

Program example as STL-source

```
ORGANIZATION_BLOCK "CYC_INT5"
TITLE = "Cyclic Interrupt"
VERSION : 0.0

VAR_TEMP
  OB35_EV_CLASS : BYTE ; //Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
  OB35_STRT_INF : BYTE ; //16#36 (OB 35 has started)
  OB35_PRIORITY : BYTE ; //11 (Priority of 1 is lowest)
  OB35_OB_NUMBR : BYTE ; //35 (Organization block 35, OB35)
  OB35_RESERVED_1 : BYTE ; //Reserved for system
  OB35_RESERVED_2 : BYTE ; //Reserved for system
  OB35_PHASE_OFFSET : WORD ; //Phase offset (msec)
  OB35_RESERVED_3 : INT ; //Reserved for system
  OB35_EXC_FREQ : INT ; //Frequency of execution (msec)
  OB35_DATE_TIME : DATE_AND_TIME ; //Date and time OB35 started
END_VAR

BEGIN
NETWORK
TITLE =Inputdriver

  CALL "EncoderCPU314C" , "DB_Enc314C" (
    Axis := "db_axis".Ax,
    Init := "db_axis".Ax.Init.I0);

//----< Call up an inputdriver for further axis >-----

NETWORK
TITLE =Blocks for your application
//call up here the EMC-blocks you need for your application.

  CALL "Example" , "DB_Example" ;

NETWORK
TITLE =Output
//Drive-enable signal (output A 124.0) and analog signal for the drive if
//EMC correctly works
  U "DB_MoveControl".DriveEnabled;
  = A 124.0;

//Outputdriver
  CALL "OutputCPU314C" , "DB_Out314C" (
    EnableDrive := "DB_MoveControl".DriveEnabled,
    Axis := "db_axis".Ax,
    Init := "db_axis".Ax.Init.I9);

//----< Call up an outputdriver for further axis >-----

  BE ;

END_ORGANIZATION_BLOCK
```