SIEMENS

Application description • 04/2014

Resuming to drill a pattern of holes (Block Search)

SINUMERIK 840D sl

http://support.automation.siemens.com/WW/view/en/89770537

Warranty and liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These application examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.
If there are any deviations between the recommendations provided in these application examples and other Siemens publications − e.g. Catalogs − the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of Siemens Industry Sector.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens’ products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit http://support.automation.siemens.com.

Table of contents

Warranty and liability 2

1 Task 4

1.1 Aim of this application 4
1.2 Resuming with Block Search 4
1.3 Scenario 5

2 Cases of use 6

2.1 Case n.1: holes positions programmed one-by-one 6
2.2 Case n.2: holes positions programmed with DIN cycle 8
2.3 Case n.3: holes positions programmed with Shopmill 9

3 Solution for Case n.2 10

3.1 Automatic call of a sub-routine after Block Search (to show a screen) 10
3.2 Example of a customized screen for Case n.2 12
3.3 Adaptation of the DIN-ISO program for Case n.2 15

4 Related literature 17

5 Contact 17

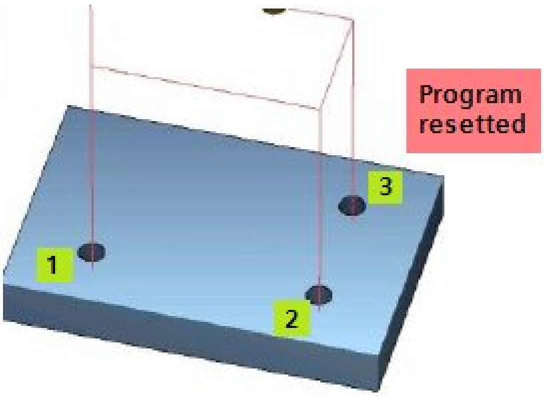
6 History 17

1 Task

1.1 Aim of this application

The design of a program, for the drilling of a pattern of holes, is easily possible with SINUMERIK Operate. During the machining, however, the program could be reset because of an error or because of a manual reset (e.g. when boring deep holes).

Figure 1-1 Machining of a pattern with 4 holes in a pattern, interrupted (reset)



How can the program being restarted, after the reset, from the last drilled position and not from the beginning? How can the manufacturer provide this resuming?

1.2 Resuming with Block Search

“Block Search” is provided in SINUMERIK Operate as a standard.

When the program has been resetted, the softkey “Block search” can be pressed. SINUMERIK Operate lets then to choose, how to prepare the program for the resuming (here is a simplified description):

- With calculation: the calculations, which are stated in the program before the seeked block, will be taken into account for the resuming
  - with approach (to contour): the end position of the block which is prior to the target block is found with <CYCLESTART>. The program runs in the same way as in normal program processing. This is used to be able to approach the contour in any circumstance.
  - without approach (block end point): the end position of the target block (or the next programmed position) is approached, using the type of interpolation valid in the target block. Only the axes programmed in the target block are moved. If machine data 11450 SEARCH_RUN_MODE bit 1=1 ist set, the rotary axes of the active swivel data record ar pre-positioned after Block Search. If necessary, a collision-free initial situation must be created manually on the machine in JOG REPOS mode before starting the program execution. This is used to approach a target machine position in the program (e. g. tool change position).
- Without calculation: the Block Search will find the desired block, but without performing the calculation stated in the part-program (e.g. without outputting auxiliary M-codes or action blocks after the NC Start). With this search mode, a desired block can be found very fast, when it’s not necessary to resume the program for the machining.

- With program test (SERUPRO): the NC starts the selected program in the program test mode, e.g. perform multi-channel Block Search with calculation. If the NC reaches the specified target block in the actual channel, it stops at the beginning of the target block and deselects program test mode. After continuing the program with NC start (after REPOS motion) the auxiliary functions of the target block are output.

NOTE More possibilities for “Block search” are explained in the official Sinumerik documentation.

1.3 Scenario

To describe the examples in this document:

- SINUMERIK 840D sl version 4.5 SP2
- SINUMERIK Operate version 4.5 SP2 with Run MyScreen

have been used.

2 Cases of use

This chapter provides a background description of the different situations, where Block Search can be used to resume a drilling program. Here are described 3 Cases of use, about how a drilling process can be programmed and, as a consequence, how Block Search can differently be used.

