

1. Přílohy

Kódy a proměnné:

DUTs:

```
TYPE DUT_I_MyAxis :  
STRUCT  
    Enable: BOOL;  
    Positive: BOOL;  
    Negativ: BOOL;  
    JogForward: BOOL;  
    JogBackWard: BOOL;  
END_STRUCT  
END_TYPE
```

GVLs:

```
{attribute 'qualified_only'}  
VAR_GLOBAL  
    ax1: AXIS_REF;  
    ax2: AXIS_REF;  
    ax3: AXIS_REF;  
END_VAR
```

Proměnné MAIN:

```
PROGRAM MAIN
```

VAR

```
Ax1: DUT_I_MyAxis;  
Ax2: DUT_I_MyAxis;  
Ax3: DUT_I_MyAxis;  
Manual : BOOL;  
Automat : BOOL;  
Lamp1 AT %I*: BOOL;  
Lamp2 AT %I*: BOOL;  
Lamp3 AT %I*: BOOL;  
Lamp1N : BOOL;  
Lamp2N : BOOL;  
Lamp3N : BOOL;  
Homovani : BOOL;  
Pozice1 : BOOL;  
Pozice2 : BOOL;  
Pozice3 : BOOL;  
DelkaText : STRING;  
DelkaWord: INT;  
DelkaWord2: INT;  
DelkaWord3: INT;  
Reset: BOOL;  
RychlostNum : INT;  
RychlostAktualniNum: INT;  
RychlostZmenit: BOOL;  
Velocity1: INT;  
Velocity2: INT;  
Velocity3: INT;  
AktivovatOsuX: BOOL;  
AktivovatOsuY: BOOL;  
AktivovatOsuZ: BOOL;
```

```
fb_MC_Power: MC_Power;  
fb_MC_Jog: MC_Jog;  
fb_MC_Power2: MC_Power;  
fb_MC_Jog2: MC_Jog;  
fb_MC_Power3: MC_Power;  
fb_MC_Jog3: MC_Jog;  
fb_MC_MoveAbsolute: MC_MoveAbsolute;  
fb_MC_MoveAbsolute2: MC_MoveAbsolute;  
fb_MC_MoveAbsolute3: MC_MoveAbsolute;  
fb_MC_Reset1: MC_Reset;  
fb_MC_Reset2: MC_Reset;  
fb_MC_Reset3: MC_Reset;
```

```
ZpracujKod: BOOL;  
fbSimpleNciSequence: FB_SimpleNciSequence;  
bExecSimpleNci: BOOL := FALSE;  
in_stItpToPlc AT %I*: NcToPlc_NciChannel_Ref;  
out_stPlcToItp AT %Q*: PLCTONC_NCICHANNEL_REF;  
bUserEnableAxes: BOOL := TRUE;  
fUserOverride: LREAL := 100.0;
```

```
END_VAR
```

Program MAIN:

```
IF AktivovatOsuX THEN
    Main.Ax1.Enable := TRUE;
    Main.Ax1.Negativ := TRUE;
    Main.Ax1.Positive := TRUE;
END_IF;
```

```
IF AktivovatOsuY THEN
    Main.Ax2.Enable := TRUE;
    Main.Ax2.Negativ := TRUE;
    Main.Ax2.Positive := TRUE;
END_IF;
```

```
IF AktivovatOsuZ THEN
    Main.Ax3.Enable := TRUE;
    Main.Ax3.Negativ := TRUE;
    Main.Ax3.Positive := TRUE;
END_IF;
```

```
IF RychlostZmenit THEN
    fbSimpleNciSequence.iAxisOvr := INT_TO_UDINT(RychlostNum)*10000;
    RychlostZmenit := FALSE;
END_IF;
RychlostAktualniNum := UDINT_TO_INT(fbSimpleNciSequence.oAxisOvr/10000);
```

```
Lamp1N := NOT Lamp1;
Lamp2N := NOT Lamp2;
Lamp3N := NOT Lamp3;
```

```
IF manual THEN
    fb_MC_Power.Enable := ax1.Enable;
```

```
fb_MC_Power.Enable_Positive := ax1.Positive;  
fb_MC_Power.Enable_Negative := ax1.Negative;  
fb_MC_Jog.JogForward := ax1.JogForward;  
fb_MC_Jog.JogBackwards := ax1.JogBackWard;  
fb_MC_MoveAbsolute.Position := DelkaWord;  
fb_MC_MoveAbsolute.Execute := Pozice1;
```

```
fb_MC_Power2.Enable := ax2.Enable;  
fb_MC_Power2.Enable_Positive := ax2.Positive;  
fb_MC_Power2.Enable_Negative := ax2.Negative;  
fb_MC_Jog2.JogForward := ax2.JogForward;  
fb_MC_Jog2.JogBackwards := ax2.JogBackWard;
```

```
fb_MC_Power3.Enable := ax3.Enable;  
fb_MC_Power3.Enable_Positive := ax3.Positive;  
fb_MC_Power3.Enable_Negative := ax3.Negative;  
fb_MC_Jog3.JogForward := ax3.JogForward;  
fb_MC_Jog3.JogBackwards := ax3.JogBackWard;
```

