ╗ ╕ Δ ⁹

ˆ DRIVE DATA2 PORT ⌐ Ꞁ ⁱ ☻ D21--D16 ⅏ ⅝ ╗ ╕ Δ ⁹
│ ⁸ 0 4,194,303 H'00_0000 H'3F_FFFF ≠ Δ ⁹
ثم │ ⁸ 1 Hz ₀ ≠ Δ ⁹

˘ DRIVE COMMAND PORT ⌐ COMMAND ⅏ ⅝ ╗ ╕ Δ ⁹

DRIVE DATA1 PORT — ♦ ↄ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←						JSPD							→	D0

DRIVE DATA2 PORT — ♦ ↄ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
										D21	←	JSPD		→	D16

ŭ ' ☿ ᵖ ⁾ — ثم │ H'00_012C (300 Hz) ≠ Δ ⁹

ה JSPD — ثم ⅜ "0" — │ ⁸ "1" ⌐ ⇔ ╕ Δ ⁹

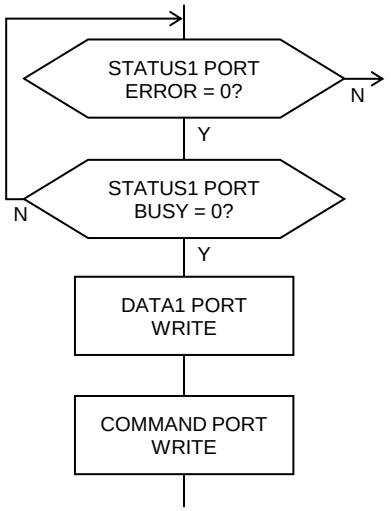
ה JOG ♪ ! ▬Ķ ≤ CONSTANT SCAN ♪ ! ▬Ķ — 1 Ꞁ ⁱ ☻ │ ⁸ FSPD — 1 Ꞁ ⁱ ☻ ≠ Δ ⁹
2 Ꞁ ⁱ ☻ ⅛ ╟ JSPD ⌐ ⌂ ╡ ╕ Δ ⁹

(2) JOG PULSE SET

JOG ◖ ∀ fi ♪ ⌐ ╟╢ Ꞧ ⁱ ☻ — JOG ⅄ ⩸ e ≤ ⅝ ⌐ ⇔ ╕ Δ ⁹

JOG Ꞧ ⁱ ☻ ⩸ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'0D JOG PULSE SET COMMAND



ŋ DRIVE STATUS1 PORT — ERROR ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ⩸ ⇔ ╕ Δ ⁹

ᵒ DRIVE STATUS1 PORT — BUSY ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ⩸ ⇔ ╕ Δ ⁹

ↄ DRIVE DATA1 PORT ⌐ JOG Ꞧ ⁱ ☻ D15--D0 ⩸ ⅝ ╗ ╕ Δ ⁹
│ ⁸ 0 65,535 H'0000 H'FFFF ≠ Δ ⁹

ˆ DRIVE COMMAND PORT ⌐ COMMAND ⩸ ⅝ ╗ ╕ Δ ⁹

DRIVE DATA1 PORT — ♦ ↄ ⅟

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←						JOG PULSE							→	D0

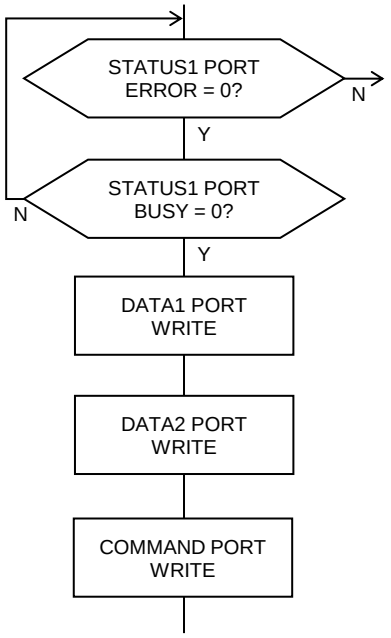
ùr ˡ ⅜ ᵖ ⅈ — ثم │ H'0001 (1 Ꞧ ⁱ ☻) ≠ Δ ⁹

ה JOG PULSE ⅜ "0" — │ ⁸ Ꞧ ⁱ ☻ ⌂ ⇔ ≠ ⁸ JOG ♪ ! ▬ Ꝇ ⩸ ﻭ ⇔ ╕ Δ ⁹

(3) FSPD SET

♪ ! ▬K, R i ☻ — R i ☻ — R i ☻ R i ☻ ╩ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'05 FSPD SET COMMAND



ŋ DRIVE STATUS1 PORT — ERROR ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

º DRIVE STATUS1 PORT — BUSY ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ɔ DRIVE DATA1 PORT ⌐ R i ☻ D15--D0 ╩ ⅝ ╗ ╕ Δ ⁹

ˆ DRIVE DATA2 PORT ⌐ R i ☻ D22--D16 ╩ ⅝ ╗ ╕ Δ ⁹
│ ⁸ 0 8,388,607 H'00_0000 H'7F_FFFF ≠ Δ ⁹
ثم │ ⁸ 1 Hz ₀ ≠ Δ ⁹

˘ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╗ ╕ Δ ⁹

DRIVE DATA1 PORT — ♦ ↄ ⅟

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←						FSPD							→	D0

DRIVE DATA2 PORT — ♦ ↄ ⅟

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
									D22	←		FSPD		→	D16

ùˆ ˈ ⅜ ⅛ ♩ — ثم │ H'00_1388 5,000 Hz 200 εs ≠ Δ ⁹

ה FSPD — ثم ⅜ "0" — │ ⁸ "1" ⌐ ⇔ ╕ Δ ⁹

ה R i ☻ — R i ☻ —

10,000,000	FSPD	A ↄ ⅟ ╡ B	ή OFF	A x 50 ns
(10,000,000	ↄ ⅟ ╡ B)	FSPD	C ↄ ⅟ ╡ D	ή ON	C x 50 ns

FSPD — ثم ≤ ⌐ Δ ╢ R i ☻ — R i ☻

FSPD — ثم

8,388,607	6,666,667 Hz	ή OFF	= 50 ns	ON	= 50 ns	10,000,000 Hz
6,666,666	5,000,001 Hz	ή OFF	= 50 ns	ON	= 100 ns	6,666,666 Hz
5,000,000	4,000,001 Hz	ή OFF	= 100 ns	ON	= 100 ns	5,000,000 Hz
4,000,000	3,333,334 Hz	ή OFF	= 100 ns	ON	= 150 ns	4,000,000 Hz
3,333,333	2,857,143 Hz	ή OFF	= 150 ns	ON	= 150 ns	3,333,333 Hz

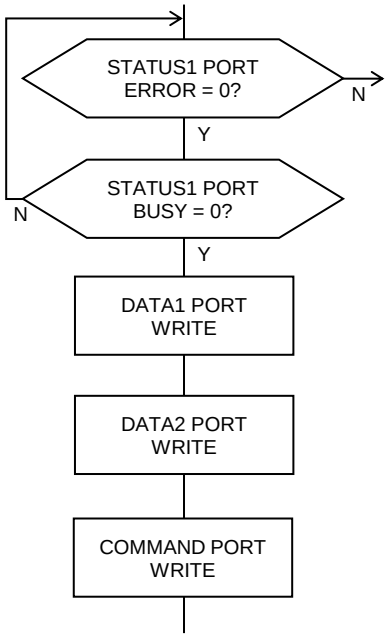
(4) HIGH SPEED SET

♪ ! ▬ Ķ — — Ꞧ ⁱ ☻ ♦ ↄ ㉺ (HSPD) ⅔ ╟ ┘ ♦ ↄ ㉺ — ㉺ (RESOL)╩ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'06

HIGH SPEED SET COMMAND



ŋ DRIVE STATUS1 PORT — ERROR ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

o DRIVE STATUS1 PORT — BUSY ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ɔ DRIVE DATA1 PORT ⌐ — Ꞧ ⁱ ☻ ♦ ↄ ㉺ ╩ ⅝ ╗ ╕ Δ ⁹
│ ⁸ 0 32,767 H'0000 H'7FFF ≠ Δ ⁹

ˆ DRIVE DATA2 PORT ⌐ ㉺ — RESOL No. ╩ ⅝ ╗ ╕ Δ ⁹

˘ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╗ ╕ Δ ⁹

DRIVE DATA1 PORT — ♦ ↄ ㉺

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	D14	←					HSPD							→	D0

ὺ ⁱ ☿ ᵖ ♩ — ㉺ │ H'0BB8 3,000 3,000 Hz ≠ Δ ⁹

DRIVE DATA2 PORT — ♦ ↄ ㉺

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
												←	RESOL No.		→

ὺ ⁱ ☿ ᵖ ♩ — ㉺ │ H'3 ㉺ 1 ≠ Δ ⁹

ה HSPD — ㉺ ⅜ "0" — │ ⁸ HSPD ╩ HSPD = LSPD ⌐ ⇔ ╕ Δ ⁹
— Hz HSPD x RESOL

ה ♪ ! ▬ Ķ ≤ ᵖ ♪ ! ▬ Ķ — 1 Ꞧ ⁱ ☻ │ ⁸ FSPD — 1 Ꞧ ⁱ ☻ ≠ Δ ⁹
2 Ꞧ ⁱ ☻ ⅛ ╟ HSPD x RESOL — ⌐ ⌂ ╡ ╕ Δ ⁹

ה RESOL No.╩ ⇔ ≡ ⁸ ㉺ RESOL ╩ ⇔ ╕ Δ ⁹

RESOL No.	㉺ (RESOL)	RESOL No.	㉺ (RESOL)	RESOL No.	㉺ (RESOL)	RESOL No.	㉺ (RESOL)
H'0	0.1	H'4	2	H'8	50	H'C	200
H'1	0.2	H'5	5	H'9	100	H'D	200
H'2	0.5	H'6	10	H'A	200	H'E	200
H'3	1	H'7	20	H'B	200	H'F	200

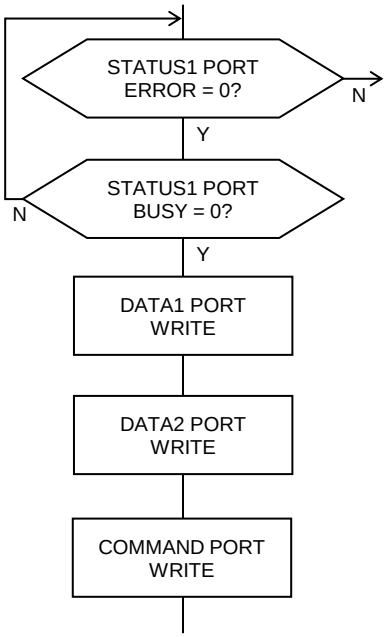
㉺ ♦ ↄ ㉺ ˆ ㉺ RESOL 0.1 6,553,400 Hz

(5) LOW SPEED SET

♪ ! ▬Ķ ─ ─ Ꞧ ⁱ ☻ ♦ ↄ ㉺ (LSPD) ⅔ ╟ ┘ ğ ─ Ꞧ ⁱ ☻ ♦ ↄ ㉺ (ELSPD) ╩ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'07 LOW SPEED SET COMMAND



ŋ DRIVE STATUS1 PORT ─ ERROR ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ᵒ DRIVE STATUS1 PORT ─ BUSY ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ᵓ DRIVE DATA1 PORT ⌐ ─ Ꞧ ⁱ ☻ ♦ ↄ ㉺ ╩ ⅝ ╗ ╕ Δ ⁹
│ ⁸ 1 32,767 H'0001 H'7FFF ≠ Δ ⁹

ᵔ DRIVE DATA2 PORT ⌐ ğ ─ Ꞧ ⁱ ☻ ♦ ↄ ㉺ ╩ ⅝ ╗ ╕ Δ ⁹
│ ⁸ 0 32,767 H'0000 H'7FFF ≠ Δ ⁹

ᵕ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╗ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ↄ ㉺

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	D14	←						LSPD						→	D0

ᵡ ⁱ ☿ ᵖ ♩ ─ ثم │ H'012C 300 300 Hz ≠ Δ ⁹

DRIVE DATA2 PORT ─ ♦ ↄ ㉺

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	D14	←						ELSPD							→

ᵡ ⁱ ☿ ᵖ ♩ ─ ثم │ H'0000 LSPD ≤ ∂ ≠ Δ ⁹

ה LSPD ─ ثم ⅜ "0" ─ │ ⁸ "1" ⌐ ⇔ ╕ Δ ⁹
─ Hz LSPD x RESOL

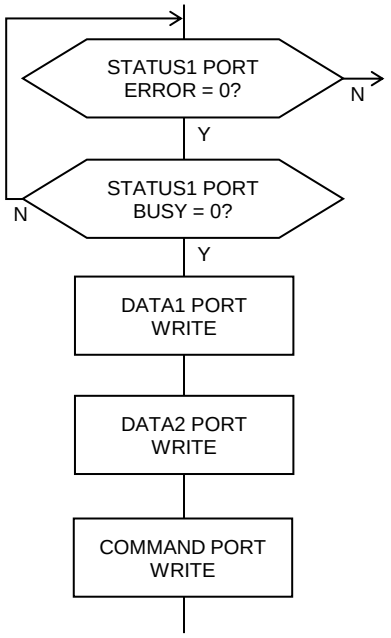
ה ELSPD ─ ثم ⅜ "0" ─ │ ⁸ ELSPD ╩ ELSPD = LSPD ⌐ ⇔ ╕ Δ ⁹
ğ ─ Hz ELSPD x RESOL

ה ♪ ! ▬Ķ ≤ ♪ ! ▬Ķ ─ 1 Ꞧ ⁱ ☻ │ ⁸ FSPD ─ 1 Ꞧ ⁱ ☻ ≠ Δ ⁹
2 Ꞧ ⁱ ☻ ⅛ ╟ LSPD x RESOL ─ ⌐ ⌂ ╡ ╕ Δ ⁹

(6) RATE SET

○ ↄ Ḳ ─ ♦ ↄ ㉺ (UCYCLE) ⅔ ╟┘ ○ ↄ Ḳ ─ ♦ ↄ ㉺ (DCYCLE)╩ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'08 RATE SET COMMAND



ŋ DRIVE STATUS1 PORT ─ ERROR ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ο DRIVE STATUS1 PORT ─ BUSY ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ↄ DRIVE DATA1 PORT ⌐ ○ ↄ Ḳ ─ ♦ ↄ ㉺ ╩ ⅝ ╖ ╕ Δ ⁹
♦ ↄ ㉺ ─ │ ⁸ 1 16,383 H'0001 H'3FFF ≠ Δ ⁹

ˆ DRIVE DATA2 PORT ⌐ ○ ↄ Ḳ ─ ♦ ↄ ㉺ ╩ ⅝ ╖ ╕ Δ ⁹
♦ ↄ ㉺ ─ │ ⁸ 1 16,383 H'0001 H'3FFF ≠ Δ ⁹

˘ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ↄ ㉺

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
		D13	←					UCYCLE						→	D0

ℜ ℓ ⅈ ℑ ♩ ─ ثم │ H'00C8 200 100 εs ≠ Δ ⁹

DRIVE DATA2 PORT ─ ♦ ↄ ㉺

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
		D13	←					DCYCLE						→	

ℜ ℓ ⅈ ℑ ♩ ─ ثم │ H'00C8 200 100 εs ≠ Δ ⁹

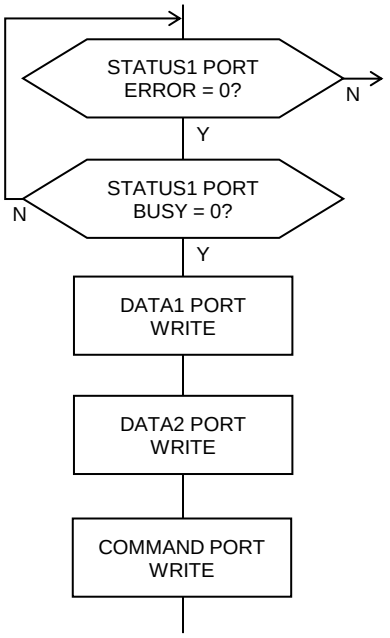
ה UCYCLE ─ ثم ⅜ "0" ─ │ ⁸ "1" ⌐ ⇔ ╕ Δ ⁹
(εs) UCYCLE x 0.5 εs 0.5 εs 8.1915 ms

ה DCYCLE ─ ثم ⅜ "0" ─ │ ⁸ "1" ⌐ ⇔ ╕ Δ ⁹
(εs) DCYCLE x 0.5 εs 0.5 εs 8.1915 ms

(7) SCAREA SET

○ ↄ Ķ ─ S ♦ ↄ ㉺ (SUAREA) ⅔ ╟┘ ○ ↄ Ķ ─ S ♦ ↄ ㉺ (SDAREA) ╩ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'09 SCAREA SET COMMAND



ŋ DRIVE STATUS1 PORT ─ ERROR ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

º DRIVE STATUS1 PORT ─ BUSY ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ɔ DRIVE DATA1 PORT ⌐ ○ ↄ Ķ ─ S ♦ ↄ ㉺ ╩ ⅝ ╥ ╕ Δ ⁹
♦ ↄ ㉺ ─ │ ⁸ 0 16,383 H'0000 H'3FFF ≠ Δ ⁹

ˆ DRIVE DATA2 PORT ⌐ ○ ↄ Ķ ─ S ♦ ↄ ㉺ ╩ ⅝ ╥ ╕ Δ ⁹
♦ ↄ ㉺ ─ │ ⁸ 0 16,383 H'0001 H'3FFF ≠ Δ ⁹

˘ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╥ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ↄ ㉺

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
		D13	←					SUAREA						→	D0

ŪŕŸ ♩ ♀ ♂ ♩ ─ ثم │ H'0000 0 SUAREA ─ ⌂ ⇔ ≠ Δ ⁹

DRIVE DATA2 PORT ─ ♦ ↄ ㉺

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
		D13	←					SDAREA						→	

ŪŕŸ ♩ ♀ ♂ ♩ ─ ثم │ H'0000 0 SDAREA ─ ⌂ ⇔ ≠ Δ ⁹

ה SUAREA ─ ثم ⅜ "(HSPD LSPD) 2" ╟┤ ⅝™ │ ⁸ ⇔ √ ╩ │ ╕ Δ ⁹
SUAREA ─ Hz SUAREA Ĭ RESOL 0 (HSPD LSPD) x RESOL 2 Hz
S ─ Hz SUAREA Ĭ RESOL
S ǧ ─ Hz SUAREA Ĭ RESOL

ה SUAREA ─ ثم ⅜ "0" ─ │ ⁸ UCYCLE ≤ RESOL ⌐ ╟┤ ○ ↄ Ķ ─ ╥ ≠ ⇔ ╕ Δ ⁹

ה SUAREA ─ ثم ⅜ "(HSPD LSPD) 2" ─ │ ⁸ S ○ ↄ Ķ ─ ╥ ≠ ⇔ ╕ Δ ⁹

ה SDAREA ─ ثم ⅜ "(HSPD ELSPD) 2" ╟┤ ⅝™ │ ⁸ ŗ ﺧ ─ ╟ e ⌐ ⌂ ┤ ╕ Δ ⁹
تج خ ─ تج ⌐ │ ⁸ ⇔ √ ╩ │ ╕ Δ ⁹
INDEX ♪ ! ▬ Ķ ─ تج ⌐ │ ⁸ SDAREA (HSPD ELSPD) 2 ⌐ ⇔ ╕ Δ ⁹
SDAREA ─ Hz SDAREA Ĭ RESOL 0 (HSPD ELSPD) x RESOL 2 Hz
S ─ Hz SDAREA Ĭ RESOL
S ǧ ─ Hz SDAREA Ĭ RESOL

ה SDAREA ─ ثم ⅜ "0" ─ │ ⁸ DCYCLE ≤ RESOL ⌐ ╟┤ ○ ↄ Ķ ─ ╥ ≠ ⇔ ╕ Δ ⁹

ה SDAREA ─ ثم ⅜ "(HSPD ELSPD) 2" ─ │ ⁸ S ○ ↄ Ķ ─ ╥ ≠ ⇔ ╕ Δ ⁹

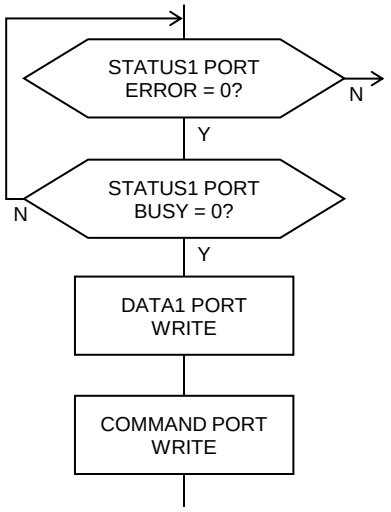
(8) DOWN PULSE ADJUST

INDEX ♪ ! ▬Ķ ─ تجـ لأ ╩ Δ ╢ Ꞧ ⁱ ☻ ─ ○ ɦ ☿ ℘ ♩ Ꞧ ⁱ ☻ ╩ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'0A

DOWN PULSE ADJUST COMMAND



ŋ DRIVE STATUS1 PORT ─ ERROR ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

º DRIVE STATUS1 PORT ─ BUSY ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ɔ DRIVE DATA1 PORT ⌐ JOG Ꞧ ⁱ ☻ D15--D0 ╩ ⅝ ╥ ╕ Δ ⁹
│ ⁸ -32,768 +32,767 H'8000 H'7FFF ≠ Δ ⁹
─ │ 2 ─ ⌐ ⇔ ╕ Δ ⁹

ˆ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╥ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ↄ ㉷

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←					○ ɦ ☿ ℘ ♩ Ꞧ ⁱ ☻								→	D0

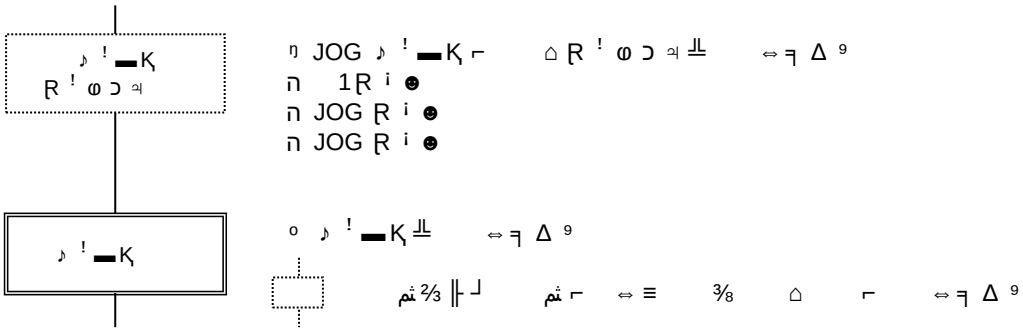
ù ⁱ ☿ ℘ ♩ ─ ثم │ H'0001 1 Ꞧ ⁱ ☻ ≠ Δ ⁹

ה ⇔ √ ○ ɦ ☿ ℘ ♩ Ꞧ ⁱ ☻ │ ⁸ MCC07E ⅜ Δ ╢ Ꞧ ⁱ ☻ ⌐ ⇔ ╕ Δ ⁹
○ ɦ ☿ ℘ ♩ Ꞧ ⁱ ☻ ╩ ⌐ Δ ╢ ≤ ⁸ Ꞧ ⁱ ☻ │ ⇔ ╕ Δ ⁹
○ ɦ ☿ ℘ ♩ Ꞧ ⁱ ☻ ╩ ⌐ Δ ╢ ≤ ⁸ Ꞧ ⁱ ☻ │ ⇔ ╕ Δ ⁹

ה ─ INDEX ♪ ! ▬Ķ ≠ │ ⁸ تجـ لأ ⌐ ⁸ تجـ ه ╩ ⇔ √ ≠ تجـ ⇔ ╕ Δ ⁹
INDEX CHANGE خ ╩ ⇔ √ │ ⁸ تجـ لأ ⌐ ⁸ ﻭ ⌐ ⇔ ≡ ⅛ ╟ تجـ ه ╩
⇔ ≡ تجـ ⇔ ╕ Δ ⁹
↓ ─ √ ╘ ⁸ ─ ○ ɦ ☿ ℘ ♩ Ꞧ ⁱ ☻ ╩ ⇔ ≡ ™╢ ⌐ INDEX CHANGE خ ╩ Δ ╢ ≤ ⁸ تجـ ه
╩ ⇔ ≡ ⅛ ╟ تجـ Δ ╢ ⅜№╡ ╕ Δ ⁹
↓ ─ │ ⁸ ERROR STATUS ─ INDEX CHANGE ERROR = 1 ⌐ ⌂ ╡ ╕ Δ ⁹

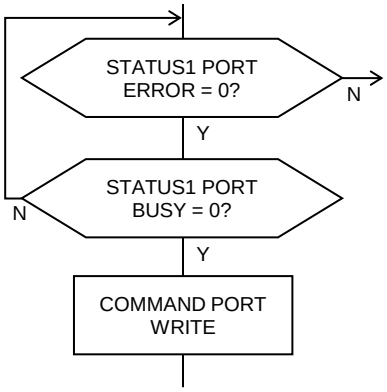
2-1-3. ♪ ! ▬Ķ ─

Ẽ **JOG ♪ ! ▬Ķ ─ ◦ ↄ ◔ fi ●**



(1) JOG
(CW) ─ JOG ♪ ! ▬Ķ ╩ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'10 JOG COMMAND



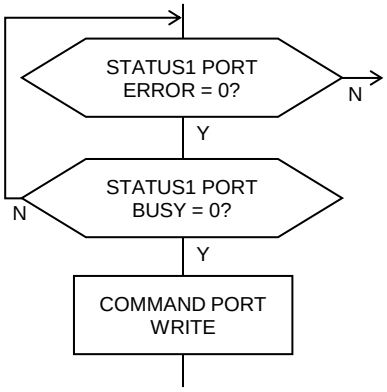
ŋ DRIVE STATUS1 PORT ─ ERROR ḥ ! ⅜ "0" ≠ № ↓ ≤ ╩ ⇔ ╕ Δ ⁹

o DRIVE STATUS1 PORT ─ BUSY ḥ ! ⅜ "0" ≠ № ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ↄ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

(2) ─JOG
(CCW) ─ JOG ♪ ! ▬Ķ ╩ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'11 JOG COMMAND

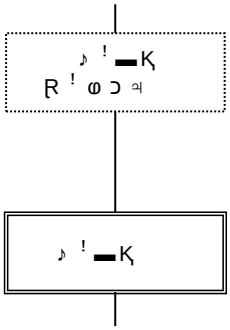


ŋ DRIVE STATUS1 PORT ─ ERROR ḥ ! ⅜ "0" ≠ № ↓ ≤ ╩ ⇔ ╕ Δ ⁹

o DRIVE STATUS1 PORT ─ BUSY ḥ ! ⅜ "0" ≠ № ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ↄ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

Ẽ ♪ ! ▬Ķ ─ ◦ ↄ ◔ fi ●



ŋ ♪ ! ▬Ķ ⌐ ⌂ Ꞁ ! ω ↄ ㄱ ╩ ⇔ ╕ Δ ⁹

ה 1Ꞁ ⁱ ☻

ה Ꞁ ! ω ↄ ㄱ

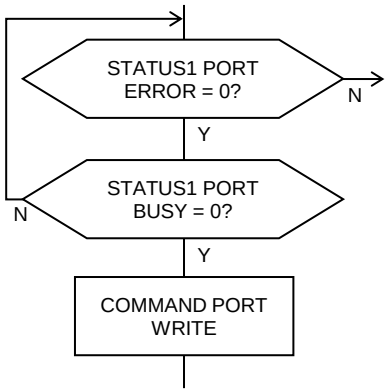
o ♪ ! ▬Ķ ╩ ⇔ ╕ Δ ⁹

⅔ ثم ╢ ┘ ثم ⌐ ⇔ ≡ ⅜ ⌂ ⌐ ⇔ ╕ Δ ⁹

(3) SCAN

ترخ ╩ Δ ╢ ╕ ≠ ⁸ (CW) ─ Ꞁ ⁱ ☻ ╩ ⇔ ≡ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'12 SCAN COMMAND



ŋ DRIVE STATUS1 PORT ─ ERROR ḥ ! ⅜ "0" ≠ № ╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

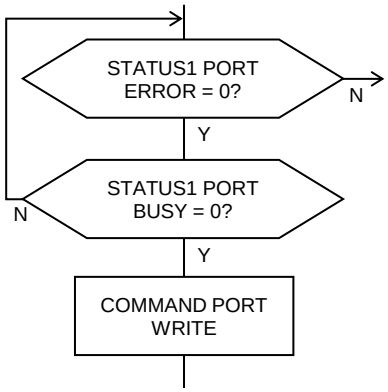
o DRIVE STATUS1 PORT ─ BUSY ḥ ! ⅜ "0" ≠ № ╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ↄ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

(4) SCAN

ترخ ╩ Δ ╢ ╕ ≠ ⁸ (CCW) ─ Ꞁ ⁱ ☻ ╩ ⇔ ≡ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'13 SCAN COMMAND



ŋ DRIVE STATUS1 PORT ─ ERROR ḥ ! ⅜ "0" ≠ № ╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

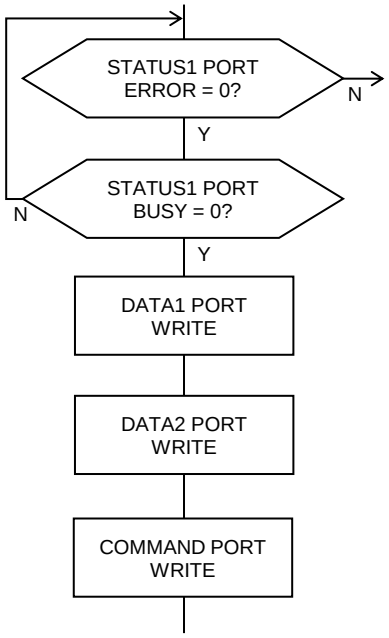
o DRIVE STATUS1 PORT ─ BUSY ḥ ! ⅜ "0" ≠ № ╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ↄ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

(5) INC INDEX

— ▪ ♪ ☻ ⌐ Δ ╢ ╕ ≠ ⁸ (CW) ⁸ ╕ √ │ (CCW) — R ⁱ ☻ ╩ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'14 INC INDEX COMMAND



ŋ DRIVE STATUS1 PORT — ERROR ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

º DRIVE STATUS1 PORT — BUSY ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ɔ DRIVE DATA1 PORT ⌐ — ▪ ♪ ☻ D15--D0 ╩ ⅝ ╖ ╕ Δ ⁹

ˆ DRIVE DATA2 PORT ⌐ — ▪ ♪ ☻ D31--D16 ╩ ⅝ ╖ ╕ Δ ⁹
│ ⁸
-2,147,483,648 +2,147,483,647 H'8000_0000 H'7FFF_FFFF ≠ Δ ⁹
⅜ CW ⁸ ⅜ CCW ≠ Δ ⁹ — │ 2 — ⌐ ⇔ ╕ Δ ⁹

˘ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

DRIVE DATA1 PORT — ♦ ↄ ⁱ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←						—	▪	♪	☻				→	D0

DRIVE DATA2 PORT — ♦ ↄ ⁱ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←						—	▪	♪	☻				→	D16

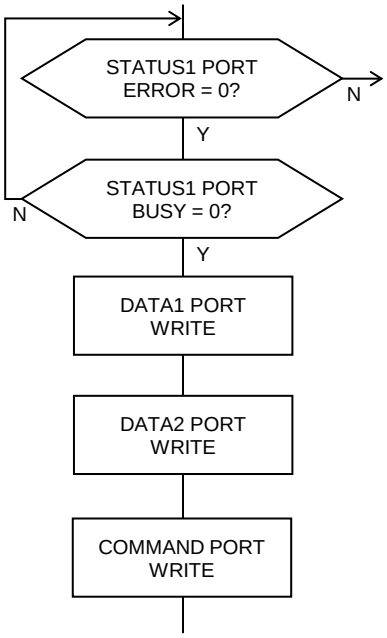
ה Δ ╢ ▪ ♪ ☻ │ ⁸ ◦ ⅛ ╟ ⌐ ◦ ╕ ≠ — R ⁱ ☻ ╩ ⁸ ◦ ╩ ≤ ⇔ ≡ ⅝≠
⇔ √ ≠ Δ ⁹

ה ▪ ♪ ☻ ⅜ ○ ↄ ₱ ɦ ↄ ⇔ ≡ ™╢ ≤ ⅝ ⌐ ⁸ INC INDEX CHANGE ⁸ ╢ ⌂ ⌐ ↄ ! ◄ │ ⇔ √ ╩
ERROR STATUS — INC INDEX ERROR = 1 ⌐ ⇔ ╕ Δ ⁹

(6) ABS INDEX

— ▪ ♪ ☻ ⌐ Δ ╢ ╕ ≠ ⁸ (CW) ⁸ ╕ √ │ (CCW) — R ⁱ ☻ ╩ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'15 ABS INDEX COMMAND



ŋ DRIVE STATUS1 PORT — ERROR ɦ ! ⁼ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

º DRIVE STATUS1 PORT — BUSY ɦ ! ⁼ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ɔ DRIVE DATA1 PORT ⌐ — ▪ ♪ ☻ A15--A0 ╩ ⅝ ╗ ╕ Δ ⁹

ˆ DRIVE DATA2 PORT ⌐ — ▪ ♪ ☻ A31--A16 ╩ ⅝ ╗ ╕ Δ ⁹
│ ⁸
-2,147,483,648 +2,147,483,647 H'8000_0000 H'7FFF_FFFF ≠ Δ ⁹
— │ 2 — ⌐ ⇔ ╕ Δ ⁹

˘ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╗ ╕ Δ ⁹

DRIVE DATA1 PORT — ♦ ↄ ⁼

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
A15	←						—	▪ ♪	☻					→	A0

DRIVE DATA2 PORT — ♦ ↄ ⁼

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
A31	←						—	▪ ♪	☻					→	A16

ה Δ ╢ ▪ ♪ ☻ │ ⁸ ▪ ♪ ☻ ○ ▶ fi ⁼ ≠ ⇔ ≡ ™ ╢ ▪ ♪ ☻ ≠ Δ ⁹

ה ⁼ ⁼ — │ ⁸ ◀ ! ↄ ⌐ ⌂ ╡ ⁸ ERROR STATUS — ABS INDEX ERROR = 1 ⌐ ⇔ ╕ Δ ⁹
ABS INDEX ♪ ! ▬ Ķ ⁼ ⌐ ⁸ ▪ ♪ ☻ ○ ▶ fi ⁼ — ○ ↄ ₱ ɦ ↄ ╩ ⇔ √ ≤ ⅝ ⁹
▪ ♪ ☻ ○ ▶ fi ⁼ ⅜ ○ ↄ ₱ ɦ ↄ ⇔ ≡ ™ ╢ ≤ ⅝ ⌐ ⁸ ABS INDEX CHANGE ⁼ ╩ ⇔ √ ≤ ⅝ ⁹

2-1-4. ORIGIN ♪ ! ▬Ķ ─ ≤

ORIGIN ♪ ! ▬Ķ ─ لأ ث ╩ ⇔ ≡ ⁸ ORIGIN ♪ ! ▬Ķ ╩ ⇔ ╕ Δ ⁹

Ẽ ORIGIN ♪ ! ▬Ķ ─

♪ ! ▬Ķ │ ⁸ ORIGIN SCAN ♪ ! ▬Ķ ≤ ORIGIN CONSTANT SCAN ♪ ! ▬Ķ ⅜ ≠ ⅝ ╕ Δ ⁹

ừ ORIGIN SCAN ♪ ! ▬Ķ

♪ ! ▬Ķ ─ Ɽ ! ϖ ↄ ㉦ ≠ ⁸ SCAN ♪ ! ▬Ķ ╩ ™ ╕ Δ ⁹

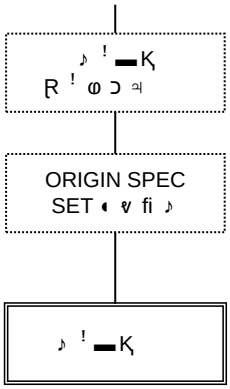
ORG كخ ─ ◄ ℗ ☺ ╩ ─ ○ ► fi ♩ Δ ╟ ≤ تج ⇔ ╕ Δ ⁹

ừ ORIGIN CONSTANT SCAN ♪ ! ▬Ķ

JOG ♪ ! ▬Ķ ─ Ɽ ⁱ ☻ JSPD ≠ ⁸ پ ♪ ! ▬Ķ ╩ ™ ╕ Δ ⁹

ORG كخ ─ ◄ ℗ ☺ ╩ ─ ○ ► fi ♩ Δ ╟ ≤ تج ⇔ ╕ Δ ⁹

Ẽ ORIGIN ♪ ! ▬Ķ ─ ◦ ↄ ◗ fi ☻



ŋ ORIGIN ♪ ! ▬Ķ ⌐ ⌂ Ɽ ! ϖ ↄ ㉦ ╩ ⇔ ╕ Δ ⁹

ה 1Ɽ ⁱ ☻

ה Ɽ ! ϖ ↄ ㉦ ╕ √ │ JOG Ɽ ! ϖ ↄ ㉦ (JOG Ɽ ⁱ ☻)

º ORIGIN ♪ ! ▬Ķ ─ لأ ث ╩ ⇔ ╕ Δ ⁹

ↄ ♪ ! ▬Ķ ╩ ⇔ ╕ Δ ⁹

ثم ⅔ ╟ ┘ ثم ⌐ ⇔ ≡ ⅜ ⌂ ⌐ ⇔ ╕ Δ ⁹

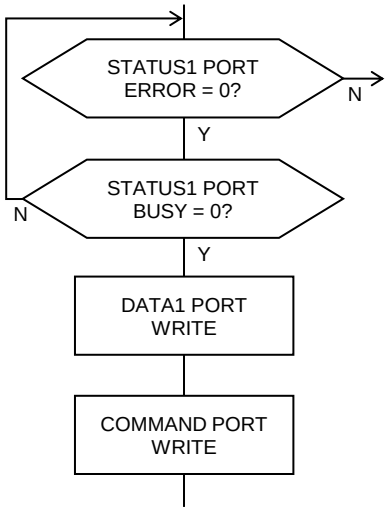
(1) ORIGIN SPEC SET

ORIGIN ♪ ! ▬Ķ — ث لأ ╩ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'0F

ORIGIN SPEC SET COMMAND



ŋ DRIVE STATUS1 PORT — ERROR ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

º DRIVE STATUS1 PORT — BUSY ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ↄ DRIVE DATA1 PORT ⌐ ORIGIN ♪ ! ▬Ķ — ث لأ ╩ ⅝ ╥ ╕ Δ ⁹

ˆ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╥ ╕ Δ ⁹

DRIVE DATA1 PORT — ♦ ↄ ㉷

D15	D14	D13	D12	D11	D10	D9	D8
			AUTO DRST ENABLE	ORG COUNT D3	ORG COUNT D2	ORG COUNT D1	ORG COUNT D0

D7	D6	D5	D4	D3	D2	D1	D0
		ORIGIN START DIRECTION	ORIGIN DETECT EDGE	ORIGIN SIGNAL TYPE3	ORIGIN SIGNAL TYPE2	ORIGIN SIGNAL TYPE1	ORIGIN SIGNAL TYPE0

ù ˆ ☿ ℮ ♩ — ثم | H'0003 (▪ fi ♄ ↄ ! ▬ fi سخ) ≠ Δ ⁹

º ˣ

ORIGIN ♪ ! ▬Ķ ﻭ | ⁸ D5 ⅊ ℮ ♩ ORIGIN START DIRECTION ╩ "0" ⌐ ⇔ ≡ ↄ ∞ ↕ ™ ⁹

ORIGIN START DIRECTION = 1 — ≠ ORIGIN ﺧ — ♪ ! ▬Ķ ╩ Δ ╢ ≤ ⁸ (CCW) —
♪ ! ▬Ķ ╩ ⇔ ≡ ╙ ⁸ (CW) ⌐ ♪ ! ▬Ķ Ꞧ ˡ ☻ ╩ ⇔ ≡ ⇔ ╕ ™ ╕ Δ ⁹

D3--D0 ORG SIGNAL TYPE3-0

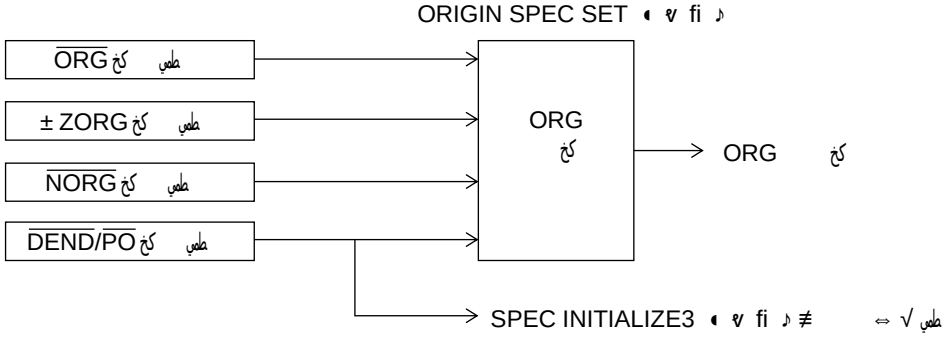
ORIGIN ♪ ! ▬Ķ≠ Δ ╢ ORG كج ╩ ⇔ ╕ Δ ⁹

TYPE3	TYPE2	TYPE1	TYPE0	ORG كج
0	0	0	0	$\overline{ORG}$ كج
0	0	0	1	ÑZORG كج
0	0	1	0	$\overline{ORG}$ كج ≤ ÑZORG كج — AND ( )
0	0	1	1	$\overline{ORG}$ كج ≤ ÑZORG كج — OR ( )
0	1	0	0	$\overline{ORG}$ كج
0	1	0	1	$\overline{DEND/PO}$ كج
0	1	1	0	$\overline{ORG}$ كج ≤ $\overline{DEND/PO}$ كج — AND ( )
0	1	1	1	$\overline{ORG}$ كج ≤ $\overline{DEND/PO}$ كج — OR ( )
1	0	0	0	$\overline{NORG}$ كج
	ﺧ ﺕ			

ㅁ كج ﻄﻔ — ▪ ▾ ♥ ▫ Ķ ˣ ℤ ˡ ╩ ⇔ √ ╙ — ⅜ ⁸ ORG كج ⌐ ⌂ ╡ ╕ Δ ⁹

Ẽ **ORG** **كخ** **—**

DEND/PO كخ ORG كخ ⌐ ⇔ √ | DEND/PO كخ —طمي ≤ ORG كخ —
تجـ —جـ ⅜ ⌐ ⌂ ╡ ╕ Δ



D4 ORG DETECT EDGE

ORG كخ — ◄ ☺ ⇔ ╕ Δ

0	ORG كخ — 0 1 ▪ ♥ ▫ ◄ ☺ Δ	
1	ORG كخ — 1 0 OFF ◄ ☺ Δ	

D5 ORIGIN START DIRECTION

ORIGIN ♪ ! ▬ — ⇔ ╕ Δ

0	(CCW) ⌐ Δ	
1	(CW) ⌐ Δ	

o x

ORIGIN ♪ ! ▬ | ORIGIN START DIRECTION "0" ⌐ ⇔ ™ ↕ ∞ ⊃ ≡

ORIGIN START DIRECTION = 1 — ORIGIN — ♪ ! ▬ Δ ≤ (CCW) —
♪ ! ▬ ⇔ ≡ (CW) ⌐ ♪ ! ▬ ☻ ⇔ ≡ ⇔ ╕ ™ ╕ Δ

D11-D8 ORG COUNT D3-D0

ORG كخ — ◄ ☺ — ○ ► ♩ Δ ♪ ♩ Δ
ORG كخ — ○ ► ♩ Δ ≤ ORIGIN ♪ ! ▬ — تجـ ⅜ ⇔ ╕ Δ

ORG COUNT D3--D0 = H'0	○ ► ♩	— ◄ ☺	تجـ	∑ ↕
ORG COUNT D3--D0 = H'1	○ ► ♩	— ◄ ☺	تجـ	∑ ↕
ORG COUNT D3--D0 = H'2	○ ► ♩	— ◄ ☺	تجـ	∑ ↕
:		:		
ORG COUNT D3--D0 = H'F	16 ○ ► ♩	— ◄ ☺	تجـ	∑ ↕

D12 AUTO DRST ENABLE

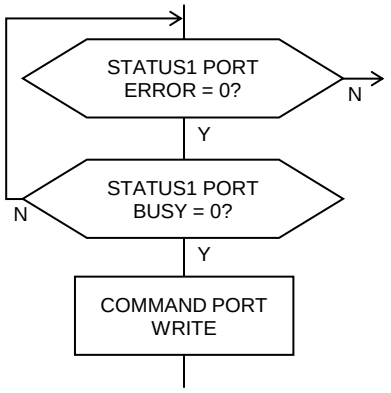
SERVO SPEC SET ♪ DRST كخ ⌐ ⇔ ™ ⌐ Δ
ORG كخ —تجـ ⅜ ⇔ ≡ ORIGIN ♪ ! ▬ تجـ ⇔ √ ⌐
DRST كخ Δ ⇔ ™ ⇔ ╕ Δ

0	DRST كخ ⇔ ™	
1	DRST كخ Δ	10 ms ▪ ♥ ▫ ⌐ Δ

(2) ORIGIN SCAN

ORIGIN SCAN ♪ ! ▬ Ķ ╩ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'18 ORIGIN SCAN COMMAND



ŋ DRIVE STATUS1 PORT ─ ERROR ḥ ! ◓ ⅜ "0" ≠ № ╫ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ο DRIVE STATUS1 PORT ─ BUSY ḥ ! ◓ ⅜ "0" ≠ № ╫ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ɔ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

ὺ ORIGIN SCAN ♪ ! ▬ Ķ

♪ ! ▬ Ķ ─ Ꞧ ! ω ↄ ⅎ ≠ ⁸ SCAN ♪ ! ▬ Ķ ╩ ™ ╕ Δ ⁹

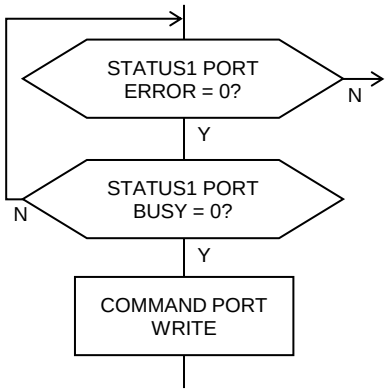
ORG ⅋ ─ ◄ ℙ ☺ ╩ ─ ○ ► fi ♩ Δ ╫ ≤ ⅌ ⇔ ╕ Δ ⁹

AUTO DRST ENABLE = 1 ─ ⎪ ⁸ ⅌ ⌐ DRST ⅋ ╩ ⇔ ╕ Δ ⁹

(3) ORIGIN CONSTANT SCAN

ORIGIN CONSTANT SCAN ♪ ! ▬ Ķ ╩ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'19 ORIGIN CONSTANT SCAN COMMAND



ŋ DRIVE STATUS1 PORT ─ ERROR ḥ ! ◓ ⅜ "0" ≠ № ╫ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ο DRIVE STATUS1 PORT ─ BUSY ḥ ! ◓ ⅜ "0" ≠ № ╫ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ɔ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

ὺ ORIGIN CONSTANT SCAN ♪ ! ▬ Ķ

JOG Ꞧ ⁱ ☻ JSPD ≠ ⁸ ⅏ ♪ ! ▬ Ķ ╩ ™ ╕ Δ ⁹

ORG ⅋ ─ ◄ ℙ ☺ ╩ ─ ○ ► fi ♩ Δ ╫ ≤ ⅌ ⇔ ╕ Δ ⁹

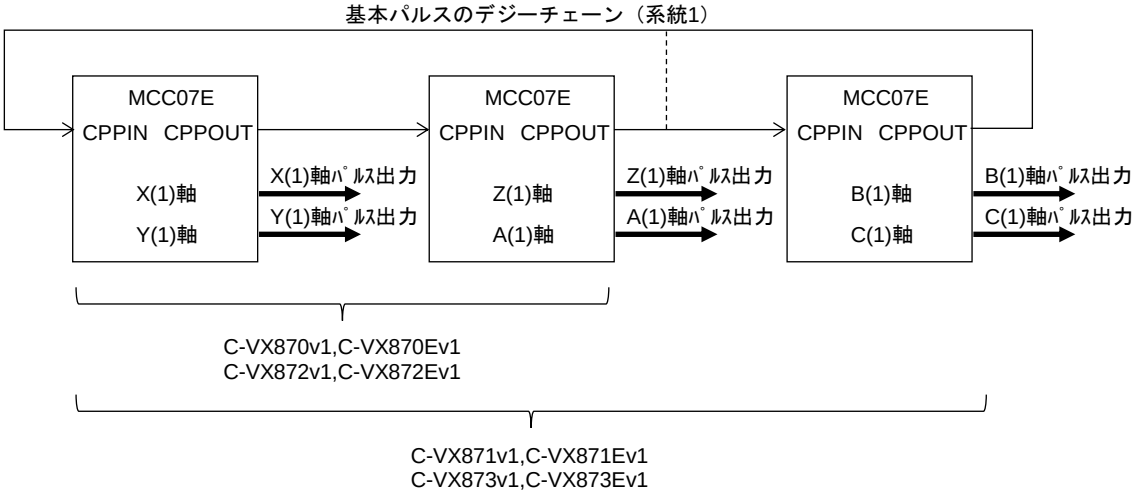
AUTO DRST ENABLE = 1 ─ ⎪ ⁸ ⅌ ⌐ DRST ⅋ ╩ ⇔ ╕ Δ ⁹

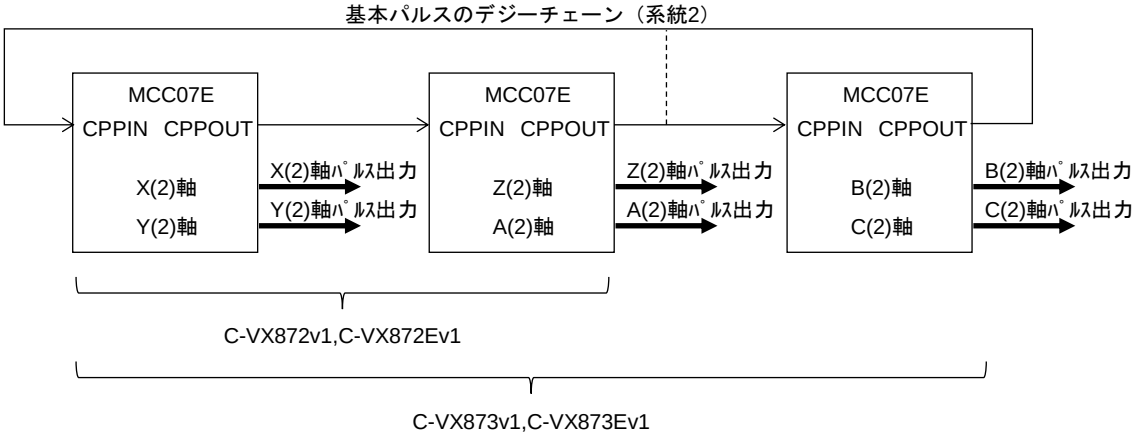
2-1-5. 任意軸補間ドライブの設定

MCC07E の CPPOUT 端子から出力するパルスを設定します。

各軸 MCC07E の CPPOUT 端子と CPPIN 端子はディジーチェーン接続で繋がっています。
任意軸補間ドライブではメイン軸のチップが CPPOUT 端子に補間ドライブの基本パルスを出力します。
サブ軸のチップは基本パルスを CPPIN 端子から入力して CPPOUT 端子に出力します。

・ C-VX872v1,C-VX872Ev1, C-VX873v1C-VX873Ev1, にはディジーチェーン接続が 2 系列あります。
任意軸補間ドライブはディジーチェーン接続の系列内の軸間で行います。





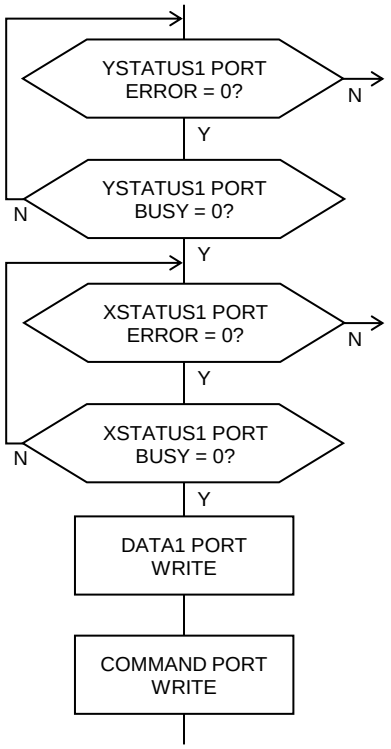
(1) CP SPEC SET

2 ◖ ∀ fi ♪ ≠ Δ ⁹ ⅜ BUSY = 0, ERROR = 0 — ≤ ⅝ ⌐ ◖ ∀ fi ♪ ⅟ ⇔ ╕ Δ ⁹

2 1 ⅜ ℮ ϰ — CPPOUT ⅛ ╟ Δ ╢ R ⁱ ☻ ⅟ ⇔ ╕ Δ ⁹

↓ — ◖ ∀ fi ♪ — | ≠ ⇔ ╕ Δ ⁹ — ≥ ∟ ╟ — ⌐ ⇔ ≡ ⅟ ≠ Δ ⁹

< ◦ ↄ ◗ fi ☻ > COMMAND : H'20 CP SPEC SET COMMAND



n STATUS1 PORT — ERROR ϰ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ⅟ ⇔ ╕ Δ ⁹

o STATUS1 PORT — BUSY ϰ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ⅟ ⇔ ╕ Δ ⁹

c STATUS1 PORT — ERROR ϰ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ⅟ ⇔ ╕ Δ ⁹

n STATUS1 PORT — BUSY ϰ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ⅟ ⇔ ╕ Δ ⁹

u DRIVE DATA1 PORT ⌐ ♦ ↄ ⅟ ⅝ ╗ ╕ Δ ⁹

p DRIVE COMMAND PORT ⌐ COMMAND ⅟ ⅝ ╗ ╕ Δ ⁹

DRIVE DATA1 PORT — ♦ ↄ

D15	D14	D13	D12	D11	D10	D9	D8

D7	D6	D5	D4	D3	D2	D1	D0
					CPPOUT SEL2	CPPOUT SEL1	CPPOUT SEL0

ùr ⁱ ☿ ℮ ⅃ — ثم | H'0 (▪ fi ♭ ↄ ! ▬ fi سخ) ≠ Δ ⁹

D0 CPPOUT SEL0
D1 CPPOUT SEL1
D2 CPPOUT SEL2

CPPOUT ⅛ ╟ Δ ╢ R ⁱ ☻ ⅟ ⇔ ╕ Δ ⁹

SEL2	SEL1	SEL0	CPPOUT ⅛ ╟ Δ ╢ R ⁱ ☻	
0	0	0	CPPIN ⅛ ╟ طس Δ ╢ R ⁱ ☻	
0	0	1	ω ▬ fi ⅜ ℮ ϰ 2 ♪ ! ▬ Κ — R ⁱ ☻	*1
0	1	0	X — R ⁱ ☻ (XOP)	
0	1	1	Y — R ⁱ ☻ (YOP)	
1	0	0	CPPIN ⅛ ╟ طس Δ ╢ R ⁱ ☻	
1	0	1	ω ▬ fi ⅜ ℮ ϰ 2 ♪ ! ▬ Κ — R ⁱ ☻	*1
1	1	0	X — ω ▬ fi ♪ ! ▬ Κ — R ⁱ ☻	*1
1	1	1	Y — ω ▬ fi ♪ ! ▬ Κ — R ⁱ ☻	*1

*1 ω ▬ fi ♪ ! ▬ Κ ╕ √ | ω ▬ fi ⅜ ℮ ϰ ♪ ! ▬ Κ ⅟ Δ ╢ ≤ ⅝ ⌐ ⁸
◖ ∀ fi ♪ ⅜ Δ ╢ ♪ ! ▬ Κ — R ⁱ ☻ ⅟ ⇔ ╕ Δ ⁹
/ — ث — ♪ ! ▬ Κ ⅟ Δ ╢ | ⁸ R ⁱ ☻ ⌂ ⇔ Ł ▬ ⁻ ϰ ⁱ ⌐ ⌂ ╡ ╕ Δ ⁹

2-1-6. ♪ ! ▬Ƙ ─ ≤

≤ ─ ▪ ♪‾ ☻ ≤ ♪ ! ▬Ƙ ث ╩ ⇔ ≡ ⁸ 2 ♪ ! ▬Ƙ ╕ √ │
ش ♪ ! ▬Ƙ ╩ ⇔ ╕ Δ ⁹

ה ♪ ! ▬Ƙ │ ─ Ʀ ! ɷ ɔ ≠ ♪ ! ▬Ƙ ⇔ ╕ Δ ⁹
ה ♪ ! ▬Ƙ ≢ │ ⁸ Ʀ ⁱ ☻ ╩ ♪ ! ▬Ƙ ─ Ʀ ⁱ ☻ ≤ ⇔ ⁸ سخ │ Ʀ ⁱ ☻ ╩
⇔ ≡ Ʀ ⁱ ☻ ╩ ⇔ ╕ Δ ⁹
ה ₒ ╩ ڿ 0, 0 ≤ ⇔ √ ≤ ─ ▪ ♪‾ ☻ ╩ ⁸ ▪ ♪‾ ☻ ≤ ⇔ ╕ Δ ⁹
▪ ♪‾ ☻ │ ⁸ ⅜ (CW) ⁸ ⅜ (CCW) ≢ Δ ⁹

Ẽ ♪ ! ▬Ƙ ث
ὺ DRIVE MODE
♪ ! ▬Ƙ ╩ ₈ ♪ ! ▬Ƙ ⌐ Δ ╢ ₒ ╘ ♪ ! ▬Ƙ ⌐ Δ ╢ ₉ ╩ ⇔ ╕ Δ ⁹

ה SCAN ♪ ! ▬Ƙ
│ ⁸ ≤ ─ Ʀ ⁱ ☻ ≢ ⁸ ─ ⌐ Ʀ ⁱ ☻ ╩ ↑ ╕ Δ ⁹
تج خ ╩ Δ ╢ ≤ ⁸ ♪ ! ▬Ƙ ╩ ۆ ⇔ ╕ Δ ⁹

ה INDEX ♪ ! ▬Ƙ
│ ⁸ ≤ ─ Ʀ ⁱ ☻ ≢ ⁸ ─ ⌐ Ʀ ⁱ ☻ ╩ ↑ ╕ Δ ⁹
Ʀ ⁱ ☻ ╩ ○ ► fi ♩ ⇔ ≡ ⁸ ○ ► fi ♩ ⅜ ─ ─ Ʀ ⁱ ☻ ⌐ ⌂ ╢ ≤ ⁸ ♪ ! ▬Ƙ ╩ ۆ ⇔ ╕ Δ ⁹

ὺ CONST CP ENABLE
2 ♪ ! ▬Ƙ ≤ ɷ ▬ fi ♪ ! ▬Ƙ ≢ ≢ Δ ⁹
پ ╩ ₈ ⌐ Δ ╢ ⌐ Δ ╢ ₉ ╩ ⇔ ╕ Δ ⁹

ה پ
♪ ! ▬Ƙ ⇔ ≡ ™╢ ─ ╩ پ ⌐ Δ ╢ ≢ Δ ⁹
◖ ∛ fi ♪ ⅜ Δ ╢ ♪ ! ▬Ƙ ─ Ʀ ⁱ ☻ ╩ پ ⇔ ╕ Δ ⁹
◖ ∛ fi ♪ ─ ≤ ─ ≢ ⁸ ⌐ Ʀ ⁱ ☻ ⇔ √ ≤ ⅝ ⌐ ⁸ ─ Ʀ ⁱ ☻ ─
╩ 1.414 ی ⌐ ⇔ ╕ Δ ⁹
پ ≢ ♪ ! ▬Ƙ ╩ e ≤ ⁸ ─ ۆ ≢ ─ ♪ ! ▬Ƙ ⅜ ɔ ⌂ ╡ ╕ Δ ⁹

ὺ CPP STOP ENABLE
2 ♪ ! ▬Ƙ ≤ ɷ ▬ fi ♪ ! ▬Ƙ ≢ ≢ Δ ⁹
ɷ ▬ fi ─ CPP STOP ╩ ₈ ⌐ Δ ╢ ⌐ Δ ╢ ₉ ╩ ⇔ ╕ Δ ⁹

ה CPP STOP ENABLE = 1 ╩ ⇔ √ │ ⁸ CPPIN ⌐ طمی ≢ ⅝ ╢ Ʀ ⁱ ☻ ─ ثم │ ⁸
2.5 MHz ⌐ ⌂ ╡ ╕ Δ ⁹

ₒ ɷ ▬ fi ─ CPP STOP ₓ
2 ♪ ! ▬Ƙ ≤ ɷ ▬ fi ♪ ! ▬Ƙ ڿ ⌐ ⇔ ╕ Δ ⁹

ה ɷ ▬ fi ─ CPP STOP ENABLE = 1 ⌐ Δ ╢ ≤ ⁸ ɷ ▬ fi ⅜ Δ ╢ ♪ ! ▬Ƙ ─ Ʀ ⁱ ☻ ≤ CPPIN
⅛ ╠ طمی Δ ╢ Ʀ ⁱ ☻ ╩ ئج ○ ► fi ♩ ⇔ ╕ Δ ⁹

ה CPPIN ─ Ʀ ⁱ ☻ ⅜ ⁸ ɷ ▬ fi ─ Ʀ ⁱ ☻ ╠ ╡ Ʀ ⁱ ☻ ⌂ ɔ ⌂ ╢ ≤ ⁸ ɷ ▬ fi ─ ♪ ! ▬Ƙ ╩
ۆ ⇔ ≡ ⁸ ɷ ▬ fi ⅜ Δ ╢ ♪ ! ▬Ƙ ─ Ʀ ⁱ ☻ ╩ تج ⇔ ╕ Δ ⁹
2 ♪ ! ▬Ƙ ≢ │ ⁸ ج ─ ♪ ! ▬Ƙ ╩ ۆ ⇔ ≡ ⁸ Ʀ ⁱ ☻ ╩ تج ⇔ ╕ Δ ⁹

ה CPP STOP ≢ ♪ ! ▬Ƙ ╩ ۆ ⇔ √ │ ⁸ ɷ ▬ fi ─ ◄ ! ⌐ ⌂ ╡ ╕ Δ ⁹
ERROR STATUS ─ CPP STOP ERROR = 1 ⌐ ⇔ ╕ Δ ⁹

ה ɷ ▬ fi │ CPP STOP ≢ ♪ ! ▬Ƙ ╩ ۆ ⇔ ╕ Δ ⅜ ⁸ ◘ Ƙ │ ♪ ! ▬Ƙ ╩ ۆ ⇔ ╕ ∑ ╪ ⁹
ɷ ▬ fi ⅜ CPP STOP ≢ ♪ ! ▬Ƙ ╩ ۆ ⇔ √ │ ⁸ Δ ═ ≡ ─ ◘ Ƙ ⌐ تج خ ╩ ⇔ ≡ ⁸
♪ ! ▬Ƙ ╩ ۆ ↕ ∑ ≡ ɔ ∞ ↕ ™ ⁹

ὺ CPP MASK ENABLE

◘ Ķ ♪ ! ▬Ķ≠ ≠ Δ ⁹

◘ Ķ — CPPIN ∿ ☻ ▾ ╩ ₈ ⌐ Δ ╢ ⌐ Δ ╢ ₉ ╩ ⇔ ╕ Δ ⁹

₀ ◘ Ķ — CPP STOP ˣ

◘ Ķ ♪ ! ▬Ķ ج ⌐ ⇔ ╕ Δ ⁹

ה ◘ Ķ — CPP MASK ENABLE = 1 ⌐ Δ ╢ ≤ ⁸ ◘ Ķ ⅜ ◄ ! ↄ STATUS1 PORT — ERROR = 1 ⌐
⌂ ╢ ≤ ⁸ CPPIN ⅛ ╟ طس Δ ╢ Ꞧ ⁱ ☻ ╩ ∿ ☻ ▾ ⇔ ╕ Δ ⁹

ה CPPOUT SEL ≠ CPPOUT ╩ ₈ CPPIN ⅛ ╟ طس Δ ╢ Ꞧ ⁱ ☻ ₉ ⌐ ⇔ ≡ ™ ╢ │ ⁸ CPPIN —
∿ ☻ ▾ ⌐ ╟ ╡ ⁸ CPPOUT │ Ł ▬ ˣ Ƶ ⁱ ⌐ ⌂ ╡ ╕ Δ ⁹

ה CPPIN ∿ ☻ ▾ ≠ CPPIN ╩ ∿ ☻ ▾ ⇔ √ │ ⁸ STATUS5 PORT — CPP MASK = 1 ⌐ ⌂ ╡ ╕ Δ ⁹
CPP MASK = 1 — │ ⁸ CPPIN — ∿ ☻ ▾ ╩ کا ⇔ ╕ Δ ⁹
CPP MASK = 1 │ ⁸ STATUS1 PORT — ERROR = 1 ́ 0 ≠ CPP MASK = 0 ⌐ ⌂ ╡ ╕ Δ ⁹

ה C-VX870v1 ◦ ⁱ ↄ ☼ ≠ │ ⁸ 2 — MCC07E ╩ CPPIN ≤ CPPOUT ╩ ♦ ☺ ↄ ⌘ ▼ ↄ fi ⇔ ≡ ™ ╕ Δ ⁹
CPPIN ≤ CPPOUT ╩ ♦ ☺ ↄ ⌘ ▼ ↄ fi ⇔ √ ♪ ! ▬Ķ≠ ⁸ CPP STOP ≤ CPPIN ∿ ☻ ▾
╩ ⌐ Δ ╢ ≤ ⁸ ◘ Ķ ⌐ ◄ ! ↄ ⅜ ⇔ √ ⌐ Δ ═ ≡ — — Ꞧ ⁱ ☻ ╩ تج ↕ ∑ ╢ ↓ ≤ ⅜
≠ ⅝ ╕ Δ ⁹

ה ╕ √ ⁸ ♪ ! ▬Ķ — ╩ ⁸ ω ▬ fi ♪ ! ▬Ķ ╕ √ │ ω ▬ fi ⌘ ᵖ ᵏ ♪ ! ▬Ķ≠
⇔ ≡ CPP STOP ╩ ⌐ Δ ╢ ≤ ⁸ تج ⅜ خ ╕ √ │ ◄ ! ↄ ⌐ ╟ ╡ ♪ ! ▬Ķ ╩ ۓ ⇔ √
⌐ ⁸ Δ ═ ≡ — ♪ ! ▬Ķ ╩ ۓ ↕ ∑ ╢ ↓ ≤ ⅜≠ ⅝ ╕ Δ ⁹

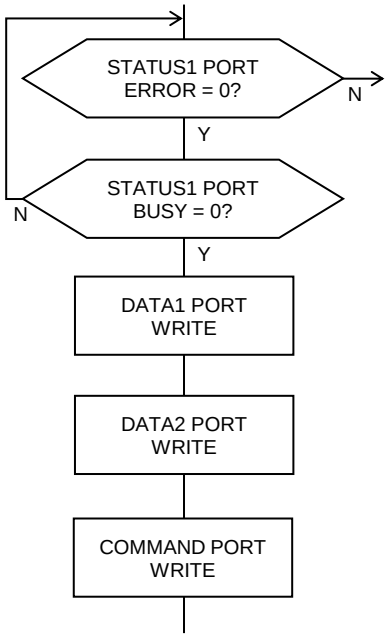
LONG POSITION
SET
SHORT POSITION
SET
LONG POSITION
SET
SHORT POSITION
SET

CP SEPEC SET
CP SPEC SET
LONG POSITION
SET
SHORT POSITION
SET
LONG POSITION
SET
SHORT POSITION
SET
CPPOUT
CPPOUT SEL = "CPPIN
CPPOUT SEL =
CPPIN

(1) LONG POSITION SET

直線補間ドライブの、長軸の座標アドレスを設定します。

< ◦ ↄ ◔ fi ☻ >　　COMMAND : H'22　　LONG POSITION SET COMMAND



ŋ DRIVE STATUS1 PORT ─ ERROR ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ 9

º DRIVE STATUS1 PORT ─ BUSY ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ 9

ɔ DRIVE DATA1 PORT ⌐ ─ ─ ▪ ♪ ☻ ╩ ⅝ ╥ ╕ Δ 9

ˆ DRIVE DATA2 PORT ⌐ ─ ─ ▪ ♪ ☻ ╩ ⅝ ╥ ╕ Δ 9
│ 8
-2,147,483,648　+2,147,483,647　H'8000_0000　H'7FFF_FFFF　≠ Δ 9
⅜ CW 8　⅜ CCW　≠ Δ 9　─　│ 2 ─　⌐ ⇔ ╕ Δ 9

˘ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╥ ╕ Δ 9

DRIVE DATA1 PORT ─　♦ ↄ ♘

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←					─	─		▪ ♪	☻				→	D0

DRIVE DATA2 PORT ─　♦ ↄ ♘

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←					─	─		▪ ♪	☻				→	D16

● リセット後の初期値は H'0000_0000 です。

・ 指定する座標アドレスは、現在位置を座標中心（0, 0）とした相対アドレスです。

・「長軸の目的地の座標アドレス」には、補間軸の中で補間パルス数が大きい補間軸（長軸）の目的地を設定します。

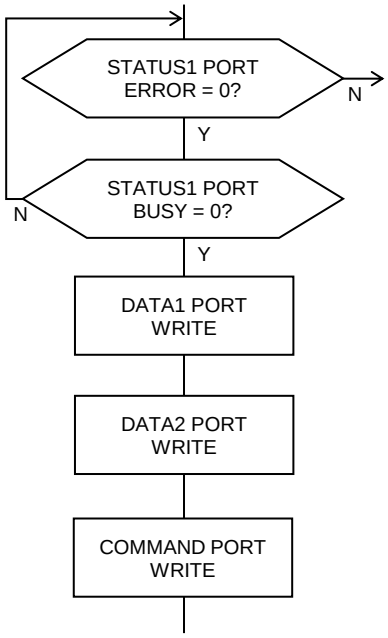
・ ドライブ実行コマンドの PULSE SEL で指定した軸の座標アドレスの符号が、出力する補間パルスの動作方向になります。

(2) SHORT POSITION SET

直線補間ドライブの、短軸の座標アドレスを設定します。

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'23　SHORT POSITION SET COMMAND



ŋ DRIVE STATUS1 PORT ― ERROR ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

º DRIVE STATUS1 PORT ― BUSY ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ɔ DRIVE DATA1 PORT ⌐ ― ― ▪ ♪ ☻ ╩ ⅝ ╖ ╕ Δ ⁹

ˆ DRIVE DATA2 PORT ⌐ ― ― ▪ ♪ ☻ ╩ ⅝ ╖ ╕ Δ ⁹
│ ⁸
-2,147,483,648　+2,147,483,647　H'8000_0000　H'7FFF_FFFF　≠ Δ ⁹
⅜ CW ⁸　⅜ CCW　≠ Δ ⁹　―　│ 2 ―　⌐ ⇔ ╕ Δ ⁹

˘ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

DRIVE DATA1 PORT ―　♦ ↄ ㉰

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←					―	―		▪ ♪	☻				→	D0

DRIVE DATA2 PORT ―　♦ ↄ ㉰

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←					―	―		▪ ♪	☻				→	D16

● リセット後の初期値は H'0000_0000 です。

・ 指定する座標アドレスは、現在位置を座標中心（0, 0）とした相対アドレスです。

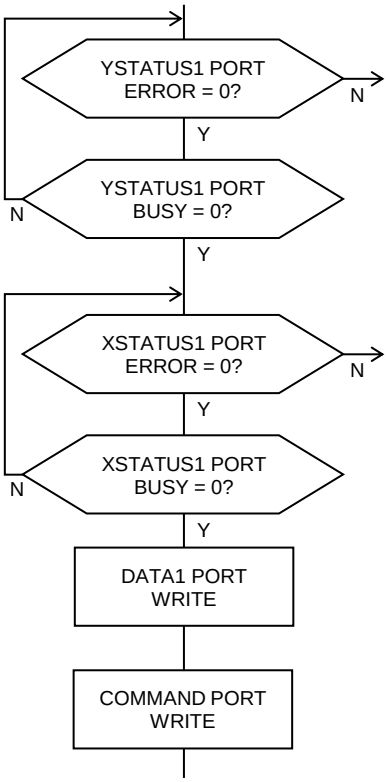
・ 「短軸の目的地の座標アドレス」には、短軸の目的地（符号付きアドレス）を設定します。
「短軸の目的地の座標アドレス」は「長軸の座標アドレスの絶対値≧短軸の座標アドレスの絶対値」となる様に設定してください。

・ ドライブ実行コマンドの PULSE SEL で指定した軸の座標アドレスの符号が、出力する補間パルスの動作方向になります。

(3) MAIN XY STRAIGHT CP

2 軸相関コマンドです。相関軸両軸が BUSY = 0, ERROR = 0 のときにコマンドを実行します。
メインチップの相関 2 軸直線補間ドライブを実行します。相関軸のどちらの軸に実行しても有効です。
直線補間ドライブは実行軸の加減速パラメータで動作します。

< ◦ ↄ ◣ fi ☻ > COMMAND : H'32 MAIN XY STRAIGHT CP COMMAND



ŋ 相関軸 STATUS1 PORT の ERROR フラグが "0" であることを確認します。

º 相関軸 STATUS1 PORT の BUSY フラグが "0" であることを確認します。

③ 実行軸 STATUS1 PORT の ERROR フラグが "0" であることを確認します。

ˆ STATUS1 PORT ー BUSY ɦ ! ◓ ⅜ "0" ≠ № ╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