2.1 Case n.1: holes positions programmed one-by-one

In *Case n.1*, the following DIN-ISO program will be considered:

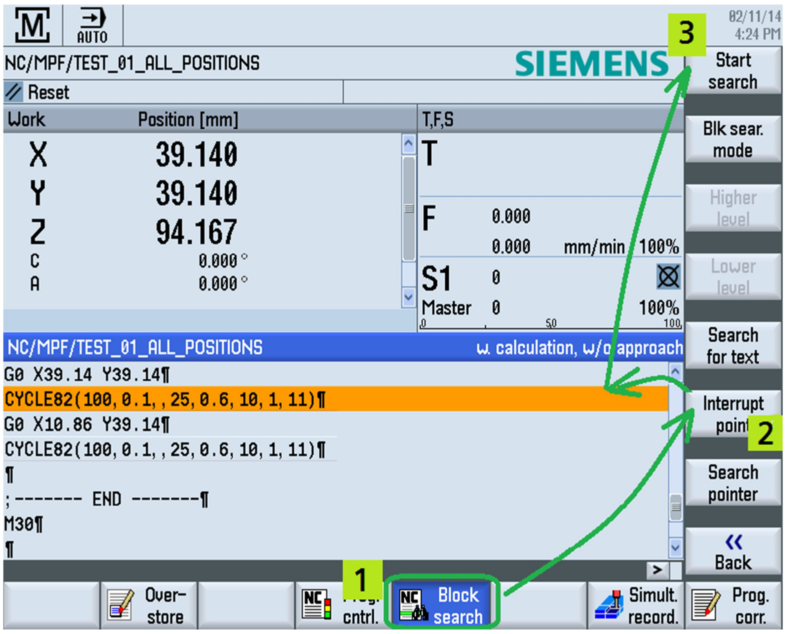
TEST_01_ALL_POSITIONS.MPF

```
            ;------- INITIAL DECLARATIONS, TOOL SELECTION -------
            T="CENTERDRILL"
            D1
            M6
            S10 M3 F1000
            X0 Y0 Z150 G0

            ;------- CIRCLE OF HOLES (PROGRAM OF EVERY HOLE POSITION): ------
Position 1  G0 X10.86 Y10.86
  Drilling  CYCLE82(100,0,1,,25,0.6,10,1,11)
Position 2  G0 X39.14 Y10.86
  Drilling  CYCLE82(100,0,1,,25,0.6,10,1,11)
       ...  G0 X39.14 Y39.14
            CYCLE82(100,0,1,,25,0.6,10,1,11)
            G0 X10.86 Y39.14
            CYCLE82(100,0,1,,25,0.6,10,1,11)

            ;------- END -------
            M30
```

Example: interruption (reset) at this block

In this situation, Block Search finds the block, which corresponds exactly to the last drilled hole position (where the program reset happened).

