Smarter. Greener. Together.

Industrial Automation Headquarters

Delta Electronics, Inc.
Taoyuan Technology Center
No.18, Xinglong Rd., Taoyuan City,
Taoyuan County 33068, Taiwan
TEL: 886-3-362-6301 / FAX: 886-3-371-6301

Asia

Delta Electronics (Jiangsu) Ltd.
Wujiang Plant 3
1688 Jiangxing East Road,
Wujiang Economic Development Zone
Wujiang City, Jiang Su Province, P.R.C. 215200
TEL: 86-512-6340-3008 / FAX: 86-769-6340-7290

Delta Greentech (China) Co., Ltd.
238 Min-Xia Road, Pudong District,
ShangHai, P.R.C. 201209
TEL: 86-21-58635678 / FAX: 86-21-58630003

Delta Electronics (Japan), Inc.
Tokyo Office
2-1-14 Minato-ku Shibadaimon,
Tokyo 105-0012, Japan
TEL: 81-3-5733-1111 / FAX: 81-3-5733-1211

Delta Electronics (Korea), Inc.
1511, Byucksan Digital Valley 6-cha, Gasan-dong,
Geumcheon-gu, Seoul, Korea, 153-704
TEL: 82-2-515-5303 / FAX: 82-2-515-5302

Delta Electronics Int'l (S) Pte Ltd.
4 Kaki Bukit Ave 1, #05-05, Singapore 417939
TEL: 65-6747-5155 / FAX: 65-6744-9228

Delta Electronics (India) Pvt. Ltd.
Plot No 43 Sector 35, HSIIDC
Gurgaon, PIN 122001, Haryana, India
TEL : 91-124-4874900 / FAX : 91-124-4874945

Americas

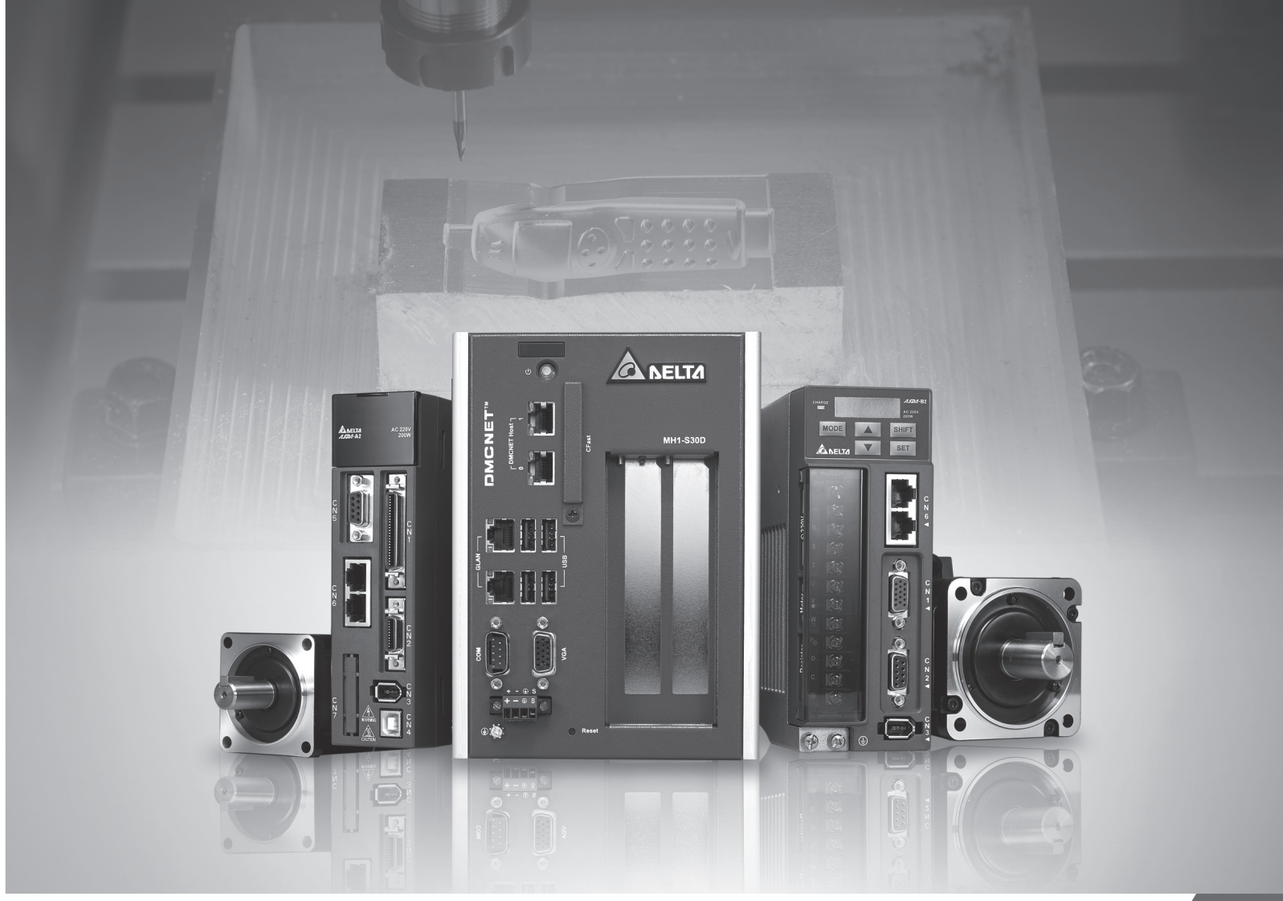
Delta Products Corporation (USA)
Raleigh Office
P.O. Box 12173,5101 Davis Drive,
Research Triangle Park, NC 27709, U.S.A.
TEL: 1-919-767-3800 / FAX: 1-919-767-8080

Delta Greentech (Brasil) S.A.
Sao Paulo Office
Rua Itapeva, 26 - 3° andar Edificio Itapeva One-Bela Vista
01332-000-São Paulo-SP-Brazil
TEL: 55 11 3568-3855 / FAX: 55 11 3568-3865

Europe

Deltronics (The Netherlands) B.V.
Eindhoven Office
De Witbogt 20, 5652 AG Eindhoven, The Netherlands
TEL : +31-40-2592850 / FAX : +31-40-2592851

V1.0
DELTA_IA-IPC_SNC-ISO_PM_EN_20150313



Delta ISO SNC programming User Manual

www.deltaww.com

Smarter. Greener. Together.

About this Manual

User Information

Please store this manual in a safe location.

This manual is subject to change without notice due to the release of new products, improvements and changes in technologies or modifications to data and forms.

Trademarks

Windows NT/2000/XP, Visual Studio, Visual C++, and Visual BASIC are all registered trademarks owned by Microsoft.

BCB (Borland C++ Builder) is a registered trademark owned by Borland.

The names of other products are only used for identification purposes and the registered trademarks remain the property of their respective owners.

Technical Support and Service

If you require technical support, service and related information or have any questions during the use of this product, please visit our website (http://www.delta.com.tw/industrialautomation) or contact us. We look forward to providing the best possible support and service for your needs. Our contact details are provided below:

ASIA
DELTA ELECTRONICS, INC.
Taoyuan Plant 1
31-1, XINGBANG ROAD,
GUISHAN INDUSTRIAL ZONE,
TAOYUAN COUNTY 33370, TAIWAN, R.O.C.
TEL: 886-3-362-6301
FAX: 886-3-362-7267

JAPAN
DELTA ELECTRONICS (JAPAN), INC.
Tokyo Office
DELTA SHIBADAIMON BUILDING
2-1-14 SHIBADAIMON, MINATO-KU,
TOKYO, 105-0012, JAPAN
TEL: 81-3-5733-1111
FAX: 81-3-5733-1211

NORTH/SOUTH AMERICA
DELTA PRODUCTS CORPORATION (USA)
Raleigh Office
P.O. BOX 12173
5101 DAVIS DRIVE,
RESEARCH TRIANGLE PARK, NC 27709,
U.S.A.
TEL: 1-919-767-3813
FAX: 1-919-767-3969

EUROPE
DELTRONICS (THE NETHERLANDS) B.V.
Eindhoven Office
DE WITBOGT 15, 5652 AG EINDHOVEN,
THE NETHERLANDS
TEL: 31-40-259-2850
FAX: 31-40-259-2851

Table of Contents

10 Caution Note

11 Special Parameters

12 G Code Supporting Table

(This page is intentionally left blank.)

1 DMC_SNC Introduction

Delta's DMC_SNC is a dynamic-link library designed for computer numerical control (CNC). Users can develop customized program with programming environments, such as BCB, C#, Delphi, VB, VB.Net and VC. Application commands and the related chapters are listed in this chapter for quick reference.

1.1 Introduction of Software Structure

DMC_SNC is a dynamic-link library (DLL) aiming at computer numerical control (CNC) provided by Delta. This DLL is based on DMCNET high-speed motion control system. It is mainly used to decode the G code program and complete the complicated algorithm. See its structure below (Figure 1-1). There are several standard definitions for G code and this manual will illustrate ISO standard definition supported by Delta DMC_SNC.

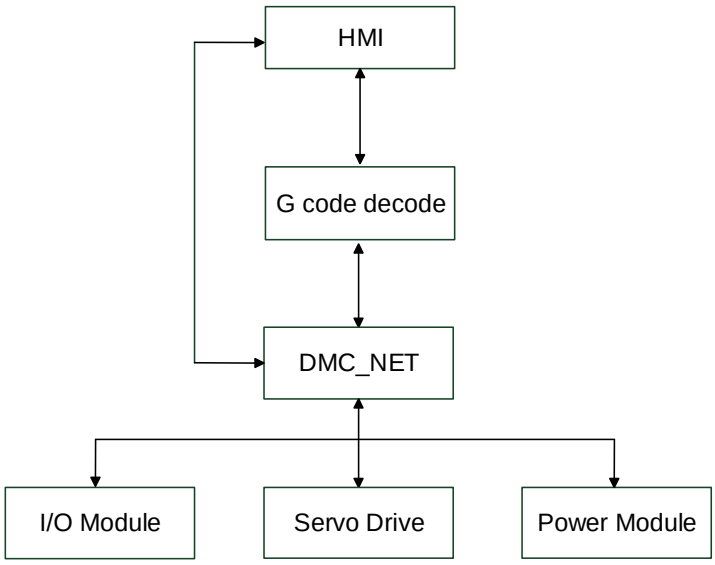


Figure 1-1 DMC_SNC Structure

Human Machine Interface (HMI)

It is the media for message exchange, communication and interaction between electromagnetic system and users. Users can easily complete the setting through the visual icons or buttons on this interface and then execute industrial automation control.

G Code Decoder

This function helps users deal with G code file. Then, it will utilize DMC_NET issues the motion command from motion card to servo drives. The auxiliary function includes Reverse, Look Ahead and etc.

DMC_NET

Issue the motion command to servo drive through DMC_NET and dynamic linking libraries (DLL), including single-axis motion control, two-/three-axis interpolation, speed control and torque control. It also supports remote control module.

I/O Module

It is a remote extension module for high-speed communication, which can process the interrupted command procedure.

Servo Drive

Delta servo drives and pulse control modules. With auto gain tuning, command smoothing and software analysis and monitoring, its control loop adopts digital signal processing (DSP) to satisfy the demand of motion control, such as high-speed moving and accurate positioning.

Power Module

Analog to Digital and Digital to Analog extension module

In the field of CNC, all products that being processed have its unique G code program. Through HMI compiled by integrator, import the G code that is going to be processed to the machine and starts processing (See figure 1-2). G code decoder and DMCNET included in dynamic-link library (DLL) from Delta's DMC_SNC (which shown in figure 1-1) start working. Then, simultaneously decode G code and issue the motion command of DMCNET until the processing is complete.

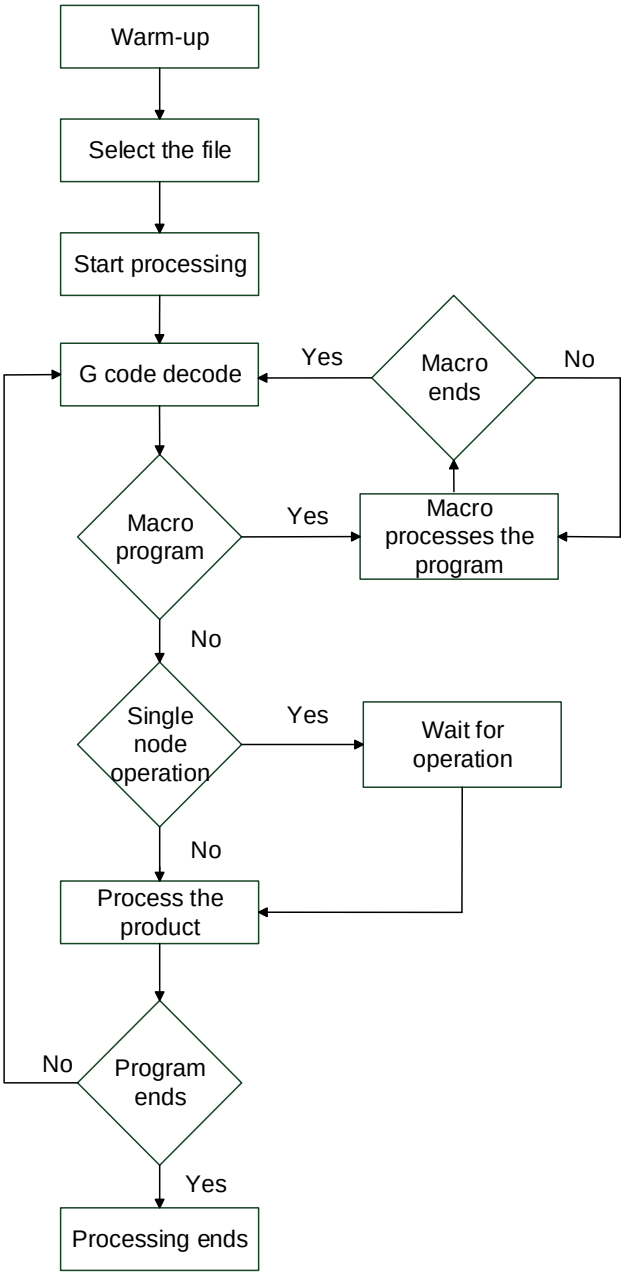


Figure 1-2 Processing Procedure

1.2 List of Application Command

Delta's Dynamic-link library provides the complete application command so as to satisfy the demand of each numerical control. The following table (table 1-1) lists each command and its description for quick reference.

