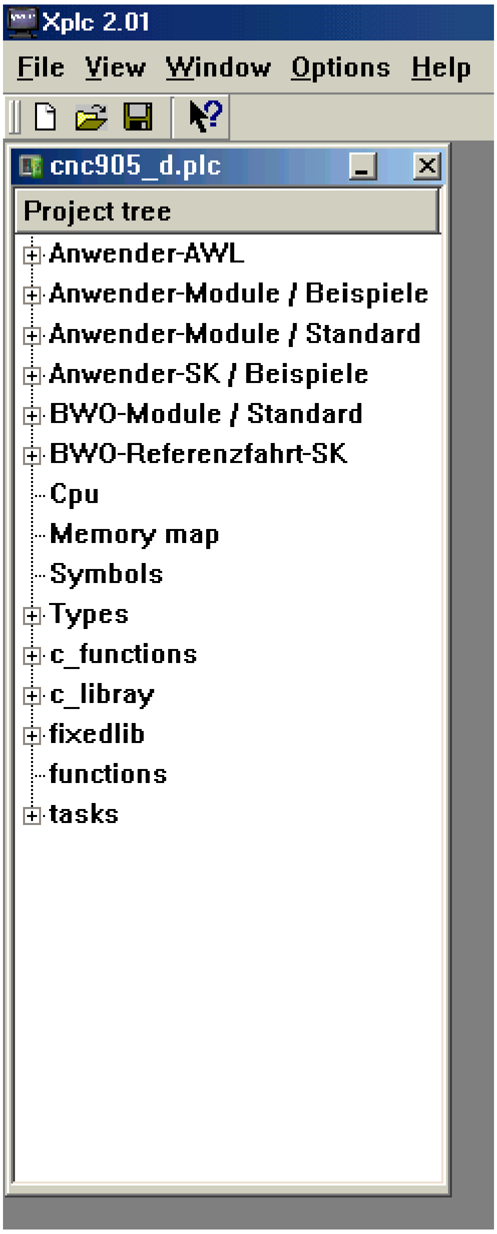
5. Handling XPLC

5.1 Main menue



Functions / Modules / Blocks

CPU monitor — Status display, Debugger

Memory Map — Online Status display, Inputs / Outputs, Flags / Registers

Symbol editor

Types definitions

Library of the standard function blocks

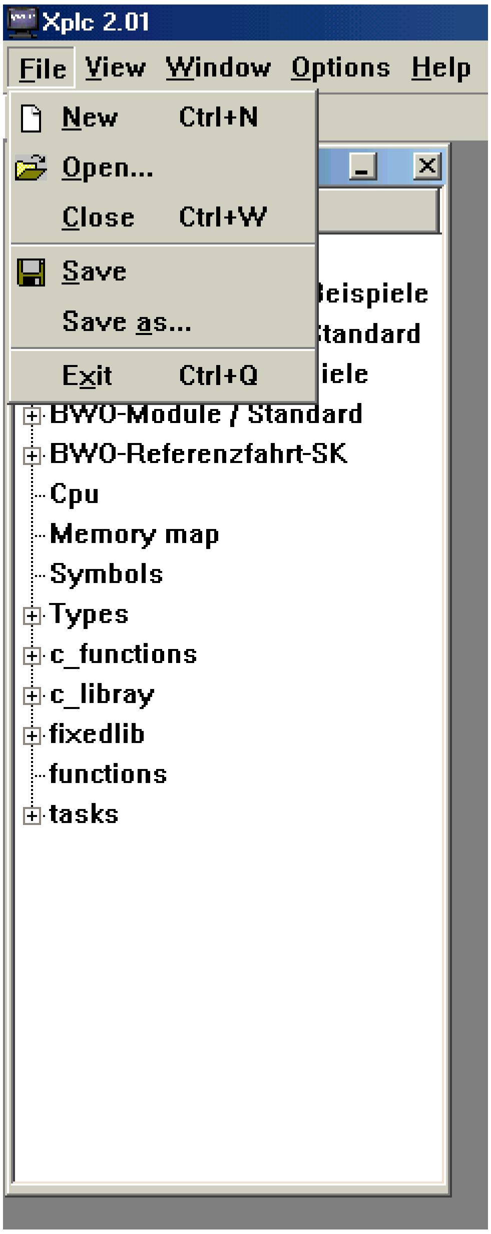
Library of further standard function blocks

Library of mathematically function blocks

Tasks — Background task, Start task

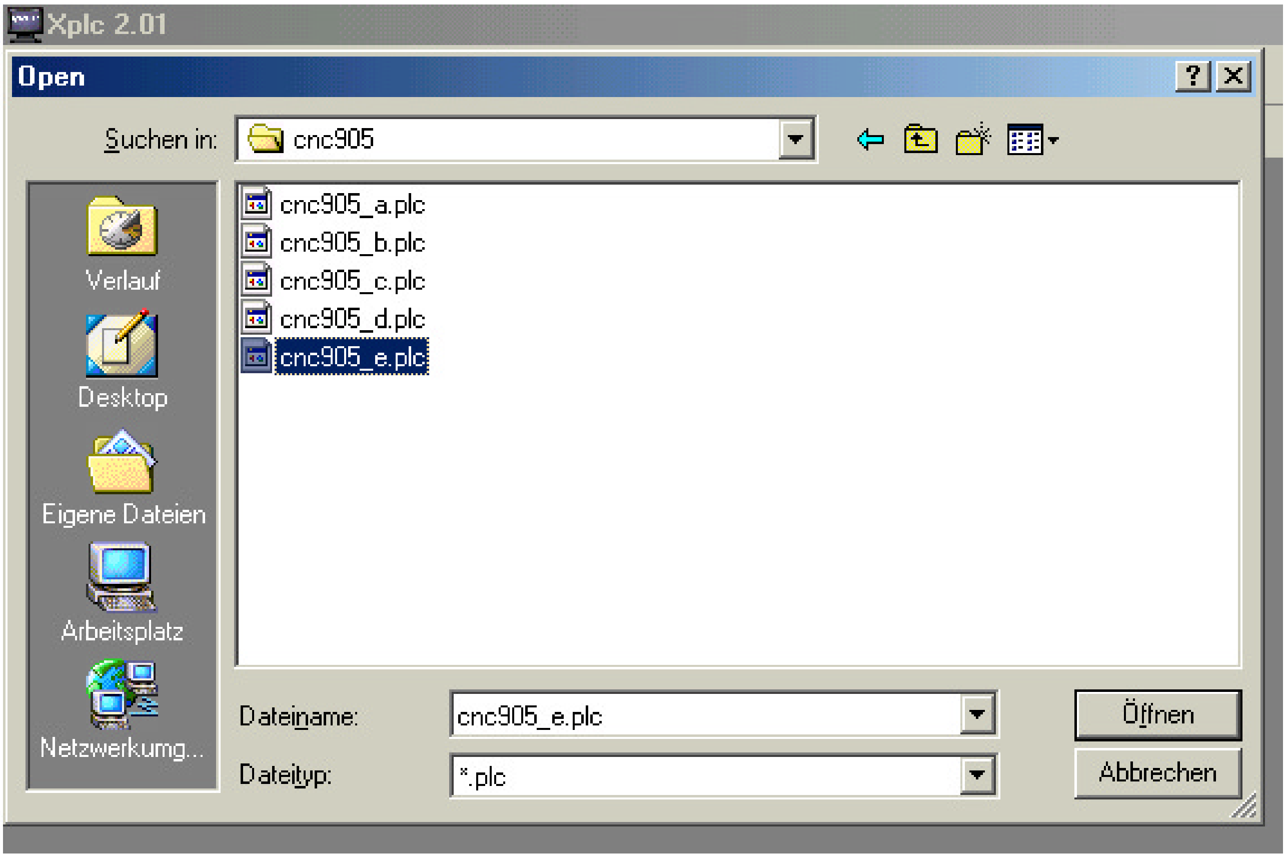
5.2 Menue File

New	create a new XPLC project
Open	open an available XPLC project
Close	clos XPLC project
Save	save a XPLC project
Save as	Save as a XPLC project
Exit	Exit handling



5.2.1 Open

Open an available XPLC project



Suchen in: cnc905

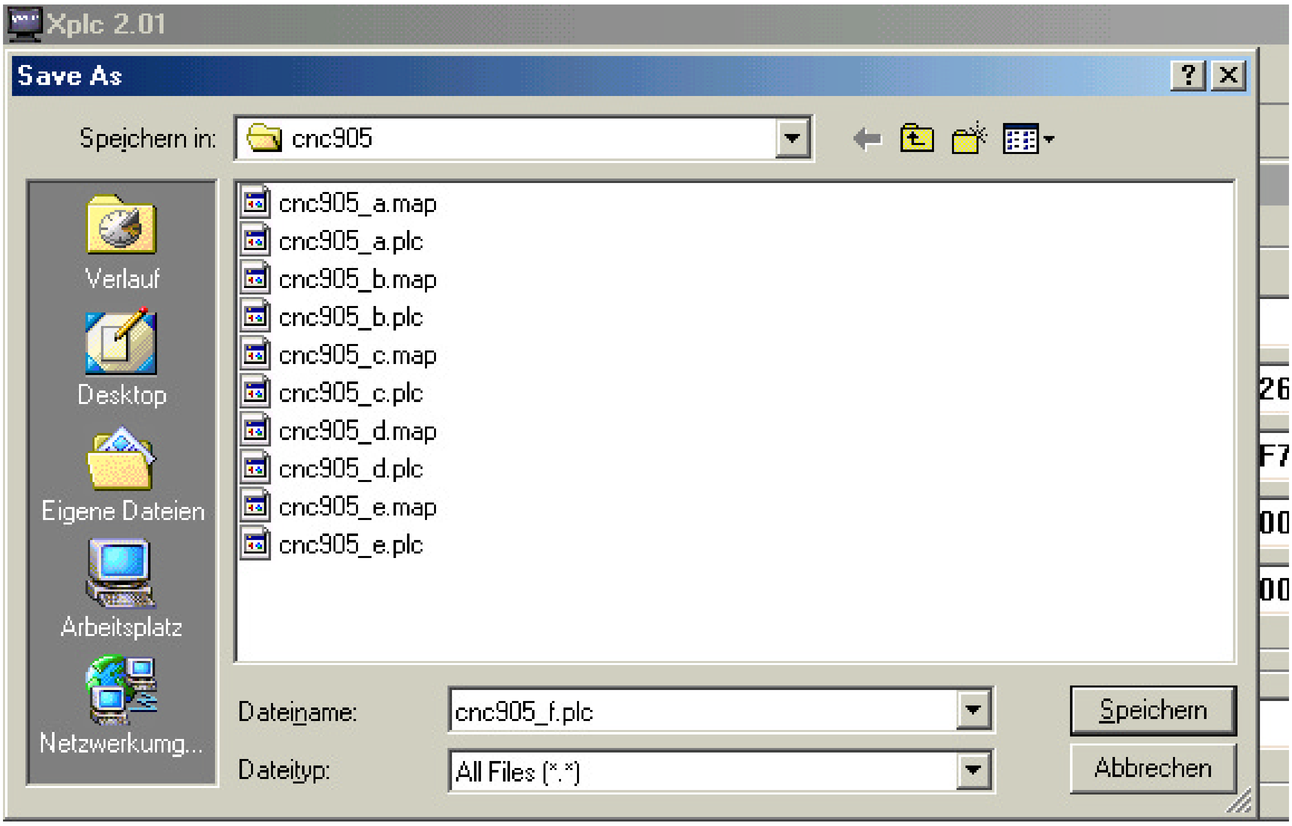
cnc905_a.plc
cnc905_b.plc
cnc905_c.plc
cnc905_d.plc
cnc905_e.plc

Dateiname: cnc905_e.plc

Dateityp: *.plc

5.2.2 Save as

Save as a XPLC project



Speichern in: cnc905

cnc905_a.map
cnc905_a.plc
cnc905_b.map
cnc905_b.plc
cnc905_c.map
cnc905_c.plc
cnc905_d.map
cnc905_d.plc
cnc905_e.map
cnc905_e.plc

Dateiname: cnc905_f.plc

Dateityp: All Files (*.*)

5.3 XPLC over WINBV start and terminate

A prerequisite WINBV version 5.5 and path specification for XPLC in the file WINBV.INI.

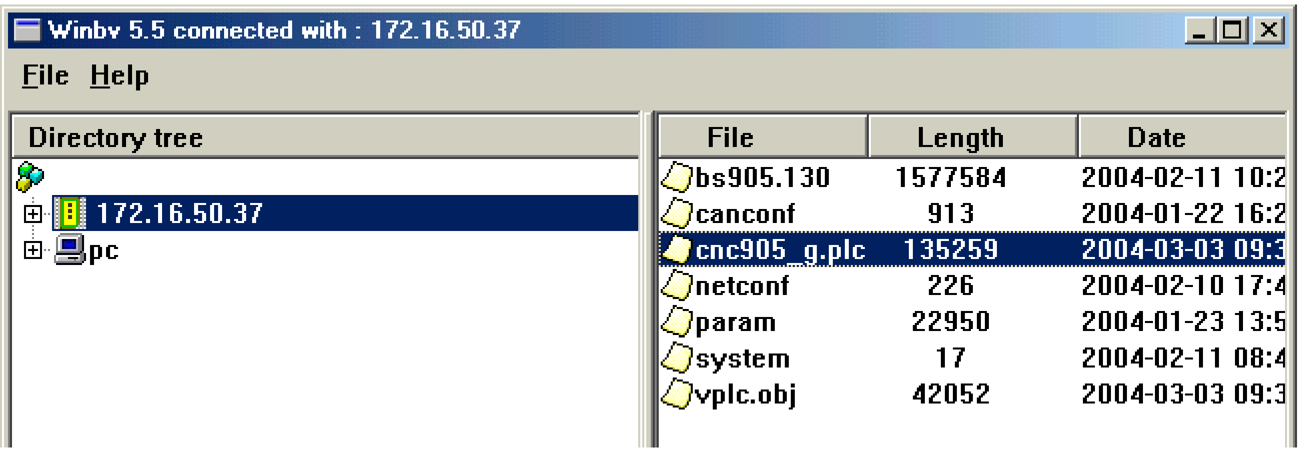
```
cnc_address=172.16.50.37
iservice=217.6.206.178
grafik_editor=c:\programme\gimp\bin\gimp.exe
xplc_editor=c:\programme\bwo\data\xplc\xplc.exe     *
identification=bwo.winbv
deutsch
plc_path=c:\plc900
xplconlinedata=c:\programme\bwo\data\xplc\online/   **
```

* Path for XPLC software

** Path to the XPLC work directory

5.3.1 XPLC project file loade

XPLC project file (source) in the general statement of the CNC CPU mark (link mouse button).



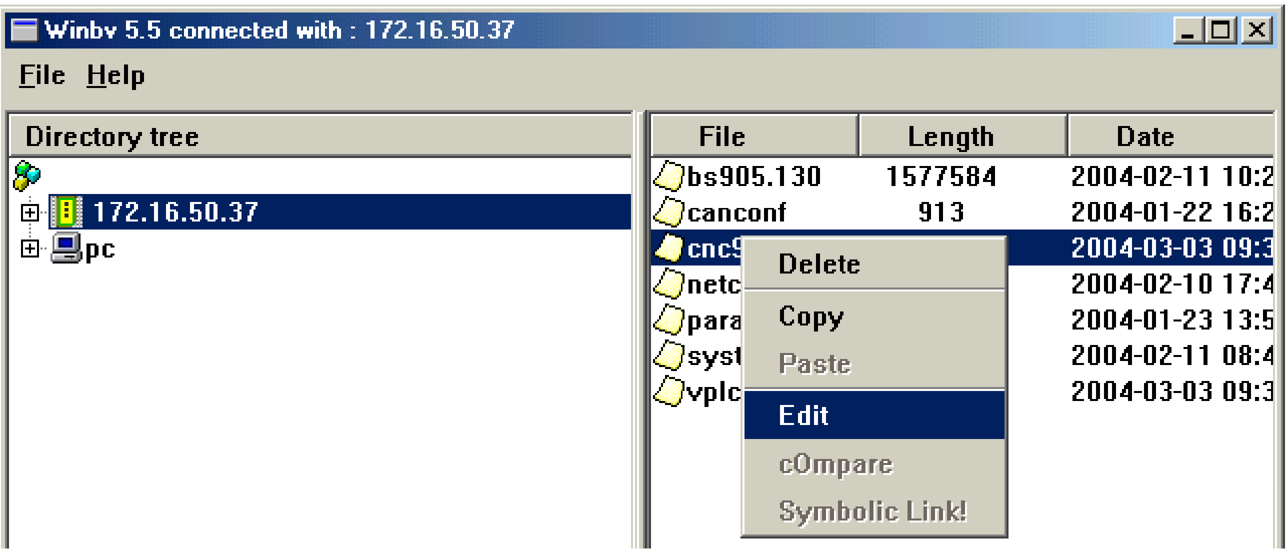
Winbv 5.5 connected with : 172.16.50.37

File	Length	Date
bs905.130	1577584	2004-02-11 10:2
canconf	913	2004-01-22 16:2
cnc905_g.plc	135259	2004-03-03 09:3
netconf	226	2004-02-10 17:4
param	22950	2004-01-23 13:5
system	17	2004-02-11 08:4
vplc.obj	42052	2004-03-03 09:3

5.3.1 XPLC project file charge (continued)

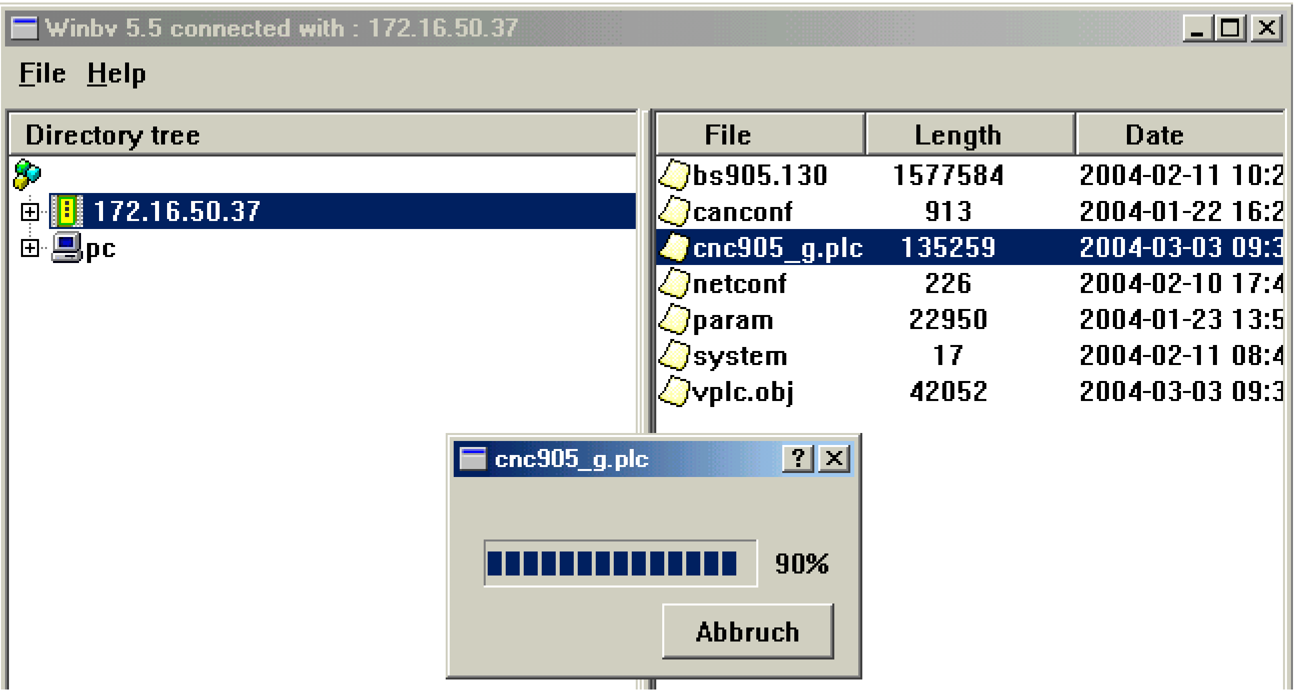
Right mouse button opens windows

Selection processing (link mouse button)



File	Length	Date
bs905.130	1577584	2004-02-11 10:2
canconf	913	2004-01-22 16:2
cnc9		2004-03-03 09:3
netc		2004-02-10 17:4
para		2004-01-23 13:5
syst		2004-02-11 08:4
vplc		2004-03-03 09:3

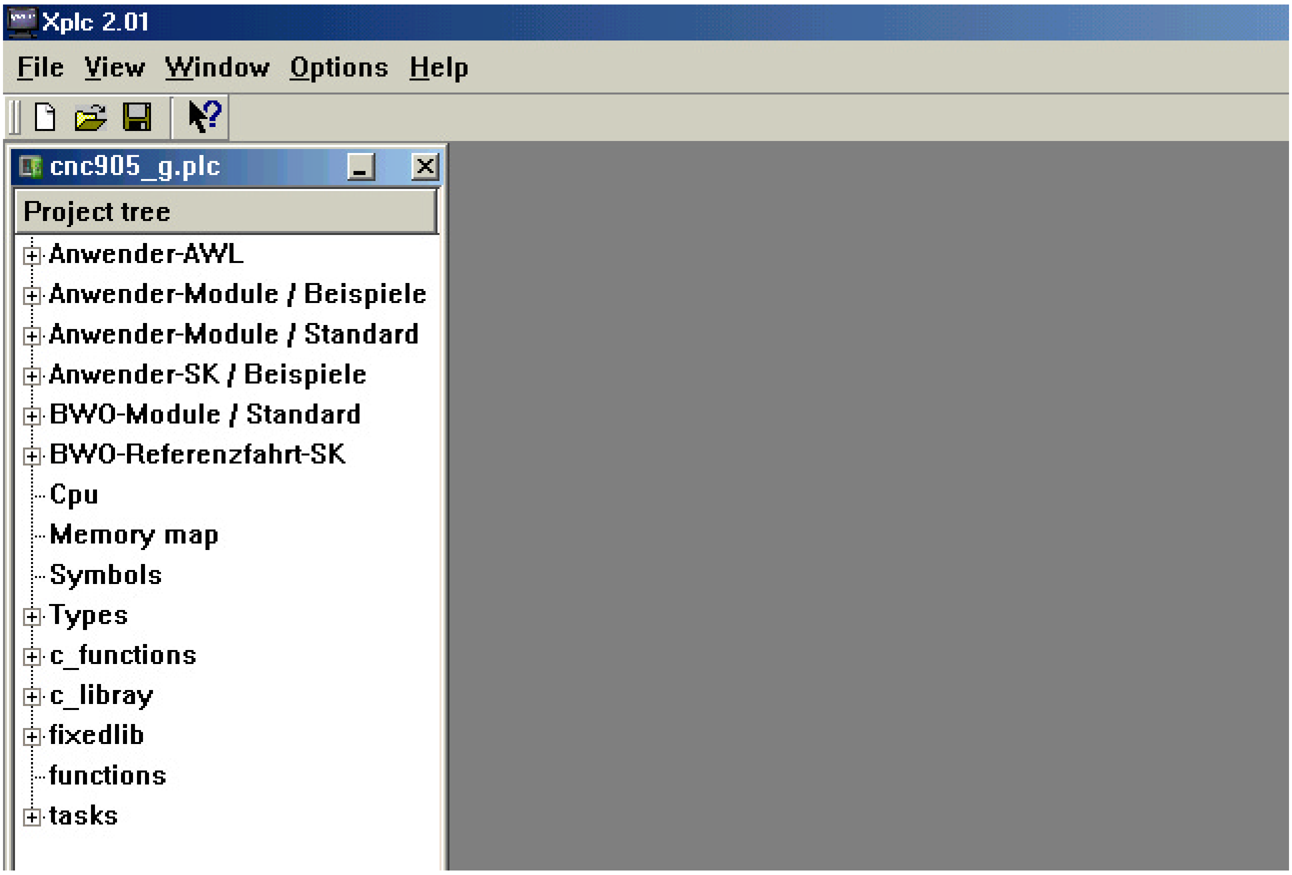
XPLC project file (source) is loaded of CNC CPU memory into the XPLC work directory by the PC (program unit).