Figure 2-1 Use of Block Search in Case n.1: starting Block Search



NC/MPF/TEST_01_ALL_POSITIONS

Reset

Work	Position [mm]
X	39.140
Y	39.140
Z	94.167
C	0.000 °
A	0.000 °

T,F,S

T

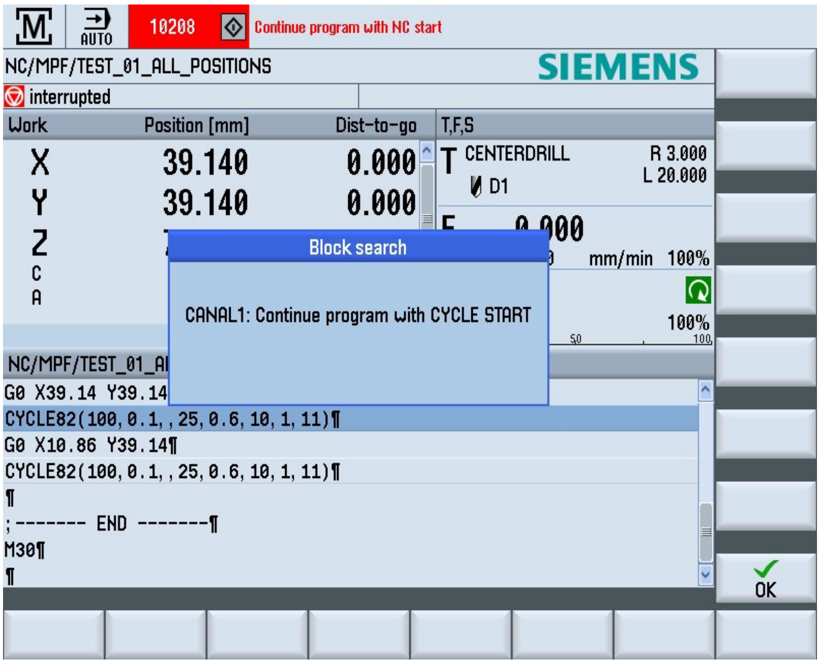
F 0.000
0.000 mm/min 100%

S1 0
Master 0 100%

NC/MPF/TEST_01_ALL_POSITIONS w. calculation, w/o approach

```
G0 X39.14 Y39.14¶
CYCLE82(100,0.1,,25,0.6,10,1,11)¶
G0 X10.86 Y39.14¶
CYCLE82(100,0.1,,25,0.6,10,1,11)¶
¶
;------- END -------¶
M30¶
¶
```

Figure 2-2 Use of Block Search in Case n.1: resume with Cycle Start



10208 Continue program with NC start

NC/MPF/TEST_01_ALL_POSITIONS

interrupted

Work	Position [mm]	Dist-to-go
X	39.140	0.000
Y	39.140	0.000
Z		
C		
A		

T,F,S

T CENTERDRILL R 3.000
D1 L 20.000

F 0.000
mm/min 100%
100%

Block search

CANAL1: Continue program with CYCLE START

NC/MPF/TEST_01_AI

```
G0 X39.14 Y39.14
CYCLE82(100,0.1,,25,0.6,10,1,11)¶
G0 X10.86 Y39.14¶
CYCLE82(100,0.1,,25,0.6,10,1,11)¶
¶
;------- END -------¶
M30¶
¶
```

2.2 Case n.2: holes positions programmed with DIN cycle

In order to describe the holes pattern, the cycle *HOLES2()* is called in the following DIN-ISO program, after a modal call of the drill instructions *MCALL CYCLE82()*.

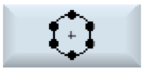
TEST_02_HOLES2.MPF

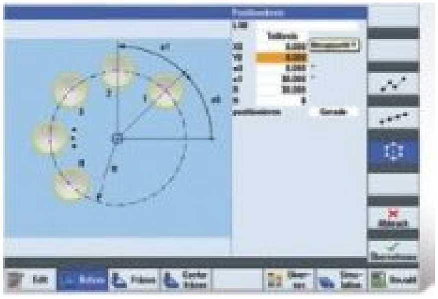
The user variable NUM_TOT_HOLES has been defined in MGUD.

```
;------- INITIAL DECLARATIONS, TOOL SELECTION -------
NUM_TOT_HOLES = 4

T="CENTERDRILL"
D1
M6
S10 M3 F1000
X0 Y0 Z150 G0
```

Sinumerik Operate with ProgramGUI-DE helps to find easily the drill positions.

```
;------- CIRCLE OF HOLES (USING "HOLES2"): -------
MCALL CYCLE82(100,0,1,,25,0.6,10,1,11)
```





Drilling will be executed several times, *Holes2* runs through the positions.