Table 1-1

Name	Description	Chapter
Initialize Setting		
_SNC_group_init	Setup SNC group number	2.2
_SNC_group_close	Close SNC group function	2.3
_SNC_group_set_parameter	Setup SNC card type and card number	2.4
_SNC_initial	Initialize SNC function	2.5
_SNC_close	Close SNC function	2.6
Processing File Setting		
_SNC_open_file	Read processing file	3.2
_SNC_set_process_data	Setup processing data	3.3
Parameters Setting and Accessing		
_SNC_set_parameter	Setup SNC parameters	4.2
_SNC_get_parameter	Acquire the content of SNC parameters	4.3
_SNC_calc_gear	Calculate gear ratio after setting up Pulse_Per_Rev and Dist_Per_Rev.	4.4
_SNC_setup_hw	Setup SNC hardware information and: (1) DMC Gear (2) SNC_Feed_Rate_Rate (Feed rate resolution) (*Use it after the card is initialized)	4.5
_SNC_set_unit	Setup the machine unit and SNC processing unit	4.6
_SNC_set_path_macro	Setup macro file path	4.7
SNC Operation Command		
_SNC_start_process	Start to process	5.1
_SNC_start_estimate	Start program pre-scanning and estimate the processing time	5.2
_SNC_get_gcode_main_current_line	Acquire the number of line that G code's main program is currently being executed.	5.3
_SNC_get_gcode_current_line	Acquire the number of line that G code's main / sub-program is being currently executed	5.4
_SNC_get_gcode_current_command	Acquire the current processing command	5.5
_SNC_get_gcode_next_command	Acquire the next processing command	5.6
_SNC_get_gcode_current_code_feedrate	Acquire the currently executed feed rate set by G code	5.7
_SNC_get_speed	Convert the feed rate to pulse speed	5.8

Name	Description	Chapter
Macro		
_SNC_set_macro_mode	Setup macro mode	6.2
_SNC_set_callback	Setup Callback function	6.3
_SNC_get_event_handle	Acquire the Event that Thread is waiting for	6.4
_SNC_get_macro_value (ISO)	Acquire macro value	6.5
_SNC_macro_done	After Macro mode is finished, ask SNC system go on the rest of G code.	6.6
_SNC_set_callback_user_macro (ISO)	Setup Callback function (user defined)	6.7
_SNC_get_user_macro_value (ISO)	Acquire macro value (user defined)	6.8
_SNC_set_iso_cycle_mode (ISO)	Setup macro mode (drilling cycle)	6.9
_SNC_set_callback_iso_cycle (ISO)	Setup Callback function (drilling cycle)	6.10
_SNC_get_event_handle_iso_cycle (ISO)	Acquire the Event that Thread is waiting for (drilling cycle)	6.11
G Code Troubleshooting		
_SNC_get_error	Acquire SNC G code error (Index-specify the specific error number)	7.2
_SNC_get_error_seq	Acquire SNC G code error (Index-display the last error number)	7.3
_SNC_get_warning	Acquire SNC G code warning (Index-specify the specific error number)	7.4
_SNC_get_warning_seq	Acquire SNC G code warning (Index-display the last error number)	7.5
_SNC_dump_param	Write the non-zero parameters set by SNC into the target file	7.6

(This page is intentionally left blank.)

2 Initialized Setting

Before applying the related commands of SNC, users have to initialize SNC through the functions mentioned in this chapter so that each command can work normally.

2.1 List of application command

Table 2-1

Name	Description
_SNC_group_init	Setup SNC group number
_SNC_group_close	Close SNC group function
_SNC_group_set_parameter	Setup SNC card type and card number
_SNC_initial	Initialize SNC function
_SNC_close	Close SNC function

2.2 _SNC_group_init

- Format

  I16 __stdcall _SNC_group_init(U16 u16_GroupNum)

- Purpose

  Setup SNC group number.

- Parameter

Name	Data Type	Description
u16_GroupNum	U16	Group number

- Example

  I16 rt;

  rt = _SNC_group_init(2);

2.3 _SNC_group_close

· Format

I16 __stdcall _SNC_group_close(void);

· Purpose

Close SNC group function.

· Parameter

No related parameter

· Example

```
I16 rt;
rt = _SNC_group_close();
```

2.4 _SNC_group_set_parameter

· Format

I16 __stdcall _SNC_group_set_parameter(U16 u16_Group, U16 u16_CardType,
U16 u16_CardNo)

· Purpose

Setup SNC card type and card number.

· Parameter

Name	Data Type	Description
u16_Group	U16	Group number
u16_CardType	U16	Card type
u16_CardNo	U16	Card number

· Example

```
I16 rt;
U16 u16_Group    = 0;
U16 u16_CardType = 0;    //0: DMC_A01(B01)
U16 u16_CardNo   = 1;
rt = _SNC_group_set_parameter(u16_Group, u16_CardType, u16_CardNo);
```

2.5 _SNC_initial

- Format

  I16 __stdcall _SNC_initial(U16 u16_Group)

- Purpose

  Initialize SNC function.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number

- Example

  I16 rt;

  rt = _SNC_initial(0);

2.6 _SNC_close

- Format

  I16 __stdcall _SNC_close(U16 u16_Group)

- Purpose

  Close SNC function.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number

- Example

  I16 rt;

  rt = _SNC_close(0);

(This page is intentionally left blank.)

3 Processing File Setting

Set the G code’s file path that is going to be processed to SNC system. Then, users can process this file ant time when they needed.

3.1 List of application command

Table 3-1

Name	Description
_SNC_open_file	Read processing file
_SNC_set_process_data	Setup processing data

3.2 _SNC_open_file

- Format

  I16 __stdcall _SNC_open_file(U16 u16_Group, char *pFilename, I32 *pi32_Lines);

- Purpose

  Set the file path to G code decoder. G code decoder will read the file and calculate the line number.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
pFilename	char*	File path
pi32_Lines	I32*	Calculate the line number and send the result back to user

- Example

  char pFilename[80] = "D:\\GCode\Test.nc"

  I16 rt;

  I32 i32_Lines;

  rt = _SNC_open_file(0, pFilename, &i32_Lines);

3.3 _SNC_set_process_data

- Format

  I16 __stdcall _SNC_set_process_data(U16 u16_Group, char **pStr, I32 i32_Lines);

- Purpose

  Send the processing content to G code decoder. If the data setting is not complete, it will send back an error message.

- Parameter

Name	Data Type	Description
iGroup	U16	Group number
pStr	char**	Processing string (Two-dimensional array)
i32_Lines	i32	Total line number of processing content

- Example

```
I16 rt;
char sData[200][128] = {...};
//allocate memory space and copy the string content
rt = _SNC_set_process_data(0, sData, 200);
```

4 Parameter Setting and Access

Apart from the function of initialization, before using SNC processing, users have to set up parameters that are required during processing for each mechanism. This chapter lists each parameter's setting and related value for users' reference.

4.1 List of application command

Table 4-1

Name	Description
_SNC_set_parameter	Setup SNC parameters (Please refer to Chapter 8 for parameter description)
_SNC_get_parameter	Acquire the content of SNC parameters (Please refer to Chapter 8 for parameter description)
_SNC_calc_gear	Calculate gear ratio after setting up Pulse_Per_Rev and Dist_Per_Rev.
_SNC_setup_hw	Setup SNC hardware information and: (1) DMC Gear (2) SNC_Feed_Rate_Rate (Feed rate resolution) (*Use is after the card is initialized)
_SNC_set_unit	Setup the machine unit and SNC processing unit
_SNC_set_path_macro	Setup macro file path

4.2 _SNC_set_parameter

- Format

  I16 __stdcall _SNC_set_parameter(U16 u16_Group, I32 i32_Idx, F64 f64_Val)

- Purpose

  Setup SNC parameters. Please refer to Chapter 8 for parameter description.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
i32_Idx	I32	Parameter number
f64_Val	F64	Parameter setting value

- Example

```
I16 rt;
F64 f64_Val = 100;
rt = _SNC_set_parameter(0, 1900, f64_Val);
```

4.3 _SNC_get_parameter

- Format

  F64 __stdcall _SNC_get_parameter(U16 u16_Group, I32 i32_Idx)

- Purpose

  Acquire the content of SNC parameters. Please refer to Chapter 8 for parameter description.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
i32_Idx	I32	Parameter number

- Example

```
I16 rt;
F64 f64_Val = _SNC_get_parameter(0, 1900);
```

4.4 _SNC_calc_gear

- Format

  I16 __stdcall _SNC_calc_gear(U16 u16_Group)

- Purpose

  Calculate gear ratio after setting up Pulse_Per_Rev and Dist_Per_Rev and SNC system will write the value into SNC_AxisX_Gear ~ SNC_Axis_W_Gear.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number

- Example

```
I16 rt;
rt = _SNC_calc_gear(0);
```

4.5 _SNC_setup_hw

- Format

  I16 __stdcall _SNC_setup_hw(U16 u16_Group)

- Purpose

  Setup SNC hardware information, DMC Gear and SNC_Feed_Rate_Rate. Then, parameter SNC_HW_Setted will be set to 1. SNC_start_process no longer needs to setup hardware information. When setting up the hardware, SNC_Feed_Rate_Running can correctly calculate the feed rate. (*Use it after the card is initialized.)

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number

- Example

```
I16 rt;
rt = _SNC_setup_hw(0);
```

4.6 _SNC_set_unit

- Format

  I16 __stdcall _SNC_set_unit(U16 u16_Group, I8 i8_Type, I8 i8_UsingType)

- Purpose

  Setup the unit for mechanism and SNC system,

  Value 20: inch

  Value 21: mm.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
i8_Type	I8	Unit for mechanism
i8_UsingType	I8	Using for SNC system

- Example

```
I16 rt;
rt = _SNC_set_unit(0, 21, 21);
```

4.7 _SNC_set_path_macro

- Format

  I16 __stdcall _SNC_set_path_macro(U16 u16_Group, char *pDirname)

- Purpose

  Setup the path of macro file

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
pDirname	char*	File path

- Example

```
I16 rt;
rt = _SNC_set_path_macro(0, "D:\SNC\Macro");
```

(This page is intentionally left blank.)

5 SNC Operation Command

The detailed description and example of each main function that will be used during SNC numeric control are listed in this chapter.

5.1 List of application command

Table 5-1

Name	Description
_SNC_start_process	Start to process
_SNC_start_estimate	Start program pre-scanning the and estimate the processing time
_SNC_get_gcode_main_current_line	Acquire the number of line that G code's main program is currently being executed.
_SNC_get_gcode_current_line	Acquire the number of line that G code's main / sub-program is currently being executed
_SNC_get_gcode_current_command	Acquire the current processing command
_SNC_get_gcode_next_command	Acquire the next processing command
_SNC_get_gcode_current_code_feedrate	Acquire the currently executed feed rate set by Gcode
_SNC_get_speed	Convert the feed rate to pulse speed

5.2 _SNC_start_process

- Format

  I16 __stdcall _SNC_start_process(U16 u16_Group)

- Purpose

  Start G code processing

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number

- Example

```
I16 rt;
rt = _SNC_start_process(0);
```

5.3 _SNC_start_estimate

- Format

  I16 __stdcall _SNC_start_estimate (U16 u16_Group)

- Purpose

  Start to pre-scan G code and estimate the processing time.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number

- Example

```
I16 rt;
rt = _SNC_start_estimate (0);
```

5.4 _SNC_get_gcode_main_current_line

· Format

I16 __stdcall _SNC_get_gcode_main_current_line(U16 u16_Group, I32 *pi32_CurLine)

· Purpose

Acquire the line number that G code's main program is currently being executed.

· Parameter

Name	Data Type	Description
u16_Group	U16	Group number
pi32_CurLine	I32*	G code's line number that is currently executed

· Example

```
I16 rt;
I32 i32_CurLine;
rt = _SNC_get_gcode_main_current_line(0, &i32_CurLine);
```

5.5 _SNC_get_gcode _current_line

· Format

I16 __stdcall _SNC_get_gcode_current_line(

U16 u16_Group, U8 *pu8_CodeIndex, I32 *pi32_CurLine)

· Purpose

Acquire the line number that G code's main / sub-program is currently being executed.

· Parameter

Name	Data Type	Description
u16_Group	U16	Group number
pu8_CodeIndex	U8*	0: acquire from main program or 1: sub-program
pi32_CurLine	I32*	G code's line number that is currently executed

· Example

```
I16 rt;
U8 u8_CodeIndex;
I32 i32_CurLine;
rt = _SNC_get_gcode_current_line(0, &u8_CodeIndex, &i32_CurLine);
```

5.6 _SNC_get_gcode _current_command

- Format

  I16 __stdcall _SNC_get_gcode_current_command(U16 u16_Group, char *pStrCurCmd)

- Purpose

  Acquire the current processing command.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
pStrCurCmd	char*	Current processing command

- Example

```
char pStrCurCmd[128];
I16 rt;
rt = _SNC_get_gcode_current_command(0, pStrCurCmd);
```

5.7 _SNC_get_gcode _next_command

- Format

  I16 __stdcall _SNC_get_gcode_next_command(U16 u16_Group, char *pStrNextCmd)

- Purpose

  Acquire next processing command.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
pStrNextCmd	char*	Next processing command

- Example

```
char pStrNextCmd[128];
I16 rt;
rt = _SNC_get_gcode_next_command(0, pStrNextCmd);
```

5.8 _SNC_get_gcode _current_code_feedrate

· Format

I32 __stdcall _SNC_get_gcode_current_code_feedrate(U16 u16_Group, F64 *pf64_FeedRate)

· Purpose

Acquire the currently executed feed rate set by G code.

