

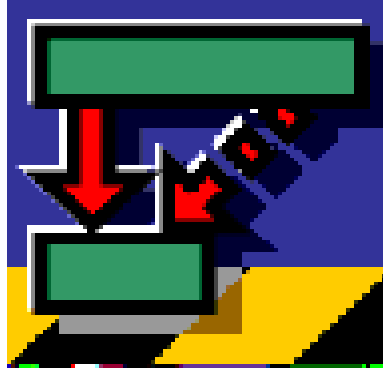
# TwinCAT

## The Windows Control and Automation Technology

TwinCAT  
Real Time System



TwinCAT  
System Manager



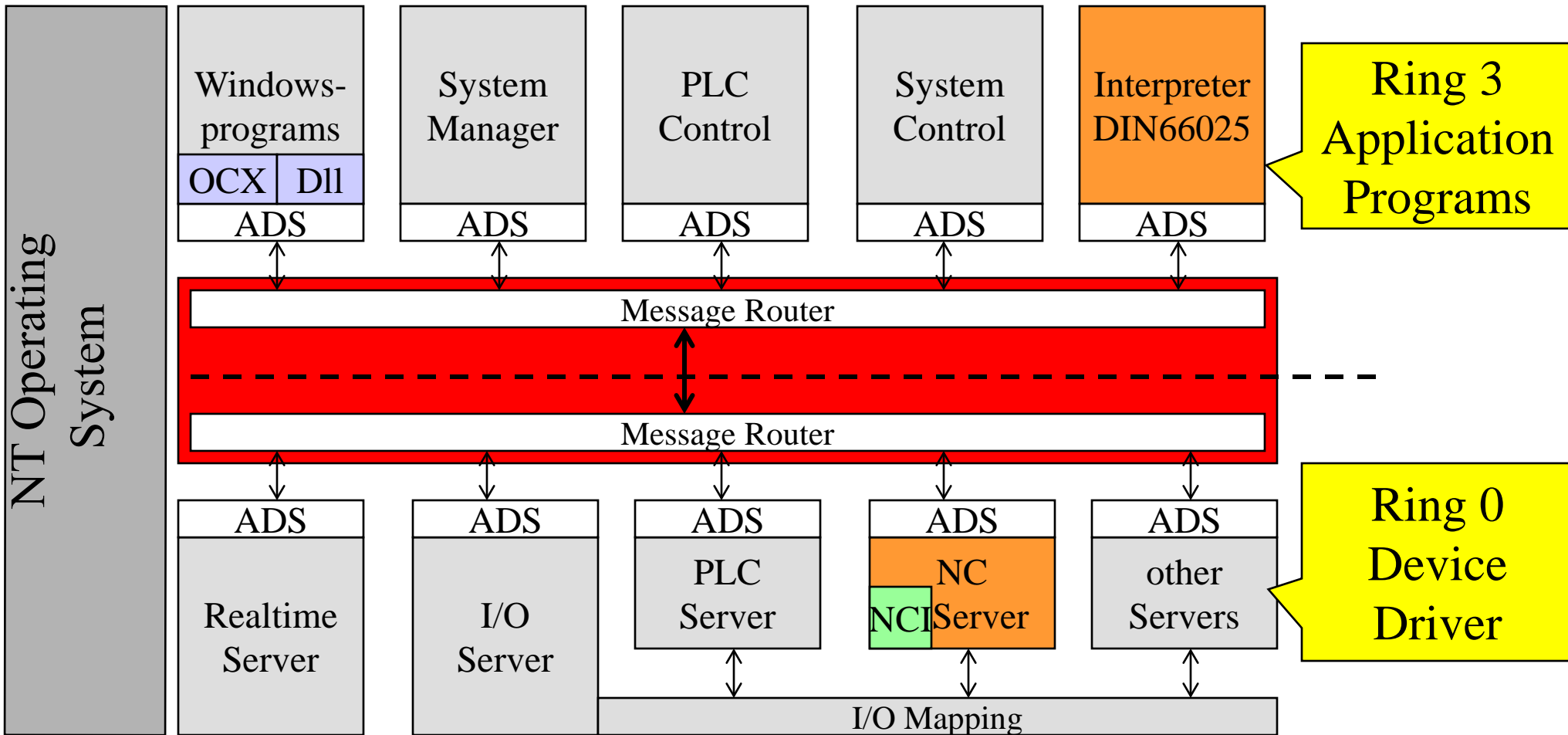
TwinCAT  
PLC



TwinCAT  
NC/NCI



# TwinCAT and Windows NT/2000



# NCI Overview 1

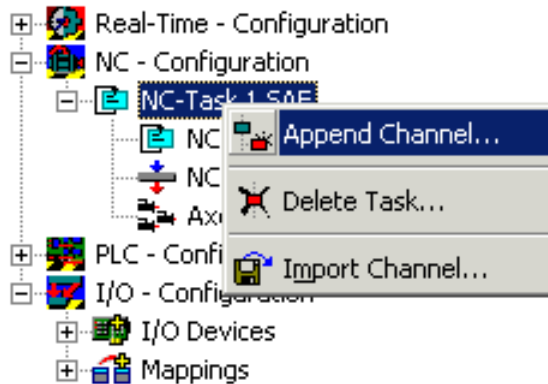
TwinCAT NCI includes

- PLC
- NC-PTP (Point to Point)
- 3D Interpolation

# NCI Overview 2

- Interfaces to run the interpolation
  - Interpreter based on DIN66025 (G-Code) or
  - PLC Interpolation Library
- Limits of the interpolation
  - 3 axes per channel (additionally Master/Slave Coupling and online reconfiguring of the axes)
  - 32 interpolation channels

# Add Interpolation Channel

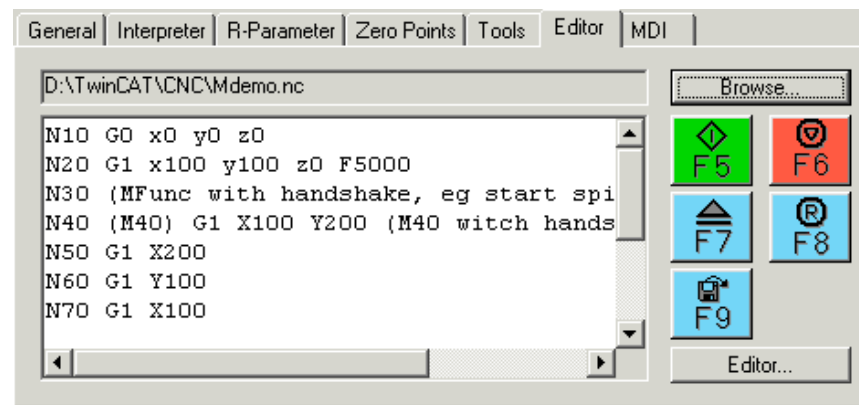


1. Select type ‚NC channel for interpolation‘
2. Save new configuration and write to the registry

# Configure Interpolation Group - Overview

1. All axes are started up in the ptp mode
2. Configure existing interpolation group (take ptp axes and put them to the interpolation group) with
  - System Manager
  - PLC (TcNcConfigLib)

# Interpreter I



# Structure of an NC Program

- Program name (optional)
- Number of blocks
- Program end

Sample:

```
% Test1 (program start)  
N10 G0 X100 Y100 Z0  
M30 (program end)
```



# Effective Duration of Words

- Modal  
commands have effects beyond the end of the block, e.g. G0, G17
- Blockwise  
commands are just valid for the actual block, e.g. G09

# Comments

- Character for a comment: curved bracket , (,
- Nested comments are not possible

*Example:*

```
N10 G0 X100 (comment)
```

# Block numbers

- Characters for a block number
  - ,:‘ main blocks
  - ,N‘ subsidiary blocks
- Block numbers are not always essential but
- Required for the target of a jump command
- Helpfull if an error occured