˘ 実行軸 DRIVE DATA1 PORT に設定データを書き込みます。

⑥ DRIVE COMMAND PORT に COMMAND を書き込みます。

DRIVE DATA1 PORT ー ♦ ↄ ㉾

D15	D14	D13	D12	D11	D10	D9	D8

D7	D6	D5	D4	D3	D2	D1	D0
		YPULSE SEL	XPULSE SEL		0	CONST CP ENABLE	DRIVE MODE

D0 : DRIVE MODE
直線補間ドライブを「連続ドライブにする／位置決めドライブにする」を選択します。
0 : 連続ドライブにする (SCAN ドライブ)
1 : 位置決めドライブにする (INDEX ドライブ)

D1 : CONST CP ENABLE
線速一定制御を「無効にする／有効にする」を選択します。
0 : 線速一定制御を無効にする
1 : 線速一定制御を有効にする

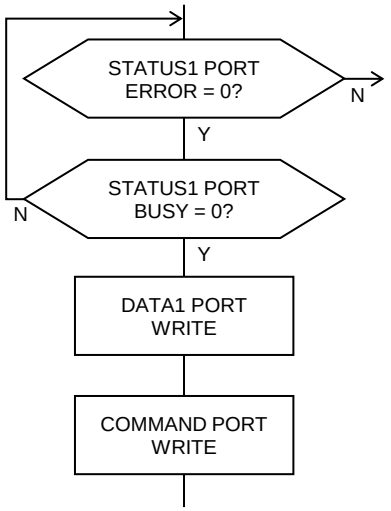
D4 : XPULSE SEL
相関軸の内、Xn 軸(または Zn 軸,Bn 軸)に出力する補間パルスを選択します。
0 : X 軸に LONG POSITION (長軸)の補間パルスを出力する
1 : X 軸に SHORT POSITION (短軸)の補間パルスを出力する

D5 : YPULSE SEL
相関軸の内、Yn 軸(または An 軸,Cn 軸)に出力する補間パルスを選択します。
0 : Y 軸に LONG POSITION (長軸)の補間パルスを出力する
1 : Y 軸に SHORT POSITION (短軸)の補間パルスを出力する

(4) SUB STRAIGHT CP

1 軸単位で直線補間ドライブを行うコマンドです。
任意軸間の直線補間、または複数軸で直線補間ドライブさせるときにサブ軸に実行します。
サブ軸の直線補間ドライブは CPPIN 入力パルスで動作します。

< ◦ ↄ ◣ fi ☻ > COMMAND : H'31 SUB STRAIGHT CP COMMAND



ŋ DRIVE STATUS1 PORT ─ ERROR ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

º DRIVE STATUS1 PORT ─ BUSY ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ↄ DRIVE DATA1 PORT に設定データを書き込みます。

ˆ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╥ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ↄ ⅈ

D15	D14	D13	D12	D11	D10	D9	D8

D7	D6	D5	D4	D3	D2	D1	D0
			PULSE SEL		CPP MASK ENABLE		DRIVE MODE

D0 : DRIVE MODE
直線補間ドライブを「連続ドライブにする／位置決めドライブにする」を選択します。
0 : 連続ドライブにする (SCAN ドライブ)
1 : 位置決めドライブにする (INDEX ドライブ)

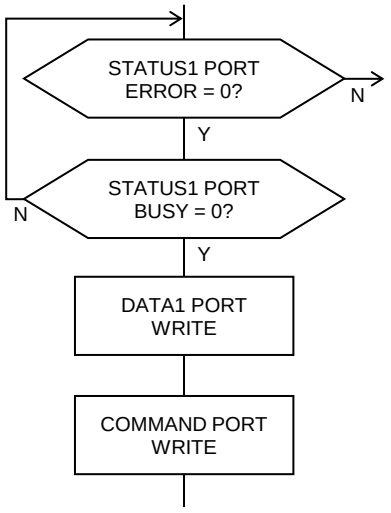
D2 : CPP MASK ENABLE
CPPIN マスク機能を「有効にする／無効にする」を選択します。
0 : CPPIN マスク機能を無効にする
1 : CPPIN マスク機能を有効にする

D4 : PULSE SEL
出力する補間パルスを選択します。
0 : LONG POSITION（長軸）の補間パルスを出力する
1 : SHORT POSITION（短軸）の補間パルスを出力する

(5) MAIN STRAIGHT CP

1 軸単位で直線補間ドライブを行うコマンドです。
任意軸間の直線補間、または複数軸で直線補間ドライブさせるときにメイン軸に実行します。
メイン軸の直線補間ドライブは実行軸の加減速パラメータで動作します。

< ◦ ↄ ◣ fi ☻ > COMMAND : H'30 MAIN STRAIGHT CP COMMAND



ŋ DRIVE STATUS1 PORT ─ ERROR ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

º DRIVE STATUS1 PORT ─ BUSY ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ɔ DRIVE DATA1 PORT に設定データを書き込みます。

ˆ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╥ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ↄ ⅟

D15	D14	D13	D12	D11	D10	D9	D8

D7	D6	D5	D4	D3	D2	D1	D0
			PULSE SEL		CPP STOP ENABLE	CONST CP ENABLE	DRIVE MODE

D0 : DRIVE MODE
直線補間ドライブを「連続ドライブにする／位置決めドライブにする」を選択します。
0 : 連続ドライブにする (SCAN ドライブ)
1 : 位置決めドライブにする (INDEX ドライブ)

D1 : CONST CP ENABLE
線速一定制御を「無効にする／有効にする」を選択します。
0 : 線速一定制御を無効にする
1 : 線速一定制御を有効にする

D2 : CPP STOP ENABLE
CPP STOP 機能を「有効にする／無効にする」を選択します。
0 : CPP STOP 機能を無効にする
1 : CPP STOP 機能を有効にする

D4 : PULSE SEL
出力する補間パルスを選択します。
0 : LONG POSITION（長軸）の補間パルスを出力する
1 : SHORT POSITION（短軸）の補間パルスを出力する

2-1-7. 円弧補間ドライブの設定と実行

現在位置の X-Y 座標アドレスと、目的地の短軸座標までの短軸パルス数と、ドライブ仕様を指定して、
相関 2 軸円弧補間ドライブ および 任意 2 軸直線補間ドライブを実行します。

・ 円弧補間ドライブは実行軸の加減速パラメータをドライブします。
・ 円弧補間ドライブでは、円弧の中心座標からみた短軸側の短軸パルスを補間ドライブの基本パルスとし、
長軸側は短軸パルスを補間演算して補間パルスを出力します。
・ 円弧の中心点座標を(0, 0)とした X 軸と Y 軸の相対アドレスを、X-Y 座標アドレスとします。
座標アドレスは、正数が ＋(CW)方向、負数が －(CCW)方向です。

■ 円弧補間ドライブ仕様

● DRIVE MODE

円弧補間ドライブを「連続ドライブにする／位置決めドライブにする」を選択します。

・ 円弧補間 SCAN ドライブ
各補間軸は、指定の円弧半径と回転方向で、補間パルス出力を続けます。
停止指令を検出すると、補間ドライブを終了します。

・ 円弧補間 INDEX ドライブ
各補間軸は、指定の円弧半径と回転方向で、補間パルス出力を続けます。
短軸パルスをカウントして、カウント数が指定の短軸パルス数になると、補間ドライブを終了します。

● CONST CP ENABLE

相関 2 軸円弧補間ドライブとメイン軸円弧補間ドライブで有効です。
線速一定制御を「無効にする／有効にする」を選択します。

・ 線速一定制御
補間ドライブしている2軸の合成速度を一定にする制御です。
コマンド実行軸が発生する補間ドライブの短軸パルスを線速一定制御します。
X 座標軸と Y 座標軸の2軸間で、2軸同時にパルス出力したときに、次の短軸パルスの出力周期を
1.414 倍にします。
線速一定で加減速ドライブを行うと、減速後の終了速度でのドライブが長くなります。

o x
ﻳ ー ﻣﺣ ♪ ! ▬ K ⅜ 8 2 ⌐ R i ☻ ⇔ √ ₀ ≠ ﻭ ⇔ √ ⌐ 8
ﺧ ⌐ Δ ╢ ﻳ ー ♪ ! ▬ K ー R i ☻ ⅜ 8
⌐ ﺛﻡ ー 1.414 ﻳﻱ ー ⌐ ﻳ ↕ ╣ ╢ ⌐ ⌂ ╡ ╕ Δ 9

ﻩ 1 軸のみパルス出力する位置（例: 0 Á90 Á180 Á270 Á）で終了した場合は正常です。
ﻩ 2 軸同時にパルス出力する位置（例:45 Á135 Á225 Á315 Á）で終了した場合に不具合が発生します。

線速一定制御有効の円弧補間ドライブ終了後は、
以下の円弧補間ドライブ（0 パルス、終了位置 0 Á）を実行して、正常終了にしてください。

ﻩ CIRCULAR XPOSITION SET コマンド	(H'28)	:H'00_0000 に設定
ﻩ CIRCULAR YPOSITION SET コマンド	(H'29)	:H'00_0000 に設定
ﻩ CIRCULAR PULSE SET コマンド	(H'2A)	:H'0000_0000 に設定
ﻩ MAIN CIRCULAR CP コマンド	(H'38)	:DATA1=H'0001 で実行

● CPP STOP ENABLE
2 軸相関円弧補間ドライブとメイン軸円弧補間ドライブで有効です。
メイン軸の CPP STOP 機能を「有効にする／無効にする」を選択します。

・ CPP STOP ENABLE = 1 を選択した場合は、CPPIN 端子に入力できるパルス速度の最高値は、2.5 MHz になります。

【メイン軸の CPP STOP 機能】
2 軸相関円弧補間ドライブとメイン軸円弧補間ドライブ実行中に機能します。

・ メイン軸の CPP STOP ENABLE = 1 にすると、メイン軸が発生する補間ドライブの基本パルスと CPPIN から入力するパルスを偏差カウントします。

・ CPPIN のパルス数が、メイン軸の基本パルス数より2パルス分少なくなると、メイン軸のドライブを終了して、メイン軸が発生する補間ドライブの基本パルス出力を停止します。
2 軸相関円弧補間ドライブでは、両軸のドライブを終了して、基本パルス出力を停止します。

・ CPP STOP 機能でドライブを終了した場合は、メイン軸のエラーになります。
ERROR STATUS の CPP STOP ERROR = 1 にします。

・ メイン軸は CPP STOP 機能でドライブを終了しますが、サブ軸はドライブを終了しません。
メイン軸が CPP STOP 機能でドライブを終了した場合は、すべてのサブ軸に停止指令を実行して、ドライブを終了させてください。

● CPP MASK ENABLE
サブ軸円弧補間ドライブで有効です。
サブ軸の CPPIN マスク機能を「有効にする／無効にする」を選択します。

【サブ軸の CPP STOP 機能】
サブ軸補間ドライブ実行中に機能します。

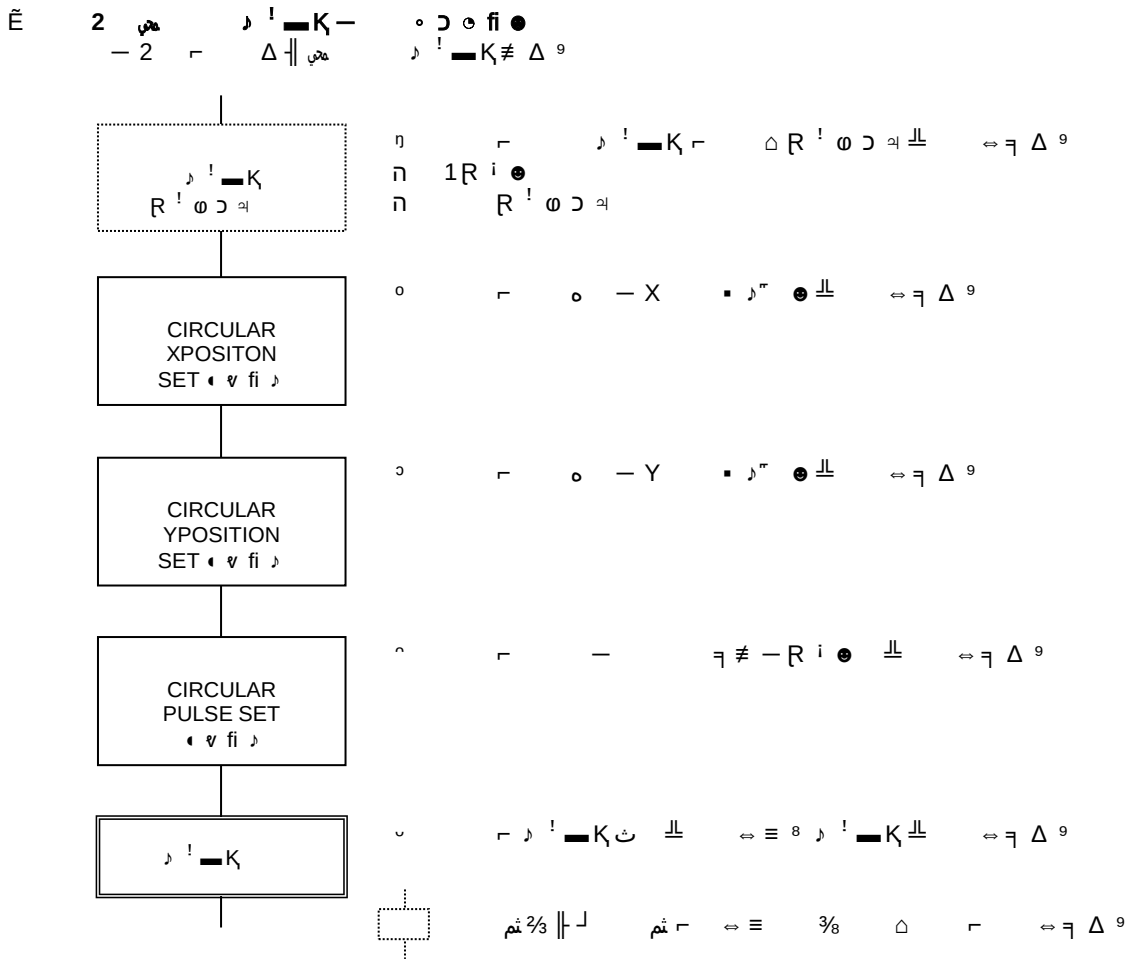
・ サブ軸の CPP MASK ENABLE = 1 にすると、サブ軸がエラー（STATUS1 PORT の ERROR = 1）になると、CPPIN から入力するパルスをマスクします。

・ CPPOUT SEL で CPPOUT 出力を「CPPIN から入力するパルス」に設定している場合は、CPPIN のマスクにより、CPPOUT 出力はハイレベル状態になります。

・ CPPIN マスク機能で CPPIN をマスクした場合は、STATUS5 PORT の CPP MASK = 1 になります。
CPP MASK = 1 の間は、CPPIN のマスク状態を保持します。
CPP MASK = 1 は、STATUS1 PORT の ERROR = 1 → 0 で CPP MASK = 0 になります。

・ C-VX870v1 シリーズでは、相関 2 軸の MCC07E 間同士を CPPIN と CPPOUT をデジーチェーン接続しています。
任意軸円弧補間ドライブで、CPP STOP 機能と CPPIN マスク機能を有効にすると、サブ軸にエラーが発生した場合にすべての補間軸のパルス出力を停止させることができます。

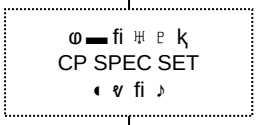
・ また、任意軸円弧補間ドライブの補間軸を、メイン軸補間ドライブまたは2軸相関円弧補間ドライブで構成して CPP STOP 機能を有効にすると、1軸が停止指令またはエラーにより補間ドライブを終了した場合に、すべての補間ドライブを終了させることができます。


CIRCULAR
XPOSITON
SET
CIRCULAR
YPOSITION
SET
CIRCULAR
PULSE SET

Ẽ ش　2 軸円弧補間ドライブの実行シーケンス

任意 2 軸間(2 チップ間)で実行する円弧補間ドライブです。

ŋ　CPPOUT
CPPOUT SEL = "CPPIN

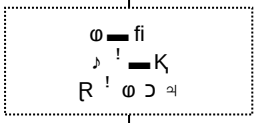


o　CPPOUT
CPPOUT SEL = "

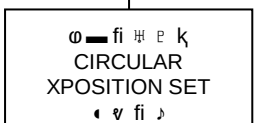
ɔ　— X 座標アドレスを設定します。

ˆ　— Y 座標アドレスを設定します。

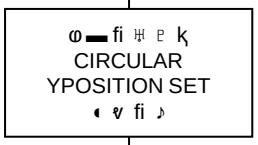
ˇ



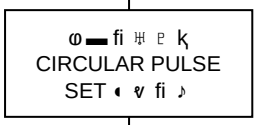
p



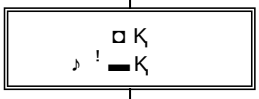
t　— X 座標アドレスを設定します。



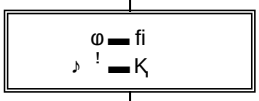
u　— Y 座標アドレスを設定します。



ɔ



⑩ サブ軸に、サブ軸円弧補間ドライブを実行します。



v

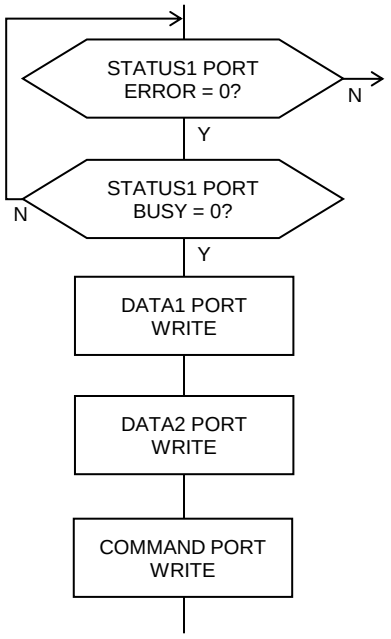
ɔ サブ軸円弧補間ドライブを実行すると、ドライブがスタンバイ状態になります。

ɔ メイン軸円弧補間ドライブを実行すると、短軸パルスを出力して、ドライブを開始します。
CPPIN

(1) CIRCULAR XPOSITION SET

円弧の中心点座標を(0, 0)とした現在位置の X 座標アドレスを設定します。
このコマンドの設定は相関軸で共有します。
実行軸のどちらの軸に設定しても有効です。

< ◦ ↄ ◣ fi ☻ > COMMAND : H'28 CIRCULAR XPOSITION SET COMMAND



ŋ DRIVE STATUS1 PORT ─ ERROR ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

º DRIVE STATUS1 PORT ─ BUSY ɦ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ɔ DRIVE DATA1 PORT ⌐ ₒ ─ X ▪ ♪ ☻ ╩ ⅝ ╗ ╕ Δ ⁹

ˆ DRIVE DATA2 PORT ⌐ ₒ ─ X ▪ ♪ ☻ ╩ ⅝ ╗ ╕ Δ ⁹
│ ⁸
-8,388,608 ～+8,388,607(H'80_0000 ～ H'7F_FFFF)です。
⅜ CW ⁸ ⅜ CCW ≠ Δ ⁹ ─ │ 2 ─ ⌐ ⇔ ╕ Δ ⁹

ˇ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╗ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ↄ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←					ₒ	─X	▪	♪	☻				→	D0

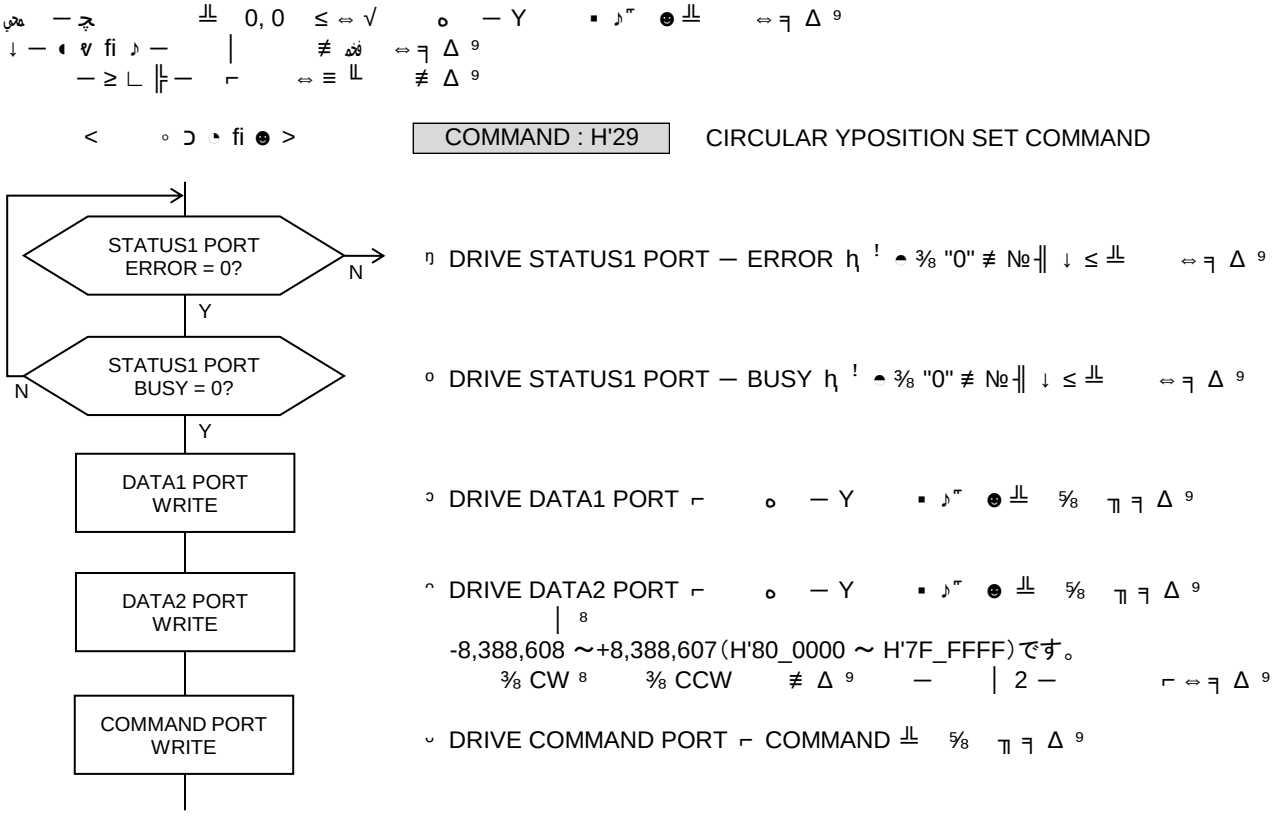
DRIVE DATA2 PORT ─ ♦ ↄ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
─	─	─	─	─	─	─	─	D23	←	ₒ	─X	▪	♪	☻ ⁾	D16

● リセット後の初期値は H'00_0000 です。

ɲ 指定する座標アドレスは、円弧の中心点座標を(0, 0)とした X 軸の相対アドレスです。

(2) CIRCULAR YPOSITION SET



COMMAND : H'29 CIRCULAR YPOSITION SET COMMAND

DRIVE STATUS1 PORT ー ERROR

DRIVE STATUS1 PORT ー BUSY

DRIVE DATA1 PORT

DRIVE DATA2 PORT

-8,388,608 ～+8,388,607（H'80_0000 ～ H'7F_FFFF）です。

DRIVE COMMAND PORT COMMAND

DRIVE DATA1 PORT ー

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←						ー Y							→	D0

DRIVE DATA2 PORT ー

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
ー	ー	ー	ー	ー	ー	ー	ー	D23	←		ー Y				D16

● リセット後の初期値は H'00_0000 です。

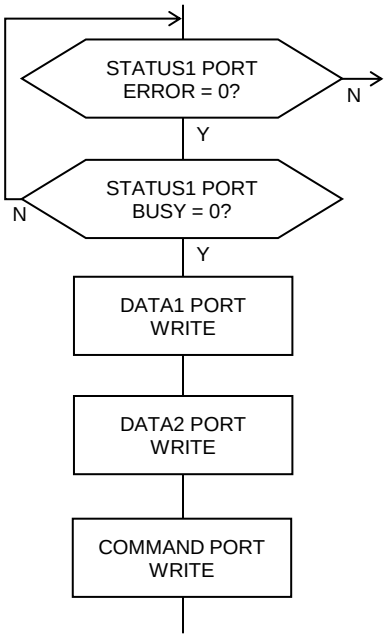
指定する座標アドレスは、円弧の中心点座標を（0, 0）とした Y 軸の相対アドレスです。

(3) CIRCULAR PULSE SET

[illegible]

< [illegible] >

COMMAND : H'2A CIRCULAR PULSE SET COMMAND



[illegible] DRIVE STATUS1 PORT [illegible] ERROR [illegible] "0" [illegible]

[illegible] DRIVE STATUS1 PORT [illegible] BUSY [illegible] "0" [illegible]

[illegible] DRIVE DATA1 PORT [illegible]

[illegible] DRIVE DATA2 PORT [illegible]

-2,147,483,648 +2,147,483,647 H'8000_0000 H'7FFF_FFFF [illegible]

[illegible] CW [illegible] CCW [illegible]

[illegible] DRIVE COMMAND PORT [illegible] COMMAND [illegible]

DRIVE DATA1 PORT [illegible]

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←					[illegible]								→	D0

DRIVE DATA2 PORT [illegible]

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←					[illegible]								→	D16

[illegible] H'00_0000 [illegible]

[illegible]

[illegible]

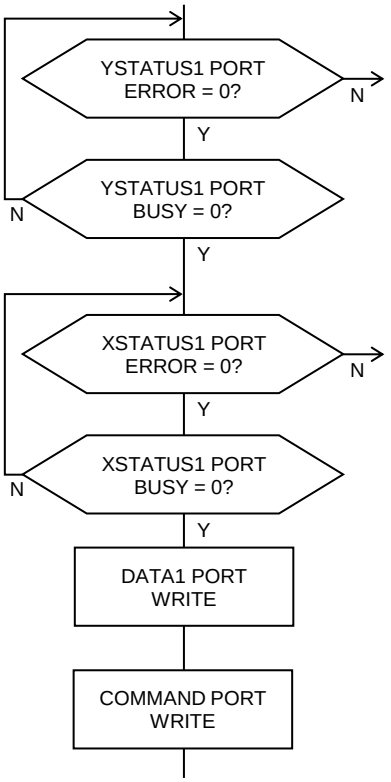
[illegible] 3-1-6.(4) [illegible]

[illegible]

(4) MAIN XY CIRCULAR CP

2 ◖ ∀ fi ♪ ≠ Δ ⁹ ⅜ BUSY = 0, ERROR = 0 — ≤ ⅝ ⌐ ◖ ∀ fi ♪ ⊥ ⇔ ╕ Δ ⁹
ω ▬ fi ⌗ ᵖ ⅊ — 2 ♪ ! ▬ Ꝗ ⊥ ⇔ ╕ Δ ⁹ — ≥ ∟ ╟ — ⌐ ⇔ ≡ ⊥ ≠ Δ ⁹
♪ ! ▬ Ꝗ │ — R ! ω ⊃ ⁿ ≠ ⅊ ⇔ ╕ Δ ⁹

< ◦ ⊃ ◗ fi ☻ > COMMAND : H'3A MAIN XY CIRCULAR CP COMMAND



n STATUS1 PORT — ERROR ḥ ! ◓ ⅜ "0" ≠ № ╢ ↓ ≤ ⊥ ⇔ ╕ Δ ⁹

o STATUS1 PORT — BUSY ḥ ! ◓ ⅜ "0" ≠ № ╢ ↓ ≤ ⊥ ⇔ ╕ Δ ⁹

c STATUS1 PORT — ERROR ḥ ! ◓ ⅜ "0" ≠ № ╢ ↓ ≤ ⊥ ⇔ ╕ Δ ⁹

n STATUS1 PORT — BUSY ḥ ! ◓ ⅜ "0" ≠ № ╢ ↓ ≤ ⊥ ⇔ ╕ Δ ⁹

u DRIVE DATA1 PORT ⌐ ♦ ⊃ ⁿ ⊥ ⅝ ╖ ╕ Δ ⁹

p DRIVE COMMAND PORT ⌐ COMMAND ⊥ ⅝ ╖ ╕ Δ ⁹

DRIVE DATA1 PORT — ♦ ⊃ ⁿ

D15	D14	D13	D12	D11	D10	D9	D8

D7	D6	D5	D4	D3	D2	D1	D0
		YPULSE SEL	XPULSE SEL		0	CONST CP ENABLE	DRIVE MODE

D0 DRIVE MODE

ⁿ ♪ ! ▬ Ꝗ ⊥ ⁸ ♪ ! ▬ Ꝗ ⌐ Δ ╢ ₀ ⅊ ♪ ! ▬ Ꝗ ⌐ Δ ╢ ⁹ ⊥ ⇔ ╕ Δ ⁹

0 ♪ ! ▬ Ꝗ ⌐ Δ ╢ (SCAN ♪ ! ▬ Ꝗ)

1 ₀ ⅊ ♪ ! ▬ Ꝗ ⌐ Δ ╢ (INDEX ♪ ! ▬ Ꝗ)

D1 CONST CP ENABLE

ᵖ ⊥ ⁸ ⌐ Δ ╢ ⌐ Δ ╢ ⁹ ⊥ ⇔ ╕ Δ ⁹

0 ᵖ ⊥ ⌐ Δ ╢

1 ᵖ ⊥ ⌐ Δ ╢

D4 XPULSE SEL

— ⁿ ⁸ Xn (╕ √ │ Zn ,Bn) ⌐ Δ ╢ R ⁱ ☻ ⊥ ⇔ ╕ Δ ⁹

0 X ⌐ ⁿ — X ▪ ♪ ☻ — R ⁱ ☻ XCP ⊥ Δ ╢

1 X ⌐ ⁿ — Y ▪ ♪ ☻ — R ⁱ ☻ YCP ⊥ Δ ╢

D5 YPULSE SEL

— ⁿ ⁸ Yn (╕ √ │ An ,Cn) ⌐ Δ ╢ R ⁱ ☻ ⊥ ⇔ ╕ Δ ⁹

0 Y ⌐ ⁿ — X ▪ ♪ ☻ — R ⁱ ☻ XCP ⊥ Δ ╢

1 Y ⌐ ⁿ — Y ▪ ♪ ☻ — R ⁱ ☻ YCP ⊥ Δ ╢

(5) SUB CIRCULAR CP

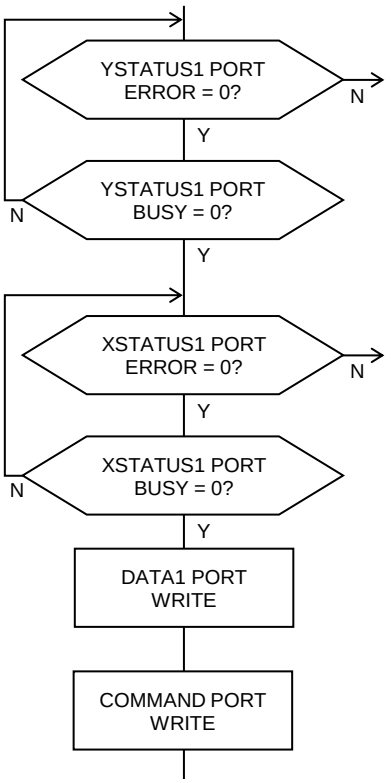
2 ◖ ∀ fi ♪ ≠ Δ ⁹ X, Y ⅜ BUSY = 0, ERROR = 0 — ≤ ⅝ ⌐ ◖ ∀ fi ♪ ╩ ⇔ ╕ Δ ⁹

1 ₀ ≠ ♥ ♪ ! ▬ Ķ ╩ e ◖ ∀ fi ♪ ≠ Δ ⁹

ش — ♥ ♪ ! ▬ Ķ ↕ ∑ ╢ ≤ ⅝ ⌐ ◘ Ķ ⌐ ⇔ ╕ Δ ⁹

◘ Ķ — ♥ ♪ ! ▬ Ķ │ CPPIN ط Ꞧ i ☻ ≠ ⅄ ⇔ ╕ Δ ⁹

< ◦ ↄ ◗ fi ☻ > COMMAND : H'39 SUB CIRCULAR CP COMMAND



ŋ STATUS1 PORT — ERROR ḫ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

₀ STATUS1 PORT — BUSY ḫ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ↄ STATUS1 PORT — ERROR ḫ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ⁿ STATUS1 PORT — BUSY ḫ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ᵘ DRIVE DATA1 PORT ⌐ ♦ ↄ ↄ ╩ ⅝ ╖ ╕ Δ ⁹

ᵖ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

DRIVE DATA1 PORT — ♦ ↄ ↄ

D15	D14	D13	D12	D11	D10	D9	D8

D7	D6	D5	D4	D3	D2	D1	D0
			PULSE SEL		CPP MASK ENABLE		DRIVE MODE

D0 DRIVE MODE

♥ ♪ ! ▬ Ķ ╩ ₈ ♪ ! ▬ Ķ ⌐ Δ ╢ ₀ ⅃ ♪ ! ▬ Ķ ⌐ Δ ╢ ₉ ╩ ⇔ ╕ Δ ⁹

0 ♪ ! ▬ Ķ ⌐ Δ ╢ (SCAN ♪ ! ▬ Ķ)

1 ₀ ⅃ ♪ ! ▬ Ķ ⌐ Δ ╢ (INDEX ♪ ! ▬ Ķ)

D2 CPP MASK ENABLE

CPPIN ∀ ☻ ▾ ╩ ₈ ⌐ Δ ╢ ⌐ Δ ╢ ₉ ╩ ⇔ ╕ Δ ⁹

0 CPPIN ∀ ☻ ▾ ╩ ⌐ Δ ╢

1 CPPIN ∀ ☻ ▾ ╩ ⌐ Δ ╢

D4 PULSE SEL

Δ ╢ Ꞧ i ☻ ╩ ⇔ ╕ Δ ⁹

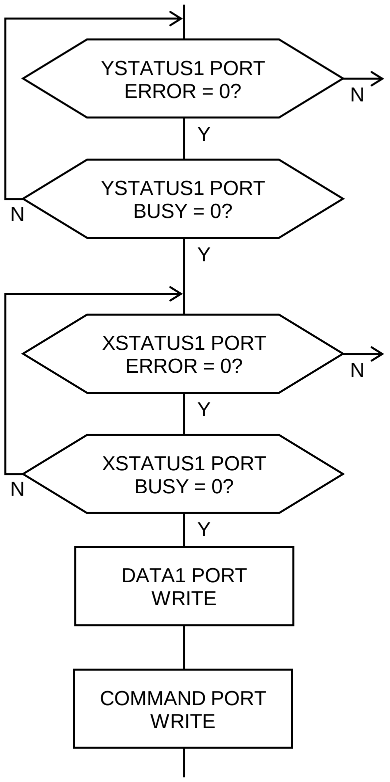
0 ♥ — X ▪ ♪ ☻ — Ꞧ i ☻ XCP ╩ Δ ╢

1 ♥ — Y ▪ ♪ ☻ — Ꞧ i ☻ YCP ╩ Δ ╢

(6) MAIN CIRCULAR CP

2 ◖ ▽ fi ♪ ≢ Δ ⁹ X, Y ⅜ BUSY = 0, ERROR = 0 ─ ≤ ⅝ ⌐ ◖ ▽ fi ♪ ╩ ⇔ ╕ Δ ⁹
1 ο ≢ ﻤﻬ ♪ ! ▬ Ķ ╩ e ◖ ▽ fi ♪ ≢ Δ ⁹
ﺶ ─ ﻤﻬ ♪ ! ▬ Ķ ↕ ∑ ╢ ≤ ⅝ ⌐ ϖ ▬ fi ⌐ ⇔ ╕ Δ ⁹
ϖ ▬ fi ─ ﻤﻬ ♪ ! ▬ Ķ │ ─ Ɽ ! ϖ ɔ ♃ ≢ ﻷ ⇔ ╕ Δ ⁹

< ◦ ɔ ◔ fi ☻ > COMMAND : H'38 MAIN CIRCULAR CP COMMAND



ŋ STATUS1 PORT ─ ERROR ɧ ! ◒ ⅜ "0" ≢ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ο STATUS1 PORT ─ BUSY ɧ ! ◒ ⅜ "0" ≢ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ɔ STATUS1 PORT ─ ERROR ɧ ! ◒ ⅜ "0" ≢ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ᵔ STATUS1 PORT ─ BUSY ɧ ! ◒ ⅜ "0" ≢ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ᵕ DRIVE DATA1 PORT ⌐ ♦ ɔ ♃ ╩ ⅝ ╖ ╕ Δ ⁹

p DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ɔ ♃

D15	D14	D13	D12	D11	D10	D9	D8

D7	D6	D5	D4	D3	D2	D1	D0
			PULSE SEL		CPP STOP ENABLE	CONST CP ENABLE	DRIVE MODE

D0 DRIVE MODE

ﻤﻬ ♪ ! ▬ Ķ ╩ ₈ ♪ ! ▬ Ķ ⌐ Δ ╢ ο ╘ ♪ ! ▬ Ķ ⌐ Δ ╢ ₉ ╩ ⇔ ╕ Δ ⁹

0 ♪ ! ▬ Ķ ⌐ Δ ╢ (SCAN ♪ ! ▬ Ķ)
1 ο ╘ ♪ ! ▬ Ķ ⌐ Δ ╢ (INDEX ♪ ! ▬ Ķ)

D1 CONST CP ENABLE

ﭗ ╩ ₈ ⌐ Δ ╢ ⌐ Δ ╢ ₉ ╩ ⇔ ╕ Δ ⁹

0 ﭗ ╩ ⌐ Δ ╢
1 ﭗ ╩ ⌐ Δ ╢

D2 CPP STOP ENABLE

CPP STOP ╩ 8 ⌐ Δ ╢ ⌐ Δ ╢ 9 ╩ ⇔ ╕ Δ 9

0 CPP STOP ╩ ⌐ Δ ╢

1 CPP STOP ╩ ⌐ Δ ╢

D4 PULSE SEL

Δ ╢ Ꞧ i ☻ ╩ ⇔ ╕ Δ 9

0 ⌂ — X ▪ ♪ ☻ — Ꞧ i ☻ XCP ╩ Δ ╢

1 ⌂ — Y ▪ ♪ ☻ — Ꞧ i ☻ YCP ╩ Δ ╢

o ⌂ x

خ ⌂ — ≠ ♪ ! ▬ Ꞧ ╩ ⇔ √ ⌐ 8 ♪ ! ▬ Ꞧ ╩ ⇔ ≡ ™ ⌂ ™ ⅛ ╬ ╚ Ꞧ i ☻ ⅜

↕ ╢ ≡ ⇔ ╕ e ⌂ ⅜ ⅜ ⇔ ╕ Δ 9

η Y ⌐ ⌂ ♪ ! ▬ Ꞧ ╩ Δ ╢ 9

o Y ⌐ ⌐ ⇔ √ ♪ ! ▬ Ꞧ ⅜ ⌂ ♪ ! ▬ Ꞧ — ≤ ⅝ 8

X ⌐ 8 ɷ ▬ fi ⌂ ♪ ! ▬ Ꞧ 9 ╕ √ │ 8 ◘ Ꞧ ⌂ ♪ ! ▬ Ꞧ 9 ╩ Δ ╢ 9

↓ — ≤ ⅝ X — ⌐ 8 ♪ ! ▬ Ꞧ ╩ ⇔ ≡ ™ ⌂ ™ Y ⅛ ╬ ╚ ⌂ ♪ ! ▬ Ꞧ —

Ꞧ i ☻ ⅜ ↕ ╢ ≡ ⇔ ╕ ™ ╕ Δ 9 ↓ — ⌂ │ X ≤ Y ⅜ ≠ ╚ ⌐ ⇔ ╕ Δ 9

╕ √ 8 ↓ — ⌂ │ ≠ ⌐ ⇔ ╕ Δ 9

o x

8 ɷ ▬ fi ⌂ ♪ ! ▬ Ꞧ 9 ⅔ ╬ ┘ 8 ◘ Ꞧ ⌂ ♪ ! ▬ Ꞧ 9 ╩ Δ ╢ ⌐ 8

♪ ! ▬ Ꞧ ╩ ⇔ ⌂ ™ ⌐ ⇔ ≡ 8 ⌂ ♪ ! ▬ Ꞧ خ — ♪ ! ▬ Ꞧ ╩ ⇔ ≡

ɔ ∞ ↕ ™ 9

ئ η Y ⌐ ⌂ ♪ ! ▬ Ꞧ ╩ Δ ╢ 9

o Y ⌐ 8 0 — ▪ ♪ ☻ INDEX ♪ ! ▬ Ꞧ 9 ╩ Δ ╢ 9

X ⌐ 8 ɷ ▬ fi ⌂ ♪ ! ▬ Ꞧ 9 ╕ √ │ 8 ◘ Ꞧ ⌂ ♪ ! ▬ Ꞧ 9 ╩

Δ ╢ 9

2-1-8. UP/DOWN/CONST ♪ ! ▬Ķ CHANGE ─ ≤

⇔ ≡ ⁸ UP/DOWN/CONST ─ ♪ ! ▬Ķ CHANGE ⇔ ╕ Δ ⁹

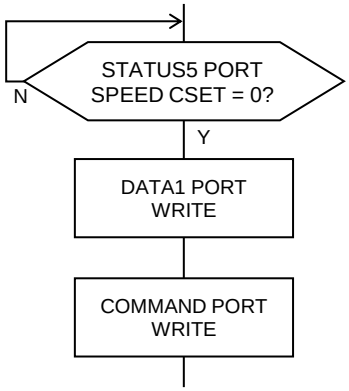
─ │ ⁸ ─ ⅜ ⌂ ⌐ ⇔ ╕ Δ ⁹

(1) UDC SPEC SET

UP/DOWN/CONST ─ ♪ ! ▬Ķ CHANGE Δ ╢ ⇔ ╕ Δ ⁹

< ◦ ↄ ◣ fi ☻ >

COMMAND : H'C0 UDC SPEC SET COMMAND



ŋ STATUS5 PORT ─ SPEED CSET ħ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ⇔ ╕ Δ ⁹

ο DRIVE DATA1 PORT ⌐ ♦ ↄ ㉯ ⅝ ╥ ╕ Δ ⁹

ↄ DRIVE COMMAND PORT ⌐ COMMAND ⅝ ╥ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ↄ ㉯

D15	D14	D13	D12	D11	D10	D9	D8
					CONST TYPE2	CONST TYPE1	CONST TYPE0

D7	D6	D5	D4	D3	D2	D1	D0
	DOWN TYPE2	DOWN TYPE1	DOWN TYPE0		UP TYPE2	UP TYPE1	UP TYPE0

ù ¹ ☿ ᵖ ♩ ─ ⁿᵐ │ H'000 (▪ fi ♭ ↄ ! ▬ fi ⁿᵒ) ≠ Δ ⁹

D2--D0 UP TYPE2--0

UP DRIVE ◖ ∿ fi ♪ ─ ♪ ! ▬Ķ CHANGE Δ ╢ ⇔ ╕ Δ ⁹

D6--D4 DOWN TYPE2--0

DOWN DRIVE ◖ ∿ fi ♪ ─ ♪ ! ▬Ķ CHANGE Δ ╢ ⇔ ╕ Δ ⁹

D10--D8 CONST TYPE2--0

CONST DRIVE ◖ ∿ fi ♪ ─ ♪ ! ▬Ķ CHANGE Δ ╢ ⇔ ╕ Δ ⁹

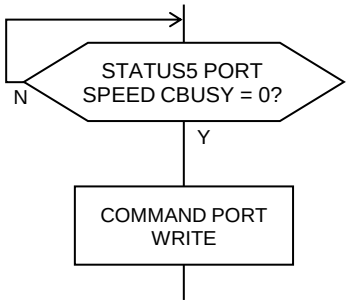
TYPE2	TYPE1	TYPE0	♪ ! ▬Ķ CHANGE Δ ╢ ⁿᵒ ↄ ⁿ ⁿ ⁿ >
0	0	0	♪ ! ▬Ķ CHANGE ◖ ∿ fi ♪ ─ ⅝ ╥ ≠ Δ ╢
0	0	1	
0	1	0	STATUS5 PORT ─ SS0 = 1 ≠ Δ ╢ ( )
0	1	1	STATUS5 PORT ─ SS1 = 1 ≠ Δ ╢ ( )
1	0	0	
1	0	1	
1	1	0	
1	1	1	

( ) SS0, SS1 ⁿᵒ │ SPEC INITIALIZE2 ◖ ∿ fi ♪ ≠ ⁸ ⁿᵒ ⁹ ⌐ ⇔ ≡ ™╢ ⌐ ≠ Δ ⁹

(2) UP DRIVE

ぇ ― R ⁱ ☻ ╩ ⁸ ╕ ≠ ╕ √ │ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'C4 UP DRIVE COMMAND



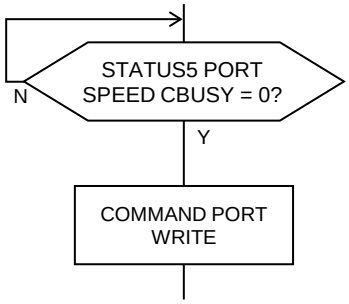
ŋ STATUS5 PORT ― SPEED CBUSY ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

º DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

(3) DOWN DRIVE

ぇ ― R ⁱ ☻ ╩ ⁸ ǧ ╕ ≠ ╕ √ │ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'C5 DOWN DRIVE COMMAND



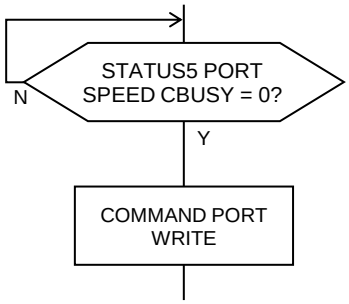
ŋ STATUS5 PORT ― SPEED CBUSY ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

º DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

(4) CONST DRIVE

ぇ ― R ⁱ ☻ ╩ ⁸ ǧ ╕ ≠ ╕ √ │ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'C6 CONST DRIVE COMMAND



ŋ STATUS5 PORT ― SPEED CBUSY ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

º DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

2-1-9. SPEED CHANGE ― ≤

[illegible] SPEED CHANGE [illegible]

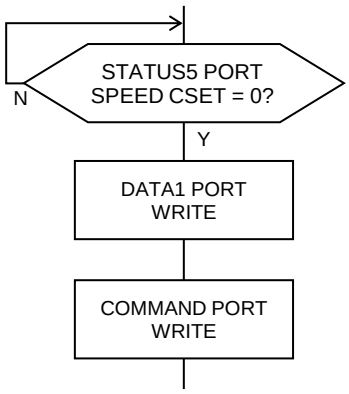
[illegible]

(1) SPEED CHANGE SPEC SET

SPEED CHANGE [illegible]

< [illegible] >

COMMAND : H'C1 SPEED CHANGE SPEC SET COMMAND



[illegible] STATUS5 PORT ― SPEED CSET [illegible]

[illegible] DRIVE DATA1 PORT [illegible]

[illegible] DRIVE COMMAND PORT [illegible] COMMAND [illegible]

DRIVE DATA1 PORT ― [illegible]

D15	D14	D13	D12	D11	D10	D9	D8

D7	D6	D5	D4	D3	D2	D1	D0
					SPEED CHANGE TYPE2	SPEED CHANGE TYPE1	SPEED CHANGE TYPE0

[illegible] H'0 ([illegible])

D0 SPEED CHANGE TYPE0
D1 SPEED CHANGE TYPE1
D2 SPEED CHANGE TYPE2

SPEED CHANGE [illegible]

TYPE2	TYPE1	TYPE0	SPEED CHANGE [illegible]
0	0	0	SPEED CHANGE [illegible]
0	0	1	
0	1	0	STATUS5 PORT ― SS0 = 1 [illegible] ( )
0	1	1	STATUS5 PORT ― SS1 = 1 [illegible] ( )
1	0	0	
1	0	1	
1	1	0	
1	1	1	

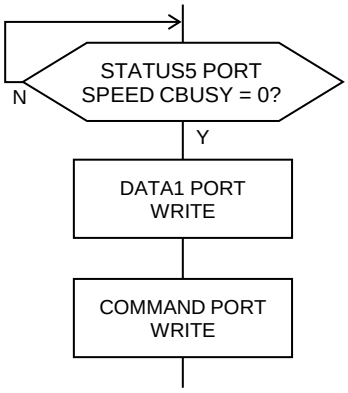
( ) SS0, SS1 [illegible] SPEC INITIALIZE2 [illegible]

(2) SPEED CHANGE

[illegible]

< [illegible] >

COMMAND : H'C8 SPEED CHANGE COMMAND



[illegible] STATUS5 PORT — SPEED CBUSY [illegible] "0" [illegible]

[illegible] DRIVE DATA1 PORT [illegible] 1 32,767 H'0001 H'7FFF [illegible]

[illegible] DRIVE COMMAND PORT [illegible] COMMAND [illegible]

DRIVE DATA1 PORT — [illegible]

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	D14	←					SPEED CHANGE [illegible]							→	D0

[illegible] SPEED CHANGE [illegible] — [illegible] "0" — [illegible] "1" [illegible]
SPEED CHANGE — Hz SPEED CHANGE [illegible] x RESOL

[illegible] SPEED CHANGE [illegible]

2-1-10. RATE CHANGE ─ ≤

RATE CHANGE ⌡ │ ⁸ ☻ Ḥ ↄ ♪ ─ ♪ ! ▬ Ḳ CHANGE ⌡ ─ ≤ ⌐ ⇔ ╕ Δ ⁹

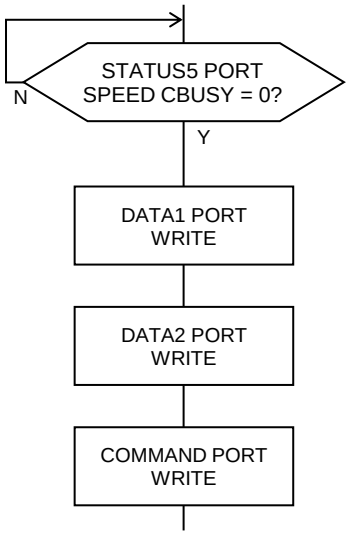
(1) RATE CHANGE

♪ ! ▬ Ḳ CHANGE ⌠ ─ ♦ ↄ ↑ ⁸ ⇔ √ ♦ ↄ ↑ ⌐ ⇔ ╕ Δ ⁹

< ◦ ↄ ▸ fi ☻ >

COMMAND : H'CA

RATE CHANGE COMMAND



ŋ STATUS5 PORT ─ SPEED CBUSY ḥ ! ⁻ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

o DRIVE DATA1 PORT ⌐ ○ ↄ Ḳ ─ ♦ ↄ ↑ ╩ ⅝ ╥ ╕ Δ ⁹
│ ⁸ 1 16,383 H'0000 H'3FFF ≠ Δ ⁹

ↄ DRIVE DATA2 PORT ⌐ ○ ↄ Ḳ ─ ♦ ↄ ↑ ╩ ⅝ ╥ ╕ Δ ⁹
│ ⁸ 1 16,383 H'0000 H'3FFF ≠ Δ ⁹

^ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╥ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ↄ ↑

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
		D13	←					UCYCLE						→	D0

DRIVE DATA2 PORT ─ ♦ ↄ ↑

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
		D13	←					DCYCLE						→	D0

n RATE CHANGE ♦ ↄ ↑ ─ ⁻ ⅜ "0" ─ │ ⁸ "1" ⌐ ⇔ ╕ Δ ⁹
○ ↄ Ḳ ─ (εs) UCYCLE x 0.5 εs 0.5 εs 8.1915 ms
○ ↄ Ḳ ─ (εs) DCYCLE x 0.5 εs 0.5 εs 8.1915 ms

n RATE CHANGE ◖ ∿ fi ♪ ╩ ⇔ ≡ ╙ ⁸ Ḷ ! ω ↄ ↑ ─ │ ╥ ╡ ╕ ∑ ╪ ⁹

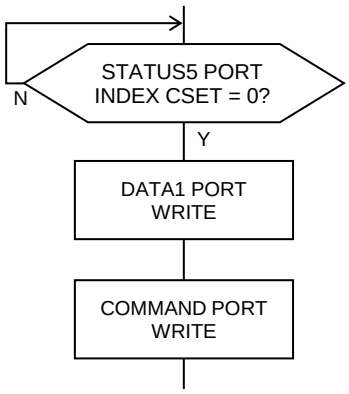
2-1-11. INDEX CHANGE ─ ≤

INDEX CHANGE

(1) INDEX CHANGE SPEC SET

INDEX CHANGE

COMMAND : H'C3 INDEX CHANGE SPEC SET COMMAND



STATUS5 PORT ─ INDEX CSET

DRIVE DATA1 PORT

DRIVE COMMAND PORT COMMAND

DRIVE DATA1 PORT ─

D15	D14	D13	D12	D11	D10	D9	D8

D7	D6	D5	D4	D3	D2	D1	D0
					INDEX CHANGE TYPE2	INDEX CHANGE TYPE1	INDEX CHANGE TYPE0

H'0

D0 INDEX CHANGE TYPE0
D1 INDEX CHANGE TYPE1
D2 INDEX CHANGE TYPE2

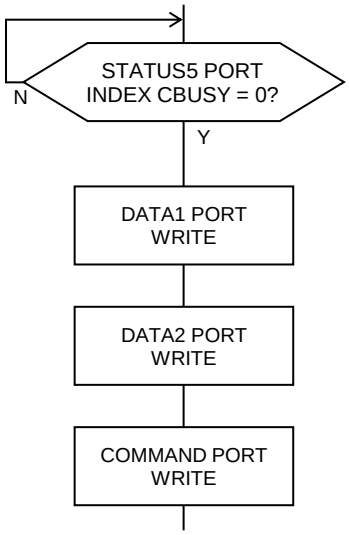
INDEX CHANGE

TYPE2	TYPE1	TYPE0	INDEX CHANGE
0	0	0	INDEX CHANGE ─
0	0	1	
0	1	0	STATUS5 PORT ─ SS0 = 1 ( )
0	1	1	STATUS5 PORT ─ SS1 = 1 ( )
1	0	0	
1	0	1	
1	1	0	
1	1	1	

( ) SS0, SS1 SPEC INITIALIZE2

(2) INC INDEX CHANGE

INC INDEX



COMMAND : H'CC

INC INDEX CHANGE COMMAND

STATUS5 PORT INDEX CBUSY "0"

DRIVE DATA1 PORT D15--D0

DRIVE DATA2 PORT D31--D16

-2,147,483,648 +2,147,483,647 H'8000_0000 H'7FFF_FFFF

CW CCW 2

DRIVE COMMAND PORT COMMAND

DRIVE DATA1 PORT

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←						INDEX CHANGE							→	D0

DRIVE DATA2 PORT

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←						INDEX CHANGE							→	D16

INC INDEX CHANGE (CW) H'0000_0000 H'7FFF_FFFF

INC INDEX CHANGE (CCW) H'8000_0000 H'FFFF_FFFF

INC INDEX CHANGE INDEX CHANGE

INC INDEX CHANGE INDEX COUNTER 32

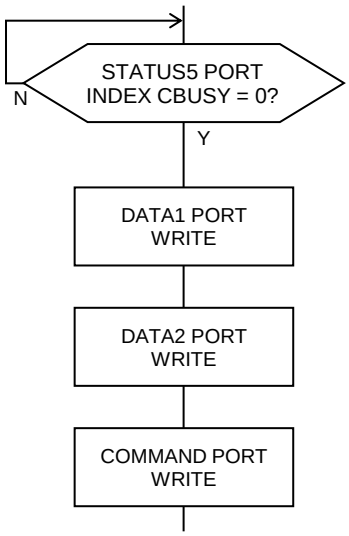
(3) ABS INDEX CHANGE

⇔ √ ♦ ⊃ ㉯ ╩ ⁸ ▪ ♪ ☻ ○ ▶ fi ㉯ ≠ ⇔ ≡ ™ ╫ ▪ ♪ ☻ ─ ⊃ ₀ ⌐ ⇔ ≡ ⁸
ABS INDEX ♪ ! ▬ Ķ ╩ ™ ╕ Δ ⁹

< ◦ ⊃ fi ☻ >

COMMAND : H'CD

ABS INDEX CHANGE COMMAND



ŋ STATUS5 PORT ─ INDEX CBUSY ḥ ! ⅜ "0" ≠ № ╫ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

º DRIVE DATA1 PORT ⌐ ─ ▪ ♪ ☻ D15--D0 ╩ ⅝ ╖ ╕ Δ ⁹

ɔ DRIVE DATA2 PORT ⌐ ─ ▪ ♪ ☻ D31--D16 ╩ ⅝ ╖ ╕ Δ ⁹
│ ⁸
-2,147,483,648 +2,147,483,647 H'8000_0000 H'7FFF_FFFF ≠ Δ ⁹
─ │ 2 ─ ⌐ ⇔ ╕ Δ ⁹
ˆ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ⊃ ㉯

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←					INDEX CHANGE ♦ ⊃ ㉯								→	D0

DRIVE DATA2 PORT ─ ♦ ⊃ ㉯

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←					INDEX CHANGE ♦ ⊃ ㉯								→	D16

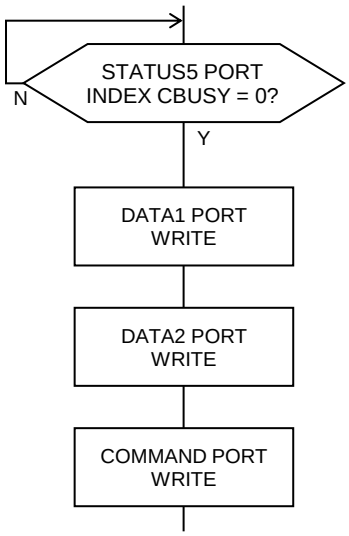
ה Δ ╫ ▪ ♪ ☻ │ ⁸ ▪ ♪ ☻ ○ ▶ fi ㉯ ≠ ⇔ ≡ ™ ╫ ▪ ♪ ☻ ≠ Δ ⁹

(4) PLS INDEX CHANGE

INC INDEX

COMMAND : H'CE

PLS INDEX CHANGE COMMAND



STATUS5 PORT INDEX CBUSY "0"

DRIVE DATA1 PORT D15--D0

DRIVE DATA2 PORT D31--D16

-2,147,483,648 +2,147,483,647 H'8000_0000 H'7FFF_FFFF

CW CCW

DRIVE COMMAND PORT COMMAND

DRIVE DATA1 PORT

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←						INDEX CHANGE							→	D0

DRIVE DATA2 PORT

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←						INDEX CHANGE							→	D16

2-1-12. [illegible]

[illegible]

[illegible]

(1) SLOW STOP

[illegible]

< [illegible] fi ☻ > COMMAND : H'FE SLOW STOP COMMAND

COMMAND PORT WRITE

[illegible] DRIVE COMMAND PORT [illegible] COMMAND [illegible] [illegible]

[illegible] STATUS1 PORT [illegible] STBY = 1 [illegible] DRIVE = 1 [illegible] [illegible]

(2) FAST STOP

[illegible]

< [illegible] fi ☻ > COMMAND : H'FF FAST STOP COMMAND

COMMAND PORT WRITE

[illegible] DRIVE COMMAND PORT [illegible] COMMAND [illegible] [illegible]

[illegible] STATUS1 PORT [illegible] STBY = 1 [illegible] DRIVE = 1 [illegible] [illegible]

[illegible] FAST STOP [illegible] BUSY = 0 [illegible]

2-1-13. ◘ ⊃ W ─

(1) SIGNAL OUT

DRST ☺ ⅛ ╟ ☺ ≤ ⇔ ≡ ↕ ╢ √ ″ ʒ ⁱ ╩ ⇔ ╕ Δ ⁹

╕ √ ⁸ OUT A,B ⅛ ╟ ☺ ≤ ⇔ ≡ ↕ ╢ √ ″ ʒ ⁱ ╩ ⇔ ╕ Δ ⁹

↓ ─ ◗ ∿ fi ♪ ─ │ ≠ Δ ⁹

< ◦ ⊃ ▪ fi ☻ >

COMMAND : H'FC

SIGNAL OUT COMMAND

DATA1 PORT WRITE

ŋ DRIVE DATA1 PORT ⌐ ″ ʒ ⁱ ╩ ⅝ ╖ ╕ Δ ⁹

COMMAND PORT WRITE

º DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ⊃ ㉻

D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	0	0	0	0

D7	D6	D5	D4	D3	D2	D1	D0
DRST OUT	0	OUT B OUT	OUT A OUT	0	0	0	0

ℹ ⁱ ☿ ℘ ♩ ─ ☺ │ H'0000 (すべて OFF レベル出力) ≠ Δ ⁹

D4 OUT A OUT

D5 OUT B OUT

☻ ♥ ⊃ ㉻ ☻ ☺ ⅜ Δ ╢″ ʒ ⁱ ╩ ⇔ ╕ Δ ⁹

0 OFF″ ʒ ⁱ

1 ▪ ◒ ♥ ▫ Ϟ″ ʒ ⁱ

ה OUT A,B OUT │ ☻ ♥ ⊃ ㉻ ☻ ╩ ⁸ ⁹ ⌐ ⇔ ≡ ™ ╢ ⌐ ≠ Δ ⁹

OUT A,B ☺ ─ │ HARD INITIALIZE1 ◗ ∿ fi ♪ ≠ ⇔ ╕ Δ ⁹

D7 DRST OUT

☺ ⅜ Δ ╢″ ʒ ⁱ ╩ ⇔ ╕ Δ ⁹

0 OFF″ ʒ ⁱ (HIGH″ ʒ ⁱ)

1 ▪ ◒ ♥ ▫ Ϟ″ ʒ ⁱ (LOW″ ʒ ⁱ)

ה DRST OUT │ $\overline{\text{DRST}}$ ☺ ─ ╩ ⁸ ⁹ ⌐ ⇔ ≡ ™ ╢ ⌐ ≠ Δ ⁹

$\overline{\text{DRST}}$ ☺ ─ │ SPEC INITIALIZE3 ◗ ∿ fi ♪ ≠ ⇔ ╕ Δ ⁹

2-1-14. ◄ ¡ ↑ ─ ≤ ╗ ⇔

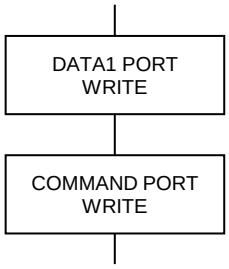
(1) ERROR STATUS MASK

ERROR ⌐ Δ ╢ ERROR STATUS ╩ ☜ ⌐ ∨ ☻ ▬ ⇔ ╕ Δ ⁹
↓ ─ ◗ ∨ fi ♪ ─ │ ≠ Δ ⁹

< ◦ ↑ ◗ fi ☻ >

COMMAND : H'E5

ERROR STATUS MASK COMMAND



ⁿ DRIVE DATA1 PORT ⌐ ∨ ☻ ▬ ♦ ↑ ⁿ ╩ ⅝ ╗ ╕ Δ ⁹

º DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╗ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ↑ ⁿ

D15	D14	D13	D12	D11	D10	D9	D8
FSSTOP ERROR MASK	1	DALM ERROR MASK	PULSE OVF ERROR MASK	ADDRESS OVF ERROR MASK	SSEND ERROR MASK	LSEND ERROR MASK	FSEND ERROR MASK

D7	D6	D5	D4	D3	D2	D1	D0
EXT PULSE ERROR MASK	CPP STOP ERROR MASK	CHANGE CLR ERROR MASK	0	0	0	COMREG CLR ERROR MASK	COMMAND ERROR MASK

ù ¹ ⁸ ⁹ ─ ⁷ │ H'FE00 ≠ Δ ⁹

D15-D0 ∨ ☻ ▬ ♦ ↑ ⁿ

ERROR ⌐ Δ ╢ ERROR STATUS ─ ∨ ☻ ▬ ♦ ↑ ⁿ ╩ ⇔ ╕ Δ ⁹
0 ∨ ☻ ▬ ⇔ ⌂ ™
1 ∨ ☻ ▬ Δ ╢

ה ERROR │ ⁸ ERROR ⌐ Δ ╢ ERROR STATUS ─ OR ≠ Δ ⁹
∨ ☻ ▬ ⇔ √ ERROR STATUS ─ │ ⁸ "0" ⌐ ⌂ ╡ ╕ Δ ⁹

ה ∨ ☻ ▬ ⇔ ≡ ╚ ⁸ ERROR STATUS │ ▬ ¹ ▪ ↕ ╢ ╕ ∑ ╪ ⁹
ERROR STATUS ╩ ▬ ¹ ▪ Δ ╢ ≤ ⅝ │ ⁸ ERROR STATUS CLR ◗ ∨ fi ♪ ╩ ⇔ ≡ ↑ ∞ ↑ ™ ⁹

ה D4,D3,D2 ─ ERROR STATUS │ ⁸ ∨ ☻ ▬ ≠ ⅝ ╕ ∑ ╪ ⁹
D15-D9 ─ ERROR STATUS │ ⁸ ¹ ⁸ ⁹ ─ ⁷ ≠ │ ⁸ ∨ ☻ ▬ Δ ╢ ⁹ ≠ Δ ⁹

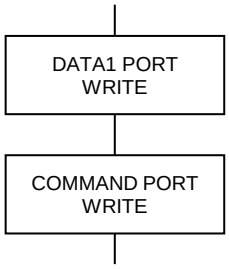
(2) ERRINT STATUS MASK

ERRINT ⌐ Δ ╢ ERROR STATUS ╩ ⌐ ⇔ ╕ Δ ⁹
↓ ─ ◗ ∨ fi ♪ ─ │ ≠ Δ ⁹

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'E6

ERRINT STATUS MASK COMMAND



ⁿ DRIVE DATA1 PORT ⌐ ∨ ☻ ▬ ♦ ↄ ㉯ ╩ ⅝ ╖ ╕ Δ ⁹

º DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ↄ ㉯

D15	D14	D13	D12	D11	D10	D9	D8
FSSTOP ERROR MASK	1	DALM ERROR MASK	PULSE OVF ERROR MASK	ADDRESS OVF ERROR MASK	SSEND ERROR MASK	LSEND ERROR MASK	FSEND ERROR MASK

D7	D6	D5	D4	D3	D2	D1	D0
EXT PULSE ERROR MASK	CPP STOP ERROR MASK	CHANGE CLR ERROR MASK	INDEX CHANGE ERROR MASK	ABS INDEX ERROR MASK	INC INDEX ERROR MASK	COMREG CLR ERROR MASK	COMMAND ERROR MASK

ㄱ ㄷ ㅎ ㅣ ㉮ ─ ㉷ │ H'FFFF (㉷ ≡ ∨ ☻ ▬ ╢ Δ) ≠ Δ ⁹

D15-D0 ∨ ☻ ▬ ♦ ↄ ㉯

ERRINT ⌐ Δ ╢ ERROR STATUS ─ ∨ ☻ ▬ ♦ ↄ ㉯ ╩ ⇔ ╕ Δ ⁹
0 ∨ ☻ ▬ ⇔ ⌂ ™
1 ∨ ☻ ▬ Δ ╢

ㅁ ERRINT │ ⁸ ERRINT ⌐ Δ ╢ ERROR STATUS ─ OR ≠ Δ ⁹
∨ ☻ ▬ ⇔ √ ERROR STATUS ─ │ ⁸ "0" ⌐ ⌂ ╡ ╕ Δ ⁹

ㅁ ∨ ☻ ▬ ⇔ ≡ ╚ ⁸ ERROR STATUS │ ▬ ⁱ ▪ ↕ ╢ ╕ ∑ ╪ ⁹
ERROR STATUS ╩ ▬ ⁱ ▪ Δ ╢ ≤ ⅝ │ ⁸ ERROR STATUS CLR ◗ ∨ fi ♪ ╩ ⇔ ≡ ↄ ∞ ↕ ™ ⁹

ㅁ ERRINT │ INT ㉰ ⅛ ╣ ≠ ⅝ ╕ Δ ⁹

(3) ERROR STATUS CLR

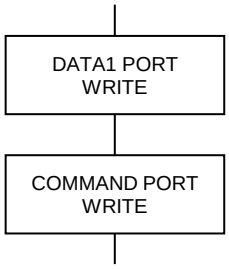
ERROR STATUS ╩ ج ⌐ ▪ ⇔ ╕ Δ ⁹

↓ ─ ◖ ∀ fi ♪ ─ │ ≠ Δ ⁹

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'E4

ERROR STATUS CLR COMMAND



ⁿ DRIVE DATA1 PORT ⌐ ▪ ♦ ↄ ⁿ ╩ ⅝ ╖ ╕ Δ ⁹

º DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ♦ ↄ ⁿ

D15	D14	D13	D12	D11	D10	D9	D8
FSSTOP ERROR CLR	0	DALM ERROR CLR	PULSE OVF ERROR CLR	ADDRESS OVF ERROR CLR	SSEND ERROR CLR	LSEND ERROR CLR	FSEND ERROR CLR

D7	D6	D5	D4	D3	D2	D1	D0
EXT PULSE ERROR CLR	CPP STOP ERROR CLR	CHANGE CLR ERROR CLR	INDEX CHANGE ERROR CLR	ABS INDEX ERROR CLR	INC INDEX ERROR CLR	COMREG CLR ERROR CLR	COMMAND ERROR CLR

D15-D0 ▪ ♦ ↄ ⁿ

ERROR STATUS ─ ▪ ♦ ↄ ⁿ ╩ ⇔ ╕ Δ ⁹

0 ▪ ⇔ ⌂ ™

1 ▪ Δ ╢

ה ◖ ∀ fi ♪ ─ ≠ ⁸ ERROR STATUS ╩ ▪ ⇔ ╕ Δ ⁹

D15--D8 ─ ERROR STATUS │ ⁸ ⅜ ⇔ ≡ ™ ╢ │ ▪ ↕ ╢ ╕ ∑ ╪ ⁹

ה ↓ ─ ◖ ∀ fi ♪ ─ ♦ ↄ ⁿ │ ⁸ ◖ ∀ fi ♪ ─ ╖ ≠ Δ ⁹ ⌋ ⁱ ● ظص

(4) ERROR STATUS READ

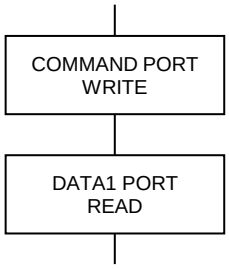
15 ⌠ ─ ERROR STATUS ⌡ ╖ ⇔ ╕ Δ ⁹

↓ ─ ◗ ∀ fi ♪ ─ │ ≠ Δ ⁹

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'D1

ERROR STATUS READ COMMAND



ⁿ DRIVE COMMAND PORT ⌐ COMMAND ⌡ ⅝ ╖ ╕ Δ ⁹

º DRIVE DATA1 PORT ⅛ ╟ ERROR STATUS ⌡ ╖ ⇔ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ╖ ⇔ ♦ ↄ ⅊ ERROR STATUS

D15	D14	D13	D12	D11	D10	D9	D8
FSSTOP ERROR	0	DALM ERROR	PULSE OVF ERROR	ADDRESS OVF ERROR	SSEND ERROR	LSEND ERROR	FSEND ERROR

D7	D6	D5	D4	D3	D2	D1	D0
EXT PULSE ERROR	CPP STOP ERROR	CHANGE CLR ERROR	INDEX CHANGE ERROR	ABS INDEX ERROR	INC INDEX ERROR	COMREG CLR ERROR	COMMAND ERROR