· Parameter

Name	Data Type	Description
u16_Group	U16	Group number
pf64_FeedRate	F64*	Feed rate set by GCode

· Example

```
I16 rt;
F64 f64_FeedRate;
rt = _SNC_get_gcode_current_code_feedrate(0,    &f64_FeedRate);
```

5.9 _SNC_get_speed

· Format

I32 __stdcall _SNC_get_speed(U16 u16_Group, char *pName, F64 f64_FeedRate)

· Purpose

Convert the feed rate to pulse speed.

· Parameter

Name	Data Type	Description
u16_Group	U16	Group number
pName	char*	String name of each axis: X, Y, Z, A, B, C, U, V, W
f64_FeedRate	F64	Feed rate

· Example

```
char pName='X';
I32     i32_Spd;
F64    f64_FeedRate = 600;
i32_Spd = _SNC_get_speed(&pName, f64_FeedRate);
```

6 Macro

Users can apply functions mentioned in this chapter for processing commands if encountering macro function when decoding G code program.

6.1 List of application command

Table 6-1

Name	Description
_SNC_set_macro_mode	Setup macro mode
_SNC_set_callback	Setup Callback function
_SNC_get_event_handle	Acquire the Event that Thread is waiting for
_SNC_get_macro_value (ISO)	Acquire macro value
_SNC_macro_done	After macro mode is finished, ask SNC system go on the rest of G codes.
_SNC_set_callback_user_macro (ISO)	Setup Callback function for user defined G code macro
_SNC_get_user_macro_value (ISO)	Acquire macro value from user defined G code
_SNC_set_iso_cycle_mode (ISO)	Setup macro mode when using drilling cycle mode
_SNC_set_callback_iso_cycle (ISO)	Setup Callback function when using drilling cycle mode
_SNC_get_event_handle_iso_cycle (ISO)	Acquire the Event that Thread is waiting for when using drilling cycle mode

6.2 _SNC_set_macro_mode

- Format

  I16 __stdcall _SNC_set_macro_mode(U16 u16_Group, U8 u8_Mode)

- Purpose

  Setup macro mode.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
u8_Mode	U8	1: Callback 2: SetEvent (Handle) 3: Scan

- Example

```
U8 u8_Mode = 3;
I16 rt;
rt = _SNC_set_macro_mode(0, u8_Mode);
```

6.3 _SNC_set_callback

- Format

  I16 __stdcall _SNC_set_callback(U16 u16_Group,

  void (__stdcall *callback)(I32 i32_VarG, I32 i32_VarM, I32 i32_VarT, I32 i32_VarS, I32i32_VarH)

)

- Purpose

  Setup SNC Callback function pointer by this function when the parameter _SNC_set_macro_mode is set as Callback mode.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
void (__stdcall.	Function address	Send Callback function address to G code decoder

- Example

```
I16 rt;
    void __stdcall callback(I32 i32_VarG, I32 i32_VarM, I32 i32_VarT, I32 i32_VarS, I32 i32_VarH)
    {
        ...
    }

    rt = _SNC_set_callback(0, callback );
```

6.4 _SNC_set_event_handle

- Format

  I16 __stdcall _SNC_get_event_handle(U16 u16_Group, HANDLE *h)

- Purpose

  Acquire the Event that Thread is waiting for.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
h	HANDLE*	Event's handle pointer

- Example

```
I16 rt;
Handle hEvent;
rt = _SNC_get_event_handle(0, &hEvent);
```

6.5 _SNC_get_macro_value (ISO)

· Format

I16 __stdcall _SNC_get_macro_value(U16 u16_Group,
I32 *pi32_VarG, I32 *pi32_VarM, I32 *pi32_VarT,
I32 *pi32_VarS, I32 *pi32_VarH, U16 *pu16_DoMacro)

· Purpose

Acquire Macro value when processed G code file is based on ISO.

· Parameter

Name	Data Type	Description
u16_Group	U16	Group number
pi32_VarG	I32*	pi32_VarG is one of the parameter of _SNC_get_macro_value (ISO). When accessing G43 (tool length compensation), HN is also needed (H = tool length; N = tool number).
pi32_VarM	I32*	M code's function number
pi32_VarT	I32*	Tval: Tool number when change the tool
pi32_VarS	I32*	Sval: Spindle speed
pi32_VarH	I32*	Hval: H code value in G43
pu16_DoMacro	U16*	Determine if it is going to execute Macro

· Example

```
I16 rt;
U16 u16_DoMacro;
I32   i32_VarG, i32_VarM, i32_VarT, i32_VarS, i32_VarH;
   rt = _SNC_get_macro_value(0, &i32_VarG, &i32_VarM, &i32_VarT, &i32_VarS, &i32_VarH,
                                                    &u16_DoMacro);
      if(1 ==   u16_DoMacro)
      {
            …
            _SNC_macro_done(0);
      }
```

6.6 _SNC_ macro_done

- Format

  I16 __stdcall _SNC_macro_done(U16 u16_Group)

- Purpose

  After Macro mode is finished, as SNC system go on the rest of G codes.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number

- Example

```
I16 rt;
rt = _SNC_macro_done(0);
```

6.7 _SNC_ set_callback_user_macro (ISO)

- Format

```
I16 __stdcall _SNC_set_callback_user_macro(U16 u16_Group,
        void (_stdcall *callback)(F64 f64_VarG, F64 f64_VarA, F64 f64_VarB, F64
f64_VarC,
                                  F64 f64_VarD, F64 f64_VarE, F64 f64_VarF,
                                  F64 f64_VarH, F64 f64_VarK, F64 f64_VarL,
                                  F64 f64_VarP, F64 f64_VarQ, F64 f64_VarS,
                                  F64 f64_VarZ) )
```

- Purpose

  Setup Callback function for user defined G code macro.

- Parameter

  All parameters on user defined macro

- Example

```
void __stdcall Callback_User_Macro(F64 f64_VarG, F64 f64_VarA, F64 f64_VarB, F64
f64_VarC,
                                   F64 f64_VarD, F64 f64_VarE, F64 f64_VarF,
                                   F64 f64_VarH, F64 f64_VarK, F64 f64_VarL,
                                   F64 f64_VarP, F64 f64_VarQ, F64 f64_VarS,
                                   F64 f64_VarZ)
{
    …
}

I16 rt = _SNC_set_callback_user_macro(0, Callback_User_Macro);
```

6.8 _SNC_ get_user_macro_value (ISO)

- Format

```
I16 __stdcall _SNC_get_user_macro_value(U16 u16_Group, F64 *pf64_VarG,
                                        F64 *pf64_VarA, F64 *pf64_VarB, F64 *pf64_VarC,
                                        F64 *pf64_VarD, F64 *pf64_VarE, F64 *pf64_VarF,
                                        F64 *pf64_VarH, F64 *pf64_VarK, F64 *pf64_VarL,
                                        F64 *pf64_VarP, F64 *pf64_VarQ, F64 *pf64_VarS,
                                        F64 *pf64_VarZ, U16 *pu16_DoMacro)
```

- Purpose

  Acquire macro value from user defined G code.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
...	F64*	All parameters on user defined macro
pu16_DoMacro	U16*	Determine if it is going to execute Macro (1: Yes; Others: No)

- Example

```
I16 rt;
U16 u16_DoMacro;
    F64 f64_VarG, f64_VarA, f64_VarB, f64_VarC,
            f64_VarD, f64_VarE, f64_VarF,
            f64_VarH, f64_VarK, f64_VarL,
            f64_VarP, f64_VarQ, f64_VarS,
            f64_VarZ;
    rt = _SNC_get_user_macro_value(0, &f64_VarG, &f64_VarA, &f64_VarB, &f64_VarC,
                    &f64_VarD, &f64_VarE, &f64_VarF,
                    &f64_VarH, &f64_VarK, &f64_VarL,
                    &f64_VarP, &f64_VarQ, &f64_VarS,
                    &f64_VarZ, &u16_DoMacro);

    if(2 ==   u16_DoMacro)
    {
        …
        _SNC_macro_done(0);
    }
```

6.9 _SNC_ get_iso_cycle_mode (ISO)

· Format

I16 __stdcall _SNC_set_iso_cycle_mode(U16 u16_Group, U8 u8_IsoCycleMode)

· Purpose

Setup macro mode when using drilling cycle mode.

· Parameter

Name	Data Type	Description
u16_Group	U16	Group number
u8_Mode	U8	1: Callback 2: SetEvent(Handle) 3: Scan

· Example

```
U8 u8_Mode = 3;
I16 rt;
rt = _SNC_set_macro_mode(0, u8_Mode);
```

6.10 _SNC_ set_callback_iso_cycle (ISO)

· Format

I16 __stdcall _SNC_set_callback_iso_cycle(U16 u16_Group, void (_stdcall *callback)(void))

· Purpose

Setup Callback function when using drilling cycle mode.

· Parameter

Name	Data Type	Description
u16_Group	U16	Group number
void (__stdcall.	Function Address	Send the Callback function address to G code decoder

· Example

```
//Please refer to Chapter 10 Parameters Descriptions for the following parameters.
void __stdcall Callback_IsoCycle(void)
{
        U16 u16_MotionType = (U16)_SNC_get_parameter(0, SNC_ISO_MotionType);
        U16 u16_Do = (U16)_SNC_get_parameter(0, SNC_ISO_Cycle_Do);
        F64 f64_Ori_Z = _SNC_get_parameter(0, SNC_ISO_Cycle_Ori_Z);
        I32 i32_Ori_Z_Pulse = (I32)_SNC_get_parameter(0,
SNC_ISO_Cycle_Ori_Z_Pulse);
        F64 f64_Target_Z = _SNC_get_parameter(0, SNC_ISO_Cycle_Target_Z);
        I32 i32_Target_Z_Pulse = (I32)_SNC_get_parameter(0,
SNC_ISO_Cycle_Target_Z_Pulse);
        F64 f64_Level_R = _SNC_get_parameter(0, SNC_ISO_Cycle_Level_R);
        I32 i32_Level_R_Pulse = (I32)_SNC_get_parameter(0,
SNC_ISO_Cycle_Level_R_Pulse);
        F64 f64_Wait_Time = _SNC_get_parameter(0, SNC_ISO_Cycle_Wait);    //Second
        F64 f64_Offset_Q = _SNC_get_parameter(0, SNC_ISO_Cycle_Offset_Q);
        I32 i32_Offset_Q = (I32)_SNC_get_parameter(0,
SNC_ISO_Cycle_Offset_Q_Pulse);
        F64 f64_Offset_d = _SNC_get_parameter(0, SNC_ISO_Cycle_Offset_d);
        F64 f64_FeedRate = _SNC_get_parameter(0, SNC_ISO_Cycle_FeedRate);
        I32 i32_FeedRate_Pulse = (I32)_SNC_get_parameter(0,
SNC_ISO_Cycle_FeedRate_Pulse);
        U8    u8_Return_Level = (U8)_SNC_get_parameter(0, SNC_ISO_Return);

        F64 f64_S = _SNC_get_parameter(0, SNC_S_Number);
```

```
        _SNC_set_parameter(0, SNC_ISO_Cycle_Do, 0);
}

I16 rt = _SNC_set_callback_iso_cycle(0, Callback_IsoCycle);
```

6.11 _SNC_ get_event_handle_iso_cycle (ISO)

- Format

  I16 __stdcall _SNC_get_event_handle_iso_cycle(U16 u16_Group, HANDLE *h)

- Purpose

  Acquire the Event that Thread is waiting for when using drilling cycle mode.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
h	HANDLE*	Event's handle pointer

- Example

```
I16 rt;
Handle hEvent;
rt = _SNC_get_event_handle_iso_cycle(0, &hEvent);
```

7 G Code Troubleshooting

During the process of decoding G code, SNC will automatically record the error and warning from G code so that users can look up the error from the provided related functions for troubleshooting.

7.1 List of application command

Table 7-1

Name	Description
_SNC_get_error	Acquire SNC G code error code (Index-specify the specific error number)
_SNC_get_error_seq	Acquire SNC G code error code (Index-display the last error number)
_SNC_get_warning	Acquire SNC G code warning code (Index-specify the specific error number))
_SNC_get_warning_seq	Acquire SNC G code warning code (Index-display the last error number)
_SNC_dump_param	Write non-zero parameters set by SNC into the target file

7.2 _SNC_get_error

· Format

I16 __stdcall _SNC_get_error(U16 u16_Group, U16 u16_Idx,
U16 *pu16_Type, U16 *pu16_ErrNo, I32 *pi32_Line)

· Purpose

Acquire SNC G code error code (Index-specify the specific error number).

· Parameter

Name	Data Type	Description
u16_Group	U16	Group number
u16_Idx	U16	Index of error code (0 ~ SNC_Err_Count) SNC_Err_Count is the error count of G code.
pu16_Type	U16*	Error type
pu16_ErrNo	U16*	Error number
pi32_Line	I32*	Error line

· Example

```
I16     rt;
U16 u16_Idx = 0;
U16 u16_Type, u16_ErrNo;
I32    i32_Line;
rt = _SNC_get_error(0, u16_Idx, &u16_Type, &u16_ErrNo, &i32_Line);
```

7.3 _SNC_get_error_seq

- Format

  I16 __stdcall _SNC_get_error_seq(U16 u16_Group,

  U16 *pu16_Type, U16 *pu16_ErrNo, I32 *pi32_Line)

- Purpose

  Acquire SNC GCode error code (Index-display the last error number). When this function is called, parameter SNC_Err_Idx will automatically accumulate the error number until it is equal to parameter SNC_Err_Count. If the returned value is -1, it means no new error occurs.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
pu16_Type	U16*	Error type
pu16_ErrNo	U16*	Error number
pi32_Line	I32*	Error line

- Example

```
I16     rt;
U16 u16_Type, u16_ErrNo;
I32     i32_Line;
rt = _SNC_get_error_seq(0, &u16_Type, &u16_ErrNo, &i32_Line);
if(rt != -1)
{
...
}
```

7.4 _SNC_get_warning

- Format

  I16 __stdcall _SNC_get_warning(U16 u16_Group, U16 u16_Idx,

  U8 *pu8_CodeIndex, U16 *pu16_WarningNo, I32 *pi32_Line)

- Purpose

  Acquire SNC G code warning code (Index-specify the specific error number).