# Programming Movements

- Referencing (homing)

Command	G74
Cancellation	End of Block

Hint:

Referencing of the axes should be done before the interpolation group is created (PTP mode)

# Programming Movements

- Referencing (2)

*Example:*

```
N10 G74 X  
N20 G74 Y
```

# Programming Movements

- Rapid Traverse

Command	G0
Cancellation	G01-G03

- Linear interpolation with the max. allowed velocity
- Generally used for positioning the tool

# Programming Movements

- Linear Intepolation (1)

Command	G1
Cancellation	G0, G3, G03
Parameters	F - feedrate

- Tool moves with the specified velocity (F) to the target position (X, Y, Z)



# Programming Movements

- Linear Interpolation (2)

*Sample:*

```
N10 G90  
N20 G01 X100.1 Y200 F6000
```



# Programming Movements

- Circular Interpolation Clockwise
  - Radius Programming
  - Centre Point Programming

Command	G02
Cancellation	G0, G01, G03

# Programming Movements

- Radius Programming - Parameters
  - Target Position (X, Y, Z)
  - Radius (B or U)
- Hint:
  - not possible to program a full circle

*Sample:*

```
N10 G01 G17 X100 Y100 F6000  
N20 G02 X200 B200
```



# Programming Movements

- Centre Point Programming – Parameters
  - Target Position (X, Y, Z)
  - Centre Point of the Circle (I, J, K) (on default relativ to the starting point)

Sample:

```
N10 G01 G17 X100 Y100 F6000  
N20 G02 I50 J0 (J is optional) X200  
N30 M30 (program end)
```



# Programming Movements

- Circular Interpolation Counter Clockwise
  - Radius programming
  - Centre point programming

Command	G03
Cancellation	G0, G01, G02

# Programming Movements

- CIP Arc (1)
  - special circle, that is doesn't belong to a plane
  - No radius/cutter compensation supported
  - Defined with 3 points on the curve
  - Not possible to program a full circle

Command	CIP
Cancellation	End of block

# Programming Movements

- CIP Arc (2)

*Sample:*

```
N10 G01 X100 Y100 F6000  
N20 CIP X200 Y200 I50 J50 K50
```

# Programming Movements

- Helix (1)
  - circular motion with a superimposed perpendicular linear movement
  - No cutter/radius compensation supported

Command	G02 / G03
Cancellation	G0, G1

# Programming Movements

- Helix (2)