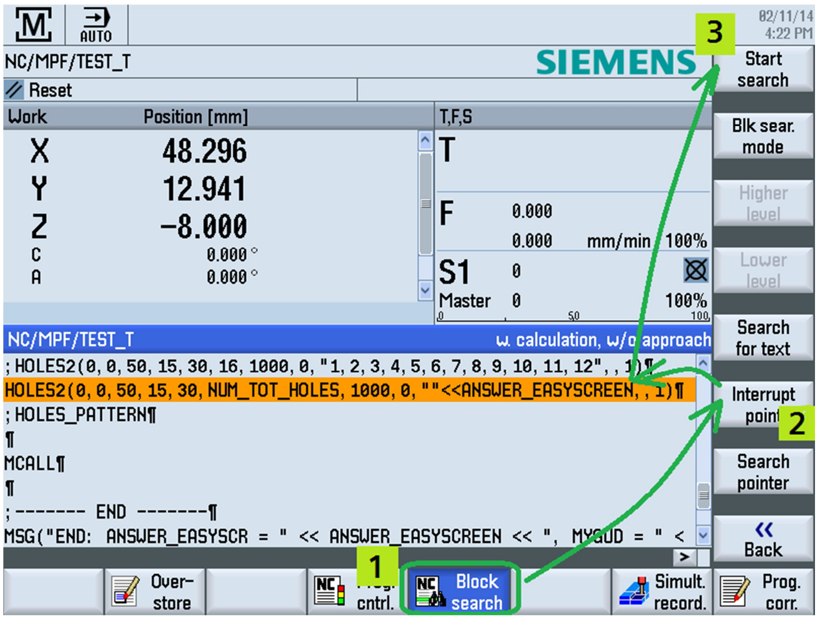
Result ->

```
HOLES2(0,0,50,15,30,NUM_TOT_HOLES,1000,0,"",,1)

MCALL

;------- END -------
M30
```

Example: interruption (reset) at this block

However, with the part program in *Case n.2*, the Block Search will find the block which includes all the hole positions, so it is not automatic the finding of the last drilled position.

Figure 2-3 Use of Block Search in Case n.2



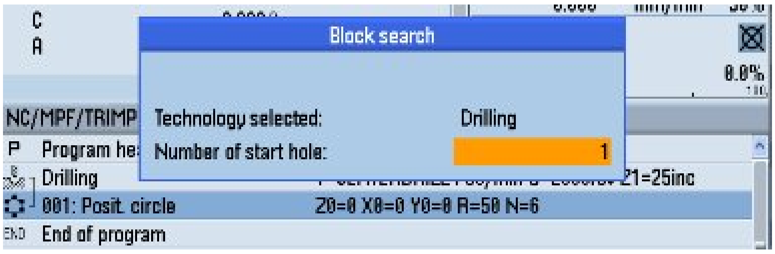
```
; HOLES2(0, 0, 50, 15, 30, 16, 1000, 0, "1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12", , 1)¶
HOLES2(0, 0, 50, 15, 30, NUM_TOT_HOLES, 1000, 0, ""<<ANSWER_EASYSCREEN, , 1)¶
; HOLES_PATTERN¶
¶
MCALL¶
¶
; ------- END -------¶
MSG("END:  ANSWER_EASYSCR = " << ANSWER_EASYSCREEN << ", MYGUD = " <
```

In the next chapter, an example of a customized application for the *Case n.2* will be described.

2.3 Case n.3: holes positions programmed with Shopmill

Situation n.3. is very similar to Situation n.2: moreover, SINUMERIK Operate will ask automatically to the operator from which hole he would like to resume from.

Figure 2-4 Use of Block Search in Case n.3



Block search

Technology selected: Drilling

Number of start hole: 1

3 Solution for Case n.2

OVERVIEW OF THE APPLICATION EXAMPLE:

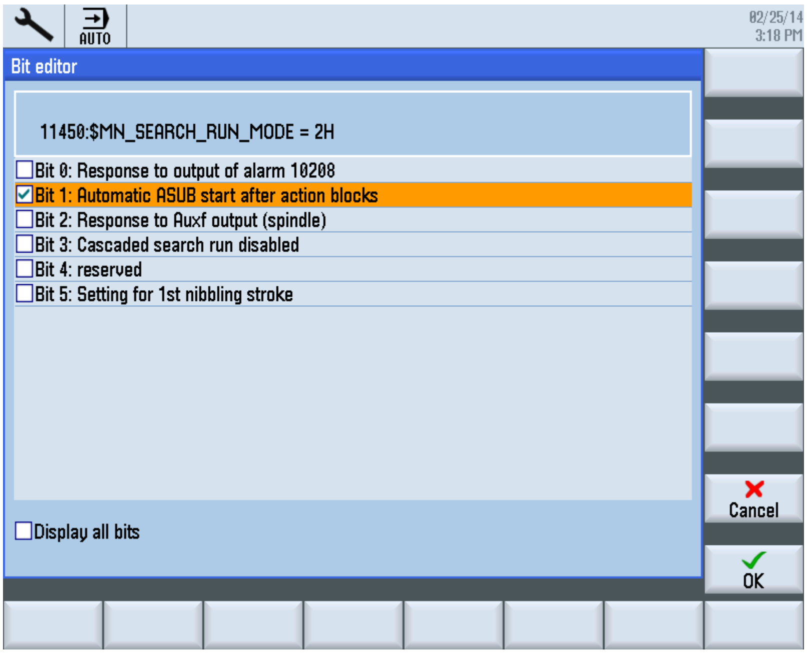
1. 3.1 Automatic call of a sub-routine after Block Search (to show a screen)
2. 3.2 Example of a customized screen for Case n.2
3. 3.3 Adaptation of the DIN-ISO program for Case n.2