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
u16_Idx	U16	Index for warning code (0 ~ SNC_Warn_Count) SNC_Warn_Count is the warning count of G code
pu8_CodeIndex	U8*	G code layer number
pu16_WarningNo	U16*	Warning number
pi32_Line	I32*	Warning line number

- Example

```
I16     rt;
U8 u8_CodeIndex;
U16 u16_Idx = 0;
U16 u16_WarningNo;
I32    i32_Line;
rt = _SNC_get_warning(0, u16_Idx, &u8_CodeIndex, &u16_WarningNo, &i32_Line);
```

7.5 _SNC_get_warning_seq

- Format

  I16 __stdcall _SNC_get_warning_seq(U16 u16_Group,
  U8 *pu8_CodeIndex, U16 *pu16_WarningNo, I32 *pi32_Line)

- Purpose

  Acquire SNC G code warning code (Index-display the last error number). When this function is called, parameter SNC_Warn_Idx will automatically accumulate the error number until it is equal to parameter SNC_Warn_Count. If the returned value is -1, it means no new warning occurs.

- Parameter

Name	Data Type	Description
u16_Group	U16	Group number
pu8_CodeIndex	U8*	G code layer number
pu16_WarningNo	U16*	Warning number
pi32_Line	I32*	Warning line number

- Example

```
I16     rt;
U8      u8_CodeIndex;
U16 u16_WarningNo;
I32     i32_Line;
rt = _SNC_get_warning_seq(0, &u8_CodeIndex, &u16_WarningNo, &i32_Line);
if(rt != -1)
{
...
}
```

7.6 _SNC_dump_param

· Format

I16 __stdcall _SNC_dump_param(U16 u16_Group, char *path)

· Purpose

Write non-zero parameters set by SNC into the target file.

· Parameter

Name	Data Type	Description
u16_Group	U16	Group number
path	char*	File location

· Example

```
I16     rt;

rt = _SNC_dump_param(0, “C:\\SNC_Dump_Param.txt”);
```

(This page is intentionally left blank.)

Parameters Descriptions

8

After SNC is initialized, users have to setup parameters according to each mechanism. This is for accurately and quickly decoding G code and executing commands. This chapter lists functions and definitions of each parameter.

8.1 List of parameters and descriptions

*Abbreviation O stands for operation; R/W stands for read and write.

Table 8-1

No.	Name	O	Description
1 ~ 500	None	R/W	Reserved
Jump function setting (Please refer to Chapter 12 for further information)			
700	SNC_Card_Type	R/W	Card type 0: DMC_B01 1: DMC_F02
701	SNC_GCode_Type	R/W	G code type 0: ISO 1: EIA
702	SNC_Machine_Type	R/W	Sub function type 0: Fanuc likes 1: Syntec likes
703	SNC_Use_MPG_Control	R/W	Percentage of MPG control speed
704	SNC_User_Scan_MCode	R/W	When pre-scanning, enable or disable the prompt box to ask if users are going to execute M code.
705	SNC_Keep_Sharp_Variables	R/W	Whether to save the variables content of sub function 0: Variables content will be cleared when SNC is executed. 1: Save
706	SNC_Keep_Work_Plane	R/W	Whether to keep SNC_Work_Plane work plane setting 0: Machine will return to G54 when being executed (default) 1: Save
721	SNC_Jump_Cond_Alpha	R/W	G code / M code jump function - start alpha
722	SNC_Jump_Cond_Number	R/W	G code / M code jump function - function number
723	SNC_Jump_Trigger	R/W	Record the triggered number of jump function

Special setting			
800	SNC_Draw_Color_Type	R/W	The line color of G code path simulation
802	SNC_User_Tool_Length_Compensation_Type	R/W	ISO. Select G43 and G44 mode: 0: Acquire tool compensation from parameters (2001~2100) 1: Acquire tool compensation from macro.
803	SNC_User_Tool_Cutter_Compensation_Type	R/W	ISO. Select G41 and G42 mode: 0: Acquire tool compensation from parameters (2201~2300) 1: Acquire tool compensation from macro.
805	SNC_Check_Tool_No	R/W	Confirm the tool that you wish to use is listed in the tool list. 0: Disable (Default) 1: Enable
807	SNC_Scan	R/W	Enable G code pre-scanning function 0: Disable 1: Enable (Default)
810	SNC_Alwasy_Check_Axis_Alarm	R/W	Whether to check if software limit / hardware limit / errors related to servo occurs when SNC is processing. If the setting is enabled, the value of SNC_Is_Alarm will be changed.
811	SNC_Reverse	R/W	The reverse working direction function 0: Disable (Default) 1: Enable
813	SNC_Work_Type	R	SNC working status: 0: process 1: forward direction 2: jump function
816	SNC_Unit_Disp	R	Unit that being used when executing SNC
817	SNC_Unit_Multiply	R	Unit multiply
818	SNC_Base_Unit	R/W	Machine unit (ball screw)
819	SNC_Unit	R/W	Machine unit
Debug and Test			
839	SNC_Err_Idx	R/W	The index of _SNC_get_error_seq
840	SNC_Err_Count	R	G code error count
841	SNC_Warn_Idx	R/W	The index of _SNC_get_warning_seq
842	SNC_Warn_Count	R	Total number of G code warning

Motion Control			
858	SNC_Ignore_NC_Kerf_Setting	R/W	Ignore the tool radius setting in NC code 0: Not to ignore (default) 1: Ignore
859	SNC_Ignore_NC_FeedRate	R/W	Ignore the feed rate setting in NC code when applying _SNC_start_process function.
860	SNC_Tolerance	R/W	Sets the tolerance of continuous cutting. Value is set as distance, e.g. 0.01 mm. Please refer to Chapter 11 for further description of parameters.
867	SNC_Circle_Tolerance	R/W	Sets the tolerance of arc calculation. Value is set as distance, e.g. 0.01 mm. Please refer to Chapter 11 for further description of parameters.
870	SNC_G00_Use_Non_Liner	R/W	System would not use the interpolation when G00 is decoded.
871 ~879	SNC_Feed_Rate_G00_AxisX ~ SNC_Feed_Rate_G00_AxisW	R/W	G00 feed rate of X~Z--axis, A~C-axis and U~W-axis
886	SNC_Kerf_Permit_Angle	R/W	Cutting kerf permission angle. When encountering lead angle, user can determine whether to use arc or linear line. Please refer to Chapter 11 for further description of parameters.
887	SNC_Corner_Control	R/W	Enable the function of corner control. Please enable the function of look ahead.
888	SNC_Corner_Angle	R/W	Corner angle
889	SNC_Corner_Speed	R/W	Corner speed
890	SNC_Use_Look_Ahead	R/W	Look ahead
891	SNC_Fix_Slope	R/W	For acceleration and deceleration: 0: time 1: acceleration
892	SNC_Curve	R/W	Acceleration curve, S-curve (2), T-curve (other)
893	SNC_Scale_Rate	R/W	ISO. The precision of the scaling function (IJK) 0: 0.001 (Default) 1: 0.00001
894	SNC_Short_Line_Warning_Or_Error	R/W	When the point to point distance being pre-scanning is too short: 0: warning 1: error

895	SNC_Look_Ahead_Speed_Up	R/W	Boost the speed of look ahead 0: Disable (Default) 1: Enable
900	SNC_Tdec_Sd_Stop	R/W	Sd_Stop; Refer to SNC_Fix_Slope: 0: time 1: acceleration
Procedure Control			
901	SNC_Is_Scanning	R	The program is being pre-scanning
902	SNC_Is_Processing	R	The status of system processing 1: processing 0: not in processing procedure
903	SNC_Is_Giveup	R/W	Set the value to 1. Give up processing and emergency stop.
904	SNC_Is_Stop	R/W	Set the value to 1. G code will stop after the incomplete command is finished.
905	SNC_Is_FeedHold	R/W	Set the value to 1 and the setting of feed rate percentage will be set to 0 (Procedure pauses)
906	SNC_Is_Step	R/W	Set the value to 1. It enters single-node operation mode.
Error			
907	SNC_ErrNo	R/W	See if any error occurs
908	SNC_Is_Alarm	R/W	See if there is any servo alarm, limit or software limit signal: 1: Yes 0: No
909	SNC_Code_ErrNo	R/W	Error code for G code decode
910	SNC_On_Soft_Limit	R	Software limit is triggered
911	SNC_API_ErrNo	R/W	Returned value of DMC Motion API
912	SNC_Setting_ErrNo	R/W	Parameter setting error
913	SNC_Device_ErrNo	R/W	Device error. (Not servo drive or pulse module (04PI))
914	SNC_Error_CardNo	R/W	Wrong number of the device card
915	SNC_Error_NodeID	R/W	Wrong station of the device
916	SNC_Error_SlotID	R/W	Wrong slot number of the device

917	SNC_System_Error	R/W	Function error of tool radius compensation
918	SNC_Tool_ErrNo	R	The system is failed to use the cutting tool when SNC_Check_Tool_No is enabled (the cutting tool is not in the list).
920	SNC_Code_Error_Line	R	Users can check error occurs in which line on SNC system (know the error type from SNC_ErrNo)
Line Number of Processing File			
941	SNC_Code_Lines	R	Total line number of G code processing files
942	SNC_Code_Lines_Macro1	R	Total line number of G code processing files (macro 1)
943	SNC_Code_Lines_Macro2	R	Total line number of G code processing files (Marco 2)
944	SNC_Code_Lines_Macro3	R	Total line number of G code processing files (macro 3)
945	SNC_Code_Lines_Macro4	R	Total line number of G code processing files (macro 4)
949	SNC_Macro_Code_Idx	R	Layer number that currently being executed by G code
Time			
961	SNC_Time_Estimate	R	Estimated time of program completion
962	SNC_Time_Processing	R	Program processing time
963	SNC_Time_Knife	R/W	Tool using time
964	SNC_Time_Remain	R	Time remaining to processing completion
966	SNC_Time_Dwell_Preset	R/W	Default processing time will be calculated by second when G04 only has number code.
969	SNC_Finish_Count	R/W	Number of complete processing
970	SNC_Process_Progress	R	The percentage of process completion
971 ~ 979	SNC_AxisX_LimitSpeed ~ SNC_AxisW_LimitSpeed	R/W	Speed limit of X~Z--axis, A~C-axis and U~W-axis (distance/minute, same as feed rate)
980	SNC_Use_LimitSpeed	R/W	Speed limit function 0: Disable (Default) 1: Enable

Max. Deceleration (Uses in Look Ahead and is for G01, G02 and G03)			
989	SNC_PermitMaxDec	R/W	Maximum deceleration: (distance / s²), Ex. mm / s²
Initial Speed			
990	SNC_Str_Vel	R/W	Initial speed. Its setting method is identical to feed rate setting (distance / per minute)
Max. Accelration / Deceleration (Uses in Look Ahead and is for G01, G02 and G03)			
991 ~ 999	SNC_AxisX_PermitMaxAcc ~ SNC_AxisW_PermitMaxAcc	R/W	The permissible acceleration / deceleration on X~Z--axis, A~C-axis and U~W-axis. Unit is: (distance / s²), Ex. mm / s²
Hardware Information			
1000	SNC_Axes	R/W	Axis amount applied by this system
1001 ~1009	SNC_AxisX_Axis ~ SNC_AxisW_Axis	R/W	Axis number of X~Z--axis, A~C-axis and U~W-axis
1011	SNC_Card_No	R/W	Card number
1021 ~1029	SNC_AxisX_Node ~ SNC_AxisW_Node	R/W	Station of X~Z--axis, A~C-axis and U~W-axis
1031 ~1039	SNC_AxisX_Slot ~ SNC_AxisW_Slot	R/W	Slot number of X~Z--axis, A~C-axis and U~W-axis
Status			
1051 ~1059	SNC_AxisX_Command ~ SNC_AxisW_Command	R	Command of X~Z--axis, A~C-axis and U~W-axis
1061 ~1069	SNC_AxisX_Feedback ~ SNC_AxisW_Feedback	R	Position of X~Z--axis, A~C-axis and U~W-axis
1070	SNC_Axes_Spd	R	Vector speed
1071 ~1079	SNC_AxisX_Speed ~ SNC_AxisW_Speed	R	Speed of X~Z--axis, A~C-axis and U~W-axis
1081 ~1089	SNC_AxisX_Done ~ SNC_AxisW_Done	R	Status of X~Z--axis, A~C-axis and U~W-axis is motion done
1091 ~1099	SNC_AxisX_Status ~ SNC_AxisW_Status	R	Motion status of X~Z--axis, A~C-axis and U~W-axis
1101 ~1109	SNC_AxisX_On_Soft_Limit ~ SNC_AxisW_On_Soft_Limit	R	Signal of software limit on X~Z--axis, A~C-axis and U~W-axis: 1: Triggered in forward direction 2: Triggered in reverse direction