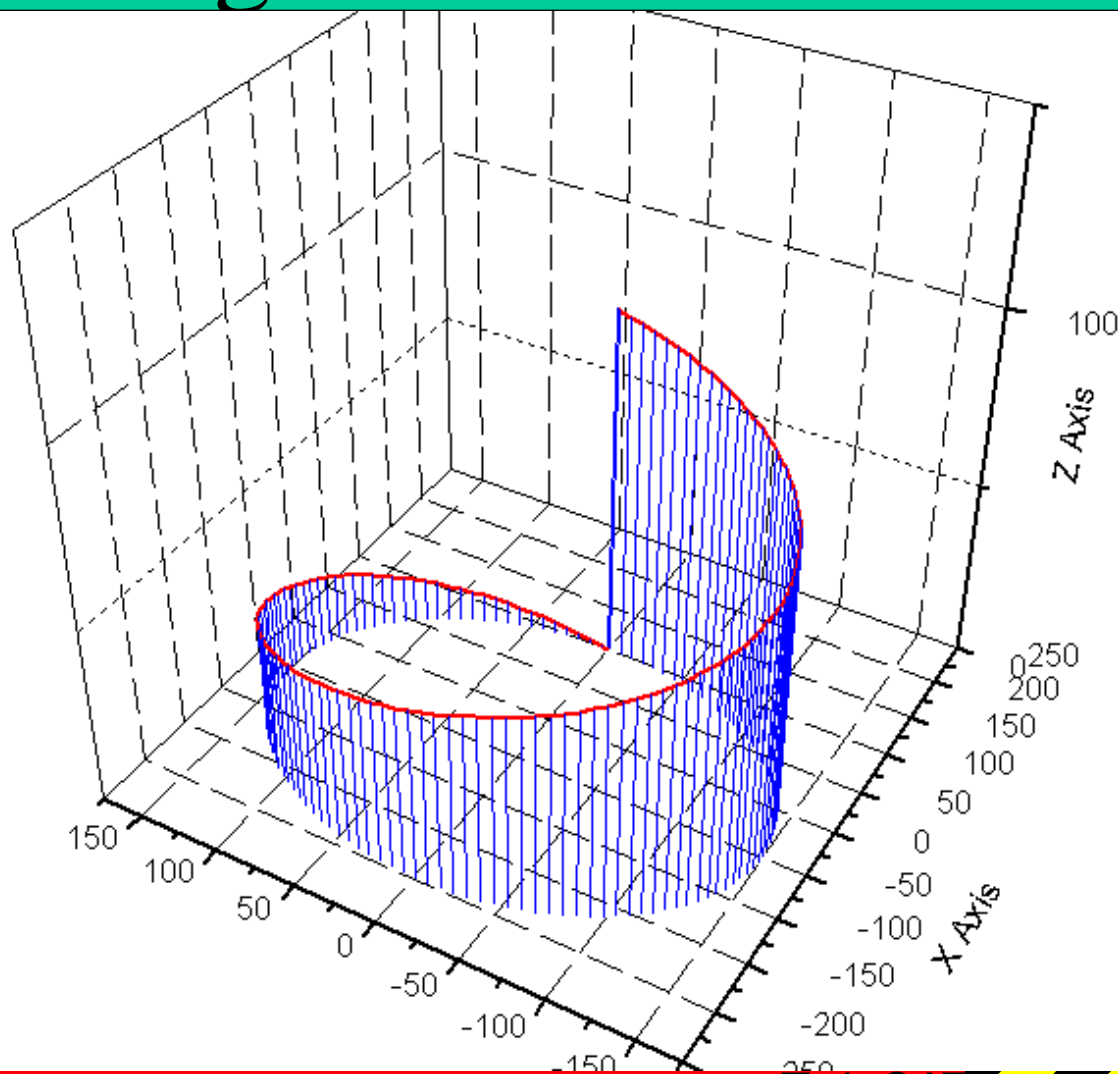
*Sample:*

```
N10 G01 G17 X100 Y0 Z0 F6000  
N20 G03 I-50 Z100  
M30
```



# Programming Movements

- Helix (3)



# 1. Exercise

- To do
  - Create new System Manager configuration with interpolation channel
  - Create ScopeView with all relevant variables
  - Write small sample programs
  - Configure interpolation group with the System Manager

# PLC NCI Libraries

- TcNcCfg.lib  
Library to build interpolation group etc
- TcNciItp.lib  
Library to manage the interpreter

# PLC NCI Libraries

- Cyclic Interfaces NciChannelFromPlc (Type):

TYPE NciChannelFromPlc :

STRUCT

nSkipLine	:	WORD;	(* Mask to skip lines *)
nItpMode	:	WORD;	
nMFuncGranted	:	WORD;	(* granted signal of the M-function *)
nReserved1	:	ARRAY[7..14] OF BYTE;	
nChaAxesOvr	:	DWORD;	(* Channel override in percent * 100 *)
nChaSpindleOvr	:	DWORD;	
nReserved2	:	ARRAY[23..150] OF BYTE;	

END\_STRUCT

END\_TYPE

# PLC NCI Libraries

- Cyclic Interfaces NciChannelFromPlc : (online )

▣ PLC\_TO\_Group(%QB1500)

.....nSkipLine = 0

.....nItpMode = 0

.....nMFuncGranted = 0

Quitt M-Function (1)

⊕ .....nReserved1

.....nChaAxesOvr = 0

Channel Override

.....nChaSpindleOvr = 0

⊕ .....nReserved2

(temp. not used)

