



() .
NS System Co., Ltd..

Unc-Series, Lnc-Series
Library

Users' Manual

1.

1.1.

() .

Serial

Computer

?

LNC- 1, 2, 4

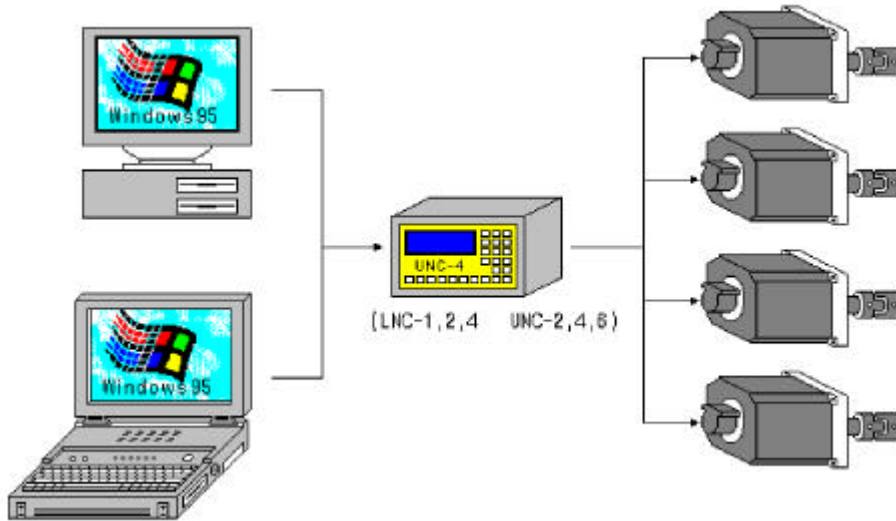
UNC-2, 4

Computer

RS232C

Parameter

1.2.



1.3.

1.3.1.

Windows 95,98

486

Computer

1.3.2. Software

A. OS Software

Microsoft Windows 95,98

Windows NT(INTEL MACHINE)

B. Software — VER 1.0

(2HD×2, 3.5 FDD)

OPTION

C.

a. Visual C / C++

b. Visual Basic

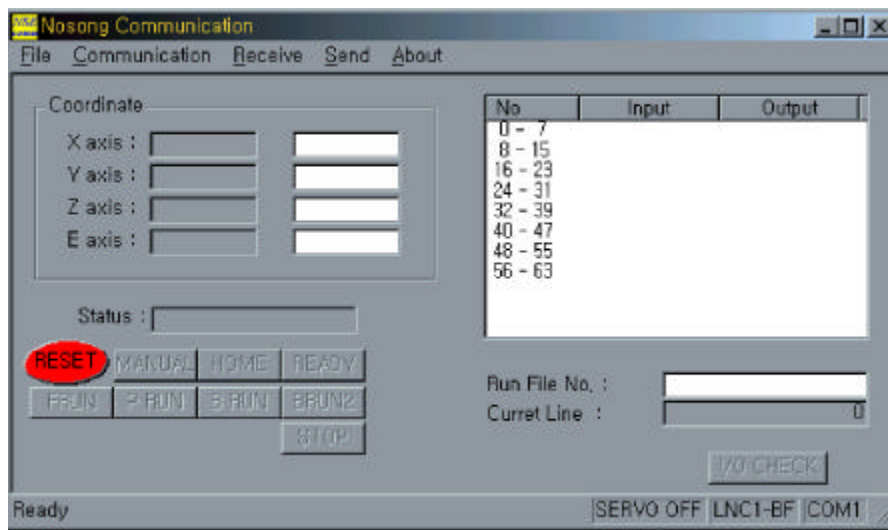
1.3.3. Controller

A. Controller PC RS232C Connector .()

B. Controller “ 18 - RS232 ” “ 1 65535 ”

C. Reset

2.



2.1.

File :
 Communication : ID 가
 Receive :
 Send :
 About :

2.2.

Coordinate : (BRUN2
 Status :
 I/O View Box :
 Run File No. : FRUN
 Current Line :

2.3.

RESET :
 MANUAL :
 HOME :
 READY :
 FRUN : Run File No.
 PRUN :
 BRUN : PC Buffer Run
 BRUN2 : Coordinate Buffer Run
 STOP :

2.4.

SERVO OFF / ON : -
 LNC1-BF :
 COM1 :

3.