			3: Triggered in both direction
E-gear Ratio			
1111	SNC_AxisX_Pulse_Per_Rev	R/W	Pulse number when the motor runs a cycle on X-axis
1112	SNC_AxisY_Pulse_Per_Rev	R/W	Pulse number when the motor runs a cycle on Y-axis
1121 ~1129	SNC_AxisX_Dist_Per_Rev ~ SNC_AxisW_Dist_Per_Rev	R/W	The distance when ball screw runs a cycle on X~Z--axis, A~C-axis and U~W-axis axis
1131 ~1139	SNC_AxisX_Gear ~ SNC_AxisW_Gear	R/W	E-gear ratio of X~Z--axis, A~C-axis and U~W-axis (Pulse / Pitch), which can be set directly or via API-6.3
Axial Moving Direction			
1141 ~1149	SNC_AxisX_Dir ~ SNC_AxisW_Dir	R/W	Moving direction of X~Z--axis, A~C-axis and U~W-axis: 1: same as the program -1: opposite direction
Software Limit			
1150	SNC_Use_Soft_Limit	R/W	Software limit function 0: Disable (Default) 1: Enable
1151 ~1159	SNC_AxisX_Soft_Limit_Pos ~ SNC_AxisW_Soft_Limit_Pos	R/W	Forward software limit position of X~Z--axis, A~C-axis and U~W-axis
1161 ~1169	SNC_AxisX_Soft_Limit_Neg ~ SNC_AxisW_Soft_Limit_Neg	R/W	Reverse software limit position of X~ Z-axis, A~C-axis and U~W-axis
Speed Control			
1170	SNC_Feed_Rate_Percent	R/W	Feed rate setting; Set the value to 0 then the motion stops.
1171	SNC_Feed_Rate_G00	R/W	G00 Feed rate speed
1172	SNC_Tacc_G00	R/W	G00, refer to SNC_Fix_Slope; 0: time 1: acceleration
1173	SNC_Tdec_G00	R/W	G00, refer to SNC_Fix_Slope; 0: time 1: acceleration
1174	SNC_Feed_Rate_G01	R/W	Speed limit of G01
1175	SNC_Tacc_G01	R/W	G01, refer to SNC_Fix_Slope; 0: time 1: acceleration

1176	SNC_Tdec_G01	R/W	G01, refer to SNC_Fix_Slope; 0: time 1: acceleration
1177	SNC_Feed_Rate_Circle	R/W	Speed limit of G02, G03
1178	SNC_Tacc_Circle	R/W	G02 and G03; refer to SNC_Fix_Slope; 0: time 1: acceleration
1179	SNC_Tdec_Circle	R/W	G02 and G03; refer to SNC_Fix_Slope; 0: time 1: acceleration
1180	SNC_Feed_Rate_Reverse	R/W	Feed rate speed of reverse direction
1181	SNC_Tacc_Reverse	R/W	Moves in reverse direction. Refer to SNC_Fix_Slope; 0: time 1: acceleration
1182	SNC_Tdec_Reverse	R/W	Moves in reverse direction. Refer to SNC_Fix_Slope; 0: time 1: acceleration
1183	SNC_Feed_Rate_G01_Default	R/W	Default feed rate speed of G01; The system will use SNC_Feed_Rate_G01 when there is no default setting
1184	SNC_Feed_Rate_Circle_Default	R/W	Default feed rate speed of G02 and G03; Use SNC_Feed_Rate_Circle when there is no default setting
1199	SNC_Feed_Rate_Rate	R/W	Feed rate resolution. The setting value is 0.01: SNC_Feed_Rate_Percent is displayed percentage. The setting value is 0.001: SNC_Feed_Rate_Percent is displayed permillage. (*This parameter is used by _SNC_setup_hw)
Coordinates			
1301 ~ 1309	SNC_AxisX_G59_1 ~ SNC_AxisW_G59_1	R/W	G59.1 of X~Z--axis, A~C-axis and U~W-axis
1311 ~ 1319	SNC_AxisX_G59_2 ~ SNC_AxisW_G59_2	R/W	G59.2 of X~Z--axis, A~C-axis and U~W-axis
1321 ~ 1329	SNC_AxisX_G59_3 ~ SNC_AxisW_G59_3	R/W	G59.3 of X~Z--axis, A~C-axis and U~W-axis
1331 ~ 1339	SNC_AxisX_G59_4	R/W	G59.4 of X~Z--axis, A~C-axis and U~W-axis

1341 ~ 1349	SNC_AxisX_G59_5 ~ SNC_AxisW_G59_5	R/W	G59.5 of X~Z--axis, A~C-axis and U~W-axis
1351 ~ 1359	SNC_AxisX_G59_6 ~ SNC_AxisW_G59_6	R/W	G59.6 of X~Z--axis, A~C-axis and U~W-axis
1361 ~ 1369	SNC_AxisX_G59_7 ~ SNC_AxisW_G59_7	R/W	G59.7 of X~Z--axis, A~C-axis and U~W-axis
1371 ~ 1379	SNC_AxisX_G59_8 ~ SNC_AxisW_G59_8	R/W	G59.8 of X~Z--axis, A~C-axis and U~W-axis
1381 ~ 1389	SNC_AxisX_G59_9 ~ SNC_AxisW_G59_9	R/W	G59.9 of X~Z--axis, A~C-axis and U~W-axis
1491 ~ 1499	SNC_AxisX_Mach_Fbk ~ SNC_AxisW_Mach_Fbk	R/W	Mechanical coordinate of X~Z--axis, A~C-axis and U~W-axis (Feedback / Gear)
1500	SNC_Work_Plane	R/W	The setting of work plane. Range is from 54 to 59 and its default value is 54.
1501 ~ 1509	SNC_AxisX_Mach_Pos ~ SNC_AxisW_Mach_Pos	R	Mechanical coordinate of X~Z--axis, A~C-axis and U~W-axis (Command / Gear)
1511 ~ 1519	SNC_AxisX_Work_Pos	R	Working coordinate of X~Z--axis, A~C-axis and U~W-axis = mechanical coordinate - work plane
1521 ~ 1529	SNC_AxisX_Target_Pos ~ SNC_AxisW_Target_Pos	R	Distance from X~Z--axis, A~C-axis and U~W-axis to target position = Target coordinate – mechanical coordinate
1531 ~ 1539	SNC_AxisX_G52 ~ SNC_AxisW_G52	R/W	G52 of X~Z--axis, A~C-axis and U~W-axis (offset of original point)
1541 ~ 1549	SNC_AxisX_G54 ~ SNC_AxisW_G54	R/W	G54 of X~Z--axis, A~C-axis and U~W-axis
1551 ~ 1559	SNC_AxisX_G55 ~ SNC_AxisW_G55	R/W	G55 of X~Z--axis, A~C-axis and U~W-axis
1561 ~ 1569	SNC_AxisX_G56 ~ SNC_AxisW_G56	R/W	G56 of X~Z--axis, A~C-axis and U~W-axis
1571 ~ 1579	SNC_AxisX_G57 ~ SNC_AxisW_G57	R/W	G57 of X~Z--axis, A~C-axis and U~W-axis
1581 ~ 1589	SNC_AxisX_G58 ~ SNC_AxisW_G58	R/W	G58 of X~Z--axis, A~C-axis and U~W-axis
1591 ~ 1599	SNC_AxisX_G59 ~ SNC_AxisW_G59	R/W	G59 of X~Z--axis, A~C-axis and U~W-axis

1601 ~ 1609	SNC_AxisX_Relative_Base ~ SNC_AxisW_Relative_Base	R/W	Standard of relative coordinate on X~Z--axis, A~C-axis and U~W-axis
1611 ~ 1619	SNC_AxisX_Relative ~ SNC_AxisW_Relative	R/W	Relative coordinate of X~Z--axis, A~C-axis and U~W-axis
Others			
1801	SNC_G_Group_MotionType	R	Motion mode, such as G81, G82, G83...
1808	SNC_ISO_Return	R	ISO. Return to G98/G99 after one cycle
1828	SNC_Feed_Rate_Running	R	Acquire the current feed rate
Tool Related Information			
1890	SNC_Tool_Max	R/W	Tool maximum number which is used by the system. If users do not setup the tool changing and G43, an error will occur. (Refer to SNC_Check_Tool_No)
1891	SNC_Tool_Multi_Set_On_One_Line	R/W	ISO. More than one T code can be set in one line.
1892	SNC_Tool_Multi_Part1_Count	R	ISO. The number of first T code group
1893	SNC_Tool_Multi_Part2_Count	R	ISO. The number of second T code group
1900	SNC_Tool_Len	R/W	Tool length compensation value SNC_User_Tool_Length_Compensation_Type sets to 1. Read this parameter after macro processing is done.
1901	SNC_Tool_Radius	R/W	Tool radius's compensation value. When parameter value = 0, the processing path remains.
Tool Management			
2001 ~ 2100	SNC_T1_Length ~ SNC_T100_Length	R/W	SNC_User_Tool_Length_Compensation_Type is set to 0. The compensation value will acquire from G43 and G44's parameters.
2201 ~ 2300	SNC_T1_Radius ~ SNC_T100_Radius	R/W	SNC_User_Tool_Cutter_Compensation_Type is set to 0. The compensation value acquire from G41 and G42's parameters.
2601 ~ 2700	SNC_T1_No ~ SNC_T100_No	R/W	Tool list; works with SNC_Check_Tool_No
2701 ~ 2800	SNC_Tool_Multi_No1 ~ SNC_Tool_Multi_No100	R	Tool number that being used in the same line

2821 ~ 2829	SNC_AxisX_G30_P2 ~ SNC_AxisW_G30_P2	R/W	The second middle coordinate of G30 on X~Z--axis, A~C-axis and U~W-axis
2831 ~ 2839	SNC_AxisX_G30_P3 ~ SNC_AxisW_G30_P3	R/W	The third middle coordinate of G30 on X~Z--axis, A~C-axis and U~W-axis.
2841 ~ 2849	SNC_AxisX_G30_P4 ~ SNC_AxisW_G30_P4	R/W	The fourth middle coordinate of G30 on X~Z--axis, A~C-axis and U~W-axis.
Cyclic Mode – User defined-parameters			
3000	SNC_ISO_Cycle_Do	R/W	ISO. When this parameter is set to 1, it means a cycle that is waiting to be executed. When it is done, use _SNC_set_parameter (N, SNC_ISO_Cycle_Do, 0) and SNC resumes.
3006	SNC_ISO_Cycle_Ori_Z	R	ISO. Initial position (absolute position_coordinate)
3007	SNC_ISO_Cycle_Ori_Z_Pulse	R	ISO. Initial position (absolute position _Pulse)
3008	SNC_ISO_Cycle_Target_Z	R	ISO. Target position (absolute position _ coordinate)
3009	SNC_ISO_Cycle_Target_Z_Pulse	R	ISO. Target position (absolute position _Pulse)
3010	SNC_ISO_Cycle_Level_R	R	ISO. Position R (absolute position _ coordinate)
3011	SNC_ISO_Cycle_Level_R_Pulse	R	ISO. Position R(absolute position _Pulse)
3012	SNC_ISO_Cycle_Wait	R	ISO. Waiting time
3014	SNC_ISO_Cycle_Offset_Q	R	ISO. Offset amount
3015	SNC_ISO_Cycle_Offset_Q_Pulse	R	ISO. Offset amount (Pulse)
3016	SNC_ISO_Cycle_Offset_d	R	ISO. Tool retrieval amount
3017	SNC_ISO_Cycle_FeedRate	R	ISO. Feed rate
3018	SNC_ISO_Cycle_FeedRate_Pulse	R	ISO. Feed rate (Pulse)
Cyclic Mode – Sets the Processing Mode (It is processed by SNC in default setting; however, SNC is not supported by tapping)			
3173	SNC_ISO_Cycle_Mode_73	R/W	ISO. G73 cyclic processing mode 0: Processed by SNC 1: Processed by users

3174	SNC_ISO_Cycle_Mode_74	R/W	ISO. G74 cyclic processing mode
3176	SNC_ISO_Cycle_Mode_76	R/W	ISO. G76 cyclic processing mode
3181	SNC_ISO_Cycle_Mode_81	R/W	ISO. G81 cyclic processing mode
3182	SNC_ISO_Cycle_Mode_82	R/W	ISO. G82 cyclic processing mode
3183	SNC_ISO_Cycle_Mode_83	R/W	ISO. G83 cyclic processing mode
3184	SNC_ISO_Cycle_Mode_84	R/W	ISO. G84 cyclic processing mode
3185	SNC_ISO_Cycle_Mode_85	R/W	ISO. G85 cyclic processing mode
3186	SNC_ISO_Cycle_Mode_86	R/W	ISO. G86 cyclic processing mode
3187	SNC_ISO_Cycle_Mode_87	R/W	ISO. G87 cyclic processing mode
3188	SNC_ISO_Cycle_Mode_88	R/W	ISO. G88 cyclic processing mode
3189	SNC_ISO_Cycle_Mode_89	R/W	ISO. G89 cyclic processing mode
Record the Tool Changing Order			
3199	SNC_ISO_PreScan_TNM6_Index	R	ISO. The current tool number (start from 0)
3200	SNC_ISO_PreScan_TNM6_Count	R	ISO. Tool number that is changed in G code
3201 ~ 3299	SNC_ISO_PreScan_TNM6_Start ~ SNC_ISO_PreScan_TNM6_End	R	ISO. Address that stores the first cutter ~ 99th cutter
Setting of each variable in G code. The setting value will determine the operation of G code			
3304	SNC_ISO_VAR_D_ALLOW_LESS_EQUAL_ZERO	R/W	ISO. Whether D variable could be smaller than or equal to 0. 0: no (default) 1: yes
3306	SNC_ISO_VAR_F_ALLOW_LESS_EQUAL_ZERO	R/W	ISO. Whether F variable could be smaller than or equal to 0. 0: no (default) 1: yes
3308	SNC_ISO_VAR_H_ALLOW_LESS_EQUAL_ZERO	R/W	ISO. Whether H variable could be smaller than or equal to 0. 0: no (default) 1: yes