ERROR STATUS │ ⁸ "1" ≠ ◀ ! ↄ ⅜ ⇔ √ ↓ ≤ ⌡ ⇔ ╕ Δ ⁹

D0 COMMAND ERROR

─ ◗ ∀ fi ♪ ⌡ ⇔ √ ↓ ≤ ⌡ ⇔ ╕ Δ ⁹

⌠ ─ ⁞ │ ⁸ ◀ ! ↄ ⌐ ⌂ ╡ ∑ ╪ ⁹

ⓗ ─ ◗ ∀ fi ♪ ⌡ ⇔ √

ⓗ SPEED CSET = 1 ─ ≤ ⅝ ⌐ ⁸ ☻ Ḥ ↄ ♪ ─ ♪ ! ▬ Ḳ CHANGE ◗ ∀ fi ♪ ⌡ ⇔ √

ⓗ SPEED CBUSY = 1 ─ ≤ ⅝ ⌐ ⁸ ☻ Ḥ ↄ ♪ ─ ♪ ! ▬ Ḳ CHANGE ◗ ∀ fi ♪ ⌡ ⇔ √

ⓗ INDEX CSET = 1 ─ ≤ ⅝ ⌐ ⁸ INDEX CHANGE ◗ ∀ fi ♪ ⌡ ⇔ √

ⓗ INDEX CBUSY = 1 ─ ≤ ⅝ ⌐ ⁸ INDEX CHANGE ◗ ∀ fi ♪ ⌡ ⇔ √

ⓗ COMREG FL = 1 ─ ≤ ⅝ ⌐ ⁸ ◗ ∀ fi ♪ ⌡ ⇔ √

D1 COMREG CLR ERROR

⁹ Δ ╕ ⇔ ⌡ ≤ ↓ √ ⇔ ▪ ⁱ ⌡ ♪ fi ∀ ◗ ⌡ ─ ⌊ ♪ fi ∀ ◗ ≠ ⇔ ≡ ™ ╢

D2 INC INDEX ERROR

⁹ Δ ╕ ⇔ ⌡ ≤ ↓ √ ⇔ ⌡ ▬ Ḳ ! ♪ ⁸ INC INDEX ⌡ ≠ ↄ ⌡ ⁊ ⌐ ⁸ ☻ ○ ↄ ⅊ ─ ○ ↄ ⅊ ⁊ ♪ ▪

ⓗ ⁸ INC INDEX CHANGE ⌐ ≤ ⅝ ⁸ ☻ ⅜ ○ ↄ ⅊ ⁊ ♪ ▪ ⇔ ≡ ™ ╢ ↄ ⌡ ⇔ √

D3 ABS INDEX ERROR

⁹ Δ ╕ ⇔ ⌡ ≤ ↓ √ ⇔ ⌡ ▬ Ḳ ! ♪ ⁸ ABS INDEX ⌡ ≠ ↄ ⌡ ⁊ ⌐ ⁸ ▶ ○ ☻ fi ⅊ ─ ○ ↄ ⅊ ⁊ ♪ ▪

ⓗ ABS INDEX ♪ ! ▬ Ḳ ↄ ⌐ ⁸ ▪ ♪ ⁊ ☻ ○ ▶ fi ⅊ ─ ○ ↄ ⅊ ⁊ ⌡ ⇔ √

ⓗ ▪ ♪ ⁊ ☻ ○ ▶ fi ⅊ ⅜ ○ ↄ ⅊ ⁊ ⌡ ↄ ⇔ ≡ ™ ╢ ≤ ⅝ ⌐ ⁸ ABS INDEX CHANGE ⌡ ⇔ √

D4 INDEX CHANGE ERROR

⁹ Δ ╕ ⇔ ⌡ ≤ ↓ √ ⇔ ⌡ ⌂ INDEX CHANGE ⌡ ⅜ ⌡

ⓗ ⌡ ⅜ ⌂ INDEX CHANGE ⌡ ⇔ √

ⓗ ABS INDEX ♪ ! ▬ Ḳ ↄ ⌐ ⁸ ▪ ♪ ⁊ ☻ ○ ▶ fi ⅊ ─ º ⅜ ↕ ╢ ⁸ ⌡ ⅜ ⌐ ⌂ ∫ √

D5 CHANGE CLR ERROR

⁹ Δ ╕ ⇔ ⌡ ≤ ↓ √ ⇔ ⌐ ⌡ ─ INDEX CHANGE ⌡ ⌊

D6 CPP STOP ERROR

♪ ! ▬ Ką — ω ▬ fi — CPP STOP ≠ ♪ ! ▬ Ką ╩ ğ ⇔ √ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D7 EXT PULSE ERROR

Ꞧ ⁱ ☻ ╩ ⌐ ⁸ ⌂ Ꞧ ⁱ ☻ ⅜ ≠ ⅝ ⌂ ⅛ ∫ √ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ה ▪ ▾ ♥ ▫ Ką — ىي — ⌐ ⁸ ○ — ▶ fi ♩ ▬ ⊢ fi ◓ ⅜ ⇔ √

D8 FSEND ERROR

BUSY = 1 — ≤ ⅝ ⌐ ⁸ STATUS1 PORT — FSEND = 1 ╩ ⇔ √ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D9 LSEND ERROR

BUSY = 1 — ≤ ⅝ ⌐ ⁸ STATUS1 PORT — LSEND = 1 ╩ ⇔ √ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D10 SSEND ERROR

BUSY = 1 — ≤ ⅝ ⌐ ⁸ STATUS1 PORT — SSEND = 1 ╩ ⇔ √ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D11 ADDRESS OVF ERROR

BUSY = 1 — ≤ ⅝ ⌐ ⁸ STATUS4 PORT — ADDRESS OVF = 1 ╩ ⇔ √ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D12 PULSE OVF ERROR

STATUS4 PORT — PULSE OVF = 1 ╩ ⇔ √ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D13 DALM ERROR

STATUS2 PORT — DALM = 1 ╩ ⇔ √ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

D15 FSSTOP ERROR

STATUS2 PORT — FSSTOP = 1 ╩ ⇔ √ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ה ERROR STATUS READ ◖ ∀ fi ♪ ╩ Δ ╢ ≤ ⁸ ERROR STATUS D15--D0 ╩ DRIVE DATA1 PORT (READ) ⌐ ☿ ♇ ♩ ⇔ ╕ Δ ⁹

ה FSEND, LSEND, SSEND ḫ ! ◓ ⅜ "1" ≠ ╚ ⁸ — BUSY = 0 ǵ 1 ≠ │ ◀ ! ↕ ╢ ╕ ∑ ╪ ⁹

BUSY = 0 ǵ 1 ≤ ⌐ ⁸ FSEND, LSEND, SSEND = 1 ǵ 0 ⌐ ⌂ ╡ ╕ Δ ⁹

2-1-15. [illegible]

(1) INT FACTOR MASK

INT [illegible] 12 [illegible]

[illegible]

[illegible]

COMMAND : H'E1 INT FACTOR MASK COMMAND

DATA1 PORT WRITE

[illegible] DRIVE DATA1 PORT [illegible]

COMMAND PORT WRITE

[illegible] DRIVE COMMAND PORT [illegible] COMMAND [illegible]

DRIVE DATA1 PORT [illegible]

D15	D14	D13	D12	D11	D10	D9	D8
1	1	1	1	ERRINT INT MASK	DFLINT INT MASK	CNTINT INT MASK	ADRINT INT MASK

D7	D6	D5	D4	D3	D2	D1	D0
SS1 INT MASK	SS0 INT MASK	DALM INT MASK	MAN INT MASK	COMREG FL INT MASK	COMREG EP INT MASK	STBY INT MASK	RDYINT INT MASK

[illegible] H'FFFF ([illegible])

D15--D0 [illegible]

12 [illegible]

0 [illegible]

1 [illegible]

[illegible] INT [illegible] 12 [illegible] OR [illegible]

[illegible] "0" [illegible]

[illegible] INT FACTOR CLR [illegible]

(2) INT FACTOR CLR

INT [illegible] 12 [illegible]

[illegible]

[illegible] COMMAND : H'E0 INT FACTOR CLR COMMAND

DATA1 PORT WRITE

COMMAND PORT WRITE

[illegible] DRIVE DATA1 PORT [illegible]

[illegible] DRIVE COMMAND PORT [illegible] COMMAND [illegible]

DRIVE DATA1 PORT [illegible]

D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	ERRINT INT CLR	DFLINT INT CLR	CNTINT INT CLR	ADRINT INT CLR

D7	D6	D5	D4	D3	D2	D1	D0
SS1 INT CLR	SS0 INT CLR	DALM INT CLR	MAN INT CLR	COMREG FL INT CLR	COMREG EP INT CLR	STBY INT CLR	RDYINT INT CLR

D15-D0 [illegible]

12 [illegible]

0 [illegible]

1 [illegible]

[illegible]

[illegible]

(3) INT FACTOR READ

INT ─ 12 ڽ ─ ╡ ╖ ╨ ╖ ⇔╕ Δ ⁹
↓ ─ ◖ ∛ fi ♪ ─ │ ≢ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'D0 INT FACTOR READ COMMAND

COMMAND PORT WRITE

ŋ DRIVE COMMAND PORT ⌐ COMMAND ╨ ⅝ ╖╕ Δ ⁹

DATA1 PORT READ

º DRIVE DATA1 PORT ⅛ ╠ ERROR STATUS ╨ ╖ ⇔╕ Δ ⁹

DRIVE DATA1 PORT ─ ╖ ⇔ ♦ ↄ ♃ INT FACTOR

D15	D14	D13	D12	D11	D10	D9	D8
0	0	0	0	ERRINT INT	DFLINT INT	CNTINT INT	ADRINT INT

D7	D6	D5	D4	D3	D2	D1	D0
SS1 INT	SS0 INT	DALM INT	MAN INT	COMREG FL INT	COMREG EP INT	STBY INT	RDYINT INT

ה INT FACTOR │ ⁸ "1" ≢ ╡ ╖ ⅜ ⇔ √ ↓ ≤ ╨ ⇔╕ Δ ⁹

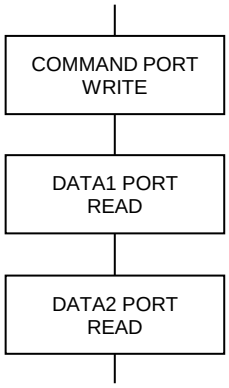
ה INT FACTOR READ ◖ ∛ fi ♪ ╨ Δ ╢ ≤ ⁸ INT FACTOR D15--D0 ╨ DRIVE DATA1 PORT READ ⌐ ☿ ୧ ♩ ⇔╕ Δ ⁹

2-1-16. ♦ ⊃ ㉯ ─ ╖ ⇔

(1) MCC SPEED READ

MCC07E ⅜ ⇔ ≡ ™ ╢ ♪ ! ▬ Ķ Ꝑ ⁱ ☻ ╩ ╖ ⇔ ╕ Δ ⁹
↓ ─ ◖ ∿ fi ♪ ─ │ ≠ Δ ⁹

< ◦ ⊃ ◔ fi ☻ > COMMAND : H'D4 MCC SPEED READ COMMAND



ŋ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

º DRIVE DATA1 PORT ⅛ ╟ Ꝑ ⁱ ☻ ♦ ⊃ ㉯ D15--D0 ╩ ╖ ⇔ ╕ Δ ⁹

ª DRIVE DATA2 PORT ⅛ ╟ Ꝑ ⁱ ☻ ♦ ⊃ ㉯ D16--D31 ╩ ╖ ⇔ ╕ Δ ⁹

DRIVE DATA1 PORT ─ ╖ ⇔ ♦ ⊃ ㉯

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←					Ꝑ ⁱ ☻		♦ ⊃ ㉯						→	D0

DRIVE DATA2 PORT ─ ╖ ⇔ ♦ ⊃ ㉯

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←					Ꝑ ⁱ ☻		♦ ⊃ ㉯						→	D16

ה ╖ Δ ♦ ⊃ ㉯ │ ⁸ ⁸ ♪ ! ▬ Ķ Ꝑ ⁱ ☻ Hz ─ 10 ⁹ ─ Ꝑ ⁱ ☻ ♦ ⊃ ㉯ ≠ Δ ⁹
♪ ! ▬ Ķ Ꝑ ⁱ ☻ Hz Ꝑ ⁱ ☻ ♦ ⊃ ㉯ 10

ה MCC SPEED READ ◖ ∿ fi ♪ ╩ Δ ╢ ≤ ⁸ MCC07E ⅜ ⇔ ≡ ™ ╢ ♪ ! ▬ Ķ Ꝑ ⁱ ☻ ─
10 ─ ♦ ⊃ ㉯ ╩ DRIVE DATA1, 2 PORT READ ⌐ ⁿ ⁿ ♩ ⇔ ╕ Δ ⁹

ה ♪ ! ▬ Ķ ⊃ │ ⁸ ω ▬ fi ─ Ꝑ ⁱ ☻ ─ ╖ ⇔ ─ ╖ ≠ Δ ⁹
ω ▬ fi ⅛ ╟ ╖ Δ ♦ ⊃ ㉯ │ ⁸ ♪ ! ▬ Ķ ─ ≤ ⌂ ╢ Ꝑ ⁱ ☻ ─ ≠ Δ ⁹

ה ⊃ ─ │ ⁸ Ꝑ ⁱ ☻ ─ ╖ ⇔ │ ≠ Δ ⁹
STATUS1 PORT ─ DRIVE = 0 ─ ≤ ⅝
STATUS1 PORT ─ EXT PULSE = 1 ─ ≤ ⅝ Ꝑ ⁱ ☻ ─ ⊃

(2) MCC SET DATA READ

MCC07E ⌐ ⇔ √ ♦ ↗ ⊥ ╖ ⇔ ╕ Δ ⁹
↓ ─ ◖ ∀ fi ♪ ─ │ ≠ Δ ⁹

< ◦ ↗ ◔ fi ☻ > COMMAND : H'D5 MCC SET DATA READ COMMAND

COMMAND PORT WRITE

ŋ DRIVE COMMAND PORT ⌐ COMMAND ⊥ ⅝ ╖ ╕ Δ ⁹

DATA1 PORT READ

º DRIVE DATA1 PORT ⅛ ╟ ⇔ √ COMMAND CODE ─ DATA1 PORT ─
♦ ↗ ⊥ ╖ ⇔ ╕ Δ ⁹

DATA2 PORT READ

ɔ DRIVE DATA2 PORT ⅛ ╟ ⇔ √ COMMAND CODE ─ DATA2 PORT ─
♦ ↗ ⊥ ╖ ⇔ ╕ Δ ⁹

ה ╖ Δ ♦ ↗ │ ⁸ MCC07E ⁿ ≠ ⇔ ≡ ™ ⌂ ™ ♦ ↗ ≠ Δ ⁹

ה ⁱ ☿ ℘ ♩ │ ⁸ ─ ♦ ↗ ─ ⅜ ╖ ↕ ╢ ╕ Δ ⁹

ה SET DATA READ ◖ ∀ fi ♪ ⊥ Δ ╢ ≤ ⇔ √ ◖ ∀ fi ♪ ─ ♦ ↗ ⊥ DRIVE DATA1, 2 PORT
(READ) ⌐ ☿ ℘ ♩ ⇔ ╕ Δ ⁹
◖ ∀ fi ♪ ≠ ⅝ ╖ ⅜ ⌂ DATA PORT ─ ♦ ↗ │ ⁸ "0" ⌐ ⌂ ╢ ╕ Δ ⁹

ù ╖ ⇔ ≠ ⅝ ╢ ♪ ! ▬ Ķ Ꞧ ! ω ↗ ≤ ─ ♦ ↗

COMMAND CODE	◖ ∀ fi ♪	
H'01	SPEC INITIALIZE1	♪ ! ▬ Ķ Ꞧ ⁱ ☻ ─ ث ─
H'02	SPEC INITIALIZE2	CWLM, CCWLM, RDYINT,SS0,SS1 ─
H'03	SPEC INITIALIZE3	DRST, DEND, DALM, STBY, ─
H'05	FSPD SET	Ꞧ ⁱ ☻ ─ Ꞧ ⁱ ☻ ─
H'06	HIGH SPEED SET	♪ ! ▬ Ķ ─ يس ≤ ─
H'07	LOW SPEED SET	♪ ! ▬ Ķ ─ ≤ ﻭ ─
H'08	RATE SET	○ ↗ Ķ ─ ─
H'09	SCAREA SET	○ ↗ Ķ ─ ─
H'0A	DOWN PULSE ADJUST	Ꞧ ⁱ ☻ ─ ○ ɧ ☿ ℘ ♩
H'0C	JSPD SET	JOG ♪ ! ▬ Ķ ─ Ꞧ ⁱ ☻ ─
H'0D	JOG PULSE SET	JOG ♪ ! ▬ Ķ ─ Ꞧ ⁱ ☻ ─
H'0F	ORIGIN SPEC SET	ORIGIN ♪ ! ▬ Ķ ─ ث لأ ─
H'20	CP SPEC SET	CPPOUT ─
H'22	LONG POSITION SET	♪ ! ▬ Ķ ─ ▪ ♪ ☻ ─
H'23	SHORT POSITION SET	♪ ! ▬ Ķ ─ ▪ ♪ ☻ ─
H'28	CIRCULAR XPOSITION SET	محني ♪ ! ▬ Ķ ─ X ▪ ♪ ☻ ─
H'29	CIRCULAR YPOSITION SET	محني ♪ ! ▬ Ķ ─ Y ▪ ♪ ☻ ─
H'2A	CIRCULAR PULSE SET	محني ♪ ! ▬ Ķ ─ Ꞧ ⁱ ☻ ─

COMMAND CODE		
H'81	ADDRESS COUNTER INITIALIZE1	
H'82	ADDRESS COUNTER INITIALIZE2	
H'87	ADDRESS COUNTER MAX COUNT SET	
H'88	ADRINT COMPARE REGISTER1 SET	ADRINT
H'89	ADRINT COMPARE REGISTER2 SET	ADRINT
H'8A	ADRINT COMPARE REGISTER3 SET	ADRINT
H'8C	ADRINT COMP1 ADD DATA SET	ADRINT COMP1 ADD
H'91	PULSE COUNTER INITIALIZE1	
H'92	PULSE COUNTER INITIALIZE2	
H'97	PULSE COUNTER MAX COUNT SET	
H'98	CNTINT COMPARE REGISTER1 SET	CNTINT
H'99	CNTINT COMPARE REGISTER2 SET	CNTINT
H'9A	CNTINT COMPARE REGISTER3 SET	CNTINT
H'9C	CNTINT COMP1 ADD DATA SET	CNTINT COMP1 ADD
H'A1	DFL COUNTER INITIALIZE1	
H'A2	DFL COUNTER INITIALIZE2	
H'A3	DFL COUNTER INITIALIZE3	
H'A8	DFLINT COMPARE REGISTER1 SET	DFLINT
H'A9	DFLINT COMPARE REGISTER2 SET	DFLINT
H'AA	DFLINT COMPARE REGISTER3 SET	DFLINT
H'AC	DFLINT COMP1 ADD DATA SET	DFLINT COMP1 ADD
H'C0	UDC SPEC SET	UP/DOWN/CONST
H'C1	SPEED CHANGE SPEC SET	SPEED CHANGE
H'C3	INDEX CHANGE SPEC SET	INDEX CHANGE
H'E1	INT FACTOR MASK	INT INT FACTOR
H'E5	ERROR STATUS MASK	ERROR ERROR STATUS
H'E6	ERRINT STATUS MASK	ERRINT ERROR STATUS
H'E8	COUNT LATCH SPEC SET	
H'F1	HARD INITIALIZE1	OUTA,OUTB
H'F4	HARD INITIALIZE4	
H'F5	HARD INITIALIZE5	
H'F6	HARD INITIALIZE6	
H'F7	HARD INITIALIZE7	
H'FC	SIGNAL OUT	

* COMMAND CODE H'88, H'98 H'A8 COMPARE REGISTER1 SET [8]

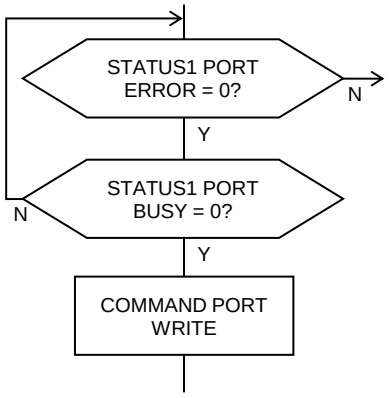
[9]

2-1-17. ノ ーث

(1) NO OPERATION

│ №╡ ╕ ∑ ╪ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'00 NO OPERATION COMMAND



ŋ DRIVE STATUS1 PORT ─ ERROR ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

o DRIVE STATUS1 PORT ─ BUSY ḥ ! ◓ ⅜ "0" ≠ №╢ ↓ ≤ ╩ ⇔ ╕ Δ ⁹

ɔ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

↓ ─ ◖ ∀ fi ♪ ─ ⌐ ╟╡ ⁸ ⁓ ⁗ ─ STATUS ḥ ! ◓ ⅜ ◒ ⁱ ▪ ↕ ╢ ╕ Δ ⁹

ה STATUS1 PORT ─ DRVEND ḥ ! ◓

ה STATUS1 PORT ─ LSEND ḥ ! ◓

ה STATUS1 PORT ─ SSEND ḥ ! ◓

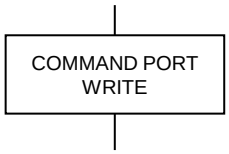
ה STATUS1 PORT ─ FSEND ḥ ! ◓

(2) MCC CHIP RESET

⁓ ─ ≥ └ ╟ ⌐ ⇔ ≡ ╩ ≠ Δ ⁹ ↓ ─ ◖ ∀ fi ♪ ─ │ ≠ Δ ⁹

MCC07E ⁓ ─ ⁓ ─ ⁓ ≡ ─ ♦ ↄ ⁓ ╩ ⇔ ≡ ⁸ ⁱ ⁓ ⁓ ⁓ ≤ ∂ ⌐ ⇔ ╕ Δ ⁹

< ◦ ↄ ◔ fi ☻ > COMMAND : H'F0 MCC CHIP RESET COMMAND



ŋ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

2-2. ○ ▶ fi ㅋ ◖ ∀ fi ♪

2-2-1. ▪ ♪ ● ○ ▶ fi ㅋ —

(1) ADDRESS COUNTER INITIALIZE1

▪ ♪ ● ○ ▶ fi ㅋ — ⇔ ╕ Δ ⁹

↓ — ◖ ∀ fi ♪ — ≠ Δ ⁹

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'81 ADDRESS COUNTER INITIALIZE1 COMMAND

DATA1 PORT WRITE

ⁿ DRIVE DATA1 PORT ⌐ ♦ ↄ ㅋ ⅝ ⅟ ╕ Δ ⁹

COMMAND PORT WRITE

ᵒ DRIVE COMMAND PORT ⌐ COMMAND ⅝ ⅟ ╕ Δ ⁹

DRIVE DATA1 PORT — ♦ ↄ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8
AUTO ADD ENABLE	AUTO CLR ENABLE	COMP GATE TYPE1	COMP GATE TYPE0	ADRINT PULSE TYPE1	ADRINT PULSE TYPE0	ADRINT TYPE1	ADRINT TYPE0

D7	D6	D5	D4	D3	D2	D1	D0
EXT COUNT DIRECTION	EXT PULSE TYPE2	EXT PULSE TYPE1	EXT PULSE TYPE0	EXT COUNT TYPE1	EXT COUNT TYPE0	COUNT PULSE SEL1	COUNT PULSE SEL0

ᵡ ᵢ ᵡ ᵖ ᵢ — ᵗᵐ | H'0030 (▪ fi ᵇ ↄ ᵢ ▬ fi ᵡ) ≠ Δ ⁹

D0 COUNT PULSE SEL0

D1 COUNT PULSE SEL1

○ ▶ fi ㅋ — ○ ▶ fi ᵢ R ᵢ ☻ ⇔ ╕ Δ ⁹

⇔ √ ○ ▶ fi ᵢ R ᵢ ☻ | ⁸ CWP, CCWP ⅛ ⅟ Δ ⁹ ♪ ᵢ ▬ K R ᵢ ☻ ⌐ ⌂ ╕ ╕ Δ ⁹

Xn, Zn, Bn ⌐ Δ

SEL1	SEL0	○ ▶ fi ᵢ R ᵢ ☻	○ ▶ fi ᵢ
0	0	(Xn,Zn,Bn)— R ᵢ ☻ ○ ▶ fi ᵢ Δ	≠ ○ ▶ fi ᵢ ▪ ᵖ k
0	1	ᵗᵇ (Yn,An,Cn)— R ᵢ ☻ ○ ▶ fi ᵢ Δ	≠ ○ ▶ fi ᵢ ᵇ ▶ fi
1	0	(Xn,Zn,Bn)— R ᵢ ☻ ᵏᵗ ○ ▶ fi ᵢ Δ	EXT COUNT DIRECTION
1	1	ᵗᵇ (Yn,An,Cn)— R ᵢ ☻ ᵏᵗ ○ ▶ fi ᵢ Δ	≠

Yn, An, Cn ⌐ Δ

SEL1	SEL0	○ ▶ fi ᵢ R ᵢ ☻	○ ▶ fi ᵢ
0	0	(Yn,An,Cn)— R ᵢ ☻ ≠ ○ ▶ fi ᵢ Δ	≠ ○ ▶ fi ᵢ ▪ ᵖ k
0	1	ᵗᵇ (Xn,Zn,Bn)— R ᵢ ☻ ○ ▶ fi ᵢ Δ	≠ ○ ▶ fi ᵢ ᵇ ▶ fi
1	0	ᵗᵇ (Xn,Zn,Bn)— R ᵢ ☻ ᵏᵗ ○ ▶ fi ᵢ Δ	EXT COUNT DIRECTION
1	1	(Yn,An,Cn)— R ᵢ ☻ ᵏᵗ ≠ ○ ▶ fi ᵢ Δ	≠

ᵐ DRIVE STATUS1 PORT —EXT PULSE = 0 ⁸ BUSY = 1 — ≤ ⅝ ⌐ ⁸ "10", "11" ⁹ ⇔ √ | ⁸

ᵗᵍ — ⇔ √ BUSY = 0 ⌐ ⁸ EXT PULSE = 1 ⁸ BUSY = 1 ⌐ ⌂ ╕ ╕ Δ ⁹

D2 EXT COUNT TYPE0
D3 EXT COUNT TYPE1

TYPE1	TYPE0		
0	0	EA, EB	
0	1	EA, EB	
1	0	EA, EB	
1	1	EA	EB

D4 EXT PULSE TYPE0
D5 EXT PULSE TYPE1
D6 EXT PULSE TYPE2

TYPE1	TYPE1	TYPE0	
0	0	0	100 ns
0	0	1	200 ns
0	1	0	500 ns
0	1	1	1.0 εs

TYPE1	TYPE1	TYPE0	
0	0	0	2.0 εs
0	0	1	5.0 εs
0	1	0	10 εs
0	1	1	20 εs

COUNT PULSE SEL

CWP, CCWP

D7 EXT COUNT DIRECTION

EA, EB

D8 ADRINT TYPE0
D9 ADRINT TYPE1

DRIVE STATUS4 PORT ≤ ADRINT

COMP1, 2, 3

TYPE1	TYPE0	COMP1, 2, 3	
0	0		DRIVE STATUS4 PORT
0	1		DRIVE STATUS4 PORT
1	0		
1	1		INT FACTOR CLR ADRINT INT CLR = 1

ADRINT PULSE TYPE

D10 ADRINT PULSE TYPE0
D11 ADRINT PULSE TYPE1

COMP1, 2, 3 ─ پ ╩ ☻ ⁱ ↄ ⌐ ⇔ √ ≤ ⅝ ─ ⁸ ╩ ⇔ ╕ Δ ⁹

TYPE1	TYPE0	پ ─
0	0	200 ns
0	1	10 εs
1	0	100 εs
1	1	1,000 εs

ה ☻ ⁱ ↄ ⌐ ○ ↓ ↄ ⁱ ▪ ╕ √ │ ╩ ⇔ √ │ ⁸ ↓ ─ ╩ ⇔ ╕ Δ ⁹
↓ ─ │ ⁱ ↓ ⁱ ● ≢ Δ ⁹

D12 COMP GATE TYPE0
D13 COMP GATE TYPE1

ADRINT ⌐ Δ ╢ COMP1, 2, 3 ─ پ ─ ⁸ ╩ ⇔ ╕ Δ ⁹

TYPE1	TYPE0	پ ─
0	0	COMP1 OR (COMP2 OR COMP3)
0	1	COMP1 OR (COMP2 AND COMP3)
1	0	COMP1 AND (COMP2 OR COMP3)
1	1	COMP1 AND (COMP2 AND COMP3)

OR ⁸ AND

D14 AUTO CLEAR ENABLE

COMP1 ─ ○ ↄ ↓ ↄ ⁱ ▪ ≢ ⁸ ○ ▶ fi ㄱ ╩ ₈ ↄ ⁱ ▪ Δ ╢ ↄ ⁱ ▪ ⇔ ⌂ ™ ₉ ╩ ⇔ ╕ Δ ⁹
0 COMP1 ─ پ ≢ ○ ▶ fi ㄱ ╩ ↄ ⁱ ▪ ⇔ ⌂ ™
1 COMP1 ─ پ ≢ ○ ▶ fi ㄱ ╩ ↄ ⁱ ▪ Δ ╢

Ẽ ○ ↄ ↓ ↄ ⁱ ▪
COMP1 ─ پ ≤ ⌐ ⁸ ▪ ♪ ☻ ○ ▶ fi ㄱ ─ ♦ ↄ ㄱ ╩ "0" ⌐ ↄ ⁱ ▪ ⇔ ╕ Δ ⁹
COMP1 ─ پ ⅜ ☻ ⁱ ↄ ─ ≤ ⅝ │ ⁸ پ ─ ╩ ⇔ ╕ Δ ⁹

D15 AUTO ADD ENABLE

COMP1 ─ ≢ ⁸ ♦ ↄ ㄱ ╩ ₈ ⁂ Δ ╢ ⁂ ⇔ ⌂ ™ ₉ ╩ ⇔ ╕ Δ ⁹
0 COMP1 ─ پ ≢ ♦ ↄ ㄱ ╩ ⁂ ⇔ ⌂ ™
1 COMP1 ─ پ ≢ ♦ ↄ ㄱ ╩ ⁂ Δ ╢

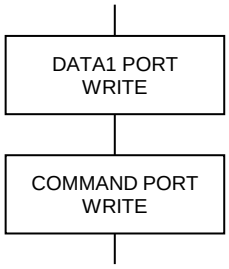
Ẽ
COMP1 ─ پ ≤ ⌐ ⁸ COMP1 ADD ♦ ↄ ㄱ ⌐ ↕ ╢ ≡ ™ ╢ ♦ ↄ ㄱ ╩ ⁸
COMPARE REGISTER1 ─ ♦ ↄ ㄱ ⌐ ⇔ ≡ ⁸ COMPARE REGISTER1 ╩ ⁂ ⇔ ╕ Δ ⁹

COMPARE REGISTER1 <= COMPARE REGISTER1 COMP1 ADD ♦ ↄ ㄱ

COMP1 ─ پ ⅜ ☻ ⁱ ↄ ─ ≤ ⅝ │ ⁸ پ ─ ╩ ⇔ ╕ Δ ⁹

o x
╩ ئو Δ ╢ │ ⁸ ↓ ↄ ⁱ ▪ ╩ AUTO CLEAR ENABLE = 0 ⌐ ⇔ ≡ ↄ ∞ ↕ ™ ⁹
* ○ ↄ ↓ ↄ ⁱ ▪ ╩ AUTO CLEAR ENABLE = 1 ⌐ Δ ╢ ≤ ⁸ │ ⌐ ⌂ ╡ ╕ Δ ⁹

(2) ADDRESS COUNTER INITIALIZE2



COMMAND : H'82 ADDRESS COUNTER INITIALIZE2 COMMAND

DRIVE DATA1 PORT

DRIVE COMMAND PORT COMMAND

DRIVE DATA1 PORT

D15	D14	D13	D12	D11	D10	D9	D8
COMP3 TYPE1	COMP3 TYPE0	COMP2 TYPE1	COMP2 TYPE0	COMP3 STOP TYPE1	COMP3 STOP TYPE0	COMP3 STOP ENABLE	COMP3 INT ENABLE

D7	D6	D5	D4	D3	D2	D1	D0
COMP2 STOP TYPE1	COMP2 STOP TYPE0	COMP2 STOP ENABLE	COMP2 INT ENABLE		COMP1 STOP TYPE	COMP1 STOP ENABLE	COMP1 INT ENABLE

H'0000

D0 COMP1 INT ENABLE

COMP1 ADRINT

0 COMP1 ADRINT

1 COMP1 ADRINT

D1 COMP1 STOP ENABLE

COMP1

0 COMP1

1 COMP1

D2 COMP1 STOP TYPE

COMP1

0

1

COMP1 COMPARE REGISTER1

D4 COMP2 INT ENABLE

COMP2 ADRINT

0 COMP2 ADRINT

1 COMP2 ADRINT

D5 COMP2 STOP ENABLE

COMP2

0 COMP2

1 COMP2

D6 COMP2 STOP TYPE0
D7 COMP2 STOP TYPE1

COMP2 ─پ ⌐ ╟┤ تج ╩ ⇔ ╕ Δ 9

TYPE1	TYPE0	COMP2 ─تج
0	0	پ ≠ R i ☻ ╩ تج Δ ┤
0	1	پ ≠ R i ☻ ╩ تج Δ ┤
1	0	پ ≠ 8 (CW) ─R i ☻ ╩ تج Δ ┤
1	1	پ ≠ 8 (CW) ─R i ☻ ╩ تج Δ ┤

D8 COMP3 INT ENABLE

COMP3 ─پ ╩ 8 ADRINT ⌐ 8 Δ ┤ ⇔ ⌂ ™ 9 ╩ ⇔ ╕ Δ 9

0 COMP3 ─پ ╩ADRINT ⌐ ⇔ ⌂ ™

1 COMP3 ─پ ╩ADRINT ⌐ Δ ┤

D9 COMP3 STOP ENABLE

COMP3 ─پ ⌐ ╟┤ تج ╩ 8 Δ ┤ ⇔ ⌂ ™ 9 ╩ ⇔ ╕ Δ 9

0 COMP3 ─پ ─تج ╩ ⇔ ⌂ ™

1 COMP3 ─پ ─تج ╩ Δ ┤

D10 COMP3 STOP TYPE0
D11 COMP3 STOP TYPE1

COMP3 ─پ ⌐ ╟┤ تج ╩ ⇔ ╕ Δ 9

TYPE1	TYPE0	COMP3 ─تج
0	0	پ ≠ R i ☻ ╩ تج Δ ┤
0	1	پ ≠ R i ☻ ╩ تج Δ ┤
1	0	پ ≠ 8 (CCW) ─R i ☻ ╩ تج Δ ┤
1	1	پ ≠ 8 (CCW) ─R i ☻ ╩ تج Δ ┤

D12 COMP2 TYPE0
D13 COMP2 TYPE1

COMP2 ─ س╩ ⇔ ╕ Δ 9

TYPE1	TYPE0	COMP2 ─ س
0	0	○ ► fi ㉯ ─ثم COMPARE REGISTER2 ─ثم
0	1	○ ► fi ㉯ ─ثم " COMPARE REGISTER2 ─ثم
1	0	○ ► fi ㉯ ─ثم ' COMPARE REGISTER2 ─ثم
1	1	

D14 COMP3 TYPE0
D15 COMP3 TYPE1

COMP3 ─ س╩ ⇔ ╕ Δ 9

TYPE1	TYPE0	COMP3 ─ س
0	0	○ ► fi ㉯ ─ثم COMPARE REGISTER3 ─ثم
0	1	○ ► fi ㉯ ─ثم " COMPARE REGISTER3 ─ثم
1	0	○ ► fi ㉯ ─ثم ' COMPARE REGISTER3 ─ثم
1	1	

(3) ADDRESS COUNTER PRESET

▪ ♪ ☻ ○ ▶ fi ㉺ — ㅇ ⇔ ㅋ Δ ⁹
↓ — ◖ ∀ fi ♪ — │ ≠ Δ ⁹

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'80 ADDRESS COUNTER PRESET COMMAND

DATA1 PORT WRITE

ㄱ DRIVE DATA1 PORT ⌐ ㅇ D15--D0 ⊥ ⅝ ㅠ ㅋ Δ ⁹

DATA2 PORT WRITE

ㅇ DRIVE DATA2 PORT ⌐ ㅇ D31--D16 ⊥ ⅝ ㅠ ㅋ Δ ⁹
│ ⁸ ▪ ♪ ☻ ○ ▶ fi ㉺ ≠ ⇔ ≡ ™ ╢ ▪ ♪ ☻ — ≠ Δ ⁹
-2,147,483,648 +2,147,483,647 H'8000_0000 H'7FFF_FFFF
— │ 2 — ⌐ ⇔ ㅋ Δ ⁹

COMMAND PORT WRITE

ㅇ DRIVE COMMAND PORT ⌐ COMMAND ⊥ ⅝ ㅠ ㅋ Δ ⁹

DRIVE DATA1 PORT — ㅠ ⇔ ♦ ↄ ㉺

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←							ㅇ						→	D0

DRIVE DATA2 PORT — ㅠ ⇔ ♦ ↄ ㉺

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←							ㅇ						→	D16

ㅠ ⅛ ⅛ ⅛ — ㅁ │ H'0000_0000 ≠ Δ ⁹

ㅎ ㅇ ⌐ │ ⁸ H'8000_0000 ⊥ Δ ╢ ↓ ≤ ⊥ ≠ ⅝ ㅋ Δ ⁹
√ ∞ ⇔ ⁸ H'8000_0000 ⊥ Δ ╢ ≤ ⁸ STATUS4 PORT — ADDRESS OVF = 1 ⌐ ⌂ ╡ ㅋ Δ ⁹

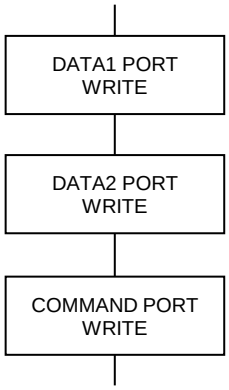
ㅎ ㅈ ㅌ — │ ⁸ ◀ ! ↄ ⌐ ⌂ ╡ ⁸ ERROR STATUS — INDEX CHANGE ERROR = 1 ⌐ ⇔ ㅋ Δ ⁹
ABS INDEX ♪ ! ▬ K ㅊ ⌐ ⁸ ADDRESS COUNTER PRESET ◖ ∀ fi ♪ — ≠ ㅇ ⅜ ↕ ╢ ⁸
⅛ ⅜ ⌂ ⌐ ⌂ ∫ √ ≤ ⅝ ⁹

(4) ADDRESS COUNTER MAX COUNT SET

アドレスカウンタの最大カウント数を設定します。
このコマンドの実行は常時可能です。

< 実行シーケンス >

COMMAND : H'87　ADDRESS COUNTER MAX COUNT SET COMMAND



① DRIVE DATA1 PORT に最大カウント数 D15--D0 を書き込みます。

② DRIVE DATA2 PORT に最大カウント数 D31--D16 を書き込みます。
設定範囲は、1 ～ 4,294,967,295（H'0000_0001 ～ H'FFFF_FFFF）です。

③ DRIVE COMMAND PORT に COMMAND を書き込みます。

DRIVE DATA1 PORT の読み出しデータ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←						最大カウント数							→	D0

DRIVE DATA2 PORT の読み出しデータ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←						最大カウント数							→	D16

● リセット後の初期値はH'FFFF_FFFF です。

・ カウント数が設定値の 1/2 に達すると、STATUS4 PORT の ADDRESS OVF = 1 になります。

・ 最大カウント数を設定しても、現在のアドレスカウンタの値は変わりません。
アドレスカウンタの値が、最大カウント数の範囲内になったときから、設定が有効になります。

・ アドレスカウンタの最大カウント数を H'FFFF_FFFF 以外に設定した場合、ABS INDEX および
ABS INDEX CHANGE コマンドを実行しないで下さい。

■ **最大カウント数**
設定値をカウンタの最大値として、リングカウントします。
STATUS4 PORT の ADDRESS OVF フラグを無視すれば、回転系のアドレス管理ができます。

・ 最大カウント数＝ 1,999 の場合（2,000 カウントで 1 回転）
＋方向のカウント：0 → 1 →…→ 999 → 1000 (ADDRESS OVF = 1)→ 1001 →…→ 1999 → 0
－方向のカウント：0 → 1999 →…→ 1001 → 1000 (ADDRESS OVF = 1)→ 999 →…→ 1 → 0

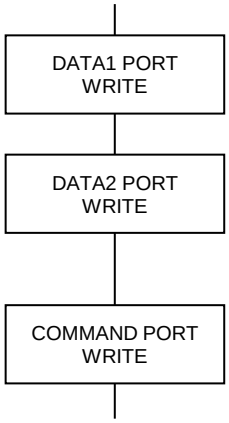
・ 最大カウント数＝ 2,000 の場合（2,001 カウントで 1 回転）
＋方向のカウント：0 → 1 →…→ 1000 → 1001 (1001 になると ADDRESS OVF = 1)→…→ 2000 → 0
－方向のカウント：0 → 2000 →…→ 1001 → 1000 (1000 になると ADDRESS OVF = 1)→…→ 1 → 0

(5) ADRINT COMPARE REGISTER1,2,3 SET

▪ ♪ ☻ ○ ► fi ㉡ — COMPARE REGISTER1, 2, 3 ⌐ ₀ ⇔ ┐ Δ ⁹
↓ — ◖ ∿ fi ♪ — │ ≠ Δ ⁹

COMMAND : H'88	ADRINT COMPARE REGISTER1 SET COMMAND
COMMAND : H'89	ADRINT COMPARE REGISTER2 SET COMMAND
COMMAND : H'8A	ADRINT COMPARE REGISTER3 SET COMMAND

< ◦ ↄ ◔ fi ☻ >



ⁿ DRIVE DATA1 PORT ⌐ ₀ D15--D0 ⇐ ⅝ ╗ ┐ Δ ⁹

ᵒ DRIVE DATA2 PORT ⌐ ₀ D31--D16 ⇐ ⅝ ╗ ┐ Δ ⁹
│ ⁸ ▪ ♪ ☻ ○ ► fi ㉡ ≠ ⇔ ≡ ™ ┤ ▪ ♪ ☻ — ≠ Δ ⁹
-2,147,483,648 +2,147,483,647 H'8000_0000 H'7FFF_FFFF
— │ 2 — ⌐ ⇔ ┐ Δ ⁹

ᵓ DRIVE COMMAND PORT ⌐ COMMAND ⇐ ⅝ ╗ ┐ Δ ⁹

DRIVE DATA1 PORT — ╗ ⇔ ♦ ↄ ㉡

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←							₀						→	D0

DRIVE DATA2 PORT — ╗ ⇔ ♦ ↄ ㉡

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←							₀						→	D16

ùr ˡ ⁸ ᵉ ˡ — ثم │ H'8000_0000 ≠ Δ ⁹

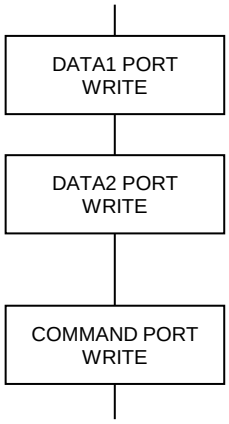
(6) ADRINT COMP1 ADD DATA SET

▪ ♪ ☻ ○ ► fi ㉺ — COMP1 — ♦ ⊃ ㉺ ⇔ ╕ Δ ⁹
↓ — ◖ ∿ fi ♪ — │ ≠ Δ ⁹

< ◦ ⊃ ◔ fi ☻ >

COMMAND : H'8C

ADRINT COMP1 ADD DATA SET COMMAND



ŋ DRIVE DATA1 PORT ⌐ COMP1 ADD ♦ ⊃ ㉺ D15--D0 ⩸ ⅝ ╗ ╕ Δ ⁹

º DRIVE DATA2 PORT ⌐ COMP1 ADD ♦ ⊃ ㉺ D31--D16 ⩸ ⅝ ╗ ╕ Δ ⁹
│ ⁸ ▪ ♪ ☻ ○ ► fi ㉺ ≠ ⇔ ≡ ™ ╢ ▪ ♪ ☻ — ≠ Δ ⁹
-2,147,483,648 +2,147,483,647 H'8000_0000 H'7FFF_FFFF
— │ 2 — ⌐ ⇔ ╕ Δ ⁹

ɔ DRIVE COMMAND PORT ⌐ COMMAND ⩸ ⅝ ╗ ╕ Δ ⁹

DRIVE DATA1 PORT — ╗ ⇔ ♦ ⊃ ㉺

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←					COMP1 ADD ♦ ⊃ ㉺								→	D0

DRIVE DATA2 PORT — ╗ ⇔ ♦ ⊃ ㉺

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←					COMP1 ADD ♦ ⊃ ㉺								→	D16

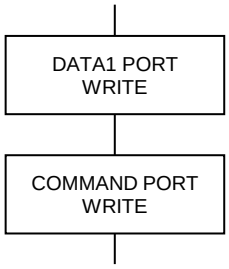
ù ¹ ⁸ ᵉ ⁾ — ⁿᵒ │ H'0000_0000 ≠ Δ ⁹

2-2-2. R i ● ○ ▶ fi ㅋ ㅡ
(1) PULSE COUNTER INITIALIZE1

R i ● ○ ▶ fi ㅋ ㅡ ╩ ⇔ ╕ Δ ⁹
↓ ㅡ ◖ ∀ fi ♪ ㅡ │ ≠ Δ ⁹

< ◦ ↄ ◗ fi ● >

COMMAND : H'91 PULSE COUNTER INITIALIZE1 COMMAND



ŋ DRIVE DATA1 PORT ⌐ ♦ ↄ ㅋ ╩ ⅝ ╖ ╕ Δ ⁹

º DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

DRIVE DATA1 PORT ㅡ ♦ ↄ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8
AUTO ADD ENABLE	AUTO CLR ENABLE	COMP GATE TYPE1	COMP GATE TYPE0	CNTINT PULSE TYPE1	CNTINT PULSE TYPE0	CNTINT TYPE1	CNTINT TYPE0

D7	D6	D5	D4	D3	D2	D1	D0
EXT COUNT DIRECTION				EXT COUNT TYPE1	EXT COUNT TYPE0	COUNT PULSE SEL1	COUNT PULSE SEL0

ù ' ⁹ ♀ ♩ ㅡ ㅊ │ H'0000 (▪ fi ♭ ↄ ' ▬ fi ㅅ) ≠ Δ ⁹

D0 COUNT PULSE SEL0
D1 COUNT PULSE SEL1

○ ▶ fi ㅋ ㅡ ○ ▶ fi ♩ R i ● ╩ ⇔ ╕ Δ ⁹

Xn, Zn, Bn ⌐ Δ ╫

SEL1	SEL0	○ ▶ fi ♩ R i ●	○ ▶ fi ♩
0	0	(Xn,Zn,Bn)ㅡ R i ● ╩ ○ ▶ fi ♩ Δ ╫	≠ ○ ▶ fi ♩ ▪ ℙ ϰ
0	1	ث (Yn,An,Cn)ㅡ R i ● ╩ ○ ▶ fi ♩ Δ ╫	≠ ○ ▶ fi ♩ ♭ ▶ fi
1	0	(Xn,Zn,Bn)ㅡ R i ● كخ ╩ ○ ▶ fi ♩ Δ ╫	EXT COUNT DIRECTION
1	1	ث (Yn,An,Cn)ㅡ R i ● كخ ╩ ○ ▶ fi ♩ Δ ╫	≠

Yn, An, Cn ⌐ Δ ╫

SEL1	SEL0	○ ▶ fi ♩ R i ●	○ ▶ fi ♩
0	0	(Yn,An,Cn)ㅡ R i ● ≠ ○ ▶ fi ♩ Δ ╫	≠ ○ ▶ fi ♩ ▪ ℙ ϰ
0	1	ث (Xn,Zn,Bn)ㅡ R i ● ╩ ○ ▶ fi ♩ Δ ╫	≠ ○ ▶ fi ♩ ♭ ▶ fi
1	0	ث (Xn,Zn,Bn)ㅡ R i ● كخ ╩ ○ ▶ fi ♩ Δ ╫	EXT COUNT DIRECTION
1	1	(Yn,An,Cn)ㅡ R i ● كخ ≠ ○ ▶ fi ♩ Δ ╫	≠

D2 EXT COUNT TYPE0
D3 EXT COUNT TYPE1

R i ● كخ طص ㅡ ○ ▶ fi ♩ ╩ ⇔ ╕ Δ ⁹

TYPE1	TYPE0	○ ▶ fi ♩	R i ● طص
0	0	EA, EB ╩ ≠ى ○ ▶ fi ♩ Δ ╫	طص كخ ٥
0	1	EA, EB ╩ ≠ى ○ ▶ fi ♩ Δ ╫	
1	0	EA, EB ╩ ≠ى ○ ▶ fi ♩ Δ ╫	
1	1	EA≠ ㅡ ○ ▶ fi ♩ ⁸ EB ≠ ㅡ ○ ▶ fi ♩	R i ● طص

D7 EXT COUNT DIRECTION

0	EA, EB
1	

D8 CNTINT TYPE0

D9 CNTINT TYPE1

DRIVE STATUS4 PORT ≤ CNTINT COMP1, 2, 3

TYPE1	TYPE0	COMP1, 2, 3	
0	0		DRIVE STATUS4 PORT
0	1		DRIVE STATUS4 PORT
1	0		
1	1		INT FACTOR CLR CNTINT INT CLR = 1

CNTINT PULSE TYPE

CNTINT TYPE1,0="1,1"

COMP1,2,3

INT FACTOR CLR

CNTINT INT CLR=1

D10 CNTINT PULSE TYPE0

D11 CNTINT PULSE TYPE1

COMP1, 2, 3

TYPE1	TYPE0	
0	0	200 ns
0	1	10 εs
1	0	100 εs
1	1	1,000 εs

D12 COMP GATE TYPE0

D13 COMP GATE TYPE1

CNTINT COMP1, 2, 3

TYPE1	TYPE0	
0	0	COMP1 OR (COMP2 OR COMP3)
0	1	COMP1 OR (COMP2 AND COMP3)
1	0	COMP1 AND (COMP2 OR COMP3)
1	1	COMP1 AND (COMP2 AND COMP3)

OR AND

D14 AUTO CLEAR ENABLE

COMP1 ─ ○ ↄ ♩ ▼ ⁱ ▪ ≠ ⁸ ○ ▶ fi ㉺ ╩ ₈ ▼ ⁱ ▪ Δ ╫ ▼ ⁱ ▪ ⇔ ⌂ ™ ₉ ╩ ⇔ ╕ Δ ⁹

0 COMP1 ─ پ ≠ ○ ▶ fi ㉺ ╩ ▼ ⁱ ▪ ⇔ ⌂ ™

1 COMP1 ─ پ ≠ ○ ▶ fi ㉺ ╩ ▼ ⁱ ▪ Δ ╫

Ẽ ○ ↄ ♩ ▼ ⁱ ▪

COMP1 ─ پ ≤ ⌐ ⁸ ▪ ♪ ☻ ○ ▶ fi ㉺ ─ ♦ ↄ ㉺ ╩"0" ⌐ ▼ ⁱ ▪ ⇔ ╕ Δ ⁹

COMP1 ─ پ ⅜ ☻ ⁱ ↄ ─ ≤ ⅝ │ ⁸ پ ─ ╩ ⇔ ╕ Δ ⁹

D15 AUTO ADD ENABLE

COMP1 ─ ≠ ⁸ ♦ ↄ ㉺ ╩ ₈ ㉺ Δ ╫ ㉺ ⇔ ⌂ ™ ₉ ╩ ⇔ ╕ Δ ⁹

0 COMP1 ─ پ ≠ ♦ ↄ ㉺ ╩ ㉺ ⇔ ⌂ ™

1 COMP1 ─ پ ≠ ♦ ↄ ㉺ ╩ ㉺ Δ ╫

Ẽ

COMP1 ─ پ ≤ ⌐ ⁸ COMP1 ADD ♦ ↄ ㉺ ⌐ ↕ ╬ ≡ ™ ╫ ♦ ↄ ㉺ ╩ ⁸

COMPARE REGISTER1 ─ ♦ ↄ ㉺ ⌐ ⇔ ≡ ⁸ COMPARE REGISTER1 ╩ ㉺ ⇔ ╕ Δ ⁹

COMPARE REGISTER1 <= COMPARE REGISTER1 COMP1 ADD ♦ ↄ ㉺

COMP1 ─ پ ⅜ ☻ ⁱ ↄ ─ ≤ ⅝ │ ⁸ پ ─ ╩ ⇔ ╕ Δ ⁹

o x

╩ ئو Δ ╫ │ ⁸ ○ ↄ ♩ ▼ ⁱ ▪ ╩ AUTO CLEAR ENABLE = 0 ⌐ ⇔ ≡ ↄ ∞ ↕ ™ ⁹

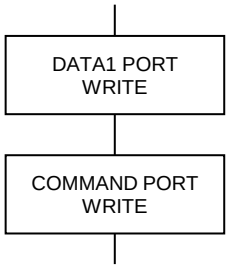
* ○ ↄ ♩ ▼ ⁱ ▪ ╩ AUTO CLEAR ENABLE = 1 ⌐ Δ ╫ ≤ ⁸ │ ⌐ ⌂ ╡ ╕ Δ ⁹

(2) PULSE COUNTER INITIALIZE2

R ⁱ ☻ ○ ► fi ㅋ — ⇔ ㄱ Δ ⁹
↓ — ◖ ∿ fi ♪ — ≠ Δ ⁹

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'92 PULSE COUNTER INITIALIZE2 COMMAND



ŋ DRIVE DATA1 PORT ⌐ ♦ ↄ ㅋ ⅝ ㄱ Δ ⁹

º DRIVE COMMAND PORT ⌐ COMMAND ⅝ ㄱ Δ ⁹

DRIVE DATA1 PORT — ♦ ↄ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8
COMP3 TYPE1	COMP3 TYPE0	COMP2 TYPE1	COMP2 TYPE0	COMP3 STOP TYPE1	COMP3 STOP TYPE0	COMP3 STOP ENABLE	COMP3 INT ENABLE

D7	D6	D5	D4	D3	D2	D1	D0
COMP2 STOP TYPE1	COMP2 STOP TYPE0	COMP2 STOP ENABLE	COMP2 INT ENABLE		COMP1 STOP TYPE	COMP1 STOP ENABLE	COMP1 INT ENABLE

ù ⁱ ☿ ℘ ♩ — شم | H'0000 (▪ fi ♭ ↄ ¹ ▬ fi سخ) ≠ Δ ⁹

D0 COMP1 INT ENABLE

COMP1 — پ ⅝ ⁸ CNTINT ⌐ ₈ Δ ╢ ⇔ ⌂ ™ ₉ ⅝ ⇔ ㄱ Δ ⁹
0 COMP1 — پ ⅝ CNTINT ⌐ ⇔ ⌂ ™
1 COMP1 — پ ⅝ CNTINT ⌐ Δ ╢

D1 COMP1 STOP ENABLE

COMP1 — پ ⌐ ╟ ╢ تج ⅝ ₈ Δ ╢ ⇔ ⌂ ™ ₉ ⅝ ⇔ ㄱ Δ ⁹
0 COMP1 — پ — تج ⅝ ⇔ ⌂ ™
1 COMP1 — پ — تج ⅝ Δ ╢

D2 COMP1 STOP TYPE

COMP1 — پ ⌐ ╟ ╢ تج ⅝ ⇔ ㄱ Δ ⁹
0 پ ≠ R ⁱ ☻ ⅝ تج Δ ╢
1 پ ≠ R ⁱ ☻ ⅝ تج Δ ╢

ה COMP1 — سد | ⁸ ₈ ○ ► fi ㅋ — شم COMPARE REGISTER1 — شم ₉ ≠ Δ ⁹

D4 COMP2 INT ENABLE

COMP2 — پ ⅝ ⁸ CNTINT ⌐ ₈ Δ ╢ ⇔ ⌂ ™ ₉ ⅝ ⇔ ㄱ Δ ⁹
0 COMP2 — پ ⅝ CNTINT ⌐ ⇔ ⌂ ™
1 COMP2 — پ ⅝ CNTINT ⌐ Δ ╢

D5 COMP2 STOP ENABLE

COMP2 — پ ⌐ ╟ ╢ تج ⅝ ₈ Δ ╢ ⇔ ⌂ ™ ₉ ⅝ ⇔ ㄱ Δ ⁹
0 COMP2 — پ — تج ⅝ ⇔ ⌂ ™
1 COMP2 — پ — تج ⅝ Δ ╢

D6 COMP2 STOP TYPE0
D7 COMP2 STOP TYPE1

COMP2 ─پ ⌐ ╟┤ تج ╩ ⇔ ╕ Δ 9

TYPE1	TYPE0	COMP2 ─تج
0	0	پ ≠ R i ☻ ╩ تج Δ ┤
0	1	پ ≠ R i ☻ ╩ تج Δ ┤
1	0	پ ≠ 8 (CW) ─R i ☻ ╩ تج Δ ┤
1	1	پ ≠ 8 (CW) ─R i ☻ ╩ تج Δ ┤

D8 COMP3 INT ENABLE

COMP3 ─پ ╩ 8 CNTINT ⌐ 8 Δ ┤ ⇔ ⌂ ™ 9 ╩ ⇔ ╕ Δ 9

0 COMP3 ─پ ╩CNTINT ⌐ ⇔ ⌂ ™

1 COMP3 ─پ ╩CNTINT ⌐ Δ ┤

D9 COMP3 STOP ENABLE

COMP3 ─پ ⌐ ╟┤ تج ╩ 8 Δ ┤ ⇔ ⌂ ™ 9 ╩ ⇔ ╕ Δ 9

0 COMP3 ─پ ─تج ╩ ⇔ ⌂ ™

1 COMP3 ─پ ─تج ╩ Δ ┤

D10 COMP3 STOP TYPE0
D11 COMP3 STOP TYPE1

COMP3 ─پ ⌐ ╟┤ تج ╩ ⇔ ╕ Δ 9

TYPE1	TYPE0	COMP3 ─تج
0	0	پ ≠ R i ☻ ╩ تج Δ ┤
0	1	پ ≠ R i ☻ ╩ تج Δ ┤
1	0	پ ≠ 8 (CCW) ─R i ☻ ╩ تج Δ ┤
1	1	پ ≠ 8 (CCW) ─R i ☻ ╩ تج Δ ┤

D12 COMP2 TYPE0
D13 COMP2 TYPE1

COMP2 ─ س╩ ⇔ ╕ Δ 9

TYPE1	TYPE0	COMP2 ─ س
0	0	○ ► fi ㉦ ─ ثم COMPARE REGISTER2 ─ ثم
0	1	○ ► fi ㉦ ─ ثم " COMPARE REGISTER2 ─ ثم
1	0	○ ► fi ㉦ ─ ثم ' COMPARE REGISTER2 ─ ثم
1	1	

D14 COMP3 TYPE0
D15 COMP3 TYPE1

COMP3 ─ س╩ ⇔ ╕ Δ 9

TYPE1	TYPE0	COMP3 ─ س
0	0	○ ► fi ㉦ ─ ثم COMPARE REGISTER3 ─ ثم
0	1	○ ► fi ㉦ ─ ثم " COMPARE REGISTER3 ─ ثم
1	0	○ ► fi ㉦ ─ ثم ' COMPARE REGISTER3 ─ ثم
1	1	

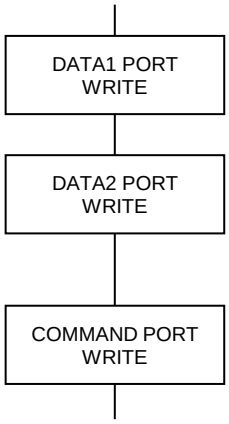
(3) PULSE COUNTER PRESET

パルスカウンタのカウント初期値を設定します。

↓ ー ◖ ∀ fi ♪ ー │ ≠ Δ ⁹

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'90 PULSE COUNTER PRESET COMMAND



ŋ DRIVE DATA1 PORT ⌐ ○ ► fi ♩ ثم D15--D0 ⩸ ⅝ ╗ ╕ Δ ⁹

º DRIVE DATA2 PORT ⌐ ○ ► fi ♩ ثم D31--D16 ⩸ ⅝ ╗ ╕ Δ ⁹

│ ⁸

-2,147,483,648 +2,147,483,647 H'8000_0000 H'7FFF_FFFF

ー │ 2 ー ⌐ ⇔ ╕ Δ ⁹

ↄ DRIVE COMMAND PORT ⌐ COMMAND ⩸ ⅝ ╗ ╕ Δ ⁹

DRIVE DATA1 PORT ー ╗ ⇔ ♦ ↄ ⅟

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←					○ ► fi ♩			ثم					→	D0

DRIVE DATA2 PORT ー ╗ ⇔ ♦ ↄ ⅟

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←					○ ► fi ♩			ثم					→	D16

ùr ˡ ☿ ᵉ ♩ ー ثم │ H'0000_0000 ≠ Δ ⁹

ה ◦ ⌐ │ ⁸ H'8000_0000 ⩸ Δ ╢ ↓ ≤ ⩸ ≠ ⅝ ╕ Δ ⁹

√ ∞ ⇔ ⁸ H'8000_0000 ⩸ Δ ╢ ≤ ⁸ STATUS4 PORT ー PULSE OVF = 1 ⌐ ⌂ ╡ ╕ Δ ⁹

(4) PULSE COUNTER MAX COUNT SET

[illegible]

< [illegible] >

COMMAND : H'97 PULSE COUNTER MAX COUNT SET COMMAND

DATA1 PORT WRITE

[illegible] DRIVE DATA1 PORT [illegible] D15--D0 [illegible]

DATA2 PORT WRITE

[illegible] DRIVE DATA2 PORT [illegible] D31--D16 [illegible]
[illegible] 1 4,294,967,295 H'0000_0001 H'FFFF_FFFF [illegible]

COMMAND PORT WRITE

[illegible] DRIVE COMMAND PORT [illegible] COMMAND [illegible]

DRIVE DATA1 PORT [illegible]

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←						[illegible]							→	D0

DRIVE DATA2 PORT [illegible]

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←						[illegible]							→	D16

[illegible] H'FFFF_FFFF [illegible]

[illegible] 1/2 [illegible] STATUS4 PORT [illegible] PULSE OVF = 1 [illegible]

[illegible]

[illegible]

[illegible]

STATUS4 PORT [illegible] PULSE OVF [illegible]

[illegible] 1,999 [illegible] 2,000 [illegible]
[illegible] 0 [illegible] 1 [illegible] 999 [illegible] 1000 (PULSE OVF = 1) [illegible] 1001 [illegible] 1999 [illegible] 0
[illegible] 0 [illegible] 1999 [illegible] 1001 [illegible] 1000 (PULSE OVF = 1) [illegible] 999 [illegible] 1 [illegible] 0

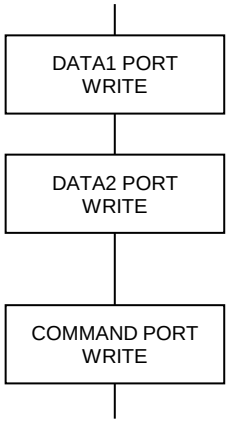
[illegible] 2,000 [illegible] 2,001 [illegible]
[illegible] 0 [illegible] 1 [illegible] 1000 [illegible] 1001 (1001 [illegible] PULSE OVF = 1) [illegible] 2000 [illegible] 0
[illegible] 0 [illegible] 2000 [illegible] 1001 [illegible] 1000 (1000 [illegible] PULSE OVF = 1) [illegible] 1 [illegible] 0

(5) CNTINT COMPARE REGISTER1,2,3 SET

R ⁱ ☻ ○ ► fi ㉻ — COMPARE REGISTER1, 2, 3 ⌐ ₀ ⩸ ⇔ ╕ Δ ⁹
↓ — ◖ ∿ fi ♪ — │ ≠ Δ ⁹

COMMAND : H'98	CNTINT COMPARE REGISTER1 SET COMMAND
COMMAND : H'99	CNTINT COMPARE REGISTER2 SET COMMAND
COMMAND : H'9A	CNTINT COMPARE REGISTER3 SET COMMAND

< ◦ ↄ ◔ fi ☻ >



ⁿ DRIVE DATA1 PORT ⌐ ₀ D15--D0 ⩸ ⅝ ╗ ╕ Δ ⁹

ᵒ DRIVE DATA2 PORT ⌐ ₀ D31--D16 ⩸ ⅝ ╗ ╕ Δ ⁹
│ ⁸
-2,147,483,648 +2,147,483,647 H'8000_0000 H'7FFF_FFFF
— │ 2 — ⌐ ⇔ ╕ Δ ⁹

ᶜ DRIVE COMMAND PORT ⌐ COMMAND ⩸ ⅝ ╗ ╕ Δ ⁹

DRIVE DATA1 PORT — ╗ ⇔ ♦ ↄ ㉻

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←							₀						→	D0

DRIVE DATA2 PORT — ╗ ⇔ ♦ ↄ ㉻

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←							₀						→	D16

ù ˡ ⁸ ᵉ ˡ — ثم │ H'8000_0000 ≠ Δ ⁹

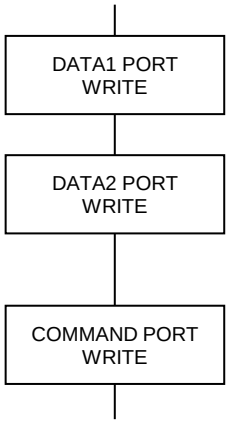
(6) CNTINT COMP1 ADD DATA SET

R ⁱ ☻ ○ ► fi ㉯ — COMP1 — ♦ ⊃ ㉯ ⩸ ⇔ ╕ Δ ⁹
↓ — ◖ ∿ fi ♪ — │ ≠ Δ ⁹

< ◦ ⊃ ◔ fi ☻ >

COMMAND : H'9C

CNTINT COMP1 ADD DATA SET COMMAND



ⁿ DRIVE DATA1 PORT ⌐ COMP1 ADD ♦ ⊃ ㉯ D15--D0 ⩸ ⅝ ╗ ╕ Δ ⁹

º DRIVE DATA2 PORT ⌐ COMP1 ADD ♦ ⊃ ㉯ D31--D16 ⩸ ⅝ ╗ ╕ Δ ⁹
│ ⁸

-2,147,483,648 +2,147,483,647 H'8000_0000 H'7FFF_FFFF
— │ 2 — ⌐ ⇔ ╕ Δ ⁹

ᵒ DRIVE COMMAND PORT ⌐ COMMAND ⩸ ⅝ ╗ ╕ Δ ⁹

DRIVE DATA1 PORT — ╗ ⇔ ♦ ⊃ ㉯

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←					COMP1 ADD ♦ ⊃ ㉯								→	D0

DRIVE DATA2 PORT — ╗ ⇔ ♦ ⊃ ㉯

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←					COMP1 ADD ♦ ⊃ ㉯								→	D16

ù ⁱ ⁸ ⁸ ⁾ — ⁿᵐ │ H'0000_0000 ≠ Δ ⁹

2-2-3. R i ● ئجر ○ ▶ fi ٥ ا —

(1) DFL COUNTER INITIALIZE1

R i ● ئجر ○ ▶ fi ٥ ا — ⇔ ┐ Δ ⁹

↓ — ◖ ٧ fi ♪ — | ≠ Δ ⁹

< ◦ ↄ ◔ fi ● >

COMMAND : H'A1

DFL COUNTER INITIALIZE1 COMMAND

DATA1 PORT WRITE

COMMAND PORT WRITE

ŋ DRIVE DATA1 PORT ⌐ ♦ ↄ ٥ ا ⅝ ╥ ┐ Δ ⁹

º DRIVE COMMAND PORT ⌐ COMMAND ⅝ ╥ ┐ Δ ⁹

DRIVE DATA1 PORT — ♦ ↄ ٥ ا

D15	D14	D13	D12	D11	D10	D9	D8
AUTO ADD ENABLE	AUTO CLR ENABLE	COMP GATE TYPE1	COMP GATE TYPE0	DFLINT PULSE TYPE1	DFLINT PULSE TYPE0	DFLINT TYPE1	DFLINT TYPE0

D7	D6	D5	D4	D3	D2	D1	D0
DIVISION TYPE	TIMER START TYPE2	TIMER START TYPE1	TIMER START TYPE0	EXT COUNT TYPE1	EXT COUNT TYPE0	COUNT PULSE SEL1	COUNT PULSE SEL0

ئر ا ⅜ ع ل — ثم | H'0000 (▪ fi ط ↄ ا ▬ fi سخ) ≠ Δ ⁹

D0 COUNT PULSE SEL0

D1 COUNT PULSE SEL1

○ ▶ fi ٥ ا — ○ ▶ fi ل R i ● ⅝ ⇔ ┐ Δ ⁹

Xn, Zn, Bn ⌐ Δ ╢

SEL1	SEL0	○ ▶ fi ل R i ●	○ ▶ fi ل R i ●
0	0	(Xn,Zn,Bn)— R i ● كج	(Xn,Zn,Bn)— R i ●
0	1	ث (Yn,An,Cn)— R i ● كج	(Xn,Zn,Bn)— R i ●
1	0	ث (Yn,An,Cn)— R i ● كج	(Xn,Zn,Bn)— R i ● كج
1	1	20MHz ◒ ع ◒	(⌂ ⇔)

Yn, An, Cn ⌐ Δ ╢

SEL1	SEL0	○ ▶ fi ل R i ●	○ ▶ fi ل R i ●
0	0	(Yn,An,Cn)— R i ● كج	(Yn,An,Cn)— R i ●
0	1	ث (Xn,Zn,Bn)— R i ● كج	(Yn,An,Cn)— R i ●
1	0	ث (Xn,Zn,Bn)— R i ● كج	(Yn,An,Cn)— R i ● كج
1	1	20MHz ◒ ع ◒	(⌂ ⇔)

Ẽ ○ ▶ fi ل

ה ○ ▶ fi ل R i ● طص ≠ ○ ▶ fi ل ▪ ع ك ⁸ طص ≠ ○ ▶ fi ل ط ▶ fi

ה ○ ▶ fi ل R i ● طص ≠ ○ ▶ fi ل ▪ ع ك ⁸ طص ≠ ○ ▶ fi ل ط ▶ fi

Ẽ ٥ ا ▬ ٧

11 ⌐ Δ ╢ ≤ ⁸ ○ ▶ fi ل R i ● ⅝ ⁸ ⌐ ○ ▶ fi ل ▪ ع ك ⇔ ┐ Δ ⁹

○ ▶ fi ل R i ● —20 MHz ◒ ع ◒ | ⁸ 1/1 1/256 ⌐ ⇔ ≡ ○ ▶ fi ل ≠ ⅝ ┐ Δ ⁹

D2 EXT COUNT TYPE0
D3 EXT COUNT TYPE1

TYPE1	TYPE0		
0	0	EA, EB	
0	1	EA, EB	
1	0	EA, EB	
1	1	EA≠ EB ≠	

D4 TIMER START TYPE0
D5 TIMER START TYPE1
D6 TIMER START TYPE2

COUNT PULSE SEL "11"

TYPE2	TYPE1	TYPE0	
0	0	0	
0	0	1	
0	1	0	DRIVE STATUS5 PORT SS0 = 1 ≠
0	1	1	DFL COUNTER INITIALIZE1 ≠
1	0	0	
1	0	1	
1	1	0	
1	1	1	

D7 DIVISION TYPE

0
1

D8 DFLINT TYPE0
D9 DFLINT TYPE1

DRIVE STATUS4 PORT ≤ CNTINT COMP1, 2, 3

TYPE1	TYPE0	COMP1, 2, 3	
0	0		DRIVE STATUS4 PORT
0	1		DRIVE STATUS4 PORT
1	0		
1	1		INT FACTOR CLR DFLINT INT CLR = 1 ≠

DFLINT PULSE TYPE ≠

DFLINT ≠ DFLINT TYPE1,0="1,1"
COMP1,2,3 INT FACTOR CLR DFLINT INT CLR=1 ≠

D10　DFLINT PULSE TYPE0
D11　DFLINT PULSE TYPE1

COMP1, 2, 3 ー ⌐ ⇔ √ ≤ ⅝ ー ⁸ ⇔ ┐ Δ ⁹

TYPE1	TYPE0	⌐ ー
0	0	200 ns
0	1	10 εs
1	0	100 εs
1	1	1,000 εs

⇔ √ ⁸ ↓ ー ⇔ ┐ Δ ⁹
↓ ー ⁹

D12　COMP GATE TYPE0
D13　COMP GATE TYPE1

DFLINT ⌐ Δ ╢ COMP1, 2, 3 ー ⌐ ー ⁸ ⇔ ┐ Δ ⁹

TYPE1	TYPE0	⌐ ー
0	0	COMP1 OR (COMP2 OR COMP3)
0	1	COMP1 OR (COMP2 AND COMP3)
1	0	COMP1 AND (COMP2 OR COMP3)
1	1	COMP1 AND (COMP2 AND COMP3)

OR ⁸ AND

D14　AUTO CLEAR ENABLE

COMP1 ー ⁸ ⁸ ⇔ ⌂ ™ ⁹ ⇔ ┐ Δ ⁹
0　COMP1 ー ⌐ ⇔ ⌂ ™
1　COMP1 ー ⌐ Δ ╢

Ẽ
COMP1 ー ⌐ ≤ ⁸ "0" ⇔ ┐ Δ ⁹
COMP1 ー ⌐ ⅜ ≤ ⅝ ⁸ ⌐ ー ⇔ ┐ Δ ⁹

D15　AUTO ADD ENABLE

COMP1 ー ⁸ Δ ╢ ⇔ ⌂ ™ ⁹ ⇔ ┐ Δ ⁹
0　COMP1 ー ⌐ ⇔ ⌂ ™
1　COMP1 ー ⌐ Δ ╢

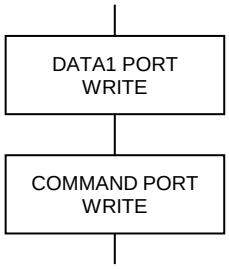
Ẽ
COMP1 ー ⌐ ≤ ⁸ COMP1 ADD ⁸
COMPARE REGISTER1 ー ⌐ ⇔ ≡ ⁸ COMPARE REGISTER1 ⇔ ┐ Δ ⁹

COMPARE REGISTER1 <= COMPARE REGISTER1　COMP1 ADD

COMP1 ー ⌐ ⅜ ≤ ⅝ ⁸ ⌐ ー ⇔ ┐ Δ ⁹

o　x
自動加算機能を使用する場合は、オートクリア機能を無効（AUTO CLEAR ENABLE = 0）にしてください。
＊ オートクリア機能を有効（AUTO CLEAR ENABLE = 1）にすると、自動加算機能は無効になります。

(2) DFL COUNTER INITIALIZE2



COMMAND : H'A2 — DFL COUNTER INITIALIZE2 COMMAND

DRIVE DATA1 PORT

DRIVE COMMAND PORT — COMMAND

DRIVE DATA1 PORT

D15	D14	D13	D12	D11	D10	D9	D8
COMP3 TYPE1	COMP3 TYPE0	COMP2 TYPE1	COMP2 TYPE0	COMP3 DETECT TYPE1	COMP3 STOP TYPE	COMP3 STOP ENABLE	COMP3 INT ENABLE

D7	D6	D5	D4	D3	D2	D1	D0
COMP2 DETECT TYPE1	COMP2 STOP TYPE	COMP2 STOP ENABLE	COMP2 INT ENABLE	COMP1 DETECT TYPE	COMP1 STOP TYPE	COMP1 STOP ENABLE	COMP1 INT ENABLE