# PLC NCI Libraries

- Cyclic Interfaces NciChannelToPlc (Type):

```
TYPE NciChannelToPlc :
```

```
STRUCT
```

```
    nJobNo          :DWORD;  
    nFastMFuncMask :ARRAY[1..5] OF DWORD;    (* Mask to evaluate fast M-functions *)  
    nHskMFuncNo    :WORD;                    (* evaluate M-function with handshake *)  
    nHskMFuncReq   :WORD;  
    nHFuncValue    :DINT;  
    nSpindleRpm    :WORD;  
    nTool          :WORD;  
    nReserved1     :ARRAY[37..132] OF BYTE;  
    nLoadedProg    :DWORD;                    (* loaded program number if exist *)  
    nItpMode       :WORD;                      (* Interpreter mode *)  
    nItpState      :WORD;                      (* Interpreter status *)  
    nItpErrCode    :WORD;                      (* Interpreter-Channel Error Code *)  
    nReserved2     :ARRAY[143..150] OF BYTE;
```

```
END_STRUCT
```

```
END_TYPE
```



# PLC NCI Libraries

- Cyclic Interfaces NciChannelToPlc : (online )

```

Group_TO_PLC(%IB1500)
├── nJobNo = 0
├── nFastMFuncMask
│   ├── nFastMFuncMask[1] = 0
│   ├── nFastMFuncMask[2] = 0
│   ├── nFastMFuncMask[3] = 0
│   ├── nFastMFuncMask[4] = 0
│   └── nFastMFuncMask[5] = 0
├── nHskMFuncNo = 0
├── nHskMFuncReq = 0
├── nHFuncValue = 0
├── nSpindleRpm = 0
├── nTool = 0
├── nReserved1
├── nLoadedProg = 0
├── nItpMode = 0
├── nItpState = 0
├── nItpErrCode = 0
├── nReserved2
└── Global_Error = 0
  
```

Dwords for Fast M-functions

Requested „M“ Nr

Signal : M Channel  
request's M-function

Additional Info:

Value of „H“ Parameter

# PLC NCI Libraries

- Cyclic Interfaces NciChannelToPlc : (online )