File	Length	Date
bs905.130	1577584	2004-02-11 10:2
canconf	913	2004-01-22 16:2
cnc905_g.plc	135259	2004-03-03 09:3
netconf	226	2004-02-10 17:4
param	22950	2004-01-23 13:5
system	17	2004-02-11 08:4
vplc.obj	42052	2004-03-03 09:3

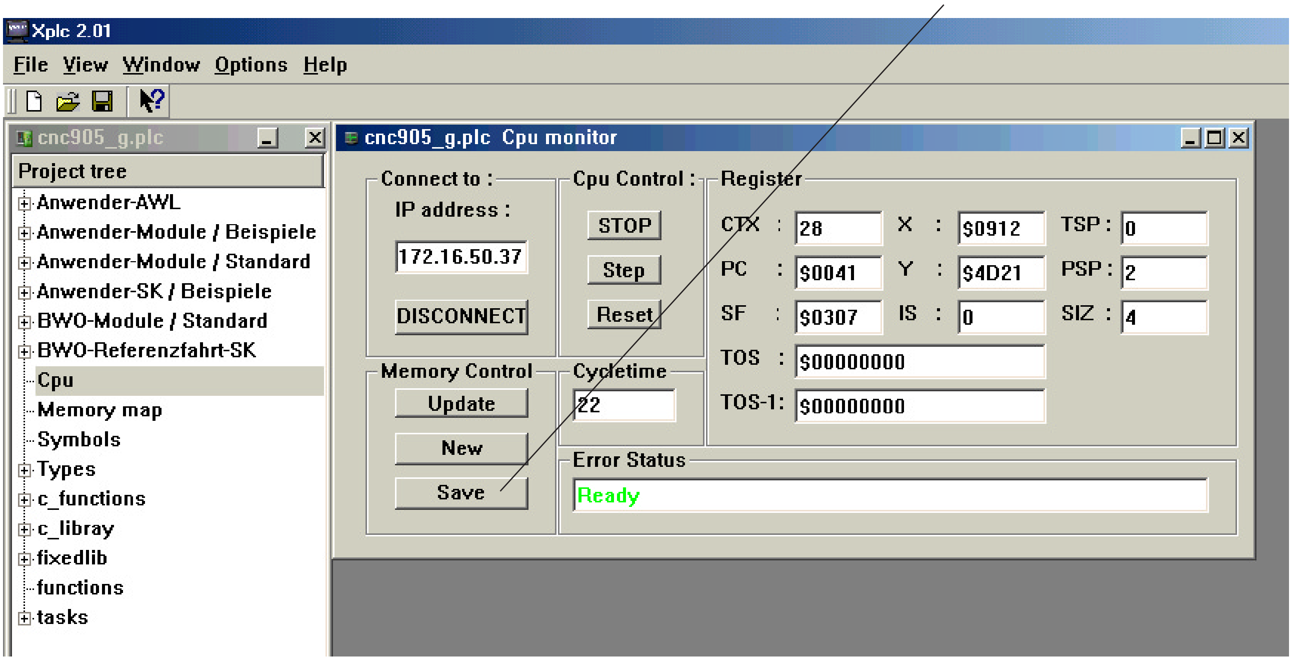
5.3.1 XPLC project file charge (continued)

The XPLC software actual started and the project file opened.



5.3.2 XPLC Projektdatei save after edit

Memory of the XPLC project file (object code) after EEPROM with key save.



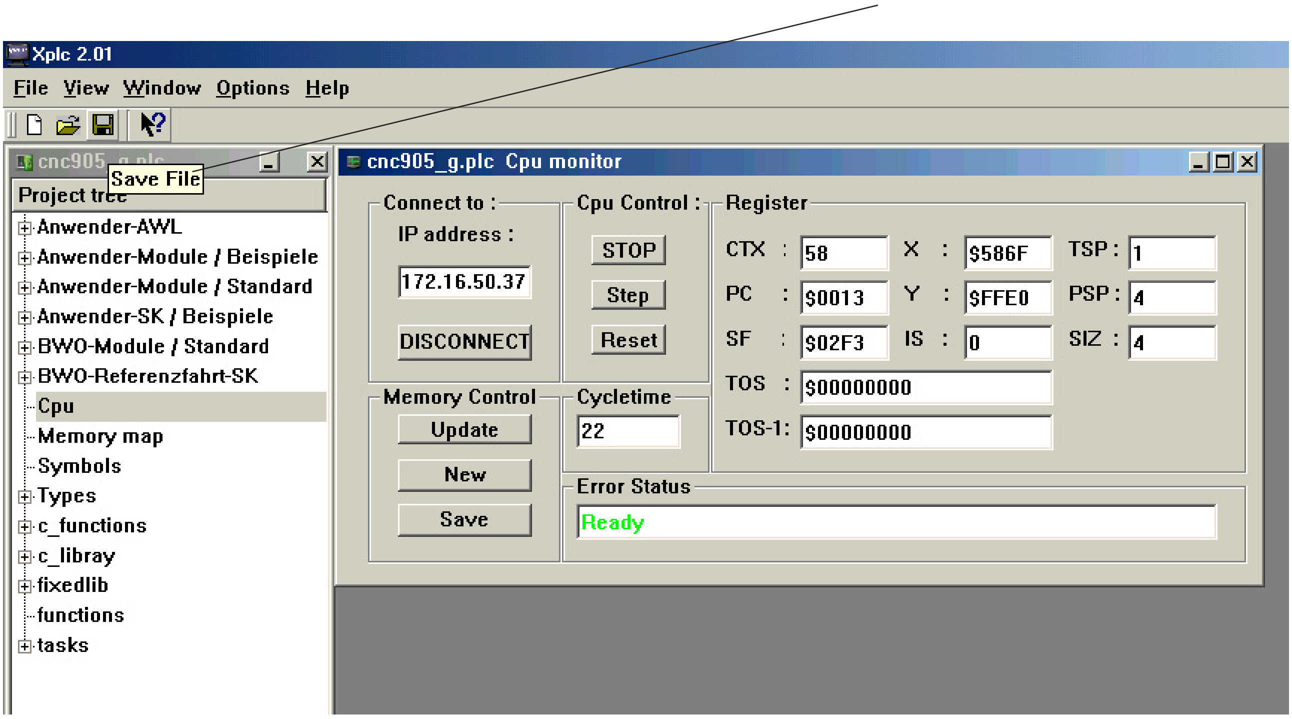
IP address : 172.16.50.37

Cycletime: 22

CTX	: 28	X	: $0912	TSP :	0
PC	: $0041	Y	: $4D21	PSP :	2
SF	: $0307	IS	: 0	SIZ :	4
TOS	: $00000000				
TOS-1:	$00000000				

Error Status: Ready

XPLC project file (source) after work directory PC (program unit) save.



IP address : 172.16.50.37

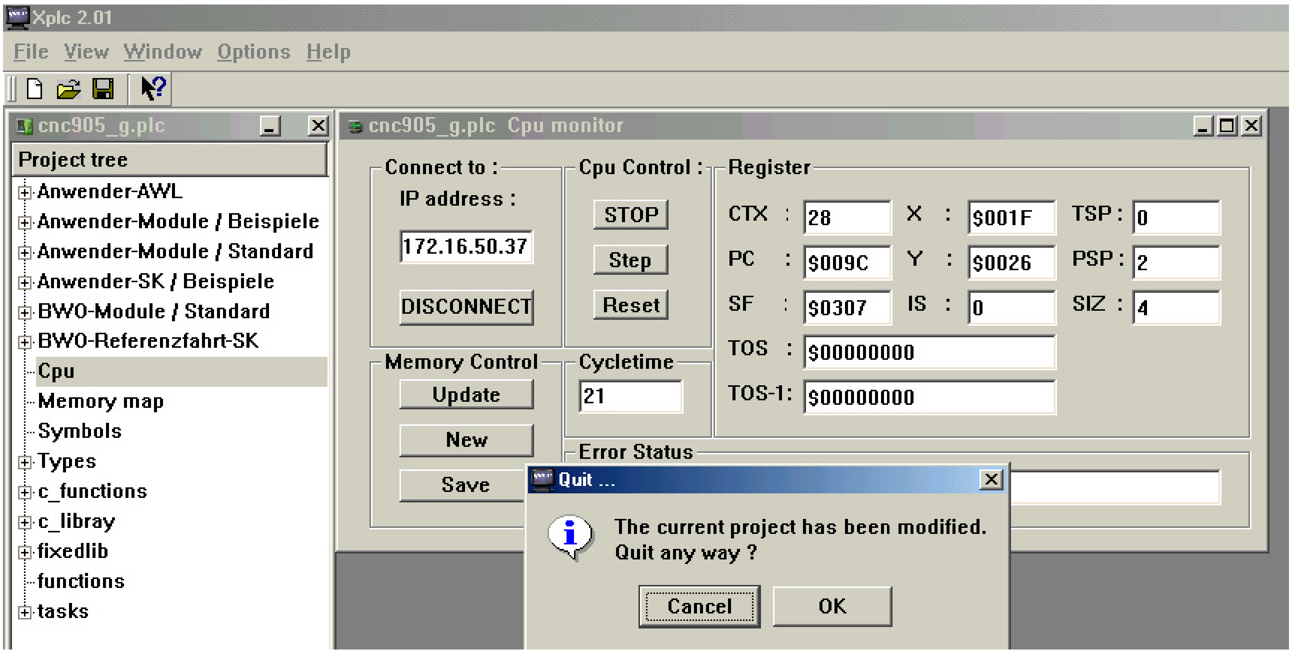
Cycletime: 22

CTX	: 58	X	: $586F	TSP :	1
PC	: $0013	Y	: $FFE0	PSP :	4
SF	: $02F3	IS	: 0	SIZ :	4
TOS	: $00000000				
TOS-1:	$00000000				

Error Status: Ready

5.3.2 XPLC Projektdatei save after edit (continued)

Command XPLC to be closed, without the XPLC project file (source) was saved into the work directory of the PC (program unit), appears the following message.



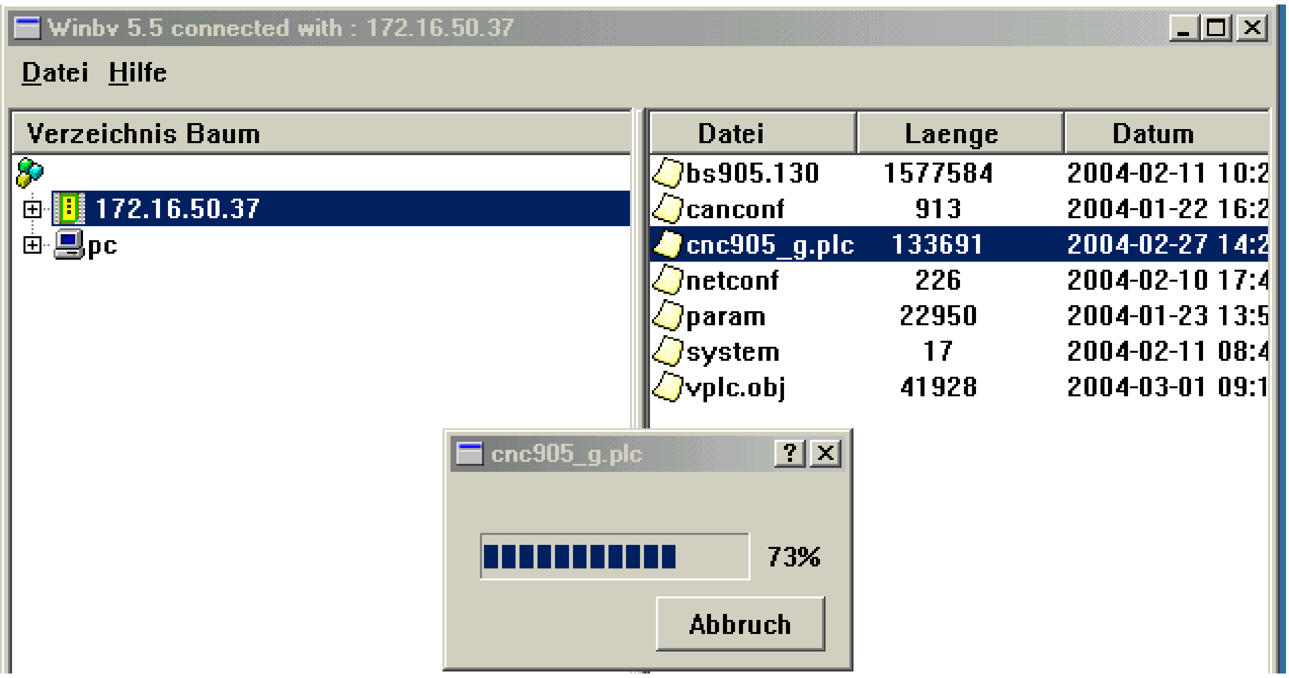
IP address : 172.16.50.37

CTX :	28	X :	$001F	TSP :	0
PC :	$009C	Y :	$0026	PSP :	2
SF :	$0307	IS :	0	SIZ :	4
TOS :	$00000000				
TOS-1:	$00000000				

Cycletime: 21

The current project has been modified.
Quit any way ?

With the closing of XPLC the XPLC project file (source) of the PC work directory in save CNC CPU back-saved.



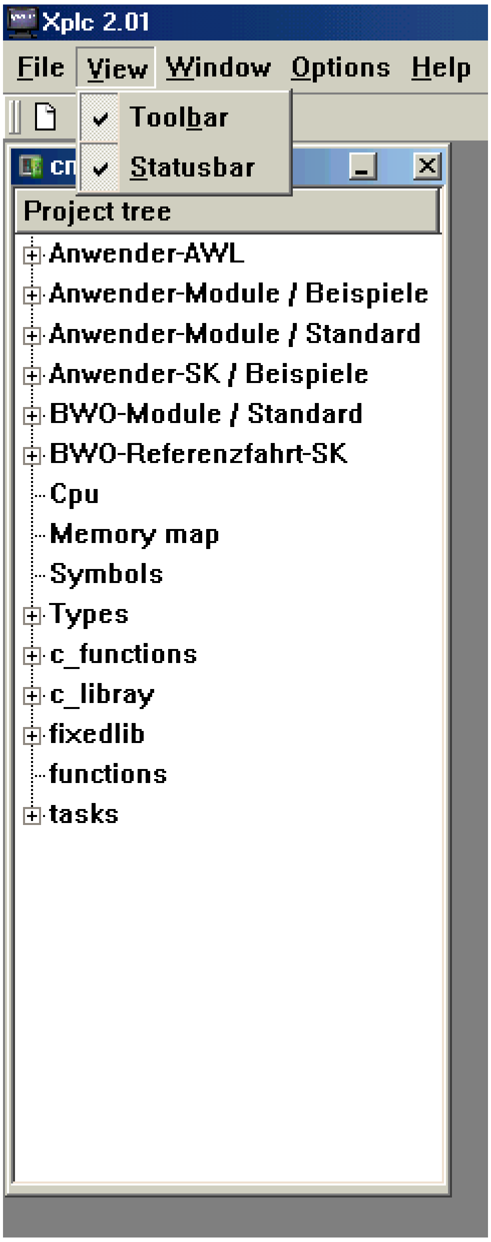
Datei	Laenge	Datum
bs905.130	1577584	2004-02-11 10:2
canconf	913	2004-01-22 16:2
cnc905_g.plc	133691	2004-02-27 14:2
netconf	226	2004-02-10 17:4
param	22950	2004-01-23 13:5
system	17	2004-02-11 08:4
vplc.obj	41928	2004-03-01 09:1

73%

5.4 Menue View

Standard Windows Function

Toolbar	on / off
Statusbar	on / off



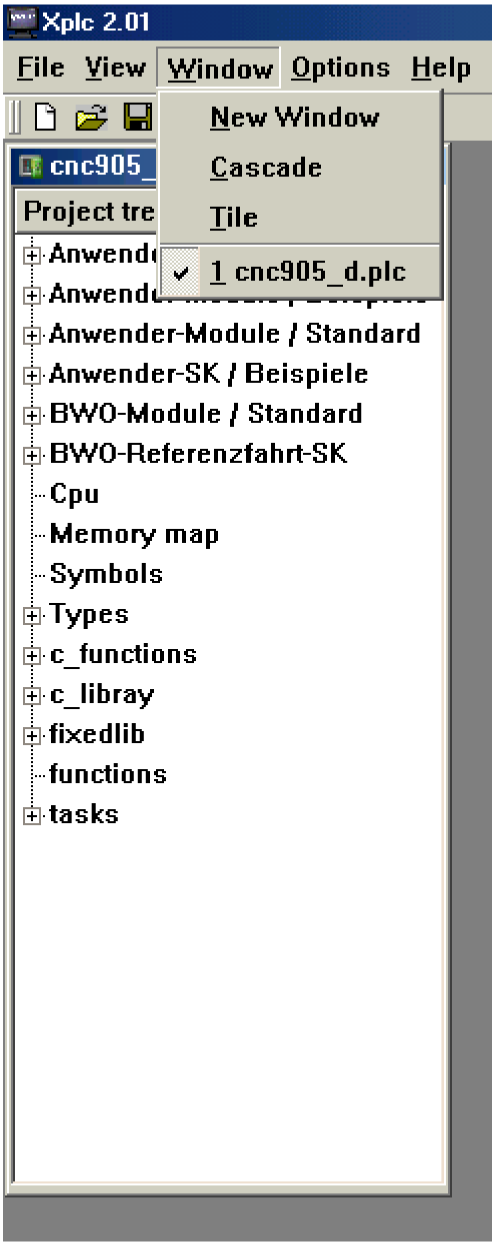
5.5 Menue Window

Standard Windows Management

New Window

Cascade

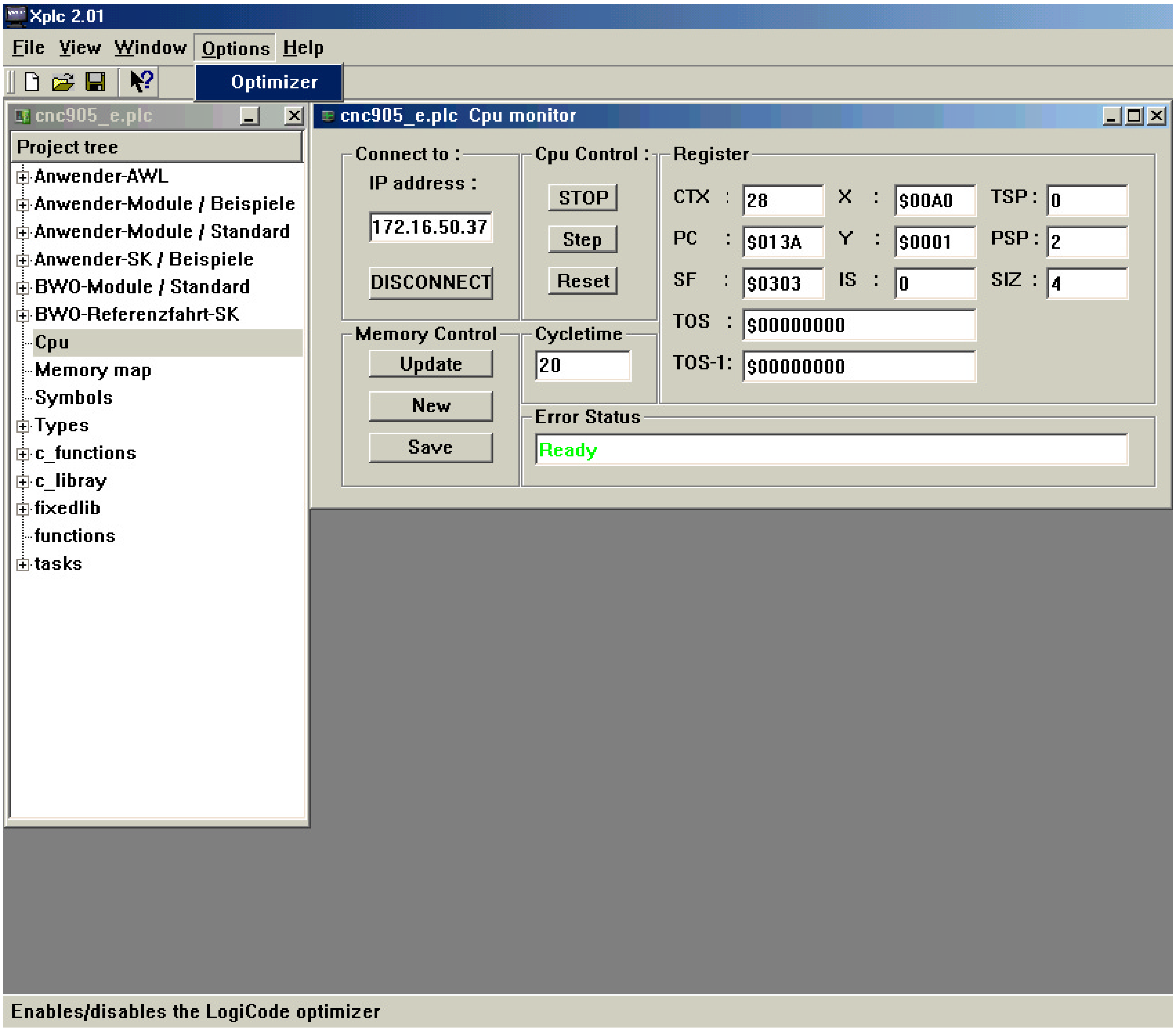
Tile



5.6 Menue Options

LogiCode optimization on / off

on > certain code sequences are accelerated.