H'9000

D0 COMP1 INT ENABLE

COMP1

0 COMP1 — DFLINT

1 COMP1 — DFLINT

D1 COMP1 STOP ENABLE

COMP1

0 COMP1

1 COMP1

D2 COMP1 STOP TYPE

COMP1

0

1

COMP1 — COMPARE REGISTER1

D3 COMP1 DETECT TYPE

COMP1

0

1

D4 COMP2 INT ENABLE

COMP2 — DFLINT

0 COMP2 — DFLINT

1 COMP2 — DFLINT

D5 COMP2 STOP ENABLE

COMP2 ─ پ ⌐ ╟─╢ تج ╩ 8 Δ ╢ ⇔ ⌂ ™ 9 ╩ ⇔ ╕ Δ 9

0 COMP2 ─ پ ─ تج ╩ ⇔ ⌂ ™

1 COMP2 ─ پ ─ تج ╩ Δ ╢

D6 COMP2 STOP TYPE

COMP2 ─ پ ⌐ ╟─╢ تج ╩ ⇔ ╕ Δ 9

0 پ ≠ R i ☻ ╩ تج Δ ╢

1 پ ≠ R i ☻ ╩ تج Δ ╢

D7 COMP2 DETECT TYPE

COMP2 ⅜ Δ ╢ ○ ► fi ㉺ ثم ─ 8 ╩ ⇔ ╕ Δ 9

0 ○ ► fi ㉺ ثم ╩ ثم ⌐ ⇔ ≡ Δ ╢

1 ○ ► fi ㉺ ثم ╩ ثم ⅝ ─ ╕ ╕ Δ ╢

D8 COMP3 INT ENABLE

COMP3 ─ پ ╩ 8 DFLINT ⌐ 8 Δ ╢ ⇔ ⌂ ™ 9 ╩ ⇔ ╕ Δ 9

0 COMP3 ─ پ ╩DFLINT ⌐ ⇔ ⌂ ™

1 COMP3 ─ پ ╩DFLINT ⌐ Δ ╢

D9 COMP3 STOP ENABLE

COMP3 ─ پ ⌐ ╟─╢ تج ╩ 8 Δ ╢ ⇔ ⌂ ™ 9 ╩ ⇔ ╕ Δ 9

0 COMP3 ─ پ ─ تج ╩ ⇔ ⌂ ™

1 COMP3 ─ پ ─ تج ╩ Δ ╢

D10 COMP3 STOP TYPE

COMP3 ─ پ ⌐ ╟─╢ تج ╩ ⇔ ╕ Δ 9

0 پ ≠ R i ☻ ╩ تج Δ ╢

1 پ ≠ R i ☻ ╩ تج Δ ╢

D11 COMP3 DETECT TYPE

COMP3 ⅜ Δ ╢ ○ ► fi ㉺ ثم ─ 8 ╩ ⇔ ╕ Δ 9

0 ○ ► fi ㉺ ثم ╩ ثم ⌐ ⇔ ≡ Δ ╢

1 ○ ► fi ㉺ ثم ╩ ثم ⅝ ─ ╕ ╕ Δ ╢

D12 COMP2 TYPE0

D13 COMP2 TYPE1

COMP2 ─ س ╩ ⇔ ╕ Δ 9

TYPE1	TYPE0	COMP2 ─ س
0	0	○ ► fi ㉺ ─ ثم COMPARE REGISTER2 ─ ثم
0	1	○ ► fi ㉺ ─ ثم " COMPARE REGISTER2 ─ ثم
1	0	○ ► fi ㉺ ─ ثم ' COMPARE REGISTER2 ─ ثم
1	1	

D14 COMP3 TYPE0

D15 COMP3 TYPE1

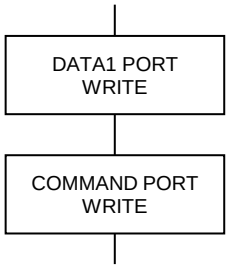
COMP3 ─ س ╩ ⇔ ╕ Δ 9

TYPE1	TYPE0	COMP3 ─ س
0	0	○ ► fi ㉺ ─ ثم COMPARE REGISTER3 ─ ثم
0	1	○ ► fi ㉺ ─ ثم " COMPARE REGISTER3 ─ ثم
1	0	○ ► fi ㉺ ─ ثم ' COMPARE REGISTER3 ─ ثم
1	1	

(3) DFL COUNTER INITIALIZE3

[illegible]

[illegible] COMMAND : H'A3 DFL COUNTER INITIALIZE3 COMMAND



[illegible] DRIVE DATA1 PORT [illegible]

[illegible] DRIVE COMMAND PORT [illegible] COMMAND [illegible]

DRIVE DATA1 PORT [illegible]

D15	D14	D13	D12	D11	D10	D9	D8

D7	D6	D5	D4	D3	D2	D1	D0
DIVISION D7	DIVISION D6	DIVISION D5	DIVISION D4	DIVISION D3	DIVISION D2	DIVISION D1	DIVISION D0

[illegible] H'00 [illegible]

D7--D0 DIVISION D7--D0

DIVISION TYPE [illegible]

D7--D0	H'FF	H'FE	H'FD		H'03	H'02	H'01	H'00
	256	255	254		4	3	2	1

[illegible] COUNT TYPE [illegible]

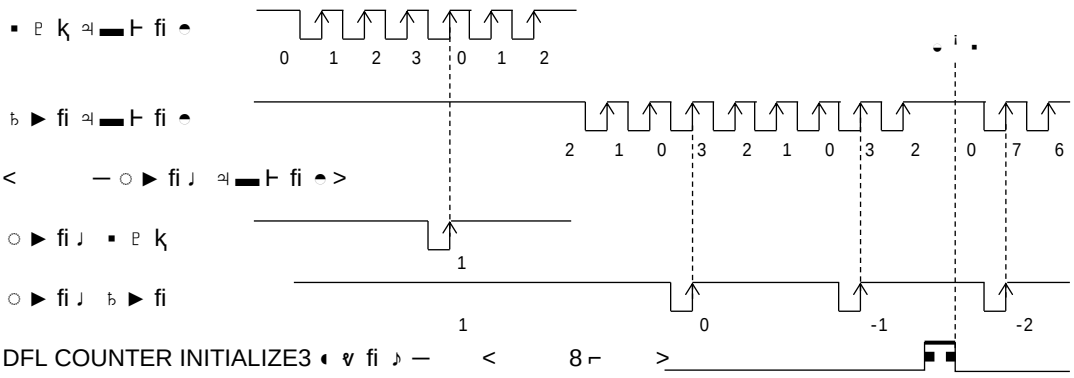
[illegible]

[illegible] COUNT PULSE SEL [illegible] DIVISION TYPE [illegible]

[illegible] COUNT TYPE [illegible]

[illegible]

[illegible] DFL COUNTER INITIALIZE3 [illegible]



DFL COUNTER INITIALIZE3 [illegible] 8 [illegible]

(4) DFL COUNTER PRESET

R ⁱ ☻ ئجر ○ ► fi ㅋ — ○ ► fi ♩ ثم ⇓ ⇔ ┐ Δ ⁹
↓ — ◖ ∀ fi ♪ — │ ≠ Δ ⁹

< ◦ ↄ ◔ fi ☻ >

COMMAND : H'A0 DFL COUNTER PRESET COMMAND

DATA1 PORT WRITE

COMMAND PORT WRITE

ŋ DRIVE DATA1 PORT ⌐ ○ ► fi ♩ ثم D15--D0 ⇓ ⅝ ╗ ┐ Δ ⁹
│ ⁸
-32,767 +32,767 H'8001 H'7FFF ≠ Δ ⁹
— │ 2 — ⌐ ⇔ ┐ Δ ⁹

º DRIVE COMMAND PORT ⌐ COMMAND ⇓ ⅝ ╗ ┐ Δ ⁹

DRIVE DATA1 PORT — ╗ ⇔ ♦ ↄ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←						○ ► fi ♩		ثم					→	D0

ŭ ˡ ☿ ᵇ ♩ — ثم │ H'0000 ≠ Δ ⁹

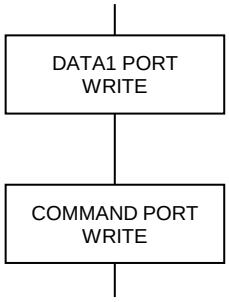
ה º ⌐ │ ⁸ H'8000 ⇓ Δ ╟ ↓ ≤ ⅝ ≠ ┐ Δ ⁹
ם ⇔ ⁸ H'8000 ⇓ Δ ╟ ≤ ⁸ STATUS4 PORT — DFL OVF = 1 ⌐ ⌂ ┤ ┐ Δ ⁹

(5) DFLINT COMPARE REGISTER1,2,3 SET

R ⁱ ☻ ئج ○ ► fi ㅋ — COMPARE REGISTER1, 2, 3 ⌐ ㅇ ╩ ⇔ ╕ Δ ⁹
↓ — ◖ ∀ fi ♪ — │ ≠ Δ ⁹

COMMAND : H'A8	DFLINT COMPARE REGISTER1 SET COMMAND
COMMAND : H'A9	DFLINT COMPARE REGISTER2 SET COMMAND
COMMAND : H'AA	DFLINT COMPARE REGISTER3 SET COMMAND

< ◦ ↄ ◔ fi ☻ >



ㅁ DRIVE DATA1 PORT ⌐ ㅇ D15--D0 ╩ ⅝ ╖ ╕ Δ ⁹
│ ⁸
-32,767 +32,767 H'8001 H'7FFF ≠ Δ ⁹
— │ 2 — ⌐ ⇔ ╕ Δ ⁹

ㅇ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

DRIVE DATA1 PORT — ╖ ⇔ ♦ ↄ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←							ㅇ						→	D0

ㅁ ⁱ ☿ ㄹ ㄱ — ㅊ │ H'8000 ≠ Δ ⁹

ㅁ ㅊ │ ⁸ DFL COUNTER INITIALIZE2 ◖ ∀ fi ♪ —COMP1, 2, 3 — COMP DETECT TYPE —
⌐ ╟ ⁸ ㅊ ╕ √ │ ⅝ ㅈ — ♦ ↄ ㅋ ⌐ ⌂ ╡ ╕ Δ ⁹

ㅁ COMP DETECT TYPE = 0 — ㅊ
ㅊ ╩ ⌐ ㅊ ⇔ ≡ ⁸ ⌐ ㅊ ⇔ √ ○ ► fi ㅋ ㅊ ≤ ⇔ ╕ Δ ⁹
|H'8001 H'FFFF| = +32,767 +1 ⌐ ⌂ ╡ ╕ Δ ⁹
|H'0000 H'7FFF| = 0 +32,767 ⌐ ⌂ ╡ ╕ Δ ⁹

ㅁ COMP DETECT TYPE = 1 — ㅈ ⅝
ㅊ │ / — ╕ ╕ ㅈ ⅝ — ㅊ ≠ ⁸ ㅈ ⅝ — ○ ► fi ㅋ ㅊ ≤ ⇔ ╕ Δ ⁹
H'8001 H'7FFF = -32,767 +32,767 ≠ Δ ⁹

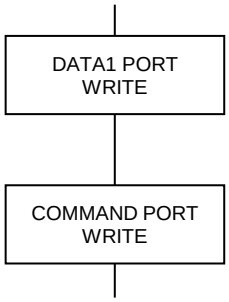
(6) DFLINT COMP1 ADD DATA SET

R ⁱ ☻ ئجر ○ ► fi ㅋ — COMP1 — ♦ ⊃ ㅋ ⇓ ⇔ ㅋ Δ ⁹
↓ — ◖ ∀ fi ♪ — | ≠ Δ ⁹

< ◦ ⊃ ◔ fi ☻ >

COMMAND : H'AC

DFLINT COMP1 ADD DATA SET COMMAND



ŋ DRIVE DATA1 PORT ⌐ COMP1 ADD ♦ ⊃ ㅋ D15--D0 ⇓ ⅝ ㅠ ㅋ Δ ⁹
| ⁸
-32,767 +32,767 H'8001 H'7FFF ≠ Δ ⁹
— | 2 — ⌐ ⇔ ㅋ Δ ⁹

º DRIVE COMMAND PORT ⌐ COMMAND ⇓ ⅝ ㅠ ㅋ Δ ⁹

DRIVE DATA1 PORT — ㅠ ⇔ ♦ ⊃ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←					COMP1 ADD ♦ ⊃ ㅋ								→	D0

ὶ ᵢ ᾗ ᵉ ᵢ — ثم | H'0000 ≠ Δ ⁹

2-2-4. [illegible]

[illegible]

[illegible] DRIVE DATA1, 2 PORT (READ) [illegible]

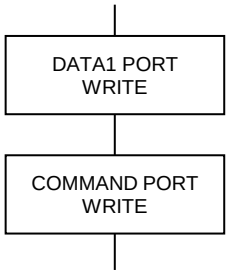
[illegible]

[illegible]

(1) COUNT LATCH SPEC SET

[illegible]

COMMAND : H'E8 — COUNT LATCH SPEC SET COMMAND



[illegible] DRIVE DATA1 PORT [illegible]

[illegible] DRIVE COMMAND PORT [illegible] COMMAND [illegible]

DRIVE DATA1 PORT [illegible]

D15	D14	D13	D12	D11	D10	D9	D8
				DFL CLR ENABLE	DFL LATCH TYPE2	DFL LATCH TYPE1	DFL LATCH TYPE0

D7	D6	D5	D4	D3	D2	D1	D0
PULSE CLR ENABLE	PULSE LATCH TYPE2	PULSE LATCH TYPE1	PULSE LATCH TYPE0	ADDRESS CLR ENABLE	ADDRESS LATCH TYPE2	ADDRESS LATCH TYPE1	ADDRESS LATCH TYPE0

[illegible] H'000 ([illegible])

D0 ADDRESS LATCH TYPE0
D1 ADDRESS LATCH TYPE1
D2 ADDRESS LATCH TYPE2

[illegible]

TYPE2	TYPE1	TYPE0	[illegible]
0	0	0	ADDRESS LATCH DATA READ [illegible]
0	0	1	
0	1	0	STATUS5 PORT [illegible] SS0 = 0 [illegible] 1 [illegible]
0	1	1	ORIGIN SPEC SET [illegible] ORG [illegible]
1	0	0	
1	0	1	
1	1	0	
1	1	1	

D3 ADDRESS CLR ENABLE

[illegible]

0 [illegible]

1 [illegible]

D4 PULSE LATCH TYPE0
D5 PULSE LATCH TYPE1
D6 PULSE LATCH TYPE2

TYPE2	TYPE1	TYPE0	
0	0	0	PULSE LATCH DATA READ
0	0	1	
0	1	0	STATUS5 PORT ― SS0 = 0
0	1	1	ORIGIN SPEC SET ORG
1	0	0	
1	0	1	
1	1	0	
1	1	1	

D7 PULSE CLR ENABLE

0
1

D8 DFL LATCH TYPE0
D9 DFL LATCH TYPE1
D10 DFL LATCH TYPE2

TYPE2	TYPE1	TYPE0	
0	0	0	DFL LATCH DATA READ
0	0	1	
0	1	0	STATUS5 PORT ― SS0 = 0
0	1	1	ORIGIN SPEC SET ORG
1	0	0	
1	0	1	
1	1	0	
1	1	1	

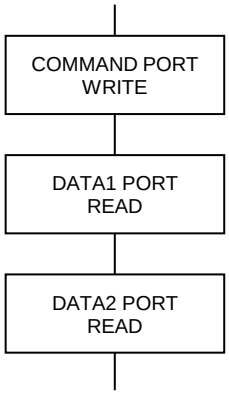
D11 DFL CLR ENABLE

0
1

2-2-5. ○ ▶ fi ♩ ♦ ↄ ㅋ ㅡ ╖ ⇔

Ẽ ○ ▶ fi ♩ ♦ ↄ ㅋ ㅡ ╖ ⇔ ◦ ↄ ◔ fi ☻

< ◦ ↄ ◔ fi ☻ >



ŋ DRIVE COMMAND PORT ⌐ COMMAND ╩ ⅝ ╖ ╕ Δ ⁹

º DRIVE DATA1 PORT ⅛ ╠ ○ ▶ fi ♩ ♦ ↄ ㅋ D15--D0 ╩ ╖ ⇔ ╕ Δ ⁹

ɔ DRIVE DATA2 PORT ⅛ ╠ ○ ▶ fi ♩ ♦ ↄ ㅋ D31--D16 ╩ ╖ ⇔ ╕ Δ ⁹

DRIVE DATA1 PORT ㅡ ╖ ⇔ ♦ ↄ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←						○ ▶ fi ♩ ♦ ↄ ㅋ							→	D0

DRIVE DATA2 PORT ㅡ ╖ ⇔ ♦ ↄ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←						○ ▶ fi ♩ ♦ ↄ ㅋ							→	D16

ㅁ COUNTER READ ◖ ∀ fi ♪ ╩ Δ ╢ ≤ ⁸ ○ ▶ fi ㅋ ㅡ ○ ▶ fi ♩ ♦ ↄ ㅋ ╩ DRIVE DATA1, 2 PORT (READ) ⌐ ⅜ ⅊ ♩ ⇔ ╕ Δ ⁹

(1) ADDRESS COUNTER READ

▪ ♪ ☻ ○ ▶ fi ㅋ ㅡ ○ ▶ fi ♩ ♦ ↄ ㅋ ╩ ╖ ⇔ ╕ Δ ⁹

↓ ㅡ ◖ ∀ fi ♪ ㅡ │ ≠ Δ ⁹

COMMAND : H'D8 ADDRESS COUNTER READ COMMAND

(2) PULSE COUNTER READ

Ꞁ ⁱ ☻ ○ ▶ fi ㅋ ㅡ ○ ▶ fi ♩ ♦ ↄ ㅋ ╩ ╖ ⇔ ╕ Δ ⁹

↓ ㅡ ◖ ∀ fi ♪ ㅡ │ ≠ Δ ⁹

COMMAND : H'D9 PULSE COUNTER READ COMMAND

(3) DFL COUNTER READ

Ꞁ ⁱ ☻ ○ ▶ fi ㅋ ㅡ ○ ▶ fi ♩ ♦ ↄ ㅋ ╩ ╖ ⇔ ╕ Δ ⁹

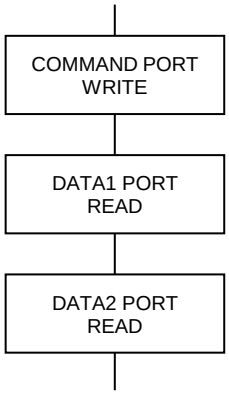
↓ ㅡ ◖ ∀ fi ♪ ㅡ │ ≠ Δ ⁹

COMMAND : H'DA DFL COUNTER READ COMMAND

2-2-6. ○ ▶ fi ♩ ♦ ⊃ ㅋ ー ! ㄹ ㅃ ♦ ⊃ ㅋ ー ╖ ⇔

Ẽ ! ㄹ ㅃ ♦ ⊃ ㅋ ー ╖ ⇔ ◦ ⊃ ◔ fi ☻

< ◦ ⊃ ◔ fi ☻ >



ŋ DRIVE COMMAND PORT ⌐ COMMAND ⇈ ⅝ ╖ ┤ Δ ⁹

º DRIVE DATA1 PORT ⅛ ╟ ○ ▶ fi ♩ ♦ ⊃ ㅋ D15--D0 ⇈ ╖ ⇔ ┤ Δ ⁹

ɔ DRIVE DATA2 PORT ⅛ ╟ ○ ▶ fi ♩ ♦ ⊃ ㅋ D31--D16 ⇈ ╖ ⇔ ┤ Δ ⁹

DRIVE DATA1 PORT ー ╖ ⇔ ♦ ⊃ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D15	←						ー ! ㄹ ㅃ ♦ ⊃ ㅋ							→	D0

DRIVE DATA2 PORT ー ╖ ⇔ ♦ ⊃ ㅋ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
D31	←						ー ! ㄹ ㅃ ♦ ⊃ ㅋ							→	D16

ㅁ ADDRESS LATCH DATA READ ◂ ∀ fi ♪ ┤ √ │ PULSE LATCH DATA READ ◂ ∀ fi ♪ ⇈ Δ ╢ ≤ ⁸
○ ▶ fi ㅋ ー ! ㄹ ㅃ ♦ ⊃ ㅋ ⇈ DRIVE DATA1, 2 PORT READ ⌐ ⅛ ㄹ ♩ ⇔ ┤ Δ ⁹

ㅁ DFL LATCH DATA READ ◂ ∀ fi ♪ ⇈ Δ ╢ ≤ ⁸ ○ ▶ fi ㅋ ー ! ㄹ ㅃ ♦ ⊃ ㅋ ⇈ DRIVE DATA1 PORT
（READ）にセットします。

(1) ADDRESS LATCH DATA READ

▪ ♪ ☻ ○ ▶ fi ㅋ ー ! ㄹ ㅃ ♦ ⊃ ㅋ ⇈ ╖ ⇔ ┤ Δ ⁹
↓ ー ◂ ∀ fi ♪ ー │ ≠ Δ ⁹

COMMAND : H'DC　ADDRESS LATCH DATA READ COMMAND

(2) PULSE LATCH DATA READ

R ⁱ ☻ ○ ▶ fi ㅋ ー ! ㄹ ㅃ ♦ ⊃ ㅋ ⇈ ╖ ⇔ ┤ Δ ⁹
↓ ー ◂ ∀ fi ♪ ー │ ≠ Δ ⁹

COMMAND : H'DD　PULSE LATCH DATA READ COMMAND

(3) DFL LATCH DATA READ

R ⁱ ☻ ○ ▶ fi ㅋ ー ! ㄹ ㅃ ♦ ⊃ ㅋ ⇈ ╖ ⇔ ┤ Δ ⁹
↓ ー ◂ ∀ fi ♪ ー │ ≠ Δ ⁹

COMMAND : H'DE　DFL LATCH DATA READ COMMAND

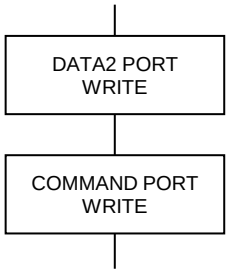
2-3. HARD CONFIG ◖ ∀ fi ♪

2-3-1. طص ث —

(1) MAN MASK

I/O ◖ Ł ▾ ㄱ (J3)— $\overline{\text{MAN}}$ كخ (MANUAL ⁄ ⊃ ♪ ⊣ ⊣ كخ ⊣)—طص ⊥⊥ ∀ ☻ ▾ ⇔ ⊓ Δ ⁹
↓ — ◖ ∀ fi ♪ — | ≠ Δ ⁹

< ◦ ⊃ ◗ fi ☻ > COMMAND : H'00 MAN MASK COMMAND



ŋ HARD CONFIG DATA2 PORT ⌐ ∀ ☻ ▾ ♦ ⊃ ㄱ ⊥⊥ ⅝ ⊓ ⊓ Δ ⁹

º HARD CONFIG COMMAND PORT ⌐ COMMAND ⊥⊥ ⅝ ⊓ ⊓ Δ ⁹

CONFIG DATA2 PORT — ♦ ⊃ ㄱ

D15	D14	D13	D12	D11	D10	D9	D8

D7	D6	D5	D4	D3	D2	D1	D0
			MAN MASK				

ÙÞ ¹ ⁸ ⁹ ♩ — ثم | ∀ ☻ ▾ ⇔ ⌂ ™≠ Δ ⁹

D4 MAN MASK

0 ∀ ☻ ▾ ⇔ ⌂ ™

1 ∀ ☻ ▾ Δ ⊣⊣

I/O ◖ Ł ▾ ㄱ — $\overline{\text{MAN}}$ كخ ⌐ ⊣⊣ MANUAL ⁄ ⊃ ♪ ⊣ ⊣ ⊥⊥ ▪ ⅊ ⁱ ◦ ⊃ ◗ ◦ ↑ fi ⅛ ⊣⊣
Δ ⊣⊣ ↓ ≤ ⅜≠ ⅝ ⊓ Δ ⁹

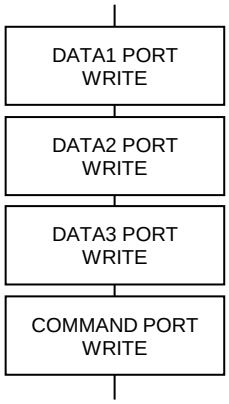
ה ⁸ ∀ ☻ ▾ Δ ⊣⊣ ⁹ ⌐ Δ ⊣⊣ ≤ ⁸ MANUAL ⁄ ⊃ ♪ — ⊣ ⊣ ⅜ $\overline{\text{MAN}}$ كخ ⅛ ⊣⊣ لأ — ⌐ ⌂ ⊣ ⊓ Δ ⁹
↓ — ≤ ⅝ ⁸ HARD CONFIG STATUS1 PORT — $\overline{\text{MAN RDY}}$=0 ⁸ I/O ◖ Ł ▾ ㄱ — $\overline{\text{MAN RDY}}$=HIGH
⌐ ⇔ ⊓ Δ ⁹

ה ⁸ ∀ ☻ ▾ ⇔ ⌂ ™ ⁹ ⌐ Δ ⊣⊣ ≤ ⁸ MANUAL ⁄ ⊃ ♪ — ⊣ ⊣ ⅜ $\overline{\text{MAN}}$ كخ ⅛ ⊣⊣ ⌂ ⌐ ⌂ ⊣ ⊓ Δ ⁹
↓ — ≤ ⅝ ⁸ HARD CONFIG STATUS1 PORT — $\overline{\text{MAN RDY}}$=1 ⁸ I/O ◖ Ł ▾ ㄱ — $\overline{\text{MAN RDY}}$=LOW
⌐ ⇔ ⊓ Δ ⁹

(2) SENSOR SIGNAL SELECT

各軸 MCC07E の多用途センサ信号(SS0,SS1 信号)に接続する信号を設定します。

< 実行シーケンス >



COMMAND : H'01　SENSOR SIGNAL SELECT COMMAND

① HARD CONFIG DATA1 PORT に設定データを書き込みます。

② HARD CONFIG DATA2 PORT に設定データを書き込みます。

③ HARD CONFIG DATA3 PORT に設定データを書き込みます。

④ HARD CONFIG COMMAND PORT に COMMAND を書き込みます。

HARD CONFIG DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	–	–	–	–	–	–	←		AXIS SEL	→	

HARD CONFIG DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	–	–	–	–	–	–	–	–	–	← SIGNAL GATE TYPE →	

HARD CONFIG DATA3 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
			←	SIGNAL2 SEL			→				←	SIGNAL1 SEL			→

■ HARD CONFIG DATA1 PORT の内容

D4-0 ：AXIS SEL

設定の対象多用途センサを選択します。

AXIS SEL	対象多用途センサ			
	C-VX870v1 C-VX870Ev1	C-VX871v1 C-VX871Ev1	C-VX872v1 C-VX872Ev1	C-VX873v1 C-VX873Ev1
H'00	X軸 SS0	X軸 SS0	X1軸 SS0	X1軸 SS0
H'01	X軸 SS1	X軸 SS1	X1軸 SS1	X1軸 SS1
H'02	Y軸 SS0	Y軸 SS0	Y1軸 SS0	Y1軸 SS0
H'03	Y軸 SS1	Y軸 SS1	Y1軸 SS1	Y1軸 SS1
H'04	Z軸 SS0	Z軸 SS0	Z1軸 SS0	Z1軸 SS0
H'05	Z軸 SS1	Z軸 SS1	Z1軸 SS1	Z1軸 SS1
H'06	A軸 SS0	A軸 SS0	A1軸 SS0	A1軸 SS0
H'07	A軸 SS1	A軸 SS1	A1軸 SS1	A1軸 SS1
H'08	設定禁止(無効)	B軸 SS0	設定禁止(無効)	B1軸 SS0
H'09	設定禁止(無効)	B軸 SS1	設定禁止(無効)	B1軸 SS1
H'0A	設定禁止(無効)	C軸 SS0	設定禁止(無効)	C1軸 SS0
H'0B	設定禁止(無効)	C軸 SS1	設定禁止(無効)	C1軸 SS1
H'0C～H'0F	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)
H'10	設定禁止(無効)	設定禁止(無効)	X2軸 SS0	X2軸 SS0
H'11	設定禁止(無効)	設定禁止(無効)	X2軸 SS1	X2軸 SS1
H'12	設定禁止(無効)	設定禁止(無効)	Y2軸 SS0	Y2軸 SS0
H'13	設定禁止(無効)	設定禁止(無効)	Y2軸 SS1	Y2軸 SS1
H'14	設定禁止(無効)	設定禁止(無効)	Z2軸 SS0	Z2軸 SS0
H'15	設定禁止(無効)	設定禁止(無効)	Z2軸 SS1	Z2軸 SS1
H'16	設定禁止(無効)	設定禁止(無効)	A2軸 SS0	A2軸 SS0
H'17	設定禁止(無効)	設定禁止(無効)	A2軸 SS1	A2軸 SS1
H'18	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)	B2軸 SS0
H'19	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)	B2軸 SS1
H'1A	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)	C2軸 SS0
H'1B	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)	C2軸 SS1
H'1C～H'1F	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)

■ HARD CONFIG DATA2 PORT の内容

D1-0：SIGNAL GATE TYPE

なしを選択した場合は、SIGNAL1 SEL で選択した信号をそのまま接続します。
ANDまたは"OR"を選択した場合は SIGNAL1 SEL と SIGNAL2 SEL で選択した信号の指定合成論理信号を接続します。

SIGNAL GATE TYPE	合成論理
H'0	なし(SIGNAL1 SEL 選択信号を使用)
H'1	AND
H'2	OR
H'3	設定禁止

● リセット後の初期値はアンダーライン側です

■ HARD CONFIG DATA3 PORT の内容

D4-0　：SIGNAL1 SEL

D12-8 ：SIGNAL2 SEL

合成論理に使用する信号を選択します。

SIGNAL SEL	対象多用途センサ			
	C-VX870v1 C-VX870Ev1	C-VX871v1 C-VX871Ev1	C-VX872v1 *1 C-VX872Ev1	C-VX873v1 *1 C-VX873Ev1
H'00	X 軸 OUTA	X 軸 OUTA	Xn 軸 OUTA	Xn 軸 OUTA
H'01	X 軸 OUTB	X 軸 OUTB	Xn 軸 OUTB	Xn 軸 OUTB
H'02	Y 軸 OUTA	Y 軸 OUTA	Yn 軸 OUTA	Yn 軸 OUTA
H'03	Y 軸 OUTB	Y 軸 OUTB	Yn 軸 OUTB	Yn 軸 OUTB
H'04	Z 軸 OUTA	Z 軸 OUTA	Zn 軸 OUTA	Zn 軸 OUTA
H'05	Z 軸 OUTB	Z 軸 OUTB	Zn 軸 OUTB	Zn 軸 OUTB
H'06	A 軸 OUTA	A 軸 OUTA	An 軸 OUTA	An 軸 OUTA
H'07	A 軸 OUTB	A 軸 OUTB	An 軸 OUTB	An 軸 OUTB
H'08	なし(LOW レベル固定)	B 軸 OUTA	なし(LOW レベル固定)	Bn 軸 OUTA
H'09	なし(LOW レベル固定)	B 軸 OUTB	なし(LOW レベル固定)	Bn 軸 OUTB
H'0A	なし(LOW レベル固定)	C 軸 OUTA	なし(LOW レベル固定)	Cn 軸 OUTA
H'0B	なし(LOW レベル固定)	C 軸 OUTB	なし(LOW レベル固定)	Cn 軸 OUTB
H'0C	$\overline{\text{SENSOR0}}$ 入力信号	$\overline{\text{SENSOR0}}$ 入力信号	$\overline{\text{SENSORn0}}$ 入力信号	$\overline{\text{SENSORn0}}$ 入力信号
H'0D	$\overline{\text{SENSOR1}}$ 入力信号	$\overline{\text{SENSOR1}}$ 入力信号	$\overline{\text{SENSORn1}}$ 入力信号	$\overline{\text{SENSORn1}}$ 入力信号
H'0E	$\overline{\text{SIGNAL IN0}}$ 入力信号	$\overline{\text{SIGNAL IN0}}$ 入力信号	$\overline{\text{SIGNAL INn0}}$ 入力信号	$\overline{\text{SIGNAL INn0}}$ 入力信号
H'0F	$\overline{\text{SIGNAL IN1}}$ 入力信号	$\overline{\text{SIGNAL IN1}}$ 入力信号	$\overline{\text{SIGNAL INn1}}$ 入力信号	$\overline{\text{SIGNAL INn1}}$ 入力信号
H'1x	なし(LOW レベル固定)	なし(LOW レベル固定)	なし(LOW レベル固定)	なし(LOW レベル固定)

● リセット後の初期値は以下の通りです。

対象多用途センサ	SIGNAL1 SEL	SIGNAL2 SEL
Zn 軸SS0	$\overline{\text{SENSORn0}}$ 入力信号	なし(LOW レベル固定)
An 軸SS0	$\overline{\text{SENSORn1}}$ 入力信号	なし(LOW レベル固定)
その他の軸SS0,SS1	なし(LOW レベル固定)	なし(LOW レベル固定)

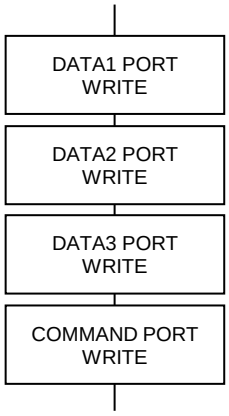
*1 C-VX872v1, C-VX872Ev1, C-VX873v1, C-VX873Ev1 の場合、
使用できる信号は設定の対象と同じ系列軸内の信号になります。
例) X2 軸 SS0 の設定の場合、n=2

(3) SIGNAL OUT SELECT

特殊 I/O コネクタの SIGNAL OUTnx 信号(ステータス出力信号)に出力する信号を設定します。

< 実行シーケンス >

COMMAND : H'04　SIGNAL OUT SELECT COMMAND



① HARD CONFIG DATA1 PORT に設定データを書き込みます。

② HARD CONFIG DATA2 PORT に設定データを書き込みます。

③ HARD CONFIG DATA3 PORT に設定データを書き込みます。

④ HARD CONFIG COMMAND PORT に COMMAND を書き込みます。

HARD CONFIG DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	–	–	–	–	–	–				← SIGNAL OUT SEL →	

HARD CONFIG DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	–	–	–	–	–	← SIGNAL OUT TYPE →		–	–	← SIGNAL GATE TYPE →	

HARD CONFIG DATA3 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
				← SIGNAL2 SEL →								← SIGNAL1 SEL →			

■ HARD CONFIG DATA1 PORT の内容

D1-0 ：SIGNAL OUT SEL

設定の対象ステータス出力信号を選択します。

SIGNAL OUT SEL	対象ステータス出力信号	
	C-VX870v1, C-VX870Ev1 C-VX871v1, C-VX871Ev1	C-VX872v1, C-VX872Ev1 C-VX873v1, C-VX873Ev1
H'0	$\overline{\text{SIGNAL OUT0}}$	$\overline{\text{SIGNAL OUT10}}$
H'1	$\overline{\text{SIGNAL OUT1}}$	$\overline{\text{SIGNAL OUT11}}$
H'2	設定禁止(無効)	$\overline{\text{SIGNAL OUT20}}$
H'3	設定禁止(無効)	$\overline{\text{SIGNAL OUT21}}$

■ HARD CONFIG DATA2 PORT の内容

D1-0 ：SIGNAL GATE TYPE1-0

なしを選択した場合は、SIGNAL1 SEL で選択した信号を出力方式に応じて出力します。
ANDまたは"OR"を選択した場合は SIGNAL1 SEL と SIGNAL2 SEL で選択した信号の指定合成論理信号を出力方式に応じて出力します。

SIGNAL GATE TYPE	合成論理
H'0	なし(SIGNAL1 SEL 選択信号を使用)
H'1	AND
H'2	OR
H'3	設定禁止

● リセット後の初期値はアンダーライン側です

D5-4 ：SIGNAL OUT TYPE

出力方式を設定します。

SIGNAL OUT TYPE	出力方式
H'0	合成論理信号をスルー出力
H'1	合成論理信号のアクティブエッジでワンショット出力
H'2	合成論理信号のノットアクティブエッジでワンショット出力
H'3	設定禁止(合成論理信号をスルー出力)

● リセット後の初期値はアンダーライン側です

・ ワンショット出力に設定した場合、合成論理信号の指定されたエッジを検出すると SIGNAL OUT TIMER SET コマンドで設定された時間、アクティブレベルを出力します。

■ HARD CONFIG DATA3 PORT の内容

D3-0 ：SIGNAL1 SEL

D11-8 ：SIGNAL2 SEL

合成論理に使用する信号を選択します。

SIGNAL SEL	合成論理に使用する信号			
	C-VX870v1 C-VX870Ev1	C-VX871v1 C-VX871Ev1	C-VX872v1 *1 C-VX872Ev1	C-VX873v1 *1 C-VX873Ev1
H'00	X 軸 OUTA	X 軸 OUTA	Xn 軸 OUTA	Xn 軸 OUTA
H'01	X 軸 OUTB	X 軸 OUTB	Xn 軸 OUTB	Xn 軸 OUTB
H'02	Y 軸 OUTA	Y 軸 OUTA	Yn 軸 OUTA	Yn 軸 OUTA
H'03	Y 軸 OUTB	Y 軸 OUTB	Yn 軸 OUTB	Yn 軸 OUTB
H'04	Z 軸 OUTA	Z 軸 OUTA	Zn 軸 OUTA	Zn 軸 OUTA
H'05	Z 軸 OUTB	Z 軸 OUTB	Zn 軸 OUTB	Zn 軸 OUTB
H'06	A 軸 OUTA	A 軸 OUTA	An 軸 OUTA	An 軸 OUTA
H'07	A 軸 OUTB	A 軸 OUTB	An 軸 OUTB	An 軸 OUTB
H'08	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	B 軸 OUTA	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	Bn 軸 OUTA
H'09	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	B 軸 OUTB	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	Bn 軸 OUTB
H'0A	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	C 軸 OUTA	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	Cn 軸 OUTA
H'0B	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	C 軸 OUTB	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	Cn 軸 OUTB
H'0C	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)
H'0D	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)
H'0E	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)
H'0F	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)
H'1x	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)	なし(ﾉｯﾄｱｸﾃｨﾌﾞ固定)

● リセット後の初期値は以下の通りです。

対象ステータス出力	SIGNAL1 SEL	SIGNAL2 SEL
$\overline{\text{SIGNAL OUTn0}}$	Xn 軸 OUTA	Xn 軸 OUTA
$\overline{\text{SIGNAL OUTn1}}$	Yn 軸 OUTA	Xn 軸 OUTA

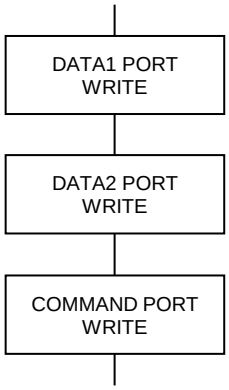
*1 C-VX872v1, C-VX872Ev1, C-VX873v1, C-VX873Ev1 の場合、
使用できる信号は設定の対象と同じ系列軸内の信号になります。
例) $\overline{\text{SIGNAL OUT20}}$ の設定の場合、n=2

(4) SIGNAL OUT TIMER SET

特殊 I/O コネクタの SIGNAL OUTnx 信号(ステータス出力信号)の出力方式をワンショット出力とした場合の出力時間を設定します。

< 実行シーケンス >

COMMAND : H'05　SIGNAL OUT TIMER SET COMMAND



① HARD CONFIG DATA1 PORT に設定データを書き込みます。

② HARD CONFIG DATA2 PORT に設定データを書き込みます。
設定範囲は、0 ～ 65,535（H'0000 ～ H'FFFF）です。
設定値は、1 μ s 単位です。

③ HARD CONFIG COMMAND PORT に COMMAND を書き込みます。

HARD CONFIG DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	–	–	–	–	–	–	–	–	–	← SIGNAL OUT SEL →	

HARD CONFIG DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
← ワンショット出力時間 →															

● リセット後の初期値は H'0001(1 μ s)です。
・ H'0000 を設定した場合、出力時間は 1 μ s に補正して出力します。

D1-0 ：SIGNAL OUT SEL 1-0
設定の対象ステータス出力信号を選択します。

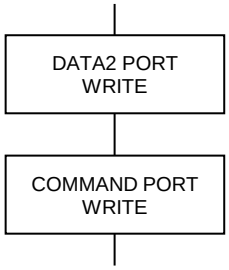
SIGNAL OUT SEL	対象ステータス出力信号	
	C-VX870v1, C-VX870Ev1 C-VX871v1, C-VX871Ev1	C-VX872v1, C-VX872Ev1 C-VX873v1, C-VX873Ev1
H'0	SIGNAL OUT0	SIGNAL OUT10
H'1	SIGNAL OUT1	SIGNAL OUT11
H'2	設定禁止(無効)	SIGNAL OUT20
H'3	設定禁止(無効)	SIGNAL OUT21

(5) SIGNAL OUT LATCH STATUS CLR

HARD CONFIG STATUS4 PORT の SIGNAL OUTnx LATCH フラグを個別にクリアします。
このコマンドの実行は常時可能です。

HARD CONFIG STATUS4 PORT の SIGNAL OUTnx LATCH フラグは対応する出力信号の
アクティブエッジを検出をラッチします。当コマンドはこのフラグのクリアに使用します。

< 実行シーケンス >　　COMMAND : H'06　　SIGNAL OUT STATUS CLR COMMAND



① HARD CONFIG DATA2 PORT にクリアデータを書き込みます。

② HARD CONFIG COMMAND PORT に COMMAND を書き込みます。

HARD CONFIG DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	
–	–	–	–	SIGNAL OUT21 LATCH CLR	SIGNAL OUT20 LATCH CLR	–	–	C-VX872v1,C-VX872Ev1 C-VX873v1,C-VX873Ev1
				–	–			

D7	D6	D5	D4	D3	D2	D1	D0	
–	–	–	–	SIGNAL OUT11 LATCH CLR	SIGNAL OUT10 LATCH CLR	–	–	C-VX872v1,C-VX872Ev1 C-VX873v1,C-VX873Ev1
				SIGNAL OUT1 LATCH CLR	SIGNAL OUT0 LATCH CLR			C-VX870v1,C-VX870Ev1 C-VX871v1,C-VX871Ev1

D3 - D2　：クリアデータ
D11-D10 ：クリアデータ
　SIGNAL OUT LATCH のクリアデータを選択します。
　　0 ：クリアしない
　　1 ：クリアする

・ コマンドの実行で、SIGNAL OUT LATCH フラグをクリアします。
　このコマンドのデータは、コマンド実行時のみ有効です。（トリガ入力）

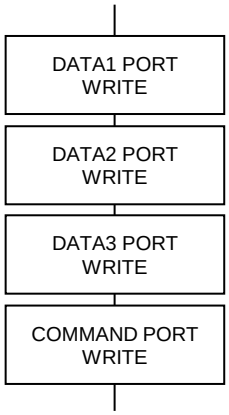
2-3-2. 同期スタート機能の設定と実行

(1) PAUSE SET SPEC

同期スタートさせる軸の PAUSE 信号を ON にする条件を設定します。
このコマンドの実行は常時可能です。

< 実行シーケンス >

COMMAND : H'11　PAUSE SET SPEC COMMAND



① HARD CONFIG DATA1 PORT に設定データを書き込みます。

② HARD CONFIG DATA2 PORT に設定データを書き込みます。

③ HARD CONFIG DATA3 PORT に設定データを書き込みます。

④ HARD CONFIG COMMAND PORT に COMMAND を書き込みます。

HARD CONFIG DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	–	–	–	–	–	–				← AXIS SEL →	

HARD CONFIG DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	–	–	–	–	–	← SIGNAL GATE TYPE →		–	–	← PAUSE SET TYPE →	

HARD CONFIG DATA3 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
				← SIGNAL2 SEL →								← SIGNAL1 SEL →			

■ HARD CONFIG DATA1 PORT の内容

D4-0 ：AXIS SEL

設定の対象軸を選択します。

AXIS SEL	設定の対象軸			
	C-VX870v1 C-VX870Ev1	C-VX871v1 C-VX871Ev1	C-VX872v1 C-VX872Ev1	C-VX873v1 C-VX873Ev1
H'00	X 軸	X 軸	X1 軸	X1 軸
H'01	Y 軸	Y 軸	Y1 軸	Y1 軸
H'02	Z 軸	Z 軸	Z1 軸	Z1 軸
H'03	A 軸	A 軸	A1 軸	A1 軸
H'04	設定禁止(無効)	B 軸	設定禁止(無効)	B1 軸
H'05	設定禁止(無効)	C 軸	設定禁止(無効)	C1 軸
H'06～H'0F	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)
H'10	設定禁止(無効)	設定禁止(無効)	X2 軸	X2 軸
H'11	設定禁止(無効)	設定禁止(無効)	Y2 軸	Y2 軸
H'12	設定禁止(無効)	設定禁止(無効)	Z2 軸	Z2 軸
H'13	設定禁止(無効)	設定禁止(無効)	A2 軸	A2 軸
H'14	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)	B2 軸
H'15	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)	C2 軸
H'16～H'1F	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)

■ HARD CONFIG DATA2 PORT の内容

D1-0 ：PAUSE SET TYPE

PAUSE SET 条件を選択します。

PAUSE SET TYPE	PAUSE SET 条件
H'0	PAUSE コマンド実行で PAUSE SET
H'1	合成論理信号のアクティブエッジ検出で PAUSE SET
H'2	合成論理信号のノットアクティブエッジ検出で PAUSE SET
H'3	設定禁止(PAUSE コマンド実行で PAUSE SET)

● リセット後の初期値はアンダーライン側です

D5-4 ：SIGNAL GATE TYPE

PAUSE SET 条件を「合成論理信号のアクティブエッジ／ノットアクティブエッジ検出」とした場合は、合成論理を設定します。

なしを選択した場合は、SIGNAL1 SEL で選択した信号のエッジ検出を PAUSE SET 条件にします。

ANDまたは"OR"を選択した場合は SIGNAL1 SEL と SIGNAL2 SEL で選択した信号の指定合成論理信号のエッジ検出を PAUSE SET 条件にします。

SIGNAL GATE TYPE	合成論理
H'0	なし(SIGNAL1 SEL 選択信号を使用)
H'1	AND
H'2	OR
H'3	設定禁止

● リセット後の初期値はアンダーライン側です

■ HARD CONFIG DATA3 PORT の内容

D3-0 ：SIGNAL1 SEL

D11-8 ：SIGNAL2 SEL

合成論理に使用する信号を選択します。

SIGNAL SEL	合成論理に使用する信号			
	C-VX870v1 C-VX870Ev1	C-VX871v1 C-VX871Ev1	C-VX872v1 *1 C-VX872Ev1	C-VX873v1 *1 C-VX873Ev1
H'00	X 軸 OUTA	X 軸 OUTA	Xn 軸 OUTA	Xn 軸 OUTA
H'01	X 軸 OUTB	X 軸 OUTB	Xn 軸 OUTB	Xn 軸 OUTB
H'02	Y 軸 OUTA	Y 軸 OUTA	Yn 軸 OUTA	Yn 軸 OUTA
H'03	Y 軸 OUTB	Y 軸 OUTB	Yn 軸 OUTB	Yn 軸 OUTB
H'04	Z 軸 OUTA	Z 軸 OUTA	Zn 軸 OUTA	Zn 軸 OUTA
H'05	Z 軸 OUTB	Z 軸 OUTB	Zn 軸 OUTB	Zn 軸 OUTB
H'06	A 軸 OUTA	A 軸 OUTA	An 軸 OUTA	An 軸 OUTA
H'07	A 軸 OUTB	A 軸 OUTB	An 軸 OUTB	An 軸 OUTB
H'08	なし(LOW レベル固定)	B 軸 OUTA	なし(LOW レベル固定)	Bn 軸 OUTA
H'09	なし(LOW レベル固定)	B 軸 OUTB	なし(LOW レベル固定)	Bn 軸 OUTB
H'0A	なし(LOW レベル固定)	C 軸 OUTA	なし(LOW レベル固定)	Cn 軸 OUTA
H'0B	なし(LOW レベル固定)	C 軸 OUTB	なし(LOW レベル固定)	Cn 軸 OUTB
H'0C	$\overline{\text{SENSOR0}}$ 入力信号	$\overline{\text{SENSOR0}}$ 入力信号	$\overline{\text{SENSORn0}}$ 入力信号	$\overline{\text{SENSORn0}}$ 入力信号
H'0D	$\overline{\text{SENSOR1}}$ 入力信号	$\overline{\text{SENSOR1}}$ 入力信号	$\overline{\text{SENSORn1}}$ 入力信号	$\overline{\text{SENSORn1}}$ 入力信号
H'0E	$\overline{\text{SIGNAL IN0}}$ 入力信号	$\overline{\text{SIGNAL IN0}}$ 入力信号	$\overline{\text{SIGNAL INn0}}$ 入力信号	$\overline{\text{SIGNAL INn0}}$ 入力信号
H'0F	$\overline{\text{SIGNAL IN1}}$ 入力信号	$\overline{\text{SIGNAL IN1}}$ 入力信号	$\overline{\text{SIGNAL INn1}}$ 入力信号	$\overline{\text{SIGNAL INn1}}$ 入力信号

● リセット後の初期値はアンダーライン側です

*1 C-VX872v1, C-VX872Ev1, C-VX873v1, C-VX873Ev1 の場合、
使用できる信号は設定の対象と同じ系列軸内の信号になります。
例) X2 軸の設定の場合、n=2

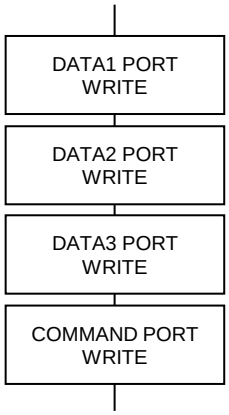
・ 対象軸の PAUSE SET 条件と PAUSE CLR 条件が同じ場合は、PAUSE 信号は ON しません。
(PAUSE コマンドの場合は除く)

(2) PAUSE CLR SPEC

同期スタートさせる軸の PAUSE 信号を OFF にする条件を設定します。
このコマンドの実行は常時可能です。

< 実行シーケンス >

COMMAND : H'12　PAUSE CLR SPEC COMMAND



① HARD CONFIG DATA1 PORT に設定データを書き込みます。

② HARD CONFIG DATA2 PORT に設定データを書き込みます。

③ HARD CONFIG DATA3 PORT に設定データを書き込みます。

④ HARD CONFIG COMMAND PORT に COMMAND を書き込みます。

HARD CONFIG DATA1 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	–	–	–	–	–	–				← AXIS SEL →	

HARD CONFIG DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
–	–	–	–	–	–	–	–	–	–	← SIGNAL GATE TYPE →		–	–	← PAUSE CLR TYPE →	

HARD CONFIG DATA3 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
				← SIGNAL2 SEL →								← SIGNAL1 SEL →			

■ HARD CONFIG DATA1 PORT の内容

D4-0 ：AXIS SEL

設定の対象軸を選択します。

AXIS SEL	設定の対象軸			
	C-VX870v1 C-VX870Ev1	C-VX871v1 C-VX871Ev1	C-VX872v1 C-VX872Ev1	C-VX873v1 C-VX873Ev1
H'00	X 軸	X 軸	X1 軸	X1 軸
H'01	Y 軸	Y 軸	Y1 軸	Y1 軸
H'02	Z 軸	Z 軸	Z1 軸	Z1 軸
H'03	A 軸	A 軸	A1 軸	A1 軸
H'04	設定禁止(無効)	B 軸	設定禁止(無効)	B1 軸
H'05	設定禁止(無効)	C 軸	設定禁止(無効)	C1 軸
H'06～H'0F	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)
H'10	設定禁止(無効)	設定禁止(無効)	X2 軸	X2 軸
H'11	設定禁止(無効)	設定禁止(無効)	Y2 軸	Y2 軸
H'12	設定禁止(無効)	設定禁止(無効)	Z2 軸	Z2 軸
H'13	設定禁止(無効)	設定禁止(無効)	A2 軸	A2 軸
H'14	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)	B2 軸
H'15	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)	C2 軸
H'16～H'1F	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)	設定禁止(無効)

■ HARD CONFIG DATA2 PORT の内容

D1-0 ：PAUSE CLR TYPE

PAUSE CLR 条件を選択します。

PAUSE CLR TYPE	PAUSE CLR 条件
H'0	PAUSE コマンド実行で PAUSE CLR
H'1	合成論理信号のアクティブエッジ検出で PAUSE CLR
H'2	合成論理信号のノットアクティブエッジ検出で PAUSE CLR
H'3	設定禁止(PAUSE コマンド実行で PAUSE CLR)

● リセット後の初期値はアンダーライン側です

D5-4 ：SIGNAL GATE TYPE

PAUSE CLR 条件を「合成論理信号のアクティブエッジ／ノットアクティブエッジ検出」とした場合は、合成論理を設定します。

なしを選択した場合は、SIGNAL1 SEL で選択した信号のエッジ検出を PAUSE SET 条件にします。

ANDまたは"OR"を選択した場合は SIGNAL1 SEL と SIGNAL2 SEL で選択した信号の指定合成論理信号のエッジ検出を PAUSE CLR 条件にします。

SIGNAL GATE TYPE	合成論理
H'0	なし(SIGNAL1 SEL 選択信号を使用)
H'1	AND
H'2	OR
H'3	設定禁止

● リセット後の初期値はアンダーライン側です

■ HARD CONFIG DATA3 PORT の内容

D3-0 ：SIGNAL1 SEL

D11-8 ：SIGNAL2 SEL

合成論理に使用する信号を選択します。

SIGNAL SEL	合成論理に使用する信号			
	C-VX870v1 C-VX870Ev1	C-VX871v1 C-VX871Ev1	C-VX872v1 *1 C-VX872Ev1	C-VX873v1 *1 C-VX873Ev1
H'00	X 軸 OUTA	X 軸 OUTA	Xn 軸 OUTA	Xn 軸 OUTA
H'01	X 軸 OUTB	X 軸 OUTB	Xn 軸 OUTB	Xn 軸 OUTB
H'02	Y 軸 OUTA	Y 軸 OUTA	Yn 軸 OUTA	Yn 軸 OUTA
H'03	Y 軸 OUTB	Y 軸 OUTB	Yn 軸 OUTB	Yn 軸 OUTB
H'04	Z 軸 OUTA	Z 軸 OUTA	Zn 軸 OUTA	Zn 軸 OUTA
H'05	Z 軸 OUTB	Z 軸 OUTB	Zn 軸 OUTB	Zn 軸 OUTB
H'06	A 軸 OUTA	A 軸 OUTA	An 軸 OUTA	An 軸 OUTA
H'07	A 軸 OUTB	A 軸 OUTB	An 軸 OUTB	An 軸 OUTB
H'08	なし(LOW レベル固定)	B 軸 OUTA	なし(LOW レベル固定)	Bn 軸 OUTA
H'09	なし(LOW レベル固定)	B 軸 OUTB	なし(LOW レベル固定)	Bn 軸 OUTB
H'0A	なし(LOW レベル固定)	C 軸 OUTA	なし(LOW レベル固定)	Cn 軸 OUTA
H'0B	なし(LOW レベル固定)	C 軸 OUTB	なし(LOW レベル固定)	Cn 軸 OUTB
H'0C	$\overline{\text{SENSOR0}}$ 入力信号	$\overline{\text{SENSOR0}}$ 入力信号	$\overline{\text{SENSORn0}}$ 入力信号	$\overline{\text{SENSORn0}}$ 入力信号
H'0D	$\overline{\text{SENSOR1}}$ 入力信号	$\overline{\text{SENSOR1}}$ 入力信号	$\overline{\text{SENSORn1}}$ 入力信号	$\overline{\text{SENSORn1}}$ 入力信号
H'0E	$\overline{\text{SIGNAL IN0}}$ 入力信号	$\overline{\text{SIGNAL IN0}}$ 入力信号	$\overline{\text{SIGNAL INn0}}$ 入力信号	$\overline{\text{SIGNAL INn0}}$ 入力信号
H'0F	$\overline{\text{SIGNAL IN1}}$ 入力信号	$\overline{\text{SIGNAL IN1}}$ 入力信号	$\overline{\text{SIGNAL INn1}}$ 入力信号	$\overline{\text{SIGNAL INn1}}$ 入力信号

● リセット後の初期値はアンダーライン側です

*1 C-VX872v1, C-VX872Ev1, C-VX873v1, C-VX873Ev1 の場合、
使用できる信号は設定の対象と同じ系列軸内の信号になります。
例) X2 軸の設定の場合、n=2

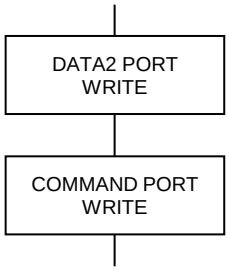
・ 対象軸の PAUSE CLR 条件と PAUSE CLR 条件が同じ場合は、PAUSE 信号は ON しません。
(PAUSE コマンドの場合は除く)

(3) PAUSE

各軸の PAUSE 信号をコマンドで操作します。
このコマンドの実行は常時可能です。

< 実行シーケンス >

COMMAND : H'10　PAUSE COMMAND



① HARD CONFIG DATA2 PORT に設定データを書き込みます。

② HARD CONFIG COMMAND PORT に COMMAND を書き込みます。

HARD CONFIG DATA2 PORT の設定データ

D15	D14	D13	D12	D11	D10	D9	D8	
–	–	C2 PAUSE	B2 PAUSE	A2 PAUSE	Z2 PAUSE	Y2 PAUSE	X2 PAUSE	C-VX872v1,C-VX872Ev1 C-VX873v1,C-VX873Ev1
		–	–	–	–	–	–	

D7	D6	D5	D4	D3	D2	D1	D0	
–	–	C1 PAUSE	B1 PAUSE	A1 PAUSE	Z1 PAUSE	Y1 PAUSE	X1 PAUSE	C-VX872v1,C-VX872Ev1 C-VX873v1,C-VX873Ev1
		C PAUSE	B PAUSE	A PAUSE	Z PAUSE	Y PAUSE	X PAUSE	C-VX870v1,C-VX870Ev1 C-VX871v1,C-VX871Ev1

C-VX870v1,C-VX870Ev1
C-VX871v1,C-VX871Ev1

C-VX872v1,C-VX872Ev1
C-VX873v1,C-VX873Ev1

D3 -D0 ：PAUSE データ
D11-D8：PAUSE データ
各軸の PAUSE 信号の状態を設定します。
0 ：PAUSE OFF
1 ：PAUSE ON

・ 当コマンドで PAUSE 信号を ON にするには、対象軸の PAUSE SET 条件が「PAUSE コマンド実行」に設定されている必要があります。
PAUSE SET 条件は PAUSE SET SPEC コマンドで設定します。

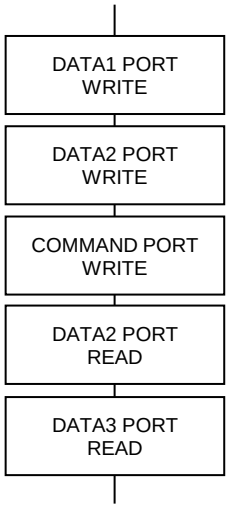
・ 当コマンドで PAUSE 信号を OFF にするには、対象軸の PAUSE CLR 条件が「PAUSE コマンド実行」に設定されている必要があります。
PAUSE CLR 条件は PAUSE CLR SPEC コマンドで設定します。

2-3-3. ♦ ↄ ㅋ ー ╗ ⇔

(1) HARD CONFIG SET DATA READ

HARD CONFIGURATION ⌐ ⇔ √ ♦ ↄ ㅋ ╩ ╗ ⇔ ╕ Δ ⁹
↓ ー ◖ ∀ fi ♪ ー │ ≠ Δ ⁹

< ◦ ↄ ◗ fi ☻ > COMMAND : H'20 HARD CONFIG SET DATA READ COMMAND



ŋ HARD CONFIG DATA1 PORT ⌐ ╗ Δ ♦ ↄ ㅋ ー
╕ √ │ ⌠ ╩ ⅝ ╗ ╕ Δ ⁹
(╗ Δ COMMAND CODE ー ー ≤ ∂ ≠ Δ)

º HARD CONFIG DATA2 PORT に読み出すデータの COMMAND CODE ╩
⅝ ╗ ╕ Δ ⁹

ɔ HARD CONFIG COMMAND PORT ⌐ COMMAND ╩ ⅝ ╗ ╕ Δ ⁹

ˆ HARD CONFIG DATA2 PORT ⅛ ╟ ⇔ √ COMMAND CODE ー
DATA2 PORT ー ♦ ↄ ㅋ ╩ ╗ ⇔ ╕ Δ ⁹

˘ HARD CONFIG DATA3 PORT ⅛ ╟ ⇔ √ COMMAND CODE ー
DATA3 PORT ー ♦ ↄ ㅋ ╩ ╗ ⇔ ╕ Δ ⁹

ה ⁱ ☿ ℙ ♩ │ ⁸ ー ♦ ↄ ㅋ ー ⅜ ╗ ↕ ╢ ╕ Δ ⁹

ה HARD CONFIG SET DATA READ ◖ ∀ fi ♪ ╩ Δ ╢ ≤ ⇔ √ ◖ ∀ fi ♪ ー ♦ ↄ ㅋ ╩ DRIVE DATA2, 3 PORT READ ⌐ ☿ ℙ ♩ ⇔ ╕ Δ ⁹
◖ ∀ fi ♪ ≠ ⅝ ╗ ⅜ ث ⌂ DATA PORT ー ♦ ↄ ㅋ │ ⁸ "0" ⌐ ⌂ ╡ ╕ Δ ⁹

ù ╗ ⇔ ≠ ⅝ ╢ ♦ ↄ ㅋ

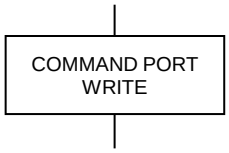
COMMAND CODE	◖ ∀ fi ♪
H'01	SENSOR SIGNAL SELECT
H'04	SIGNAL OUT SELECT
H'05	SIGNAL OUT TIMER SET
H'11	PAUSE SET SPEC
H'12	PAUSE CLR SPEC

2-3-4. ／ ー ث

(1) HARD CONFIG RESET

HARD CONFIGURATION の全てのデータを初期化して、リセット入力後と同じ状態にします。
このコマンドの実行は常時可能です。

< ◦ ↄ ◗ fi ☻ > COMMAND : H'F0 HARD CONFIG RESET COMMAND



ŋ HARD CONFIG COMMAND PORT ⌐ COMMAND ╩ ⅝ ╗ ╕ Δ ⁹

３．機能説明

3-1. ドライブ仕様

3-1-1. コマンド予約機能

各軸には、10 命令分のデータ・コマンドを格納する予約レジスタがあります。
予約レジスタには DRIVE COMMAND の汎用コマンドを予約することができます。
ドライブ起動後に次のドライブコマンドを予約することで連続したドライブを行うことができます。

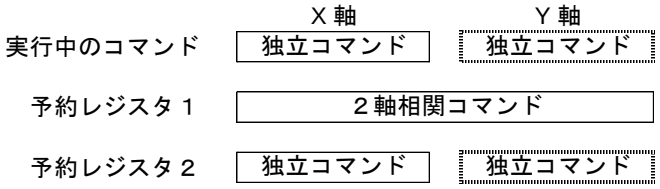
予約レジスタの状態は、STATUS1 PORT の COMREG EP と COMREG FL フラグで確認します。
BUSY = 1 で、COMREG FL = 0 のときに、DRIVE COMMAND PORT に汎用コマンドを書き込むと
DRIVE DATA1, 2 PORT のデータと汎用コマンドの 1 命令分を、予約レジスタに格納します。

予約レジスタは FIFO 構成になっています。
実行中のコマンド処理が終了すると、予約レジスタに格納したコマンドを順次実行します。

DRIVE COMMAND の特殊コマンドは予約できません。

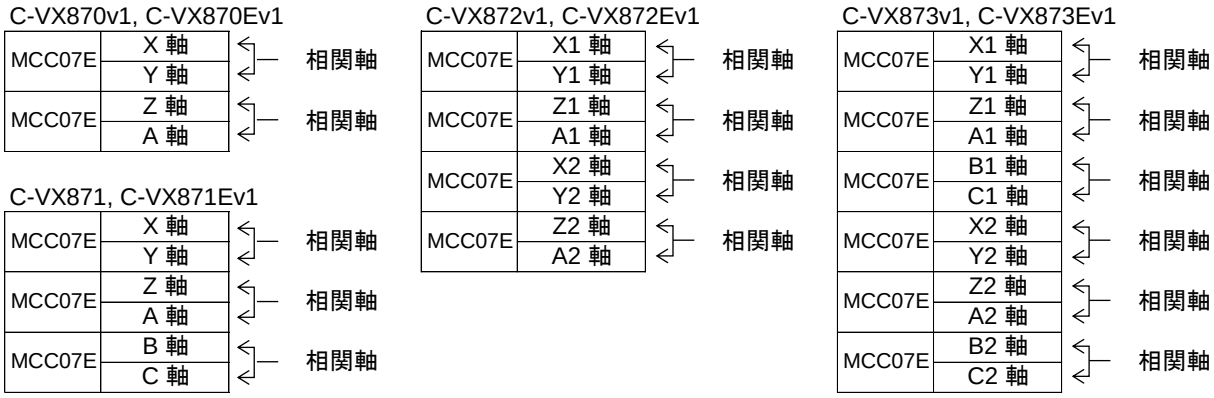
● 2 軸相関コマンドの予約
2 軸相関コマンドは、相関軸 2 軸の COMREG FL = 0 のときに予約することができます。
2 軸相関コマンドの予約は、以下のように実行します。

・ 2 軸相関コマンドの命令は、相関軸両軸の予約レジスタに格納します。
・ 片方の軸の次に実行する予約コマンドが 2 軸相関コマンドになっても、もう片方の軸が別のコマンド処理中の場合は、BUSY = 1 のまま、もう片方の軸のコマンド処理終了を待ちます。
・ 両軸の次に実行する予約コマンドが 2 軸相関コマンドになると、2 軸相関コマンドを実行します。



以下の 2 軸相関コマンド実行中は、相関軸のどちらの軸で停止指令が発生しても有効になります。
・ MAIN XY STRAIGHT CP コマンド
・ MAIN XY CIRCULAR CP コマンド

当製品での相関軸は以下の通りです。



● コマンド予約機能が自動的に無効となる状態
STATUS1 PORT の ERROR = 1 になると、予約コマンドをすべてクリアします。
また、実行待ちの予約コマンドをクリアした場合は、ERROR STATUS の COMREG CLR ERROR = 1 にします。

2 軸相関コマンドを予約している場合は、ERROR = 1 で、X, Y 軸の予約コマンドをすべてクリアします。
エラーが発生していない軸も、COMREG CLR ERROR = 1 により ERROR = 1 になります。

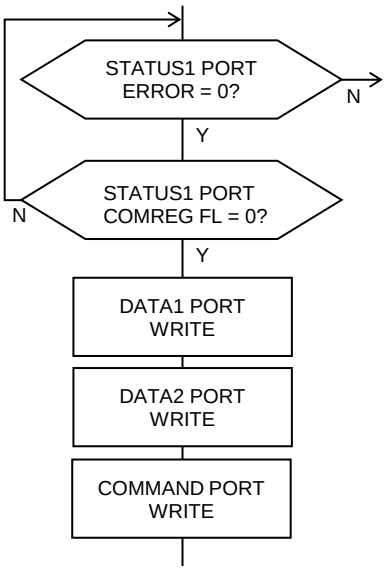
ERROR = 1 の間は、COMREG FL = 1、COMREG EP = 1 になり、予約コマンド（汎用コマンド）の
書き込みが無効になります。
ERROR = 0 にクリアすると、COMREG FL = 0 になり、予約コマンドの書き込みが有効になります。

Ẽ ◖ ∀ fi ♪ ↓ — ŭ̃

◖ ∀ fi ♪ ⇈ ↓ Δ ╟ | ⁸ BUSY = 0 — ⌂ ⇈ ╡ ⌐ COMREG FL = 0 ⇈ ⇔ ╕ Δ ⁹

↓ ◦ ⊃ ◗ fi ☻ ⌐ BUSY = 0 ⌐ ⌂ ∫ √ | ⁸ — ◖ ∀ fi ♪ ≤ ⌐ ⌂ ╡ ╕ Δ ⁹

ŭ ◖ ∀ fi ♪ ↓ ◦ ⊃ ◗ fi ☻



ŋ STATUS1 PORT — ERROR ḥ ! ⁼ ⅜ "0" ≠ №╟ ↓ ≤ ⇈ ⇔ ╕ Δ ⁹

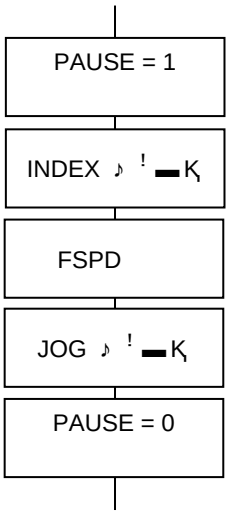
º STATUS1 PORT — COMREG FL ḥ ! ⁼ ⅜ "0" ≠ №╟ ↓ ≤ ⇈ ⇔ ╕ Δ ⁹

ɔ DRIVE DATA1 PORT ⌐ ♦ ⊃ ⅟ ⇈ ⅝ ╥ ╕ Δ ⁹

ˆ DRIVE DATA2 PORT ⌐ ♦ ⊃ ⅟ ⇈ ⅝ ╥ ╕ Δ ⁹

ˇ DRIVE COMMAND PORT ⌐ COMMAND ⇈ ⅝ ╥ ╕ Δ ⁹

ŭ INDEX ♪ ! ▬Ḳ ḱ JOG ♪ ! ▬Ḳ — ŭ̃



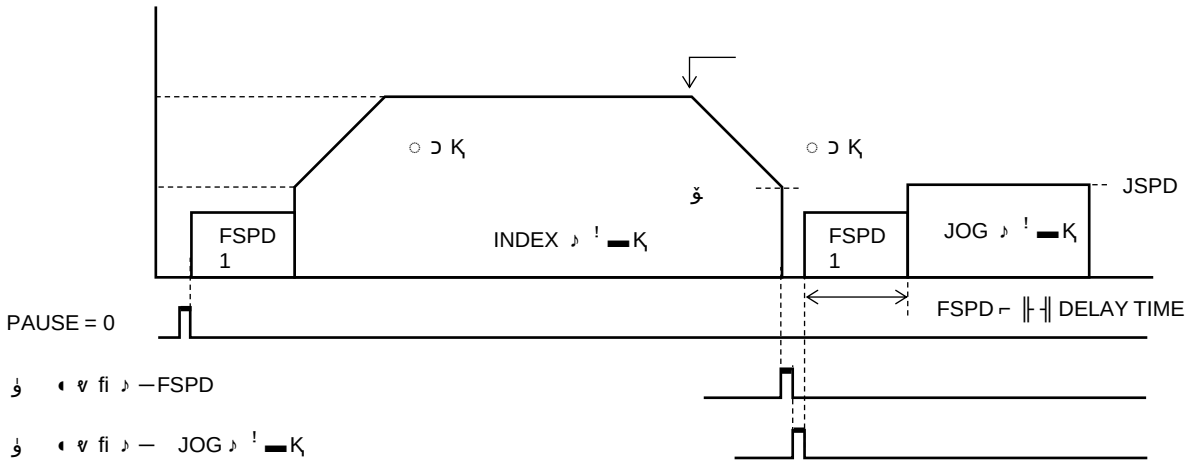
ŋ PAUSE ◖ ∀ fi ♪ (HARD CONFIG ◖ ∀ fi ♪)≠ — PAUSE ⇈ 1 ⌐ ⇔ ╕ Δ ⁹