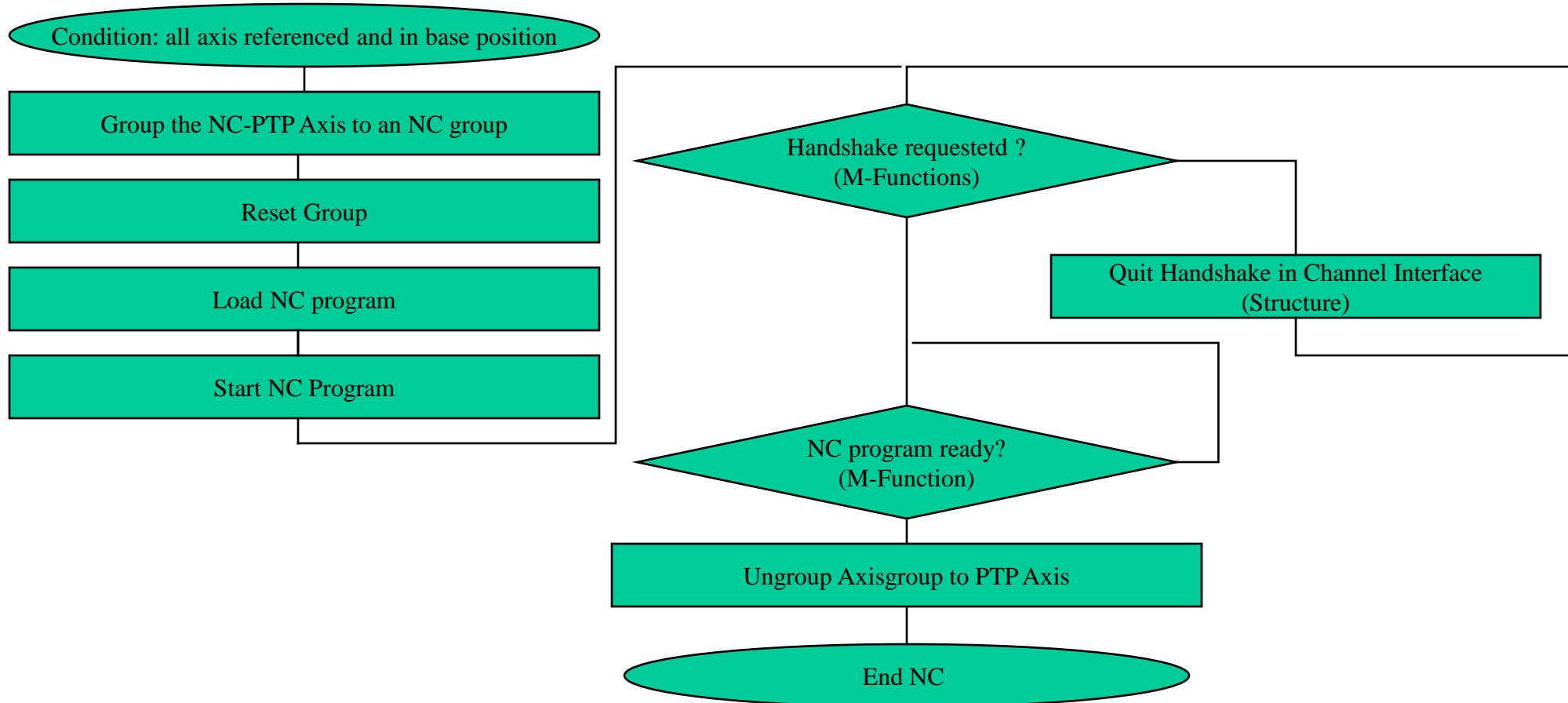
```
Group_TO_PLC(%IB1500)
├── nJobNo = 0
├── nFastMFuncMask
│   ├── nFastMFuncMask[1] = 0
│   ├── nFastMFuncMask[2] = 0
│   ├── nFastMFuncMask[3] = 0
│   ├── nFastMFuncMask[4] = 0
│   └── nFastMFuncMask[5] = 0
├── nHskMFuncNo = 0
├── nHskMFuncReq = 0
├── nHFuncValue = 0
├── nSpindleRpm = 0
├── nTool = 0
├── nReserved1
│   ├── nLoadedProg = 0
│   ├── nntpMode = 0
│   ├── nntpState = 0
│   └── nntpErrCode = 0
├── nReserved2
└── Global_Error = 0
```

State of Interpreter



# PLC NC Handshake

- Steps to load and start and handshake an NC program



# PLC NC Handshake

- FB's for this steps

Condition: all axis referenced and in base position

Group the NC-PTP Axis to an NC group

Reset Group

Load NC program

Start NC Program

Ungroup Axisgroup to PTP Axis

ItpReset

— bExecute bBusy —  
 — nChnld bErr —  
 — tTimeOut nErrld —

CfgBuild3DGroup

— bExecute bBusy —  
 — nGroupld bErr —  
 — nXAxisld nErrld —  
 — nYAxisld —  
 — nZAxisld —  
 — tTimeOut —

ItpLoadProg

— bExecute bBusy —  
 — nChnld bErr —  
 — sPrg nErrld —  
 — nLength —  
 — tTimeOut —

ItpStartStop

— bStart bBusy —  
 — bStop bErr —  
 — nChnld nErrld —  
 — tTimeOut —

CfgReconfigGroup

— bExecute bBusy —  
 — nGroupld bErr —  
 — tTimeOut nErrld —

# PLC NC Handshake

- In which library to find this FB's ?