IP address : 172.16.50.37

Cycletime: 20

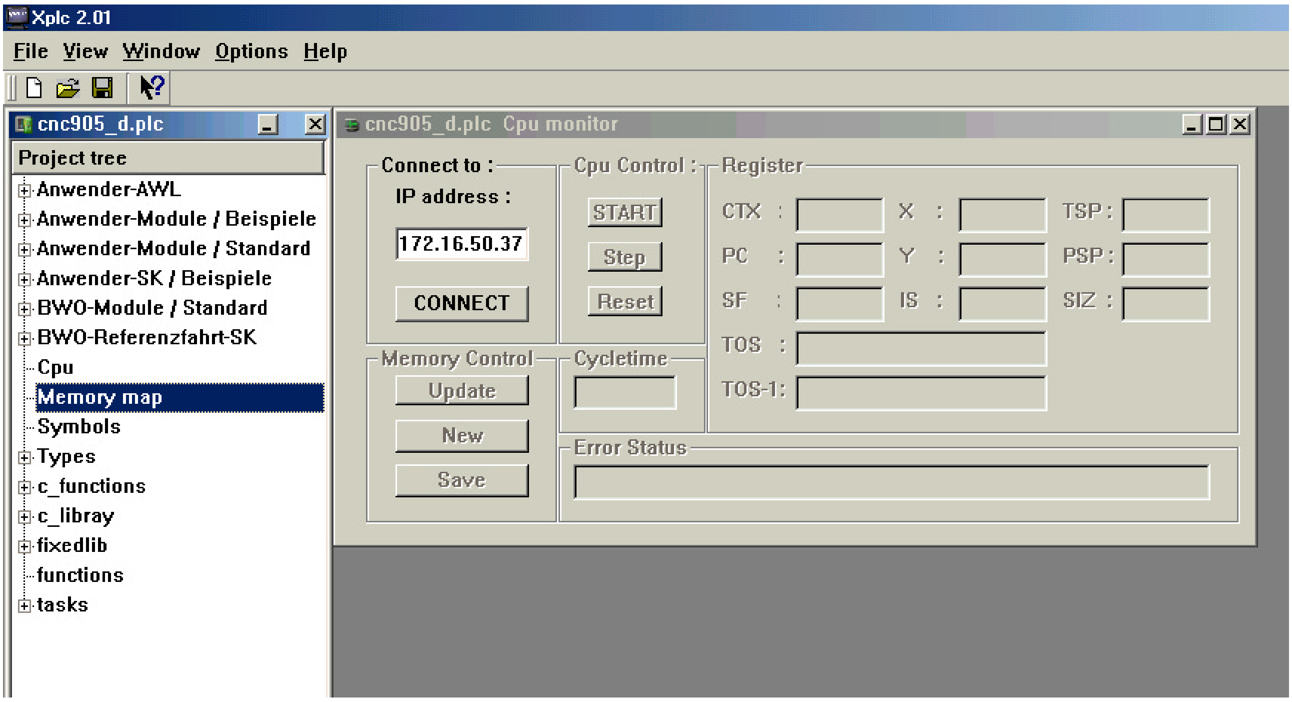
CTX	:	28	X	:	$00A0	TSP :	0
PC	:	$013A	Y	:	$0001	PSP :	2
SF	:	$0303	IS	:	0	SIZ :	4
TOS	:	$00000000					
TOS-1:		$00000000					

Error Status: Ready

Enables/disables the LogiCode optimizer

5.7 CPU Monitor

Offline Mode

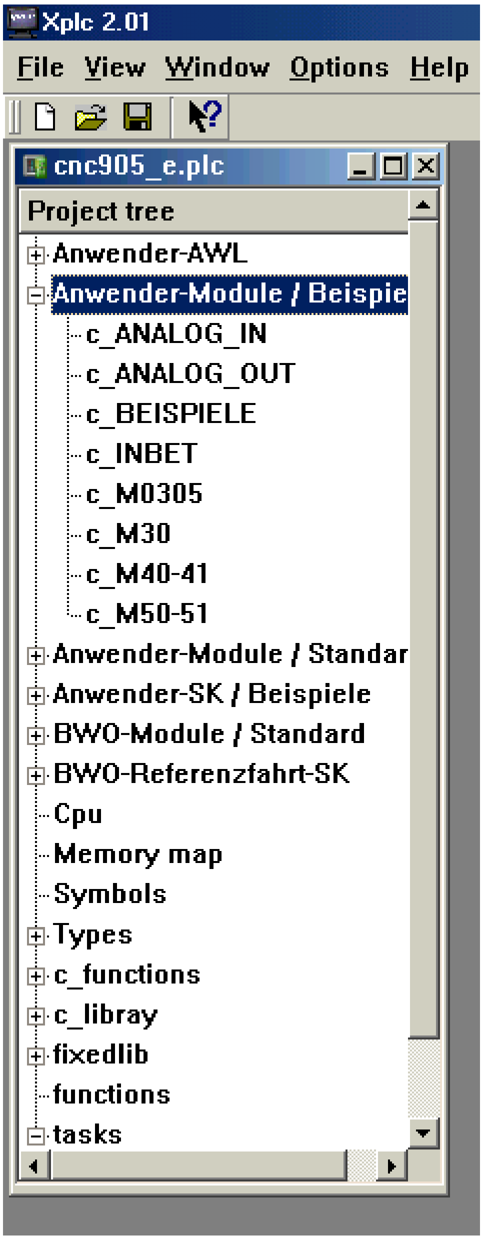


IP address:	IP address of the CNC
CONNECT:	On-line connection to the CNC make
DISCONNECT:	on-line connection to the CNC abort (Offline)

5.8 Project tree

Files **of user modules / examples**

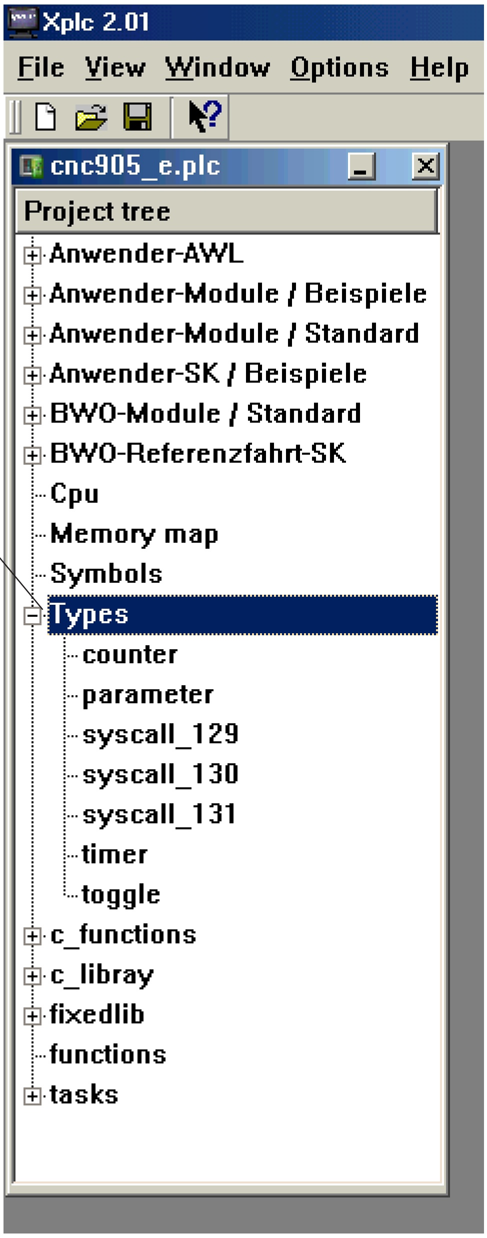
Contains various blocks with those user functions are implemented.



- Anwender-AWL
- Anwender-Module / Beispie
  - c_ANALOG_IN
  - c_ANALOG_OUT
  - c_BEISPIELE
  - c_INBET
  - c_M0305
  - c_M30
  - c_M40-41
  - c_M50-51
- Anwender-Module / Standar
- Anwender-SK / Beispiele
- BWO-Module / Standard
- BWO-Referenzfahrt-SK
- Cpu
- Memory map
- Symbols
- Types
- c_functions
- c_libray
- fixedlib
- functions
- tasks

5.8.1 Files of type

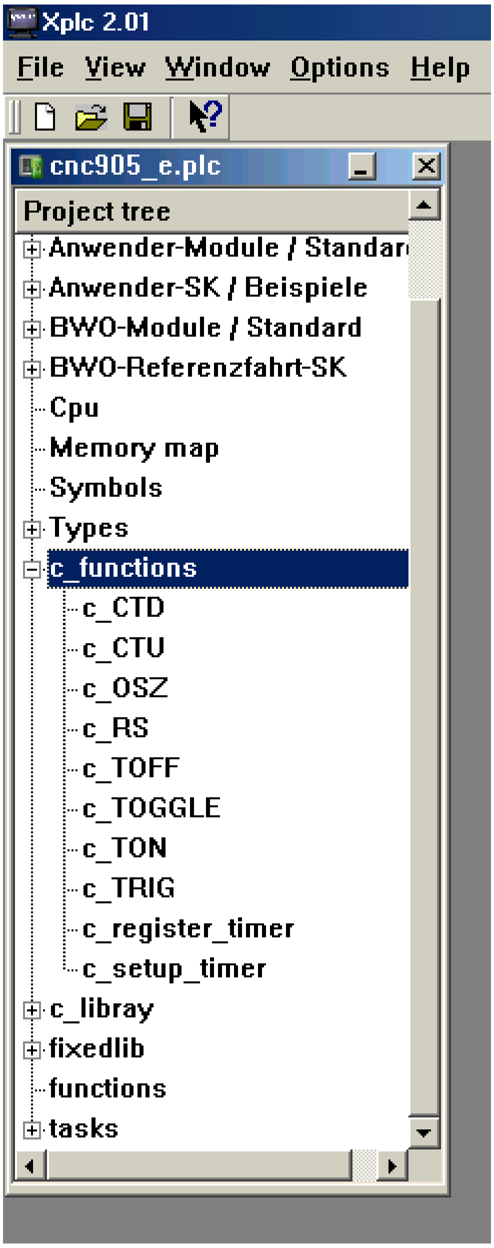
Library of the combined types
(see 4.5 type definition)



5.8.2 File c_functions

Library of the built-in function blocks
(see 4.9 standard function blocks)

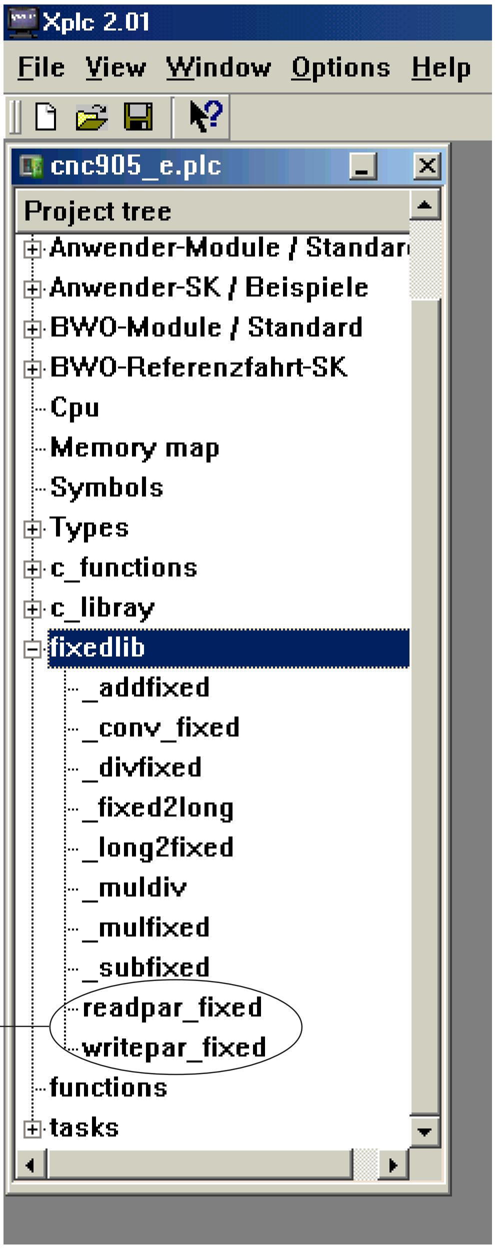
Library of further standard function blocks



5.8.3 File fixedlib

Library of the mathematical functional blocks
for fixed point arithmetic.
These functional blocks become from the compiler called, if printouts are used, those the type “ fixed “ contain.

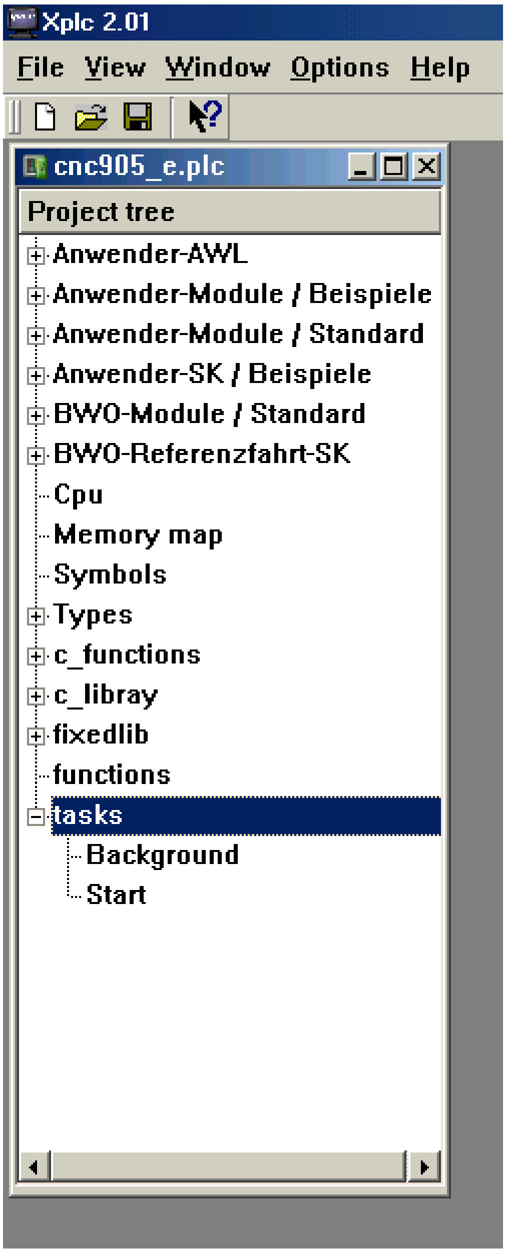
Functions for reading and writing of parameters in the fixed format.



5.8.4 File Tasks

Background task: actual the interface of the CNC to the PLC and contain the fast transfer of parameters, m-functions, messages, Bit information and clock pulse.

Start task: Contains of calls of the different standard and user modules and that program start.

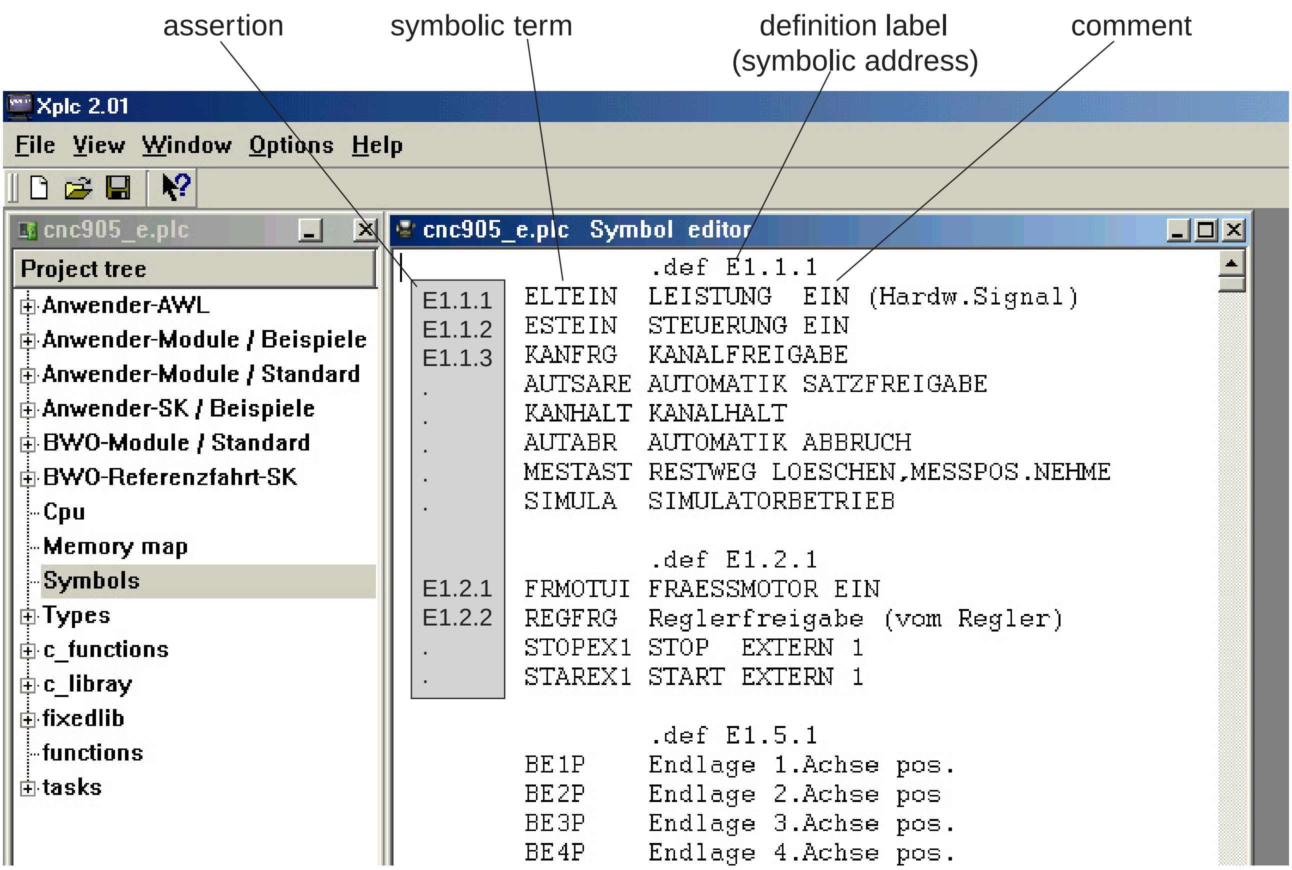


5.9 Editors

5.9.1 Symbol editor

Definition of inputs

The symbol editor serves the allocation of symbol terms in fixed symbolic addresses and for it belonging explaining comments.



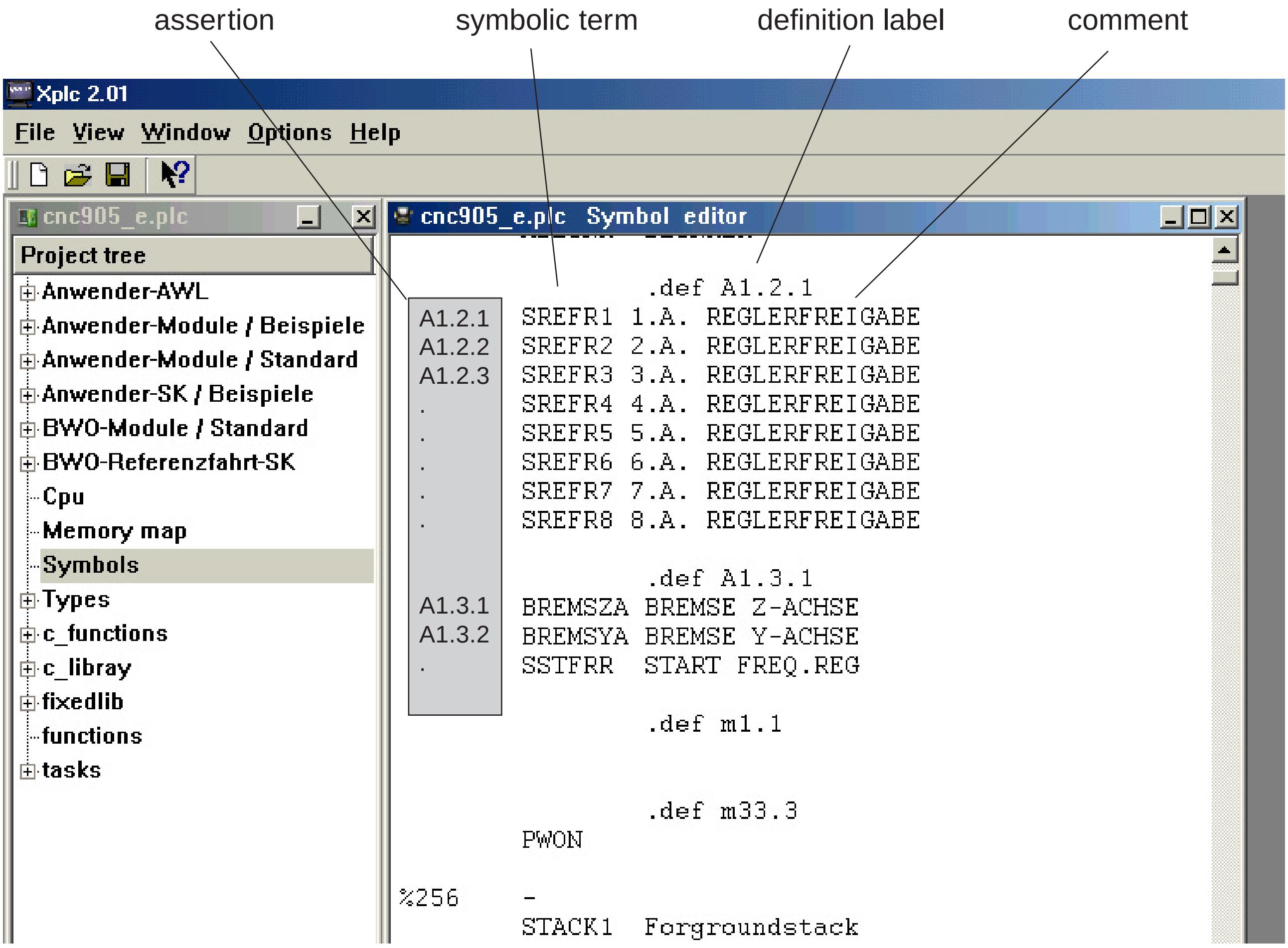
```
                .def E1.1.1
E1.1.1  ELTEIN  LEISTUNG  EIN (Hardw.Signal)
E1.1.2  ESTEIN  STEUERUNG EIN
E1.1.3  KANFRG  KANALFREIGABE
.       AUTSARE AUTOMATIK SATZFREIGABE
.       KANHALT KANALHALT
.       AUTABR  AUTOMATIK ABBRUCH
.       MESTAST RESTWEG LOESCHEN,MESSPOS.NEHME
.       SIMULA  SIMULATORBETRIEB

                .def E1.2.1
E1.2.1  FRMOTUI FRAESSMOTOR EIN
E1.2.2  REGFRG  Reglerfreigabe (vom Regler)
.       STOPEX1 STOP  EXTERN 1
.       STAREX1 START EXTERN 1

                .def E1.5.1
        BE1P    Endlage 1.Achse pos.
        BE2P    Endlage 2.Achse pos
        BE3P    Endlage 3.Achse pos.
        BE4P    Endlage 4.Achse pos.
```

With so-called definition labels the symbolic addresses of the following inputs, outputs and flags are determined. The definition labels serve the allocation of inputs on certain modules of a CAN node. Without these labels the addresses are determined by the system. They are then stringed together in blocks to in each case 8 inputs.

```
.def E1.1.1
    | | └──── input number
    | └────── module number   (1 - 8)
    └──────── node number     (1 - 5)
```

5.9.1 Symbol editor (continued)

Definition of outputs



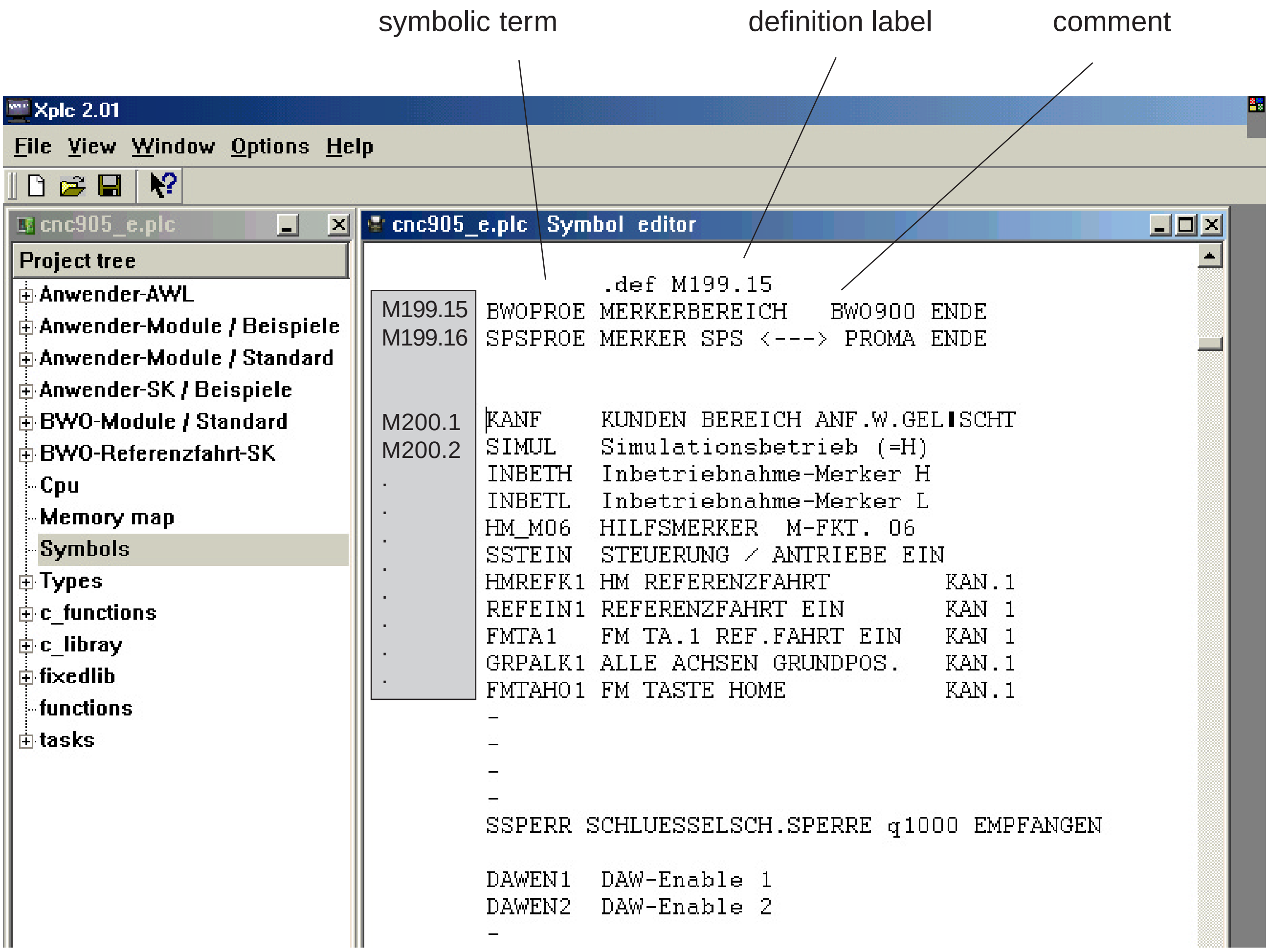
```
                   .def A1.2.1
A1.2.1   SREFR1 1.A. REGLERFREIGABE
A1.2.2   SREFR2 2.A. REGLERFREIGABE
A1.2.3   SREFR3 3.A. REGLERFREIGABE
.        SREFR4 4.A. REGLERFREIGABE
.        SREFR5 5.A. REGLERFREIGABE
.        SREFR6 6.A. REGLERFREIGABE
.        SREFR7 7.A. REGLERFREIGABE
.        SREFR8 8.A. REGLERFREIGABE

                   .def A1.3.1
A1.3.1   BREMSZA BREMSE Z-ACHSE
A1.3.2   BREMSYA BREMSE Y-ACHSE
.        SSTFRR  START FREQ.REG

                   .def m1.1


                   .def m33.3
         PWON

%256     -
         STACK1  Forgroundstack
```