º INDEX ♪ ! ▬Ḳ ⇈ ⇔ ╕ Δ ⁹
STBY = 1 — ╕ ╕ ♪ ! ▬Ḳ ⅜ ⌂ ⌐ ⌂ ╡ ╕ Δ ⁹

ɔ ↓ ◖ ∀ fi ♪ ≠ ⁸ FSPD ⌐ ╟╢ DELAY TIME ⇈ ⇔ ╕ Δ ⁹

ˆ ↓ ◖ ∀ fi ♪ ≠ ⁸ JOG ♪ ! ▬Ḳ ⇈ ⇔ ╕ Δ ⁹

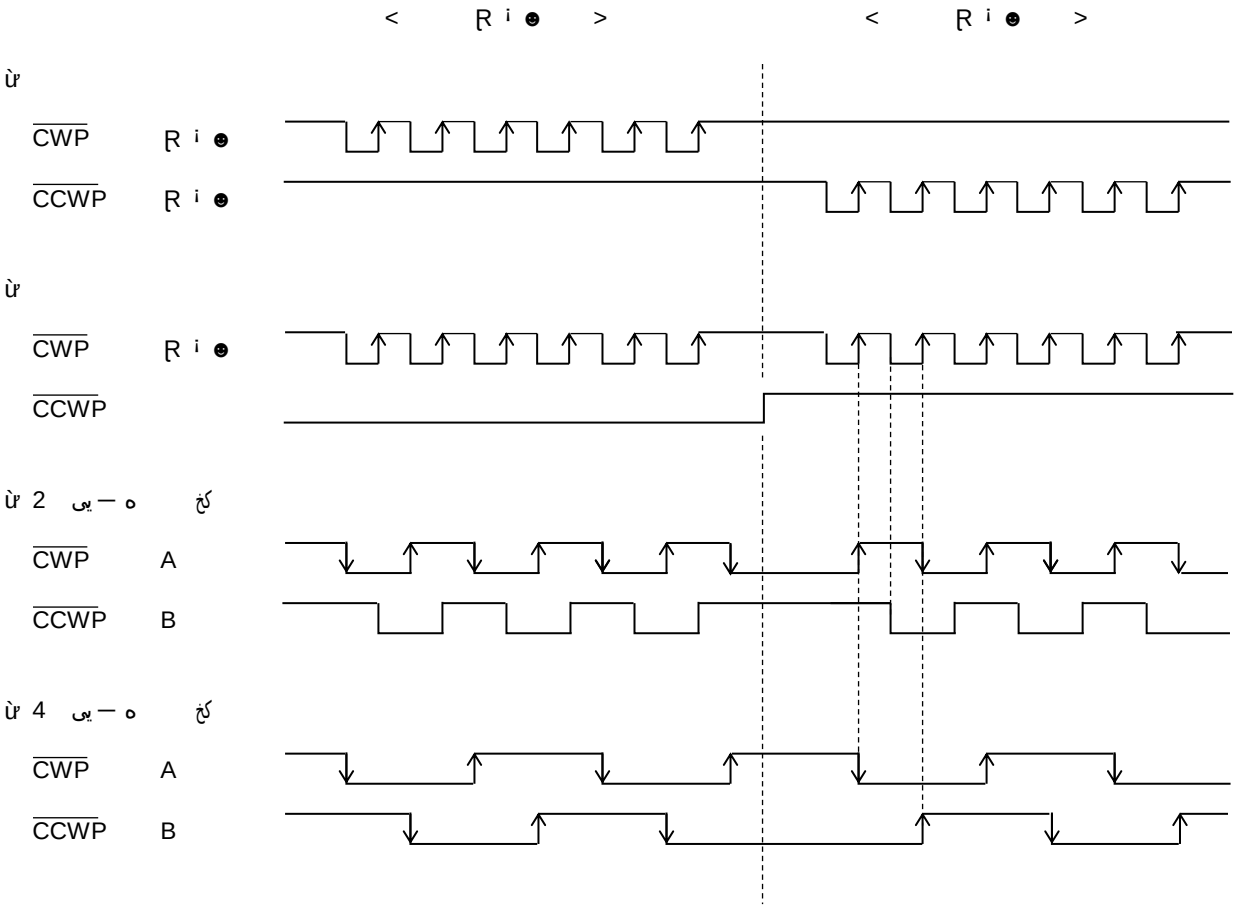
ˇ PAUSE ◖ ∀ fi ♪ (HARD CONFIG ◖ ∀ fi ♪)≠ — PAUSE ⇈ 0 ⌐ ⇔ ╕ Δ ⁹
STBY = 0 ⌐ ⌂ ╡ ⁸ INDEX ♪ ! ▬Ḳ ḱ JOG ♪ ! ▬Ḳ ⇈ ⇔ ╕ Δ ⁹



3-1-2. ط ث

(1) R i ● ث

CWP,CCWP ك ⅛ ╟ Δ ╢ R i ● ─ ╩ ث ─ 4 ─ ث ⅛ ╟ ≠ ⅝ ╕ Δ ⁹



ה │ R i ● ─ ğ ◄ ℘ ☺(▪ ♪" ● ○ ► fi ㄱ ─ ○ ► fi ♩ ◄ ℘ ☺)≠ Δ ⁹

ה ─ R i ● │ ⁸ ─ ⌐ SPEC INITIALIZE1 ◖ ℣ fi ♪ ≠ ⇔ ╕ Δ ⁹

ה ─ │ ⁸ Δ ╢ R i ● ─ ⅜ Δ ╢ ≤ ⇔ ╕ Δ ⁹
JOG, SCAN, INDEX, ORIGIN, ♪ ! ▬ Ķ ≠ │ ⁸ STBY = 1 ≠ ⅜ ⇔ ╕ Δ ⁹
ص ♪ ! ▬ Ķ ≠ │ ⁸ STBY = 1 ≠ ⇔ ⁸ R i ● ⌐ ─ R i ● ─ ⅜ ⇔ ╕ Δ ⁹
MANUAL ♪ ! ▬ Ķ ≠ │ ⁸ CWMS ╕ √ │ CCWMS ─ STBY = 1 ≠ ⅜ ⇔ ╕ Δ ⁹
R i ● ≠ │ ⁸ Δ ╢ R i ● ─ ≠ ⅜ ⇔ ╕ Δ ⁹

ה ه ك │ ⁸ ─ R i ● ğ ◄ ℘ ☺ ─ ㄱ ▬ ⊢ fi ◓ ≠ ⇔ ╕ Δ ⁹

(2) サーボ対応機能

各軸にはサーボドライバに対応する信号として以下の信号があります。

・ DRST 信号出力　　(サーボリセット出力)
・ DEND/PO 信号入力　　(サーボ位置決め完了入力／ PO 入力)
・ INn0--INn3 信号入力　　(ドライバアラーム入力など)
・ OUTn0--OUTn3 信号出力　　(サーボON など)

※ 6軸,12軸の製品は、INnx 信号入力、OUTnx 信号出力機能はありません。
汎用入力(停止機能設定可能)信号として、SENSORn0,SENSORn1 信号を使うことができます。
汎用出力信号として、DRST 信号を使用することができます。

■ DRST 信号

サーボ対応無効時は、汎用出力としてステッピングモータドライバのM.F 信号(モータ励磁電流のON/OFF)などに使用できます。
サーボ対応有効時は、ドライブ中に即時停止指令、またはLIMIT 即時停止指令を検出すると、DRST 信号が10 ms 間アクティブレベルを出力します。
DRST/MF 信号をサーボモータドライバに接続しない場合、DRST TYPE をサーボ対応には設定しないでください。

・ DRST 信号がサーボ対応でアクティブレベルを出力中はDRIVE STATUS1 PORT のBUSY=1 となります。
DRST 信号およびDEND 信号の<サーボ対応>終了後に、ドライブを終了します。
・ DRST 信号はサーボ対応の有効/無効に関わらずSIGNAL OUT コマンドでON/OFF レベルを出力することができます。
・ DRST 信号のサーボ対応はSPEC INITIALIZE3 コマンドで設定します。

■ DEND/PO 信号

サーボ対応無効時は、ステッピングモータドライバのPO 信号入力、または汎用入力として使用できます。
サーボ対応有効時は、ドライブ実行時にパルス出力が終了しても、DEND/PO 信号のアクティブレベルを検出するまでドライブを終了しません。初期設定はサーボ対応無効です。
DEND 信号の状態は、MCC07E のDRIVE STATUS2 PORT から確認することができます。

・ DEND/PO 信号がサーボ対応でアクティブレベルの検出待ちの間は、DRIVE STATUS1 PORT のBUSY = 1 、DRIVE STATUS2 PORT のDEND BUSY ＝ 1 になります。
・ 即時停止指令を検出した場合は、サーボ対応を中止してドライブを終了します。
即時停止指令の検出で、BUSY = 0 、DEND BUSY = 0 になります。
・ DEND/PO 信号のサーボ対応はSPEC INITIALIZE3 コマンドで設定します。

■ INnx 信号

汎用入力INnx 信号の入力状態は、汎用入力PORT から確認することができます。
また、MCC07E の入力機能を選択すると、ドライバからのアラーム信号に使用することができます。
DALM 信号としては、減速停止、または即時停止させることができます。初期設定は汎用入力です。
INnx 信号によるDALM 状態は、MCC07E のDRIVE STATUS2 PORT から確認することができます。
・ INnx 信号によるMCC07E のDALM 入力機能はSPEC INITIALIZE3 コマンドで設定します。
・ MCC07E へのDALM 信号入力は、汎用入力信号INn0--INn3 によって機能します。

汎用入力信号	C-VX870v1 C-VX870Ev1
IN0	X 軸
IN1	Y 軸
IN2	Z 軸
IN3	A 軸

汎用入力信号	C-VX872v1 C-VX872Ev1
IN10	X1 軸
IN11	Y1 軸
IN12	Z1 軸
IN13	A1 軸
IN20	X2 軸
IN21	Y2 軸
IN22	Z2 軸
IN23	A2 軸

■ OUTnx 信号

汎用出力として使用することができます。
サーボオン信号やステッピングモータドライバの分解能切替信号などに使用できます。
・ 汎用出力OUTnx 信号の状態は、汎用I/O 関数で確認することができます。

(3) [illegible]

[illegible]

[illegible]

[illegible]		[illegible]
CWLM, CCWLM [illegible] DEND/PO, DALM(INnx) [illegible]	[illegible] 0	10ms	HARD INITIALIZE4 [illegible]
ORG, NORG [illegible]	[illegible] 0	10ms	[illegible] HARD INITIALIZE5 [illegible]
ÑZORG [illegible]	[illegible] 0	1ms	[illegible] HARD INITIALIZE5 [illegible]
ÑEA, ÑEB [illegible]	[illegible] 0	12.75εs	[illegible] HARD INITIALIZE6 [illegible]

[illegible]

[illegible]

[illegible]

(4) [illegible]

[illegible]

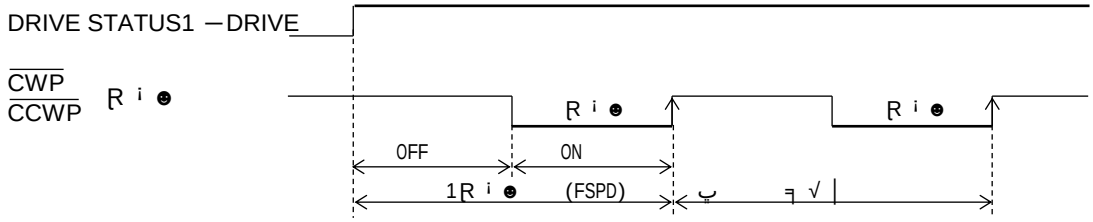
[illegible]	[illegible]	[illegible]
FSSTOP [illegible]	[illegible]	HARD INITIALIZE7 [illegible]
CWLM [illegible]	[illegible]	
CCWLM [illegible]	[illegible]	
DALM(INnx) [illegible]	[illegible]	
ORG [illegible]	[illegible]	
NORG [illegible]	[illegible]	

[illegible]

3-1-3. ♪ ! ▬Ķ Ꞁ ! ϖ ↄ ⅍
(1) 1 Ꞁ ⁱ ●

♪ ! ▬Ķ — 1 Ꞁ ⁱ ☻ | FSPD ≠ ⇔ √ Ꞁ ⁱ ☻ ╩ ⇔ ╕ Δ ⁹
◖ ∀ fi ♪ ↓ ≤ 1 Ꞁ ⁱ ☻ ╩ ╖ ╥ ∑ ╢ ↓ ≤ ≠ ⁸ ⇔ √ ♪ ! ▬Ķ ╩ ⅃ ╢ ↓ ≤ ⅜ ≠ ⅝ ╕ Δ ⁹

FSPD ה ה ה ♪ ! ▬Ķ — 1 Ꞁ ⁱ ☻ — ╩ 1Hz ₀ ≠ ⇔ ╕ Δ ⁹
(0 8,388,607Hz)



ה FSPD — ثم ≤ ⌐ Δ ╢ | ⅌ — ╡ ≤ ⌂ ╡ ╕ Δ ⁹

ثم		⌐ Δ ╢ ( )			
8,388,607	6,666,667 Hz	OFF	= 50ns	ON	= 50ns (10,000,000 Hz)
6,666,666	5,000,001 Hz	OFF	= 50ns	ON	= 100ns (6,666,666 Hz)
5,000,000	4,000,001 Hz	OFF	= 100ns	ON	= 100ns (5,000,000 Hz)
4,000,000	3,333,334 Hz	OFF	= 100ns	ON	= 150ns (4,000,000 Hz)
3,333,333	2,857,143 Hz	OFF	= 150ns	ON	= 150ns (3,333,333 Hz)

Ẽ **FSPD ⌐ ╟ ╢ DELAY TIME —** طص

FSPD — Ꞁ ⁱ ☻ | ⁸ ♪ ! ▬Ķ — ⌐ ∏ ⇔ ╕ Δ ⁹
◖ ∀ fi ♪ ↓ ( )≠ ♪ ! ▬Ķ ╩ e ⌐ | ⁸ — ♪ ! ▬Ķ — FSPD — ╩
Δ ╢ ↓ ≤ ⌐ ╟ ╡ ⁸ FSPD ╩ ♪ ! ▬Ķ — DELAY TIME ≤ ⇔ ≡ ≠ ⅝ ╕ Δ ⁹

ϋ FSPD ≠ تج ⇔ ⌂ ™ ♪ ! ▬Ķ ╩ e
— ♪ ! ▬Ķ ḱ — ♪ ! ▬Ķ ╩ ⁸ — Ꞁ ⁱ ☻ ≠ ≈ ⌂ → ╕ Δ ⁹
ה — ♪ ! ▬Ķ ⌐ ⁸ ◖ ∀ fi ♪ ≠ ₈ — ♪ ! ▬Ķ ₉ ╩ ⇔ ╕ Δ ⁹
₈ — ♪ ! ▬Ķ ₉ — FSPD ╩ ⁸ ₈ — ♪ ! ▬Ķ ₉ — ⌐ ⇔ ╕ Δ ⁹

ה MCC07E | ⁸ — ♪ ! ▬Ķ ⌐ ◖ ∀ fi ♪ — ╩ ™ ╕ Δ ⁹
₈ — ♪ ! ▬Ķ ₉ — Ꞁ ⁱ ☻ FSPD ⌐ ⁸ — ♪ ! ▬Ķ ₉ — ╩ ⇔ ╕ Δ ⁹
Ꞁ ⁱ ☻ | ⁸ ₈ — ♪ ! ▬Ķ ₉ — ⅛ ╟ Ꞁ ⁱ ☻ ⇔ ╕ Δ ⁹

ϋ FSPD ≠ ♪ ! ▬Ķ — تج ╩ طص Δ ╢
— ♪ ! ▬Ķ ḱ — ♪ ! ▬Ķ ⌐ ⁸ DELAY TIME ئو 50ms 20Hz ╩ طص ⇔ ╕ Δ ⁹
ה — ♪ ! ▬Ķ ⌐ ⁸ ◖ ∀ fi ♪ ≠ ₈ — ♪ ! ▬Ķ ₉ ╩ ⇔ ╕ Δ ⁹
₈ — ♪ ! ▬Ķ ₉ — FSPD ╩ ⁸ 20Hz ⌐ ⇔ ╕ Δ ⁹

ה MCC07E | ⁸ — ♪ ! ▬Ķ ⌐ ◖ ∀ fi ♪ — ╩ ™ ╕ Δ ⁹
₈ — ♪ ! ▬Ķ ₉ — Ꞁ ⁱ ☻ FSPD ⌐ 20 Hz ╩ ⇔ ╕ Δ ⁹
Ꞁ ⁱ ☻ | ⁸ ₈ — ♪ ! ▬Ķ ₉ — ⅛ ╟ Ꞁ ⁱ ☻ ⇔ ╕ Δ ⁹

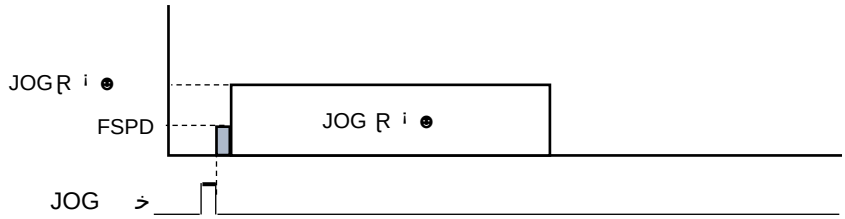
DELAY TIME — طص ≤ ⇔ ≡ | ⁸ SPEC INITIALIZE1 ◖ ∀ fi ♪ — PULSE OUTPUT MASK — ╩ ئو ⇔ ≡ ⁸
₈ Ꞁ ⁱ ☻ ╩ ∀ ☻ ⇔ √ ♪ ! ▬Ķ — ₉ ╩ DELAY TIME ≤ ⇔ ≡ Δ ╢ ↓ ≤ ╚ ≠ ⅝ ╕ Δ ⁹

(2) JOG Ɽ ¡ ɷ ɔ ㅋ

Ẽ **JOG Ɽ ⁱ ●**

JOG ♪ ! ▬ Ḳ ╩ Δ ╢ ≤ ⇔ √ JOG Ɽ ⁱ ☻ ─ ᵖ ≠ ♪ ! ▬ Ḳ ╩ ™ ╕ Δ ⁹

JOG Ɽ ⁱ ☻ ─ │ 1 4,194,303Hz(1Hz ₀)≠ Δ ⁹



ה JOG Ɽ ⁱ ☻ │ JSPD SET ◖ ∀ fi ♪ ≠ ⇔ ╕ Δ ⁹

Ẽ **JOG Ɽ ⁱ ●**

JOG ♪ ! ▬ Ḳ ╩ Δ ╢ ≤ ⇔ √ JOG Ɽ ⁱ ☻ ─ Ɽ ⁱ ☻ ╩ ⇔ ╕ Δ ⁹

JOG Ɽ ⁱ ☻ │ ⁸ 0 65,535 H'0000 H'FFFF ≠ Δ ⁹

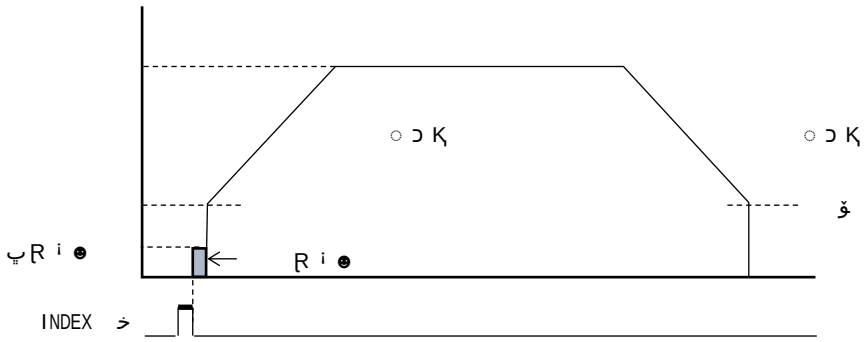
ה JOG ♪ ! ▬ Ḳ Ɽ ⁱ ☻ │ JOG PULSE SET ◖ ∀ fi ♪ ≠ ⇔ ╕ Δ ⁹

(3) R ! ω ɔ ㄹ

Ē 8 8 ǧ

♪ ! ▬Ķ — 8 8 ǧ ╩ ≠ ⅝ ┐ Δ 9

│ 0.1 6,553,400 Hz ≠ Δ 9



│ خ ﾞ — ⌐ ⇔ ┐ Δ 9

(Hz) — R i ☻ ♦ ɔ ㄹ (HSPD) × ی (RESOL)

(Hz) — R i ☻ ♦ ɔ ㄹ (LSPD) × ی (RESOL)

ǧ (Hz) ǧ — R i ☻ ♦ ɔ ㄹ (ELSPD) × ی (RESOL)

ی (RESOL) │ RESOL No. ╩ ≡ ⇔ ⇔ ┐ Δ 9

RESOL No.	ی (RESOL)	RESOL No.	ی (RESOL)	RESOL No.	ی (RESOL)	RESOL No.	ی (RESOL)
H'0	0.1	H'4	2	H'8	50	H'C	200
H'1	0.2	H'5	5	H'9	100	H'D	200
H'2	0.5	H'6	10	H'A	200	H'E	200
H'3	1	H'7	20	H'B	200	H'F	200

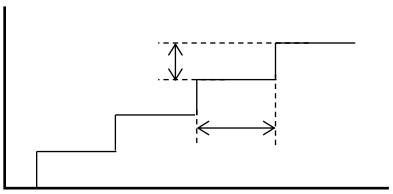
ה RESOL ≤ HSPD │ HIGH SPEED SET ◗ ∀ fi ♪ ≠ ⇔ ┐ Δ 9

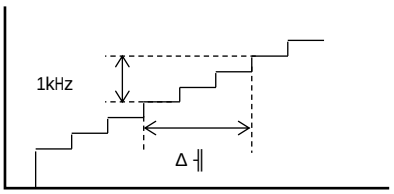
ה LSPD ≤ ELSPD │ LOW SPEED SET ◗ ∀ fi ♪ ≠ ⇔ ┐ Δ 9

Ẽ

│ ⁸ ♪ ! ▬Ķ ╩ 1kHz ↕ ∑ ╢ ─ ⌐ Δ ╢ ms/kHz ≠ Δ ⁹
≠ │ ↓ ─ ╩ RATE ≤ ⇔ ≡ ™ ╕ Δ ⁹
⅔ ╟ ┘ │ ╩ ⌐ ⅔ ╟ ┘ Δ ╢ ↓ ≤ ≠ ™ ╕ Δ ⁹

1☻ ♥ ᵖ ķ ─





RATE │ ⁿ ─ ⌐ ⇔ ╕ Δ ⁹

$$\text{RATE(ms/kHz)} = \frac{\text{○ ↄ Ķ ─ (ms)}}{\text{○ ↄ Ķ ─ (kHz)}} = \frac{\text{UCYCLE}}{\text{RESOL Ĭ 2}}$$

ה ○ ↄ Ķ ─ (ms) UCYCLE Ĭ 0.0005
ה ○ ↄ Ķ ─ (kHz) RESOL Ĭ 0.001

$$\text{RATE(ms/kHz)} = \frac{\text{○ ↄ Ķ ─ (ms)}}{\text{○ ↄ Ķ ─ (kHz)}} = \frac{\text{DCYCLE}}{\text{RESOL Ĭ 2}}$$

ה ○ ↄ Ķ ─ (ms) DCYCLE Ĭ 0.0005
ה ○ ↄ Ķ ─ (kHz) RESOL Ĭ 0.001

ה UCYCLE ≤ DCYCLE │ RATE SET ◖ ∀ fi ♪ ≠ ⇔ ╕ Δ ⁹
ה RESOL ≠ ⅜ ⇔ ╕ Δ ⁹ RESOL ╩ ↕ ↄ Δ ╢ ≤ ⅜ ╟ ⅛ ⌐ ⌂ ╡ ╕ Δ ⁹

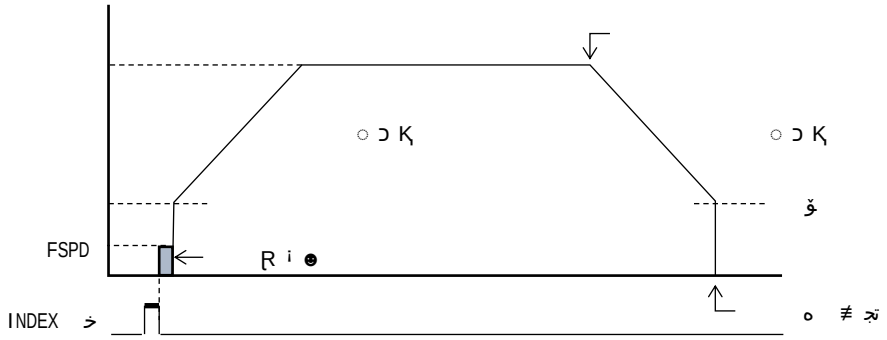
Ẽ **RATE DATA TABLE( )**

RATE (ms/kHz)	RESOL = 1 U/D CYCLE	RESOL = 5 U/D CYCLE	RESOL = 10 U/D CYCLE	RESOL = 20 U/D CYCLE	RESOL = 50 U/D CYCLE	RESOL = 200 U/D CYCLE
5,000	10,000					
3,000	6,000					
2,000	4,000					
1,000	2,000	10,000				
500	1,000	5,000	10,000			
300	600	3,000	6,000	12,000		
200	400	2,000	4,000	8,000		
100	200	1,000	2,000	4,000	10,000	
50	100	500	1,000	2,000	5,000	
30	60	300	600	1,200	3,000	12,000
20	40	200	400	800	2,000	8,000
10	20	100	200	400	1,000	4,000
5	10	50	100	200	500	2,000
3	6	30	60	120	300	1,200
2	4	20	40	80	200	800
1	2	10	20	40	100	400
0.5	1	5	10	20	50	200
0.3		3	6	12	30	120
0.2		2	4	8	20	80
0.1		1	2	4	10	40
0.05			1	2	5	20
0.03					3	12
0.02					2	8
0.01					1	4
0.005						2
0.0025						1

(4) ♪ ! ▬Ķ

♪ ! ▬Ķ | ⁸ S — ╩ "0" ⌐ ⇔ √ ○ ɔ Ķ ≤ ⁸ S — ╩ "0" ⌐ ⇔ √ ○ ɔ Ķ≠ ╩ e ♪ ! ▬Ķ≠ Δ ⁹

⅛ ╠ ╕ ≠ ⁸ S ⅜ ⌂ ™ ○ ɔ Ķ≠ ⇔ ⁸ ⅛ ╠ ǧ ╕ ≠ ⁸ S ⅜ ⌂ ™ ○ ɔ Ķ≠ ⇔ ╕ Δ ⁹



ה ⁸ ○ ɔ Ķ ≤ Δ ╢ ≤ ⅝ | ⁸ SCAREA SET ◖ ∀ fi ♪ ≠ S (SUAREA SDAREA)
╩ "0" ⌐ ⇔ ╕ Δ ⁹

ǔ ♪ ! ▬Ķ — ≤

○ ɔ Ķ — (ms)
(UCYCLE x 0.0005) ǐ (HSPD LSPD+1) Ꞁ ⁱ ☻ — (ms) TU
0 ' TU —

○ ɔ Ķ — (ms)
(DCYCLE x 0.0005) ǐ (HSPD ELSPD+1) TD
0 ' TD ǧ —

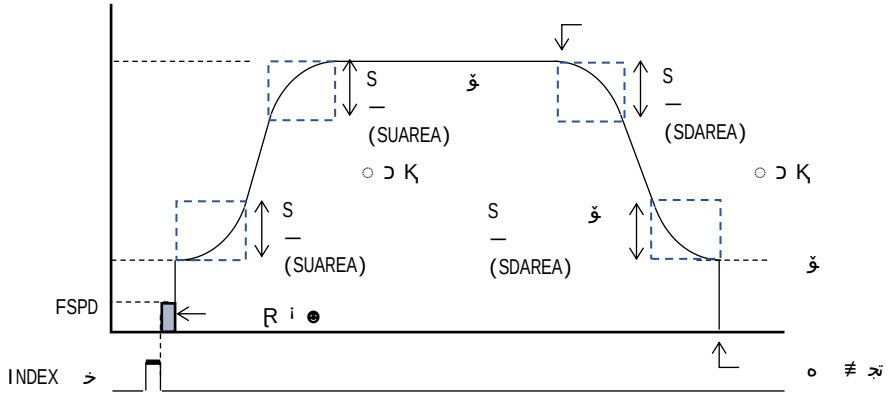
ה ɨ ɨ ≠ ɨ ≠ Δ ╢ — ≠ Δ ⁹
ה INDEX ♪ ! ▬Ķ — ɨ ⌐ | ⁸ ○ ɦ ☿ ℮ ♩ Ꞁ ⁱ ☻ — ⅜№ ╡ ╕ Δ ⁹

(5) S ♪ ! ▬Ķ

S ♪ ! ▬Ķ | ⁸ S — ⇊ ⇔ √ ○ ɔ Ķ ≤ S — ⇊ ⇔ √
○ ɔ Ķ≠ ⇊ e ♪ ! ▬Ķ≠ Δ ⁹
— S ≤ ǧ — S ⇊ ⁸ ⌐ ⌋ ⇔ √ S ○ ɔ Ķ≠ ⇔ ⁸
— S ≤ ǧ — S ⇊ ⁸ ⌐ ⌋ ⇔ √ S ○ ɔ Ķ≠ ⇔ ╕ Δ ⁹

ǜ S ○ ɔ Ķ
SCAREA SET ◖ ∀ fi ♪ — SUAREA ≠ S — ⇊ ⇔ ╕ Δ ⁹
SUAREA ≠ ⇔ √ ⅜ ⁸ — S ≤ ǧ — S ⌐ ⌂ ╡ ⁸
S ○ ɔ Ķ ⇊ ⇔ ╕ Δ ⁹ ╡ — | ⁸ UCYCLE — ○ ɔ Ķ≠ ⇔ ╕ Δ ⁹

ǜ S ○ ɔ Ķ
SCAREA SET ◖ ∀ fi ♪ — SDAREA ≠ S — ⇊ ⇔ ╕ Δ ⁹
SDAREA ≠ ⇔ √ ⅜ ⁸ — S ≤ ǧ — S ⌐ ⌂ ╡ ⁸
S ○ ɔ Ķ ⇊ ⇔ ╕ Δ ⁹ ╡ — | ⁸ DCYCLE — ○ ɔ Ķ≠ ⇔ ╕ Δ ⁹



ה SUAREA, SDAREA | SCAREA SET ◖ ∀ fi ♪ ≠ ⇔ ╕ Δ ⁹

ǜ S ♪ ! ▬Ķ — ≤

S ○ ɔ Ķ — (ms)
(UCYCLE x 0.0005) ǐ (HSPD LSPD 1 SUAREA x 2) Ꞧ ⁱ ☻ — (ms) TU
0 ' TU —

ה SUAREA (HSPD LSPD) 2 ≠ ⁸ Δ ╢ — ≠ Δ ⁹

S ○ ɔ Ķ — (ms)
(DCYCLE x 0.0005) ǐ (HSPD ELSPD 1 SDAREA x 2) TD
0 ' TD ǧ —

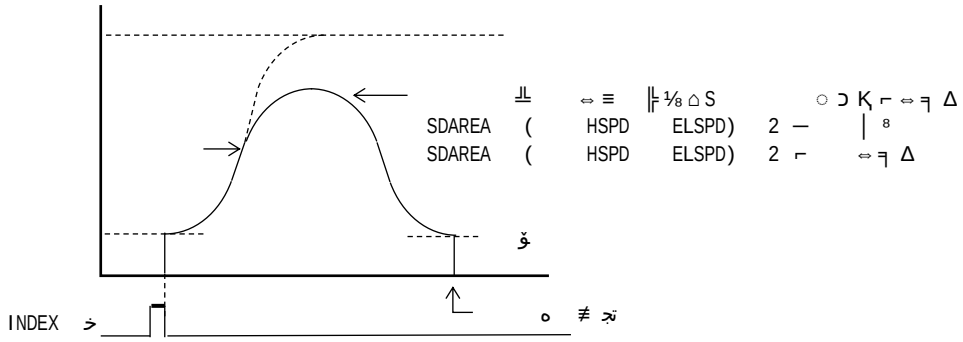
ה SDAREA (HSPD ELSPD) 2 ≠ ⁸ ⇄ ɔ ≠ ⇄ Δ ╢ — ≠ Δ ⁹
ה INDEX ♪ ! ▬Ķ — ⇄ ⌐ | ⁸ ○ ɦ ⁿ ⁿ ⁿ Ꞧ ⁱ ☻ — ⅜№╡ ╕ Δ ⁹

Ẽ **S** **INDEX** ♪ ! ▬Ķ ─ ت لأ

S ─ INDEX ♪ ! ▬Ķ ≠ ⁸ تج ه ┐ ≠ ─ Ꞧ ⁱ ☻ ⅜ ⌂ ↄ ≡ ⌐ ⇔ ⌂ ™

│ ⁸ ⌐ ╩ ⅝ ⁞ → ≡ ⁸ ╟ ⅛ ⌂ S ○ ↄ Ķ ≠ INDEX ♪ ! ▬Ķ ╩ تج ⇔ ┐ Δ ⁹

↓ ─ │ ≠ Δ ⁹

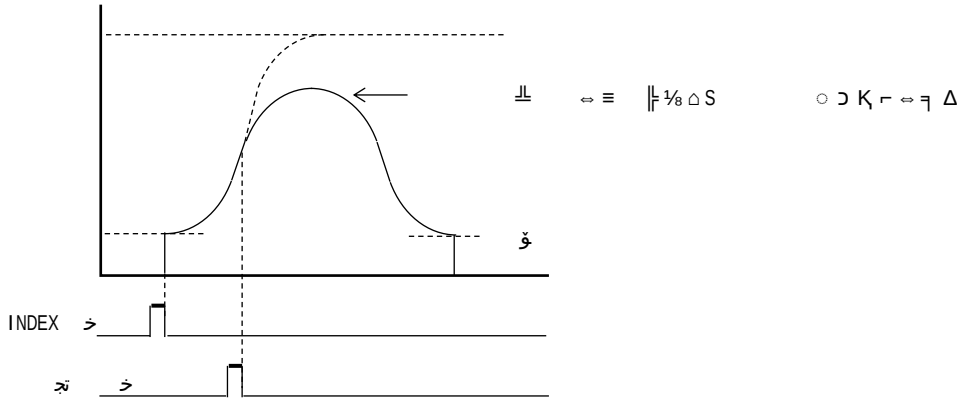


Ẽ تج خ ت لأ

S چ ⌐ تج خ ╩ ⇔ √ │ ⁸ SUAREA ─ S ǧ ○ ↄ Ķ ≠ ╟ ⅛ ⌐ ╩ ǧ ⇔ ⁸

S ○ ↄ Ķ ≠ تج ⇔ ┐ Δ ⁹

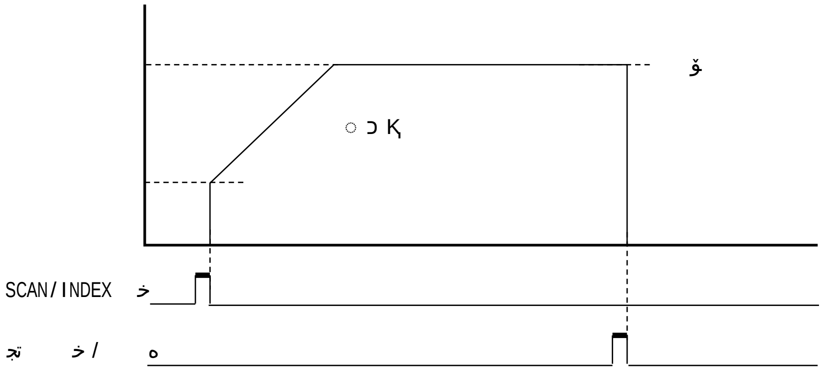
↓ ─ │ ≠ Δ ⁹



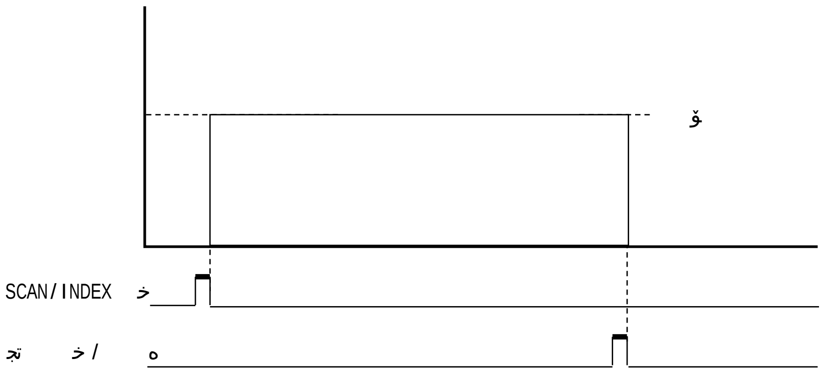
(6) ⁄ ─ٹ─ ♪ ╵ ▬Ķ

خ ﺓ ─ ╟ e ⌂ ♪ ╵ ▬Ķ ⁸ پ ♪ ╵ ▬Ķ ⁸ ♪ ╵ ▬Ķ ╨ ⁸ ◖ ∜ ﬁ ♪ ﺆ ╨ ™ ≡ ⅞ ╦ ∑ ⁸
⇔ √ ♪ ╵ ▬Ķ Ŗ ♃ ↄ ﬁ ╨ ﻷ ╣ ↓ ≤ ⅜≢ ⅝╕ ∆ ⁹

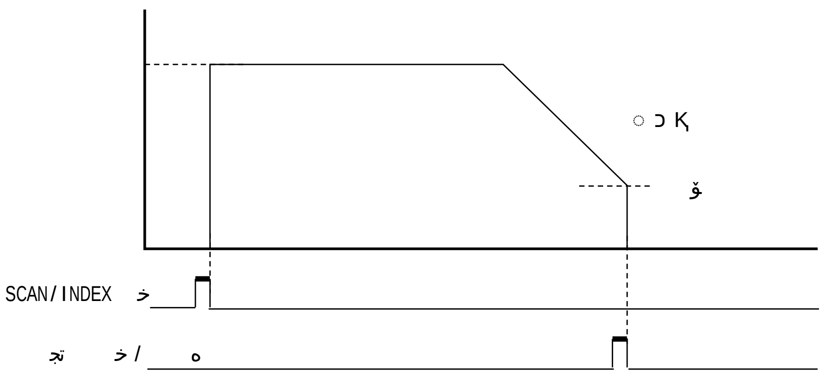
Ẽ ♪ ╵ ▬Ķ
⁸ ⁹ ⅔ ╟ ┘ ⁸ ﯙ ⁹ ⌐ ∆ ╢ ≤ ⁸ ≤ ⌐ ╟ ╢
♪ ╵ ▬Ķ ╨ ™╕ ∆ ⁹



Ẽ پ ♪ ╵ ▬Ķ
⁸ ﯙ ⁹ ╕ √ ⎮ ⁸ ﯙ ⁹ ⌐ ∆ ╢ ≤ ⁸ ≢ ─
پ ♪ ╵ ▬Ķ ╨ ™╕ ∆ ⁹



Ẽ ♪ ╵ ▬Ķ
⁸ ⁹ ⅔ ╟ ┘ ⁸ ﯙ ⁹ ⌐ ∆ ╢ ≤ ⁸ ≤ ﯙ ⌐ ╟ ╢
♪ ╵ ▬Ķ ╨ ™╕ ∆ ⁹



Ē ⁄ —ث— ♪ ! ▬Ķ

╕ √ ⁸ ⁸ ⁸ ğ ╩ ╖ ╥ ∑ ╢ ≤ ⁸ ₁ ⌂ ♪ ! ▬Ķ Ř ⁻ ↄ fi ⅜ ≠ ⅝ ╕ Δ ⁹

ǔ ⁸ ğ ⁹ ⌐ Δ ╢ ≤ ⁸ ž — ♪ ! ▬Ķ ╩ ™ ╕ Δ ⁹

ǔ ⁸ ğ ⁹ ⌐ Δ ╢ ≤ ⁸ ž — ♪ ! ▬Ķ ╩ ™ ╕ Δ ⁹

ǔ ⁸ ⅔ ╢ ┘ ⁸ ğ ⁹ ⌐ Δ ╢ ≤ ⁸ ž — ♪ ! ▬Ķ ╩
™ ╕ Δ ⁹

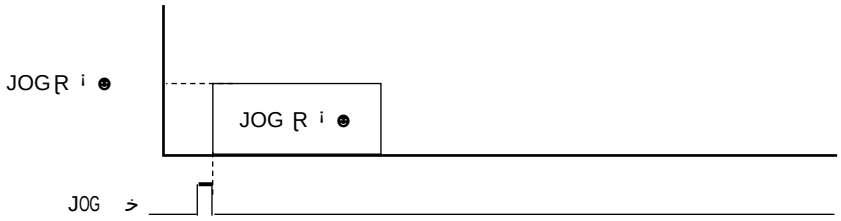
3-1-4. ♪ ! ▬Ķ

(1) JOG ♪ ! ▬Ķ

+/- JOG ◖ ∀ fi ♪ ⊥⊥ Δ ╢ ≤ ⁸ JOG Ꞁ ⁱ ☻ — ᵖ ≠ JOG Ꞁ ⁱ ☻ — Ꞁ ⁱ ☻ ⊥⊥ ⇔ ╕ Δ ⁹

تج خ ⊥⊥ Δ ╢ ≤ ⁸ Ꞁ ⁱ ☻ ⊥⊥ تج ⇔ ≡ ♪ ! ▬Ķ ⊥⊥ ǧ ⇔ ╕ Δ ⁹

تج خ ⊥⊥ Δ ╢ ≤ ⁸ Ꞁ ⁱ ☻ ⊥⊥ تج ⇔ ≡ ♪ ! ▬Ķ ⊥⊥ ǧ ⇔ ╕ Δ ⁹



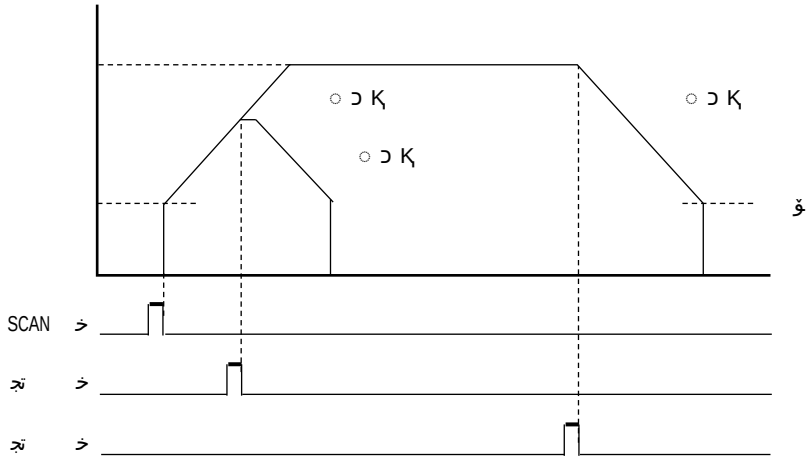
(2) SCAN ♪ ! ▬Ķ

+/- SCAN ◖ ∀ fi ♪ ⊥⊥ Δ ╢ ≤ ⁸ تج خ ⊥⊥ Δ ╢ ╕ ≠ ⇔ ≡ Ꞁ ⁱ ☻ ⊥⊥ ⇔ ╕ Δ ⁹

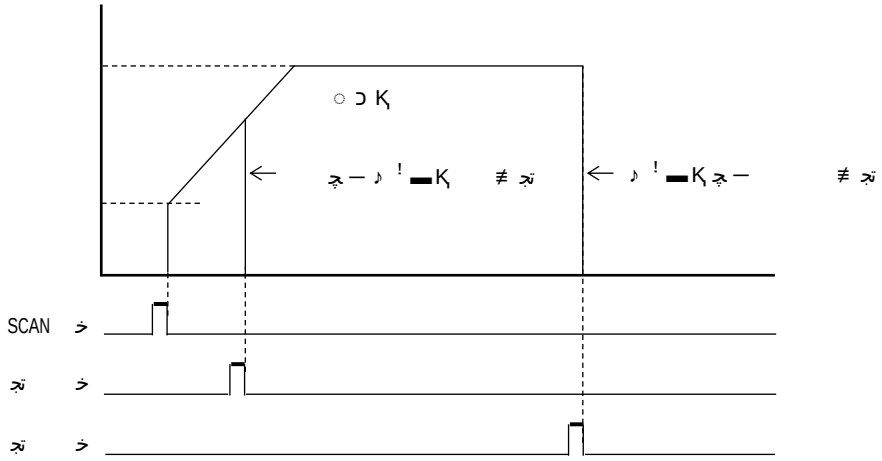
تج خ ⊥⊥ Δ ╢ ≤ ⁸ Ꞁ ⁱ ☻ ⊥⊥ تج ⇔ ≡ ♪ ! ▬Ķ ⊥⊥ ǧ ⇔ ╕ Δ ⁹

تج خ ⊥⊥ Δ ╢ ≤ ⁸ Ꞁ ⁱ ☻ ⊥⊥ تج ⇔ ≡ ♪ ! ▬Ķ ⊥⊥ ǧ ⇔ ╕ Δ ⁹

Ùª تج ╢ ⊢ خ تج لأ



Ùª تج ╢ ⊢ خ تج لأ



(3) INDEX ♪ ! ▬Ķ

INC INDEX ◖ ∀ fi ♪ ╩ Δ ╢ ≤ 8 ⇔ √ ▪ ♪ ˝ ☻ ⌐ Δ ╢ ╕ ≠ Ꞁ i ☻ ╩ ⇔ ╕ Δ 9

ABS INDEX ◖ ∀ fi ♪ ╩ Δ ╢ ≤ 8 ⇔ √ ▪ ♪ ˝ ☻ ⌐ Δ ╢ ╕ ≠ Ꞁ i ☻ ╩ ⇔ ╕ Δ 9

♪ ! ▬Ķ ج ⌐ | 8 Ꞁ i ☻ ╩ ⇔ ≡ ٥ ≠ تج ⇔ ╕ Δ 9

تج خ ╩ Δ ╢ ≤ 8 Ꞁ i ☻ ╩ تج ⇔ ≡ ♪ ! ▬Ķ ╩ ğ ⇔ ╕ Δ 9

تج خ ╩ Δ ╢ ≤ 8 Ꞁ i ☻ ╩ تج ⇔ ≡ ♪ ! ▬Ķ ╩ ğ ⇔ ╕ Δ 9

ÙÞ ⌐ ╢ ╟ تج لأ

ה ⅜ ğ ج ̈ ̄ | 8 تج خ ╩ Δ ╢ ≤ ğ ⌐ ⅛ ∫ ≡ ⇔ ╕ Δ 9

╩ Δ ╢ ≤ ğ ⌐ ⅛ ∫ ≡ ⇔ 8 ٥ ≠ Ꞁ i ☻ ╩ تج ⇔ ╕ Δ 9

3-1-5. ORIGIN ♪ ! ▬Ķ

(1) ORIGIN ♪ ! ▬Ķ

♪ ! ▬Ķ ╩ ⇔ ≡ ORIGIN ♪ ! ▬Ķ ◖ ∀ fi ♪ ╩ Δ ╢ ≤ ⁸ ORG ☵ — ◄ ℘ ☺ ╩ ⇔ ≡
♪ ! ▬Ķ ╩ ǧ ⇔ ╕ Δ ⁹
Δ ╢ ORG ☵ | ⁸ $\overline{ORG}$ ☵ , ÑZORG ☵ , $\overline{DEND}$/$\overline{PO}$ ☵ , $\overline{NORG}$ ☵ — ☵ ⅛ ╣ ⇔ ╕ Δ ⁹

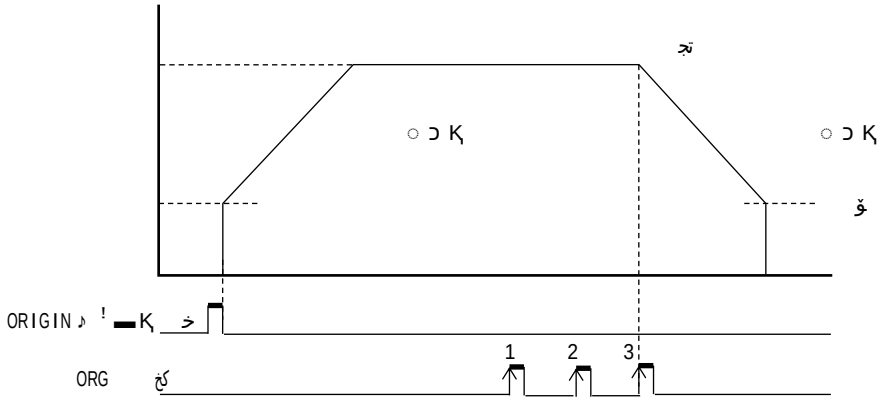
ORIGIN ♪ ! ▬Ķ ⌐ | ⁸ ☵ — ♪ ! ▬Ķ Ɽ ! ω ⊃ ⁼ — ⅜ ≠ Δ ⁹
ה ♪ ! ▬Ķ — Ɽ ! ω ⊃ ⁼
ה JSPD JOG ♪ ! ▬Ķ — Ɽ ⁱ ☻

Ẽ ORIGIN ♪ ! ▬Ķ

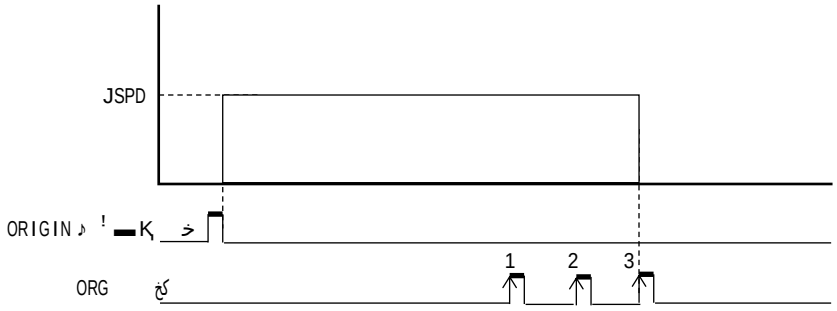
♪ ! ▬Ķ | ⁸ ORIGIN SCAN ♪ ! ▬Ķ ≤ ORIGIN CONSTANT SCAN ♪ ! ▬Ķ ⅜ ≠ ⅝ ╕ Δ ⁹

♪ ! ▬Ķ — ⁸
ORG ☵ —3○► fi ♩ — ▪ ▬ ♥ ▫ Ķ ◄ ℘ ☺ ≠ ⁸ ☵ ╩ ↕ ⅄ ∑ ╕ Δ ⁹
ה ORG DETECT EDGE = 0 ORG ☵ — 0 ǵ 1 ▪ ▬ ♥ ▫ Ķ ◄ ℘ ☺ ╩ Δ ╢
ה ORIGIN COUNT D3--D0 = H'2 3○► fi ♩ — ◄ ℘ ☺ ≠ ⁸ ☵ ╩ ↕ ⅄ ∑ ╢

ÜÜ ORIGIN SCAN ♪ ! ▬Ķ
♪ ! ▬Ķ — Ɽ ! ω ⊃ ⁼ ≠ ⁸ SCAN ♪ ! ▬Ķ ╩ ™ ╕ Δ ⁹
ORG ☵ — ◄ ℘ ☺ ╩ Δ ╢ ≤ ☵ ⇔ ╕ Δ ⁹



ÜÜ ORIGIN CONSTANT SCAN ♪ ! ▬Ķ
JOG ♪ ! ▬Ķ — Ɽ ⁱ ☻ JSPD ≠ ⁸ ♪ ! ▬Ķ ╩ ™ ╕ Δ ⁹
ORG ☵ — ◄ ℘ ☺ ╩ Δ ╢ ≤ ☵ ⇔ ╕ Δ ⁹

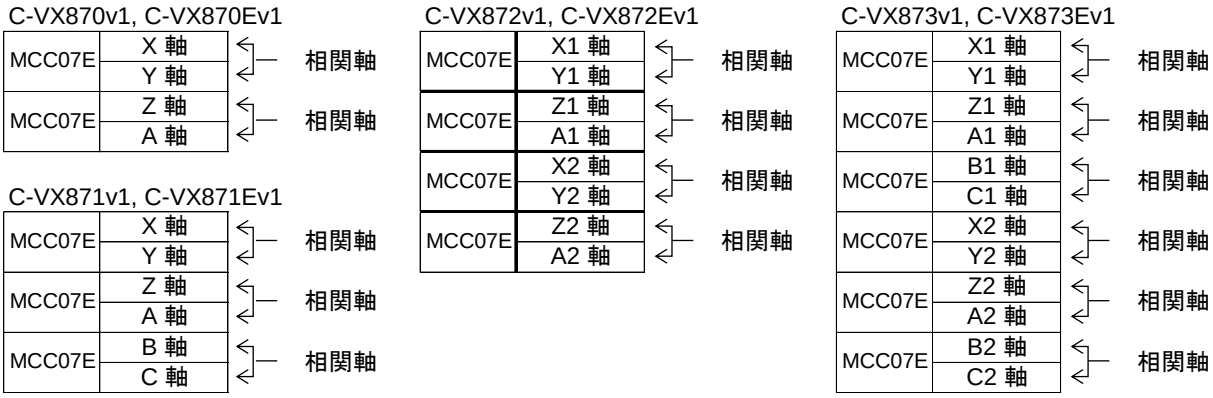


3-1-6. 補間ドライブ

補間ドライブには、相関軸で実行する相関 2 軸補間ドライブと、任意軸間で実行する任意軸補間ドライブがあります。

(1) 相関 2 軸補間ドライブ

相関軸で行う補間ドライブです。当製品における相関軸は以下の通りです。



相関軸で 2軸直線補間ドライブ、および 2軸円弧補間ドライブが行えます。
相関 2軸直線補間ドライブ … MAIN XY STRAIGHT CP コマンド
相関 2軸円弧補間ドライブ … MAIN XY CIRCULAR CP コマンド

・ 相関 2軸補間ドライブは、コマンド実行軸の加減速パラメータで、補間ドライブの基本パルスを発生します。
補間制御部は、発生した基本パルスを補間演算して補間パルスを出力します。

・ 相関 2軸補間ドライブのコマンドは、相関軸のどちらの軸に実行しても有効です。
コマンドの実行で 2軸のドライブを開始します。

・ 同期スタート機能は、コマンド実行軸で有効です。他軸の同期スタート機能は無効になります。
コマンド実行軸の STBY = 0 で、他軸も STBY = 0 になります。

・ LIMIT 停止指令、減速停止指令、即時停止指令は、相関軸両軸のどちらで発生しても有効です。

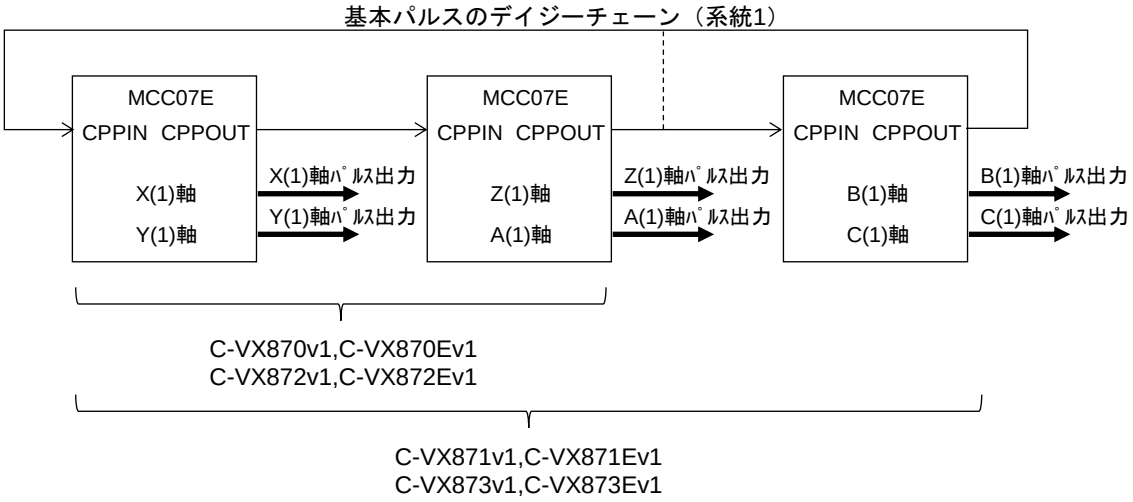
・ エラーが発生した場合は、エラー該当軸が ERROR = 1 になります。
但し、エラーによる停止機能は、X, Y 軸のどちらでエラーが発生しても有効です。

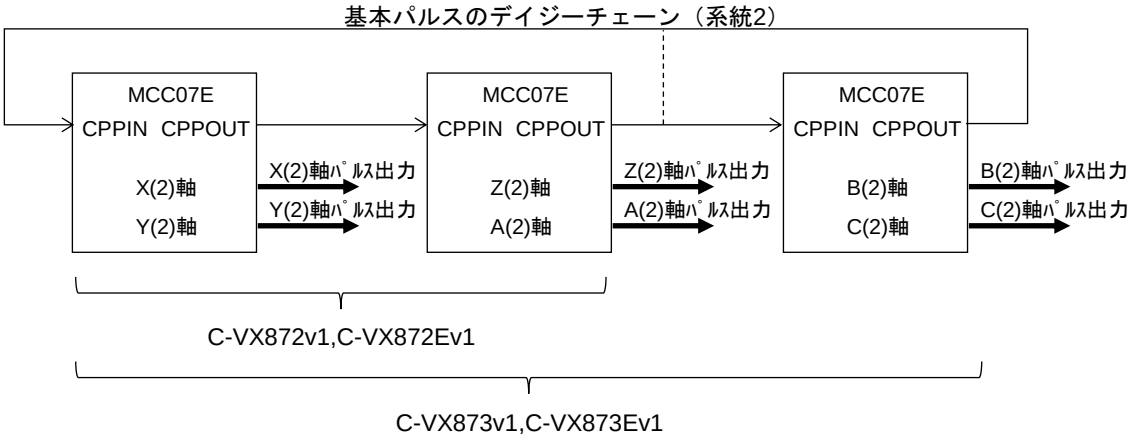
・ 両軸のドライブが終了すると、両軸が BUSY = 0 になります。
$\overline{\text{DEND}}$/$\overline{\text{PO}}$ 信号または $\overline{\text{DRST}}$ 信号を<サーボ対応>に設定している場合は、両軸の<サーボ対応>が終了した後に、両軸が BUSY = 0 になります。

(2) 任意軸補間ドライブ

基本パルスのデイジーチェーン接続を使用して任意軸で行う補間ドライブです。
各軸 MCC07E の CPPOUT 端子と CPPIN 端子はデイジーチェーン接続で繋がっています。
任意軸補間ドライブではメイン軸のチップが CPPOUT 端子に補間ドライブの基本パルスを出力します。
サブ軸のチップは基本パルスを CPPIN 端子から入力して CPPOUT 端子に出力します。

・ C-VX872v1, C-VX873v1 にはデイジーチェーン接続が 2系列あります。
　任意軸補間ドライブはデイジーチェーン接続の系列内の軸間で行います。





任意軸で多軸直線補間ドライブ、および 2軸円弧補間ドライブが行えます。

任意多軸直線補間ドライブ ･･･	MAIN STRAIGHT CP コマンド (メイン軸)
	SUB STRAIGHT CP コマンド (サブ軸)
任意 2軸円弧補間ドライブ ･･･	MAIN CIRCULAR CP コマンド (メイン軸)
	SUB CIRCULAR CP コマンド (サブ軸)

・ 任意軸補間ドライブは、メイン軸の加減速パラメータで、補間ドライブの基本パルスを発生します。
　補間制御部は、発生した基本パルスを補間演算して補間パルスを出力します。

・ サブ軸は、CPPIN 端子から入力するパルスを補間ドライブの基本パルスにします。
　補間制御部は、CPPIN 端子の入力パルスを補間演算して補間パルスを出力します。

・ メイン軸補間ドライブは、各サブ軸にサブ軸補間ドライブを実行した後に、メイン軸に実行してください。
　メイン軸はコマンドの実行で任意軸補間ドライブを開始します。

・ 同期スタート機能は、各軸で有効です。

Ẽ ش ♪ ! ▬Ķ — تج

ὶ ω▬fi ≠ تج ⅜ خ ⇔ √

ה تج خ ╩ ⇔ √ | ⁸ ω▬fi — Ꞧ ⁱ ☻ ╩ تج ⇔ ≡ ⁸ ♪ ! ▬Ķ ╩ ǧ ⇔ ╕ Δ ⁹

ה تج خ ╩ ⇔ √ | ⁸ ω▬fi — Ꞧ ⁱ ☻ ⅜ Ł ▬ ʒ ⁱ — ≤ ⅝ ⌐ ⁸ Ꞧ ⁱ ☻ ╩ تج ⇔ ≡ ⁸
♪ ! ▬Ķ ╩ ǧ ⇔ ╕ Δ ⁹

ה تج خ — ⇔ √ ω▬fi | Ꞧ ⁱ ☻ تج ⌐ ♪ ! ▬Ķ ╩ ǧ ⇔ ╕ Δ ⅜ ⁸ ◘ Ķ | ♪ ! ▬Ķ ╩
ǧ ⇔ ╕ ∑ ╪ ⁹
(م ⇔ ⁸ Ꞧ ⁱ ☻ ⅜ تج Δ ╫ — ≠ ◘ Ķ — Ꞧ ⁱ ☻ | تج ⇔ ╕ Δ)
ω▬fi ⅜ تج خ ⌐ ╫ ╡ ♪ ! ▬Ķ ╩ ǧ ⇔ √ | ⁸ ◘ Ķ Δ ═ ≡ ⌐ تج خ ╩ ⇔ ≡ ⁸
♪ ! ▬Ķ ╩ ǧ ↕ ∑ ≡ ɔ ∞ ↕ ™ ⁹

ὶ ◘ Ķ ≠ تج ⅜ خ ⇔ √

ה تج خ ╩ ⇔ √ | ⁸ — چ Ꞧ ⁱ ☻ ⅜ Ł ▬ ʒ ⁱ — ≤ ⅝ ⌐ ⁸ ♪ ! ▬Ķ ╩ ǧ ⇔ ╕ Δ ⁹

ה تج خ ╩ ⇔ √ | ⁸ — چ Ꞧ ⁱ ☻ ⅜ Ł ▬ ʒ ⁱ — ≤ ⅝ ⌐ ⁸ ♪ ! ▬Ķ ╩ ǧ ⇔ ╕ Δ ⁹

ה تج خ — ⇔ √ ◘ Ķ | Ꞧ ⁱ ☻ تج ⌐ ♪ ! ▬Ķ ╩ ǧ ⇔ ╕ Δ ⅜ ⁸ ث— | ♪ ! ▬Ķ ╩
ǧ ⇔ ╕ ∑ ╪ ⁹
(ω▬fi — Ꞧ ⁱ ☻ | تج ⇔ ≡ ™ ⌂ ™ — ≠ ⁸ ω▬fi ⅔ ╫ ┘ ث— ◘ Ķ — Ꞧ ⁱ ☻ ╙ تج ⇔ ╕ ∑ ╪)
◘ Ķ ⅜ تج خ ⌐ ╫ ╡ ♪ ! ▬Ķ ╩ ǧ ⇔ √ | ⁸ ث— Δ ═ ≡ — تج ⌐ خ ╩ ⇔ ≡ ⁸
♪ ! ▬Ķ ╩ ǧ ↕ ∑ ≡ ɔ ∞ ↕ ™ ⁹

ὶ CPP STOP ≤ CPPIN ∀ ☻ ▾
ω▬fi — CPP STOP ≤ ◘ Ķ — CPPIN ∀ ☻ ▾ ╩ ⌐ ⇔ ≡ ⁸ ش ♪ ! ▬Ķ ╩ Δ ╫ ≤ ⁸
◘ Ķ ⌐ ◄ ! ɔ ⅜ ⇔ √ ⌐ ⁸ Δ ═ ≡ — — Ꞧ ⁱ ☻ ╩ تج ↕ ∑ ╫ ↓ ≤ ⅜ ≠ ⅝ ╕ Δ ⁹
ה ◄ ! ɔ ⅜ ⇔ √ ◘ Ķ | ⁸ CPPIN ∀ ☻ ▾ ≠ CPPOUT ╩ تج ⇔ ╕ Δ ⁹

ה ω▬fi | ⁸ CPP STOP ≠ ♪ ! ▬Ķ ╩ ǧ ⇔ ⁸ CPPOUT ╩ ǧ ⇔ ╕ Δ ⁹

ה ث— ◘ Ķ | ⁸ ω▬fi — CPPOUT ǧ ≠ Ꞧ ⁱ ☻ ╩ تج ⇔ ╕ Δ ⁹

ǧ ᵒ ˣ
خ ̈ — ≠ ♪ ! ▬Ķ ╩ ⇔ √ ⌐ ⁸ ♪ ! ▬Ķ ╩ ⇔ ≡ ™ ⌂ ™ ⅛ ╬ ╙ Ꞧ ⁱ ☻ ⅜
↕ ╫ ≡ ⇔ ╕ e لحي ث ⅜ ⅜ ⇔ ╕ Δ ⁹
ŋ Y ⌐ محي ♪ ! ▬Ķ ╩ Δ ╫ ⁹
ᵒ Y ⌐ ⌐ ⇔ √ ♪ ! ▬Ķ ⅜ محي ♪ ! ▬Ķ — ≤ ⅝ ⁸
X ⌐ ⁸ ω▬fi محي ♪ ! ▬Ķ ⁹ ╕ √ | ⁸ ◘ Ķ محي ♪ ! ▬Ķ ⁹ ╩ Δ ╫ ⁹

↓ — ≤ ⅝ X — ث ⌐ ⁸ ♪ ! ▬Ķ ╩ ⇔ ≡ ™ ⌂ ™ Y ⅛ ╬ ╙ محي ♪ ! ▬Ķ —
Ꞧ ⁱ ☻ ⅜ ↕ ╫ ≡ ⇔ ╕ ™ ╕ Δ ⁹ ↓ — لحي ث | X ≤ Y ⅜ ≠ ╙ ⌐ ⇔ ╕ Δ ⁹
╕ √ ⁸ ↓ — لحي ث | ≠ ⌐ ⇔ ╕ Δ ⁹

ᵒ ˣ
⁸ ω▬fi محي ♪ ! ▬Ķ ⁹ ⅔ ╫ ┘ ⁸ ◘ Ķ محي ♪ ! ▬Ķ ⁹ ╩ Δ ╫ ⌐ ⁸
♪ ! ▬Ķ ╩ ⇔ ™ ⌂ ⌐ ⇔ ≡ ⁸ محي ♪ ! ▬Ķ خ ♪ ! ▬Ķ — ♪ ! ▬Ķ ╩ ⇔ ≡
ɔ ∞ ↕ ™ ⁹

ئ ŋ Y ⌐ محي ♪ ! ▬Ķ ╩ Δ ╫ ⁹
ᵒ Y ⌐ ⁸ 0 — ▪ ♪ ™ ☻ INDEX ♪ ! ▬Ķ ⁹ ╩ Δ ╫ ⁹
X ⌐ ⁸ ω▬fi محي ♪ ! ▬Ķ ⁹ ╕ √ | ⁸ ◘ Ķ محي ♪ ! ▬Ķ ⁹ ╩
Δ ╫ ⁹

(3) ♪ ! ▬Ķ

⇔ ≡ ⇔ ╕ Δ ⁹ ⁸ ⇔ ≤ ─ ─ ─ Ꞁ ⁱ ☻ ─ ▬Ķ ! ♪ ⁸ 5MHz (任意軸補間ドライブは 4MHz) ≠ Δ ⁹

指定直線に対する位置誤差は、Ñ0.5 LSB です。

⁸ -2,147,483,648 +2,147,483,647 32 ≠ Δ ⁹ ☻ ▪ ♪ ⅝ ≠

─ Ꞁ ⁱ ☻ ⅜ INDEX ♪ ! ▬Ķ ≤ ─ ♪ ! ▬Ķ ≤ ⌂ ╡ ╕ Δ ⁹

♪ ! ▬Ķ ─ Ꞁ ⁱ ☻ ⁸ ─ ⅞ e ⌐ ⌂ ╡ ╕ Δ ⁹

ה 2 ♪ ! ▬Ķ ⁸ ♪ fi ⅛ Ꞁ ⁱ ☻ ⇔ ╕ Δ ⁹

2 ≤ ─ Ꞁ ⁱ ☻ ⇔ ≡ Ꞁ ⁱ ☻ ⇔ ╕ Δ ⁹

ה ش ♪ ! ▬Ķ ─ ▬ fi ♪ ! ▬Ķ ⁸ ♪ fi ⅛ Ꞁ ⁱ ☻ ⇔ ╕ Δ ⁹

ה ش ♪ ! ▬Ķ ─ Ķ ♪ ! ▬Ķ ⁸ CPPIN ⅛ ⅞ Δ Ꞁ ⁱ ☻ Ꞁ ⁱ ☻ ≤ ⇔ ╕ Δ ⁹

2 ♪ ! ▬Ķ ⁸ ─ ♪ ! ▬Ķ Ꞁ ! ▬Ķ ⁸ ─ ⅜ ≠ Δ ⁹

ה ─ ♪ ! ▬Ķ ─ Ꞁ ! ▬Ķ

ה XLONG POSITION ⁸ XSHORT POSITION X ─ ≤ ─ ▪ ♪ ☻

ה YLONG POSITION ⁸ YSHORT POSITION Y ─ ≤ ─ ▪ ♪ ☻

ش ♪ ! ▬Ķ ─ ▬ fi ♪ ! ▬Ķ ⁸ ─ ♪ ! ▬Ķ Ꞁ ! ▬Ķ ─ ⅜ ≠ Δ ⁹

ה ─ ♪ ! ▬Ķ ─ Ꞁ ! ▬Ķ

ה CP SPEC CPPOUT

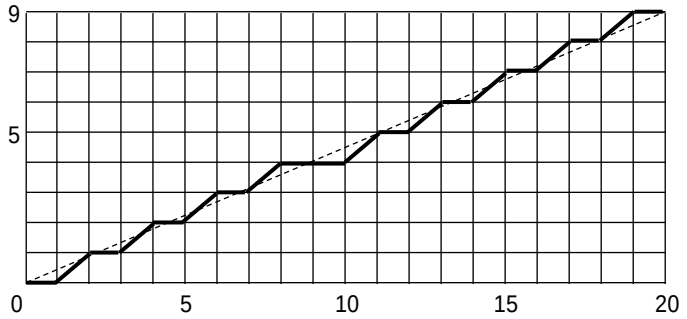
ה LONG POSITION ⁸ SHORT POSITION ─ ≤ ─ ▪ ♪ ☻

ش ♪ ! ▬Ķ ─ Ķ ♪ ! ▬Ķ ⁸ ─ ♪ ! ▬Ķ Ꞁ ! ▬Ķ ─ ⅜ ≠ Δ ⁹

ה CP SPEC CPPOUT

ה LONG POSITION ⁸ SHORT POSITION ─ ≤ ─ ▪ ♪ ☻

Ẽ **♪ ! ▬Ķ ─ (20: 9 ─ ♯)**



ה ♪ ! ▬Ķ ─ ⁸ ⁹ Δ ╕ ™ ⌐ ⊤ ⇈ ≤ ⁸ ♪ ! ▬Ķ ─ ⁸ ⁹ Δ ╕ ↑ ⇈ ⌐ Ꞁ ⁱ ☻ ─ ≠ ╕ Δ ⇈ ⁸ ♪ ! ▬Ķ ─ SCAN ה

ה INDEX ♪ ! ▬Ķ ─ ⁸ ─ Ꞁ ⁱ ☻ ⅜ ─ Ꞁ ⁱ ☻ ⌐ ⌂ ⇈ ≤ ♪ ! ▬Ķ ⇈ ⁹ Δ ╕ ⇔ ╕ Δ ⁹

ὺ ─ ≤

Ꞁ ⁱ ☻ ⅜ ⅝ ™ ─ ⅜ ⁸ ↕ ™ ─ ⅜ ⁹ Δ ╕ ╕ ⌂ ⌐

(4) [illegible]

[illegible] 5MHz (任意軸補間ドライブは 4MHz) [illegible]

指定円弧曲線に対する位置誤差は、Ñ1 LSB です。

[illegible] -8,388,608 +8,388,607 24 [illegible]

[illegible] -2,147,483,648 +2,147,483,647 32 [illegible]

[illegible] INDEX [illegible]

[illegible] 2 [illegible]

[illegible] 2 [illegible] XCP, YCP [illegible]

[illegible] CPPIN [illegible]

[illegible] 2 [illegible]

[illegible]

[illegible] CIRCULAR XPOSITION :現在位置の X 座標アドレス

[illegible] CIRCULAR YPOSITION :現在位置の Y 座標アドレス

[illegible] CIRCULAR PULSE :目的地の短軸座標までの短軸パルス数

[illegible]

[illegible] CP SPEC :CPPOUT 出力

[illegible] CIRCULAR XPOSITION :現在位置の X 座標アドレス

[illegible] CIRCULAR YPOSITION :現在位置の Y 座標アドレス

[illegible] CIRCULAR PULSE :目的地の短軸座標までの短軸パルス数

[illegible]

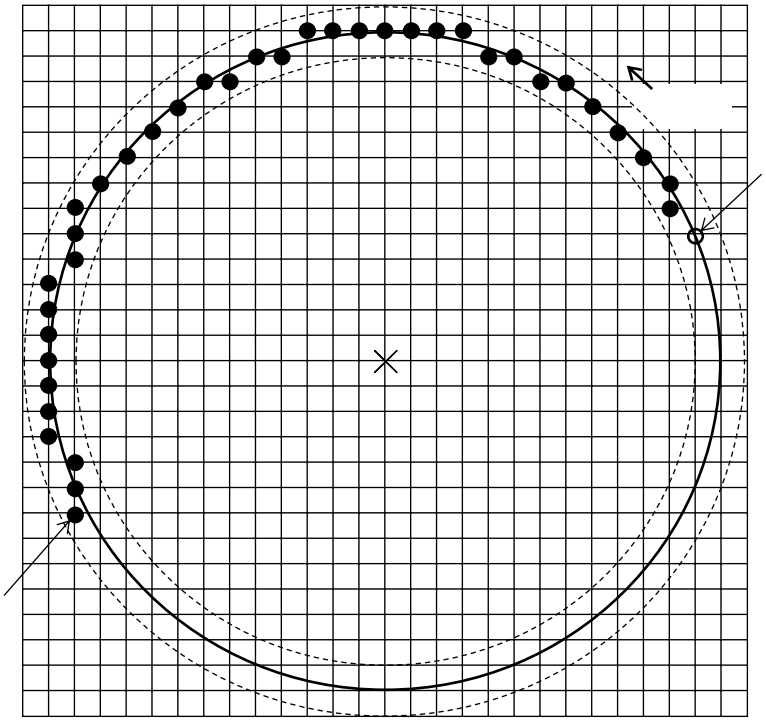
[illegible] CP SPEC :CPPOUT 出力

[illegible] CIRCULAR XPOSITION :現在位置の X 座標アドレス

[illegible] CIRCULAR YPOSITION :現在位置の Y 座標アドレス

[illegible] CIRCULAR PULSE :目的地の短軸座標までの短軸パルス数

Ẽ [illegible] **(CCW [illegible])**



[illegible]

[illegible] SCAN [illegible]

[illegible]