3.1. Library ()

STATUS			
check_controller		CHECK	.
close_comm_port	comm port		.
get_cnt			.
get_cnt_bas		BUFFER	.
get_cnt2		BUFFER	.
get_comm_status			.
get_coordinate			.
get_coordinate_bas		BUFFER	.
get_line_no		File Step	.
get_pgm_data	PGM DATA	file	.
get_pos		BUFFER	.
get_ready		On	.
get_scale	SET- UP	SCALE(Pulse/mm)	.
get_scale_bas	SET- UP	SCALE(Pulse/mm) BUFFER	.
get_scale2	SET- UP	SCALE(Pulse/mm) BUFFER	.
get_setup_data	SET- UP DATA	file	.
get_status2			.
get_status2_bas			.
get_work_cnt	counter		.
open_comm_port			.
set_buffer_data		controller brun buffer	.
set_buffer_data_bas		controller brun buffer	.
set_pgm_data	PGM DATA file	controller	.
set_pgm_data_bas	PGM DATA file	controller	.
set_setup_data	SET- UP DATA	controller	.
set_setup_data_bas	SET- UP DATA	controller	.
brun_cmd	BUFFER RUN		.
frun_cmd	PC		.
goto_xy			.
home_cmd	(HOME)		.
lnc 1_goto_x		(Lnc 1)	.
prun_cmd			.
reset_cmd	RESET		.
restart_cmd	Resear		.
stop_cmd	STOP		.
jog_cmd	(JOG)		.
manual_cmd			.
manual_in_cmd			.
manual_out_cmd			.

manual_stop_cmd

/
get_io
get_io_bas BUFFER
get_io2 BUFFER
out_off_cmd OFF
out_on_cmd ON
outport_cmd
outport_cmd bas
servo_cmd Servo On/Off

3.2. Library ()

brun_cmd

long brun_cmd (long serial)

Return Value

bstatus (button status)(2)

PARAMETER

serial (controller ID)(1)

BUFFER RUN .(READY)

check_controller

long check_controller (long serial)

Return Value

conID (controller ID)

PARAMETER

serial (controller ID)(1)

CHECK .

close_comm_port

long close_comm_port()

Return Value

PARAMETER

IO port

frun_cmd

long frun_cmd (long serial, long file no)

Return Value

bstatus (button status)(2)

PARAMETER

serial (controller ID)(1)

file no (file : 0 9) / SNC : file no (file : 0 15)

PNC-3 .(READY)

file no File

SNC filno POINT POINT

get_cnt

long get_cnt (long serial, long *cnt)

Return Value

COMOK : OK

COMERR : ERROR

PARAMETER

serial (controller ID)(1)

cnt[8] (counter value)

cnt[0] cnt[7]

get_cnt_bas

long stdcall get_cnt_bas (long serial)

Return Value

COMOK : OK

COMERR : ERROR

PARAMETER

serial (controller ID)(1)

BUFFER . (Vbasic)

get_cnt2

long stdcall get_cnt2 (long inx)

Return Value

COMOK : OK
COMERR : ERROR

PARAMETER

inx (index no. of input/output), (0 - 7)

BUFFER

get_cnt_bas() . (Vbasic)

get_comm_status

unsigned char get_comm_status()

Return Value

COMOK : OK, COMERR : ERROR, COMNOAMS :

PARAMETER

get_coordinate

long get_coordinate (long serial, long *pos)

Return Value

COMOK : OK
COMERR : ERROR

PARAMETER

serial (controller ID)(1 - 10)
pos[4] (0 : x , 1 : y , 2 : z , 3 : data)

X, Y, Z

get_coordinate_bas

long stdcall get_coordinate_bas (long serial)

Return Value

COMOK : OK
COMERR : ERROR

PARAMETER

serial (controller ID)(1 - 10)

X, Y, Z BUFFER . (Vbasic)

get_io

long get_io (long serial, unsigned char *iobuf)

Return Value

COMOK : OK

COMERR : ERROR

PARAMETER

serial (controller ID)(1)

iobuf[16] (input(0- 7), output(8- 15))

iobuf[0], iobuf[8] DATA7
iobuf[0] , iobuf[8]

get_io_bas

long stdcall get_io_bas (long serial)

Return Value

COMOK : OK

COMERR : ERROR

PARAMETER

serial (controller ID)(1)

BUFFER . (Vbasic)

get_io2

long stdcall get_io2 (long inxt)

Return Value

COMOK : OK

COMERR : ERROR

PARAMETER

inx (index no. of input/output), (0 : , 8 :)

BUFFER .
get_io_bas()

get_line_no

long get_line_no (long serial, long *file no, long *line no)

Return Value

COMOK : OK

COMERR : ERROR

PARAMETER

serial (controller ID)(1)
file no (file)
line no (line)

File Step

get_pgm_data

long get_pgm_data (long serial, char *filename, long file no)

Return Value

0 : ERROR, 1 : OK

PARAMET

serial (controller ID)(1)
filename (file name)(4)
file no (file no. (0-9))

PGM DATA file . (READY)

get_pos

long stdcall get_pos (long inx)

Return Value

PARAMETER

inx (0 : x , 1 : y , 2 : z)

X, Y, Z BUFFER
get_coordinate_bas()

get_ready

long get_ready (long serial)

Return Value

- 1 : error
0 : auto servo on
1 : manual servo on

PARAMETER

serial (controller ID)(1)

On .(READY)
" 01- READY "
0 : , , On 가 .