The definition labels serve the allocation of outputs on certain modules of a CAN node. Without these labels the addresses are determined by the system. They are stringed together by the system in blocks to in each case 8 outputs.

```
.def A1.1.1
      | | └──── output number
      | └────── module number   (1 - 8)
      └──────── node number     (1 - 5)
```

5.9.1 Symbol editor (continued)

Definition of flags



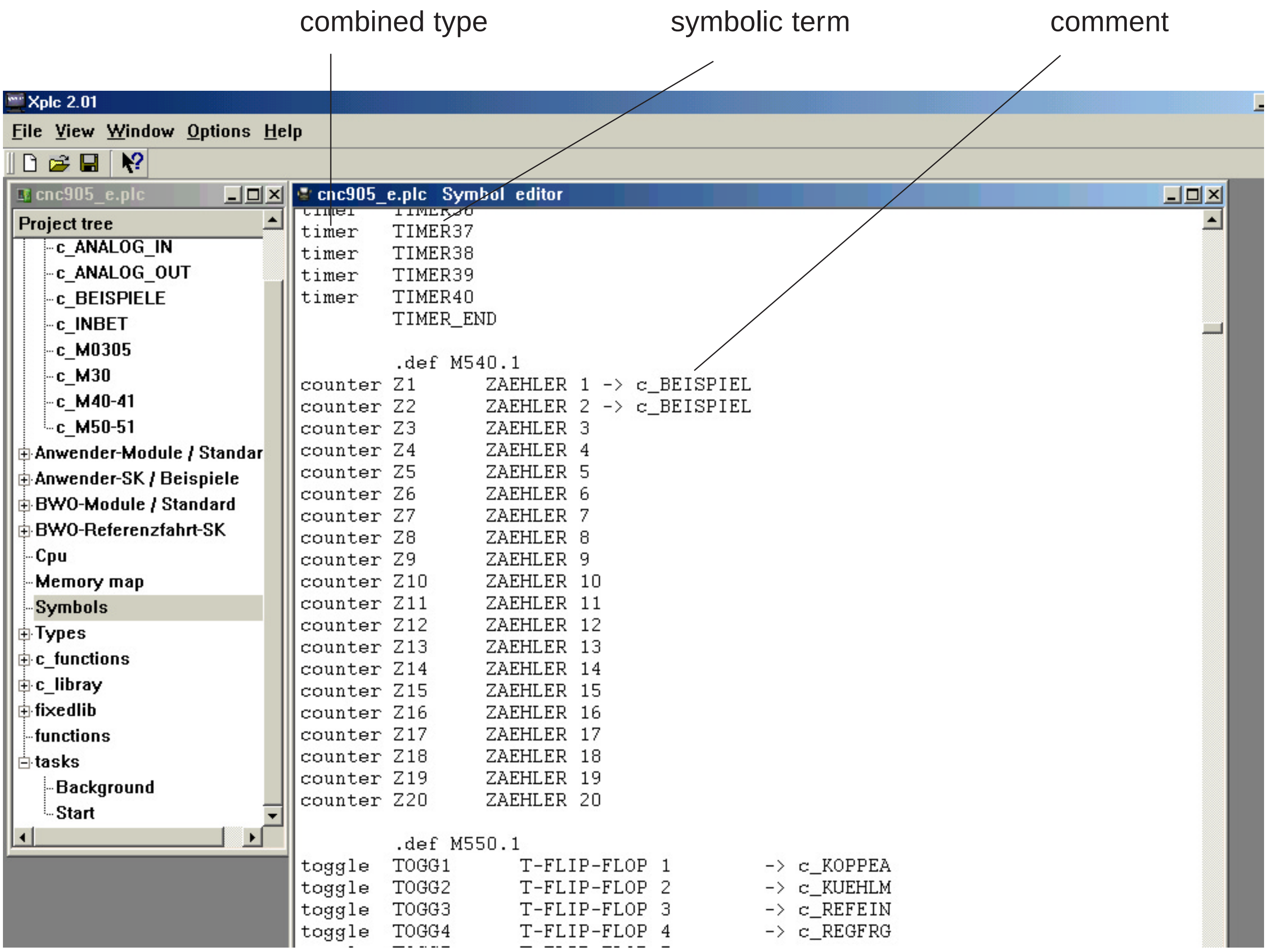
```
                .def M199.15
M199.15 BWOPROE MERKERBEREICH   BWO900 ENDE
M199.16 SPSPROE MERKER SPS <---> PROMA ENDE


M200.1  KANF    KUNDEN BEREICH ANF.W.GEL▌SCHT
M200.2  SIMUL   Simulationsbetrieb (=H)
.       INBETH  Inbetriebnahme-Merker H
.       INBETL  Inbetriebnahme-Merker L
.       HM_M06  HILFSMERKER  M-FKT. 06
.       SSTEIN  STEUERUNG / ANTRIEBE EIN
.       HMREFK1 HM REFERENZFAHRT        KAN.1
.       REFEIN1 REFERENZFAHRT EIN       KAN 1
.       FMTA1   FM TA.1 REF.FAHRT EIN   KAN 1
.       GRPALK1 ALLE ACHSEN GRUNDPOS.   KAN.1
        FMTAHO1 FM TASTE HOME           KAN.1
        -
        -
        -
        -
        SSPERR SCHLUESSELSCH.SPERRE q1000 EMPFANGEN

        DAWEN1  DAW-Enable 1
        DAWEN2  DAW-Enable 2
        -
```

With definition labels the flags can be divided in groups (for example according to functions). The definition label determines the symbolic address for the following flag. Without this labels the flags are stringed together by the system into blocks to in each case 16 flags.

```
.def M199.15
        |   └──── flag number
        └──────── flag block number
```

5.9.1 Symbol editor (continued)

Definitionen of types



```
timer   TIMER37
timer   TIMER38
timer   TIMER39
timer   TIMER40
        TIMER_END

        .def M540.1
counter Z1      ZAEHLER 1 -> c_BEISPIEL
counter Z2      ZAEHLER 2 -> c_BEISPIEL
counter Z3      ZAEHLER 3
counter Z4      ZAEHLER 4
counter Z5      ZAEHLER 5
counter Z6      ZAEHLER 6
counter Z7      ZAEHLER 7
counter Z8      ZAEHLER 8
counter Z9      ZAEHLER 9
counter Z10     ZAEHLER 10
counter Z11     ZAEHLER 11
counter Z12     ZAEHLER 12
counter Z13     ZAEHLER 13
counter Z14     ZAEHLER 14
counter Z15     ZAEHLER 15
counter Z16     ZAEHLER 16
counter Z17     ZAEHLER 17
counter Z18     ZAEHLER 18
counter Z19     ZAEHLER 19
counter Z20     ZAEHLER 20

        .def M550.1
toggle  TOGG1      T-FLIP-FLOP 1         -> c_KOPPEA
toggle  TOGG2      T-FLIP-FLOP 2         -> c_KUEHLM
toggle  TOGG3      T-FLIP-FLOP 3         -> c_REFEIN
toggle  TOGG4      T-FLIP-FLOP 4         -> c_REGFRG
```

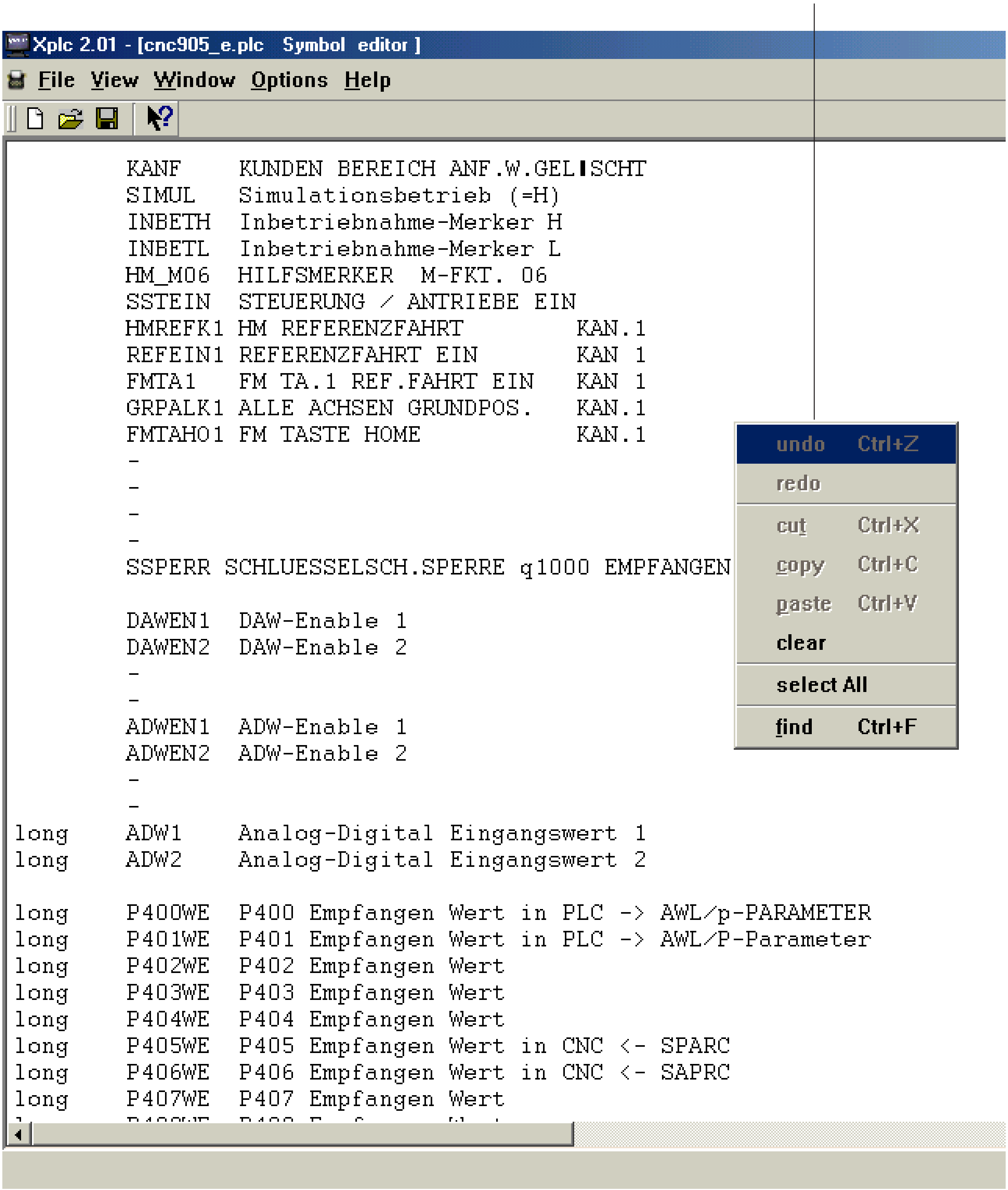
Combined types, for example: „ timers „ / „ counter „ / „ toggle „

Z1 counter 1 starts with the symbolic address M540.1 and occupies due to the type definition additionally the following 5 flags.

5.9.1 Symbol editor (continued)

Tool in the symbol editor

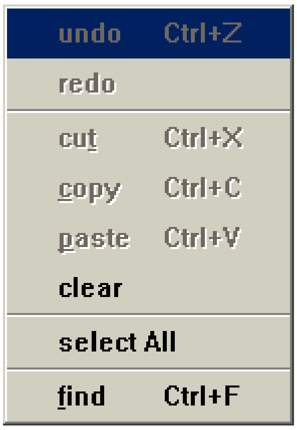
Rights mouse button



```
        KANF    KUNDEN BEREICH ANF.W.GEL▮SCHT
        SIMUL   Simulationsbetrieb (=H)
        INBETH  Inbetriebnahme-Merker H
        INBETL  Inbetriebnahme-Merker L
        HM_M06  HILFSMERKER  M-FKT. 06
        SSTEIN  STEUERUNG / ANTRIEBE EIN
        HMREFK1 HM REFERENZFAHRT        KAN.1
        REFEIN1 REFERENZFAHRT EIN       KAN 1
        FMTA1   FM TA.1 REF.FAHRT EIN   KAN 1
        GRPALK1 ALLE ACHSEN GRUNDPOS.   KAN.1
        FMTAHO1 FM TASTE HOME           KAN.1
        -
        -
        -
        -
        SSPERR SCHLUESSELSCH.SPERRE q1000 EMPFANGEN

        DAWEN1  DAW-Enable 1
        DAWEN2  DAW-Enable 2
        -
        -
        ADWEN1  ADW-Enable 1
        ADWEN2  ADW-Enable 2
        -
        -
long    ADW1    Analog-Digital Eingangswert 1
long    ADW2    Analog-Digital Eingangswert 2

long    P400WE  P400 Empfangen Wert in PLC -> AWL/p-PARAMETER
long    P401WE  P401 Empfangen Wert in PLC -> AWL/P-Parameter
long    P402WE  P402 Empfangen Wert
long    P403WE  P403 Empfangen Wert
long    P404WE  P404 Empfangen Wert
long    P405WE  P405 Empfangen Wert in CNC <- SPARC
long    P406WE  P406 Empfangen Wert in CNC <- SAPRC
long    P407WE  P407 Empfangen Wert
```

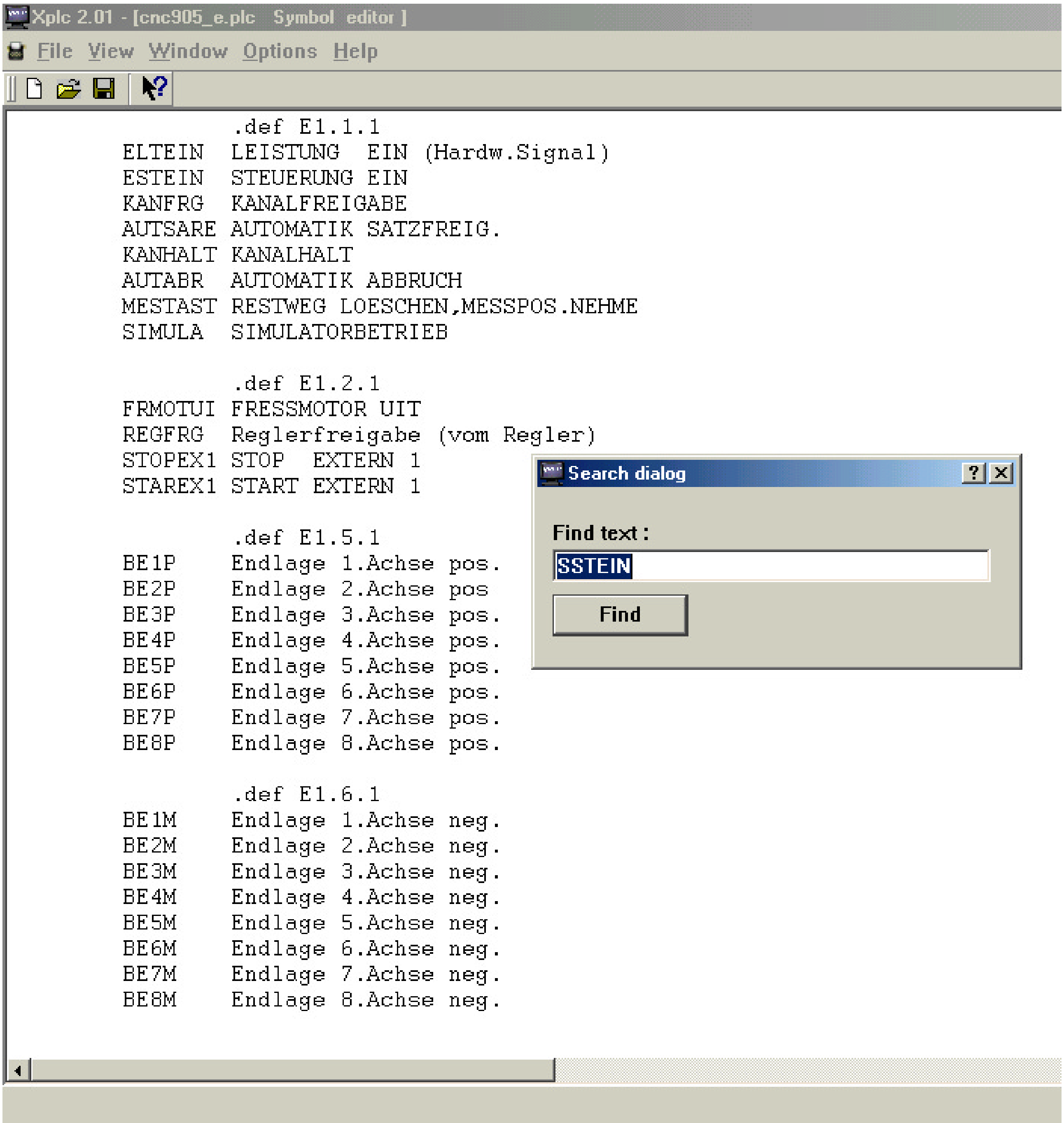
5.9.1 Symbol editor (continued)

The symbol editor makes the following **tools available.**
Hereby sections can be transferred off the symbolism also by a project to on different one.



5.9.1 Symbol editor (continued)

The search function **find** (Ctrl+F) enables a looking up of symbol terms in the symbolism section.



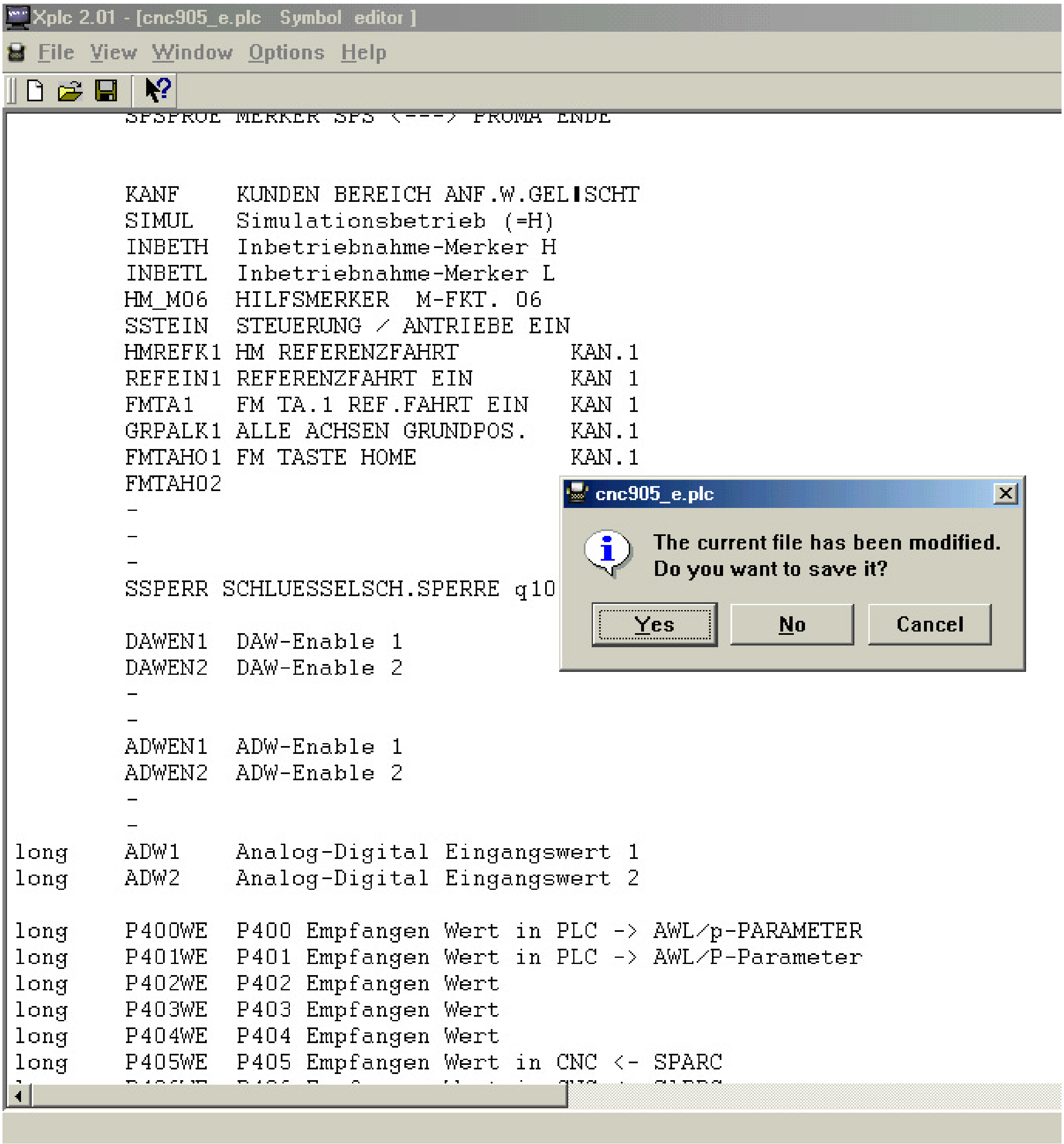
Xplc 2.01 - [cnc905_e.plc Symbol editor]
File View Window Options Help
Search dialog
Find text :
Find

```
              .def E1.1.1
       ELTEIN  LEISTUNG  EIN (Hardw.Signal)
       ESTEIN  STEUERUNG EIN
       KANFRG  KANALFREIGABE
       AUTSARE AUTOMATIK SATZFREIG.
       KANHALT KANALHALT
       AUTABR  AUTOMATIK ABBRUCH
       MESTAST RESTWEG LOESCHEN,MESSPOS.NEHME
       SIMULA  SIMULATORBETRIEB

               .def E1.2.1
       FRMOTUI FRESSMOTOR UIT
       REGFRG  Reglerfreigabe (vom Regler)
       STOPEX1 STOP  EXTERN 1
       STAREX1 START EXTERN 1

               .def E1.5.1
       BE1P    Endlage 1.Achse pos.
       BE2P    Endlage 2.Achse pos
       BE3P    Endlage 3.Achse pos.
       BE4P    Endlage 4.Achse pos.
       BE5P    Endlage 5.Achse pos.
       BE6P    Endlage 6.Achse pos.
       BE7P    Endlage 7.Achse pos.
       BE8P    Endlage 8.Achse pos.

               .def E1.6.1
       BE1M    Endlage 1.Achse neg.
       BE2M    Endlage 2.Achse neg.
       BE3M    Endlage 3.Achse neg.
       BE4M    Endlage 4.Achse neg.
       BE5M    Endlage 5.Achse neg.
       BE6M    Endlage 6.Achse neg.
       BE7M    Endlage 7.Achse neg.
       BE8M    Endlage 8.Achse neg.
```

Find text : SSTEIN

5.9.1 Symbol editor (continued)

Modifications in the symbolism definition store.