3.1 Automatic call of a sub-routine after Block Search (to show a screen)

With *SINUMERIK*, a special sub-program named "PROG_EVENT" can be called upon the occurrence of specific events, e.g. after a Block Search. Here is shown only how to enable it after a Block Search; for further settings, e.g. behavior with single block selected (MD 20106), please refer to the documentation (4, \6\).

1 – Modify MD 11450 as in the figure below:

Figure 3-1 MD 11450 SEARCH_RUN_MODE bit 2: enable PROG_EVENT after Block Search



Bit editor

11450:$MN_SEARCH_RUN_MODE = 2H

□ Bit 0: Response to output of alarm 10208
☑ Bit 1: Automatic ASUB start after action blocks
□ Bit 2: Response to Auxf output (spindle)
□ Bit 3: Cascaded search run disabled
□ Bit 4: reserved
□ Bit 5: Setting for 1st nibbling stroke

□ Display all bits

NOTE

The *PROG_EVENT* and the Asynchronous Subroutines (ASUB) require the runtime license *Multiple Mode Actions*.

“PROG_EVENT” (**PE**) can be customized by the machine manufacturer: to do this, among other possibilities, a special subprogram can be created in the NC folder “Manufacturer Cycles” (CMA.DIR) and named “CYC**PE**_MA.SPF”.

In our example, the “MMC()” command will be used in “CYCPE_MA.SPF” to show the prompt screen, as shown below.

2 – Create your customized CYCPE_MA.SPF, similar to the text below:

CMA.DIR / CYCPE_MA.SPF

```
;...
;...
```

Identify condition of PROG_EV.: Block Search and Boring.

```
; ---------- TEST, TO IDENTIFY A BLOCK SEARCH ---------
IF(($P_PROG_EVENT==5) AND (JOB_BOR==TRUE))  ;5= AFTER A BLOCK S.
; more customization: IF($P_SEARCHL>0) ;4=WITH CALCUL,WITHOUT AP.
```

Show the user screen

```
MMC("CYCLES,PICTURE_ON,mein_mmc2.com,MaskMMC","A")
```

Wait for user confirmation

```
M0
STOPRE    ;IMPORTANT,OTHERWISE THE CHANGES IN EASYSCR.DONT WORK!
```

Hide the screen

```
MMC("CYCLES,PICTURE_OFF", "N")
; -> HERE THE GUD ANSWER_EASYSCREEN HAS BEEN MODIFIED IN EASYSCR.
```

Message for debug:

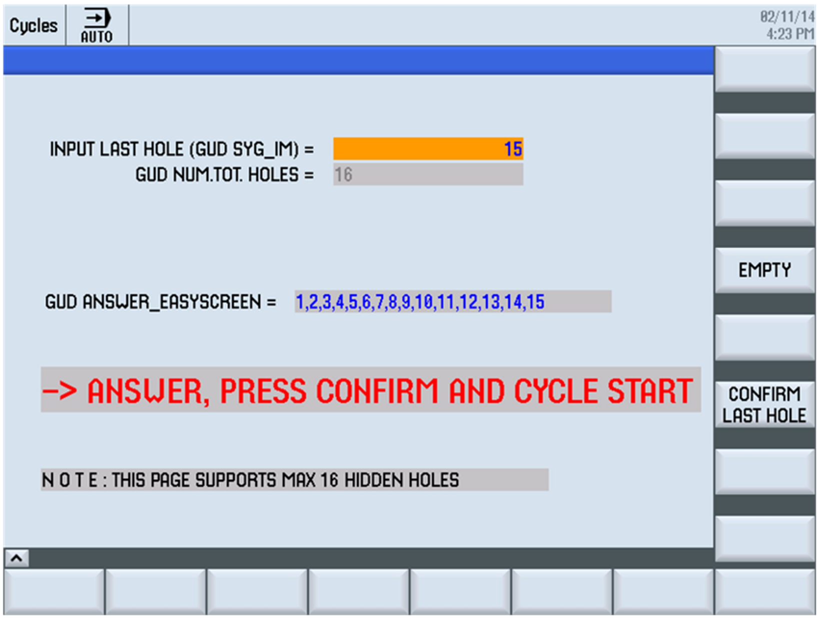
```
MSG("PROG_EVENT CYCPE1_MA, AFTER BLOCK SEARCH. ANSWER_EASYSCR = "
<< ANSWER_EASYSCREEN)
G4F2
MSG()

ENDIF
; ---------- END OF TEST ---------
;...
;...

M17
```