3312	SNC_ISO_VAR_L_ALLOW_LESS_EQUAL_ZERO	R/W	ISO. Whether L variable could be smaller than or equal to 0. 0: no (default) 1: yes
3316	SNC_ISO_VAR_P_ALLOW_LESS_EQUAL_ZERO	R/W	ISO. Whether P variable could be smaller than or equal to 0. 0: no (default) 1: yes
3317	SNC_ISO_VAR_Q_ALLOW_LESS_EQUAL_ZERO	R/W	ISO. Whether Q variable could be smaller than or equal to 0. 0: no 1: yes (default)
3319	SNC_ISO_VAR_S_ALLOW_LESS_EQUAL_ZERO	R/W	ISO. Whether S variable could be smaller than or equal to 0. 0: no (default) 1: yes
3500 ~ 4499	SNC_Macro_Spend_Time_Start ~ SNC_Macro_Spend_Time_End	R/W	Time it takes from M0 to M999
EIA M Code			
4511 ~ 4519	SNC_EIA_AxisX_M11_M12_Offset ~ SNC_EIA_AxisW_M11_M12_Offset	R/W	EIA. Relative moving distance of M11(+) and M12(-) on X~Z--axis, A~C-axis and U~W-axis
4521 ~ 4529	SNC_EIA_AxisX_M72_M73_Offset ~ SNC_EIA_AxisW_M72_M73_Offset	R/W	EIA. Relative moving distance of M73(+) and M72(-) on X~Z--axis, A~C-axis and U~W-axis
4531 ~ 4539	SNC_EIA_AxisX_M274_M275_Offset ~ SNC_EIA_AxisW_M274_M275_Offset	R/W	EIA. Relative moving distance of M275 (+) and M274(-) on X~Z--axis, A~C-axis and U~W-axis
4541 ~ 4549	SNC_EIA_AxisX_M276_M277_Offset ~ SNC_EIA_AxisW_M276_M277_Offset	R/W	EIA. Relative moving distance of M277 (+) and M276(-) on X~Z--axis, A~C-axis and U~W-axis
4551 ~ 4559	SNC_EIA_AxisX_M278_M279_Offset ~ SNC_EIA_AxisW_M278_M279_Offset	R/W	EIA. Relative moving distance of M279 (+) and M278(-) on X~Z--axis, A~C-axis and U~W-axis
4561 ~ 4569	SNC_EIA_AxisX_M280_M281_Offset ~ SNC_EIA_AxisW_M280_M281_Offset	R/W	EIA. Relative moving distance of M281 (+) and M280(-) on X~Z--axis, A~C-axis and U~W-axis
4571 ~ 4579	SNC_EIA_AxisX_M282_M283_Offset ~ SNC_EIA_AxisW_M282_M283_Offs	R/W	EIA. Relative moving distance of M283 (+) and M282(-) on X~Z--axis, A~C-axis and U~W-axis

	et		
4581 ~ 4589	SNC_EIA_AxisX_M284_M285_Offset ~ SNC_EIA_AxisW_M284_M285_Offset	R/W	EIA. Relative moving distance of M285 (+) and M284(-) on X~Z--axis, A~C-axis and U~W-axis
4591 ~ 4599	SNC_EIA_AxisX_M286_M287_Offset ~ SNC_EIA_AxisW_M286_M287_Offset	R/W	EIA. Relative moving distance of M287 (+) and M286(-) on X~Z--axis, A~C-axis and U~W-axis
4601 ~ 4609	SNC_EIA_AxisX_M288_M289_Offset ~ SNC_EIA_AxisW_M288_M289_Offset	R/W	EIA. Relative moving distance of M289 (+) and M288(-) on X~Z--axis, A~C-axis and U~W-axis
4611 ~ 4619	SNC_EIA_AxisX_M290_M291_Offset ~ SNC_EIA_AxisW_M290_M291_Offset	R/W	EIA. Relative moving distance of M291 (+) and M290 (-) on X~Z--axis, A~C-axis and U~W-axis
4621 ~ 4629	SNC_EIA_AxisX_M292_M293_Offset ~ SNC_EIA_AxisW_M292_M293_Offset	R/W	EIA. Relative moving distance of M293 (+) and M292 (-) on X~Z--axis, A~C-axis and U~W-axis
4711 ~ 4719	SNC_EIA_AxisX_M79_Home_Position1 ~ SNC_EIA_AxisW_M79_Home_Position1	R/W	EIA. The absolute position of M79 T1 moves on X~Z--axis, A~C-axis and U~W-axis
4721 ~ 4729	SNC_EIA_AxisX_M79_Home_Position2 ~ SNC_EIA_AxisW_M79_Home_Position2	R/W	EIA. The absolute position of M79 T2 moves on X~Z--axis, A~C-axis and U~W-axis
4731 ~ 4739	SNC_EIA_AxisX_M79_Home_Position3 ~ SNC_EIA_AxisW_M79_Home_Position3	R/W	EIA. The absolute position of M79 T3 moves on X~Z--axis, A~C-axis and U~W-axis
4741 ~ 4749	SNC_EIA_AxisX_M79_Home_Position4 ~ SNC_EIA_AxisW_M79_Home_Position4	R/W	EIA. The absolute position of M79 T4 moves on X~Z--axis, A~C-axis and U~W-axis
4751 ~ 4759	SNC_EIA_AxisX_M79_Home_Position5 ~ SNC_EIA_AxisW_M79_Home_Position5	R/W	EIA. The absolute position of M79 T5 moves on X~Z--axis, A~C-axis and U~W-axis
4761 ~ 4769	SNC_EIA_AxisX_M79_Home_Position6 ~ SNC_EIA_AxisW_M79_Home_Position6	R/W	EIA. The absolute position of M79 T6 moves on X~Z--axis, A~C-axis and U~W-axis
4771 ~ 4779	SNC_EIA_AxisX_M79_Home_Position7 ~ SNC_EIA_AxisW_M79_Home_Position7	R/W	EIA. The absolute position of M79 T7 moves on X~Z--axis, A~C-axis and U~W-axis

4781 ~ 4789	SNC_EIA_AxisX_M79_Home_Position8 ~ SNC_EIA_AxisW_M79_Home_Position8	R/W	EIA. The absolute position of M79 T8 moves on X~Z--axis, A~C-axis and U~W-axis
4791 ~ 4799	SNC_EIA_AxisX_M79_Home_Position9 ~ SNC_EIA_AxisW_M79_Home_Position9	R/W	EIA. The absolute position of M79 T9 moves on X~Z--axis, A~C-axis and U~W-axis
4801 ~ 4809	SNC_EIA_AxisX_M79_Home_Position10 ~ SNC_EIA_AxisW_M79_Home_Position10	R/W	EIA. The absolute position of M79 T10 moves on X~Z--axis, A~C-axis and U~W-axis
4811 ~ 4819	SNC_EIA_AxisX_M79_Home_Position11 ~ SNC_EIA_AxisW_M79_Home_Position11	R/W	EIA. The absolute position of M79 T11 moves on X~Z--axis, A~C-axis and U~W-axis
4821 ~ 4829	SNC_EIA_AxisX_M79_Home_Position12 ~ SNC_EIA_AxisW_M79_Home_Position12	R/W	EIA. The absolute position of M79 T12 moves on X~Z--axis, A~C-axis and U~W-axis
4831 ~ 4839	SNC_EIA_AxisX_M79_Home_Position13 ~ SNC_EIA_AxisW_M79_Home_Position13	R/W	EIA. The absolute position of M79 T13 moves on X~Z--axis, A~C-axis and U~W-axis
4841 ~ 4849	SNC_EIA_AxisX_M79_Home_Position14 ~ SNC_EIA_AxisW_M79_Home_Position14	R/W	EIA. The absolute position of M79 T14 moves on X~Z--axis, A~C-axis and U~W-axis
4851 ~ 4859	SNC_EIA_AxisX_M79_Home_Position15 ~ SNC_EIA_AxisW_M79_Home_Position15	R/W	EIA. The absolute position of M79 T15 moves on X~Z--axis, A~C-axis and U~W-axis
4861 ~ 4869	SNC_EIA_AxisX_M79_Home_Position16 ~ SNC_EIA_AxisW_M79_Home_Position16	R/W	EIA. The absolute position of M79 T16 moves on X~Z--axis, A~C-axis and U~W-axis
EIA Variable			
5000	SNC_EIA_D_Var_Start	R/W	EIA. Variable D starts (Range is from 5000 ~ 5999, and it is unchangeable)
5201	SNC_EIA_G66_Rotate_Angle	R/W	EIA. The rotate angle after G66 executed.
5301	SNC_EIA_D_Var_301	R/W	EIA. Feed rate speed of G00
5302	SNC_EIA_D_Var_302	R/W	EIA. Default feed rate speed of G01
5303	SNC_EIA_D_Var_303	R/W	EIA. Default feed rate speed of G02, G03

5999	SNC_EIA_D_Var_End	R/W	EIA. Variable D ends (Range is from 5000 ~ 5999, and it is unchangeable)
6000	SNC_EIA_V_Var_Start	R/W	EIA. Variable V starts (Range is from 6000 ~ 6999, and it is unchangeable)
6999	SNC_EIA_V_Var_End	R/W	EIA. Variable V ends (Range is from 6000 ~ 6999, and it is unchangeable)
Variable			
50000 ~100000	SNC_Sharp_Variables_Start ~ SNC_Sharp_Variables_End	R/W	Address that stores variable#
100000 ~150000	SNC_At_Variables_Start ~ SNC_At_Variables_End	R/W	Address that stores variable@
150000 ~200000	SNC_Registry_Start ~ SNC_Registry_End	R/W	Register address

(This page is intentionally left blank.)

Description of SNC Error Parameter

9

This chapter can be regarded as the list for looking up error codes or error parameters during SNC operation.

9.1 List of error parameters

Table 9-1

Data and Information Error		
1	ERR_FILE_NOT_EXIST	File does not exist
2	ERR_NO_DATA	No string is found
3	ERR_DATA_NOT_COMPLETE	File string error
4	ERR_START_OVER	The start line exceeds the total line number
5	ERR_DMC_01_DLL_Not_Full_Version	PCI_DMC_01.dll, wrong version
6	ERR_CUTTING_LINE_TOO_SHORT	Error / warning occurs when the cutting line is too short
7	ERR_GOTO_LINE_WRONG	Use goto but cannot find the corresponding tag
8	ERR_GOTO_LINE_REDEFINED	Use goto but tag is duplicate
Duplicate Definition		
101	ERR_GCODE_MULTIPLE_A_WORDS_ON_ONE_LINE	Duplicate definition of variable A in the same line of G code
102	ERR_GCODE_MULTIPLE_B_WORDS_ON_ONE_LINE	Duplicate definition of variable B in the same line of G code
103	ERR_GCODE_MULTIPLE_C_WORDS_ON_ONE_LINE	Duplicate definition of variable C in the same line of G code
104	ERR_GCODE_MULTIPLE_D_WORDS_ON_ONE_LINE	Duplicate definition of variable D in the same line of G code
105	ERR_GCODE_MULTIPLE_E_WORDS_ON_ONE_LINE	Duplicate definition of variable E in the same line of G code
106	ERR_GCODE_MULTIPLE_F_WORDS_ON_ONE_LINE	Duplicate definition of variable F in the same line of G code
107	ERR_GCODE_MULTIPLE_H_WORDS_ON_ONE_LINE	Duplicate definition of variable H in the same line of G code
108	ERR_GCODE_MULTIPLE_I_WORDS_ON_ONE_LINE	Duplicate definition of variable I in the same line of G code
109	ERR_GCODE_MULTIPLE_J_WORDS_ON_ONE_LINE	Duplicate definition of variable J in the same line of G code
110	ERR_GCODE_MULTIPLE_K_WORDS_ON_ONE_LINE	Duplicate definition of variable K in the same line of G code
111	ERR_GCODE_MULTIPLE_L_WORDS_ON_ONE_LINE	Duplicate definition of variable L in the same line of G code
112	ERR_GCODE_MULTIPLE_M_WORDS_ON_ONE_LINE	Duplicate definition of variable M in the same line of G code
113	ERR_GCODE_MULTIPLE_P_WORDS_ON_ONE_LINE	Duplicate definition of variable P in the same line of G code

114	ERR_GCODE_MULTIPLE_Q_WORDS_ON_ONE_LINE	Duplicate definition of variable Q in the same line of G code
115	ERR_GCODE_MULTIPLE_R_WORDS_ON_ONE_LINE	Duplicate definition of variable R in the same line of G code
116	ERR_GCODE_MULTIPLE_S_WORDS_ON_ONE_LINE	Duplicate definition of variable S in the same line of G code
117	ERR_GCODE_MULTIPLE_T_WORDS_ON_ONE_LINE	Duplicate definition of variable T in the same line of G code
118	ERR_GCODE_MULTIPLE_U_WORDS_ON_ONE_LINE	Duplicate definition of variable U in the same line of G code
119	ERR_GCODE_MULTIPLE_V_WORDS_ON_ONE_LINE	Duplicate definition of variable V in the same line of G code
120	ERR_GCODE_MULTIPLE_W_WORDS_ON_ONE_LINE	Duplicate definition of variable W in the same line of G code
121	ERR_GCODE_MULTIPLE_X_WORDS_ON_ONE_LINE	Duplicate definition of variable X in the same line of G code
122	ERR_GCODE_MULTIPLE_Y_WORDS_ON_ONE_LINE	Duplicate definition of variable Y in the same line of G code
123	ERR_GCODE_MULTIPLE_Z_WORDS_ON_ONE_LINE	Repeated definition of variable Z in the same line of G code
Negative Variable Number		
201	ERR_GCODE_NEGATIVE_D_WORD	Variable D is a negative number
202	ERR_GCODE_NEGATIVE_F_WORD	Variable F is a negative number
203	ERR_GCODE_NEGATIVE_G_WORD	Variable G is a negative number
204	ERR_GCODE_NEGATIVE_H_WORD	Variable H is a negative number
205	ERR_GCODE_NEGATIVE_L_WORD	Variable L is a negative number
206	ERR_GCODE_NEGATIVE_M_WORD	Variable M is a negative number
207	ERR_GCODE_NEGATIVE_P_WORD	Variable P is a negative number
Unknown Variables		
301	ERR_GCODE_BAD_CHARACTER	Variable range is not within A to Z
302	ERR_GCODE_UNKNOWN_CHARACTER	Unknown variable
303	ERR_GCODE_UNKNOWN_G_CODE	Unknown G code
304	ERR_GCODE_UNKNOWN_WORD_STARTING_WITH_A	Unknown function starting with A
305	ERR_GCODE_UNKNOWN_WORD_STARTING_WITH_C	Unknown function starting with C
306	ERR_GCODE_UNKNOWN_WORD_STARTING_WITH_E	Unknown function starting with E
307	ERR_GCODE_UNKNOWN_WORD_STARTING_WITH_F	Unknown function starting with F