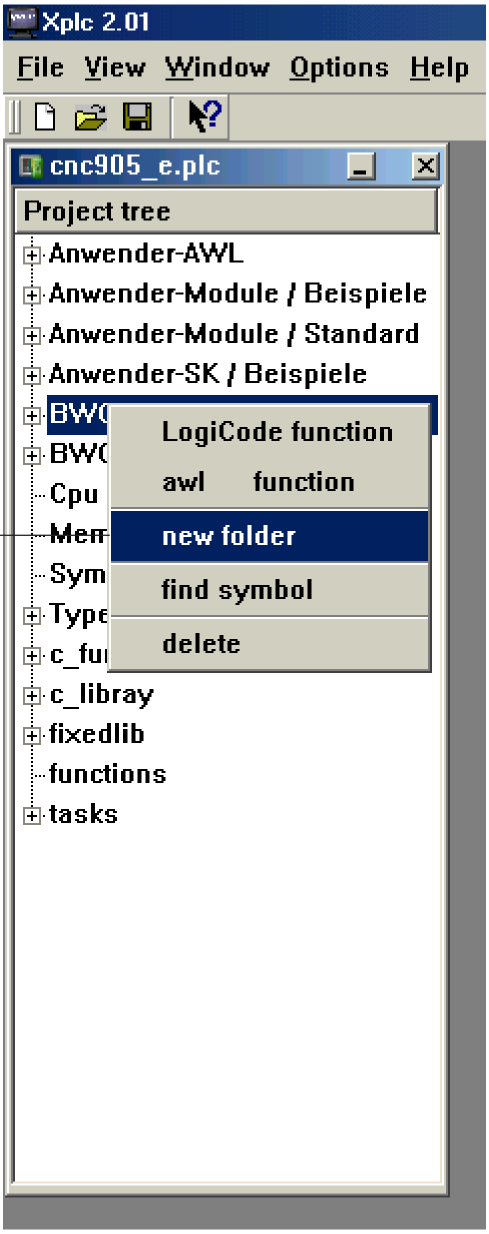

Xplc 2.01 - [cnc905_e.plc Symbol editor]
File View Window Options Help
cnc905_e.plc
Yes
No
Cancel

```
        SPSPROE MERKER SPS <---> PROMA ENDE


        KANF    KUNDEN BEREICH ANF.W.GEL▮SCHT
        SIMUL   Simulationsbetrieb (=H)
        INBETH  Inbetriebnahme-Merker H
        INBETL  Inbetriebnahme-Merker L
        HM_M06  HILFSMERKER  M-FKT. 06
        SSTEIN  STEUERUNG / ANTRIEBE EIN
        HMREFK1 HM REFERENZFAHRT        KAN.1
        REFEIN1 REFERENZFAHRT EIN       KAN 1
        FMTA1   FM TA.1 REF.FAHRT EIN   KAN 1
        GRPALK1 ALLE ACHSEN GRUNDPOS.   KAN.1
        FMTAH01 FM TASTE HOME           KAN.1
        FMTAH02
        -
        -
        -
        SSPERR SCHLUESSELSCH.SPERRE q10

        DAWEN1  DAW-Enable 1
        DAWEN2  DAW-Enable 2
        -
        -
        ADWEN1  ADW-Enable 1
        ADWEN2  ADW-Enable 2
        -
        -
long    ADW1    Analog-Digital Eingangswert 1
long    ADW2    Analog-Digital Eingangswert 2

long    P400WE  P400 Empfangen Wert in PLC -> AWL/p-PARAMETER
long    P401WE  P401 Empfangen Wert in PLC -> AWL/P-Parameter
long    P402WE  P402 Empfangen Wert
long    P403WE  P403 Empfangen Wert
long    P404WE  P404 Empfangen Wert
long    P405WE  P405 Empfangen Wert in CNC <- SPARC
long    P406WE  P406 Empfangen Wert in CNC <- SPARC
```

The current file has been modified.
Do you want to save it?

5.9.2 Program editor

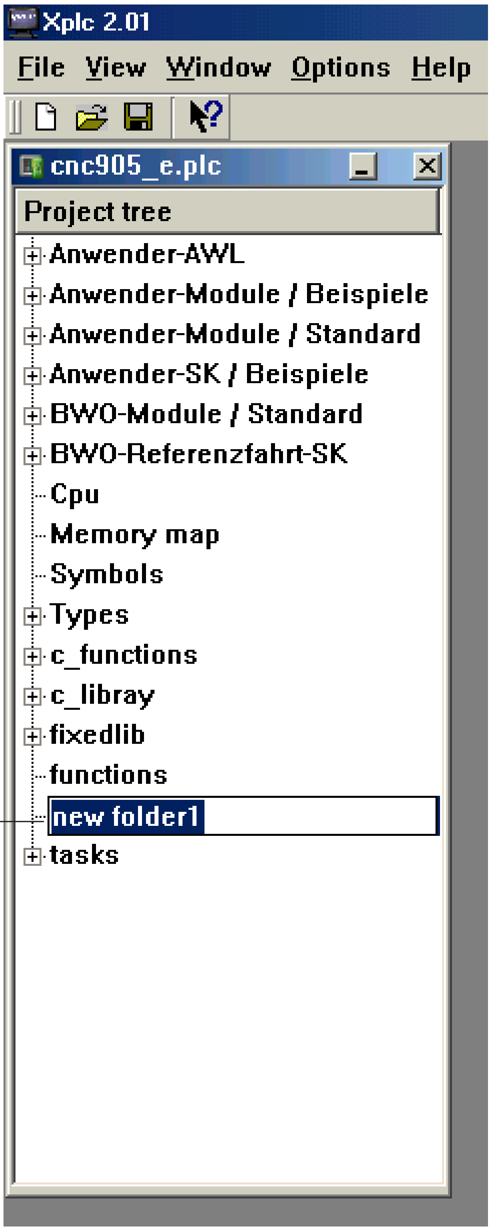
Function **new folder** :
New file for further blocks create.



Right mouse button

5.9.2 Program editor (continued)

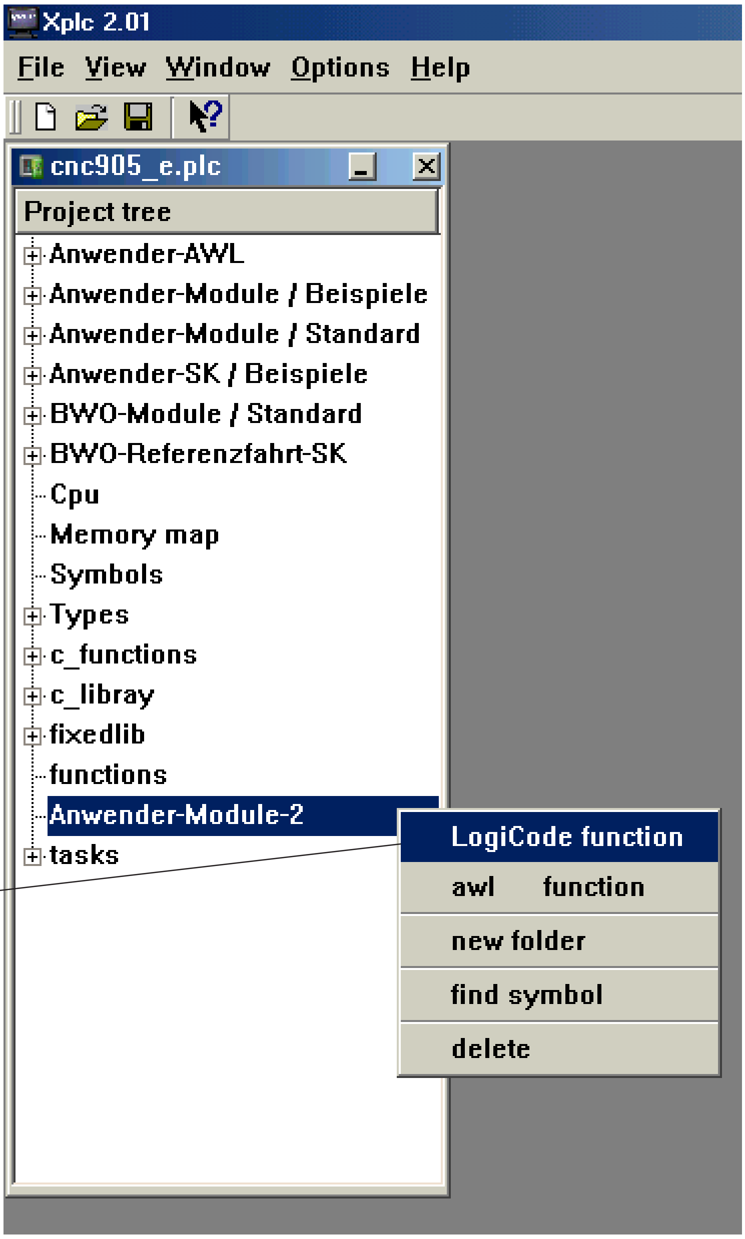
New created file rename.



Link mouse button
new term inputs, e.g. user modules 2

5.9.2 Program editor (continued)

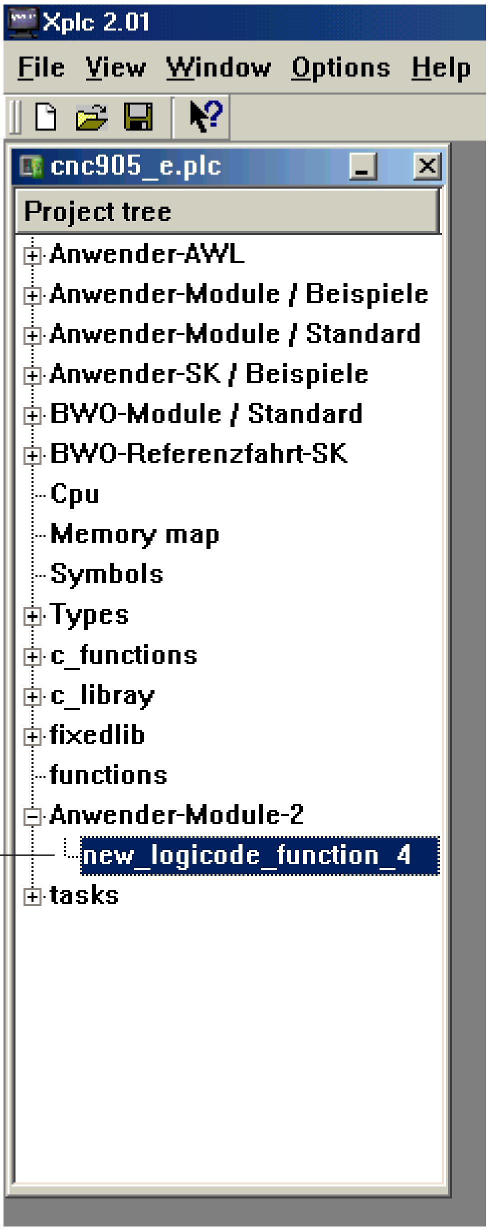
New LogiCode function create.



Right mouse button

5.9.2 Program editor (continued)

New created LogiCode function rename.



Xplc 2.01

File View Window Options Help

cnc905_e.plc

Project tree

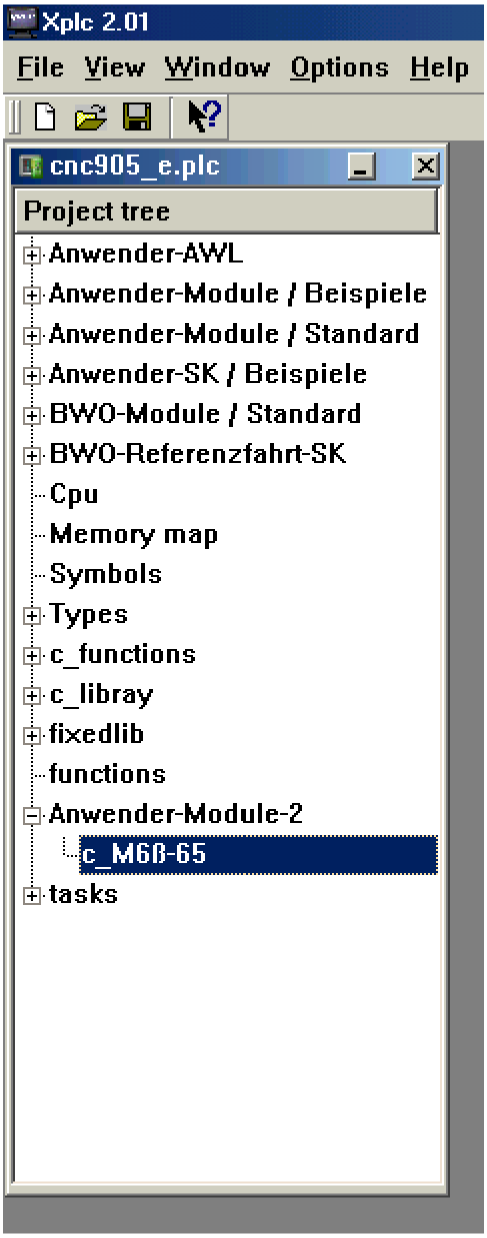
- Anwender-AWL
- Anwender-Module / Beispiele
- Anwender-Module / Standard
- Anwender-SK / Beispiele
- BWO-Module / Standard
- BWO-Referenzfahrt-SK
- Cpu
- Memory map
- Symbols
- Types
- c_functions
- c_libray
- fixedlib
- functions
- Anwender-Module-2
  - new_logicode_function_4
- tasks

Link mouse button
new term inputs, e.g. C_M60-65

5.9.2 Program editor (continued)

LogiCode function edit.

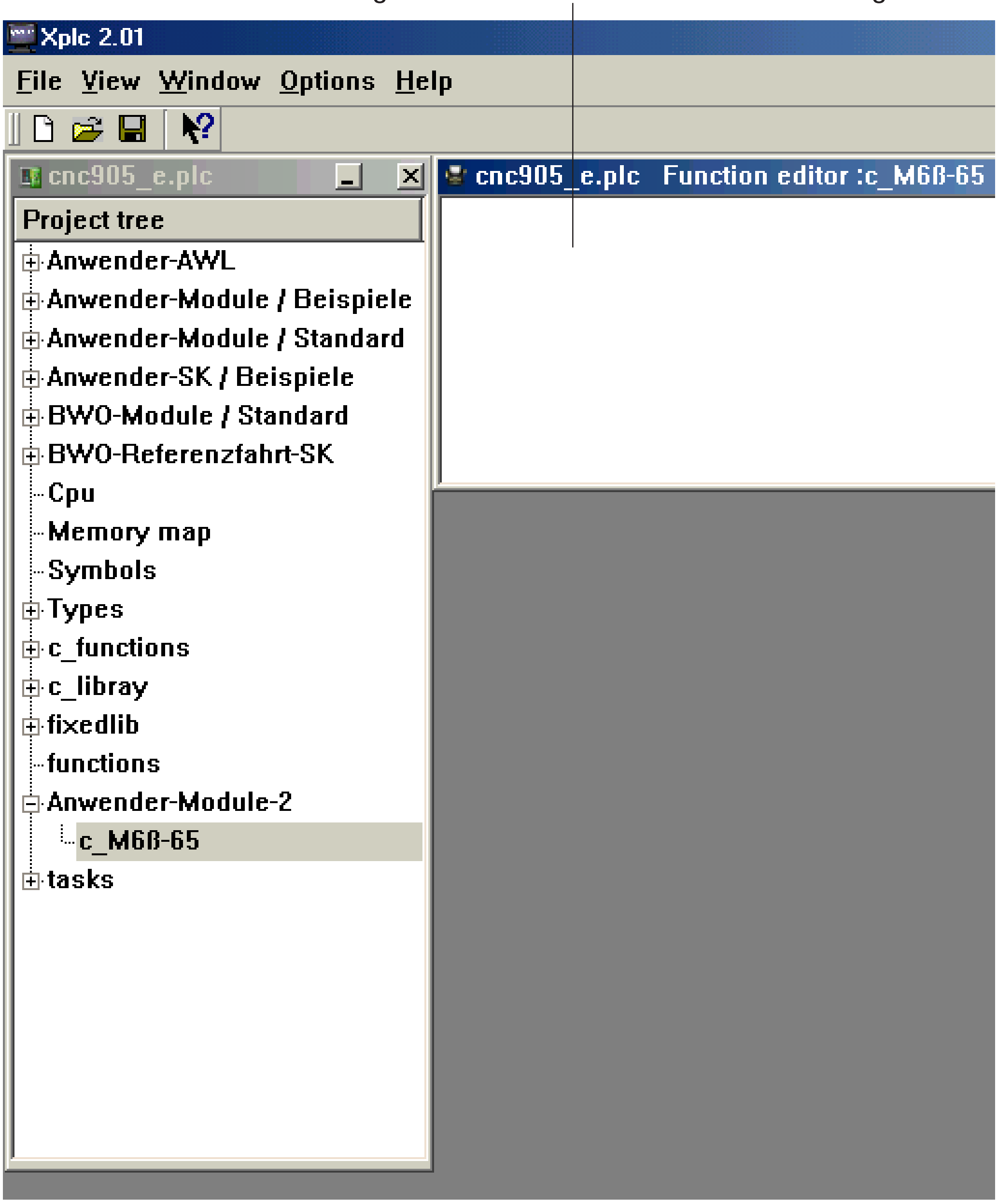
Doubleclick left mouse button opens the window the selected function (see next picture).



5.9.2 Program editor (continued)

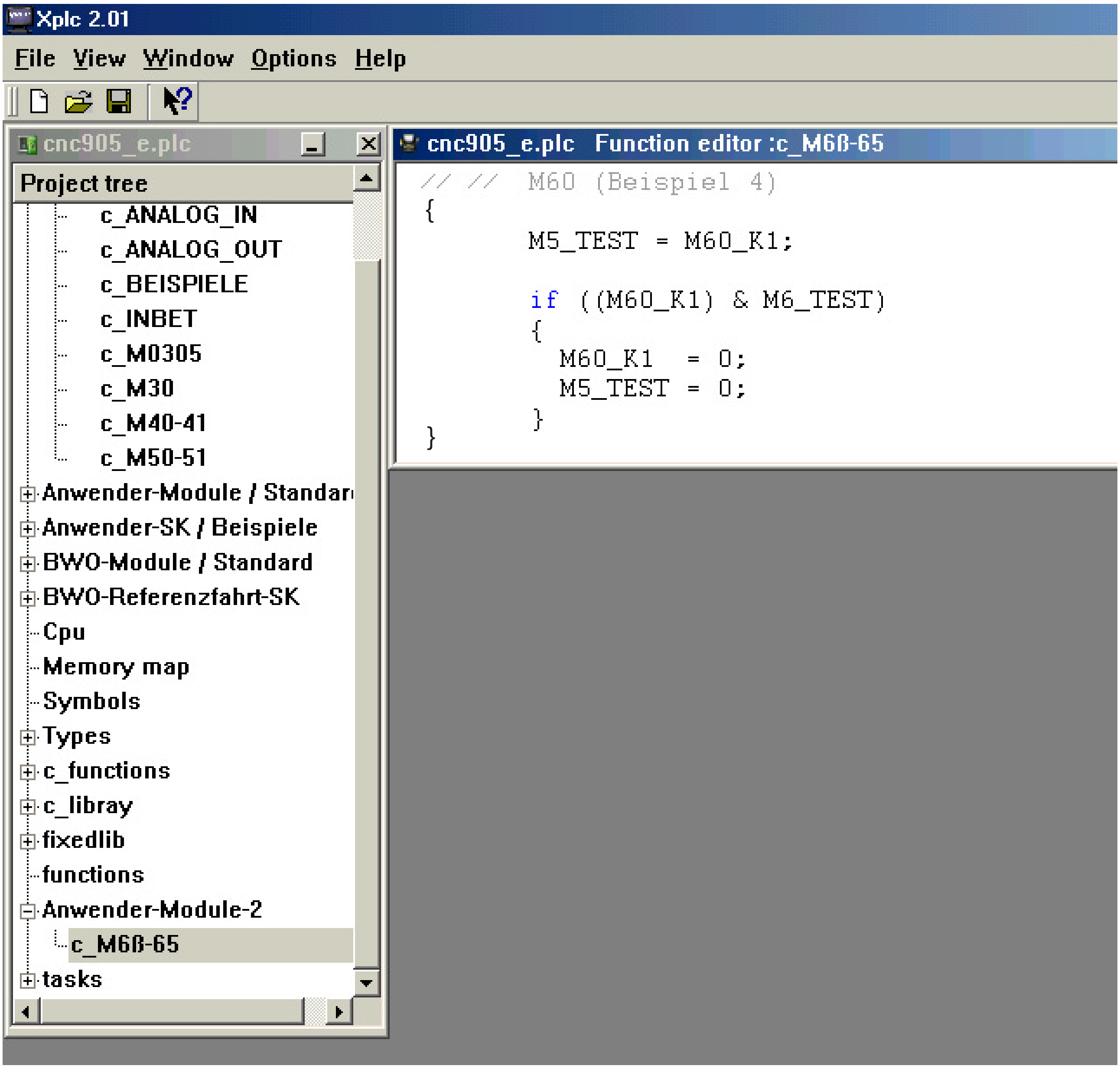
LogiCode function edit.

Window for LogiCode function which can be created again



5.9.2 Program editor (continued)

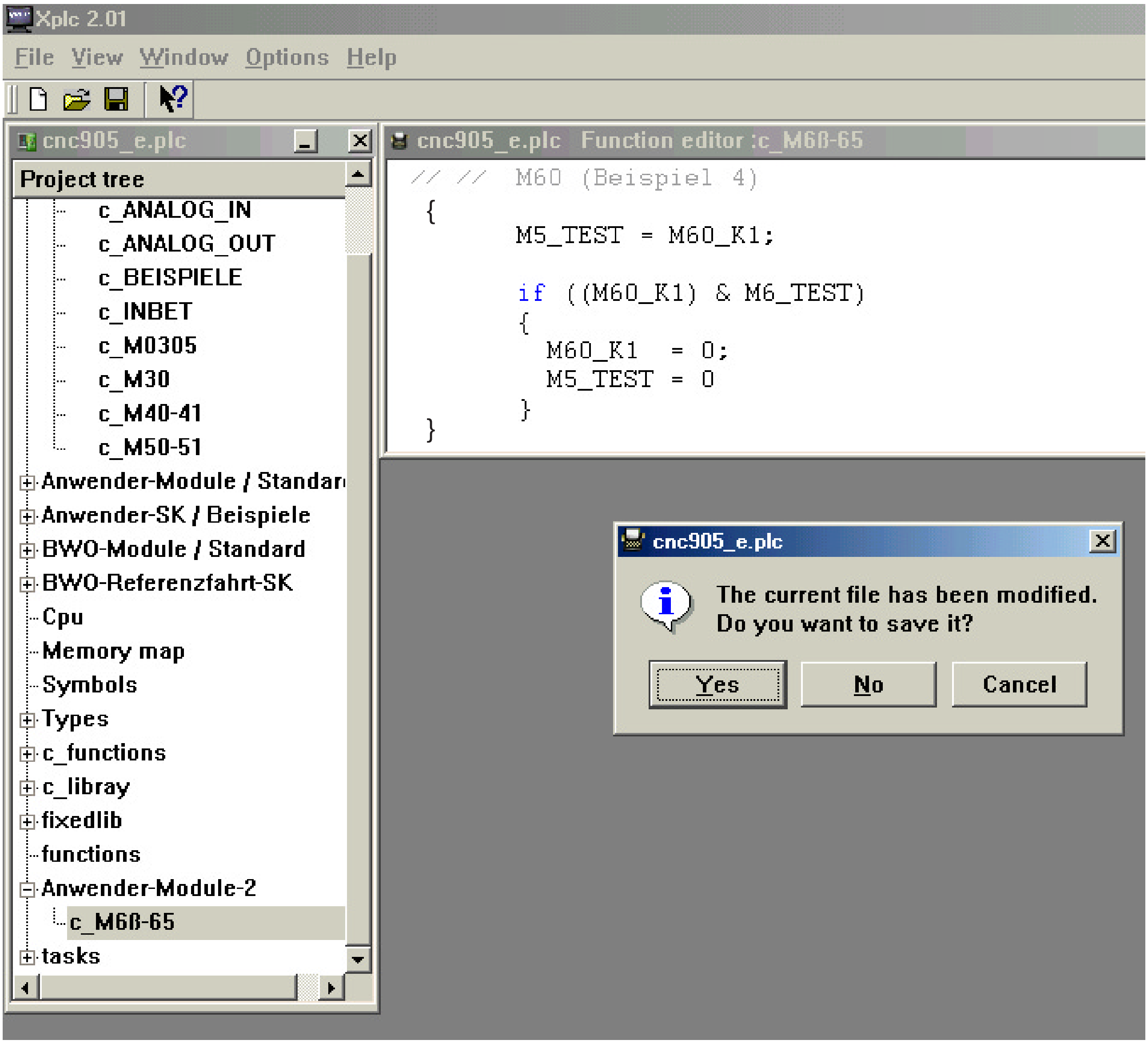
LogiCode function edit (block).



```
// //  M60 (Beispiel 4)
{
        M5_TEST = M60_K1;

        if ((M60_K1) & M6_TEST)
        {
          M60_K1  = 0;
          M5_TEST = 0;
        }
}
```

5.9.2 Program editor (continued)

Created block store.
With closing query whether edited function are stored command.