3.2 Example of a customized screen for Case n.2

In the *Cases n.1* and *n.3*, the hole drilled as last (before the program interruption) can be easily determined, as explained in the previous chapter.

With the part-program of *Case n.2*, the machine manufacturer can implement a customized application, in order to let the user choose the last drilled position before resuming the machining. An example of application is described here, showing a screen after a Block Search to the user, which will ask the next hole number.

Figure 3-2 Customized prompt screen, shown after Block Search before resuming the machining



INPUT LAST HOLE (GUD SYG_IM) = 15

GUD NUM.TOT. HOLES = 16

GUD ANSWER_EASYSCREEN = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15

-> ANSWER, PRESS CONFIRM AND CYCLE START

N O T E : THIS PAGE SUPPORTS MAX 16 HIDDEN HOLES

The prompt screen in this example is implemented with *Easyscreen* (*Run MyScreens*) (file: *mein_mmc2.com*). Basically it executes the following actions:

- asks the user to write the hole number to start from (example: 15)
- checks if the input number is >0 and < than total number of holes in the pattern
- writes the GUD "ANSWER_EASYSCREEN", which will be used in the part-program as an appropriate argument for the cycle HOLES2().

The settings to enable the MMC commands *PICTURE_ON* and *PICTURE_OFF* are already done by default in the original hmi file *systemconfiguration.ini*. For more information, please refer to the documentation (4, \5\).

NOTE Please, pay attention to the lower-case letter in the file names. Linux-based system, like the NCU system, distinguish lower-case from capital letters.

3 – Create your customized Easyscreen file (e.g. *mein_mmc2.com*), similar to the text below.

The easyscreen.ini will be skipped, because a softkey is not needed to show the screen in Figure 3-2 (the MMC command is used).

In this paragraph, an example of source code for the prompt screen is shown. The main file is:

[*/card* or */hmisl*] */oem/sinumerik/hmi/proj/mein_mmc2.com*.

FILE: mein_mmc2.com (Screen with the question for the user)

```
//M(MaskMMC)

DEF GUDMMC0=(IDB///"","INPUT LAST HOLE =
","",""/WR2,AL1///0,50,250/260,50,150/6,4)
DEF N_TOT_HOLES=(S///"","GUD NUM.TOT. HOLES =
","",""/WR1,AL1//"GUD/NUM_TOT_HOLES"/0,70,250/260,70,150/5,4)
DEF DESCR1=(S//"HOLES TO HIDE: 1, 2, .. :
"/"","","",""/WR1,AL1///0,150,10/20,150,380/6,4)
DEF GUD_ANSWER=(S///"","GUD ANSWER_EASYSCREEN =
","",""/WR1,AL1//"GUD/ANSWER_EASYSCREEN"/20,170,200/230,170,250/6,4)
DEF DESCR2=(S//"-> ANSWER, PRESS CONFIRM AND CYCLE
START"/"","","",""/WR1,FS2///0,230,10/30,230,520/7,4)
DEF DESCR3=(S//"N O T E : THIS PAGE SUPPORTS MAX 16 HIDDEN
HOLES"/"","","",""/WR1,AL1///0,310,10/30,310,400/0,4)


HS1=("")
HS2=("")
HS3=("")
HS4=("")
HS5=("")
HS6=("")
HS7=("")
HS8=("")
VS1=("")
VS2=("")
VS3=("")
VS4=("EMPTY")
VS5=("")
VS6=("CONFIRM LAST HOLE")
VS7=("")
VS8=("")


PRESS(RECALL)
LM("MaskMMC")
END_PRESS


PRESS(VS4)
GUD_ANSWER.VAL = ""
END_PRESS
```

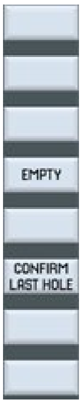
NOTES:

Beginning of mask (screen)

Text and I/O fields on the screen

(the variables NUM_TOT_ HOLES and ANSWER_ EASYSCREEN are defined in MGUDs).