308	ERR_GCODE_UNKNOWN_WORD_STARTING_WITH_L	Unknown function starting with L
309	ERR_GCODE_UNKNOWN_WORD_STARTING_WITH_P	Unknown function starting with P
310	ERR_GCODE_UNKNOWN_WORD_STARTING_WITH_R	Unknown function starting with R
311	ERR_GCODE_UNKNOWN_WORD_STARTING_WITH_S	Unknown function starting with S
312	ERR_GCODE_UNKNOWN_WORD_STARTING_WITH_T	Unknown function starting with T
313	ERR_GCODE_UNKNOWN_OPERATION	Unknown operator
314	ERR_GCODE_BUG_UNKNOWN_OPERATION	Unknown operation
315	ERR_GCODE_UNKNOWN_OPERATION_NAME_STARTING_WITH_A	Unknown operator starting with A
316	ERR_GCODE_UNKNOWN_OPERATION_NAME_STARTING_WITH_M	Unknown operator starting with M
317	ERR_GCODE_UNKNOWN_OPERATION_NAME_STARTING_WITH_O	Unknown operator starting with O
318	ERR_GCODE_UNKNOWN_OPERATION_NAME_STARTING_WITH_X	Unknown operator starting with X
319	ERR_GCODE_UNKNOWN_WORD_WHERE_UNARY_OPERATION_COULD_BE	Unknown function
Exceeding the Range		
403	ERR_GCODE_M_CODE_TOO_BIG	M code exceeds the range, 0 ~ 255
405	ERR_GCODE_PARAMETER_NUMBER_OUT_OF_RANGE	SNC parameter exceeds the range, 0 ~ 5000
406	ERR_GCODE_H_WORD_EMPTY	H code has not set yet
408	ERR_GCODE_GLOBAL_PARAMETER_NUMBER_OUT_OF_RANGE	Variable @ exceeds the range
Others		
-1	ERR_AXIS_UNKNOWN	Unknown axis, X ~ Z, A ~ C, U ~ W
501	ERR_GCODE_NEGATIVE_OR_ZERO_Q_VALUE	Value Q is smaller than or equals to 0
502	ERR_GCODE_NEGATIVE_SPINDLE_SPEED	Value S is smaller than 0
503	ERR_GCODE_NEGATIVE_TOOL_ID	Value T is smaller than 0
504	ERR_GCODE_TWO_G_CODES_USED_FROM_SAME_MODAL_GROUP	Duplicate setting of G code group
511	ERR_GCODE_G51_AXES_NOT_EQUAL_TWO	Scaling value has to set with two axes
512	ERR_GCODE_G51_X_SCALE_VALUE_ZER	Scaling point is set, but scaling value on

	O	X axis is not assigned
513	ERR_GCODE_G51_Y_SCALE_VALUE_ZERO	Scaling point is set, but scaling value on Y axis is not assigned
514	ERR_GCODE_G51_Z_SCALE_VALUE_ZERO	Scaling point is set, but scaling value on Z axis is not assigned
516	ERR_GCODE_G51_1_AXES_NOT_ASSIGN	Mirror axis is not assigned
522	ERR_GCODE_G68_ROTATE_ANGLE_NOT_ASSING	Angle of rotation is not assigned
530	ERR_Cycle_Repet_Cnt_Negative	The repeated number of peck drilling cycle is negative
701	ERR_SNC_INITIAL_FAILED	SNC initialized failed
702	ERR_CANT_SET_WHEN_PROCESSING	Parameter cannot be accesssed during processing
703	ERR_AXIS_OUT_OF_RNG	Axis number exceeds the range,1 ~ 9
704	ERR_AXIS_REDEFINE	Axis number is duplicate
706	ERR_AXES_ZERO	Total axis amount cannot be zero
707	ERR_AXES_OUT_OF_RNG	Total axis amount exceeds the range, 1 ~ 9
708	ERR_MACRO_MODE_OUT_OF_RNG	The setting of macro mode is wrong
709	ERR_CALLBACK_NULL	CallBack function is not assigned
711	ERR_GEAR_ZERO	E-gear ratio is zero
713	ERR_G00_SPD_ZERO	G00 speed cannot be zero
715	ERR_WRONG_PLANE	G code plane setting error
723	ERR_MACRO_OVER_RNG	Layer of sub function exceeds four layers
727	ERR_G02_G03_PARAM	Work plane used by G02 and G03 does not match to the parameter
728	ERR_G02_G03_PLANE	G02 and G03 use the wrong work plane
729	ERR_G02_G03_CALC	G02 and G03 cannot calculate coordinate
730	ERR_G02_G03_AXES_OVER	Axis amount used by G02 and G03 exceeds 3
731	ERR_PROCESSING_IS_RUNNING	The system is in process. The setting is invalid
732	ERR_TOOL_MAX_OVER_RNG	Tool number exceeds the range (1 ~100)
733	ERR_CUTTER_COMPENSATION_ARC_PLANE_NOT_SUPPORT	Function Tool radius compensation only supports XY plane
734	ERR_CUTTER_COMPENSATION_CANT_USE_HELI	Tool radius compensation cannot be used in HELI

735	ERR_CUTTER_COMPENSATION_CALC	Tool radius compensation cannot be calculated
736	ERR_CUTTER_COMPENSATION_MEMORY	Memory error during tool radius compensation
737	ERR_CUTTER_FIRST_MOTION_ARC	The first motion of tool radius compensation shall be straight line
738	ERR_CUTTER_NOT_FINISH	Macro function is about to be executed, but function of tool radius compensation is not complete.
741	ERR_MEMORY_ALLOC_FAIL	SNC memory accessing error
742	ERR_USER_CALLBACK_NULL	Use the function of User Macro but does not set up callback function
761	ERR_ISO_CYCLE_MODE_OUT_OF_RNG	Wrong setting of drilling mode
762	ERR_CALLBACK_ISO_CYCLE_NULL	The macro function in drilling mode does not setup callback function
763	ERR_ISO_CYCLE_NOT_SUPPORT	Cycle function is not supported
802	ERR_SETTING_GEAR	Gear setting error
803	ERR_SETTING_AXIS	Wrong axis number
805	ERR_TOOL_RADIUS_TOO_SHORT	When applying G43 and G44, the value of tool radius is too small.
806	ERR_SETTING_TOOL_MAX_ZERO	Confirm the tool but has not setup the tool number
807	ERR_SETTING_DIRECT	Wrong direction (-1, 1)
809	ERR_SETTING_UINT	Unit setting error
810	ERR_TOOL_PARTS_OVER_RNG	T code group number in the same line exceeds the range (use ‘/’ to separate)
901	ERR_DEVICE_04PI_MODE1	Device is 04PI Mode1
902	ERR_DEVICE_RM_MODULE	Device is RM module
903	ERR_DEVICE_NO_DEVICE	This station is unable to find the device
904	ERR_DEVICE_UNKNOWN	Unknown device
911	ERR_API_ERRNO	Basic function returns error. Please access parameter SNC_API_ErrNo
1001	ERR_GROUP_INIT_FIRST	Group number has not been set up
1002	ERR_GRUOP_OVER_RANGE	Operation group exceeds the setting range
1004	ERR_GRUOP_CARD_TYPE	Wrong card type
1101	ERR_EIA_G59_NO_PARAMS	EIA G59 parameter error
1102	ERR_EIA_G59_D_VAR_NO_X	EIA D variable error

1103	ERR_EIA_G59_V_VAR_NO_F	EIA V variable error
1104	ERR_EIA_CALL_FUNC_PATH	EIA. The system cannot find the file specified by macro
1105	ERR_EIA_TOO_MUCH_M40	M40 function code using error. M40 comes after the same function code (M40).
1106	ERR_EIA_TOO_MUCH_M41	M41 function code using error. M41 comes after the same function code (M41).
1107	ERR_EIA_M40_NOT_M41	M40 and M41 using error. (without M41)
1108	ERR_EIA_TOO_MUCH_G97	G97 function code using error. G97 comes after the same function code (G97).
1109	ERR_EIA_TOO_MUCH_G98	G98 function code using error. G98 comes after the same function code (G98).
1110	ERR_EIA_G97_NOT_G98	G97 and G98 using error. (without G98)
1111	ERR_EIA_M79_OVER_RNG	EIA M79 exceeds the range (1 ~ 16)
2000	ERR_TRIGGER_SOFT_LIMIT	Software limit is triggered
9999	ERR_SECURITY_FAILED	Security authentication failed

(This page is intentionally left blank.)

10 Caution Note

When applying the Delta's SNC function, please pay special attention to SNC initialized setting, SNC default value, G43 and G44 program scanning and instruction of SNC gear that mentioned in this chapter.

10.1 SNC Initialized Setting

Please use _SNC_initial function initialization before setting up each SNC parameter. This is for avoiding the parameter being cleared or covered during initialization.

10.2 G43 and G44 Program Pre-Scanning

When SNC_User_Tool_Length_Compensation_Type is 1, it means the system has to measures the tool length first before confirming the value. Scanning of G43 and G44 will be skipped because the compensation value is unknown.

10.3 Instruction of SNC Gear

Procedure of applying E-gear is shown as below:

(1) _SNC_group_init

(2) _SNC_initial

(3) Setup each parameter of SNC

(4) Setup parameter Gear (SNC_AxisX_Gear ~ SNC_AxisW_Gear)

(5) Use DMC API to activate the motion card.

(6) Set the value of Position and Command to zero

(7) _SNC_setup_hw

(8) When applying _SNC_setup_hw, make sure the value of Comamnd and Position is zero.

10.4 SNC Default Value

Table 10-1

Parameter	Default Value	Description
SNC_Fix_Slope	0	For acceleration and deceleration: 0: time 1: acceleration
SNC_AxisX_Dir	1	Moving direction of X-axis 1: same as the program -1: opposite direction
SNC_AxisY_Dir	1	Moving direction of Y-axis 1: same as the program -1: opposite direction
SNC_AxisZ_Dir	1	Moving direction of Z-axis 1: same as the program -1: opposite direction
SNC_AxisA_Dir	1	Moving direction of A-axis 1: same as the program -1: opposite direction
SNC_AxisB_Dir	1	Moving direction of B-axis 1: same as the program -1: opposite direction
SNC_AxisC_Dir	1	Moving direction of C-axis 1: same as the program -1: opposite direction
SNC_AxisU_Dir	1	Moving direction of U-axis 1: same as the program -1: opposite direction
SNC_AxisV_Dir	1	Moving direction of V-axis 1: same as the program -1: opposite direction
SNC_AxisW_Dir	1	Moving direction of W-axis 1: same as the program -1: opposite direction
SNC_Feed_Rate_G00	1	In default setting, G00 speed is F1 (mm / per minute)
SNC_Tacc_G00	0.1	In default setting, G00 acceleration time is 0.1 second.
SNC_Tdec_G00	0.1	In default setting, G00 deceleration time is 0.1 second.
SNC_Feed_Rate_G01	1	In default setting, G01 speed is F1 (mm / per minute)
SNC_Tacc_G01	0.1	In default setting, G01 acceleration time is 0.1 second.
SNC_Tdec_G01	0.1	In default setting, G01 deceleration time is 0.1 second.
SNC_Feed_Rate_Circle	1	In default setting, G02 and G03 speed is F1 (mm / per minute)

SNC_Tacc_Circle	0.1	In default setting, G02 and G03 acceleration time is 0.1 second.
SNC_Tdec_Circle	0.1	In default setting, G02 and G03 deceleration time is 0.1 second.
SNC_Feed_Rate_Reverse	1	In default setting, reverse speed is F1 (mm / per minute)
SNC_Tacc_Reverse	0.1	In default setting, acceleration time of reverse is 0.1 second.
SNC_Tdec_Reverse	0.1	In default setting, deceleration time of reverse is 0.1 second.
SNC_Work_Plane	54	In default setting, the working plane is G54.

11 Special Parameters

Part of special parameters mentioned in Chapter 8 are illustrated in this chapter.