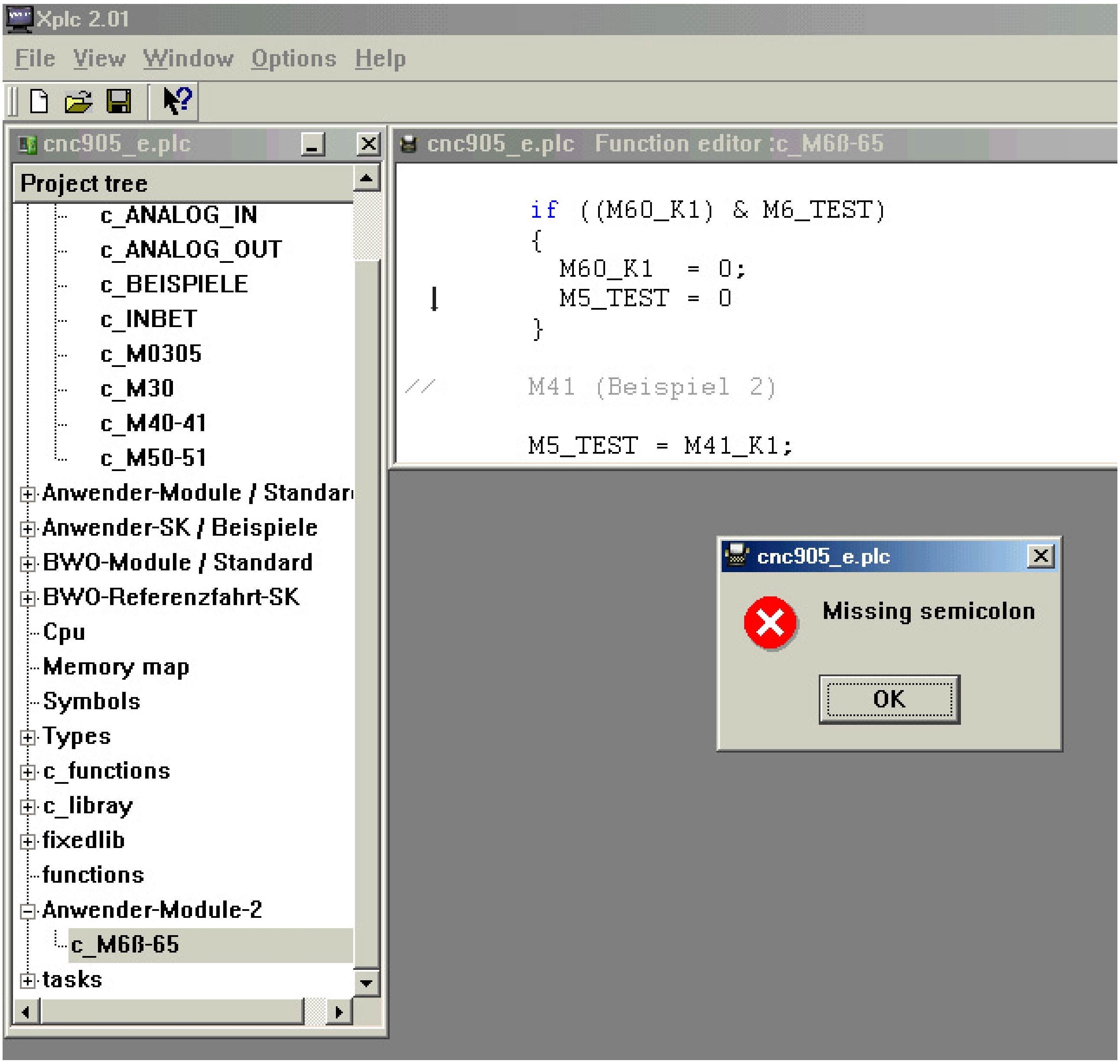
```
// //  M60 (Beispiel 4)
 {
        M5_TEST = M60_K1;

        if ((M60_K1) & M6_TEST)
        {
          M60_K1  = 0;
          M5_TEST = 0
        }
 }
```

The current file has been modified.
Do you want to save it?

Yes No Cancel

5.9.2 Program editor (continued)

Syntax error message



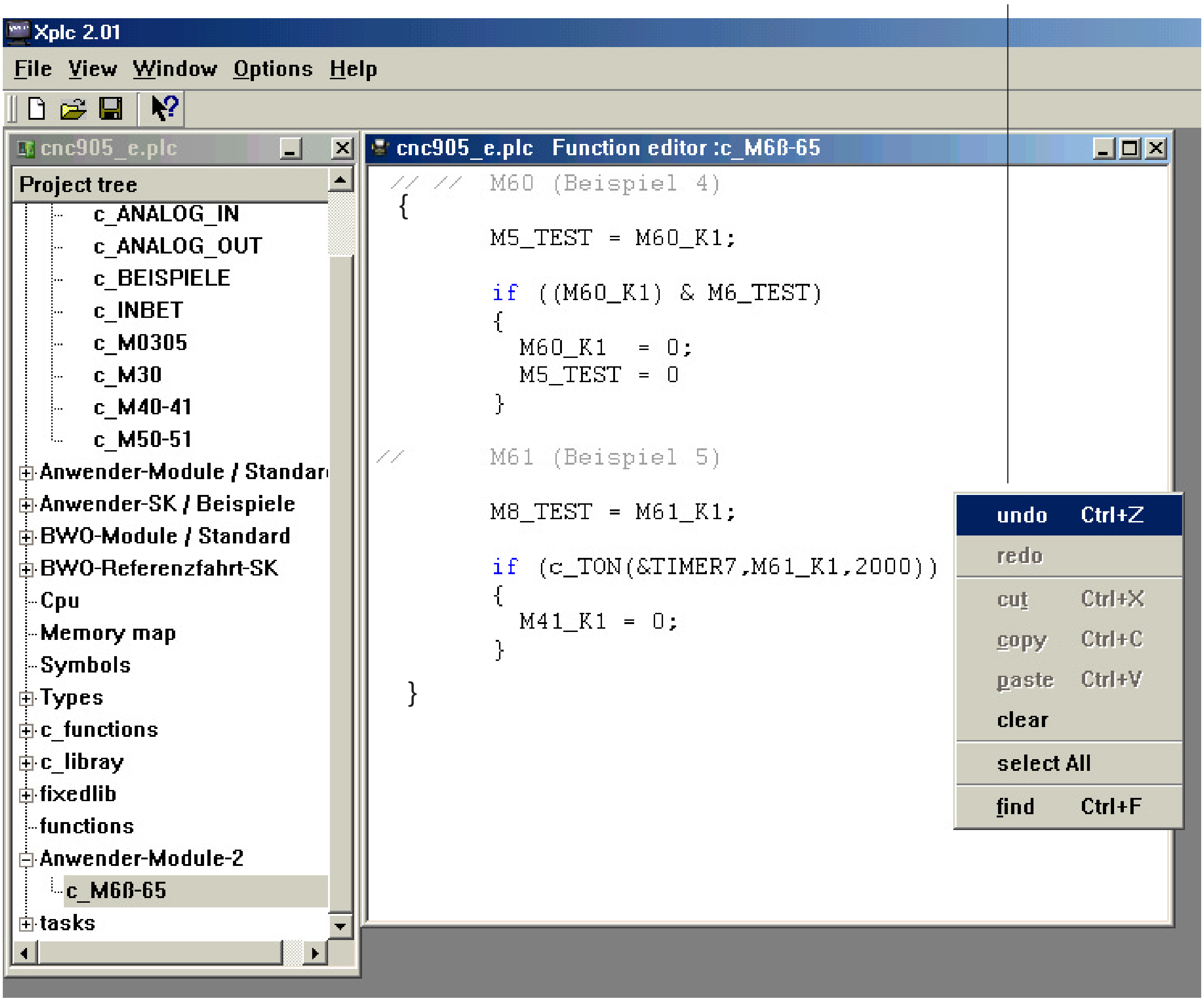
```
        if ((M60_K1) & M6_TEST)
        {
          M60_K1  = 0;
          M5_TEST = 0
        }

//      M41 (Beispiel 2)

        M5_TEST = M41_K1;
```

Missing semicolon

5.9.2 Program editor (continued)

The program editor makes the following tools available.

right mouse button

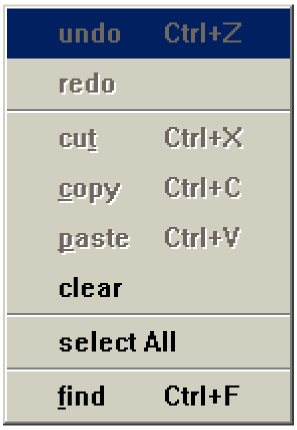


```
 // //   M60 (Beispiel 4)
 {
          M5_TEST = M60_K1;

          if ((M60_K1) & M6_TEST)
          {
            M60_K1  = 0;
            M5_TEST = 0
          }

//        M61 (Beispiel 5)

          M8_TEST = M61_K1;

          if (c_TON(&TIMER7,M61_K1,2000))
          {
            M41_K1 = 0;
          }

  }
```

Thus program sections can be transferred by a block to another.

5.9.2 Program editor (continued)

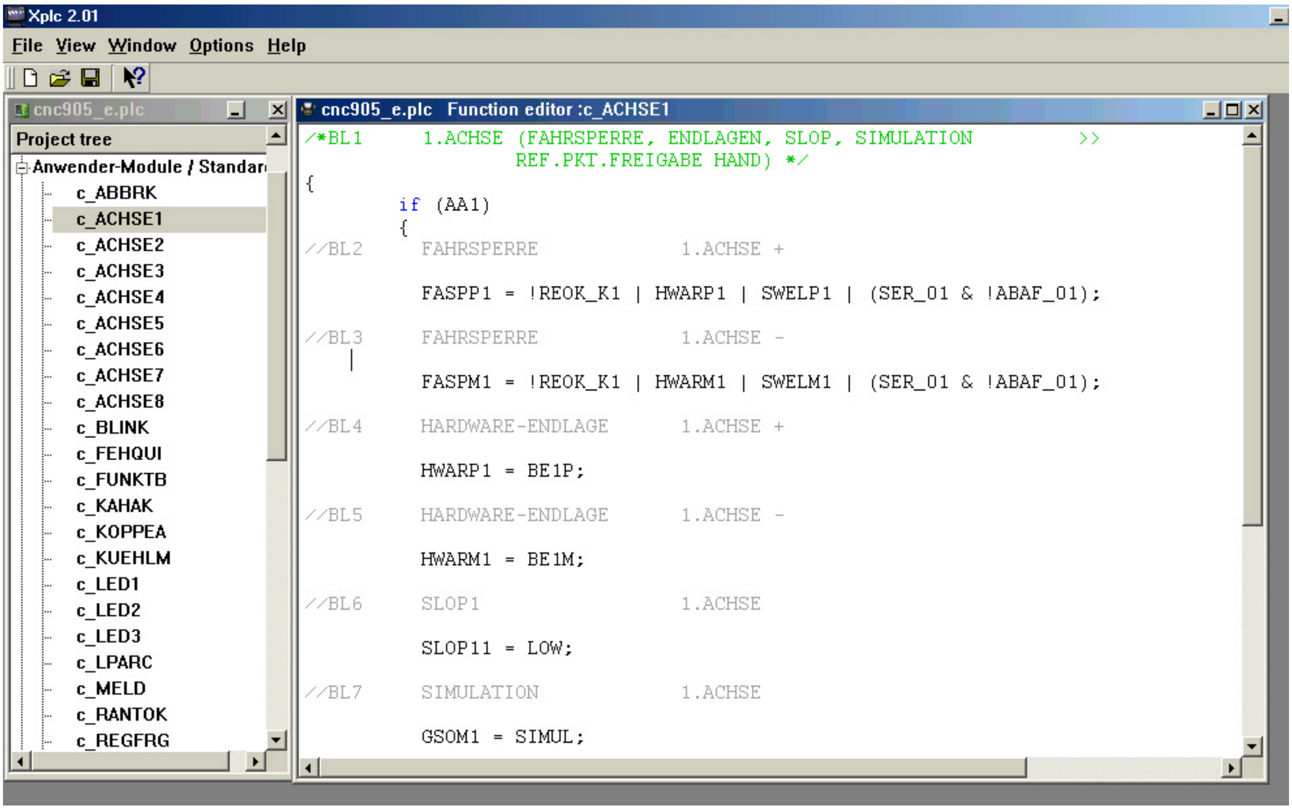
The program editor makes the following tools available.



5.9.2 Program editor (continued)

Example: User module c_ACHSE1

Doubleclick left mouse button opens windows.

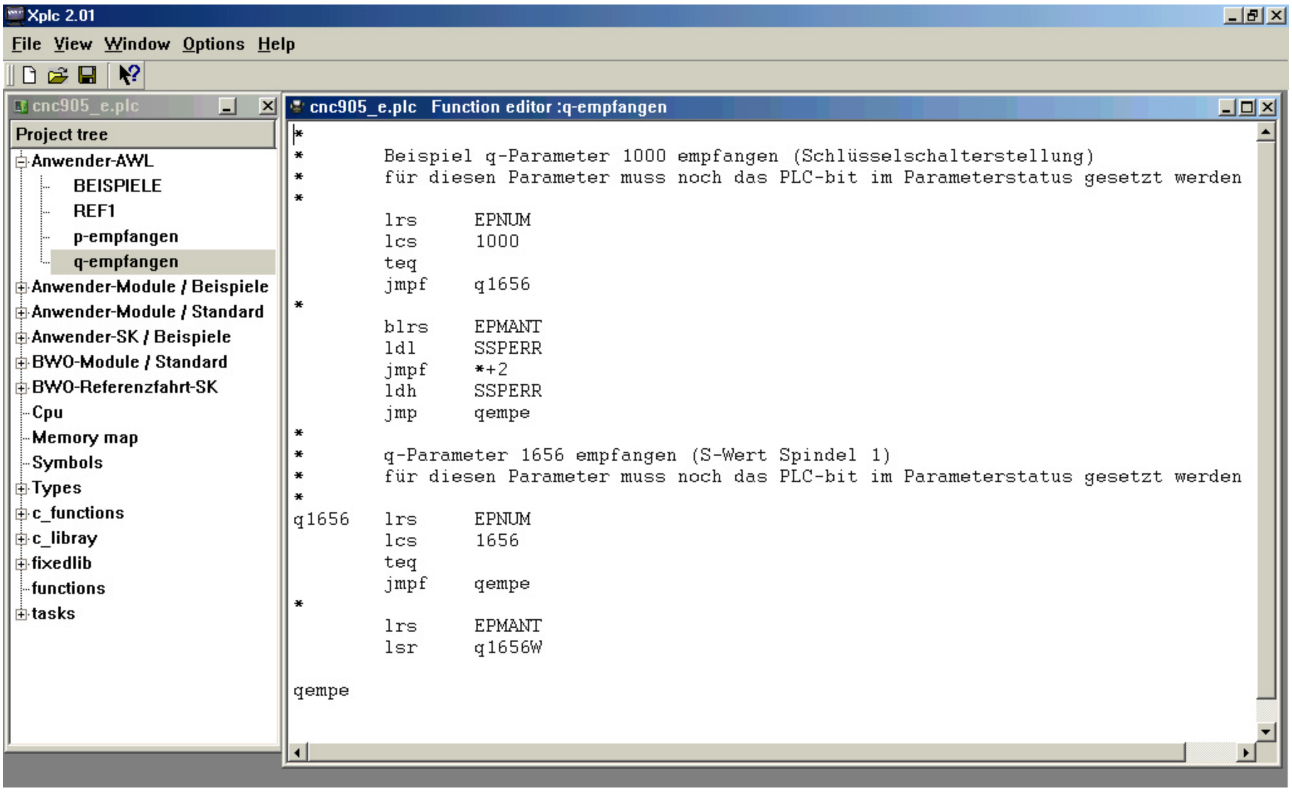


```
/*BL1      1.ACHSE (FAHRSPERRE, ENDLAGEN, SLOP, SIMULATION          >>
                   REF.PKT.FREIGABE HAND) */
{
          if (AA1)
          {
//BL2      FAHRSPERRE            1.ACHSE +

           FASPP1 = !REOK_K1 | HWARP1 | SWELP1 | (SER_01 & !ABAF_01);

//BL3      FAHRSPERRE            1.ACHSE -

           FASPM1 = !REOK_K1 | HWARM1 | SWELM1 | (SER_01 & !ABAF_01);

//BL4      HARDWARE-ENDLAGE      1.ACHSE +

           HWARP1 = BE1P;

//BL5      HARDWARE-ENDLAGE      1.ACHSE -

           HWARM1 = BE1M;

//BL6      SLOP1                 1.ACHSE

           SLOP11 = LOW;

//BL7      SIMULATION            1.ACHSE

           GSOM1 = SIMUL;
```

5.9.2 Program editor (continued)

Example: User AWL module q_empfangen

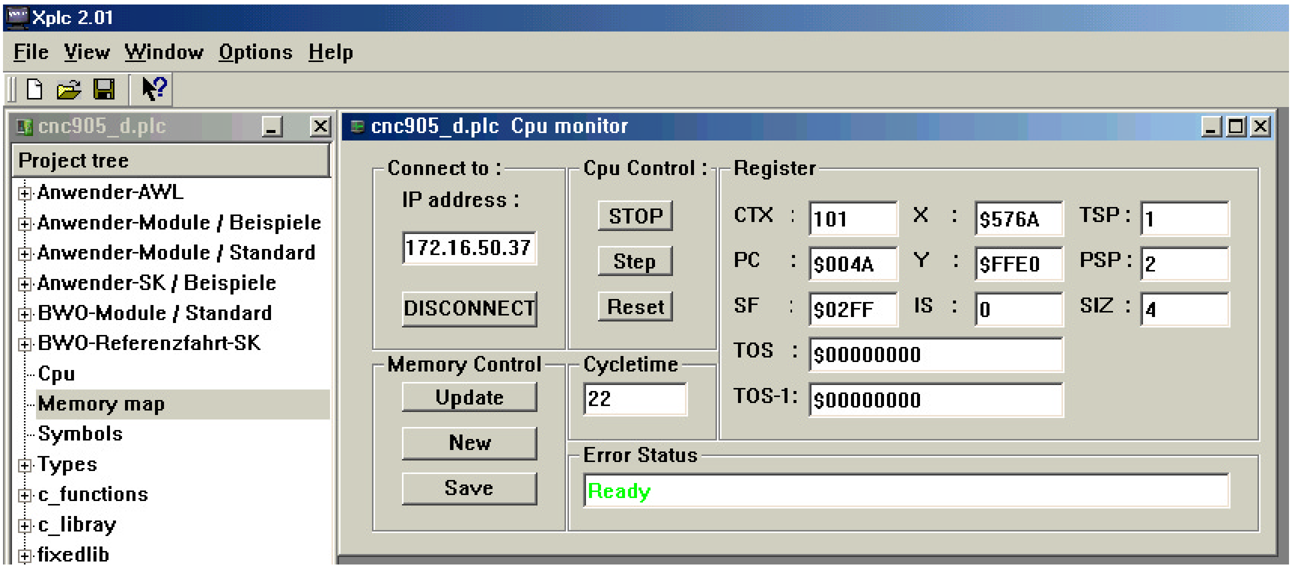
Doubleclick left mouse button opens windows.



```
*
*       Beispiel q-Parameter 1000 empfangen (Schlüsselschalterstellung)
*       für diesen Parameter muss noch das PLC-bit im Parameterstatus gesetzt werden
*
        lrs     EPNUM
        lcs     1000
        teq
        jmpf    q1656
*
        blrs    EPMANT
        ldl     SSPERR
        jmpf    *+2
        ldh     SSPERR
        jmp     qempe
*
*       q-Parameter 1656 empfangen (S-Wert Spindel 1)
*       für diesen Parameter muss noch das PLC-bit im Parameterstatus gesetzt werden
*
q1656   lrs     EPNUM
        lcs     1656
        teq
        jmpf    qempe
*
        lrs     EPMANT
        lsr     q1656W

qempe
```

5.10 On-line diagnosis

CPU monitor



IP address : 172.16.50.37

CTX : 101 X : $576A TSP : 1

PC : $004A Y : $FFE0 PSP : 2

SF : $02FF IS : 0 SIZ : 4

TOS : $00000000

TOS-1: $00000000

Cycletime 22

Error Status Ready

CTX context register, module address
PC program counter, address counter, block
SF stack frame for functionl module parameter transfer
IS internal status register for AWL
X x-index-register for AWL
Y y-index-register for AWL
TSP arithmetic stack pointer (number of register on arithmetic stack)
PSP program stack pointer (uses for UP)
SIZ amount of the adjusted word width or byte
TOS first batch item on the arithmetic stack
TOS-1 second batch item on the arithmetic stack

Memory control
Update Program in the CPU update. Compiler is started. Only those become Blocks updates, which were edited and to have also actually changed.

New one The complete program in the CPU is reset. With a following update is updated the total program (lasts therefore longer).

Save Binary program off the RAM is copied in the EEPROM.