<p>TcNcCfg.Lib</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p style="text-align: center;">CfgBuild3DGroup</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">— bExecute</td> <td style="width: 50%;">bBusy</td> </tr> <tr> <td>— nGroupId</td> <td>bErr</td> </tr> <tr> <td>— nXAxisId</td> <td>nErrId</td> </tr> <tr> <td>— nYAxisId</td> <td></td> </tr> <tr> <td>— nZAxisId</td> <td></td> </tr> <tr> <td>— tTimeOut</td> <td></td> </tr> </table> </div>	— bExecute	bBusy	— nGroupId	bErr	— nXAxisId	nErrId	— nYAxisId		— nZAxisId		— tTimeOut		<p>TcNciIPT.lib</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p style="text-align: center;">ItpLoadProg</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">— bExecute</td> <td style="width: 50%;">bBusy</td> </tr> <tr> <td>— nChnId</td> <td>bErr</td> </tr> <tr> <td>— sPrg</td> <td>nErrId</td> </tr> <tr> <td>— nLength</td> <td></td> </tr> <tr> <td>— tTimeOut</td> <td></td> </tr> </table> </div>	— bExecute	bBusy	— nChnId	bErr	— sPrg	nErrId	— nLength		— tTimeOut		<p>TcNcCfg.Lib</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p style="text-align: center;">CfgReconfigGroup</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">— bExecute</td> <td style="width: 50%;">bBusy</td> </tr> <tr> <td>— nGroupId</td> <td>bErr</td> </tr> <tr> <td>— tTimeOut</td> <td>nErrId</td> </tr> </table> </div>	— bExecute	bBusy	— nGroupId	bErr	— tTimeOut	nErrId
— bExecute	bBusy																																
— nGroupId	bErr																																
— nXAxisId	nErrId																																
— nYAxisId																																	
— nZAxisId																																	
— tTimeOut																																	
— bExecute	bBusy																																
— nChnId	bErr																																
— sPrg	nErrId																																
— nLength																																	
— tTimeOut																																	
— bExecute	bBusy																																
— nGroupId	bErr																																
— tTimeOut	nErrId																																
<p>TcNciIPT.lib</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p style="text-align: center;">ItpReset</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">— bExecute</td> <td style="width: 50%;">bBusy</td> </tr> <tr> <td>— nChnId</td> <td>bErr</td> </tr> <tr> <td>— tTimeOut</td> <td>nErrId</td> </tr> </table> </div>	— bExecute	bBusy	— nChnId	bErr	— tTimeOut	nErrId	<p>TcNciIPT.lib</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p style="text-align: center;">ItpStartStop</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">— bStart</td> <td style="width: 50%;">bBusy</td> </tr> <tr> <td>— bStop</td> <td>bErr</td> </tr> <tr> <td>— nChnId</td> <td>nErrId</td> </tr> <tr> <td>— tTimeOut</td> <td></td> </tr> </table> </div>	— bStart	bBusy	— bStop	bErr	— nChnId	nErrId	— tTimeOut																	
— bExecute	bBusy																																
— nChnId	bErr																																
— tTimeOut	nErrId																																
— bStart	bBusy																																
— bStop	bErr																																
— nChnId	nErrId																																
— tTimeOut																																	

# Mdefs file for M-functions

- Description for Interpreter how to handle M-functions
- File must be under “TwinCAT/CNC”
- Changes are valid after TwinCAT restart

# Mdefs file for M-functions

- Construction of an M-defs file

```

//-----
// Beispiel für eine Datei zur Festlegung
// der M-Funktionsregeln.
//-----
//
// M-Nummer      00 ... 159
//
// Regel-Bits   01h =  1 = Handshake before Move
//              02h =  2 = Set FastBit before Move
//              04h =  4 = Handshake after Move
//              08h =  8 = Set FastBit after Move
//              10h = 16 = Reset FastBit before Move
//              20h = 32 = Reset FastBit after Move
//              40h = 64 = donot use
//              80h = 128 = AutoReset FastBit at Line End
//
// Abgeloescht[10]
//
//-----
30 140 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
60 130 64 65 66 67 68 69 70 71 -1 -1
64  2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
65  2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
66  2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
67  2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
68  2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
69  2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
70  2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
71  2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

```