11.1 Table of SNC_Use_Look_Ahead acceleration / deceleation parameters

Table 11-1

G Code Type	When SNC_Use_Look_Ahead = 1:	When SNC_Use_Look_Ahead = 0:
G00	SNC_Tacc_G00(1172), SNC_Tdec_G00(1173)	
G01	SNC_AxisX_PermitMaxAcc (991) ~ SNC_AxisW_PermitMaxAcc (999)	SNC_Tacc_G01 (1175), SNC_Tdec_G01 (1176)
G02, G03	SNC_AxisX_PermitMaxAcc (991) ~ SNC_AxisW_PermitMaxAcc (999)	SNC_Tacc_Circle (1178), SNC_Tdec_Circle (1179)

11.2 SNC_Tolerance

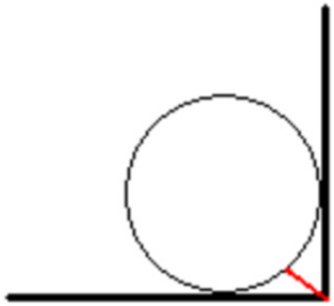


Figure 11-1

Description: Distance from the rim of arc to right angle

11.3 SNC_Circle_Tolerance

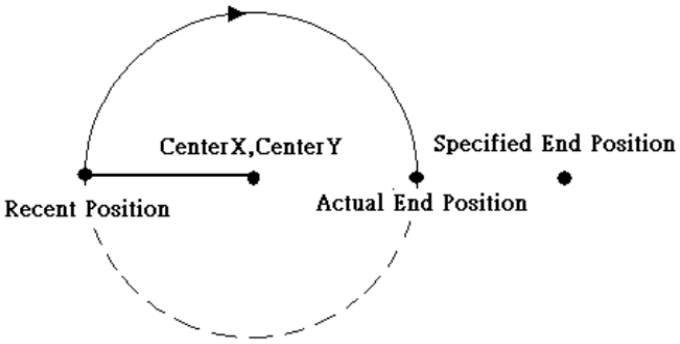
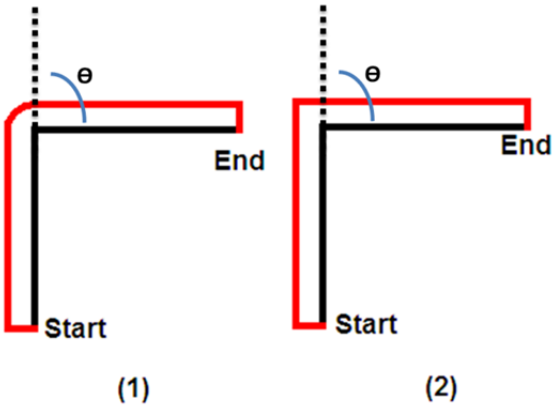


Figure 11-2

Description: When the difference between expected and actual position exceeds the setting error, SNC will return ERR_G02_G03_CALC and calculate the arc error.

11.4 SNC_Kerf_Permit_Angle mode 1

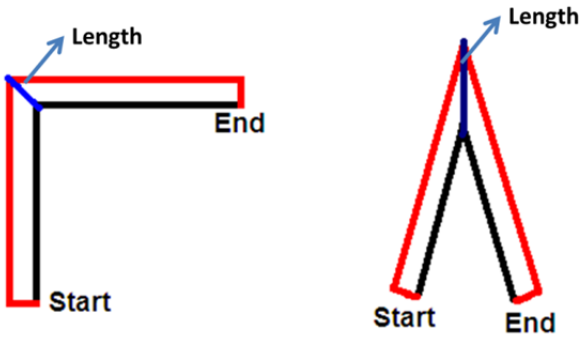


Description: There are two paths when tool radius compensation is applied to two linear commands:

(1) When included angle between two lines (Θ) > the value of SNC_Kerf_Permit_Angle, please apply to figure (1).

(2) When included angle between two lines (Θ) < the value of SNC_Kerf_Permit_Angle, please apply to figure (2).

11.5 SNC_Kerf_Permit_Angle mode 2



Description: When applying the second path, distance (Length) between the intersection point of original line and the one in tool compensation line is different. See the figure below:

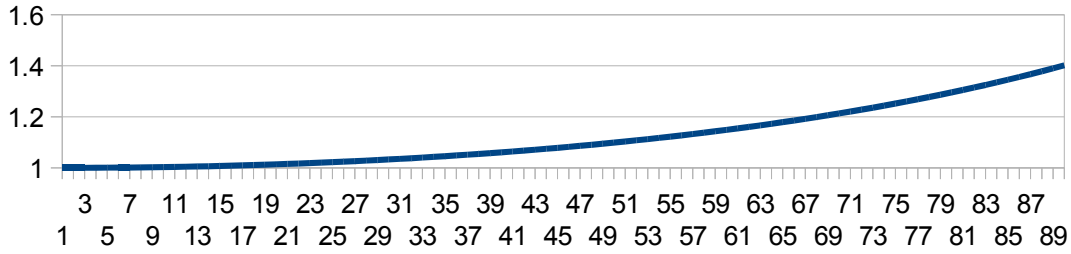


Figure 11-3: X axis: included angle between two lines (0 ~ 90 degrees); Y axis: proportion (actual distance (Length) = proportion * tool radius)

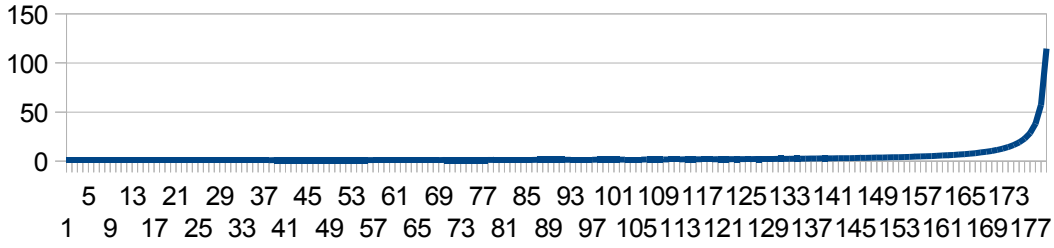


Figure 11-4: X axis: included angle between two lines (0 ~ 180 degrees); Y axis: proportion (actual distance (Length) = proportion * tool radius)

11.6 SNC_ PermitMaxDec

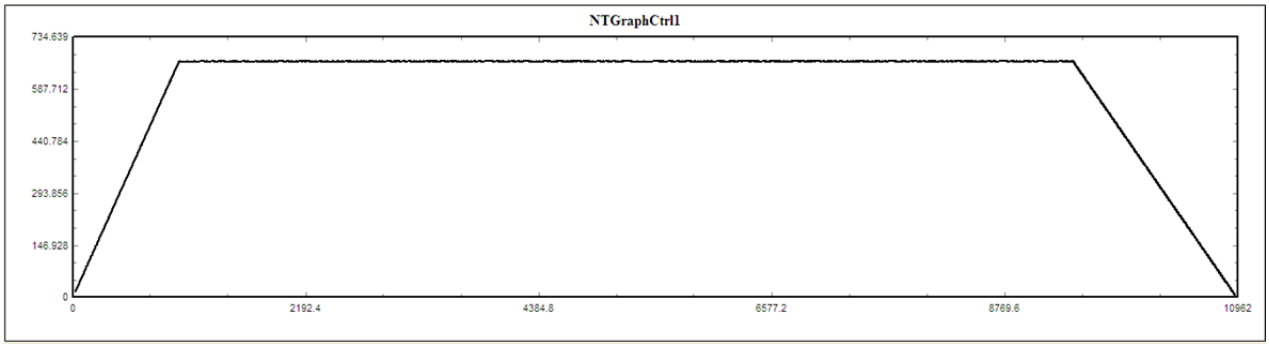


Figure 11-5: Parameter is set to 0

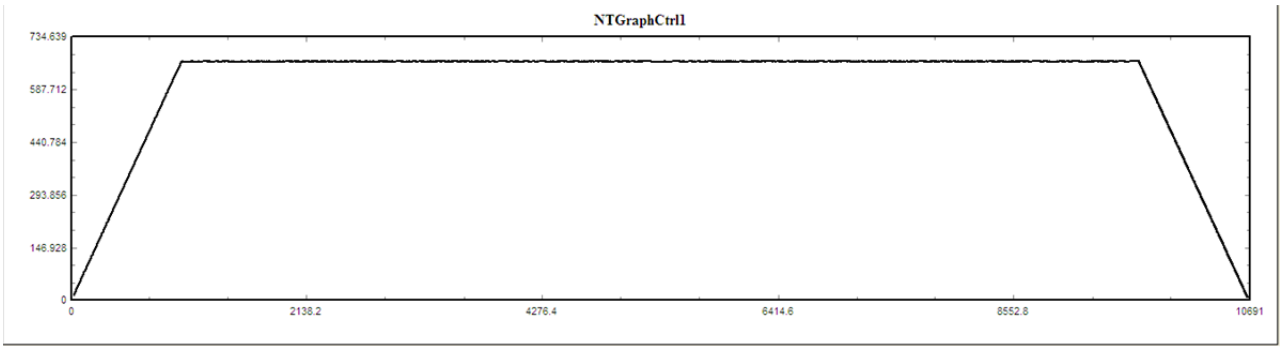


Figure 11-6: Parameter is set to 100

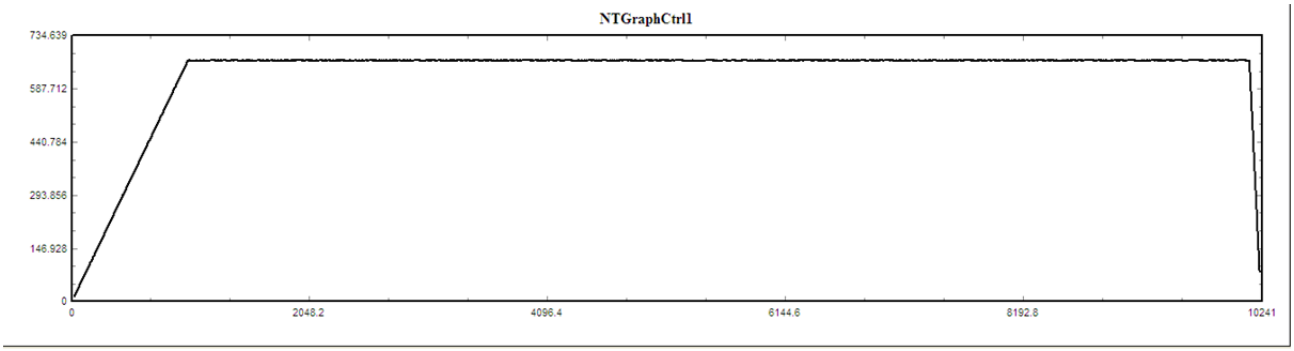
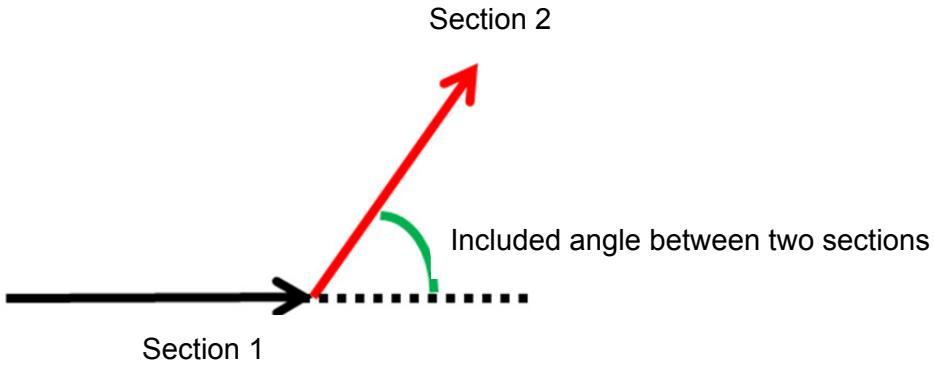


Figure 11-7: Parameter is set to 1000

Description: This parameter influences the motion when Look Ahead decelerates to stop. When the value is smaller, it means the machine tolerates smaller speed variation. That is, speed change will be moderate. If the value is bigger, speed change becomes abrupt.

11.7 SNC_Corner_Control



Description:

When SNC_Corner_Control is enabled, if the included angle between two lines is over the value of SNC_Corner_Angle, corner speed is equal to SNC_Corner_Speed; If included angle between two lines is smaller than the value of SNC_Corner_Angle, corner speed is the one calculated by LookAhead.

When SNC_Corner_Control is disabled, corner speed is the one calculated by LookAhead.

G code supporting table 12

Users can look up G code types and corresponding functions supported by Delta's SNC in G code supporting table.

12.1 G code supporting table

Table 12-1

G code	Group	Function description
G00	01	Rapid positioning command
G01	01	Linear interpolation command
G02	01	Clockwise circular interpolation (CW)
G03	01	Counterclockwise circular interpolation (CCW)
G04	00	Pause command (Dwell)
G09	00	Exact stop check
G17	02	X-Y plane selection
G18	02	Z-X plane selection
G19	02	Y-Z plane selection
G20	06	Programming in inches
G21	06	Programming in millimeters (mm)
G28	00	Homing through the reference origin point
G29	00	Homing to the starting point
G30	00	Auto homing of the second, third, and fourth reference point
G40	07	Tool radius compensation cancelling
G41	07	Tool radius left compensation
G42	07	Tool radius right compensation
G43	08	Tool length positive direction compensation
G44	08	Tool length negative direction compensation
G49	08	Tool length compensation cancelling
G50	11	Scale cutting cancelling
G51	11	Scale cutting
G52	00	Local coordinate system setup
G53	00	Mechanical coordinate system setup
G54	12	The first machining coordinate system selection

G55	12	The second machining coordinate system selection
G56	12	The third machining coordinate system selection
G57	12	The fourth machining coordinate system selection
G58	12	The fifth machining coordinate system selection
G59	12	The sixth machining coordinate system selection
G61	13	Exact stop check mode
G64	13	cutting mode (cancel exact stop check mode)
G65	00	Non-continuous effect macro command calling
G66	14	Continuous effect macro command calling
G67	14	Continuous effect macro command calling cancelling
G68	15	Coordinate system rotation command
G69	15	Coordinate system rotation command cancelling
G80	09	Constant loop cancelling
G81	09	Drilling cycle
G82	09	Countersunk drilling cycle
G83	09	Deep hole peck drilling cycle
G85	09	Right spiral tapping cycle
G90	03	Absolute coordinate value system
G91	03	Incremental coordinate value system
G92	00	Coordinate system setup
G98	10	Return to the initial point of the fixed cycle
G99	10	Return to the R point of the fixed cycle

* Each G code function with the same group cannot be programed in the same line of a G code file.

(This page is intentionally left blank.)