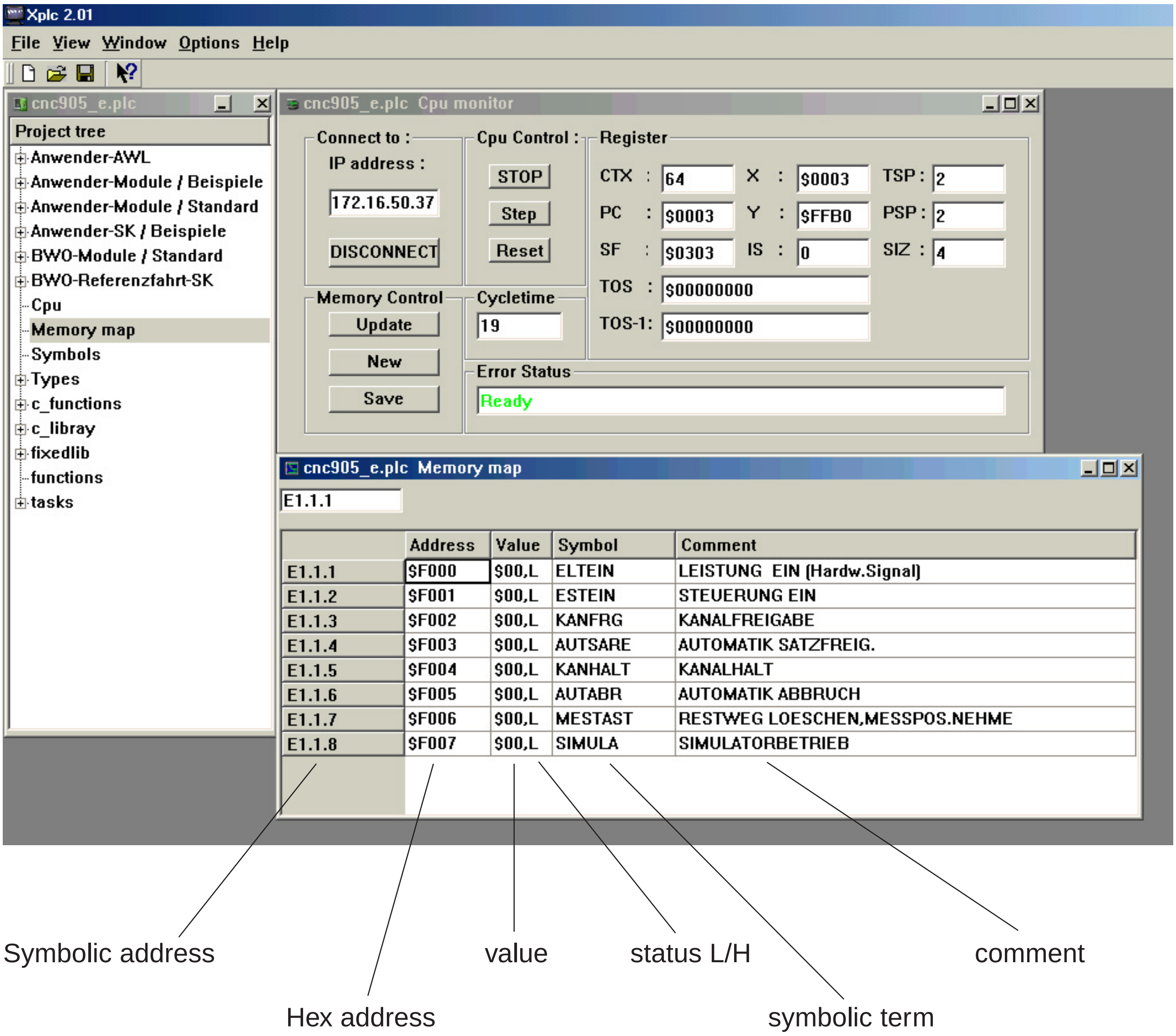
CPU control
Stop program stop
Start program start
Step single step operation
RESET sequence counters reset

5.10 On-line diagnosis (continued)

Function „ Memory map „

Status indication: **Inputs, outputs**

Doubleclick left mouse button opens windows.



IP address : 172.16.50.37

CTX : 64 X : $0003 TSP : 2
PC : $0003 Y : $FFB0 PSP : 2
SF : $0303 IS : 0 SIZ : 4
TOS : $00000000
TOS-1: $00000000

Cycletime: 19

Error Status: Ready

E1.1.1

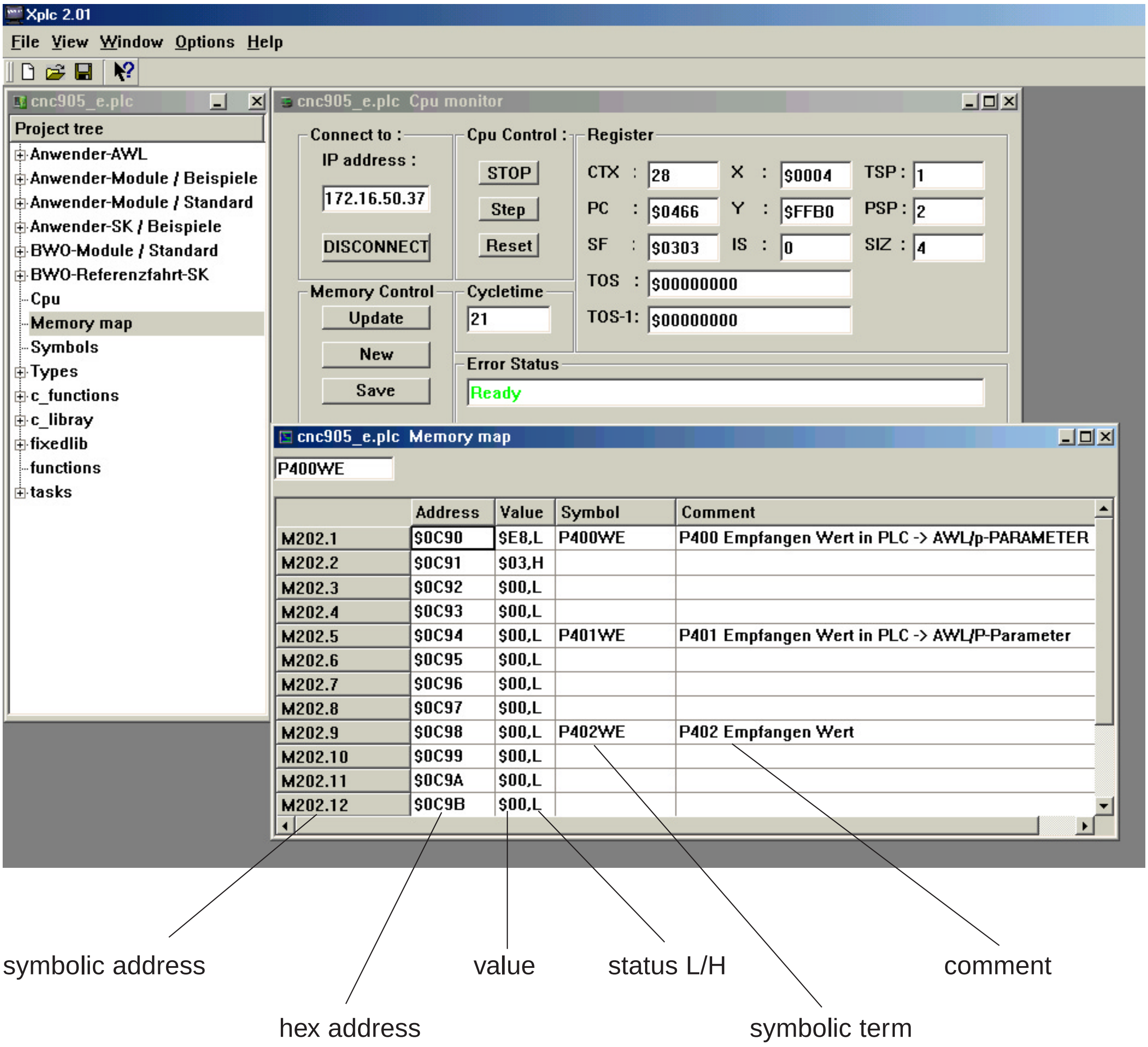
	Address	Value	Symbol	Comment
E1.1.1	$F000	$00,L	ELTEIN	LEISTUNG EIN (Hardw.Signal)
E1.1.2	$F001	$00,L	ESTEIN	STEUERUNG EIN
E1.1.3	$F002	$00,L	KANFRG	KANALFREIGABE
E1.1.4	$F003	$00,L	AUTSARE	AUTOMATIK SATZFREIG.
E1.1.5	$F004	$00,L	KANHALT	KANALHALT
E1.1.6	$F005	$00,L	AUTABR	AUTOMATIK ABBRUCH
E1.1.7	$F006	$00,L	MESTAST	RESTWEG LOESCHEN,MESSPOS.NEHME
E1.1.8	$F007	$00,L	SIMULA	SIMULATORBETRIEB

5.10 On-line diagnosis (continued)

Function „ Memory map „

Status indication: **Flag, register**

Doubleclick left mouse button opens windows.



IP address : 172.16.50.37

CTX :	28	X :	$0004	TSP :	1
PC :	$0466	Y :	$FFB0	PSP :	2
SF :	$0303	IS :	0	SIZ :	4
TOS :	$00000000				
TOS-1:	$00000000				

Cycletime: 21

Error Status: Ready

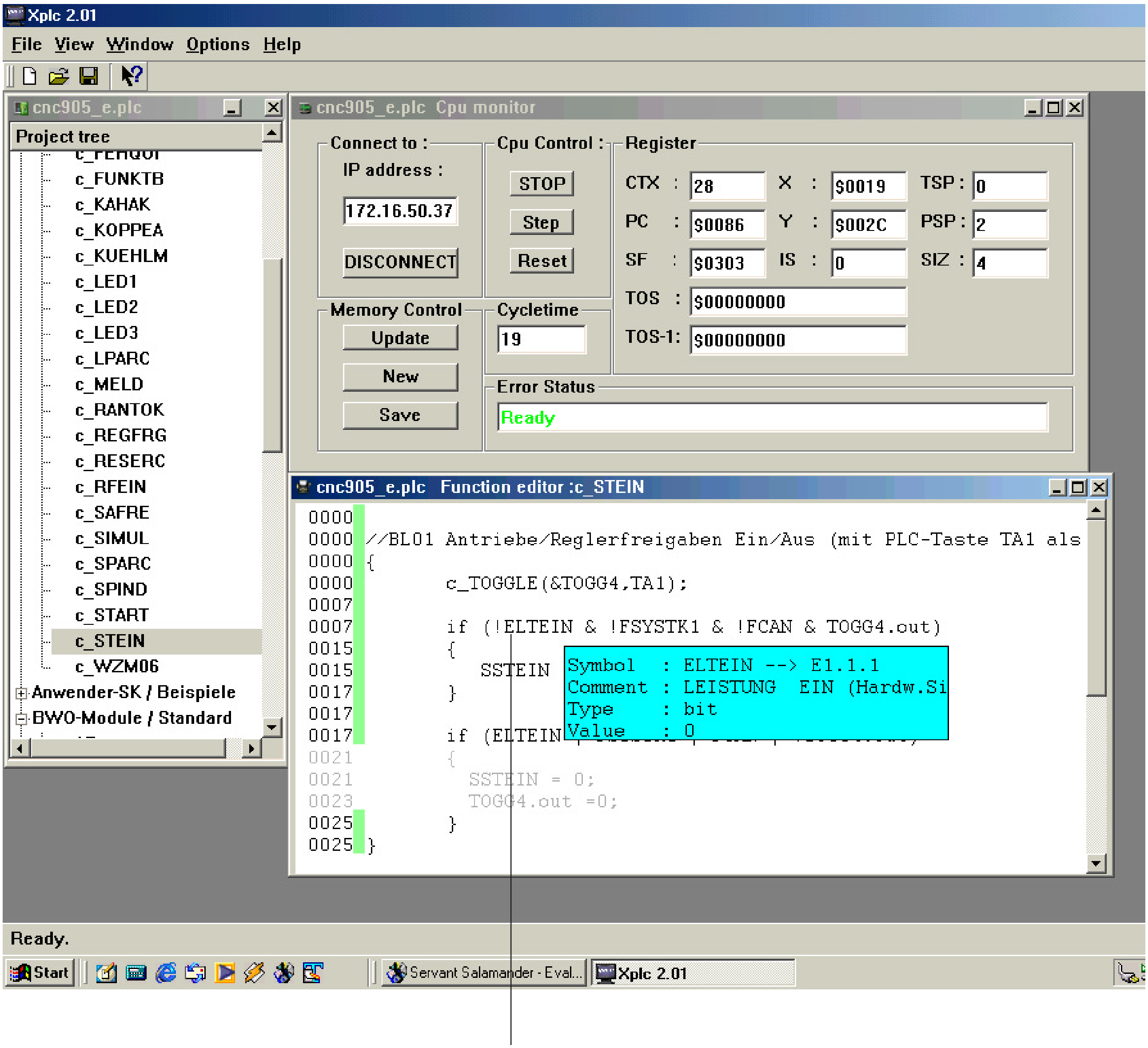
P400WE

	Address	Value	Symbol	Comment
M202.1	$0C90	$E8,L	P400WE	P400 Empfangen Wert in PLC -> AWL/p-PARAMETER
M202.2	$0C91	$03,H		
M202.3	$0C92	$00,L		
M202.4	$0C93	$00,L		
M202.5	$0C94	$00,L	P401WE	P401 Empfangen Wert in PLC -> AWL/P-Parameter
M202.6	$0C95	$00,L		
M202.7	$0C96	$00,L		
M202.8	$0C97	$00,L		
M202.9	$0C98	$00,L	P402WE	P402 Empfangen Wert
M202.10	$0C99	$00,L		
M202.11	$0C9A	$00,L		
M202.12	$0C9B	$00,L		

5.10 On-line diagnosis (continued)

Status indication in **LogiCode block**
I/O items, flag, counter, words

Doubleclick left mouse button opens windows of the selected block.



c_FEHQUI
c_FUNKTB
c_KAHAK
c_KOPPEA
c_KUEHLM
c_LED1
c_LED2
c_LED3
c_LPARC
c_MELD
c_RANTOK
c_REGFRG
c_RESERC
c_RFEIN
c_SAFRE
c_SIMUL
c_SPARC
c_SPIND
c_START
c_STEIN
c_WZM06
Anwender-SK / Beispiele
BWO-Module / Standard

IP address : 172.16.50.37

CTX : 28 X : $0019 TSP : 0
PC : $0086 Y : $002C PSP : 2
SF : $0303 IS : 0 SIZ : 4
TOS : $00000000
TOS-1: $00000000

Cycletime: 19

Error Status: Ready

cnc905_e.plc Function editor :c_STEIN

```
0000
0000 //BL01 Antriebe/Reglerfreigaben Ein/Aus (mit PLC-Taste TA1 als
0000 {
0000        c_TOGGLE(&TOGG4,TA1);
0007
0007        if (!ELTEIN & !FSYSTK1 & !FCAN & TOGG4.out)
0015        {
0015           SSTEIN
0017        }
0017
0017        if (ELTEIN
0021        {
0021          SSTEIN = 0;
0023          TOGG4.out =0;
0025        }
0025 }
```

```
Symbol   : ELTEIN --> E1.1.1
Comment  : LEISTUNG  EIN (Hardw.Si
Type     : bit
Value    : 0
```

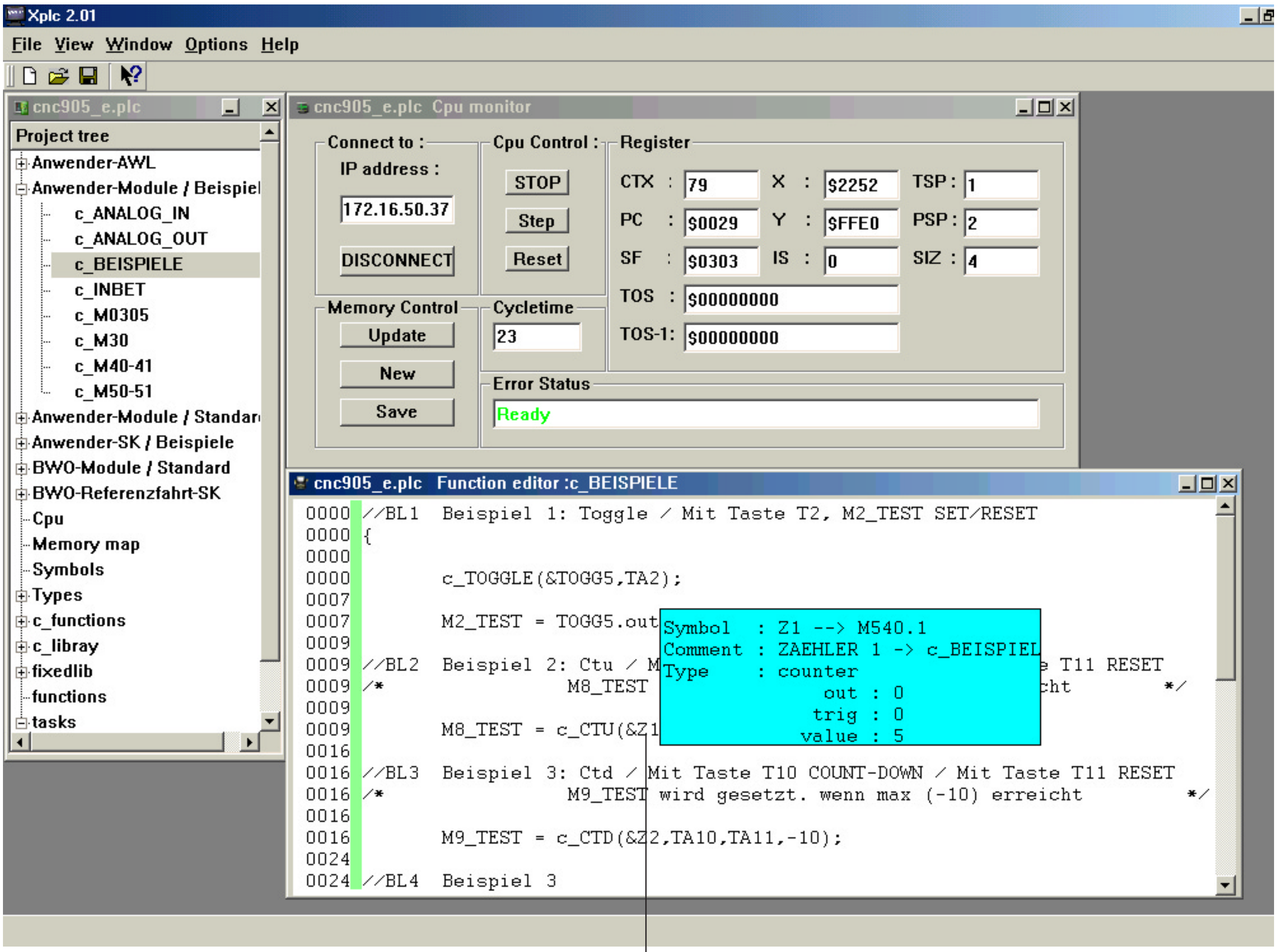
Ready.

Mouse pointers on the symbolic terms position (status window opens), e.g. status of input E.1.1.1

5.10 On-line diagnosis (continued)

Status indication in **LogiCode block**
I/O items, flag, counter, words

Doubleclick left mouse button opens windows of the selected block.



IP address : 172.16.50.37

CTX	:	79	X	:	$2252	TSP :	1
PC	:	$0029	Y	:	$FFE0	PSP :	2
SF	:	$0303	IS	:	0	SIZ :	4
TOS	:	$00000000					
TOS-1:		$00000000					

Cycletime: 23

Error Status: Ready

```
0000 //BL1   Beispiel 1: Toggle / Mit Taste T2, M2_TEST SET/RESET
0000 {
0000
0000         c_TOGGLE(&TOGG5,TA2);
0007
0007         M2_TEST = TOGG5.out
0009
0009 //BL2   Beispiel 2: Ctu / M                                      e T11 RESET
0009 /*                 M8_TEST                                       cht          */
0009
0009         M8_TEST = c_CTU(&Z1
0016
0016 //BL3   Beispiel 3: Ctd / Mit Taste T10 COUNT-DOWN / Mit Taste T11 RESET
0016 /*                 M9_TEST wird gesetzt. wenn max (-10) erreicht          */
0016
0016         M9_TEST = c_CTD(&Z2,TA10,TA11,-10);
0024
0024 //BL4   Beispiel 3
```

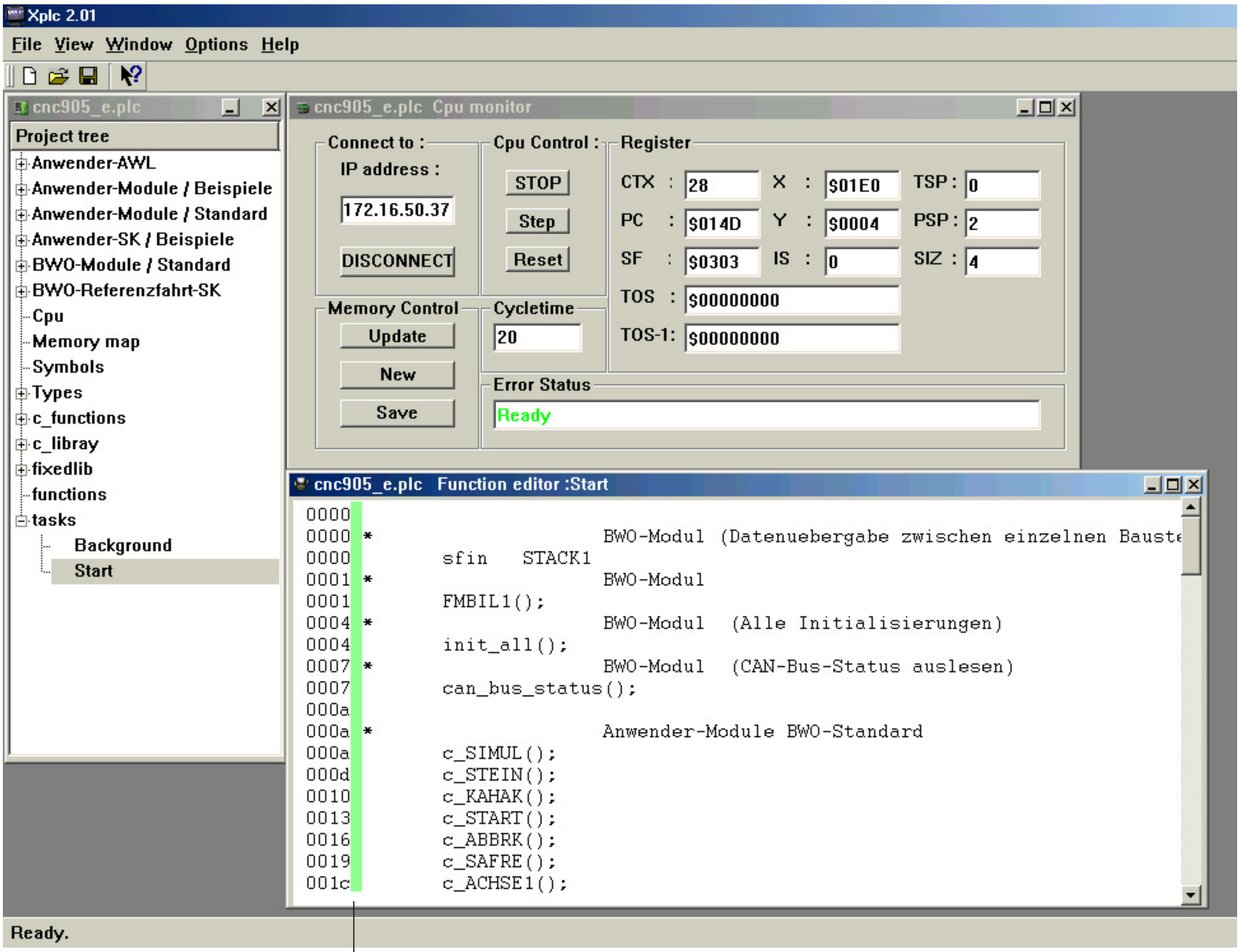
```
Symbol   : Z1 --> M540.1
Comment  : ZAEHLER 1 -> c_BEISPIEL
Type     : counter
              out  : 0
             trig  : 0
            value  : 5
```

Mouse pointers on the symbolic terms position (status window opens), e.g. status counter Z1.

5.10 On-line diagnosis (continued)

Status indication in **LogiCode block**
program track

Doubleclick left mouse button opens windows.