[illegible] INDEX [illegible]

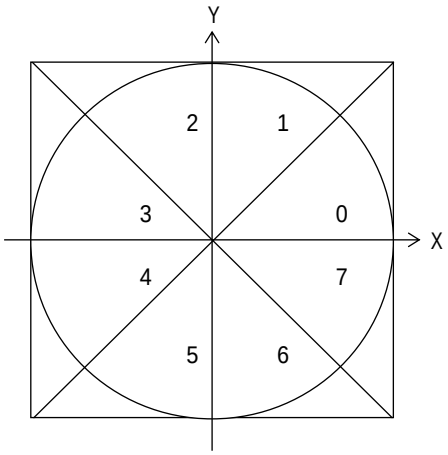
Ù² ⌂ ⌂ —

⌂ — ⌂ ⌂ 0, 0 ≤ ⇔ √ ⌂ ⌂ — X, Y ⌐ ⅔™ ≡

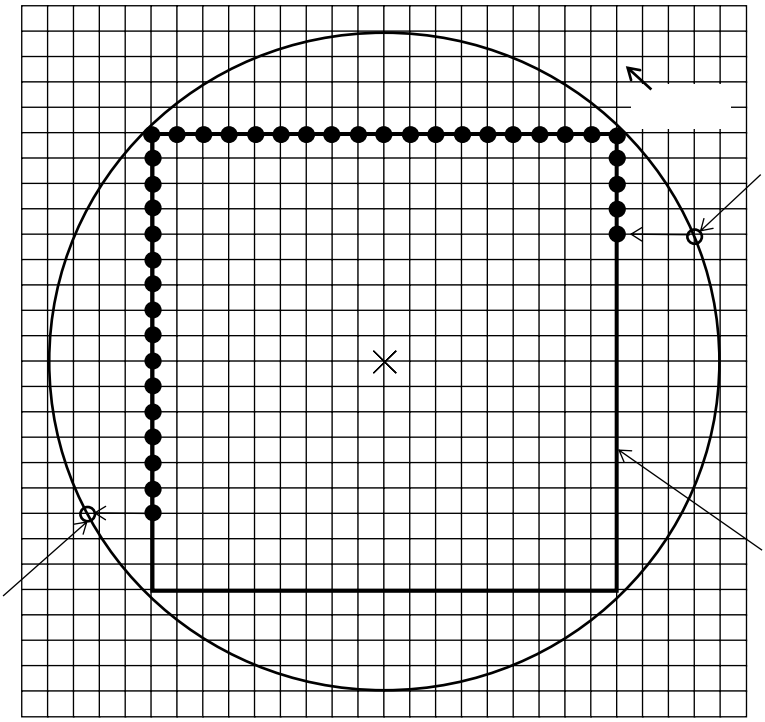
X, Y — ⌂ ⅜ ⌂ ↕™ — ⅜ ⌐ ⌂ ⌂ ¬ Δ ⁹

— 1, 2, 5, 6 | X ⅜ ⁸ 0, 3, 4, 7 |

Y ⅜ ≢ Δ ⁹



Ẽ ⌂ ♪ ! ▬Ķ Ꞧ i ● — (CCW — ⌂)



⌂ —

⌂ تج

⌂ ♪ ! ▬Ķ | ⌂ ⁸ ⌂ — ⌂ ⅛ ⌂ √ ⌂ ⅜ سخ ♪ ! ▬Ķ — Ꞧ i ☻(Ꞧ i ☻)⌂

⌐ ⇔ ⁸ سخ | Ꞧ i ☻(Ꞧ i ☻)⌂ ⇔ ≡ Ꞧ i ☻ ⌂ ⇔ ¬ Δ ⁹

ù Ɽ ⁱ ☻ —
R —ھي ⌐ ⅔ ↑ ╢ √ ╡ — Ɽ ⁱ ☻ Ps │ ⁸ ژ خ — س ≢ ⇔ ╕ Δ ⁹

K int(R l 2) int() │ ژ خ ╩ ╡ ≡ √
(1) R^{2} ' K^{2} $(K 1)^{2}$ — ≤ ⅝

س | K^{2} $(K 1)^{2}$ R^{2} | | K^{2} K^{2} R^{2} | — ≤ ⅝ │ ⁸ Ps K
| K^{2} $(K 1)^{2}$ R^{2} | ' | K^{2} K^{2} R^{2} | — ≤ ⅝ │ ⁸ Ps K 1/2

(2) R^{2} K^{2} $(K 1)^{2}$ — ≤ ⅝

س | K^{2} $(K 1)^{2}$ R^{2} | | $(K 1)^{2}$ $(K 1)^{2}$ R^{2} | — ≤ ⅝ │ ⁸ Ps K 1
| K^{2} $(K 1)^{2}$ R^{2} | ' | $(K 1)^{2}$ $(K 1)^{2}$ R^{2} | — ≤ ⅝ │ ⁸ Ps K 1/2

Ɽ ⁱ ☻ P int(— Ɽ ⁱ ☻ —)
1 √ ╡ — Ɽ ⁱ ☻ — 8 يى Ps x 8 ⅜ ⁸ 1 — Ɽ ⁱ ☻ ⌐ ⌂ ╡ ╕ Δ ⁹

ù Ɽ ⁱ ☻ — ئ 1 CCW
ڇ 0, 0 ⌐ ⇔ ≡ ⁸ ه ╩ 10, 5 ⁸ ╩ -10, -5 ≤ ⇔ ≡ CCW ↕ ∑ ╢ —
— ╕ ≢ — Ɽ ⁱ ☻ P │ ⁸ ژ خ — ╟ e ⌐ ⌂ ╡ ╕ Δ ⁹

R l (10^{2} 5^{2}) l 125, K int(R l 2) int(7.9) 7
R^{2} 125, K^{2} $(K 1)^{2}$ 113, ⌂ —≢ R^{2} K^{2} $(K 1)^{2}$

| K^{2} $(K 1)^{2}$ R^{2} | 12, | $(K 1)^{2}$ $(K 1)^{2}$ R^{2} | 3, ⌂ —≢
| K^{2} $(K 1)^{2}$ R^{2} | | $(K 1)^{2}$ $(K 1)^{2}$ R^{2} | — ≤ ⅝ │ ⁸ Ps K 1 8

Ɽ ⁱ ☻ P int(0 — Ɽ ⁱ ☻ 1, 2, 3 — Ɽ ⁱ ☻ 4 — Ɽ ⁱ ☻)
(8 5) (8 8 8) 5 32
CCW │ ≢ Δ ╢ —≢ ⁸ CIRCULAR PULSE -32 H'FFFF_FFE0

CCW — ه — — Ɽ ⁱ ☻ │ ⁸ ژ خ — ╟ e ⌐ ⌂ ╡ ╕ Δ ⁹
ה ه ⅜ 0, 2, 4, 6 — Ɽ ⁱ ☻ Ps (ه — — ثم)
ה ه ⅜ 1, 3, 5, 7 — Ɽ ⁱ ☻ ه — — ثم

CCW — ه — — Ɽ ⁱ ☻ │ ⁸ ژ خ — ╟ e ⌐ ⌂ ╡ ╕ Δ ⁹
ה ه ⅜ 0, 2, 4, 6 — Ɽ ⁱ ☻ ه — — ثم
ה ه ⅜ 1, 3, 5, 7 — Ɽ ⁱ ☻ Ps (ه — — ثم)

ù Ɽ ⁱ ☻ — ئ CW
ڇ 0, 0 ⌐ ⇔ ≡ ⁸ ه ╩ 20, 5 ⁸ ╩ -20, -5 ≤ ⇔ ≡ CW ↕ ∑ ╢ —
— ╕ ≢ — Ɽ ⁱ ☻ P │ ⁸ ژ خ — ╟ e ⌐ ⌂ ╡ ╕ Δ ⁹

R l (20^{2} 5^{2}) l 425, K int(R l 2) int(14.6) 14
R^{2} 425, K^{2} $(K 1)^{2}$ 421, ⌂ —≢ R^{2} K^{2} $(K 1)^{2}$

K^{2} $(K 1)^{2}$ R^{2} | 4, | $(K 1)^{2}$ $(K 1)^{2}$ R^{2} | 25, ⌂ —≢
| K^{2} $(K 1)^{2}$ R^{2} | ' | $(K 1)^{2}$ $(K 1)^{2}$ R^{2} | — ≤ ⅝ │ ⁸ Ps K 1/2 14.5

Ɽ ⁱ ☻ P int(0 — Ɽ ⁱ ☻ 7, 6, 5 — Ɽ ⁱ ☻ 4 — Ɽ ⁱ ☻)
5 (14.5 14.5 14.5) (14.5 5) 58
CW │ ≢ Δ ╢ —≢ ⁸ CIRCULAR PULSE 58 H'0000_003A

CW — ه — — Ɽ ⁱ ☻ │ ⁸ ژ خ — ╟ e ⌐ ⌂ ╡ ╕ Δ ⁹
ה ه ⅜ 0, 2, 4, 6 — Ɽ ⁱ ☻ ه — — ثم
ה ه ⅜ 1, 3, 5, 7 — Ɽ ⁱ ☻ Ps (ه — — ثم)

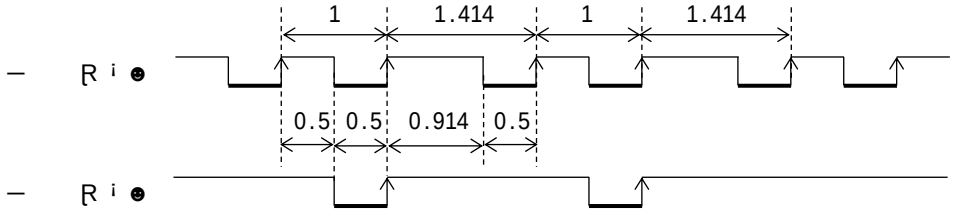
CW — ه — — Ɽ ⁱ ☻ │ ⁸ ژ خ — ╟ e ⌐ ⌂ ╡ ╕ Δ ⁹
ה ه ⅜ 0, 2, 4, 6 — Ɽ ⁱ ☻ Ps (ه — — ثم)
ה ه ⅜ 1, 3, 5, 7 — Ɽ ⁱ ☻ ه — — ثم

(5) [illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

- 1 軸のみパルス出力する位置（例: 0 Á90 Á180 Á270 Á）で終了した場合は正常です。
- 2 軸同時にパルス出力する位置（例:45 Á135 Á225 Á315 Á）で終了した場合に不具合が発生します。

線速一定制御有効の円弧補間ドライブ終了後は、
以下の円弧補間ドライブ（0 パルス、終了位置 0 Á）を実行して、正常終了にしてください。

- CIRCULAR XPOSITION SET コマンド (H'28) :H'00_0000 に設定
- CIRCULAR YPOSITION SET コマンド (H'29) :H'00_0000 に設定
- CIRCULAR PULSE SET コマンド (H'2A) :H'0000_0000 に設定
- MAIN CIRCULAR CP コマンド (H'38) :DATA1=H'0001 で実行

3-1-7. ♪ ! ▬Ķ CHANGE

♪ ! ▬Ķ ⌐ ⁸ ♪ ! ▬Ķ CHANGE ⌐ ⁹ ⁸ ⅝ ≤ ⅜ ↓ ≤ ⁹

(1) UP/DOWN/CONST ♪ ! ▬Ķ CHANGE

UP/DOWN/CONST — ♪ ! ▬Ķ CHANGE ⁸ STBY = 1 ⅛ ⁹

⁸ UP/DOWN/CONST — ♪ ! ▬Ķ CHANGE ⁹

⁸ ♪ ! ▬Ķ CHANGE ⁹

INDEX ♪ ! ▬Ķ — ⁸ ♪ ! ▬Ķ CHANGE ⁹

UP DRIVE ⁸ ⁹

DOWN DRIVE ⁸ ⁹

CONST DRIVE ⁸ ⁹

ה ♪ ! ▬Ķ — ⁸ CHANGE ⁹

ה S ♪ ! ▬Ķ — ⁸ S ⅛ ⁹

UP/DOWN/CONST — ♪ ! ▬Ķ CHANGE ⁸ ⁹

ה UDC SPEC UP/DOWN/CONST — ♪ ! ▬Ķ CHANGE

Ẽ UP/DOWN/CONST ♪ ! ▬Ķ ⅜ ≤ ♪

ORIGIN ♪ ! ▬Ķ ≤ MANUAL ♪ ! ▬Ķ ⁸ — SCAN ♪ ! ▬Ķ ⁹

♪ ! ▬Ķ ⁸ — CHANGE ⁹

COMMAND CODE	♪		
H'12	+SCAN	*P	SCAN ♪ ! ▬Ķ —
H'13	-SCAN	*P	SCAN ♪ ! ▬Ķ —
H'14	INC INDEX	*P	INDEX ♪ ! ▬Ķ —
H'15	ABS INDEX	*P	INDEX ♪ ! ▬Ķ —
H'18	ORIGIN SCAN	*P	ORIGIN SCAN ♪ ! ▬Ķ —
H'32	MAIN XY STRAIGHT CP	*P	2 ♪ ! ▬Ķ —
H'3A	MAIN XY CIRCULAR CP	*P	2 ♪ ! ▬Ķ —
H'30	MAIN STRAIGHT CP	*P	♪ ! ▬Ķ —
H'38	MAIN CIRCULAR CP	*P	♪ ! ▬Ķ —
		*P	MANUAL ♪ ! ▬Ķ — SCAN ♪ ! ▬Ķ —

ה ♪ ! ▬Ķ — ⁸ STBY = 1 SPEED CBUSY = 0 ⁹

♪ ! ▬Ķ — ⁸ SPEED CBUSY = 1 — ⁹

ה ♪ ! ▬Ķ CHANGE — ⅝ ⁸ SPEED CBUSY = 1 ⁸ SPEED CSET = 1 ⁹

ה ♪ ! ▬Ķ CHANGE — ⁸ SPEED CBUSY = 0 ⁸ SPEED CSET = 0 ⁹

ה — ⁸ SPEED CBUSY = 1 ⁸ SPEED CSET = 0 ⁹

ה INDEX ♪ ! ▬Ķ ⁸ ORIGIN SCAN ♪ ! ▬Ķ ⁸ MANUAL SCAN ♪ ! ▬Ķ — —

ה LSEND = 1 ⁸ SSEND = 1 ⁸ FSEND = 1 ⁸ ERROR = 1 —

SPEED CSET ⁸ ♪ ! ▬Ķ CHANGE ⁹

Ẽ **UP/DOWN/CONST ♪ ! ▬Ķ CHANGE كخ**

SS0, SS1 كخ — لأ≠ ⁸ UP/DOWN/CONST — ♪ ! ▬Ķ CHANGE ⅜≠ ⅝╕ Δ ⁹

SS0, SS1 كخ — ♪ ! ▬Ķ CHANGE ≠ │ ⁸ لأ — │ ⌐ ⌂ ╡ ╕ Δ ⁹

SS0, SS1 كخ — ♪ ! ▬Ķ CHANGE │ ⁸ SPEC INITIALIZE2 ◖ ∀ fi ♪ ≠ ⇔ ╕ Δ ⁹

ה SS0, SS1 كخ ≠ ⁸ UP DRIVE, DOWN DRIVE, CONST DRIVE — ♪ ! ▬Ķ CHANGE خ ⅜ ≠ ⅝╕ Δ ⁹

ה SS0 كخ —╖ ╨ ئو Δ ╢ ≤ ⁸ UP DRIVE — ♪ ! ▬Ķ CHANGE خ ⅜ ≠ ⅝╕ Δ ⁹

ה SS1 كخ —╖ ╨ ئو Δ ╢ ≤ ⁸ DOWN DRIVE — ♪ ! ▬Ķ CHANGE خ ⅜ ≠ ⅝╕ Δ ⁹

SS0 ╕ √ │ SS1 كخ —ᵐ ϛ ⁱ — ≠ ⁸ ♪ ! ▬Ķ CHANGE خ ╨ ⇔ ╕ Δ ⁹

ὺ SS0, SS1 كخ ⌐ ╟╫ ♪ ! ▬Ķ CHANGE لأ

ORIGIN ♪ ! ▬Ķ ≤ MANUAL ♪ ! ▬Ķ≠ │ ⁸ — SCAN ♪ ! ▬Ķ ⌐ ≠ Δ ⁹

♪ ! ▬Ķ≠ │ ⁸ ω▬ fi — Ꞧ ! ω ↄ ⌐ ⇔ ≡ CHANGE خ ╨ ⇔ ╕ Δ ⁹

ה كخ —ᵐ ϛ ⁱ — │ ⁸ SPEED CBUSY = 0 ≤ ⌂ ╢ ♪ ! ▬Ķ — STBY = 1 ⅛ ╟ ⌐ ⌂ ╡ ╕ Δ ⁹

STBY = 1 ≠ ⇔ √ كخ —ᵐ ϛ ⁱ ⅜ ♪ ! ▬Ķ CHANGE خ — │ ⁸ STBY = 1 ⅛ ╟ ♪ ! ▬Ķ

CHANGE خ ╨ ⇔ ╕ Δ ⁹

ה ♪ ! ▬Ķ CHANGE خ ╨ Δ ╢ ≤ ⁸ SPEED CBUSY = 1 ⁸ SPEED CSET = 1 ⌐ ⌂ ╡ ╕ Δ ⁹

SPEED CBUSY = 1 — │ ⁸ كخ —ᵐ ϛ ⁱ — │ كا ⌐ ⌂ ╡ ╕ Δ ⁹

ה ♪ ! ▬Ķ CHANGE خ — ۏ ≠ ⁸ SPEED CBUSY = 0 ⁸ SPEED CSET = 0 ⌐ ⌂ ╡ ╕ Δ ⁹

⌐ ⁸ ♪ ! ▬Ķ CHANGE كخ —ᵐ ϛ ⁱ — ⅜ ⌐ ⌂ ╡ ╕ Δ ⁹

ה ♪ ! ▬Ķ CHANGE خ ⌐ ⁸ ≤ ∂ كخ ᵐ ϛ ⁱ ╕ √ │ ⌂ ⇔ ╨ ⇔ √

│ ⁸ SPEED CBUSY = 0 ⁸ SPEED CSET = 0 —╕ ╕ ≠ Δ ⁹

♪ ! ▬Ķ CHANGE خ ⌐ ⁸ ≤ ⌂ ╢ كخ ᵐ ϛ ⁱ ⌂ ╢ ♪ ! ▬Ķ CHANGE خ ╨

⇔ √ │ ⁸ SPEED CBUSY = 1 ⁸ SPEED CSET = 1 ⌐ ⌂ ╡ ⁸ — ♪ ! ▬Ķ CHANGE خ ╨

⇔ ╕ Δ ⁹

ה ڎ — ╨ Δ ╢ ≤ ⁸ SPEED CBUSY = 1 ⁸ SPEED CSET = 0 ▪ ⁱ ▬ ⌐ ⌂ ╡ ╕ Δ ⁹

ה INDEX ♪ ! ▬Ķ ⁸ ORIGIN SCAN ♪ ! ▬Ķ ⁸ MANUAL SCAN ♪ ! ▬Ķ — —

ה LSEND = 1 ⁸ SSEND = 1 ⁸ FSEND = 1 ⁸ ERROR = 1 —

SPEED CSET ╨ ▬ ⁱ ▪ ⇔ √ │ ⁸ ♪ ! ▬Ķ CHANGE خ │ ⌐ ⌂ ╡ ╕ Δ ⁹

(2) SPEED CHANGE

SPEED CHANGE ⁸ STBY = 1 ⅛ ⁹
⁸ SPEED CHANGE ⁹

⁸ SPEED CHANGE ⁹
INDEX ⁸ SPEED CHANGE ⁹

SPEED CHANGE ⁸ ⁹
⁸ ⅝ ⅔ ⁹
STBY = 1 ⁸ SPEED CHANGE ⁸ SPEED CHANGE ⁹
⁹
SPEED CHANGE ⁸ SPEED CHANGE ⁹
⁹
S ⁸ SPEED CHANGE ⁹
⁸ ⅛ ⁹ ⅛ ○

SPEED CHANGE ⁸ ⁹
SPEED CHANGE ⁹

SPEED CHANGE ⁸ ⅜ ⁹
SPEED CHANGE SPEC SPEED CHANGE

Ẽ **SPEED CHANGE** ⅜

ORIGIN MANUAL ⁸ SCAN ⁹
⁸ CHANGE ⁹

COMMAND CODE			
H'12	+SCAN	*P	SCAN
H'13	-SCAN	*P	SCAN
H'14	INC INDEX	*P	INDEX
H'15	ABS INDEX	*P	INDEX
H'18	ORIGIN SCAN	*P	ORIGIN SCAN
H'32	MAIN XY STRAIGHT CP	*P	2
H'3A	MAIN XY CIRCULAR CP	*P	2
H'30	MAIN STRAIGHT CP	*P	
H'38	MAIN CIRCULAR CP	*P	
		*P	MANUAL SCAN

STBY = 1 ⁸ SPEED CBUSY = 0 ⁹
⁸ SPEED CBUSY = 1 ⁹

SPEED CHANGE ⅝ ⁸ SPEED CBUSY = 1 ⁸ SPEED CSET = 1 ⁹

SPEED CHANGE ⁸ SPEED CBUSY = 0 ⁸ SPEED CSET = 0 ⁹

⁸ SPEED CBUSY = 1 ⁸ SPEED CSET = 0 ⁹
INDEX ⁸ ORIGIN SCAN ⁸ MANUAL SCAN
LSEND = 1 ⁸ SSEND = 1 ⁸ FSEND = 1 ⁸ ERROR = 1

SPEED CSET ⁸ CHANGE ⁹

(3) RATE CHANGE

RATE CHANGE ⊥ ⁸ STBY = 1 ⅛ ⌐ ⌂ ⊣ ⊤ Δ ⁹
RATE CHANGE ⊥ ⁸ ⊤ ⊥ — ♪ ! ▬Ϟ CHANGE ⊥ — ≤ ⌐ ⇔ ⊤ Δ ⁹
ה UP DRIVE ⁸ DOWN DRIVE ⁸ CONST DRIVE ⁸ SPEED CHANGE

⊥ ⊤ √ | ◄ ! ⊥ Δ ⊣ ≤ ⁸ RATE CHANGE ⊥ | ⌐ ⌂ ⊣ ⊤ Δ ⁹
INDEX ♪ ! ▬Ϟ — ⊥ Δ ⊣ ≤ ⁸ RATE CHANGE ⊥ | ⌐ ⌂ ⊣ ⊤ Δ ⁹

RATE CHANGE ⊥ | ⁸ ♪ ! ▬Ϟ CHANGE ⊥ — ♦ ⊃ ⊂ — ≠ Δ ⁹
RATE CHANGE ⊥ ⊥ Δ ⊣ ≤ ⁸ ♪ ! ▬Ϟ CHANGE ⊥ — ○ ⊃ ⊂ Ϟ ≤ ○ ⊃ ⊂ Ϟ — ♦ ⊃ ⊂ ⊥ ⁸
⇔ √ ♦ ⊃ ⊂ ⌐ ⇔ ⊤ Δ ⁹
ה RATE CHANGE ⊥ Δ ⊣ ≤ ⁸ ♪ — ث ! ▬Ϟ CHANGE ⊥ — ≤ ⌐ RATE CHANGE ⊥
⇔ ⁸ ♦ ⊃ ⊂ ⊥ ⇔ ⊤ Δ ⁹
ה ⇔ √ ♦ ⊃ ⊂ | ⁸ — ♪ ! ▬Ϟ CHANGE ⊥ ≠ Δ ⁹
— ♪ ! ▬Ϟ ⅜ ⊥ Δ ⊣ ≤ ⁸ ⇔ √ ♦ ⊃ ⊂ | ⌐ ⌂ ⊣ ⊤ Δ ⁹
ה ⊥ | ⁸ RATE SET ◄ ♪ fi ♪ ≠ ⇔ √ DCYCLE — ≠ ⊥ ⇔ ⊤ Δ ⁹

RATE CHANGE ⊥ ⌐ ⊣ ⊢ ♦ ⊃ ⊂ — | ⁸ ♪ ! ▬Ϟ CHANGE ⊥ — ⊤ — ≠ Δ ⁹
RATE CHANGE ⊥ ⇔ ≡ ⁸ R ! ω ⊃ ⊂ — | ⊤ ⊣ ⊤ ∑ ⊣ ⁹

Ē **RATE CHANGE ⊥ ⅜ ≤ ⌂ ⊣ ◄ ♪ fi ♪**
ORIGIN ♪ ! ▬Ϟ ≤ MANUAL ♪ ! ▬Ϟ ≠ | ⁸ — SCAN ♪ ! ▬Ϟ ⌐ ≠ Δ ⁹
♪ ! ▬Ϟ ≠ | ⁸ — R ! ω ⊃ ⊂ ⌐ ⇔ ≡ CHANGE ⊥ ⇔ ⊤ Δ ⁹

COMMAND CODE	◄ ♪ fi ♪		
H'12	+SCAN	*P	SCAN ♪ ! ▬Ϟ —
H'13	-SCAN	*P	SCAN ♪ ! ▬Ϟ —
H'14	INC INDEX	*P	▪ ♪ ☻ INDEX ♪ ! ▬Ϟ —
H'15	ABS INDEX	*P	▪ ♪ ☻ INDEX ♪ ! ▬Ϟ —
H'18	ORIGIN SCAN	*P	ORIGIN SCAN ♪ ! ▬Ϟ —
H'32	MAIN XY STRAIGHT CP	*P	2 ♪ ! ▬Ϟ —
H'3A	MAIN XY CIRCULAR CP	*P	2 ♪ ! ▬Ϟ —
H'30	MAIN STRAIGHT CP	*P	ش — ω ▬ fi ♪ ! ▬Ϟ —
H'38	MAIN CIRCULAR CP	*P	ش — ω ▬ fi ♪ ! ▬Ϟ —
		*P	MANUAL ♪ ! ▬Ϟ — SCAN ♪ ! ▬Ϟ —

ה ث — ♪ ! ▬Ϟ ⊥ Δ ⊣ ≤ ⁸ STBY = 1 ≠ SPEED CBUSY = 0 ⌐ ⌂ ⊣ ⊤ Δ ⁹
ث — ♪ ! ▬Ϟ خ — ≠ | ⁸ SPEED CBUSY = 1 — ⊤ ⊤ ≠ Δ ⁹

ה RATE CHANGE ⊥ — ⅝ ⊤ ≠ ⁸ RATE CSET = 1 ⌐ ⌂ ⊣ ⊤ Δ ⁹
RATE CSET = 1 ≠ ⊥ ⁸ RATE CHANGE ⊥ — ⅝ ⊤ | ≠ Δ ث ⅝ ⇔ ⊤ Δ ⁹
ث — ♪ ! ▬Ϟ CHANGE ⊥ — ⅝ ⊤ ≠ ⁸ SPEED CBUSY = 1 ⁸ SPEED CSET = 1 ⌐ ⌂ ⊣ ⊤ Δ

ה ث — ♪ ! ▬Ϟ CHANGE ⊥ — ≤ ⌐ RATE CHANGE ⊥ ⇔ ⁸ RATE CSET = 0 ⌐ ⌂ ⊣ ⊤ Δ ⁹
ث — ♪ ! ▬Ϟ CHANGE ⊥ — ≠ ⁸ SPEED CBUSY = 0 ⁸ SPEED CSET = 0 ⌐ ⌂ ⊣ ⊤ Δ

ה ⊤ — ⊥ Δ ⊣ ≤ ⁸ SPEED CBUSY = 1 ⁸ RATE CSET = 0 ▪ ⌐ ⌂ ⊣ ⊤ Δ ⁹
ה INDEX ♪ ! ▬Ϟ ⁸ ORIGIN SCAN ♪ ! ▬Ϟ ⁸ MANUAL SCAN ♪ ! ▬Ϟ — —
ה LSEND = 1 ⁸ SSEND = 1 ⁸ FSEND = 1 ⁸ ERROR = 1 —

RATE CSET ⊥ ▪ ⇔ √ | ⁸ ⊥ — RATE CHANGE ⊥ | ⌐ ⌂ ⊣ ⊤ Δ ⁹

(4) INDEX CHANGE

INDEX CHANGE ⁸ STBY = 1 ⅛ ⁹

⁸ INDEX CHANGE ⁹

⁸ INDEX CHANGE ⁹

⁸ INDEX CHANGE ⁹

INDEX CHANGE ⅜ ⁸ ⁹

ERROR STATUS — CHANGE CLR ERROR = 1 ⁹

⅜ INDEX CHANGE ⁸ ⁹

⁸ ERROR STATUS — INDEX CHANGE ERROR = 1 ⁹

INDEX CHANGE ⁸ INC/ABS/PLS INDEX CHANGE — ⁹

INC INDEX CHANGE ⁸ ⁹

⁸ INC INDEX ⁹

* INC INDEX CHANGE ⁸ ⁹

⁸ INC INDEX CHANGE ⁸ ⁹

ABS INDEX CHANGE ⁸ ⁹

⁸ ABS INDEX ⁹

PLS INDEX CHANGE ⁸ ⁹

⁸ INC INDEX ⁹

— INDEX ⁸ ⁹

INDEX CHANGE ⁸ ⁹

INDEX CHANGE ⅜ ⁸ ⁹

INDEX CHANGE SPEC INDEX CHANGE

Ẽ **INC/ABS/PLS INDEX CHANGE ⅜**

COMMAND CODE		
H'12	+SCAN *P	SCAN —
H'13	-SCAN *P	SCAN —
H'14	INC INDEX *P	INDEX —
H'15	ABS INDEX *P	INDEX —

⁸ STBY = 1 INDEX CBUSY = 0 ⁹

⁸ INDEX CBUSY = 1 — ⁹

INDEX CHANGE — ⅝ ⁸ INDEX CBUSY = 1 ⁸ INDEX CSET = 1 ⁹

INDEX CHANGE — ⁸ INDEX CBUSY = 0 ⁸ INDEX CSET = 0 ⁹

⅜ ⁸ DRIVE = 0 INDEX CBUSY = 1 ⁸ INDEX CSET = 0 ⁹

— ⁸ INDEX CBUSY = 1 ⁸ INDEX CSET = 0 ⁹

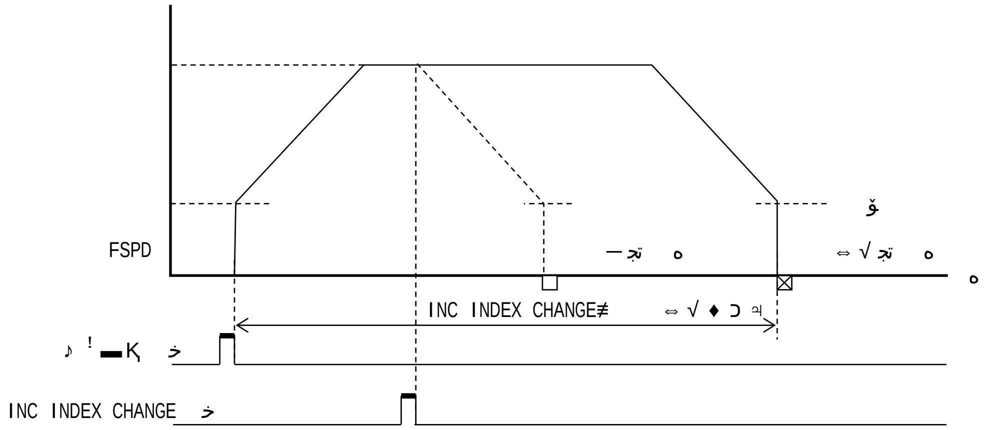
LSEND = 1 ⁸ SSEND = 1 ⁸ FSEND = 1 ⁸ ERROR = 1 —

INDEX CSET ⁸ INDEX CHANGE ⁹

INDEX CHANGE ⅜ ⁸ ERROR = 1 ⁹

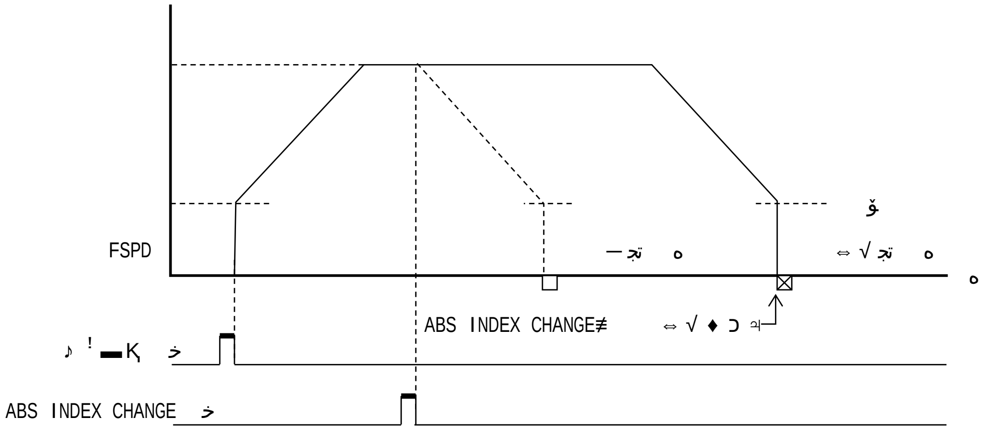
Ẽ **INC INDEX CHANGE — ﻷ**

⇔ √ ♦ ↄ ㅋ ╩ ⁸ ｏ ╩ ≤ Δ ╣ ▪ ♪‾ ☻ — ﺗﺠ ｏ ⌐ ⇔ ╕ Δ ⁹



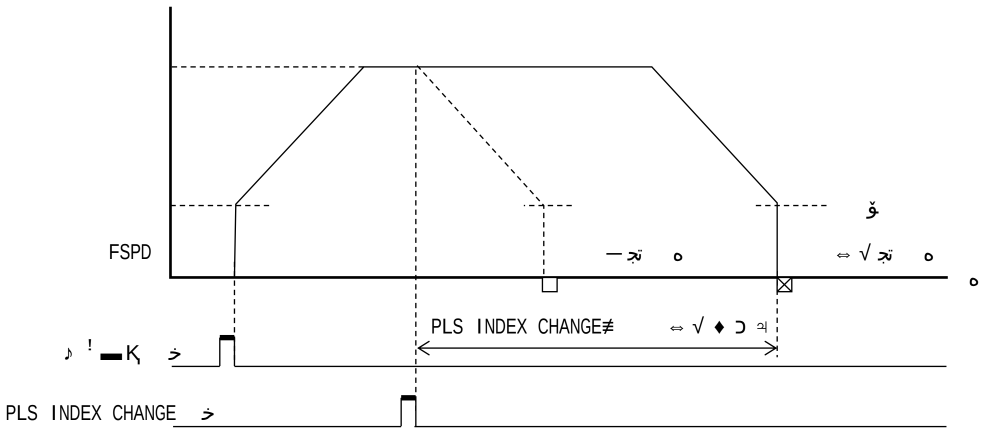
Ẽ **ABS INDEX CHANGE — ﻷ**

⇔ √ ♦ ↄ ㅋ ╩ ⁸ ▪ ♪‾ ☻ ○ ▶ ﬁ ㅋ ≢ ⇔ ≡ ™ ╣ ▪ ♪‾ ☻ — ﺗﺠ ｏ ⌐ ⇔ ╕ Δ ⁹



Ẽ **PLS INDEX CHANGE — ﻷ**

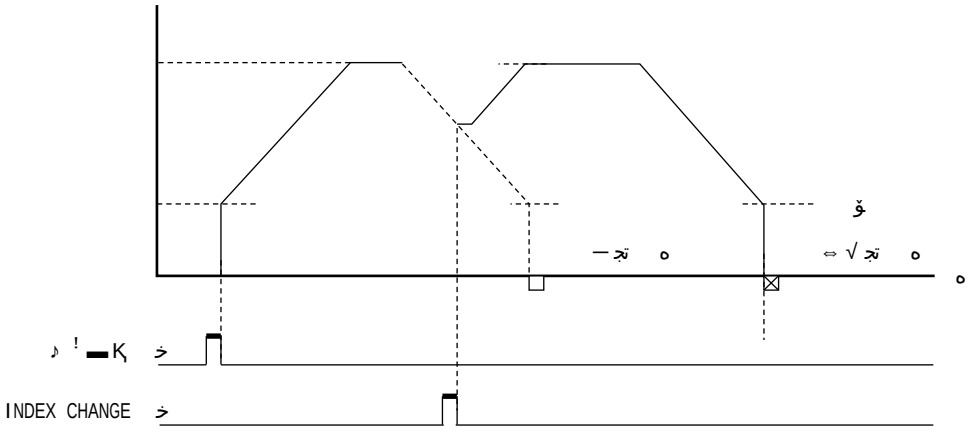
⇔ √ ♦ ↄ ㅋ ╩ ⁸ ﻷ — ｏ ╩ ≤ Δ ╣ ▪ ♪‾ ☻ — ﺗﺠ ｏ ⌐ ⇔ ╕ Δ ⁹



Ẽ **INDEX CHANGE**

ừ INDEX CHANGE

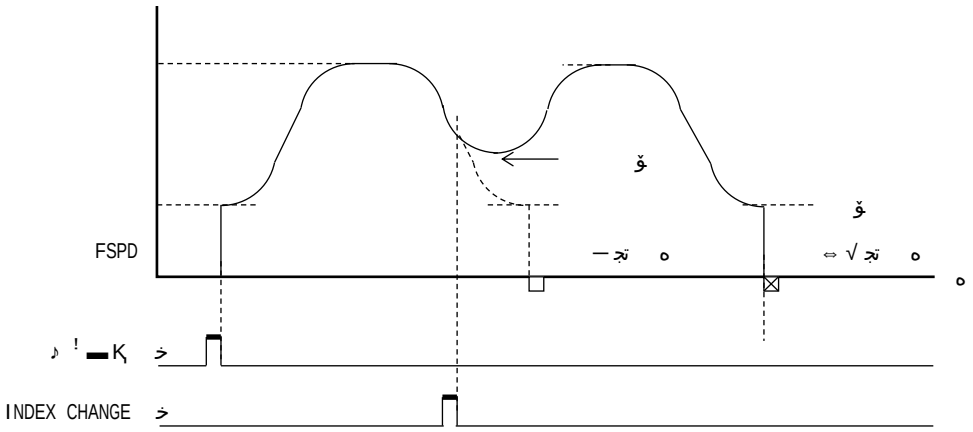
INDEX CHANGE



ừ S INDEX CHANGE

S INDEX CHANGE

S



3-1-8. MANUAL ドライブ

■ MANUAL モード

特殊 I/O コネクタの $\overline{\text{MAN}}$ 信号入力を LOW にすると全軸が MANUAL モードになります。
MANUAL モードでは、特殊 I/O コネクタの SEL A ～ SEL D 信号と $\overline{\text{CWMS}}$, $\overline{\text{CCWMS}}$ 信号入力の操作で
指定軸をドライブさせることができます。

・ $\overline{\text{MAN}}$ 信号入力が HIGH(オープン)のときはバスインターフェイスによりドライブを実行するモードです。
これを BUS モードと呼称します。

・ MANUAL モードでは全軸 STATUS1 PORT の MAN = 1、BUSY = 1 となります。
MANUAL モードへの切り替えは、全軸 BUSY = 0 のときに行ってください。

・ $\overline{\text{MAN}}$ 信号入力による MANUAL モードへの切り替えをアプリケーションから禁止することができます。
$\overline{\text{MAN}}$ 信号入力の禁止は MAN MASK コマンドで行います。
MANUAL モードの切り替え可能な状態では特殊 I/O コネクタの $\overline{\text{MAN RDY}}$ 出力が LOW になります。
切り替え禁止の状態では特殊 I/O コネクタの $\overline{\text{MAN RDY}}$ 出力が HIGH になります。

・ MANUAL ドライブを実行する際は LIMIT 停止機能を即時停止で御使用ください。

■ MANUAL ドライブ

MANUAL ドライブには MANUAL SCAN ドライブと MANUAL JOG ドライブがあります。
・ MANUAL ドライブの選択(SCAN / JOG)は SPEC INITIALIZE1 コマンドで行います。

● MANUAL ドライブの軸指定
MANUAL モード時に SEL A, SEL B, SEL C, SEL D 信号の操作で、MANUAL ドライブする軸を指定します。

SEL D	SEL C	SEL B	SEL A	C-VX870v1 C-VX870Ev1	C-VX871v1 C-VX871Ev1	C-VX872v1 C-VX872Ev1	C-VX873v1 C-VX873Ev1
LOW	**LOW**	**LOW**	**LOW**	X 軸	X 軸	X1 軸	X1 軸
LOW	**LOW**	**LOW**	HIGH	Y 軸	Y 軸	Y1 軸	Y1 軸
LOW	**LOW**	HIGH	**LOW**	Z 軸	Z 軸	Z1 軸	Z1 軸
LOW	**LOW**	HIGH	HIGH	A 軸	A 軸	A1 軸	A1 軸
LOW	HIGH	**LOW**	**LOW**	無効	B 軸	無効	B1 軸
LOW	HIGH	**LOW**	HIGH	無効	C 軸	無効	C1 軸
LOW	HIGH	HIGH	**LOW**	無効	無効	無効	無効
LOW	HIGH	HIGH	HIGH	無効	無効	無効	無効
HIGH	**LOW**	**LOW**	**LOW**	無効	無効	X2 軸	X2 軸
HIGH	**LOW**	**LOW**	HIGH	無効	無効	Y2 軸	Y2 軸
HIGH	**LOW**	HIGH	**LOW**	無効	無効	Z2 軸	Z2 軸
HIGH	**LOW**	HIGH	HIGH	無効	無効	A2 軸	A2 軸
HIGH	HIGH	**LOW**	**LOW**	無効	無効	無効	B2 軸
HIGH	HIGH	**LOW**	HIGH	無効	無効	無効	C2 軸
HIGH	HIGH	HIGH	**LOW**	無効	無効	無効	無効
HIGH	HIGH	HIGH	HIGH	無効	無効	無効	無効

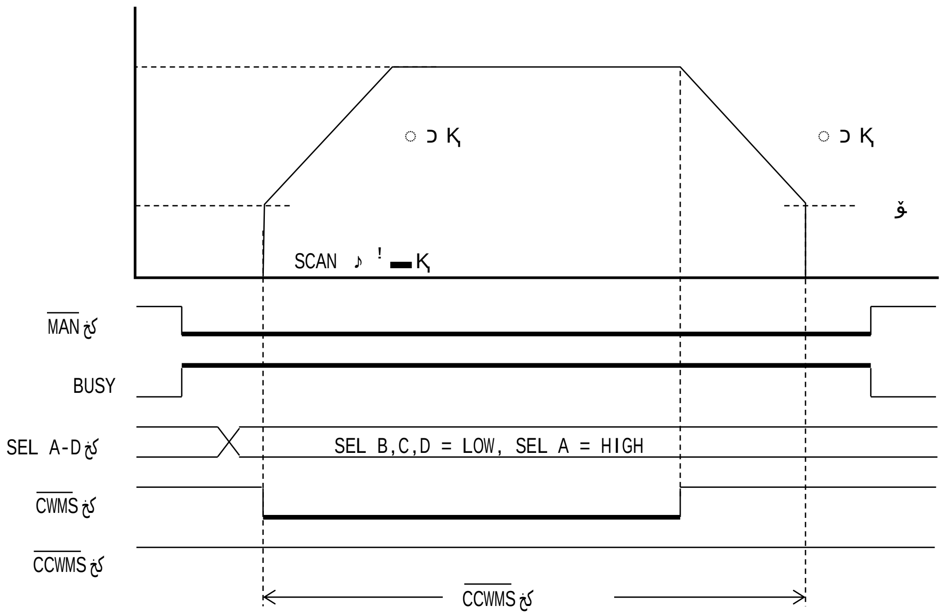
● MANUAL ドライブの実行
MANUAL モード時に $\overline{\text{CWMS}}$ 信号または $\overline{\text{CCWMS}}$ 信号を LOW レベルにすると、SEL A ～ SEL D 信号で
指定した軸がドライブを起動します。
$\overline{\text{CWMS}}$ 信号で＋方向に、$\overline{\text{CCWMS}}$ 信号で－方向にドライブします。
・ MANUAL ドライブのドライブパラメータは、各軸現在の設定値です。
・ MANUAL SCAN ドライブは $\overline{\text{CWMS}}$ 信号,または $\overline{\text{CCWMS}}$ 信号を HIGH にすると減速停止します。

● MANUAL SCAN ドライブの速度変更
MANUAL モード時、特殊 I/O コネクタの $\overline{\text{SS0}}$, $\overline{\text{SS1}}$ 信号は SEL A ～ SEL D 信号で指定した軸の
多用途センサ信号(SS0,SS1)に割り付けられます。
多用途センサ信号の入力機能を「UP/DOWN/CONST ドライブ CHANGE 指令入力」に設定した後、
MANUAL SCAN ドライブを起動すると、$\overline{\text{SS0}}$, $\overline{\text{SS1}}$ 信号の操作によって速度を変更することができます。
・ 汎用センサ信号の入力機能は SPEC INITIALIZE2 コマンドで設定します。
・ UP/DOWN/CONST ドライブ CHANGE 指令

$\overline{\text{SS1}}$	$\overline{\text{SS0}}$	ドライブ CHANGE 動作
HIGH	HIGH	CONST DRIVE
HIGH	**LOW**	UP DRIVE
LOW	HIGH	DOWN DRIVE
LOW	**LOW**	CONST DRIVE

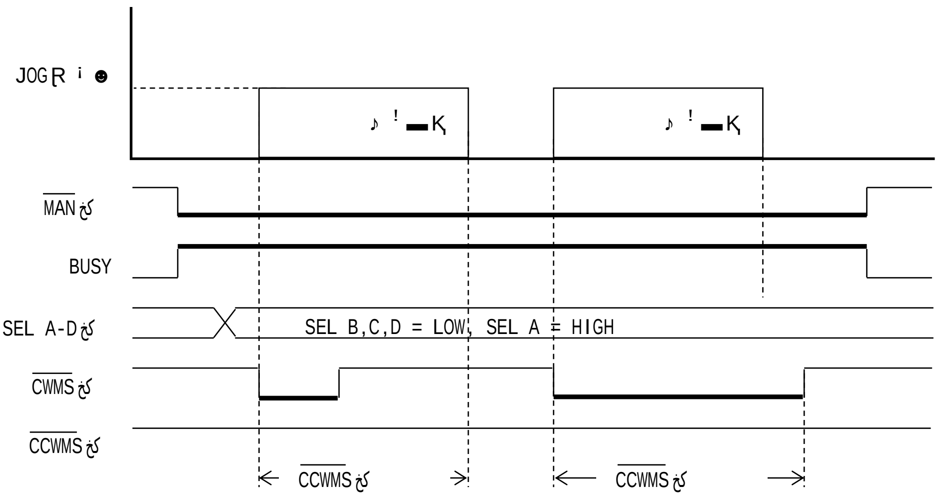
・ UP/DOWN/CONST ドライブ CHANGE 機能をご覧ください。

Ẽ **MANUAL SCAN ♪ ! ▬Ķ ﻸ ﯿ (Y ⁸)**



ŋ ﰋ BUSY = 0 ─ ≤ ⅝⌐ ⁸ MAN ﯼ ╨ LOW‾ ʐ ¡ ⌐ ⇔╕ Δ ⁽(MANUAL ≠ ɔ ♪ ⌐ ⌂╡ ╕ Δ ⁹)
º SEL A-D ﯼ ≢ Y ╨ ⇔╕ Δ ⁹
ɔ CWMS ﯼ ╨ LOW‾ ʐ ¡ ⌐ ⇔╕ Δ ⁹ (+ ─SCAN ♪ ! ▬Ķ ╨ ⇔╕ Δ ⁹)
ˆ CWMS ﯼ ╨ HIGH‾ ʐ ¡ ⌐ ⇔╕ Δ ⁹ (+ ─SCAN ♪ ! ▬Ķ ╨ ﭻ ⇔╕ Δ ⁹)
ˇ MAN ﯼ ╨ HIGH‾ ʐ ¡ ⌐ ⇔╕ Δ ⁹ (BUS ≠ ɔ ♪ ⌐ ╡ ╕ Δ ⁹)

Ẽ **MANUAL JOG ♪ ! ▬Ķ ﻸ ﯿ (Y ⁸)**



ŋ ﰋ BUSY = 0 ─ ≤ ⅝⌐ ⁸ MAN ﯼ ╨ LOW‾ ʐ ¡ ⌐ ⇔╕ Δ ⁽(MANUAL ≠ ɔ ♪ ⌐ ⌂╡ ╕ Δ ⁹)
º SEL A-D ﯼ ≢ Y ╨ ⇔╕ Δ ⁹
ɔ CWMS ﯼ ╨ LOW‾ ʐ ¡ ⌐ ⇔╕ Δ ⁹ (+ ─ JOG ♪ ! ▬Ķ ╨ ⇔╕ Δ ⁹)
(JOG Ʀ ¡ ☻ ─ Ʀ ¡ ☻ ╨ Δ ╢ ≤ ♪ ! ▬Ķ ╨ ǒ ⇔╕ Δ ⁹)
ˆ JOG ♪ ! ▬Ķ ⌐ CWMS ﯼ ╨ HIGH‾ ʐ ¡ ⌐ ⇔╕ Δ ⁹
ˇ MAN ﯼ ╨ HIGH‾ ʐ ¡ ⌐ ⇔╕ Δ ⁹ (BUS ≠ ɔ ♪ ⌐ ╡ ╕ Δ ⁹)

3-1-9. R i ● تج

R i ● تج | 8 چ — ♪ ! ▬ Ką ╩ ۆ ↕ ∑ ╫ ≠ Δ 9
R i ● تج ⌐ | 8 تج 8 تج 8 LIMIT تج 8 LIMIT تج ⅜№ ╡ ╕ Δ 9

(1) تج

تج خ — ▪ ◒ ♥ ▫ Ką ╩ Δ ╫ ≤ 8 چ — ♪ ! ▬ Ką R i ● ╩ ۆ ╕ ≠ ⇔ ≡ ⅛ ╬ 8
R i ● ╩ تج ⌐ ♪ ! ▬ Ką ╩ ۆ ⇔ ╕ Δ 9 تج ⌐ | 8 ژ خ — تج خ ⅜№ ╡ ╕ Δ 9
ה SLOW STOP ◖ ∀ fi ♪
ה تج ╩ تج ⌐ ⇔ √ ○ ▶ fi ♘ — ◖ fi R" ⊃ ♘
ה طس ╩ تج ⌐ ⇔ √ DALM كج
ה طس ╩ تج ⌐ ⇔ √ ☿ fi ◘ كج (SS0, SS1)

تج | DRIVE STATUS1 PORT —STBY = 1 ╕ √ | DRIVE = 1 — ≤ ⅝ ⌐ ⌐ ⌂ ╫ تج ≠ Δ 9
تج خ — ▪ ◒ ♥ ▫ Ką ≤ ⌐ 8 DRIVE STATUS1 PORT —SSEND = 1 ⌐ ⌂ ╡ ╕ Δ 9

(2) تج

تج خ — ▪ ◒ ♥ ▫ Ką ╩ Δ ╫ ≤ 8 چ — ♪ ! ▬ Ką ╩ ۆ ⇔ ╕ Δ 9
تج ⌐ | 8 ژ خ — تج خ ⅜№ ╡ ╕ Δ 9
ה FAST STOP ◖ ∀ fi ♪
ה تج ╩ تج ⌐ ⇔ √ ○ ▶ fi ♘ — ◖ fi R" ⊃ ♘
ה طس ╩ تج ⌐ ⇔ √ DALM كج
ה طس ╩ تج ⌐ ⇔ √ ☿ fi ◘ كج (SS0, SS1)
ה FSSTOP كج (FSSTOP1,2 كج) 8 FSSTOP كج
تج | DRIVE STATUS1 PORT —BUSY = 1 — ≤ ⅝ ⌐ ⌐ ⌂ ╫ تج ≠ Δ 9
تج خ — ▪ ◒ ♥ ▫ Ką ≤ ⌐ 8 DRIVE STATUS1 PORT —FSEND = 1 ⌐ ⌂ ╡ ╕ Δ 9
♦ ⊃ ♘ ◖ ∀ fi ♪ چ | 8 تج خ ╩ ⇔ ≡ ╚ ۆ ⇔ ╕ ∑ ╪ 9 FSEND ɥ ! ▬ ╚ ⇔ ╕ ∑ ╪ 9

(3) LIMIT تج

LIMIT تج | — ♪ ! ▬ Ką تج ≠ Δ 9 تج ⅛ تج ╩ ≠ ⅝ ╕ Δ 9
تج — 8 LIMIT تج خ — ▪ ◒ ♥ ▫ Ką ╩ Δ ╫ ≤ 8 چ — ♪ ! ▬ Ką R i ● ╩ ۆ ╕ ≠
⇔ ≡ ⅛ ╬ 8 R i ● ╩ تج ⌐ ♪ ! ▬ Ką ╩ ۆ ⇔ ╕ Δ 9
تج — 8 LIMIT تج خ — ▪ ◒ ♥ ▫ Ką ╩ Δ ╫ ≤ 8 چ — ♪ ! ▬ Ką ╩ ۆ ⇔ ╕ Δ 9
LIMIT تج | 8 SPEC INITIALIZE2 ◖ ∀ fi ♪ ≠ ⇔ ╕ Δ 9

LIMIT تج ⌐ | 8 ژ خ —LIMIT تج خ ⅜№ ╡ ╕ Δ 9
ה + ♪ ! ▬ Ką چ —LIMIT تج خ
CWLM كج
○ ▶ fi ♘ —COMP2 ◖ fi R" ⊃ ♘
ה - ♪ ! ▬ Ką چ —LIMIT تج خ
CCWLM كج
○ ▶ fi ♘ —COMP3 ◖ fi R" ⊃ ♘

DRIVE STATUS1 PORT —DRIVE = 1 — ≤ ⅝ ⌐ ⌐ ⌂ ╫ تج ≠ Δ 9
╕ √ 8 DRIVE STATUS2 PORT —DEND BUSY = 1 — ≤ ⅝ ⌐ | 8 LIMIT تج — —╖ ™╕ Δ 9
LIMIT تج خ ≤ ⌐ 8 DRIVE STATUS1 PORT —LSEND = 1 ⌐ ⌂ ╡ ╕ Δ 9

o ۆ x
LIMIT تج ╩ تج ≠ ئو Δ ╫ 8 LIMIT تج خ ⅜ ▪ ◒ ♥ ▫ Ką ≠ ╚
LIMIT ⌐ ♪ ! ▬ Ką ╩ Δ ╫ ≤ 8 R i ● ╩ ⇔ ≡ تج ⇔ ╕ Δ 9

تج ⌐ ╫ ◖ ! ⊃

تج — تج ♥ ⊃ ♘ ● — ● ♥ ⊃ ♘ ▫ Ką طس ♥ ◒ ▪ — كج تج | √ ╕ ≠ ◖ ! ⊃ Δ ╫ ↓ ≤ ⅜ ≠ ⅝ ╕ Δ 9
⇔ ⊃ | 5-1-7. 8 ◖ ! ⊃ 9 ╩ ⇔ ≡ ⊃ ∞ ↕ ™ 9

[◖ ! ⊃ ≠ ⅝ ╫ تج ♥ ⊃ ♘ ●]
ה STATUS1 PORT —FSEND
ה STATUS1 PORT —LSEND
ה STATUS1 PORT —SSEND

[◖ ! ⊃ ≠ ⅝ ╫ تج كج]
ה FSSTOP كج (FSSTOP1,2 كج) 8 FSSTOP كج
ה DALM كج (INnx كج)

3-1-10. ◄ ! ↄ

ぇ ぅ ―15 ぢ ―ERROR STATUS ― ⁸ ⇔ √ ERROR STATUS ⇓ Δ ╢ ≤ ⁸ DRIVE STATUS1 PORT ―
ERROR ɦ ! ⁼ ⇓ ⇔ ╕ Δ ⁹
ERROR ɦ ! ⁼ = 1 ― │ ◖ ∀ fi ♪ ― ⅝ ⅞ ⌐ ⌂ ╡ ⁸ ▬ fi ㌔ ↄ ㎜ ℘ ▾ ⌐ ⌂ ╡ ╕ Δ ⁹
ERROR ɦ ! ⁼ = 1 │ ⇔ √ ERROR STATUS ⇓ ▾ ⁱ ▪ Δ ╢ ↓ ≤ ≠ ERROR ɦ ! ⁼ =0 ⌐ ╡ ╕ Δ ⁹

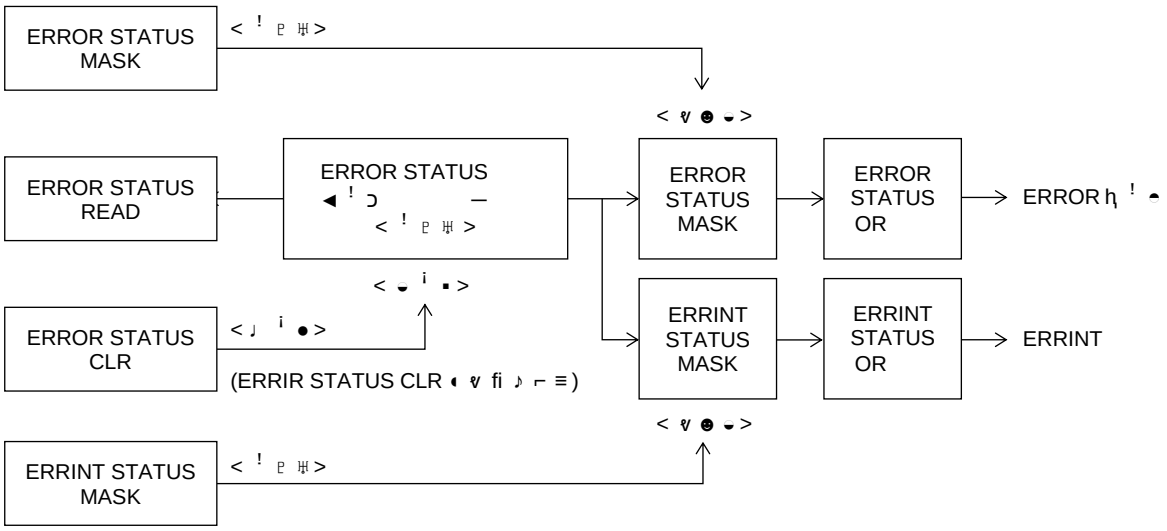
ɲ ◖ ∀ fi ♪ ゎ ( )⌐ ╟ ╡ ◖ ∀ fi ♪ ⇓ ゎ ⇔ ≡ ™ ╢ ≠ ERROR ɦ ! ⁼ = 1 ⌐ ⌂ ╢ ≤ ⁸
ゎ ◖ ∀ fi ♪ ⇓ ㎜ ≡ ▾ ⁱ ▪ ⇔ ≡ ▬ fi ㌔ ↄ ㎜ ℘ ▾ ⌐ ⌂ ╡ ╕ Δ ⁹
ɲ ERROR ɦ ! ⁼ = 1 ― │ COMREG FL = 1, COMREG EP = 1 ⌐ ⌂ ╡ ╕ Δ ⁹

Ẽ ERROR STATUS

ERROR STATUS	◄ ! ↄ ㎜
COMMAND ERROR	― ◖ ∀ fi ♪ ⇓ ⇔ √ ⁹
COMREG CLR ERROR	◖ ∀ fi ♪ ゎ ≠ ⇔ ≡ ™ ╢ ∟ ― ゎ ◖ ∀ fi ♪ ⇓ ▾ ⁱ ▪ ⇔ √ ⁹
INC INDEX ERROR	▪ ♪ ″ ☻ ― ○ ↄ ℘ ɦ ㎜ ↄ ≠ ⁸ INC INDEX ♪ ! ▬ Ꝗ ⇓ ゔ ⇔ √ ⁹
ABS INDEX ERROR	▪ ♪ ″ ☻ ○ ▶ fi ㌔ ― ○ ↄ ℘ ɦ ㎜ ↄ ≠ ⁸ ABS INDEX ♪ ! ▬ Ꝗ ⇓ ゔ ⇔ √ ⁹
INDEX CHANGE ERROR	⅛ ⅜ ⌂ INDEX CHANGE ぇ ⇓ ⇔ √ ⁹
CHANGE CLR ERROR	∟ ― INDEX CHANGE ぇ ⇓ ⌐ ⇔ √ ⁹
CPP STOP ERROR	♪ ! ▬ Ꝗ ― ω ▬ fi ― CPP STOP ≠ ♪ ! ▬ Ꝗ ⇓ ゔ ⇔ √ ⁹
EXT PULSE ERROR	Ꝛ ⁱ ☻ ⇓ ぢ ⌐ ⌂ Ꝛ ⁱ ☻ ⅜ ≠ ⅝ ⌂ ⅛ ∫ √ ⁹
FSEND ERROR	BUSY = 1 ― ≤ ⅝ ⌐ ⁸ DRIVE STATUS1 PORT ― FSEND = 1 ⇓ ⇔ √ ⁹
LSEND ERROR	BUSY = 1 ― ≤ ⅝ ⌐ ⁸ DRIVE STATUS1 PORT ― LSEND = 1 ⇓ ⇔ √ ⁹
SSEND ERROR	BUSY = 1 ― ≤ ⅝ ⌐ ⁸ DRIVE STATUS1 PORT ― SSEND = 1 ⇓ ⇔ √ ⁹
ADDRESS OVF ERROR	BUSY = 1 ― ≤ ⅝ ⌐ ⁸ DRIVE STATUS4 PORT ― ADDRESS OVF = 1 ⇓ ⇔ √ ⁹
PULSE OVF ERROR	DRIVE STATUS4 PORT ― PULSE OVF = 1 ⇓ ⇔ √ ⁹
DALM ERROR	DALM ぢ ― ▪ ▾ ♥ ▫ Ꝗ ⇓ ⇔ √ ⁹
FSSTOP ERROR	FSSTOPn ぢ │ FSSTOP ぢ ― ▪ ▾ ♥ ▫ Ꝗ ⇓ ⇔ √ ⁹

ɲ ERROR STATUS │ ⁸ ERROR STATUS READ ◖ ∀ fi ♪ ≠ ⅞ Δ ↓ ≤ ⅜ ≠ ⅝ ╕ Δ ⁹
ɲ ERROR STATUS │ ⁸ ERROR STATUS CLR ◖ ∀ fi ♪ ≠ ▾ ⁱ ▪ Δ ╢ ↓ ≤ ⅜ ≠ ⅝ ╕ Δ ⁹

Ẽ ◄ ! ↄ ≤ ERROR ―



Ẽ **ERROR ɧ ! ◓**

DRIVE STATUS1 PORT ─ERROR ɧ ! ◓ │ ⁸ ERROR STATUS ─ (OR) ≢ Δ ⁹

ERROR ɧ ! ◓ ⌐ Δ ╢ ERROR STATUS │ ⁸ ERROR STATUS MASK ◖ ∀ fi ♪ ≢ ☜ ⌐ ∀ ☻ ◒ Δ ╢ ↓ ≤ ⅜≢ ⅝╕ Δ ⁹

ᵐ ⇔ ⁸ ☜ ⠿ ─ERROR STATUS │ ∀ ☻ ◒ Δ ╢ ↓ ≤ │ ≢ ⅝╕ ∑ ╪ ⁹

ה COMMAND ERROR

ה COMREG CLR ERROR

ה INC INDEX ERROR

ה ABS INDEX ERROR

ה INDEX CHANGE ERROR

ⁱ ⠿ ⠏ ⠇ ─ │ ⁸ ☜ ⠿ ─ERROR STATUS ⅜ ∀ ☻ ◒ ↕ ╢ ≡ ™╕ Δ ⁹

ה LSEND ERROR

ה SSEND ERROR

ה ADDRESS OVF ERROR

ה PULSE OVF ERROR

ה DALM ERROR

ה FSSTOP ERROR

[ ]

☜ ╩ERROR = 1 ─ ⌐ ⇔ ≡ ™╢ ≢ ⁸

ﻭ ◖ ∀ fi ♪ ╩ ⇔ √ ♪ ! ▬Ϟ╩ ⇔ ≡ ⁸ ERROR = 1 ⅜ ⇔ √ │ ⁸

STATUS1 PORT ─DRVEND, LSEND, SSEND ɧ ! ◓ ⅜"1" ⌐ ⌂ ╟ ⌂ ™ ⅜№╡ ╕ Δ ⁹

ﻭ ◖ ∀ fi ♪ ╩ ⇔ √ ♪ ! ▬Ϟ╩ ⇔ ≡ ERROR = 1 ⅜ ⇔ √ │ ⁸

☜ ⠿ ─ ɧ ! ◓ ≢ ☜ ה ﻭ╩ ⇔ ≡ ↕ ∞ ⊃ ™ ⁹

ה ☜ │ ⁸ ERROR STATUS ─FSEND ERROR, LSEND ERROR, SSEND ERROR ≢ Δ ╢ ⁹

ה ♪ ! ▬Ϟ─ ﻭ │ ⁸ STATUS1 PORT ─BUSY=0 ≢ Δ ╢ ⁹

Ẽ **ERRINT**

ERRINT │ ERRINT ⌐ Δ ╢ ERROR STATUS ─ (OR) ≢ Δ ⁹

ERRINT ⌐ Δ ╢ ERROR STATUS │ ⁸ ERRINT STATUS MASK ◖ ∀ fi ♪ ≢ ☜ ⌐ ∀ ☻ ◒ Δ ╢ ↓ ≤ ⅜ ≢ ⅝╕ Δ ⁹

ERRINT │ ╡ ╥ ≤ ⇔ ≡ INT ☜ ⅛ ╟ ≢ ⅝╕ Δ ⁹

RESET ─ │ ⁸ ᵐᵐ ≡ ─ ERROR STATUS ⅜ ∀ ☻ ◒ ↕ ╢ ≡ ™╕ Δ ⁹

3-1-11. [illegible]

[illegible] PCI [illegible] (OR) [illegible] (INTA#) [illegible]
[illegible] (IRQ No.) [illegible] PCI Configuration Register [illegible]

[illegible]

[illegible]	[illegible]	[illegible]	[illegible]
INT	RDYINT	[illegible] RDYINT = 0 [illegible] 1	INT FACTOR CLR [illegible]
	STBY	STATUS1 PORT — STBY = 0 [illegible] 1	
	COMREG EP	STATUS1 PORT — COMREG EP = 0 [illegible] 1	
	nCOMREG FL	STATUS1 PORT — COMREG FL = 1 [illegible] 0	
	MAN	STATUS1 PORT — MAN = 0 [illegible] 1	
	DALM	STATUS2 PORT — DALM = 0 [illegible] 1	
	SS0	STATUS5 PORT — SS0 = 0 [illegible] 1	
	SS1	STATUS5 PORT — SS1 = 0 [illegible] 1	
	ADRINT	[illegible] — ADRINT = 0 [illegible] 1	
	CNTINT	[illegible] — CNTINT = 0 [illegible] 1	
	DFLINT	[illegible] — DFLINT = 0 [illegible] 1	
	ERRINT	ERRINT STATUS — ERRINT = 0 [illegible] 1	

[illegible] INT [illegible] (OR) [illegible]
[illegible] INT [illegible] OFF [illegible] ON [illegible] "1" [illegible]
INT [illegible]
[illegible] INT [illegible] INT FACTOR MASK [illegible]
[illegible] INT FACTOR READ [illegible]
[illegible] INT [illegible] STATUS3 PORT [illegible]
[illegible] INT FACTOR CLR [illegible]

[illegible]

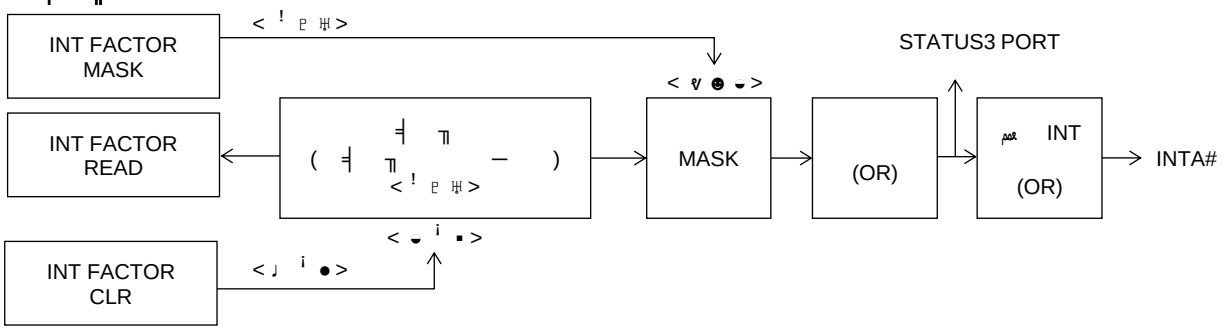
[illegible]	[illegible]	[illegible]
ADRINT		[illegible]
	COMP1	COMP1 — [illegible]
	COMP2	COMP2 — [illegible]
	COMP3	COMP3 — [illegible]
CNTINT		[illegible]
	COMP1	COMP1 — [illegible]
	COMP2	COMP2 — [illegible]
	COMP3	COMP3 — [illegible]
DFLINT		[illegible]
	COMP1	COMP1 — [illegible]
	COMP2	COMP2 — [illegible]
	COMP3	COMP3 — [illegible]

[illegible] ADRINT, CNTINT, DFLINT
[illegible]
[illegible] COUNTER INITIALIZE2 [illegible]

[illegible] RDYINT
[illegible] RDYINT [illegible] SPEC INITIALIZE2 [illegible]
RDYINT TYPE [illegible]

[illegible] ERROR=1 [illegible]
ERROR=1 [illegible] DRIVE [illegible] DRIVE 終了割り込み(RDYINT)が発生しない場合があります。

[illegible]



3-1-12.