1 : , , (READY) On 가 .

get_scale

long get_scale (long serial, long *scale)

Return Value

COMOK : OK

COMERR : ERROR

PARAMETER

serial (controller ID)(1)

scale[4] (0 : x , 1 : y , 2 : z , 3 : data)

SET- UP X, Y, Z SCALE(Pulse/mm) .

get_scale_bas

long stdcall get_scale_bas (long serial)

Return Value

COMOK : OK

COMERR : ERROR

PARAMETER

serial (controller ID)(1)

SET- UP X, Y, Z SCALE(Pulse/mm) BUFFER .(Vbasic)

get_scale2

long stdcall get_scale2 (long inx)

Return Value

SCALE(Pulse/mm)

PARAMETER

inx (0 : x , 1 : y , 2 : z)

SET- UP X, Y, Z SCALE(Pulse/mm) BUFFER .
get_scale_bas() .

get_setup_data

long get_setup_data (long serial, char *file name)

Return Value

0 : ERROR, 1 : OK

PARAMETER

serial (controller ID)(1)
filename (file name)(3)

SET-UP DATA file . (READY)

get_status2

long get_status2 (long serial, char *status)

Return Value

- 1 : COMM ERROR
0 : RUN
1 : RUN STOP
2 : HOME
3 : MANUAL
4 : READY
5 : HOME STOP
6 : MANUAL JOG
7 : ERROR

PARAMETER

serial (controller ID)(1)
status (status string)

get_status2_bas

long get_status2_bas (long serial, char *status)

get_status2 Vbasic .

get_work_cnt

long get_work_cnt (long serial, long *wcnt)

Return Value

COMOK : OK
COMERR : ERROR

PARAMETER

serial (controller ID)(1)
wcnt (work counter value)

counter .

goto_xy

long goto_xy (long serial, long x, long y)

Return Value

0 : ERROR, 1 : OK

PARAMETER

serial (controller ID)(1)

x (x)

y (y)

goto_xy2

long goto_xy2 (long acc, long speed, long serial, long x, long y)

Return Value

0 : ERROR, 1 : OK

PARAMETER

acc (가)

speed ()

serial (controller ID)(1)

x (x)

y (y)

home_cmd

long home_cmd (long serial)

Return Value

bstatus (button status)(2)

PARAMETER

serial (controller ID)(1)

(HOME) .(READY)

jog_cmd

long jog_cmd (long serial)

Return Value

bstatus (button status)(2)

PARAMETER

serial (controller ID)(1)

(JOG) .()

lnc 1_goto_x

long lnc 1_goto_x (long serial, long speed, long acc, long x)

Return Value

0 : ERROR, 1 : OK

PARAMETER

serial (controller ID)(1)

speed :

acc : 가

x (x)

. (Lnc 1)

manual_cmd

long manual_cmd (long serial, long updown, long axis)

Return Value

0 : OK, 1 : ERROR, 2 : ERROR,

3 : 가

PARAMETER

serial (controller ID)(1)

updown (up or down direction) ('U' or 'D')

axis (axis selection) (0 : X , 1 : Y , 2 : Z , 3 : data)

.()

manual_in_cmd

long manual_in_cmd (long serial, long *axisuse)

Return Value

MANUAL : OK

: ERROR

PARAMETER

serial (controller ID)(1)

axisuse (axis use flag) 1 : , 0 :

bit 1 (E AXIS)

bit 2 (X AXIS)

bit 3 (Y AXIS)

bit 4 (Z AXIS)

.(READY)

manual_out_cmd

long manual_out_cmd (long serial)

Return Value

COMOK : OK

COMERR : ERROR

PARAMETER

serial (controller ID)(1)

.()

manual_stop_cmd

long manual_stop_cmd (long serial)

Return Value

COMOK : OK

COMERR : ERROR

PARAMETER

serial (controller ID)(1)

.()

open_comm_port

long open_comm_port(long portno)

Return Value

- 1 : ERROR, portno : OK

PARAMETER

portno(com1=0, com2=1)

open_comm_port2

long open_comm_port2(long portno)

Return Value

- 1 : ERROR, portno : OK

PARAMETER

portno(com1=0, com2=1)