IP address : 172.16.50.37

CTX	:	28	X	:	$01E0	TSP :	0
PC	:	$014D	Y	:	$0004	PSP :	2
SF	:	$0303	IS	:	0	SIZ :	4
TOS	:	$00000000					
TOS-1:		$00000000					

Cycletime: 20

Error Status: Ready

```
0000
0000 *                  BWO-Modul (Datenuebergabe zwischen einzelnen Bauste
0000       sfin   STACK1
0001 *                  BWO-Modul
0001       FMBIL1();
0004 *                  BWO-Modul  (Alle Initialisierungen)
0004       init_all();
0007 *                  BWO-Modul  (CAN-Bus-Status auslesen)
0007       can_bus_status();
000a
000a *                  Anwender-Module BWO-Standard
000a       c_SIMUL();
000d       c_STEIN();
0010       c_KAHAK();
0013       c_START();
0016       c_ABBRK();
0019       c_SAFRE();
001c       c_ACHSE1();
```

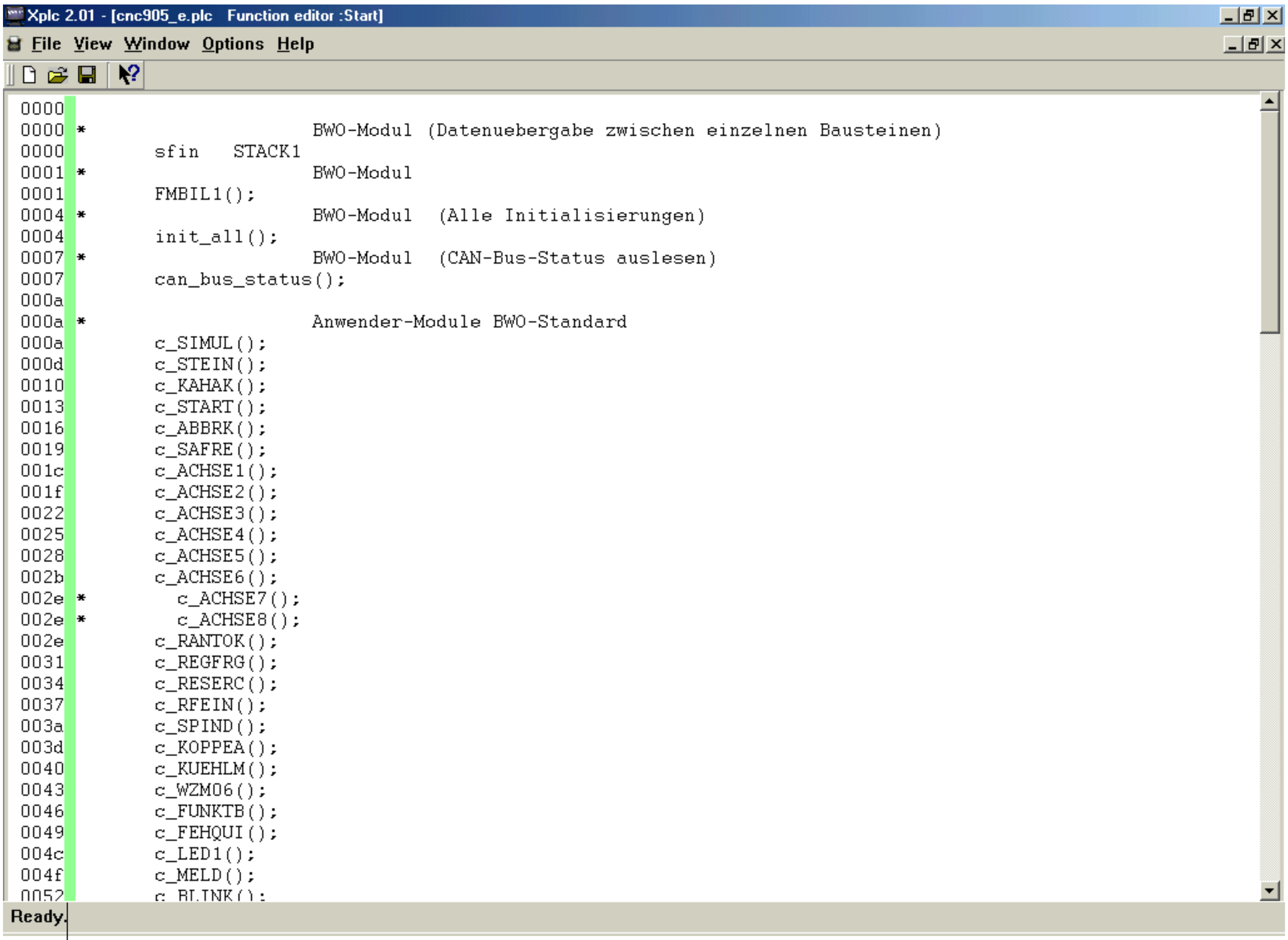
Program track display (green beam)

“ run mode “ actively: Functions are passed through.

5.10 On-line diagnosis (continued)

Status indication in **LogiCode block**
program track

Doubleclick left mouse button opens windows.


Xplc 2.01 - [cnc905_e.plc Function editor :Start]
File View Window Options Help
Ready.

```
0000
0000 *                    BWO-Modul  (Datenuebergabe zwischen einzelnen Bausteinen)
0000       sfin    STACK1
0001 *                    BWO-Modul
0001       FMBIL1();
0004 *                    BWO-Modul   (Alle Initialisierungen)
0004       init_all();
0007 *                    BWO-Modul   (CAN-Bus-Status auslesen)
0007       can_bus_status();
000a
000a *                    Anwender-Module BWO-Standard
000a       c_SIMUL();
000d       c_STEIN();
0010       c_KAHAK();
0013       c_START();
0016       c_ABBRK();
0019       c_SAFRE();
001c       c_ACHSE1();
001f       c_ACHSE2();
0022       c_ACHSE3();
0025       c_ACHSE4();
0028       c_ACHSE5();
002b       c_ACHSE6();
002e *       c_ACHSE7();
002e *       c_ACHSE8();
002e       c_RANTOK();
0031       c_REGFRG();
0034       c_RESERC();
0037       c_RFEIN();
003a       c_SPIND();
003d       c_KOPPEA();
0040       c_KUEHLM();
0043       c_WZM06();
0046       c_FUNKTB();
0049       c_FEHQUI();
004c       c_LED1();
004f       c_MELD();
0052       c_BLINK();
```

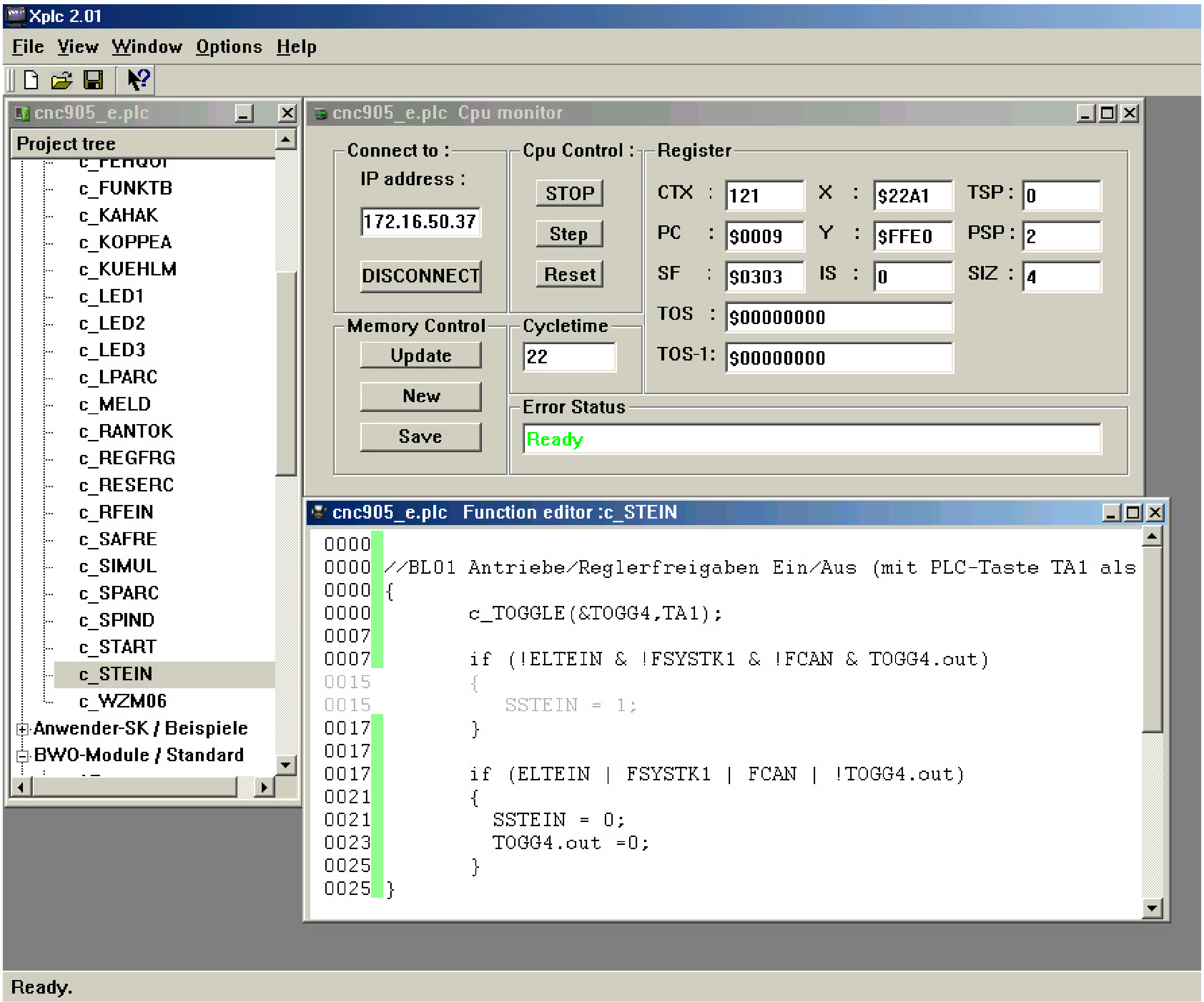
Program track display (green beam)

“ run mode “ actively: Functions are passed through.

5.10 On-line diagnosis (continued)

Status indication in **LogiCode block**
program track

Doubleclick left mouse button opens windows.



IP address : 172.16.50.37

Register

CTX :	121	X :	$22A1	TSP :	0
PC :	$0009	Y :	$FFE0	PSP :	2
SF :	$0303	IS :	0	SIZ :	4
TOS :	$00000000				
TOS-1:	$00000000				

Cycletime: 22

Error Status: Ready

```
0000
0000 //BL01 Antriebe/Reglerfreigaben Ein/Aus (mit PLC-Taste TA1 als
0000 {
0000         c_TOGGLE(&TOGG4,TA1);
0007
0007         if (!ELTEIN & !FSYSTK1 & !FCAN & TOGG4.out)
0015         {
0015            SSTEIN = 1;
0017         }
0017
0017         if (ELTEIN | FSYSTK1 | FCAN | !TOGG4.out)
0021         {
0021           SSTEIN = 0;
0023           TOGG4.out =0;
0025         }
0025 }
```

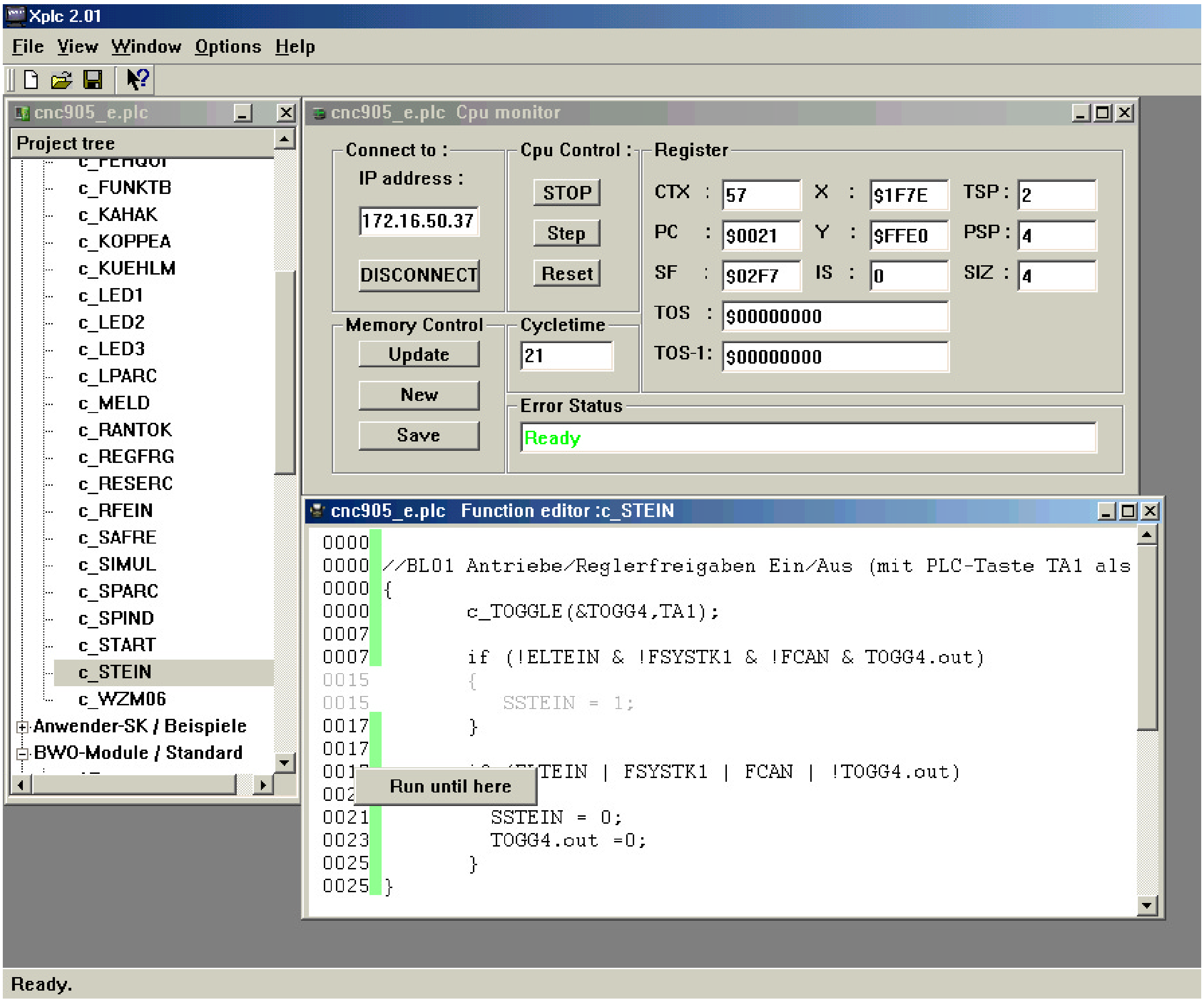
with track Instruction are active and become duchlaufen. Writing more strongly deposits.

without track Instruction are not actively, there condition efuellt actual. Writing more weakly deposits.

5.10 On-line diagnosis (continued)

Single step: Selection, up to which address the program run command.

Doubleclick left mouse button opens windows of block.



IP address : 172.16.50.37

Register

CTX :	57	X :	$1F7E	TSP :	2
PC :	$0021	Y :	$FFE0	PSP :	4
SF :	$02F7	IS :	0	SIZ :	4
TOS :	$00000000				
TOS-1:	$00000000				

Cycletime: 21

Error Status: Ready

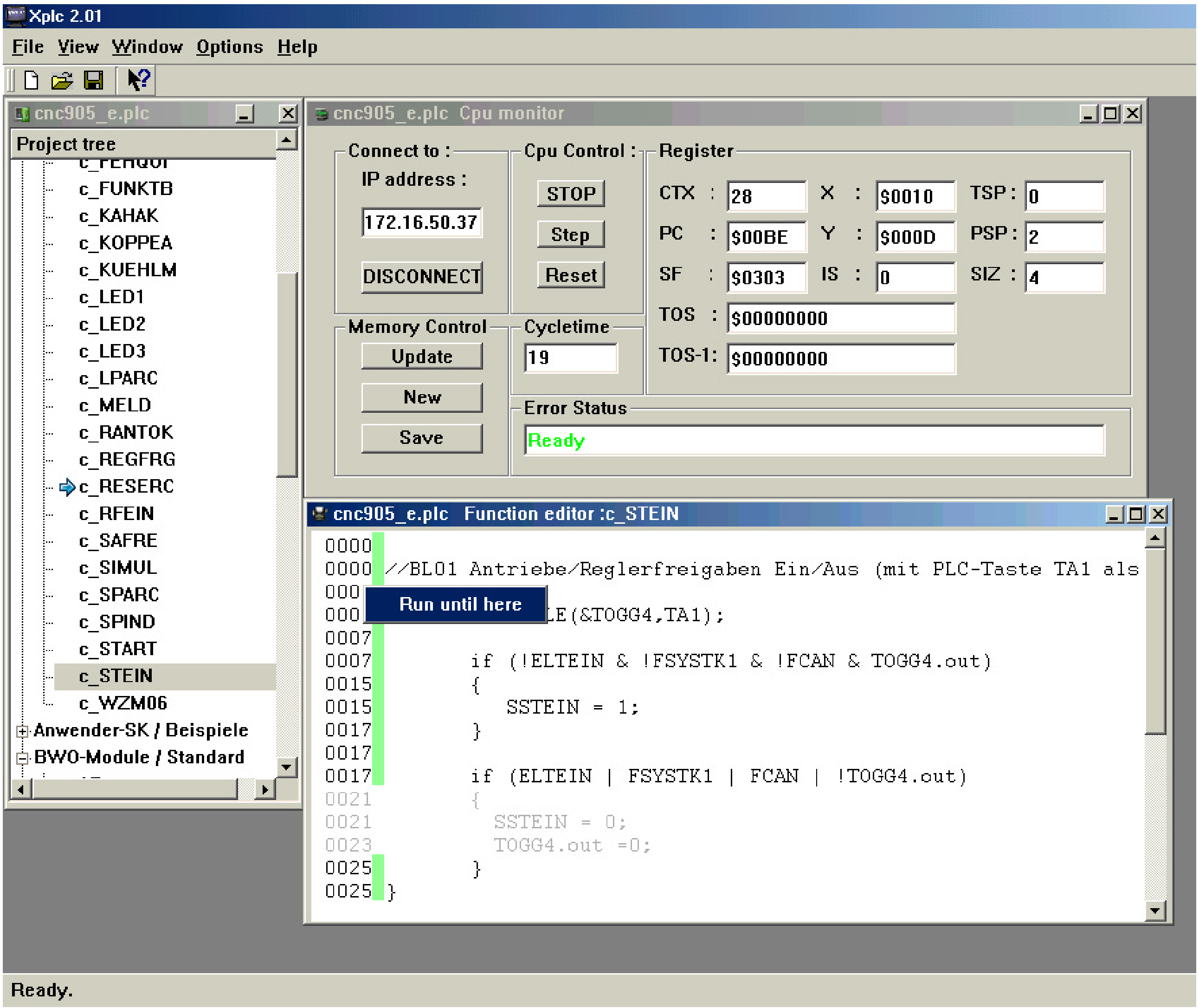
```
0000
0000 //BL01 Antriebe/Reglerfreigaben Ein/Aus (mit PLC-Taste TA1 als
0000 {
0000         c_TOGGLE(&TOGG4,TA1);
0007
0007         if (!ELTEIN & !FSYSTK1 & !FCAN & TOGG4.out)
0015         {
0015            SSTEIN = 1;
0017         }
0017
001            TEIN | FSYSTK1 | FCAN | !TOGG4.out)
002
0021           SSTEIN = 0;
0023           TOGG4.out =0;
0025         }
0025 }
```

Mouse pointers on block address counter position.
Right mouse button opens windows: “ run until here “

5.10 On-line diagnosis (continued)

Single step: Selection, up to which address the program run command.

The program stops in this address.



IP address : 172.16.50.37

Cycletime: 19

CTX : 28 X : $0010 TSP : 0

PC : $00BE Y : $000D PSP : 2

SF : $0303 IS : 0 SIZ : 4

TOS : $00000000

TOS-1: $00000000

Error Status: Ready

```
0000
0000 //BL01 Antriebe/Reglerfreigaben Ein/Aus (mit PLC-Taste TA1 als
000
000       LE(&TOGG4,TA1);
0007
0007         if (!ELTEIN & !FSYSTK1 & !FCAN & TOGG4.out)
0015         {
0015           SSTEIN = 1;
0017         }
0017
0017         if (ELTEIN | FSYSTK1 | FCAN | !TOGG4.out)
0021         {
0021           SSTEIN = 0;
0023           TOGG4.out =0;
0025         }
0025 }
```

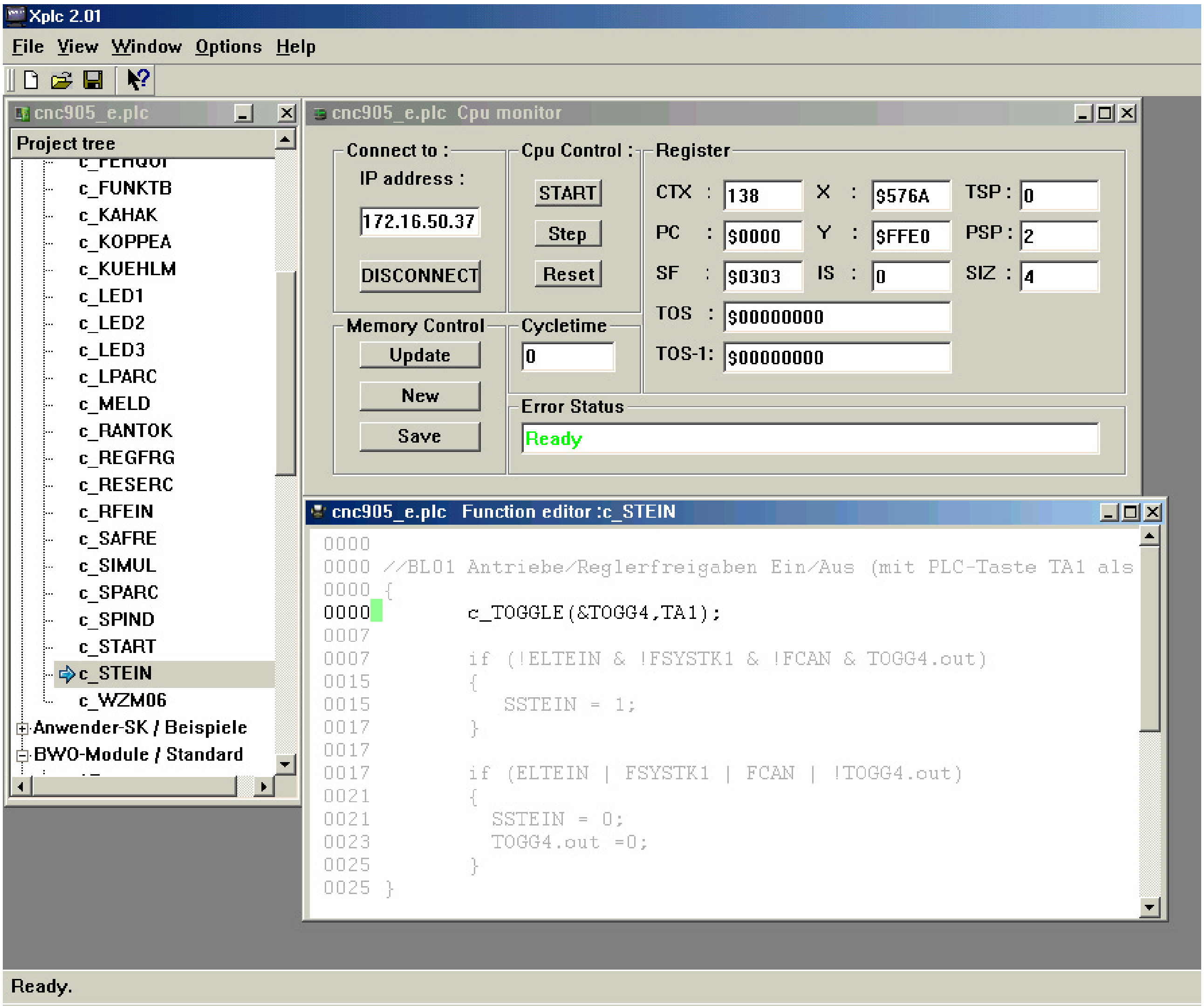
Ready.

Left mouse button stops program in the selected address.

5.10 On-line diagnosis (continued)

Single step: Selection, up to which address the program run command.

The program stops in this address.



IP address : 172.16.50.37

Register

CTX	: 138	X	: $576A	TSP :	0
PC	: $0000	Y	: $FFE0	PSP :	2
SF	: $0303	IS	: 0	SIZ :	4
TOS	: $00000000				
TOS-1:	$00000000				

Cycletime: 0

Error Status: Ready

```
0000
0000 //BL01 Antriebe/Reglerfreigaben Ein/Aus (mit PLC-Taste TA1 als
0000 {
0000        c_TOGGLE(&TOGG4,TA1);
0007
0007        if (!ELTEIN & !FSYSTK1 & !FCAN & TOGG4.out)
0015        {
0015           SSTEIN = 1;
0017        }
0017
0017        if (ELTEIN | FSYSTK1 | FCAN | !TOGG4.out)
0021        {
0021          SSTEIN = 0;
0023          TOGG4.out =0;
0025        }
0025 }
```

Ready.

Program stop at the selected address. Writing more strongly deposits.

Further in a single step with the space key.

Far in the “ run mode “ with key start.