Define softkeys horiz.- vertic.:



```
PRESS(VS6)
;------- BUILD STRING GUD_ANSWER (GUD ANSWER_EASYSCREEN) -------
GUD_ANSWER = ""
IF (GUDMMC0>0)
  IF (GUDMMC0>N_TOT_HOLES.VAL-1)
    GUDMMC0 = N_TOT_HOLES.VAL -1  ;-1 :NUM.HOLES TO DO MUST BE >0
  ENDIF
  IF (GUDMMC0<0)
    GUDMMC0 = 0
  ENDIF
  IF (GUDMMC0 == 0)
    GUD_ANSWER.VAL = ""
  ENDIF
  IF (GUDMMC0 == 1)
    GUD_ANSWER.VAL = "1"
  ENDIF
  IF (GUDMMC0 == 2)
    GUD_ANSWER.VAL = "1,2"
  ENDIF
  ;...
  ;...
  ;...
  IF (GUDMMC0 == 15)
    GUD_ANSWER.VAL = "1,2,3,4,5,6,7,8,9,10,11,12,13,14,15"
  ENDIF
  IF (GUDMMC0 == 16)
    GUD_ANSWER.VAL = "1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16"
  ENDIF
  WNP("GUD/ANSWER_EASYSCREEN",GUD_ANSWER.VAL)
ELSE
  GUD_ANSWER = ""
ENDIF
END_PRESS


LOAD
GUDMMC0.MAX = -1 + RNP("GUD/NUM_TOT_HOLES")
GUDMMC0.MIN = 0
FOC="GUDMMC0"
END_LOAD


//END
```

PRESS methods

(actions on sofktey press)

On press of “Confirm” by the user, the string for the HOLES2() cycle in the part program will be builded.

Actions at the load of the screen.

NOTE The *Easyscreen* functionality is freely included in SINUMERIK Operate up to 5 screens, after which a runtime license *Run MyScreens* is required.

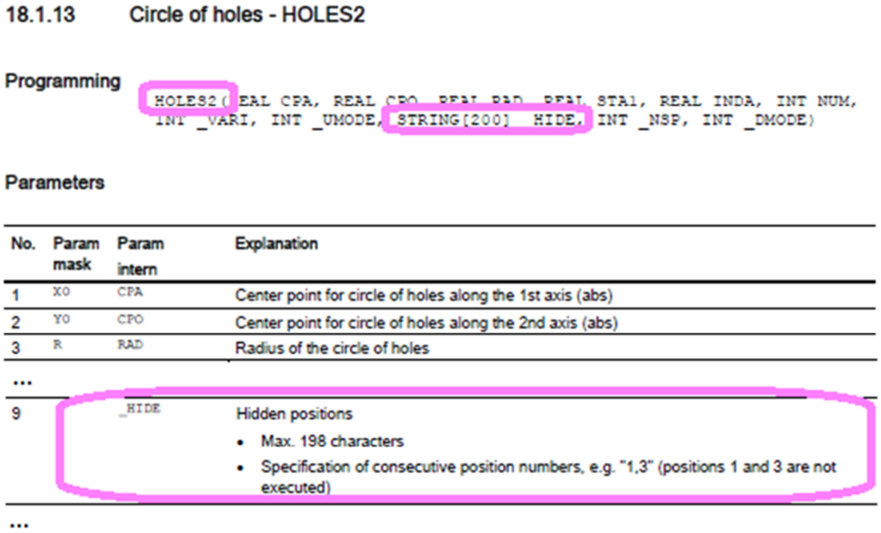
3.3 Adaptation of the DIN-ISO program for Case n.2

The user need to modify manually the DIN-ISO program, even after using the ProgramGuide mask.