(1)

STATUS PORT

(2)

SET DATA READ

(3)

MCC SPEED READ

(4)

ERROR STATUS READ

(5)

INT FACTOR READ

(6)

READ

(7)

LATCH DATA READ

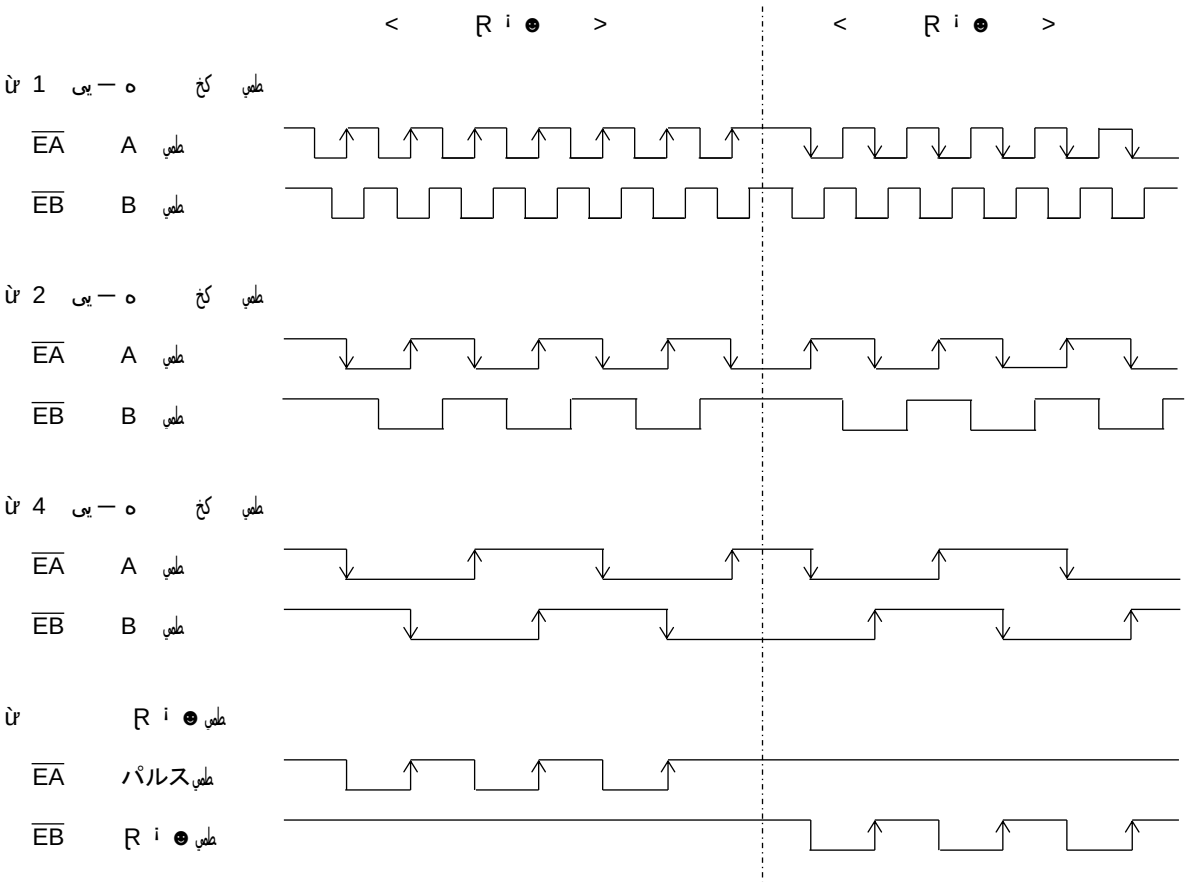
3-2. ○ ▶ fi ث

3-2-1. ◀ fi ◖ ⊃ ♭ R i ☻ طص

EA, EB كخ ⌐ R i ☻ كخ ╩طص ⇔ ≡ ○ ▶ fi ث ≠ ○ ▶ fi ♩ ≠ ⅝ ╕ Δ ⁹

○ ▶ fi ♩ │ ○ ▶ fi ث ⌐ ⌐ ⅞ — 4 — ⅛ ╟ ≠ ⅝ ╕ Δ ⁹

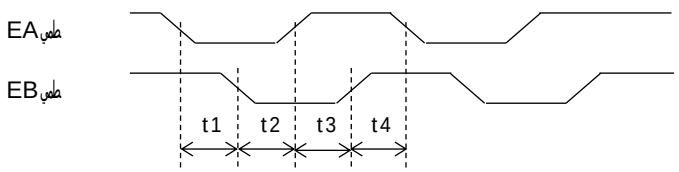
β 6 ⁸ 12 │ ◀ fi ◖ ⊃ ♭ R i ☻ طص │ № ╡ ╕ ∑ ╪ ⁹



ה │ طص R i ☻ — ○ ▶ fi ♩ ◀ ℘ ☺ ≠ Δ ⁹

ה ○ ▶ fi ث — R i ☻ ○ ▶ fi ♩ │ COUNTER INITIALIZE1 ◖ ∀ fi ♪ ≠ ™ ╕ Δ ⁹

[٥ كخ طص ث ▬ ⊢ fi ◓]



ה ▪ ♪ ™ ☻ ○ ▶ fi ث —

2 يى — ≤ ⅝ t1,t2,t3,t4 50ns

t1 t2" 200ns

t3 t4" 200ns

4 يى — ≤ ⅝ t1,t2,t3,t4" 200ns

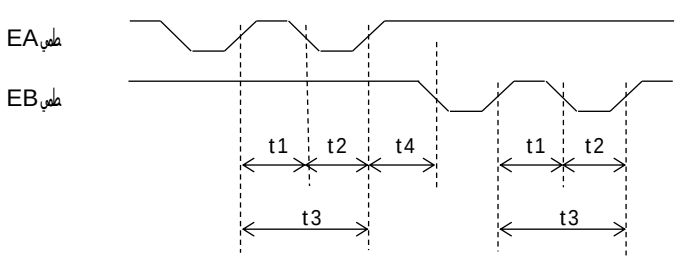
ה / — ث — ○ ▶ fi ث —

2 يى — ≤ ⅝ t1 t2＞50ns

t3 t4＞50ns

4 يى — ≤ ⅝ t1,t2,t3,t4 50ns

[R i ☻ طص ث ▬ ⊢ fi ◓]



ה ▪ ♪ ™ ☻ ○ ▶ fi ث —

t1,t2,t4 50ns

t3" 200ns

ה / — ث — ○ ▶ fi ث —

t1,t2,t4 50ns

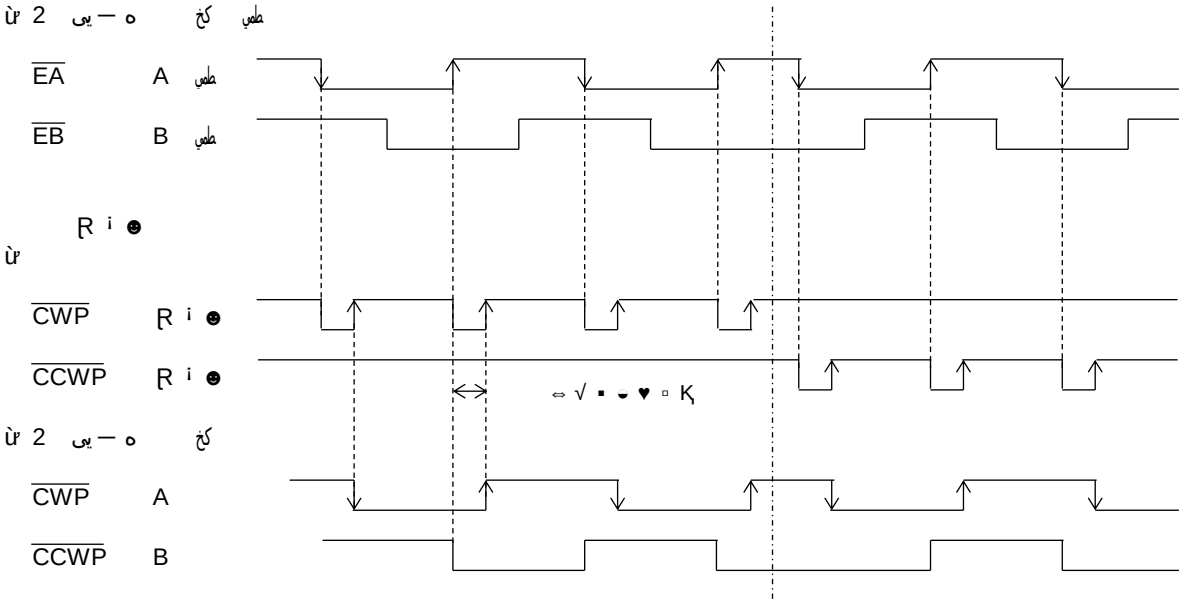
t3 100ns

3-2-2. R i ●

▪ ♪ ☻ ○ ▶ fi ▬ — ○ ▶ fi ♩ R i ☻ ⊥ 8 R i ☻ 9 ⌐ Δ ╫ ≤ 8 EA, EB ⌐ ↕ ╫ ╫ R i ☻ —
○ ▶ fi ♩ ▬ ⊢ fi ⊥ ⇔ √ ▪ ♥ ▫ K — R i ☻ ⌐ ⇔ ≡ 8 CWP, CCWP ⅛ ⇔ ┐ Δ 9

β 6 8 12 ◀ fi ◖ ↄ ♭ R i ☻ ⌐ ╫ ╫ R i ☻ № ┤ ┐ ∑ ╪ 9

R i ☻
2 — o



ה ▪ ♪ ☻ ○ ▶ fi ▬ — ○ ▶ fi ♩ R i ☻ — ADDRESS COUNTER INITIALIZE1 ◖ fi ♪ ≠ ™ ┐ Δ 9

ה ⇔ √ ▪ ♥ ▫ K — 2 — ⌐ 8 — R i ☻ — ○ ▶ fi ♩ ▬ ⊢ fi ⅜ ⇔ √ 8
⌂ R i ☻ ⅜ ≠ ⅝ ┐ ∑ ╪ 9 ↓ — 8 ◀ ! ↄ ⌐ ⌂ ┤ ┐ Δ 9
(ERROR STATUS —EXT PULSE ERROR = 1 ⌐ ⇔ ┐ Δ 9)

ה LIMIT ⊥ Δ ╫ ≤ 8 — R i ☻ ⊥ ⇔ ≡ 8 STBY ⌐ ⇔ ┐ Δ 9

ה 8 ┐ √ DRIVE STATUS1 PORT —ERROR = 1 ⊥ Δ ╫ ≤ 8 R i ☻
⊥ ⇔ ≡ 8 R i ☻ ⊥ ⌐ ⇔ ┐ Δ 9

ה R i ☻ ⅜ ≠ ◖ fi ♪ 8 ☻ ▬ ↄ ♩ 8 DEND, DRST —
ↄ W ⅜ ≠ Δ 9
┐ √ 8 DRIVE STATUS1, 2 PORT — — ! ⅜ ≠ Δ 9
DRIVE STATUS1 ה ה i BUSY 8 STBY 8 DRIVE 8 ERROR 8 LSEND 8 SSEND
FSEND 8 PAUSE 8 COMREG EP 8 COMREG FL
DRIVE STATUS2 ה ה i DEND BUSY

ה — 8 ○ ▶ fi ♩ ▬ ⊢ fi — ≠ R i ☻ — ⅜ Δ ╫ √ 8
— ≤ ▪ ♥ ▫ K — ⅜ ◀ ☺ ⅜ ⌐ ⌂ ┤ ┐ Δ 9

ה 2 — o — 8 EXT PULSE TYPE ≠ ⇔ √ ▪ ♥ ▫ K ⅜ 8 — o
⌐ ⌂ ┤ ┐ Δ 9

Ẽ Ꞁ i ● ⊂ ― ● ♥ ⊃ ⊂ ● ≤ ⊂

Ꞁ i ☻ ⅜ ▪ ▾ ♥ ▫ Ꞁ ⊂ i ⊂ ⌐ 8 Ꞁ i ☻ ― ⊂ ⊂ √ | 8

⊂ ― Ꞁ i ☻ ― ▪ ▾ ♥ ▫ Ꞁ ⊂ ⊂ ⇔ √ ⌐ Ꞁ i ☻ ⊂ ⊂ ⇔ ╕ Δ 9

Ꞁ i ☻ ⊂ ― ☻ ♥ ⊃ ⊂ ● ɦ ! ⊂ | 8 ⊂ ― ╟ e ⌐ ⇔ ╕ Δ 9

ỳ Ꞁ i ☻ ― ≤ ǧ

ה EXT PULSE = 0 8 BUSY = 0 8 ERROR = 0 ― ≤ ⅝ ⌐ 8

COUNT PULSE SEL ― 8 01, 10, 11 9 ث ― Ꞁ i ☻ 8 Ꞁ i ☻ ⊂ ⊂ Δ ╢ ≤ 8

EXT PULSE = 1 8 BUSY = 1 8 STBY = 1 8 DRIVE = 0 ⌐ ⌂ ╡ ╕ Δ 9

ה EXT PULSE = 0 8 BUSY = 1 ― ≤ ⅝ ⌐ 8 COUNT PULSE SEL ⊂ 8 01, 10, 11 9 ⌐ Δ ╢ ≤ 8

―BUSY = 1 ǧ ⌐ 8 EXT PULSE = 1 8 BUSY = 1 ⌐ ⌂ ╡ ╕ Δ 9

ה EXT PULSE = 1 8 STBY = 1 ― | 8 Δ ╢ Ꞁ i ☻ ⊂ ― ⊂ ⊂ ― ≢ Δ 9

ה Δ ╢ Ꞁ i ☻ ⊂ ⊂ Δ ╢ ≤ 8 Ꞁ i ☻ ⊂ ⇔ ≡ 8

EXT PULSE = 1 8 BUSY = 1 8 STBY = 0 8 DRIVE = 1 ⌐ ⌂ ╡ ╕ Δ 9

EXT PULSE = 1 8 DRIVE = 1 ― | 8 Ꞁ i ☻ ⊂ ― ≢ Δ 9

ה EXT PULSE = 1 ― ≤ ⅝ ⌐ 8 COUNT PULSE SEL ― 8 00 9 ― Ꞁ i ☻ ⊂ Δ ╢ ≤ 8

EXT PULSE = 0 8 BUSY = 0 ⌐ ⌂ ╡ ╕ Δ 9

EXT PULSE = 0 8 BUSY = 0 ― | 8 Ꞁ i ☻ ⊂ ǧ ⇔ √ ≢ Δ 9

ה STBY = 1 ╕ √ | DRIVE = 1 ― ≤ ⅝ ⌐ COUNT PULSE SEL ― 8 00 9 ⊂ ⇔ √ | 8

DEND/PO ⊂ ―<◘ ⊃ W ⊂ > ⊂ ⇔ ╕ Δ 9

DEND/PO ⊂ ―<◘ ⊃ W ⊂ > ⊂ | 8 BUSY = 1 ⌐ ⌂ ╡ ╕ Δ 9

ỳ LIMIT ⊂ ⌐ ╟ ╢ Ꞁ i ☻ ― ⊂

ה EXT PULSE = 1 ― ≤ ⅝ ⌐ 8 LIMIT ⊂ ⊂ ⊂ Δ ╢ ≤ 8 Ꞁ i ☻ ⊂ ⊂ ⇔ ≡ 8

EXT PULSE = 1 8 BUSY = 1 8 STBY = 1 8 DRIVE = 0 ⌐ ⌂ ╡ ╕ Δ 9

EXT PULSE = 1 8 STBY = 1 ― | 8 Δ ╢ Ꞁ i ☻ ⊂ ― ⊂ ⊂ ― ≢ Δ 9

LIMIT ⊂ ⊂ ⅜ ▪ ▾ ♥ ▫ Ꞁ ≢ ⊂ 8 LIMIT ⊂ ⊂ ≤ ― Ꞁ i ☻ ⅜ ≢ ⅝ ╕ Δ 9

ה LSEND ɦ ! ⊂ ⊂ ⇔ ╕ Δ 9 DEND/PO ⊂ ╕ √ | DRST ⊂ ―<◘ ⊃ W ⊂ > ⊂ ⇔ ╕ Δ 9

DEND/PO ⊂ ╕ √ | DRST ⊂ ―<◘ ⊃ W ⊂ > ⊂ | 8 STBY = 0 ⌐ ⌂ ╡ ╕ Δ 9

ה LIMIT ⊂ ⊂ | 8 DRIVE = 0 ǵ 1 ― ≤ DRIVE = 1 8 DEND BUSY = 1 ― ≤ ⅝ ⌐ ⇔ ╕ Δ 9

LIMIT ⊂ ⊂ | 8 DRIVE = 0 ǵ 1 ― ≤ DRIVE = 1 8 DEND BUSY = 1 ― ≤ ⅝ ⌐ ⇔ ╕ Δ 9

ỳ / ― ث ― ⊂ ⌐ ╟ ╢ Ꞁ i ☻ ―

ה EXT PULSE = 1 ― ≤ ⅝ ⌐ 8 ⊂ ⊂ 8 ⊂ ⊂ ╕ √ | ERROR = 1 ⊂ Δ ╢ ≤ 8

Ꞁ i ☻ ⊂ ⊂ ⇔ ≡ 8 EXT PULSE = 1 8 BUSY = 1 8 STBY = 0 8 DRIVE = 0 ⌐ ⌂ ╡ ╕ Δ 9

EXT PULSE = 1 8 BUSY = 1 8 STBY = 0 8 DRIVE = 0 ― | 8 Ꞁ i ☻ ⅜ ― ≢ Δ 9

ה SSEND 8 FSEND ɦ ! ⊂ ⊂ ⇔ ╕ Δ 9

DEND/PO ⊂ ╕ √ | DRST ⊂ ―<◘ ⊃ W ⊂ > ⊂ ⇔ ╕ Δ 9

ה ⊂ ⊂ | 8 STBY = 1 ╕ √ | DRIVE = 1 ― ≤ ⅝ ⌐ ⇔ ╕ Δ 9

⊂ ⊂ ⅔ ╟ ┘ ERROR = 1 | 8 BUSY = 1 ― ≤ ⅝ ⌐ ⇔ ╕ Δ 9

ה SSEND = 1 8 FSEND = 1 ╕ √ | ERROR = 1 ≢ Ꞁ i ☻ ⊂ ⊂ ⇔ √ | 8

COUNT PULSE SEL ⊂ 8 00 9 ⌐ ⇔ ≡ 8 Ꞁ i ☻ ⊂ ǧ ↕ ∑ ≡ ⊃ ∞ ↕ ™ 9

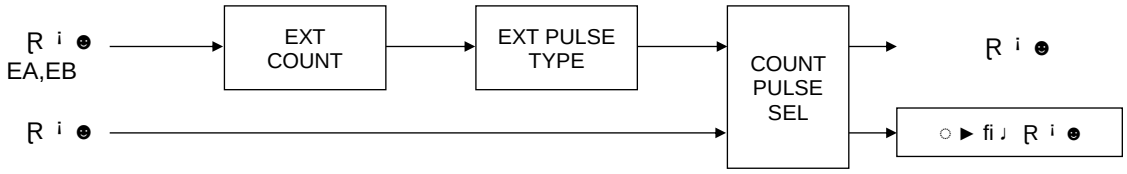
3-2-3. ▪ ♪ ● ○ ▶ fi

▪ ♪ ● ○ ▶ fi │ CWP, CCWP ⌐ Δ ♪ ▬ ▶ fi ⇔ ≡ ▪ ♪ ● Δ
32 ▶ fi ≠ Δ

+(CW) ▶ fi ▪ -(CCW) ▶ fi ▶ fi ⇔ Δ

○ ▶ fi │ -2,147,483,647 +2,147,483,647 H'8000_0001 H'7FFF_FFFF ≠ Δ
│ 2 ⌐ Δ

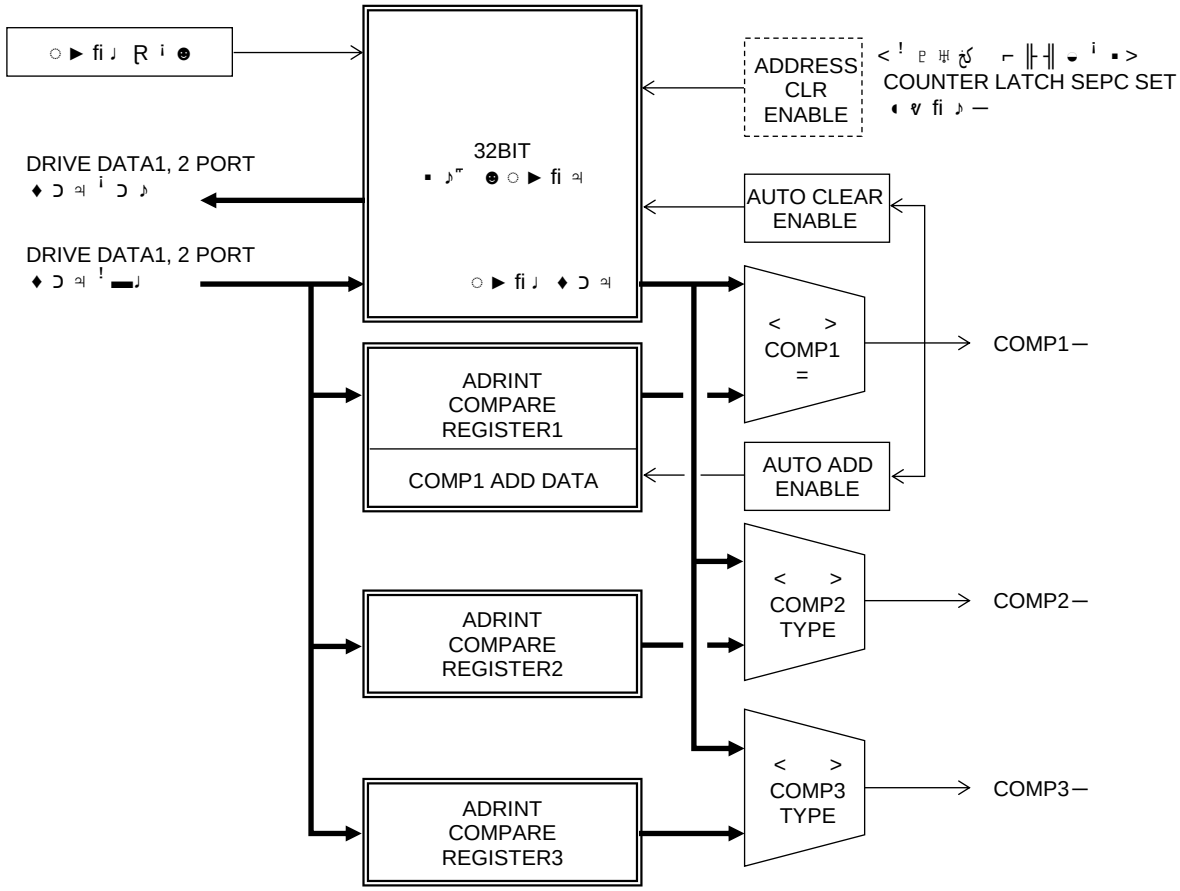
Ẽ ▪ ♪ ● ○ ▶ fi



▪ ♪ ● ○ ▶ fi │ ADDRESS COUNTER INITIALIZE1 ≠ Δ

▪ ♪ ● ○ ▶ fi ≠ ⇔ √ CWP, CCWP ⅛ Δ │
5-2-2. ⇔ ≡ ™ ∞ ↕

Ẽ ▪ ♪ ● ○ ▶ fi ≤ fi ♪ ⌐ ▶ fi



▪ ♪ ● ○ ▶ fi ≤ fi ♪ ⌐ ▶ fi │ ADDRESS COUNTER INITIALIZE1 ≠ Δ

▪ ♪ ● ○ ▶ fi │ ADDRESS COUNTER PRESET ≠ Δ

▪ ♪ ● ○ ▶ fi │ ADDRESS COUNTER READ ≠ ∑ Δ

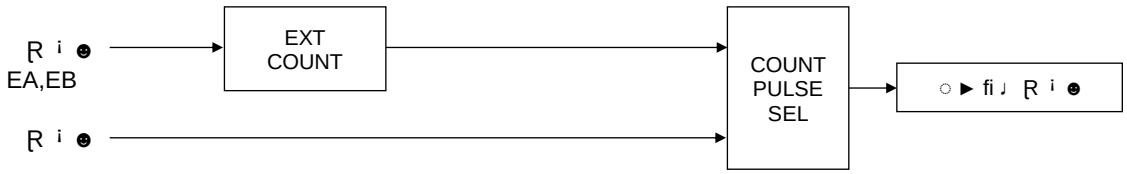
3-2-4. Ri●○▶fi㉦

Ri●○▶fi㉦│⁸◀fi◖Ↄ♭ Ri●╩○▶fi♩ ⇔≡⁸ ₀ ╩ Δ╫32 ⅈ℘♩ ─○▶fi㉦≠Δ⁹
♪!▬Ki Ri●─○▶fi♩ ╩≠⅝╕Δ⁹

ㅁ +(CW) ─Ri●≠○▶fi♩ ▪℘ki⁸-(CCW) ─Ri●≠○▶fi♩ ♭▶fi⇔╕Δ⁹

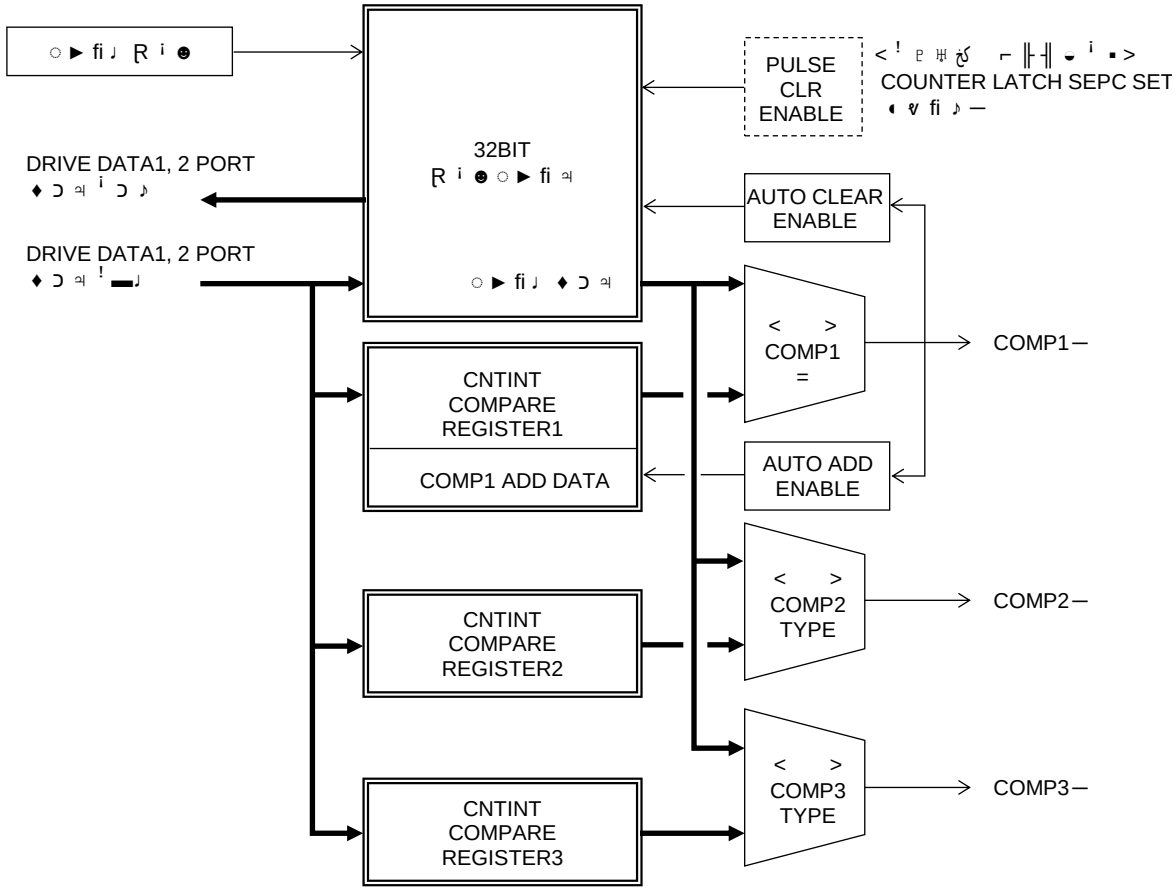
ㅁ ○▶fi㉦─ │⁸-2,147,483,647 +2,147,483,647 H'8000_0001 H'7FFF_FFFF ≠Δ⁹
─ │⁸2─ ⌐⌂╡╕Δ⁹

Ẽ Ri●○▶fi㉦─Ri●



ㅁ Ri●○▶fi㉦─Ri● │PULSE COUNTER INITIALIZE1◖∀fi♪≠ ⇔╕Δ⁹

Ẽ Ri●○▶fi㉦≤◖fiR™ Ↄ㉦─



ㅁ Ri●○▶fi㉦≤◖fiR™ Ↄ㉦─ │PULSE COUNTER INITIALIZE1◖∀fi♪≠ ⇔╕Δ⁹

ㅁ Ri●○▶fi㉦─ ⌠│PULSE COUNTER PRESET◖∀fi♪≠ ⇔╕Δ⁹

ㅁ Ri●○▶fi㉦─ ⌠│PULSE COUNTER READ◖∀fi♪≠ ╥ ∑╕Δ⁹

3-2-5. R i ● ئجـ ○ ▶ fi ص

R i ● ئجـ ○ ▶ fi ص | ◀ fi ◖ ⊃ ♭ طص R i ● ≤ ♪ ! ▬ Ką R i ● ╩ ○ ▶ fi ♩ ⇔ ≡ ⁸ R i ● — ئجـ ╩ Δ ╫
16 ⅟ ℘ ♩ — ○ ▶ fi ص ≠ Δ ⁹
◦ ☻ ♥ ⊢ ▾ ℘ ▾ (20MHz) — ╥ ╩ ○ ▶ fi ♩ ⇔ ≡ ص ▬ ∀ ≤ ⇔ ≡ ئو Δ ╫ ↓ ≤ ╚ ≠ ⅝ ╕ Δ ⁹

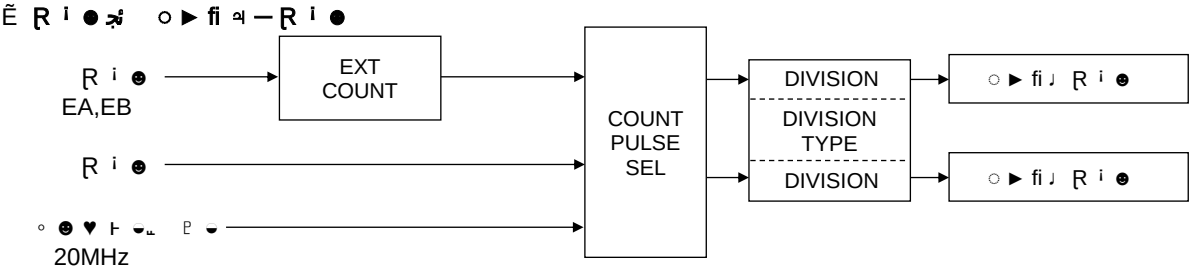
β 6 ⁸ 12 | ◀ fi ◖ ⊃ ♭ R i ● طص ⌐ ╫ ئجـ ○ ▶ fi ص | №╡ ╕ ∑ ╪ ⁹

ה ◀ fi ◖ ⊃ ♭ طص R i ● | +(CW) — R i ● ≠ ○ ▶ fi ♩ ▪ ℘ ką ⁸ -(CCW) — R i ● ≠
○ ▶ fi ♩ ♭ ▶ fi ⇔ ╕ Δ ⁹

ה ♪ ! ▬ Ką R i ● | -(CCW) — R i ● ≠ ○ ▶ fi ♩ ▪ ℘ ką ⁸ +(CW) — R i ● ≠
○ ▶ fi ♩ ♭ ▶ fi ⇔ ╕ Δ ⁹

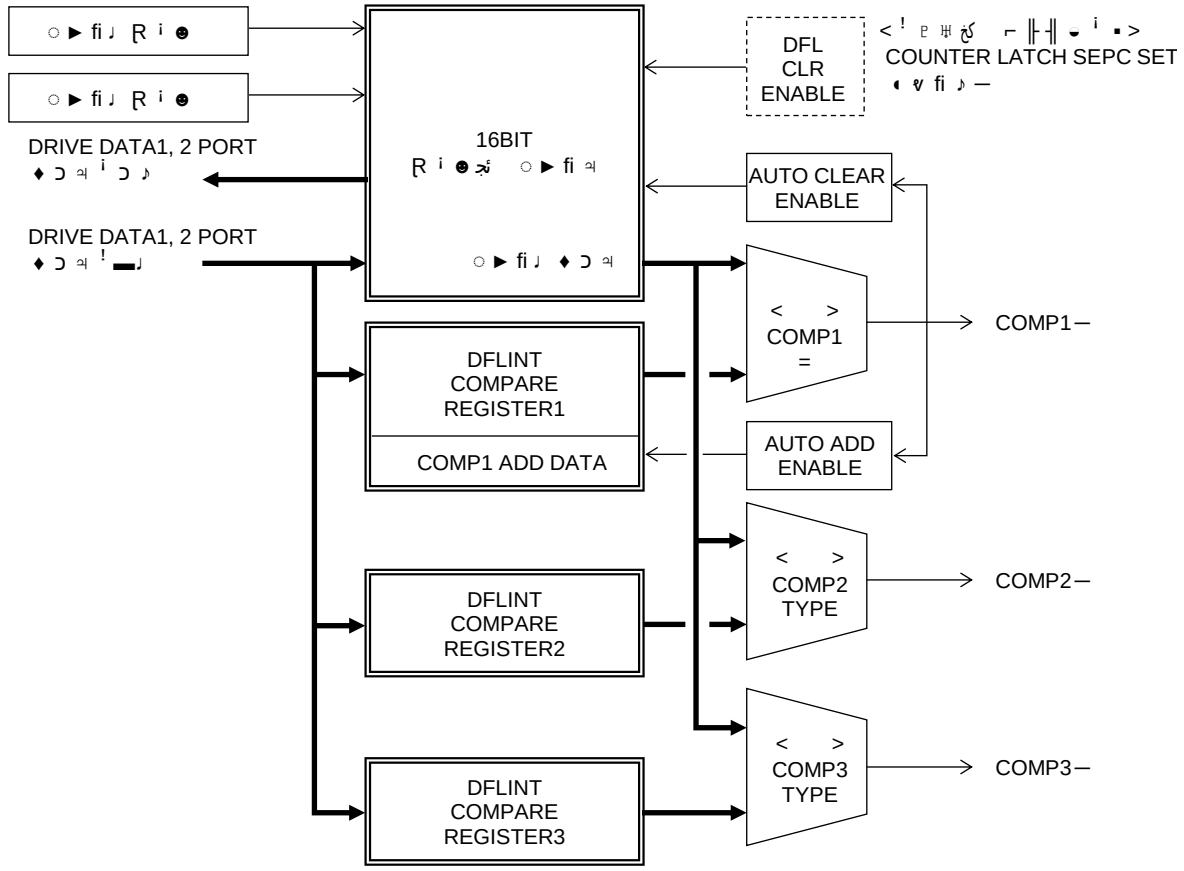
ה ○ ▶ fi ص — | ⁸ -32,767 +32,767 H'8001 H'7FFF ≠ Δ ⁹
— | ⁸ 2 — ⌐ ⌂ ╡ ╕ Δ ⁹

Ẽ R i ● ئجـ ○ ▶ fi ص — R i ●



ה R i ● ئجـ ○ ▶ fi ص — R i ● | DFL COUNTER INITIALIZE1 ◖ ∀ fi ♪ ≠ ⇔ ╕ Δ ⁹

Ẽ R i ● ○ ▶ fi ص ≤ ◖ fi R⁻ ⊃ ص —



ה R i ● ئجـ ○ ▶ fi ص ≤ ◖ fi R⁻ ⊃ ص — | DFL COUNTER INITIALIZE1 ◖ ∀ fi ♪ ≠ ⇔ ╕ Δ ⁹

ה R i ● ئجـ ○ ▶ fi ص — ثم | DFL COUNTER PRESET ◖ ∀ fi ♪ ≠ ⇔ ╕ Δ ⁹

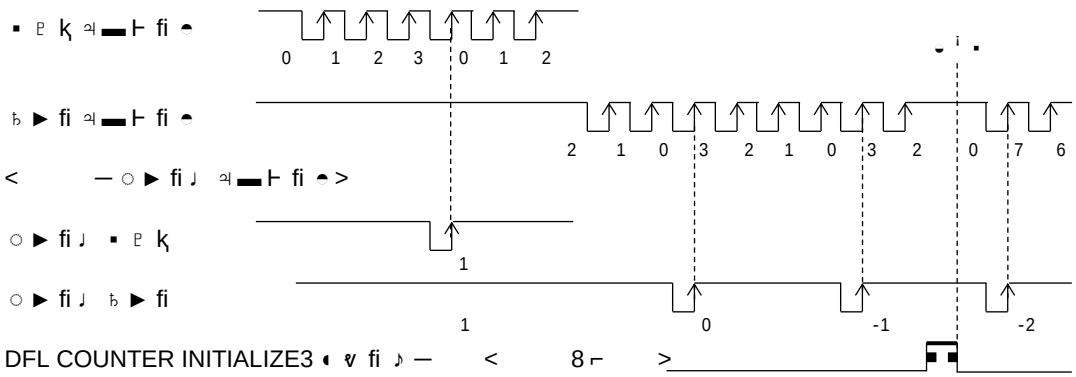
ה R i ● ئجـ ○ ▶ fi ص — ثم | DFL COUNTER READ ◖ ∀ fi ♪ ≠ ╥ ∑ ╕ Δ ⁹

Ẽ

Ꞁ ⁱ ☻ ئجـ ○ ► fi ㄹ≠ │COUNT PULSE SEL ≠ ⇔ √ ○ ► fi ♩ Ꞁ ⁱ ☻ — ○ ► fi ♩ ㄹ▬ ⊢ fi ◓ ╩ Δ ╢
↓ ≤ ⅜≠ ⅝ ╕ Δ ⁹
○ ► fi ㄹ │ ⇔ √ ○ ► fi ♩ ㄹ▬ ⊢ fi ◓≠ ○ ► fi ♩ ▪ ℘ ϟ ⁸ ╕ √ │ ○ ► fi ♩ ♭ ► fi ⇔ ╕ Δ ⁹

ùᵛ ○ ► fi ♩ ㄹ▬ ⊢ fi ◓ ╩4 Δ ╢

<○ ► fi ♩ Ꞁ ⁱ ☻ — ظص >



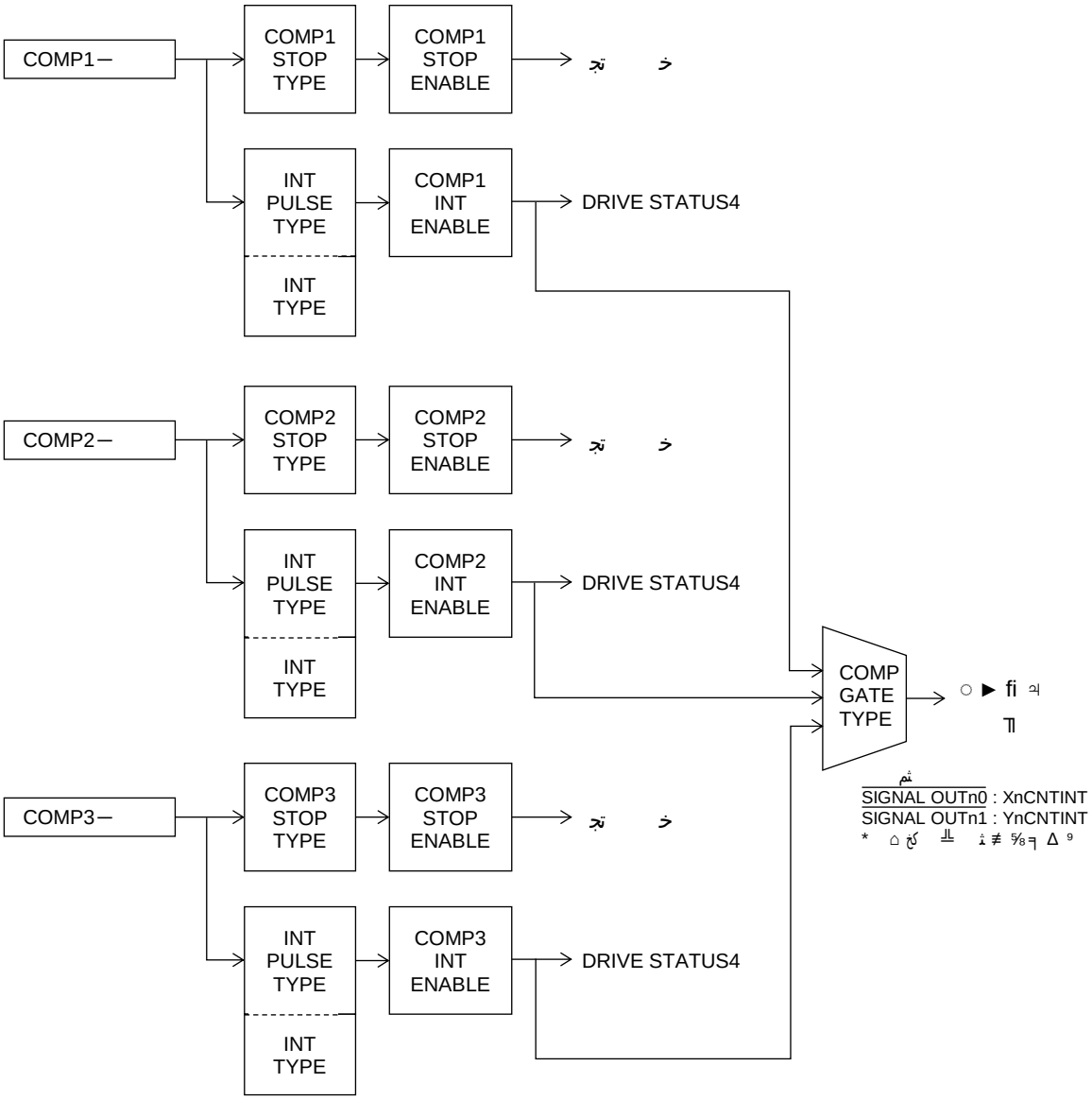
ה Ꞁ ⁱ ☻ ئجـ ○ ► fi ㄹ — │DFL COUNTER INITIALIZE3 ◖ ∀ fi ♪ ≠ ⇔ ╕ Δ ⁹

ה DFL COUNTER INITIALIZE3 ◖ ∀ fi ♪ ╩ Δ ╢ ≤ ـجـ — ○ ► fi ♩ ثم ╩ ◓ ⁱ ▪ ⇔ ╕ Δ ⁹

3-2-6. ◖ fi Ꞧ ↄ ㅋ

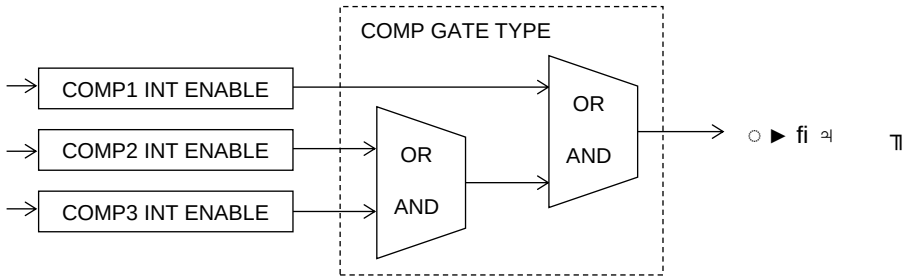
○ ▶ fi ㅋ ⌐ │ 3 ㅋ — ◖ fi Ꞧ ↄ ㅋ ⅜ ㅈ ™ ≡ ⅔ ㅓ ⁸ ○ ▶ fi ㅋ ㅁ ≤ COMPARE REGISTER1, 2, 3 — ㅁ ╩
⇔ ≡ ⁸ ㅅ ⅜ ㅍ Δ ╢ ≤ ㅌ ㅎ ㄱ √ │ ○ ▶ fi ㅋ ㅓ ╗ ╩ ⇔ ㄱ Δ ⁹
○ ▶ fi ㅋ ㅓ ╗ — │ STATUS4 PORT ≠ ≠ ⅝ ㄱ Δ ⁹

Ẽ ◖ fi Ꞧ ↄ ㅋ —



ㅎ ◖ fi Ꞧ ↄ ㅋ │ ○ ▶ fi ㅋ COUNTER INITIALIZE1, 2 ◖ ∀ fi ♪ ≠ ⇔ ㄱ Δ ⁹

Ẽ **COMP GATE TYPE —**



ㅎ COMP GATE TYPE │ ○ ▶ fi ㅋ COUNTER INITIALIZE1 ◖ ∀ fi ♪ ≠ ⇔ ㄱ Δ ⁹

Ẽ ◖ fi ʀ̃ ↄ ㆍ ث ≤ ◒ ⁱ ▪ **(INT TYPE)**

◖ fi ʀ̃ ↄ ㆍ — ث ⅛ ≠ ⅝ ╗ Δ ⁹

COMP1, 2, 3 — پ — ث	◒ ⁱ ▪ ش
پ ⅝ ≤ — پ ث ⅜ ش DRIVE STATUS4 PORT — ⁱ ◒ ≠ ♪ ↄ ⁱ ▪	Δ ╢ ≡ ⇔ ㆍ ! ⁱ ʐ ⅝
پ DRIVE STATUS4 PORT — ⁱ ◒ ≠ ♪ ↄ ⁱ ▪	Δ ╢ ≡ ⇔ ㆍ ! ☺ ◀
پ ش — پ ث — ⁱ ◒ ≠ ▪	Δ ╢ ≠ ↄ ⁱ ☻ ╗ ╗ — /
پ INT FACTOR CLR ◖ ∀ fi ♪ — ADR(CNT,DFL)INT INT CLR = 1 — ⁱ ◒ ≠ ▪	Δ ╢ ≡ ⇔ ㆍ ! ☺ ◀

ה ◖ fi ʀ̃ ↄ ㆍ ث ≤ ◒ ⁱ ▪ (INT TYPE) | ○ ▶ fi ㆍ COUNTER INITIALIZE1 ◖ ∀ fi ♪ ≠ ⇔ ╗ Δ ⁹

ה ʐ ⁱ ! ㆍ — | ⁸ ش ⅜ پ ⇔ ≡ ™ ╢ | ◒ ⁱ ▪ ≠ ⅝ ╗ ∑ ╪ ⁹

ה ☻ ⁱ ↄ — | ⁸ ⅜ ≠ ⅝ ╗ Δ ⁹

Ẽ ○ ↄ ↓ ◒ ⁱ ▪

COMP1 — پ ≤ ⌐ ○ ▶ fi ㆍ — ثم "0" ⌐ ◒ ⁱ ▪ ⇔ ╗ Δ ⁹

ה ○ ↄ ↓ ◒ ⁱ ▪ | ○ ▶ fi ㆍ COUNTER INITIALIZE1 ◖ ∀ fi ♪ ≠ ⇔ ╗ Δ ⁹

Ẽ

COMP1 — پ ≤ ⌐ ⁸ COMP1 ADD ♦ ↄ ㆍ ⌐ ↕ ╢ ≡ ™ ╢ ثم COMPARE REGISTER1 ⌐
⇔ ≡ ⁸ COMPARE REGISTER1 ⇔ ╗ Δ ⁹

COMPARE REGISTER1 <= COMPARE REGISTER1 COMP1 ADD ♦ ↄ ㆍ

ה | ○ ▶ fi ㆍ COUNTER INITIALIZE1 ◖ ∀ fi ♪ ≠ ⇔ ╗ Δ ⁹

【注意】
自動加算機能を使用する場合は、オートクリア機能を無効（AUTO CLEAR ENABLE = 0）にしてください。

* ○ ↄ ↓ ◒ ⁱ ▪ AUTO CLEAR ENABLE = 1 ⌐ Δ ╢ ≤ ⁸ | ⌐ ⌂ ╡ ╗ Δ ⁹

3-2-7. ╹ fi ◓ ○ ▶ fi ⌓

▪ ♪ᵐ ☻ ○ ▶ fi ⌓ ⁸ ⅔ ╟┘ R ⁱ ☻ ○ ▶ fi ⌓ │ ⁸ ○ ▶ fi ♩ ثم ╩ Δ ╢ ≤ ⁸ ثم ╩ ○ ▶ fi ⌓ — ثم ≤ ⇔ ≡
ⁱ fi ◓ ○ ▶ fi ♩ ⇔ ╕ Δ ⁹
— ▪ ♪ᵐ ☻ ⌐ سخ ≢ Δ ⁹

ὺ ○ ▶ fi ♩ 1,999 — (2,000 ○ ▶ fi ♩ ≢ 1)
ה ○ ▶ fi ♩ 1,999 — 2,000 ○ ▶ fi ♩ ≢

STATUS4 PORT—OVF = 1
ὶ
— ○ ▶ fi ♩ 0 ή ⁷ 1 ή é ή 999 ή 1000 ή 1001 ή é ή 1999 ή 0
— ○ ▶ fi ♩ 0 ή 1999 ή é ή 1001 ή 1000 ή 999 ή é ή 1 ή 0
ħ
STATUS4 PORT—OVF = 1

ה ○ ▶ fi ♩ 2,000 — 2,001 ○ ▶ fi ♩ ≢

STATUS4 PORT—OVF = 1
ὶ
— ○ ▶ fi ♩ 0 ή 1 ή é ή 1000 ή 1001 ή é ή 2000 ή 0
— ○ ▶ fi ♩ 0 ή 2000 ή é ή 1001 ή 1000 ή é ή 1 ή 0
ħ
STATUS4 PORT—OVF = 1

ה R ⁱ ☻ ئج ○ ▶ fi ⌓ ⌐ │ ⁱ fi ◓ ○ ▶ fi ⌓ │№╡ ╕ ∑ ╪ ⁹

ה ○ ▶ fi ♩ ⅜ ثم — 1/2 ⌐ Δ ╢ ≤ ⁸ STATUS4 PORT — ○ ▶ fi ⌓ — OVF=1 ⌐ ⌂ ╡ ╕ Δ ⁹

ה ○ ▶ fi ♩ ثم │ ○ ▶ fi ⌓ — COUNTER MAX COUNT SET ◖ ∀ fi ♪ ≢ ⇔ ╕ Δ ⁹

ה ▪ ♪ᵐ ☻ ○ ▶ fi ⌓ ⌐ ○ ▶ fi ♩ ثم ╩ ⇔ ≡ ⁸ ABS INDEX ♪ ! ▬ Ķ ≢ ○ ▶ fi ♩ —ثخ ثم
▪ ♪ᵐ ☻ ╩ ⇔ √ ⁸ ▪ ♪ᵐ ☻ ○ ▶ fi ⌓ ⅜ OVF ⇔ √ ≢ ♪ ! ▬ Ķ │ تج ⇔ ⁸ ◀ ! ↄ ≤ ⌂ ╡ ╕ Δ ⁹

3-2-8. ○ ▶ fi ♩ ♦ ↄ ㅋ — ! ㄹ ㅃ ㅎ ◒ ' ▪

Ẽ ○ ▶ fi ㅋ — ! ㄹ ㅃ

⇔ √ ! ㄹ ㅃ ㅋ ▬ ⊢ fi ◓ — ▪ ◒ ♥ ▫ Ḳ ◀ ㄹ ☺ ≠ ⁸ ○ ▶ fi ㅋ — ○ ▶ fi ♩ ♦ ↄ ㅋ ╩ ! ㄹ ㅃ ⇔ ╕ Δ ⁹

! ㄹ ㅃ ⇔ √ ♦ ↄ ㅋ │ — ! ㄹ ㅃ ㅋ ▬ ⊢ fi ◓ — ▪ ◒ ♥ ▫ Ḳ ◀ ㄹ ☺ ╩ Δ ╢ ╕ ≠ ㅋ ⇔ ╕ Δ ⁹

! ㄹ ㅃ ⇔ √ ♦ ↄ ㅋ │ ○ ▶ fi ㅋ LATCH DATA READ ◖ ∿ fi ♪ ≠ ㅟ Δ ↓ ≤ ⅜ ≠ ⅝ ╕ Δ ⁹

ὺ ≠ ⅝ ╢ ! ㄹ ㅃ ㅋ ▬ ⊢ fi ◓

! ㄹ ㅃ ㅋ ▬ ⊢ fi ◓ < ◀ ㄹ ☺ >
ㅎ ○ ▶ fi ㅋ LATCH DATA READ ◖ ∿ fi ♪ — ≠ ! ㄹ ㅃ Δ ╢
ㅎ ⁑ fi ◘ ↄ̌ SS0 = 0 ́ 1 ≠ ! ㄹ ㅃ Δ ╢
ㅎ ORIGIN SPEC SET ◖ ∿ fi ♪ — ORG ↄ̌ — ◀ ㄹ ☺ ≠ ! ㄹ ㅃ Δ ╢

ㅎ ○ ▶ fi ㅋ — ! ㄹ ㅃ ㅋ ▬ ⊢ fi ◓ │ COUNT LATCH SPEC SET ◖ ∿ fi ♪ ≠ ⇔ ╕ Δ ⁹

Ẽ ○ ▶ fi ㅋ — ◒ ⁱ ▪

○ ▶ fi ♩ ♦ ↄ ㅋ — ! ㄹ ㅃ ≤ ⌐ ○ ▶ fi ♩ ♦ ↄ ㅋ ╩ "0" ⌐ ◒ ⁱ ▪ ⇔ ╕ Δ ⁹

○ ▶ fi ㅋ — ○ ▶ fi ♩ ㅋ ▬ ⊢ fi ◓ ⅜ ⌐ ⇔ √ │ ◒ ⁱ ▪ ㅋ ▬ ⊢ fi ◓ ≤ ◒ ⁱ ▪ ╩ ⇔ ╕ Δ ⁹

ㅎ ○ ▶ fi ㅋ — ! ㄹ ㅃ ◒ ⁱ ▪ │ COUNT LATCH SPEC SET ◖ ∿ fi ♪ ≠ ⇔ ╕ Δ ⁹

3-3. HARD CONFIG 仕様

3-3-1. 入出力仕様

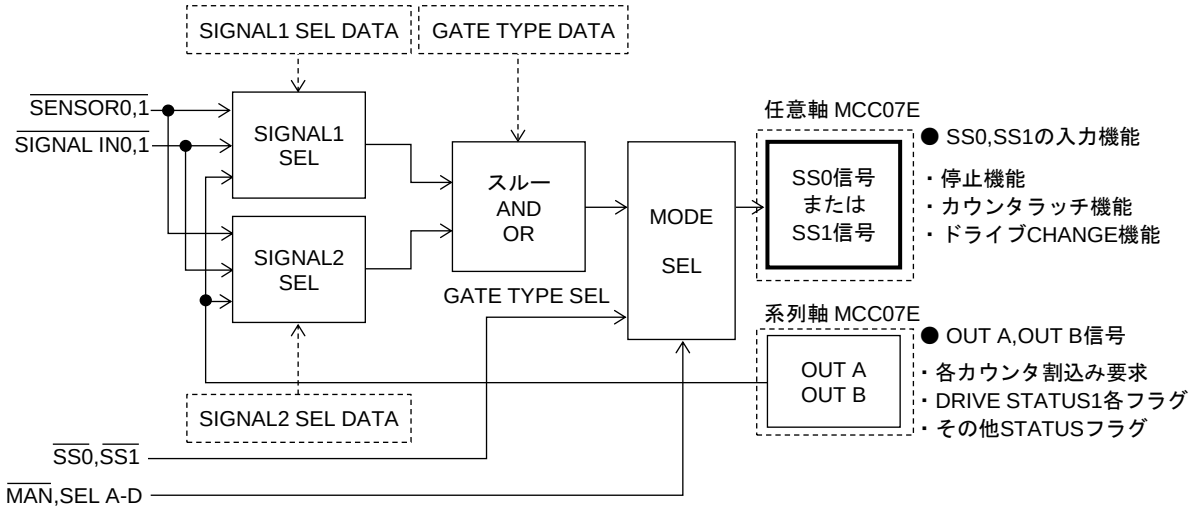
(1) 多用途センサ機能

各軸には停止機能、カウンタのラッチ機能、ドライブ CHANGE 機能のトリガとして使用可能な多用途センサ信号 SS0, SS1 があります。
これら SS0,SS1 信号に、下記のうち任意な信号を接続することができます。

多用途センサ信号	接続できる信号	
SS0,SS1	ユーザ I/O	$\overline{\text{SENSOR0}}$ 信号
		$\overline{\text{SENSOR1}}$ 信号
	特殊 I/O	$\overline{\text{SIGNAL IN0}}$ 信号
		$\overline{\text{SIGNAL IN1}}$ 信号
	MCC07E ステータス信号	任意軸 OUT A 信号
		任意軸 OUT B 信号

・ C-VX872v1, C-VX872Ev1, C-VX873v1, C-VX873Ev1 の場合、
SS0,SS1 信号に接続できる信号は同じ系列軸内の信号になります。

■ **多用途センサ信号の接続構成**



・ 接続の設定は、HARD CONFIGURATION の SENSOR SIGNAL SELECT コマンドにより設定します。
・ SIGNAL2 SEL と GATE TYPE SEL を使用すると任意 2信号の合成論理信号を多用途センサに接続できます。
・ MANUAL モード時はユーザ接続設定が無効となり、MANUAL ドライブ指定軸の多用途センサ SS0,SS1 に特殊 I/O の $\overline{\text{SS0}}$,$\overline{\text{SS1}}$ 信号が接続されます。
BUS モードに復帰するとユーザ接続設定が有効になります。

■ **多用途センサ信号の入力機能**

多用途センサ信号 SS0, SS1 の入力機能をそれぞれ以下の内から選択することができます。
・ 減速停止信号として使用する。
・ 即時停止信号として使用する。
・ カウンタラッチ機能(SS0 のみ)、ドライブ CHANGE 機能のトリガ信号として使用する。
・ ドライブ CHANGE の UP/DOWN/CONS 指令入力として使用する。

・ SS0,SS1 機能の設定は、MCC07E の SPEC INITIALIZE2 コマンドにより設定します。

■ **OUT A,B 信号**

MCC07E の OUT A,B 信号は以下のステータスを出力することができます。

・ カウンタ割り込み要求の ARDINT
・ カウンタ割り込み要求の CNTINT
・ カウンタ割り込み要求の DFLINT
・ コマンド終了割り込みの RDYINT
・ STATUS1 の STBY フラグ
・ STATUS1 の DRIVE フラグの反転
・ STATUS5 の SPEED CBUSY の反転
・ STATUS5 の INDEX CBUSY の反転
・ STATUS1 の UP フラグ
・ STATUS1 の DOWN フラグ
・ STATUS1 の CONST フラグ
・ STATUS1 の EXT PULSE フラグ
・ STATUS2 の PULSE MASK フラグの反転
・ STATUS2 の ORG SIGNAL フラグ
・ 汎用出力としての出力状態
・ STATUS4 の PULSE OVF フラグ

・ OUT A,OUT B に出力する信号の設定は、MCC07E の HARD INITIALIZE1 コマンドにより設定します。

(2) ステータス外部出力機能

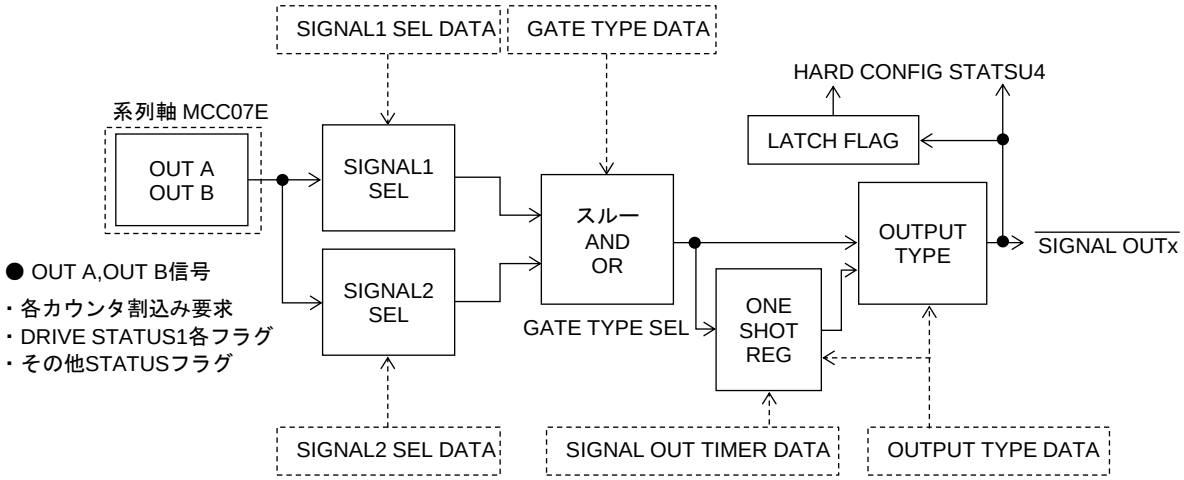
任意な軸のステータス信号(OUT A,B 信号)を、特殊 I/O コネクタ(J3)の $\overline{\text{SIGNAL OUTx}}$ 信号に接続し、外部にステータス信号を出力することができます。

外部出力信号	接続できる信号	
$\overline{\text{SIGNAL OUT0,1}}$	MCC07E ステータス信号	任意軸 OUT A 信号 任意軸 OUT B 信号

・ C-VX872v1, C-VX872Ev1, C-VX873v1, C-VX873Ev1 の場合、
$\overline{\text{SIGNAL OUTn0,n1}}$ 信号に接続できる信号は同じ系列軸内の信号になります。

$\overline{\text{SIGNAL OUTx}}$ 信号の状態は、HARD CONFIG STATUS4 PORT から読み出すことができます。
また、$\overline{\text{SIGNAL OUTx}}$ 信号のアクティブ出力検出を HARD CONFIG STATUS4 PORT のラッチフラグで確認することができます。
ラッチフラグのクリアは、SIGNAL OUT LATCH CLR コマンドで行います。

■ ステータス外部出力信号の接続構成



・ 接続の設定は、HARD CONFIGURATION に SIGNAL OUT SELECT コマンドにより設定します。
・ SIGNAL2 SEL と GATE TYPE SEL を使用すると任意 2信号の合成論理信号を外部出力することができます。
・ OUTPUT TYPE を設定すると任意信号の任意エッジ検出でワンショット出力することができます。
ワンショット出力時間は 1us ～ 65.535ms の範囲で 1us 単位で設定します。
・ ワンショット出力時間の設定は、HARD CONFIGURATION に SIGNAL OUT TIMER SET コマンドで設定します。

■ OUT A,B 信号

MCC07E の OUT A,B 信号は以下のステータスを出力することができます。

・ カウンタ割り込み要求の ARDINT
・ カウンタ割り込み要求の CNTINT
・ カウンタ割り込み要求の DFLINT
・ コマンド終了割り込みの RDYINT
・ STATUS1 の STBY フラグ
・ STATUS1 の DRIVE フラグの反転
・ STATUS5 の SPEED CBUSY の反転
・ STATUS5 の INDEX CBUSY の反転
・ STATUS1 の UP フラグ
・ STATUS1 の DOWN フラグ
・ STATUS1 の CONST フラグ
・ STATUS1 の EXT PULSE フラグ
・ STATUS2 の PULSE MASK フラグの反転
・ STATUS2 の ORG SIGNAL フラグ
・ 汎用出力としての出力状態
・ STATUS4 の PULSE OVF フラグ

OUT A,OUT B に出力する信号の設定は、MCC07E の HARD INITIALIZE1 コマンドにより設定します。

3-3-2. 同期スタート機能

任意複数軸のドライブ開始を同時にスタートさせることができます。
各軸は STBY 解除条件(PAUSE=0)を検出するまでドライブパルス出力の開始を保留します。
各軸の PAUSE SET および PAUSE CLR 条件を設定することで、同時パルス出力開始や他軸のドライブに連動したパルス出力開始を行うことができます。

■ **STBY フラグ**

各軸はドライブパルス出力の準備(データ処理)が完了すると、STBY 状態(STBY=1)となります。
STBY 状態で STBY 解除条件(PAUSE=0)を検出すると、STBY 状態を解除(STBY=0)し、ドライブパルス出力を開始します。

・ 各軸 MCC07E の OUT A 信号に STBY フラグを出力すると、HARD CONFIG STATUS3 PORT で一括して各軸の STBY 状態を確認することができます。

■ **PAUSE 信号**

各軸は PAUSE SET 条件を検出すると PAUSE 信号=1 となり、PAUSE CLR 条件を検出すると PAUSE 信号=0 となります。
PAUSE 信号=1 のときは、STBY 状態(STBY=1)を保持して、ドライブパルス出力の開始を保留します。
PAUSE 信号=0 で、STBY 状態を解除(STBY=0)し、ドライブパルス出力を開始します。

各軸の PAUSE SET 条件および PAUSE CLR 条件は以下からそれぞれ選択することができます。

PAUSE SET 条件
・ PAUSE コマンドの実行で SET する。 ・ 選択信号のアクティブエッジ検出で SET する。 ・ 選択信号のノットアクティブエッジ検出で SET する。

PAUSE CLR 条件
・ PAUSE コマンドの実行で CLR する。 ・ 選択信号のアクティブエッジ検出で CLR する。 ・ 選択信号のノットアクティブエッジ検出で CLR する。

・ 各軸の PAUSE SET 条件は、HARD CONFIGURATION に PAUSE SET SPEC コマンドで設定します。
・ 各軸の PAUSE CLR 条件は、HARD CONFIGURATION に PAUSE CLR SPEC コマンドで設定します。

● PAUSE コマンドの実行で SET または CLR する
HARD CONFIG DATA2 PORT の対象軸のビットに PAUSE データを設定し、PAUSE コマンドで実行します。
PAUSE データは"1"で PAUSE 信号=1(SET)、"0"で PAUSE 信号=0(CLR)となります。

・ PAUSE データ="1"での PAUSE コマンド実行は、対象軸の PAUSE SET 条件が「PAUSE コマンドの実行でSET する」に設定されている場合にのみ有効です。
・ PAUSE データ="0"での PAUSE コマンド実行は、対象軸の PAUSE CLR 条件が「PAUSE コマンドの実行でCLR する」に設定されている場合にのみ有効です。

● 選択信号のアクティブエッジ／ノットアクティブエッジ検出で SET または CLR する
以下の信号のうち、任意な信号のエッジ検出で PAUSE 信号を 1(SET)または 0(CLR)にします。

エッジ検出に使用できる信号	
ユーザ I/O	$\overline{\text{SENSOR0}}$ 信号
	$\overline{\text{SENSOR1}}$ 信号
特殊 I/O	$\overline{\text{SIGNAL IN0}}$ 信号
	$\overline{\text{SIGNAL IN1}}$ 信号
MCC07E ステータス信号	任意軸 OUT A 信号
	任意軸 OUT B 信号

・ C-VX872v1, C-VX872Ev1, C-VX873v1, C-VX873Ev1 の場合、
エッジ検出に使用できる信号は同じ系列軸内の信号になります。
・ PAUSE SET 条件におけるエッジ検出信号の選択は PAUSE SET SPEC コマンドで設定します。
・ PAUSE CLR 条件におけるエッジ検出信号の選択は PAUSE CLR SPEC コマンドで設定します。
・ エッジ検出信号の選択は任意 2 信号の合成論理信号とすることもできます。

Ẽ **OUT A,B** كخ

MCC07E — OUT A,B كخ | ⁸ MCC07E — HARD INITIALIZE1 ◖ ∀ fi ♪ ⌐ ╟╡ ⇔ ╕ Δ ⁹

ה ○ ▶ fi ♃ ╡ ╥ — ARDINT	ה STATUS1 — UP ɧ ! ◓
ה ○ ▶ fi ♃ ╡ ╥ — CNTINT	ה STATUS1 — DOWN ɧ ! ◓
ה ○ ▶ fi ♃ ╡ ╥ — DFLINT	ה STATUS1 — CONST ɧ ! ◓
ה ◖ ∀ fi ♪ ǧ ╡ ╥ — RDYINT	ה STATUS1 — EXT PULSE ɧ ! ◓
ה STATUS1 — STBY ɧ ! ◓	ה STATUS2 — PULSE MASK ɧ ! ◓ —
ה STATUS1 — DRIVE ɧ ! ◓ —	ה STATUS2 — ORG SIGNAL ɧ ! ◓
ה STATUS5 — SPEED CBUSY —	ה ≤ ⇔ ≡ —
ה STATUS5 — INDEX CBUSY —	ה STATUS4 — PULSE OVF ɧ ! ◓

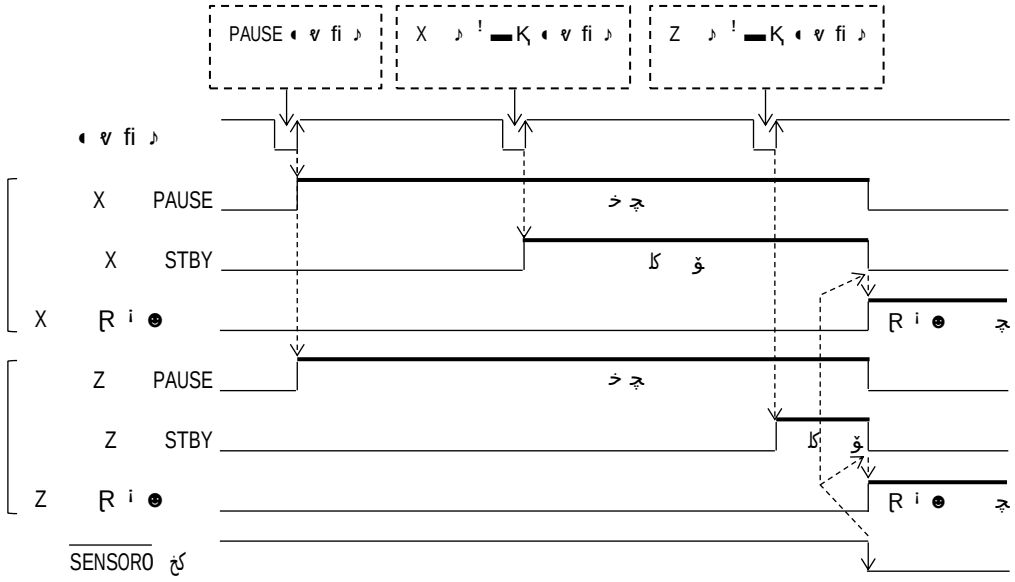
ה OUT A,OUT B ⌐ Δ ╟ كخ — | ⁸ MCC07E — HARD INITIALIZE1 ◖ ∀ fi ♪ ⌐ ╟╡ ⇔ ╕ Δ ⁹

Ẽ ● ♃ ↄ ↓ لأ ⅜ **SENSOR0** كخ ≡ ● ♃ ↄ ↓) **(X ,Z ⅜ لأ**

ئ

ה X	STBY	⌠	PAUSE=0
ה X	PAUSE SET	⌠	PAUSE ◖ ∀ fi ♪
ה X	PAUSE CLR	⌠	SENSOR0 طص كخ — ▪ ▬ ♥ ▫ Ϟ ◀ ℘ ☺
ה Z	STBY	⌠	PAUSE=0
ה Z	PAUSE SET	⌠	PAUSE ◖ ∀ fi ♪
ה Z	PAUSE CLR	⌠	SENSOR0 طص كخ — ▪ ▬ ♥ ▫ Ϟ ◀ ℘ ☺

ئ لأ



3-3-3. ╥ ⇔

(1) ● ♥ ⊃ ∝ ● ╥ ⇔

STATUS PORT ╩ ╥ Δ ↓ ≤ ≠ ⁸ $\overline{\text{MAN RDY}}$ ☜ — ⁸ $\overline{\text{SIGNAL OUTx}}$, $\overline{\text{SIGNAL INx}}$ ☜ — ⁸

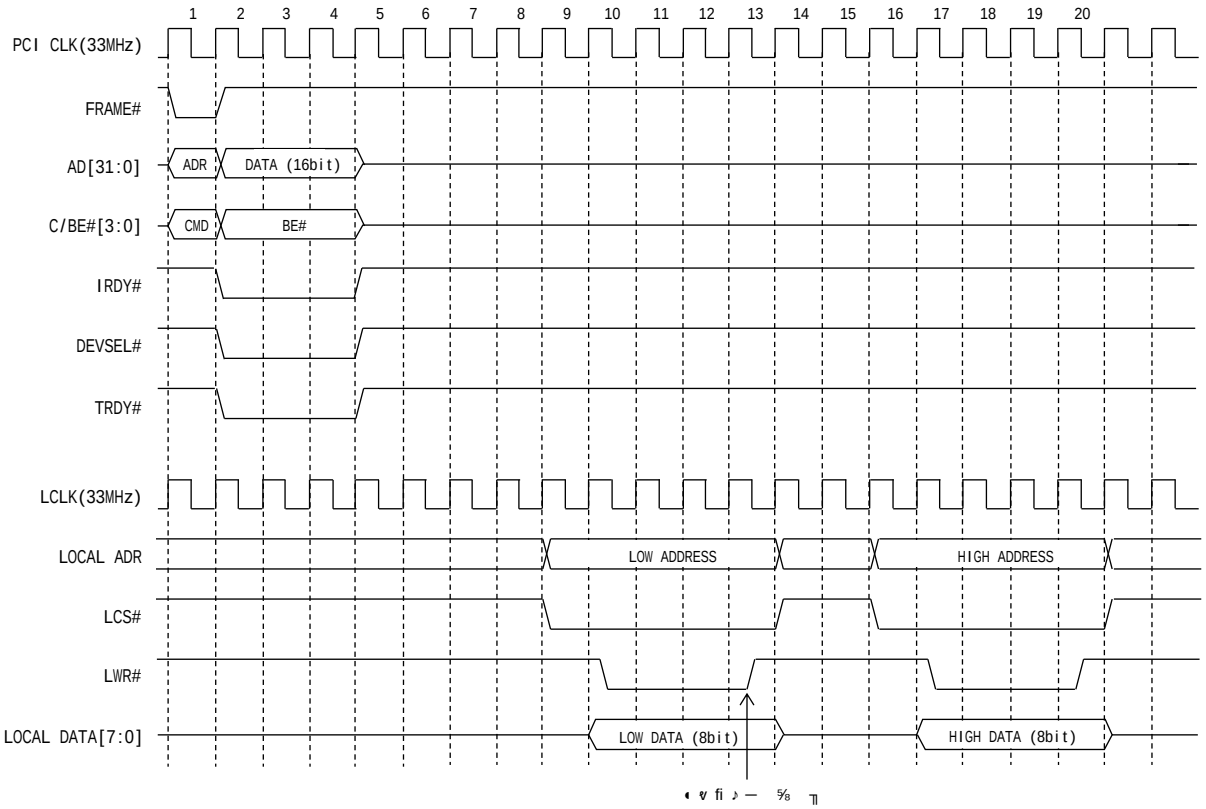
PAUSE ☜ — ⌂ ≥ ⅜ ╥ ∑ ╕ Δ ⁹

(2) ♦ ⊃ ∝ ╥ ⇔

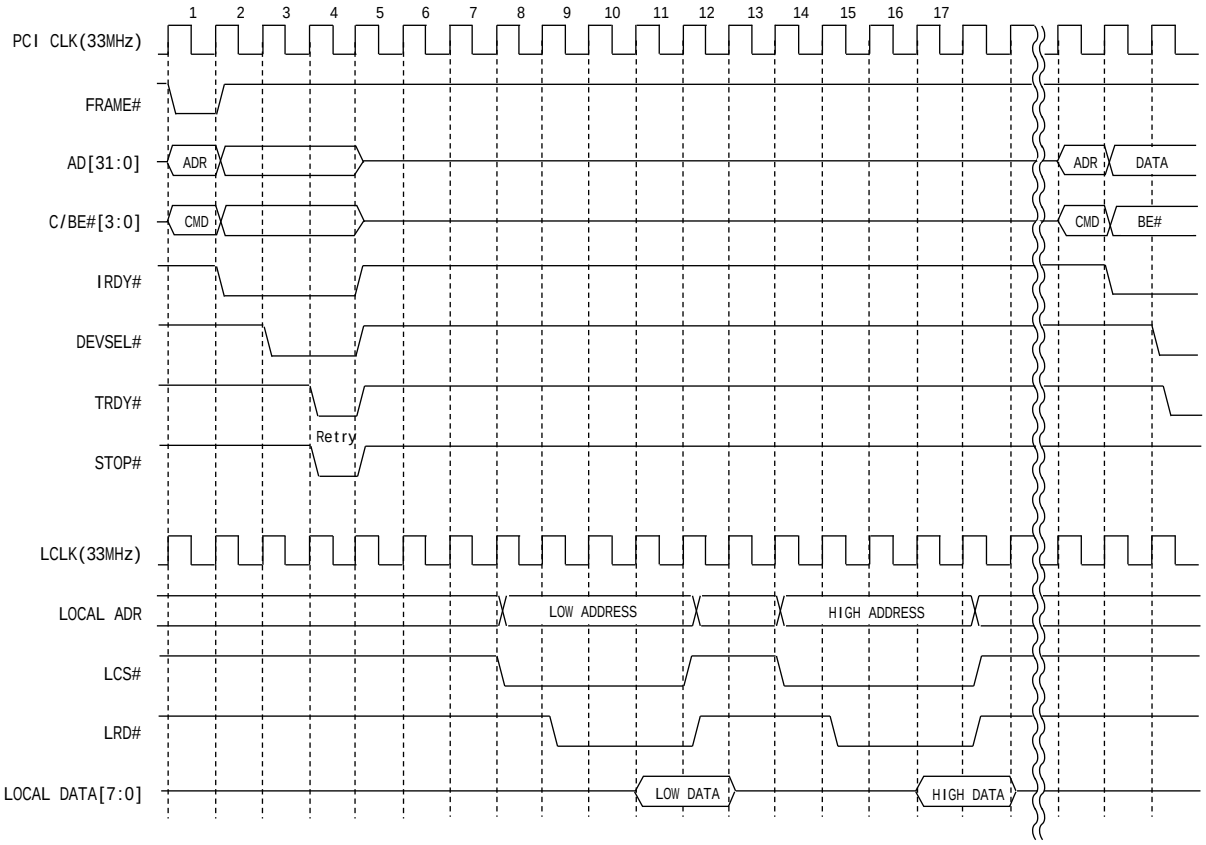
HARD CONFIGURATION ⌐ HARD CONFIG SET DATA READ ◖ ∀ fi ♪ ╩ Δ ╢ ≤ ⁸