. (RNC)

out_off_cmd

long_stdcall out_off_cmd (long serial, long outno)

Return Value

COMOK : OK

COMERR : ERROR

PARAMETER

serial (controller ID)(1)

outno (output port number)

OFF .

out_on_cmd

long_stdcall out_on_cmd (long serial, long outno)

Return Value

COMOK : OK

COMERR : ERROR

PARAMETER

serial (controller ID)(1)

outno (output port number)

ON .

outport_cmd

long outport_cmd (long serial, unsigned char *outbuf)

Return Value

COMOK : OK

COMERR : ERROR

PARAMETER

serial (controller ID)(1)

outbuf[8] (output port buffer)

outbuf[0] DATA .(.)

outport_cmd bas

long stdcall outport_cmd bas (long serial, long inx, long outbuf)

Return Value

COMOK : OK
COMERR : ERROR

PARAMETER

serial (controller ID)(1)
outbuf (output port buffer)
inx (index no. of output)
.
.(Vbasic)

prun_cmd

long prun_cmd (long serial)

Return Value

bstatus (button status)(2)

PARAMETER

serial (controller ID)(1)
.
.(READY)
" 00- FILE # " File .
SNC SEL POINT POINT

reset_cmd

long reset_cmd (long serial)

Return Value

bstatus (button status)(2)

PARAMETER

serial (controller ID)(1)
RESET .()

restart_cmd

long restart_cmd (long serial)

Return Value

bstatus (button status)(2)

PARAMETER

serial (controller ID)(1)

Reset (STOP)

servo_cmd

long servo_cmd (long serial, long cmd)

Return Value

COMOK : OK

COMERR : ERROR

PARAMETER

serial (controller ID)(1)

cmd (ON / OFF)

Servo On/Off (READY)

set_buffer_data

long set_buffer_data (long serial, char *filename, char *errmsg)

Return Value

0 : ERROR, 1 : OK

PARAMETER

serial (controller ID)(1)

filename (file name : 8)(4)

errmsg[100] (error message string)

controller brun buffer (READY)
brun_cmd

set_buffer_data_bas

long set_buffer_data_bas (long serial, char *filename, char *errmsg)

set_buffer_data Vbasic

set_pgm_data

long set_pgm_data (long serial, char *filename, long file no, char *errmsg)

Return Value

0 : ERROR, 1 : OK

PARAMETER

serial (controller ID)(1)
filename (file name)(4)
fileno (file no. (0-9))
emsg[100] (error message string)

PGM DATA file controller . (READY)

set_pgm_data_bas

long set_pgm_data_bas (long serial, char *filename, long fileno, char *emsg)

set_pgm_data Vbasic .

set_setup_data

long set_setup_data (long serial, char *filename, char *emsg)

Return Value

0 : ERROR, 1 : OK

PARAMETER

serial (controller ID)(1)
filename (file name)(3)
emsg[100] (error message string)

SET-UP DATA controller . (READY)

set_setup_data_bas

long set_setup_data_bas (long serial, char *filename, char *emsg)

set_setup_data Vbasic .

stop_cmd

long stop_cmd (long serial)

Return Value

bstatus (button status)(2)

PARAMETER

serial (controller ID)(1)

STOP . (RUN, HOME)

1) Serial (controller ID) 1 65535 , " 18- RS232 "

2) bstatus output label RUN=0, STOP=1, HOME=2, MANUAL=3, READY=4, STOPH=5, SYSEERROR=6, INVALID=7 8가

3) IPM DATA EDIT

```
FILE#      0 ;
READY      0 ;
PULSE      0 ;
RANGE      1 ;
MANSP      20 ;
JOGSP      100 ;
MACCT      10 ;
ESSPD      200 ;
ESACC      10 ;
ESDIM      0 ;
EUNIT      0 ;
HTIME      0 ;
HTYPE      0 ;
SVDLY      1 ;
ONDLY      1 ;
XXXXX      0 ;
```

● ()
● (+)
● (-)
● ()
● (SET- UP)
ERROR가

4) PGM DATA EDIT

```
● (大)
```

```
G91 ;
F00      500 ;
G08      10 ;
X00      1000 ;
Y00      1000 ;
Z00      2000 ;
G00 ;
G04      50 ;
X00      -1000 ;
Y00      -1000 ;
Z00      -2000 ;
G00 ;
G04      100 ;
M50      1 ;
```

● ()
● (+)
● (-)
● ()

5) ERROR NO. 0 49 SYSTEM ERROR

● ERROR NO. 50 98 50 (SYNTAX ERROR) ERROR
● ERROR NO. 100 50 ERROR

4. (Cable)