This is needed, because the customized screen in Figure 3-2 writes a GUD variable (ANSWER_EASYSCREEN) and this variable must be an argument of the HOLES2() cycle.

In fact after a block search, a HOLES2() cycle would always begin from the first hole, but it provides a parameter to choose from which hole to start again (_HIDE). The _HIDE parameter is described with more details in the **Fehler! Verweisquelle konnte nicht gefunden werden.** below.

Figure 3-3 Reference of cycle HOLES2, from Programming Manual Job Planning



18.1.13 Circle of holes - HOLES2

Programming

```
HOLES2(REAL CPA, REAL CPO, REAL RAD, REAL STA1, REAL INDA, INT NUM,
INT _VARI, INT _UMODE, STRING[200] _HIDE, INT _NSP, INT _DMODE)
```

Parameters

No.	Param mask	Param intern	Explanation
1	X0	CPA	Center point for circle of holes along the 1st axis (abs)
2	Y0	CPO	Center point for circle of holes along the 2nd axis (abs)
3	R	RAD	Radius of the circle of holes
...			
9		_HIDE	Hidden positions • Max. 198 characters • Specification of consecutive position numbers, e.g. "1,3" (positions 1 and 3 are not executed)

...

The application here described uses the GUD variable ANSWER_EASYSCREEN to write the _HIDE parameter

4 – Add manually the pink words from the text below to your MPF Part-Program.

Without this manual modification, the screen will appear but the drilling process will not resume from the hole number coming after the point of interrupting.

NOTE

This GUD *ANSWER_HIDE_HOLES* must be changed after a Block Search (to provide the proper list of the holes to skip) and then reset at the end of part-program (e.g. ANSWER_HIDE_HOLES = “”). In this way, a machining started normally (not after a block search) will not be affected. An alternative can be to show the prompt screen always at the beginning of the part-program.

MAIN_PROGRAM_MODIFIED.MPF

The var.NUM _TOT_HOLE, ANSWER_EASYSCREEN, JOB_BOR are defined in MGUDs.

```
;------- INITIAL DECLARATIONS, TOOL SELECTION -------
NUM_TOT_HOLES = 16
ANSWER_EASYSCREEN = ""
JOB_BOR = TRUE

T="CENTERDRILL"
D1
M6
S10 M3 F1000
X0 Y0 Z150 G0


;------- CIRCLE OF HOLES: -------
;N O T E: HOLES2(0,0,50,15,30,16,1000,0, example:"1,2,3" ,,1)
HOLES2(0,0,50,15,30,NUM_TOT_HOLES,1000,0,ANSWER_EASYSCREEN,,1)

MCALL

;------- END -------
JOB_BOR = FALSE
M30
```

Example: interruption (reset) at this block

Then, when Block Search is executed by the user, the prompt screen will appear.

With this customized application, Block Search can be used in *Case n.2*, to resume a program where the GUD-variable “ANSWER_EASYSCREEN” is used in the _HIDE argument of the HOLES2() cycle.

4 Related literature

Table 4-1

	Topic	Title / Link
\1\	Siemens Industry Online Support	http://support.automation.siemens.com
\2\	Download page of this entry	http://support.automation.siemens.com/WW/view/en/89770537
\3\	SINUMERIK 840D sl/828D Operating Manual	http://support.automation.siemens.com/WW/view/en/64895700
\4\	SINUMERIK, Programming Manual Job Planning 03/2013	http://support.automation.siemens.com/WW/view/en/70265038
\5\	SINUMERIK Operate (IM9)	http://support.automation.siemens.com/WW/view/en/74856813
\6\	SINUMERIK Integrate Run MyScreens (BE2)	http://support.automation.siemens.com/WW/view/en/74800442
\7\	SINUMERIK Basic Functions (8.7.3 Automatic start of an ASUB after block search)	http://support.automation.siemens.com/WW/view/en/74613000

5 Contact

Siemens AG

Industry Sector
I DT MC MTS APC Tech Team
Frauenauracher Strasse 80
91056 Erlangen
Germany
mailto: MC-MTS-APC-Tech.Team.i-dt@siemens.com

6 History

Table 6-1

Version	Date	Modifications
V1.0	04/2014	First version

Version	Date	Modifications