⇔ √ ♦ ⊃ ∝ ⅜ ╥ ∑ ╕ Δ ⁹

4-1. ±¬▬⊢ fi ◓
(1) PCI Ᵽ ●
Ẽ **WRITE**

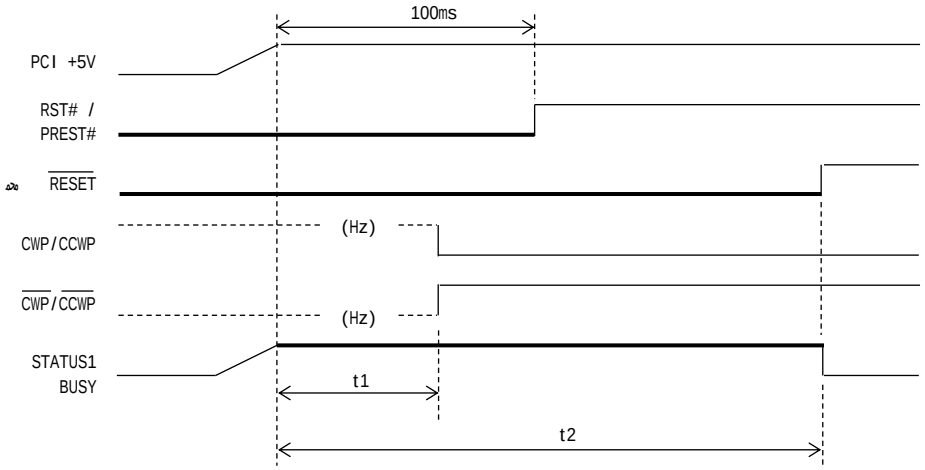


Ẽ **READ**



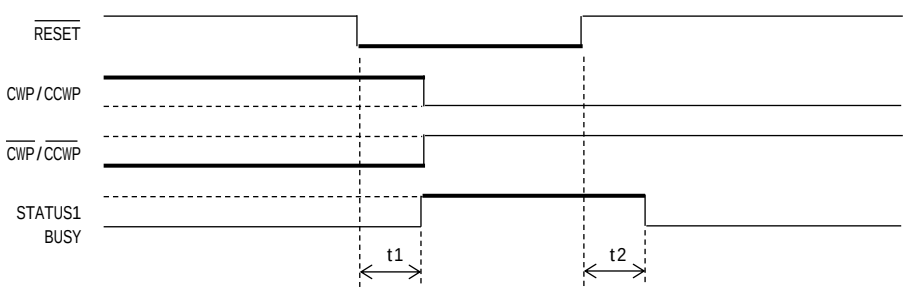
(2) [illegible]

Ê POWER ON



t1 20ms
t2 400ms

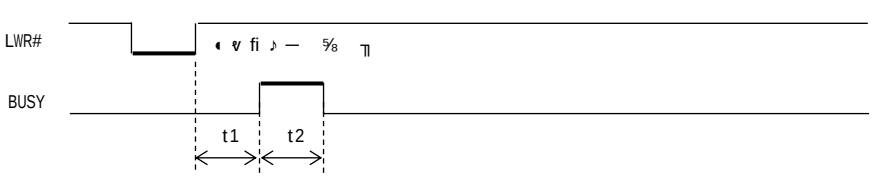
Ê RESET [illegible]



t1 5ms
t2 5ms

(3) [illegible]

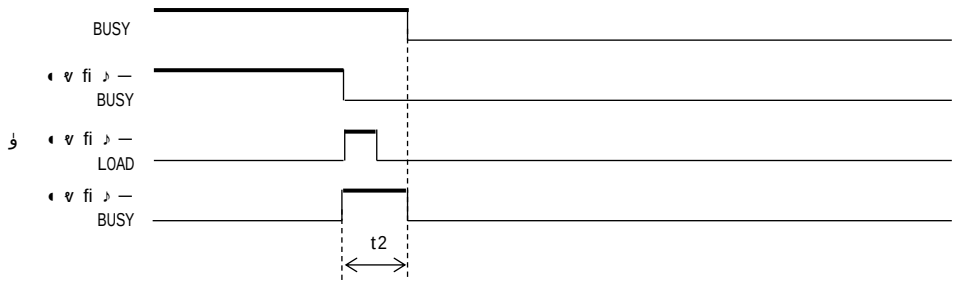
Ê [illegible]



75ns t1 125ns
t2 100ns

BUSY [illegible]

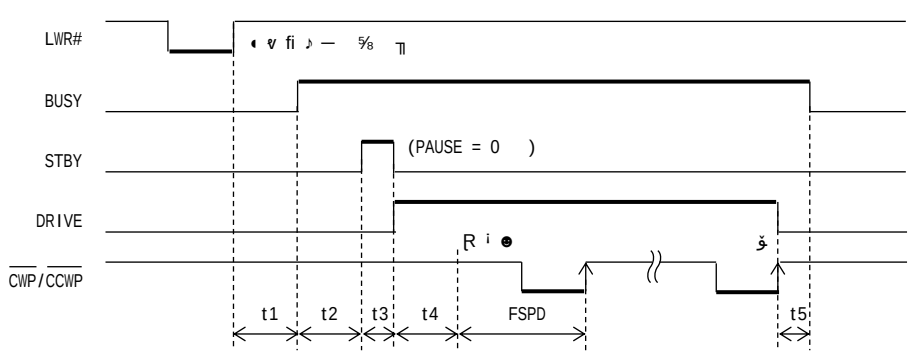
Ê [illegible]



t2 100ns

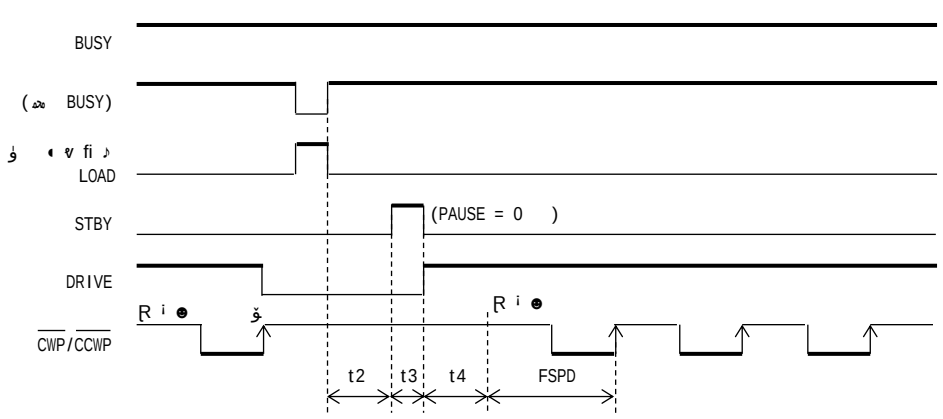
(4) ♪ ! ▬Ķ —

Ẽ ♪ ! ▬Ķ — ≤ ğ



75ns	t1	125ns
t2	100ns	
t3	50ns	
t4	100ns	
t5	50ns	

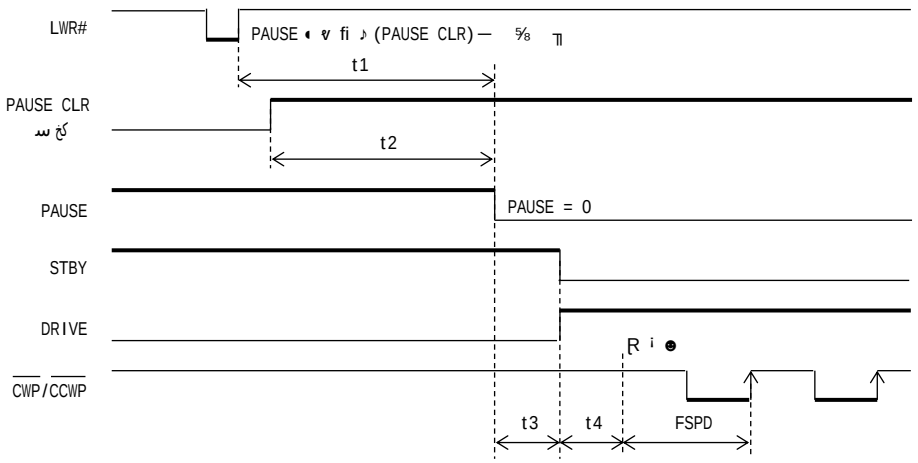
Ẽ ◖ ∀ fi ♪ ů ⌐ ╟┤ ♪ ! ▬Ķ



t2	100ns	
t3	50ns	
t4	100ns	

ה ♪ ! ▬Ķ ♪ ! ▬Ķ ⌐ ◖ ∀ fi ♪ ů ⇔ √ 8 ◖ ∀ fi ♪ 1 ◖ ∀ fi ♪ √ ╡ —
│ 100ns ⌐ ⌂ ╡ ╕ Δ 9

Ẽ ● ◓ ↄ ⌐ ╟┤ ♪ ! ▬Ķ —

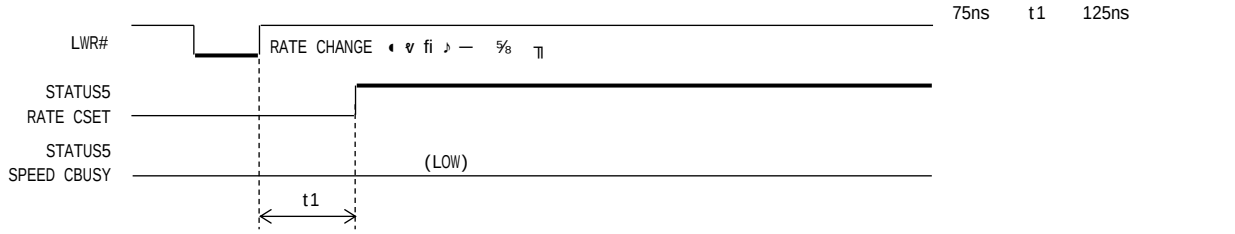


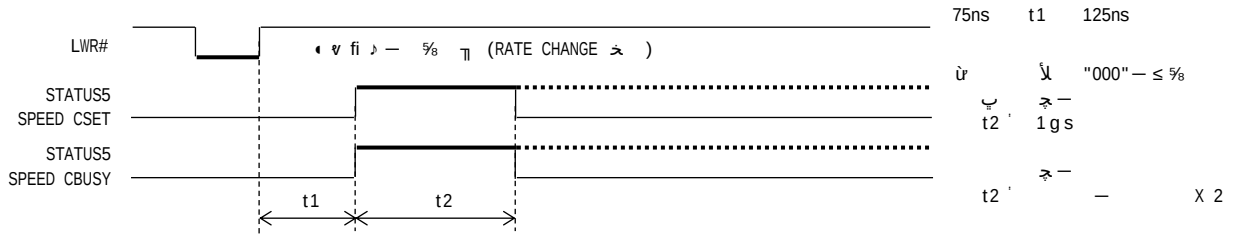
t1	200ns	
t2	1ms	*1
	10 g s	*2
	150ns	*3
75ns	t3	125ns
t4	100ns	

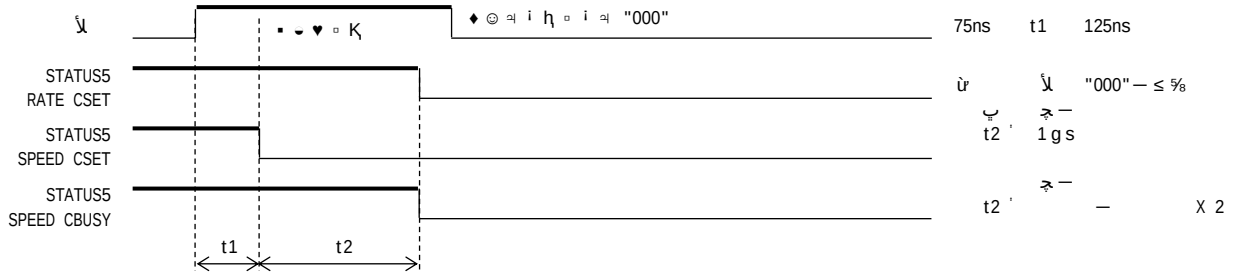
ה SENSORnx ⌐ ╟┤ PAUSE CLR — ≢ Δ 9 *1
ה SIGNAL INnx ⌐ ╟┤ PAUSE CLR — ≢ Δ 9 *2
ה ش ☻ ♥ ↄ ◓ ☻ (OUT A,B) ⌐ ╟┤ PAUSE CLR — ≢ Δ 9 *3

(5) CHANGE ♪ ! ▬Ķ ─

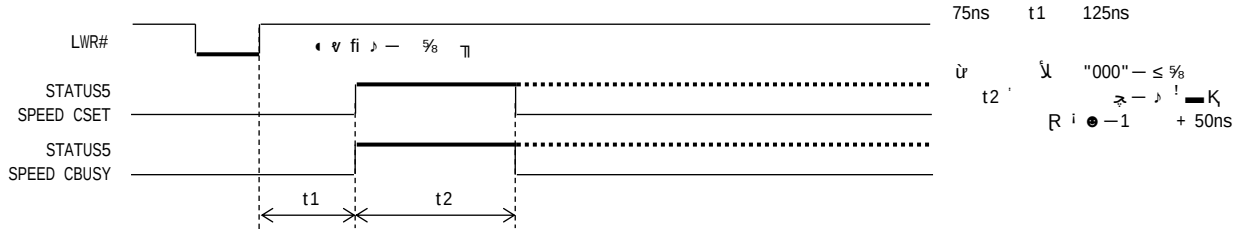
Ẽ ● Ḥ ↄ ♪ CHANGE ♪ ! ▬Ķ

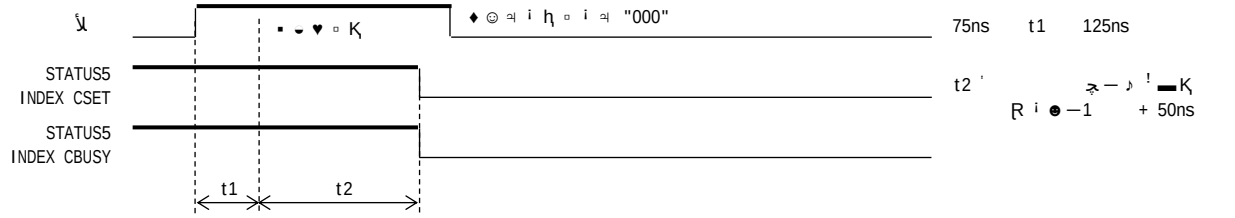






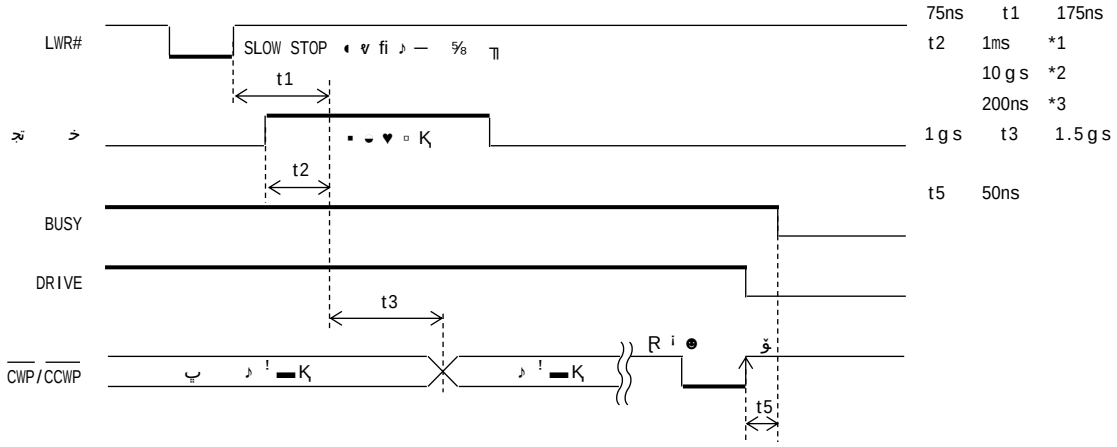
Ẽ **INDEX CHANGE ♪ ! ▬Ķ ● Ḥ ↄ ♪**





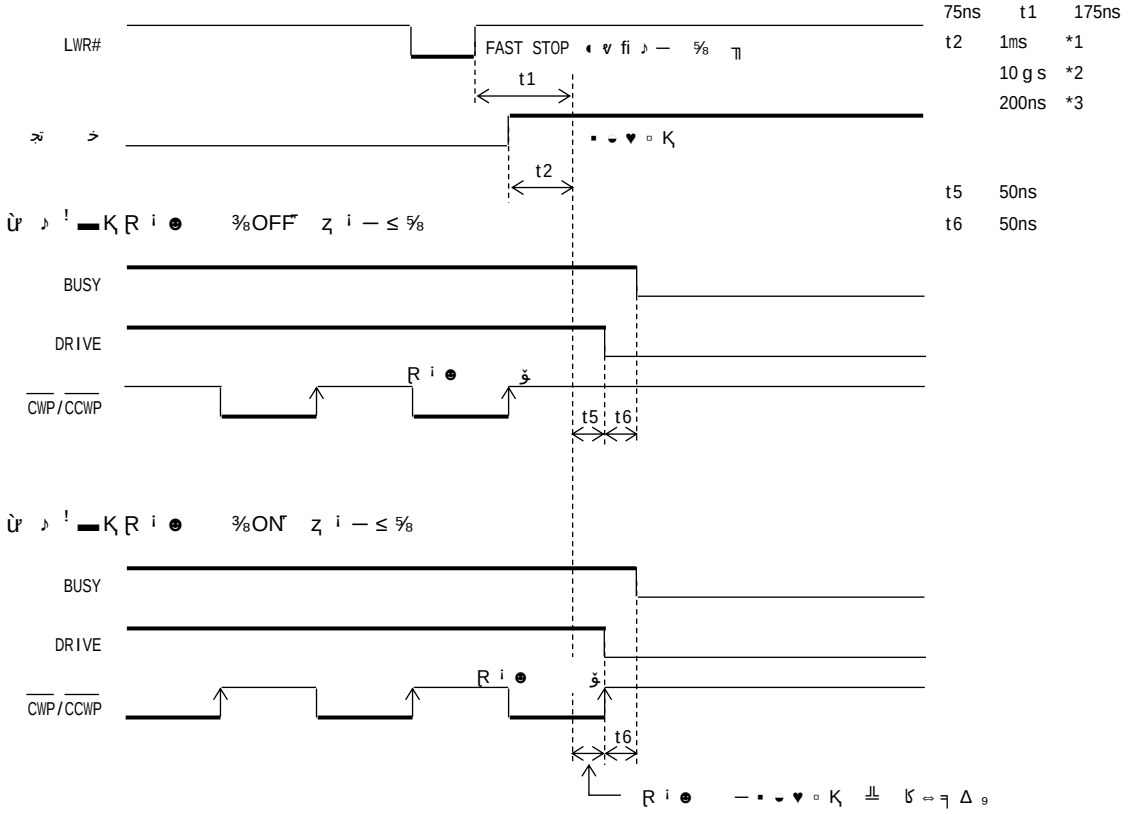
(6) **تجـ**

Ē **تجـ ה LIMIT減速停止**



ה CWLM, CCWLM, SENSORnx, DALM(INnx) כיל ⌐ ╟┼╢ תבג חכ — ≢ Δ ⁹ *1
ה SIGNAL INnx כיל ⌐ ╟┼╢ תבג חכ — ≢ Δ ⁹ *2
ה ○ ▶ fi ינ פ ⁸ ש ☻ ♥ כ ינ ☻ כיל (OUT A,B כיל) ⌐ ╟┼╢ תבג חכ — ≢ Δ ⁹ *3

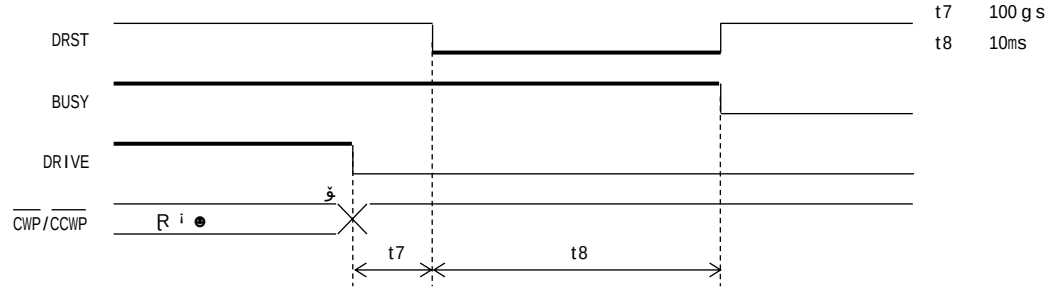
Ē **תבג ה LIMIT即時停止**



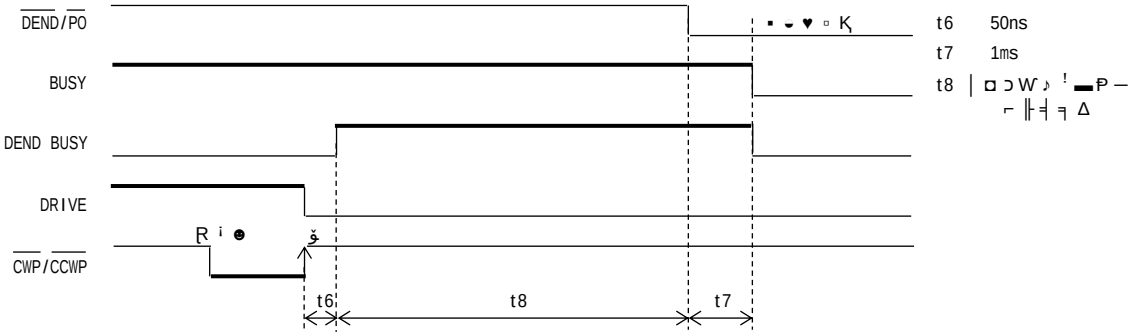
ה CWLM, CCWLM, SENSORnx, DALM(INnx) כיל ⌐ ╟┼╢ תבג חכ — ≢ Δ ⁹ *1
ה SIGNAL INnx כיל ⌐ ╟┼╢ תבג חכ — ≢ Δ ⁹ *2
ה ○ ▶ fi ינ פ ⁸ ש ☻ ♥ כ ינ ☻ כיל (OUT A,B כיל) ⌐ ╟┼╢ תבג חכ — ≢ Δ ⁹ *3

(7) ◘ ⊃ W̄

Ē **DRST 信号のアクティブ出力**



Ē **DEND/PO 信号のアクティブ検出**



4-2. ث پ
(1)

7 7	ث	◖ ∀ fi ♪
■ Ꞁ ⁱ ☻		
Ꞁ ⁱ ☻		SPEC INITIALIZE1 ◖ ∀ fi ♪
Ꞁ ⁱ ☻ ∀ ☻ ⌄	∀ ☻ ⌄ ⇔ ⌂ ™	
MANUAL ♪ ! ▬ Ꞁ ⁼ ⊃ ♪	SCAN ♪ ! ▬ Ꞁ	
■ LIMIT تج		
CWLM كج طص	+ — LIMIT تج طص	SPEC INITIALIZE2 ◖ ∀ fi ♪
CCWLM كج طص	- — LIMIT تج طص	
■ ⁼ ⅡI		
RDYINT ث	DRVEND ≠ RDYINT	SPEC INITIALIZE2 ◖ ∀ fi ♪
■ ☿ fi ◘		
☿ fi ◘ (SS0)	طص ● ⁱ ⌋— ⁸ طص	SPEC INITIALIZE2 ◖ ∀ fi ♪
☿ fi ◘ (SS1)	طص ● ⁱ ⌋— ⁸ طص	
■ ◘ ⊃ W		
DRST كج		SPEC INITIALIZE3 ◖ ∀ fi ♪
DEND/PO كج طص	طص	
DALM كج طص	طص *1	
■ ☻ ⁼ ⊃ ♩		
STBY سد	PAUSE=0 ≠	SPEC INITIALIZE3 ◖ ∀ fi ♪
■ تج		
DOWN PULSE ∀ ☻ ⌄	∀ ☻ ⌄ ⇔ ⌂ ™	SPEC INITIALIZE3 ◖ ∀ fi ♪
■ طص كج — ♦ ☺ ⁼ ⁱ ⁿ ▫ ⁱ ⁼		
CWLM, CCWLM كج	0 50ns	HARD INITIALIZE4 ◖ ∀ fi ♪
DEND/PO, DALM(INnx) كج	0 50ns	
ORG, NORG كج	0 50ns	HARD INITIALIZE5 ◖ ∀ fi ♪
ÑZORG كج	0 50ns	
ÑEA, ÑEB كج	0 50ns	HARD INITIALIZE6 ◖ ∀ fi ♪
■ طص كج — ▪ ⌄ ♥ ▫ Ꞁ		
FSSTOP كج		HARD INITIALIZE7 ◖ ∀ fi ♪
CWLM كج		
CCWLM كج		
NORG كج		
ORG كج		
DALM(INnx) كج		
■ ◄ ! ⊃		
COMMAND ERROR	∀ ☻ ⌄ ⇔ ⌂ ™	ERROR STATUS MASK ◖ ∀ fi ♪
COMREG CLR ERROR	∀ ☻ ⌄ ⇔ ⌂ ™	
INC INDEX ERROR	∀ ☻ ⌄ ⇔ ⌂ ™(≠ ⅝ ¬ ∑ ╪)	
ABS INDEX ERROR	∀ ☻ ⌄ ⇔ ⌂ ™(≠ ⅝ ¬ ∑ ╪)	
INDEX CHANGE ERROR	∀ ☻ ⌄ ⇔ ⌂ ™(≠ ⅝ ¬ ∑ ╪)	
CHANGE CLR ERROR	∀ ☻ ⌄ ⇔ ⌂ ™	
CPP STOP ERROR	∀ ☻ ⌄ ⇔ ⌂ ™	
EXT PULSE ERROR	∀ ☻ ⌄ ⇔ ⌂ ™	
FSEND ERROR	∀ ☻ ⌄ ⇔ ⌂ ™	
LSEND ERROR	∀ ☻ ⌄ Δ ╢	
SSEND ERROR	∀ ☻ ⌄ Δ ╢	
ADDRESS OVF ERROR	∀ ☻ ⌄ Δ ╢	
PULSE OVF ERROR	∀ ☻ ⌄ Δ ╢	
DALM ERROR	∀ ☻ ⌄ Δ ╢	
FSSTOP ERROR	∀ ☻ ⌄ Δ ╢	

7 7	ث	◖ ∇ fi ♪
■ ERRINT		
COMMAND ERROR	∇ ☻ ▾ Δ ┤	ERRINT STATUS MASK ◖ ∇ fi ♪
COMREG CLR ERROR	∇ ☻ ▾ Δ ┤	
INC INDEX ERROR	∇ ☻ ▾ Δ ┤	
ABS INDEX ERROR	∇ ☻ ▾ Δ ┤	
INDEX CHANGE ERROR	∇ ☻ ▾ Δ ┤	
CHANGE CLR ERROR	∇ ☻ ▾ Δ ┤	
CPP STOP ERROR	∇ ☻ ▾ Δ ┤	
EXT PULSE ERROR	∇ ☻ ▾ Δ ┤	
FSEND ERROR	∇ ☻ ▾ Δ ┤	
LSEND ERROR	∇ ☻ ▾ Δ ┤	
SSEND ERROR	∇ ☻ ▾ Δ ┤	
ADDRESS OVF ERROR	∇ ☻ ▾ Δ ┤	
PULSE OVF ERROR	∇ ☻ ▾ Δ ┤	
DALM ERROR	∇ ☻ ▾ Δ ┤	
FSSTOP ERROR	∇ ☻ ▾ Δ ┤	
■ ╡ ╗		
RDYINT	∇ ☻ ▾ Δ ┤	INT FACTOR MASK ◖ ∇ fi ♪
STBY INT	∇ ☻ ▾ Δ ┤	
COMREG EP INT	∇ ☻ ▾ Δ ┤	
nCOMREG FL INT	∇ ☻ ▾ Δ ┤	
MAN INT	∇ ☻ ▾ Δ ┤	
DALM INT	∇ ☻ ▾ Δ ┤	
SS0 INT	∇ ☻ ▾ Δ ┤	
SS1 INT	∇ ☻ ▾ Δ ┤	
ADRINT	∇ ☻ ▾ Δ ┤	
CNTINT	∇ ☻ ▾ Δ ┤	
DFLINT	∇ ☻ ▾ Δ ┤	
ERRINT	∇ ☻ ▾ Δ ┤	
■ ▪ ♪ ☻ ○ ▶ fi ⅁		
○ ▶ fi ♩ R ⁱ ☻	R ⁱ ☻	ADDRESS COUNTER INITIALIZE1 ◖ ∇ fi ♪
◀ fi ◖ ⊃ ♭ ظ R ⁱ ☻ ○ ▶ fi ♩	1 ى	
R ⁱ ☻ — ▪ ▾ ♥ ▫ Ꝇ	1 εs	
ADRINT ث	˝ ʒ ⁱ ! ᵖ ᵸ	
ADRINT ☻ ⁱ ⊃	200ns	
COMP	(OR)	
COMP1 ▾ ⁱ ▪	▾ ⁱ ▪ ⇔ ⌂ ™	
COMP1	⅌ ⇔ ⌂ ™	
COMP1,2,3 INT ENABLE	⇔ ⌂ ™	ADDRESS COUNTER INITIALIZE2 ◖ ∇ fi ♪
COMP1,2,3 STOP ENABLE	تج ⇔ ⌂ ™	
COMP1,2,3 STOP TYPE	تج	
COMP2,3 س	(پ)	
○ ▶ fi ⅁ ثم	H'0000_0000	ADDRESS COUNTER PRESET ◖ ∇ fi ♪
○ ▶ fi ♩	H'FFFF_FFFF	ADDRESS COUNTER MAX COUNT SET ◖ ∇ fi ♪
COMPARE REGISTER ثم(1,2,3)	H'8000_0000	ADRINT COMPARE REGISTER1,2,3 SET コマンド
COMP1 ثم	H'0000_0000	ADRINT COMP ADD DATA SET ◖ ∇ fi ♪

7 7	ث	◂ ∀ fi ♪
■ Ꞧ i ☻ ○ ▶ fi ⍝		
○ ▶ fi ⍼ Ꞧ i ☻	Ꞧ i ☻	PULSE COUNTER INITIALIZE1 ◂ ∀ fi ♪
◀ fi ◂ ⊃ ♭ ظ Ꞧ i ☻ ○ ▶ fi ⍼	1 ي	
CNTINT ث	⍞ Ƶ i ! ℘ ⍓	
CNTINT ☻ i ⊃	200ns	
COMP	(OR)	
COMP1 ⍥ i ▪	⍥ i ▪ ⇔ ⌂ ™	
COMP1	⍝ ⇔ ⌂ ™	
COMP1,2,3 INT ENABLE	⇔ ⌂ ™	PULSE COUNTER INITIALIZE2 ◂ ∀ fi ♪
COMP1,2,3 STOP ENABLE	⍨ ⇔ ⌂ ™	
COMP1,2,3 STOP TYPE	⍨	
COMP2,3 س	(پ)	
○ ▶ fi ⍝ ثم	H'0000_0000	PULSE COUNTER PRESET ◂ ∀ fi ♪
○ ▶ fi ⍼	H'FFFF_FFFF	PULSE COUNTER MAX COUNT SET ◂ ∀ fi ♪
COMPARE REGISTER ثم(1,2,3)	H'8000_0000	CNTINT COMPARE REGISTER1,2,3 SET ◂ ∀ fi ♪
COMP1 ثم	H'0000_0000	CNTINT COMP ADD DATA SET ◂ ∀ fi ♪
■ Ꞧ i ☻ ⍨ ○ ▶ fi ⍝		
○ ▶ fi ⍼ Ꞧ i ☻ 1	◀ fi ◂ ⊃ ♭ ظ Ꞧ i ☻	DFL COUNTER INITIALIZE1 ◂ ∀ fi ♪
○ ▶ fi ⍼ Ꞧ i ☻ 2	Ꞧ i ☻	
◀ fi ◂ ⊃ ♭ ظ Ꞧ i ☻ ○ ▶ fi ⍼	1 ي	
⍥ ℘ ⍥ ○ ▶ fi ⍼ ⍝ ▬ ⊢ fi ⍩	○ ▶ fi ⍼ ⇔ ⌂ ™	
Δ ⍦ ○ ▶ fi ⍼ Ꞧ i ☻	○ ▶ fi ⍼ Ꞧ i ☻ 1	
DFLINT ث	⍞ Ƶ i ! ℘ ⍓	
DFLINT ☻ i ⊃	200ns	
COMP	(OR)	
COMP1 ⍥ i ▪	⍥ i ▪ ⇔ ⌂ ™	
COMP1	⍝ ⇔ ⌂ ™	
COMP1,2,3 INT ENABLE	⇔ ⌂ ™	DFL COUNTER INITIALIZE2 ◂ ∀ fi ♪
COMP1,2,3 STOP ENABLE	⍨ ⇔ ⌂ ™	
COMP1,2,3 STOP TYPE	⍨	
COMP1,2,3	○ ▶ fi ⍝ ثم ⍦ ثم ⍈ ⇔ ≡ Δ ⍦	
COMP2 س	⍞ COMP2	
COMP3 س	⍞ COMP3	
○ ▶ fi ⍼ Ꞧ i ☻	0 (⌂ ⇔)	DFL COUNTER INITIALIZE3 ◂ ∀ fi ♪
○ ▶ fi ⍝ ثم	H'0000	DFL COUNTER PRESET ◂ ∀ fi ♪
COMPARE REGISTER ثم(1,2,3)	H'8000	DFLINT COMPARE REGISTER1,2,3 SET ◂ ∀ fi ♪
COMP1 ثم	H'0000	DFLINT COMP ADD DATA SET ◂ ∀ fi ♪
■ ○ ▶ fi ⍝ — ! ℘ ⍓ ⍝ ⍥ i ▪		
▪ ♪ ⍞ ☻ ○ ▶ fi ⍝ ! ℘ ⍓ ⍝ ▬ ⊢ fi ⍩	ADDRESS LATCH DATA READ ◂ ∀ fi ♪ —	COUNT LATCH SPEC SET ◂ ∀ fi ♪
ADDRESS CLR ENABLE	⍥ i ▪ ⇔ ⌂ ™	
Ꞧ i ☻ ○ ▶ fi ⍝ ! ℘ ⍓ ⍝ ▬ ⊢ fi ⍩	PULSE LATCH DATA READ ◂ ∀ fi ♪ —	
PULSE CLR ENALBE	⍥ i ▪ ⇔ ⌂ ™	
Ꞧ i ☻ ⍨ ○ ▶ fi ⍝ ! ℘ ⍓ ⍝ ▬ ⊢ fi ⍩	DFL LATCH DATA READ ◂ ∀ fi ♪ —	
DFL CLR ENABLE	⍥ i ▪ ⇔ ⌂ ™	

項　　目	初期仕様	対応コマンド
■ 多用途センサ機能		
各軸 SS0 入力機能	汎用入力、各種機能のトリガ入力	SPEC INITIALIZE2 コマンド
各軸 SS1 入力機能	汎用入力、各種機能のトリガ入力	
Xn 軸 SS0 接続信号	なし(LOW 固定)	SENSOR SELECT コマンド (HARD CONFIG コマンド)
Xn 軸 SS1 接続信号	なし(LOW 固定)	
Yn 軸 SS0 接続信号	なし(LOW 固定)	
Yn 軸 SS1 接続信号	なし(LOW 固定)	
Zn 軸 SS0 接続信号	SENSORn0 信号	
Zn 軸 SS1 接続信号	なし(LOW 固定)	
An 軸 SS0 接続信号	SENSORn1 信号	
An 軸 SS1 接続信号	なし(LOW 固定)	
Bn 軸 SS0 接続信号	なし(LOW 固定) *2	
Bn 軸 SS1 接続信号	なし(LOW 固定) *2	
Cn 軸 SS0 接続信号	なし(LOW 固定) *2	
Cn 軸 SS1 接続信号	なし(LOW 固定) *2	
■ ステータス外部出力機能		
各軸 OUT A 出力ステータス	CNTINT	HARD INITIALIZE1 コマンド
各軸 OUT B 出力ステータス	DFLINT	
SIGNAL OUTn0 出力信号	Xn 軸 OUT A	SIGNAL OUT SELECT コマンド (HARD CONFIG コマンド)
SIGNAL OUTn1 出力信号	Yn 軸 OUT A	
■ 同期スタート機能		
各軸 STBY 解除条件	PAUSE=0 で解除	SPEC INITIALIZE3 コマンド
各軸 PAUSE SET 条件	PAUSE コマンドの実行	PAUSE SET SPEC コマンド (HARD CONFIG コマンド)
各軸 PAUSE CLR 条件	PAUSE コマンドの実行	PAUSE CLR SPEC コマンド (HARD CONFIG コマンド)

*1　C-VX870v1, C-VX870Ev1, C-VX872v1, C-VX872Ev1 のみ

*2　C-VX871v1, C-VX871Ev1, C-VX873v1, C-VX873Ev1 のみ

(2) ♪ ! ▬Ķ Ŗ ! ϖ ↄ ⌐

7 7	ث	◖ ∀ fi ♪
■ 1 Ŗ ⁱ ☻ (FSPD)	5,000Hz	FSPD SET ◖ ∀ fi ♪
■ JOG Ŗ ! ϖ ↄ ⌐		
JOG Ŗ ⁱ ☻	300Hz	JSPD SET ◖ ∀ fi ♪
JOG Ŗ ⁱ ☻	1PULSE	JOG PULSE SET ◖ ∀ fi ♪
■ Ŗ ! ϖ ↄ ⌐		
ي (RESOL)	1 (No.3)	HIGH SPEED SET ◖ ∀ fi ♪
— ♦ ↄ ⌐ (HSPD)	3,000	
— ♦ ↄ ⌐ (LSPD)	300	LOW SPEED SET ◖ ∀ fi ♪
ϱ̆ — ♦ ↄ ⌐ (ELSPD)	300	
○ ↄ Ķ ♦ ↄ ⌐ (UCYCLE)	200	RATE SET ◖ ∀ fi ♪
○ ↄ Ķ ♦ ↄ ⌐ (DCYCLE)	200	
○ ↄ Ķ S ♦ ↄ ⌐ (SUAREA)	0	SCAREA SET ◖ ∀ fi ♪
○ ↄ Ķ S ♦ ↄ ⌐ (SDAREA)	0	
Ŗ ⁱ ☻ — ○ ϩ ϱ̆ ♀ ♩ Ŗ ⁱ ☻	1 Ŗ ⁱ ☻	DOWN PULSE ADJUST ◖ ∀ fi ♪
■ ORIGIN ♪ ! ▬Ķ Ŗ ! ϖ ↄ ⌐		
ORG كي	ORG كي ≤ ÑZORG كي — OR( )	ORIGIN SPEC SET ◖ ∀ fi ♪
ORG كي — ◀ ♀ ☺	ORG كي — 0 ŕ 1(▪ ▾ ♥ ▫ Ķ) ◀ ♀ ☺	
ORG ♪ ! ▬Ķ	(CCW)	
◀ ♀ ☺ — ○ ▶ fi ♩	1 ○ ▶ fi ♩ — ◀ ♀ ☺ ≢ تج	
تج DRST	⇔ ⌂ ™	
■ ♪ ! ▬Ķ Ŗ ! ϖ ↄ ⌐		
CPPOUT ⅛ ⌠ Δ ⌡ Ŗ ⁱ ☻	CPPIN ⅛ ⌠ ضي Δ ⌡ Ŗ ⁱ ☻	CP SPEC SET ◖ ∀ fi ♪
— — ▪ ♪ ˝ ☻	H'0000_0000	LONG POSITION SET ◖ ∀ fi ♪
— — ▪ ♪ ˝ ☻	H'0000_0000	SHORT POSITION SET ◖ ∀ fi ♪
مضي ₀ — X ▪ ♪ ˝ ☻	H'00_0000	CIRCULAR XPOSITION SET ◖ ∀ fi ♪
مضي ₀ — Y ▪ ♪ ˝ ☻	H'00_0000	CIRCULAR YPOSITION SET ◖ ∀ fi ♪
مضي ⌐ ≢ — Ŗ ⁱ ☻	H'0000_0000	CIRCULAR PULSE SET ◖ ∀ fi ♪
■ ♪ ! ▬Ķ CHANGE Ŗ ! ϖ ↄ ⌐		
UP DRIVE لأ	UP DRIVE CHANGE ◖ ∀ fi ♪ —	UDC SPEC SET ◖ ∀ fi ♪
DOWN DRIVE لأ	DOWN DRIVE CHANGE ◖ ∀ fi ♪ —	
CONST DRIVE لأ	CONST DRIVE CHANGE ◖ ∀ fi ♪	
SPEED CHANGE لأ	SPEED CHANGE ◖ ∀ fi ♪ —	SPEED CHANGE SPEC SET ◖ ∀ fi ♪
INDEX CHANGE لأ	INDEX CHANGE ◖ ∀ fi ♪ —	INDEX CHANGE SPEC SET ◖ ∀ fi ♪

4-3. ♪ ! ▬Ķ ◖ ∀ fi ♪ ب

MCC07E ♪ ! ▬Ķ ◖ ∀ fi ♪ — ب ╩ ⇔ ╕ Δ ⁹

ὺ ◖ ∀ fi ♪ DRIVE STATUS1 PORT — BUSY=0 ⌐ DRIVE COMMAND PORT ⌐
⅝ ╥ ⌂ ◖ ∀ fi ♪ ≠ Δ ⁹

ὺ ◖ ∀ fi ♪ DRIVE COMMAND PORT ⌐ ⅝ ╥ ⌂ ◖ ∀ fi ♪ ≠ Δ ⁹
○ ▶ fi ♃ ◖ ∀ fi ♪ ╩ ♪ ! ▬Ķ ◖ ∀ fi ♪ — ج ⌐ ╕ ╢ ╕ Δ ⁹

(1) ◖ ∀ fi ♪

COMMAND CODE	◖ ∀ fi ♪			( )	PAGE
H'00	NO OPERATION		⌂ ⇔	225 ns	97
H'01	SPEC INITIALIZE1		♪ ! ▬Ķ Ꞁ i ☻ — ث —	225 ns	25
H'02	SPEC INITIALIZE2		CWLM, CCWLM, RDYINT, SS0, SS1 —	225 ns	26
H'03	SPEC INITIALIZE3		DRST, DEND/PO, DALM, STBY —	225 ns	28
H'04					
H'05	FSPD SET		Ꞁ i ☻ — Ꞁ i ☻ —	225 ns	39
H'06	HIGH SPEED SET		♪ ! ▬Ķ — ≤ ⁿ —	225 ns	40
H'07	LOW SPEED SET		♪ ! ▬Ķ — ≤ ǧ —	225 ns	41
H'08	RATE SET		○ ɔ Ķ — —	225 ns	42
H'09	SCAREA SET		○ ɔ Ķ — S —	225 ns	43
H'0A	DOWN PULSE ADJUST		Ꞁ i ☻ — ○ ɦ ⁿ ⁿ ♪	225 ns	44
H'0B					
H'0C	JSPD SET		JOG ♪ ! ▬Ķ — Ꞁ i ☻ —	225 ns	37
H'0D	JOG PULSE SET		JOG ♪ ! ▬Ķ — Ꞁ i ☻ —	225 ns	38
H'0E					
H'0F	ORIGIN SPEC SET		ORIGIN ♪ ! ▬Ķ — لأ ث —	225 ns	50
H'10	+JOG	*P	JOG ♪ ! ▬Ķ —	375 ns	45
H'11	-JOG	*P	JOG ♪ ! ▬Ķ —	375 ns	45
H'12	+SCAN	*P	SCAN ♪ ! ▬Ķ —	375 ns	46
H'13	-SCAN	*P	SCAN ♪ ! ▬Ķ —	375 ns	46
H'14	INC INDEX	*P	▪ ♪ ☻ INDEX ♪ ! ▬Ķ —	375 ns	47
H'15	ABS INDEX	*P	▪ ♪ ☻ INDEX ♪ ! ▬Ķ —	375 ns	48
H'16					
H'17					
H'18	ORIGIN SCAN	*P	ORIGIN SCAN ♪ ! ▬Ķ —	375 ns	52
H'19	ORIGIN CONSTANT SCAN	*P	ORIGIN CONSTANT SCAN ♪ ! ▬Ķ —	375 ns	52
H'1A					
H'1B					
H'1C					
H'1D					
H'1E					
H'1F					
H'20	CP SPEC SET	*1	CPPOUT —	225 ns	54
H'21					
H'22	LONG POSITION SET		♪ ! ▬Ķ — ▪ ♪ ☻ —	225 ns	59
H'23	SHORT POSITION SET		♪ ! ▬Ķ — ▪ ♪ ☻ —	225 ns	60
H'24					
H'25					
H'26					
H'27					
H'28	CIRCULAR XPOSITION SET		♪ ! ▬Ķ — X ▪ ♪ ☻ —	225 ns	68
H'29	CIRCULAR YPOSITION SET		♪ ! ▬Ķ — Y ▪ ♪ ☻ —	225 ns	69
H'2A	CIRCULAR PULSE SET		♪ ! ▬Ķ — Ꞁ i ☻ —	225 ns	70
H'2B					
H'2C					
H'2D					
H'2E					
H'2F					

COMMAND CODE	◖ ∇ fi ♪		( )	PAGE
H'30	MAIN STRAIGHT CP *P	ω▬ fi ♪ ! ▬Ķ ─	375 ns	63
H'31	SUB STRAIGHT CP *P	◘ Ķ ♪ ! ▬Ķ ─	375 ns	62
H'32	MAIN XY STRAIGHT CP *1 *P	ω▬ fi ⌘ ℮ ķ ♪ ! ▬Ķ ─	375 ns	61
H'33				
H'34				
H'35				
H'36				
H'37				
H'38	MAIN CIRCULAR CP *P	ω▬ fi ♪ ! ▬Ķ ─	375 ns	73
H'39	SUB CIRCULAR CP *P	◘ Ķ ♪ ! ▬Ķ ─	375 ns	72
H'3A	MAIN XY CIRCULAR CP *1 *P	ω▬ fi ⌘ ℮ ķ ♪ ! ▬Ķ ─	375 ns	71
H'3B				
H'3C				
H'3D				
H'3E				
H'3F				

*1　2 軸相関 ◖ ∇ fi ♪
*P　Ꞧ ⁱ ☻　╩ق e ◖ ∇ fi ♪

(2) ◖ ∀ fi ♪

COMMAND CODE	◖ ∀ fi ♪		( )	PAGE
H'80	ADDRESS COUNTER PRESET	▪ ♪ ☻ ○ ▶ fi ㅋ — ₀ —	225 ns	103
H'81	ADDRESS COUNTER INITIALIZE1	▪ ♪ ☻ ○ ▶ fi ㅋ — —	225 ns	98
H'82	ADDRESS COUNTER INITIALIZE2	▪ ♪ ☻ ○ ▶ fi ㅋ — —	225 ns	101
H'83				
H'84				
H'85				
H'86				
H'87	ADDRESS COUNTER MAX COUNT SET	▪ ♪ ☻ ○ ▶ fi ㅋ — ○ ▶ fi ♩ —	225 ns	104
H'88	ADRINT COMPARE REGISTER1 SET	ADRINT — ◖ fi ⊏ ▪ ☺ ☻ ㅋ —	225 ns	105
H'89	ADRINT COMPARE REGISTER2 SET	ADRINT — ◖ fi ⊏ ▪ ☺ ☻ ㅋ —	225 ns	105
H'8A	ADRINT COMPARE REGISTER3 SET	ADRINT — ◖ fi ⊏ ▪ ☺ ☻ ㅋ —	225 ns	105
H'8B				
H'8C	ADRINT COMP1 ADD DATA SET	ADRINT — COMP1 ADD ♦ ⊃ ㅋ —	225 ns	106
H'8D				
H'8E				
H'8F				
H'90	PULSE COUNTER PRESET	R ⁱ ☻ ○ ▶ fi ㅋ — ○ ▶ fi ♩ ᄉᆞᆷ —	225 ns	112
H'91	PULSE COUNTER INITIALIZE1	R ⁱ ☻ ○ ▶ fi ㅋ — —	225 ns	107
H'92	PULSE COUNTER INITIALIZE2	R ⁱ ☻ ○ ▶ fi ㅋ — —	225 ns	110
H'93				
H'94				
H'95				
H'96				
H'97	PULSE COUNTER MAX COUNT SET	R ⁱ ☻ ○ ▶ fi ㅋ — ○ ▶ fi ♩ —	225 ns	113
H'98	CNTINT COMPARE REGISTER1 SET	CNTINT — ◖ fi ⊏ ▪ ☺ ☻ ㅋ —	225 ns	114
H'99	CNTINT COMPARE REGISTER2 SET	CNTINT — ◖ fi ⊏ ▪ ☺ ☻ ㅋ —	225 ns	114
H'9A	CNTINT COMPARE REGISTER3 SET	CNTINT — ◖ fi ⊏ ▪ ☺ ☻ ㅋ —	225 ns	114
H'9B				
H'9C	CNTINT COMP1 ADD DATA SET	CNTINT — COMP1 ADD ♦ ⊃ ㅋ —	225 ns	115
H'9D				
H'9E				
H'9F				
H'A0	DFL COUNTER PRESET	R ⁱ ☻ ᄉᆞᆷ ○ ▶ fi ㅋ — ○ ▶ fi ♩ ᄉᆞᆷ —	225 ns	122
H'A1	DFL COUNTER INITIALIZE1	R ⁱ ☻ ᄉᆞᆷ ○ ▶ fi ㅋ — —	225 ns	116
H'A2	DFL COUNTER INITIALIZE2	R ⁱ ☻ ᄉᆞᆷ ○ ▶ fi ㅋ — —	225 ns	119
H'A3	DFL COUNTER INITIALIZE3	R ⁱ ☻ ᄉᆞᆷ ○ ▶ fi ㅋ — —	225 ns	121
H'A4				
H'A5				
H'A6				
H'A7				
H'A8	DFLINT COMPARE REGISTER1 SET	DFLINT — ◖ fi ⊏ ▪ ☺ ☻ ㅋ —	225 ns	123
H'A9	DFLINT COMPARE REGISTER2 SET	DFLINT — ◖ fi ⊏ ▪ ☺ ☻ ㅋ —	225 ns	123
H'AA	DFLINT COMPARE REGISTER3 SET	DFLINT — ◖ fi ⊏ ▪ ☺ ☻ ㅋ —	225 ns	123
H'AB				
H'AC	DFLINT COMP1 ADD DATA SET	DFLINT — COMP1 ADD ♦ ⊃ ㅋ —	225 ns	124
H'AD				
H'AE				
H'AF				
H'B0--H'BF				
H'C0	UDC SPEC SET	UP/DOWN/CONST — ᄅᆞᆯ —	225 ns	75
H'C1	SPEED CHANGE SPEC SET	SPEED CHANGE — ᄅᆞᆯ —	225 ns	77
H'C2				
H'C3	INDEX CHANGE SPEC SET	INDEX CHANGE — ᄅᆞᆯ —	225 ns	80
H'C4	UP DRIVE	UP DRIVE —	225 ns	76
H'C5	DOWN DRIVE	DOWN DRIVE —	225 ns	76
H'C6	CONST DRIVE	CONST DRIVE —	225 ns	76
H'C7				
H'C8	SPEED CHANGE	SPEED CHANGE —	225 ns	78
H'C9				

COMMAND CODE	◖ ∇ fi ♪		( )	PAGE
H'CA	RATE CHANGE	RATE CHANGE —	225 ns	79
H'CB				
H'CC	INC INDEX CHANGE	INC INDEX CHANGE —	225 ns	81
H'CD	ABS INDEX CHANGE	ABS INDEX CHANGE —	225 ns	82
H'CE	PLS INDEX CHANGE	PLS INDEX CHANGE —	225 ns	83
H'CF				
H'D0	INT FACTOR READ	INT FACTOR — ╗ ⇔	225 ns	93
H'D1	ERROR STATUS READ	ERROR STATUS — ╗ ⇔	225 ns	89
H'D2				
H'D3				
H'D4	MCC SPEED READ	♪ ! ▬Ķ R ⁱ ☻ — ╗ ⇔	225 ns	94
H'D5	MCC SET DATA READ	♦ ↄ ㉿ — ╗ ⇔	275 ns	95
H'D6				
H'D7				
H'D8	ADDRESS COUNTER READ	▪ ♪ ☻ ○ ▶ fi ㉿ — ╗ ⇔	225 ns	127
H'D9	PULSE COUNTER READ	R ⁱ ☻ ○ ▶ fi ㉿ — ╗ ⇔	225 ns	127
H'DA	DFL COUNTER READ	R ⁱ ☻ ○ ▶ fi ㉿ — ╗ ⇔	225 ns	127
H'DB				
H'DC	ADDRESS LATCH DATA READ	▪ ♪ ☻ ○ ▶ fi ㉿ — ! ㉿ ♦ ↄ ㉿ — ╗ ⇔	325 ns	128
H'DD	PULSE LATCH DATA READ	R ⁱ ☻ ○ ▶ fi ㉿ — ! ㉿ ♦ ↄ ㉿ — ╗ ⇔	325 ns	128
H'DE	DFL LATCH DATA READ	R ⁱ ☻ ○ ▶ fi ㉿ — ! ㉿ ♦ ↄ ㉿ — ╗ ⇔	325 ns	128
H'DF				
H'E0	INT FACTOR CLR	INT FACTOR — ▾ ⁱ ▪	225 ns	92
H'E1	INT FACTOR MASK	INT ⌐ Δ ╢ INT FACTOR — ∇ ☻ ▾	225 ns	91
H'E2				
H'E3				
H'E4	ERROR STATUS CLR	ERROR STATUS — ▾ ⁱ ▪	225 ns	88
H'E5	ERROR STATUS MASK	ERROR ⌐ Δ ╢ ERROR STATUS — ∇ ☻ ▾	225 ns	86
H'E6	ERRINT STATUS MASK	ERRINT ⌐ Δ ╢ ERROR STATUS — ∇ ☻ ▾	225 ns	87
H'E7				
H'E8	COUNT LATCH SPEC SET	○ ▶ fi ♩ ♦ ↄ ㉿ — ! ㉿ ㉿ ▬ ⊢ fi ◓ —	225 ns	125
H'E9				
H'EA				
H'EB				
H'EC				
H'ED				
H'EE				
H'EF				
H'F0	MCC CHIP RESET	MCC07E — —	475 ns	97
H'F1	HARD INITIALIZE1	OUT A,B —	225 ns	31
H'F2				
H'F3				
H'F4	HARD INITIALIZE4	— ♦ ☺ ㉿ ⁱ ɧ ▫ ⁱ ㉿ —	225 ns	32
H'F5	HARD INITIALIZE5	— ♦ ☺ ㉿ ⁱ ɧ ▫ ⁱ ㉿ —	225 ns	33
H'F6	HARD INITIALIZE6	R ⁱ ☻ — ♦ ☺ ㉿ ⁱ ɧ ▫ ⁱ ㉿ —	225 ns	34
H'F7	HARD INITIALIZE7	ظص كج — ▪ ▾ ♥ ▫ Ķ —	225 ns	35
H'F8				
H'F9				
H'FA				
H'FB				
H'FC	SIGNAL OUT	كج — لأ	225 ns	85
H'FD				—
H'FE	SLOW STOP	تج —	225 ns	84
H'FF	FAST STOP	تج —	225 ns	84

4-4 HARD CONFIG コマンド一覧表

COMMAND CODE	◖ ∀ fi ♪		( )	PAGE
H'00	MAN MASK	MAN ⌚ ⌚ ― ∀ ☻ ▾	200 ns	129
H'01	SENSOR SIGNAL SELECT	☿ fi ◘ ⌚ (SS0,SS1) ⌚ ―	200 ns	130
H'02				
H'03				
H'04	SIGNAL OUT SELECT	SIGNAL OUT ⌚ ―	200 ns	132
H'05	SIGNAL OUT TIMER SET	SIGNAL OUT ―	200 ns	134
H'06	SIGNAL OUT STATUS LATCH CLR	SIGNAL STATUS ― ! ℗ ℔ ℏ ! ◓ ― ▾ ¡ ▪	200 ns	135
H'07				
H'08				
H'09				
H'0A				
H'0B				
H'0C				
H'0D				
H'0E				
H'0F				
H'10	PAUSE	PAUSE ― SET/CLR ⌚	200 ns	140
H'11	PAUSE SET SPEC	PAUSE ― SET ⌚	200 ns	136
H'12	PAUSE CLR SPEC	PAUSE ― CLR ⌚	200 ns	138
H'13				
H'14				
H'15				
H'16				
H'17				
H'18				
H'19				
H'1A				
H'1B				
H'1C				
H'1D				
H'1E				
H'1F				
H'20	HARD CONFIG SET DATA READ	HARD CONFIG ♦ ↄ ㉿ ― ⌅ ⇔	200 ns	141
H'21				
H'22				
H'23				
H'24				
H'25				
H'26				
H'27				
H'28				
H'29				
H'2A				
H'2B				
H'2C				
H'2D				
H'2E				
H'2F				
H'30_H'EF				
H'F0	HARD CONFIG RESET	HARD CONFIGURATION ― ―	200 ns	141
H'F1_H'FF				

4-5. ボード仕様一覧

一般仕様

項目	C-VX870v1	C-VX871v1	C-VX872v1	C-VX873v1	C-VX870Ev1	C-VX871Ev1	C-VX872Ev1	C-VX873Ev1
制御対象	ステッピングモータ、およびサーボモータドライバユニット(パルス列入力方式)							
制御軸数	4 軸	6 軸	8 軸	12 軸	4 軸	6 軸	8 軸	12 軸
搭載PG	MCC07E(弊社製)							
PC 消費電流	5V,1.0A以下	5V,1.2A以下	5V,1.6A以下	5V,1.8A以下	3.3V,1.4A以下	3.3V,1.6A以下	3.3V,2.3A以下	3.3V,2.5A以下
I/F 消費電流 +24V	200mA以下	250mA以下	400mA以下	500mA以下	200mA以下	250mA以下	400mA以下	500mA以下
使用条件	0-45 ℃、80 % RH 以下(非結露)							
PCI バス仕様	PCI Local Bus Specification Rev2.3 32bit, 33MHz, 3.3V/5V ユニバーサル対応 (このボードは拡張スロットから+5V 電源の供給が必要です。)				PCI Express Base Specification Rev2.1			
システムリソース I/O 領域	128 バイト＋ 256 バイト占有				128 バイト＋256 バイト占有			
割り込み	1 点(INTA#)							
外形寸法(mm)	107mm × 170mm × 17mm				107mm × 170mm × 17mm			
ユーザI/O コネクタ	DX10A -100S(50) (ヒロセ電機製)		HDRA-E100W1LFDT1EC-SL (本多製)		DX10A -100S(50) (ヒロセ電機製)		HDRA-E100W1LFDT1EC-SL (本多製)	
特殊I/O コネクタ	MIL 20P							
質量	約0.2kg							

パルス出力

項目	C-VX870v1	C-VX871v1	C-VX872v1	C-VX873v1	C-VX870Ev1	C-VX871Ev1	C-VX872Ev1	C-VX873Ev1
パルス出力	独立出力/方向指定出力/位相差信号出力 ラインドライバ出力(非絶縁:AM26LS31 相当)							
出力周波数	0.1Hz ～ 6.5 MHz(独立ドライブ時)							
出力電流	± 20mA							
加減速時定数	5000ms/1kHz ～ 0.0025ms/1kHz(台形/S 字)							
加減速形状	台形/S 字							

エンコーダ入力信号

項目	C-VX870v1	C-VX871v1	C-VX872v1	C-VX873v1	C-VX870Ev1	C-VX871Ev1	C-VX872Ev1	C-VX873Ev1
EA/EB 相	○(各軸)	× *1	○(各軸)	× *1	○(各軸)	× *1	○(各軸)	× *1
	エンコード形式 : インクリメンタル(位相差信号入力/独立入力) 入力回路 : ラインレシーバ入力(非絶縁:AM26LS32 相当) 終端抵抗 : 220 Ω 入力周波数 : ～ 5MHz 信号延長距離 : 10m(差動出力と接続)							
Z 相	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)
	入力回路 : ラインレシーバ入力(非絶縁:AM26LS32 相当) 終端抵抗 : 220 Ω 入力周波数 : ～ 100kHz 信号延長距離 : 10m(差動出力と接続)							

*1 6軸,12軸の製品は、エンコーダEA/EB 相入力はありません。

センサ信号入力

項目	C-VX870v1	C-VX871v1	C-VX872v1	C-VX873v1	C-VX870Ev1	C-VX871Ev1	C-VX872Ev1	C-VX873Ev1
$\overline{\text{ORG}}$ 信号	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)
$\overline{\text{NORG}}$ 信号	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)
CWLM 信号	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)
CCWLM 信号	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)
$\overline{\text{SENSORn0}}$ 信号	○	○	○ *1	○ *1	○	○	○ *1	○ *1
$\overline{\text{SENSORn1}}$ 信号	○	○	○ *1	○ *1	○	○	○ *1	○ *1
各信号共通	入力回路 : フォトカプラ入力(絶縁) 応答時間 : 1ms 入力抵抗 : 6.8k Ω 入力ON 電流 : 2.5mA 以上 入力OFF 電流 : 0.8mA 以下 外部I/F 電源 : +24VDC(± 10%)							

*1 8軸,12軸の製品は100 ピンコネクタ毎に各1 点あります。

制御用入力/汎用入力信号

項目	C-VX870v1	C-VX871v1	C-VX872v1	C-VX873v1	C-VX870Ev1	C-VX871Ev1	C-VX872Ev1	C-VX873Ev1
DEND/PO 信号	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)
FSSTOPn 信号	○	○	○ *1	○ *1	○	○	○ *1	○ *1
RESETn 信号	○	○	○ *1	○ *1	○	○	○ *1	○ *1
INn0--INn3 信号	○	× *2	○ *1	× *2	○	× *2	○ *1	× *2
各信号共通	入力回路 : フォトカプラ入力(絶縁) 応答時間 : RESET 信号以外 1ms : RESET 信号 5ms 入力抵抗 : 6.8k Ω 入力ON 電流 : 2.5mA 以上 入力OFF 電流 : 0.8mA 以下 外部I/F 電源 : +24VDC(± 10%)							

*1 8軸,12軸製品は、同機能の信号が100 ピンコネクタ毎に各1 点あります。
*2 6軸,12軸製品は、汎用入力はありません。

制御用出力/汎用出力信号

項目	C-VX870v1	C-VX871v1	C-VX872v1	C-VX873v1	C-VX870Ev1	C-VX871Ev1	C-VX872Ev1	C-VX873Ev1
DRST 信号	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)	○(各軸)
OUTn0--OUTn3 信号	○	× *2	○ *1	× *2	○	× *2	○ *1	× *2
各信号共通	出力回路 : トランジスタオープンコレクタ出力(フォトカプラ絶縁) 応答時間 : 1ms 出力ON 電流 : 30mA (Vce=1V 以下) 出力OFF 電流 : 0.1mA 以下 外部I/F 電源 : +24VDC(± 10%)							

*1 8軸製品は、同機能の信号が100 ピンコネクタ毎に各1 点あります。
*2 6軸,12軸製品は、汎用出力はありません。

MANUAL 機能/特殊I/O 入力信号

項目	C-VX870v1	C-VX871v1	C-VX872v1	C-VX873v1	C-VX870Ev1	C-VX871Ev1	C-VX872Ev1	C-VX873Ev1
MAN,CWMS,CCWMS SS0,SS1 信号	○	○	○	○	○	○	○	○
SIGNAL INnx 信号	○ (2 点)	○ (2 点)	○ (4 点)	○ (4 点)	○ (2 点)	○ (2 点)	○ (4 点)	○ (4 点)
FSSTOP 信号	○	○	○	○	○	○	○	○
各信号共通	入力回路 : TTL レベルCMOS シュミット入力(非絶縁) 応答時間 : MAN,CWMS,CCWMS 信号 5ms 以下 SS0,SS1,FSSTOP 信号 1ms 以下 SIGNAL INnx 信号 10 μs 以下 ローレベル : 0.8V 以下 ハイレベル : オープン インターフェイス電源 : ５ V							

SIGNAL IN 信号は、 MANUAL 動作時の軸選択信号(SEL 信号)と兼用です。

特殊I/O 出力信号

項目	C-VX870v1	C-VX871v1	C-VX872v1	C-VX873v1	C-VX870Ev1	C-VX871Ev1	C-VX872Ev1	C-VX873Ev1
SIGNAL OUTnx 信号	○ (2 点)	○ (2 点)	○ (4 点)	○ (4 点)	○ (2 点)	○ (2 点)	○ (4 点)	○ (4 点)
各信号共通	出力回路 : オープンコレクタ出力(非絶縁) 応答時間 : 1 μs 以下 出力ON 電流 : 10mA(Vce=0.6V 以下) 出力OFF 電流 : 0.3mA 以下 インターフェース電圧 : +30V 以下							

５．トラブルシューティング

No.	現　象	チェックポイント
1.	アプリケーションからコマンドを実行 しているが受け付けない。	・ボードがスロットに差し込まれているか確認して ください。 ・ボード番号の設定を確認してください。 ボード番号のスイッチ設定が重複していないか確認 してください。 ・LIMIT 信号、および FSSTOP 信号の配線を確認して ください。 上記信号は未使用のときでも GND に接続されていない とパルス出力を行いません。 ・STATUS1 内の ERROR BIT を調べてください。 ERROR BIT が 1 の時は汎用コマンドを受け付けません ERROR STATUS CLR コマンドを実行して ERROR BIT を 0 にクリアしてください。
2.	アクセスは正常に行われているようだが ドライブコマンドを書き込んでもドライブ が行われない。 この時 STATUS 内 DRIVE BIT,BUSY BIT が共に 0 である。	・STATUS1 内の ERROR BIT を調べてください。 ERROR BIT が 1 の時は汎用コマンドを受け付けません ERROR STATUS CLR コマンドを実行して ERROR BIT を 0 にクリアしてください。 ・出力 PULSE が 0 の INDEX DRIVE ではありませんか？ （指定した絶対 ADDRESS が現在位置の場合など）
3.	ドライブを開始したが、いつまでも ドライブが終了しない。	・SCAN,ORIGIN ドライブではありませんか？ ・INDEX ドライブの場合 INCREMENTAL 指定の時 … 設定された PULSE 数が多い。 ABSOLUTE 指定の時 …… 設定された ADDRESS が遠い。 と思われます。この場合はいずれ停止します
4.	ドライブのパルス出力は終了したが、 いつまでも STATUS 内 BUSY BIT が 0 に ならない。	・$\overline{\text{DEND}}$/$\overline{\text{PO}}$ 信号のサーボ対応が設定されており、 $\overline{\text{DEND}}$/$\overline{\text{PO}}$ 信号が戻って来ていない状態では ありませんか？ $\overline{\text{DEND}}$/$\overline{\text{PO}}$ 信号が ON になることにより STATUS1 内の BUSY BIT は 0 となります。 ・$\overline{\text{DRST}}$ 信号のサーボ対応が設定されており、 $\overline{\text{DRST}}$ 信号がサーボドライバと未接続状態では ありませんか？ $\overline{\text{DRST}}$ 信号がサーボドライバと未接続の場合は、 $\overline{\text{DRST}}$ 信号の出力機能をサーボ対応には設定しないで ください。
5.	機械原点検出(ORIGIN ドライブ)が 正常にできない。 または、いつまでたっても終了しない。	・センサの論理(入光時 ON 、あるいは入光時 OFF)は 合っていますか？ ・センサの接続(特に GND ライン)は合っていますか？ ・サーボ対応が設定されており、$\overline{\text{DEND}}$/$\overline{\text{PO}}$ 信号が 戻って来ていない状態ではありませんか？
6.	読み出したカウンタ値が正しくない。	・読み出したい COUNTER の READ コマンドを実行して いますか？ 読み出す際は読み出したい COUNTER の READ コマンドを都度実行する必要があります。
7.	読み出したドライブパルス速度が正しく ない	・SPEED READ コマンドを実行していますか？ 読み出す際は SPEED READ コマンドを都度実行する 必要があります。 ・読み出されるデータは実際速度の 10 倍の値です。

No.		⌘ ▼ ℘ ▾ w▬ fi ♩
8.	ADRINT,CNTINT,DFLINT ⅜ ⇔ √ ثم ≤ ⌂ ╢ ○ ► fi ㉺ ثم ≠ ⇔ ≡ ™ ╢ ≠ №╢ ⁹	ה DATA — COMPARE REGISTER ⅜ ⇔ ⁸ ⌐ COUNTER — ○ ► fi ㉺ ثم ⅜ ○ ↄ ₱ ɦ ↄ ⇔ ≡ ™ ╕ ∑ ╪ ⅛ COMPARE REGISTER │ ⁸ ⁱ ☿ ℘ ♩ ○ ↄ ₱ ɦ ↄ ثم ≤ ∂ ⌐ ↕ ╢ ╢ √ ╘ DATA — COMPA RE REGISTER ⅜№╢ ≤ ○ ↄ ₱ ɦ ↄ ثم ≠ ○ ► fi ㉺ INT كج ╩ ⇔ ╕ Δ ⁹ ئو — COMPARE REGISTER — COMP INT │ ⁸ COUNTER INITIALIZE COMMAND ≠ ⇔ ≡ ↄ ∞ ↕ ™ ⁹
9.	R ⁱ ☻ ⅜ ثم ≤ ⌂ ∫ ≡ ™ ╢ ≠ №╢ ⁹	ה — ╩ ⇔ √ ⁸ ثم ≤ — ثم ⅜ ⌂ ╢ ⅜№╡ ╕ Δ ⁹
10.	R ⁱ ☻ — ⅜ ثم ≤ ∫ ≡ ™ ╢ ≠ №╢ ⁹	ה 3-1-3. ♪ ! ▬ Ꝗ R ! ϖ ↄ ㉺ ╩ ⇔ ⁸ ⌐ ≠ ™ ≡ ╩ ⇔ ≡ ↄ ∞ ↕ ™ ⁹
11.	R ⁱ ☻ — ⅜ ⇔ √ ⌐ ⌂ ╟ ⌂ ™ ⁹	ה INDEX DRIVE — ⁸ INDEX ⅜ ⌂ ™ ≤ ⌐ ⇔ ⌂ ™ ⅜№╡ ╕ Δ ⁹
12.	طص كج طص⅜ ╢ ≤ ⁸ ♪ ! ▬ ₱ ⅜ تج ⇔ ≡ ⇔ ╕ e ⁹	ה MCC07E — DALM — ⅜ تج ⌐ ↕ ╢ ≡ ™ ╕ ∑ ╪ ⅛ طص ≤ ⇔ ≡ ئو Δ ╢ ≤ ⅝ │ ⁸ طص ⌐ ⇔ ⁸ طص — │ ⁸ طص PORT ≠ Δ ╢ ╟ e ⌐ ⇔ ≡ ↄ ∞ ↕ ™ ⁹

本版で改訂された主な箇所

箇　　所	内　　容
P2,9,11,12, 21,22,28,29, 53,104,130, 131,132,133, 134,135,136, 137,138,139, 140,142,145, 159,160,174, 192,193,194, 206,213,214, 215	【R1】 C-VX872Ev1, C-VX873Ev1　追加

■製品保証

保証期間と保証範囲について

● 納入品の保証期間は、納入後２ヶ年と致します。
● 上記保証期間中に当社の責により故障を生じた場合は、その修理を当社の責任において行います。
（日本国内のみ）
ただし、次に該当する場合は、この保証対象範囲から除外させて頂きます。
(1) お客様の不適当な取り扱い、ならびに使用による場合。
(2) 故障の原因が、当製品以外からの事由による場合。
(3) お客さまの改造、修理による場合。
(4) 製品出荷当時の科学・技術水準では予見が不可能だった事由による場合。
(5) その他、天災、災害等、当社の責にない場合。
(注1)ここでいう保証は、納入品単体の保証を意味するもので、納入品の故障により誘発される損害はご容赦頂きます。
(注2)当社において修理済みの製品に関しましては、保証外とさせて頂きます。

技術相談のお問い合わせ
販売に関するお問い合わせ

TEL.(042)664-5384　FAX.(042)666-2031
E-mail　s-support@melec-inc.com

株式会社 メレック
〒193-0834 東京都八王子市東浅川町516-10
www.melec-inc.com

記載内容は、製品改良のため予告なく変更することがありますのでご了承ください。
C2501