ELSIF automat THEN

```
fb_MC_Power.Enable := TRUE;  
fb_MC_Power2.Enable := TRUE;  
fb_MC_Power3.Enable := TRUE;
```

```
fb_MC_Power.Enable_Positive := TRUE;  
fb_MC_Power2.Enable_Positive := TRUE;  
fb_MC_Power3.Enable_Positive := TRUE;
```

```
fb_MC_Power.Enable_Negative := TRUE;  
fb_MC_Power2.Enable_Negative := TRUE;  
fb_MC_Power3.Enable_Negative := TRUE;
```

```

IF Homovani THEN
    fb_MC_Jog3.JogForward := NOT lamp3n;
    IF lamp3n THEN
        fb_MC_Jog.JogBackwards := NOT lamp1n;
        fb_MC_Jog2.JogBackwards := NOT lamp2n;
    ELSE
        fb_MC_Jog.JogBackwards := FALSE;
        fb_MC_Jog2.JogBackwards := FALSE;
    END_IF

    IF lamp1n AND lamp2n AND lamp3n THEN
        Homovani := FALSE;
    END_IF
END_IF

ELSE
    fb_MC_Power.Enable := FALSE;
    fb_MC_Power.Enable_Positive := FALSE;
    fb_MC_Power.Enable_Negative := FALSE;
    fb_MC_Jog.JogForward := FALSE;
    fb_MC_Jog.JogBackwards := FALSE;
    ax1.Enable := FALSE;
    ax1.Positive := FALSE;
    ax1.Negativ := FALSE;
    ax1.JogForward := FALSE;
    ax1.JogBackWard := FALSE;

    fb_MC_Power2.Enable := FALSE;
    fb_MC_Power2.Enable_Positive := FALSE;
    fb_MC_Power2.Enable_Negative := FALSE;
    fb_MC_Jog2.JogForward := FALSE;

```

```
fb_MC_Jog2.JogBackwards := FALSE;  
ax2.Enable := FALSE;  
ax2.Positive := FALSE;  
ax2.Negativ := FALSE;  
ax2.JogForward := FALSE;  
ax2.JogBackWard := FALSE;
```

```
fb_MC_Power3.Enable := FALSE;  
fb_MC_Power3.Enable_Positive := FALSE;  
fb_MC_Power3.Enable_Negative := FALSE;  
fb_MC_Jog3.JogForward := FALSE;  
fb_MC_Jog3.JogBackwards := FALSE;  
ax3.Enable := FALSE;  
ax3.Positive := FALSE;  
ax3.Negativ := FALSE;  
ax3.JogForward := FALSE;  
ax3.JogBackWard := FALSE;
```

```
END_IF
```

```
IF reset THEN
```

```
fb_MC_Reset1.Execute := TRUE;  
fb_MC_Reset2.Execute := TRUE;  
fb_MC_Reset3.Execute := TRUE;  
reset := false;
```

```
ELSE
```

```
fb_MC_Reset1.Execute := FALSE;  
fb_MC_Reset2.Execute := FALSE;  
fb_MC_Reset3.Execute := FALSE;
```

```
END_IF
```

```
fb_MC_Power(  

```

```
Axis:= gvl.ax1,  
Override:= ,  
BufferMode:= ,  
Options:= ,  
Status=> ,  
Busy=> ,  
Active=> ,  
Error=> ,  
ErrorID=> );
```

```
fb_MC_Jog(  
Axis:= gvl.ax1,  
Mode:= ,  
Position:= ,  
Velocity:=100 ,  
Acceleration:= ,  
Deceleration:= ,  
Jerk:= ,  
Done=> ,  
Busy=> ,  
Active=> ,  
CommandAborted=> ,  
Error=> ,  
ErrorID=> );
```

```
fb_MC_Power2(  
Axis:= gvl.ax2,  
Override:= ,  
BufferMode:= ,  
Options:= ,  
Status=> ,
```



```
Busy=> ,  
Active=> ,  
Error=> ,  
ErrorID=> );
```

```
fb_MC_Jog2(  
    Axis:= gvl.ax2,  
    Mode:= ,  
    Position:= ,  
    Velocity:= fbSimpleNciSequence.oAxisOvr,  
    Acceleration:= ,  
    Deceleration:= ,  
    Jerk:= ,  
    Done=> ,  
    Busy=> ,  
    Active=> ,  
    CommandAborted=> ,  
    Error=> ,  
    ErrorID=> );
```

```
fb_MC_Power3(  
    Axis:= gvl.ax3,  
    Override:= ,  
    BufferMode:= ,  
    Options:= ,  
    Status=> ,  
    Busy=> ,  
    Active=> ,  
    Error=> ,
```

```
ErrorID=> );
```

```
fb_MC_Jog3(  
  Axis:= gvl.ax3,  
  Mode:= ,  
  Position:= ,  
  Velocity:= fbSimpleNciSequence.oAxisOvr ,  
  Acceleration:= ,  
  Deceleration:= ,  
  Jerk:= ,  
  Done=> ,  
  Busy=> ,  
  Active=> ,  
  CommandAborted=> ,  
  Error=> ,  
  ErrorID=> );
```

```
fb_MC_MoveAbsolute(  
  Axis:= gvl.ax1 ,  
  Velocity:= fbSimpleNciSequence.oAxisOvr,  
  Acceleration:= ,  
  Deceleration:= ,  
  Jerk:= ,  
  BufferMode:= ,  
  Options:= ,  
  Done=> ,  
  Busy=> ,  
  Active=> ,  
  CommandAborted=> ,  
  Error=> ,  
  ErrorID=> );
```

```
fb_MC_MoveAbsolute2(  
    Axis:= gvl.ax2,  
    Velocity:= fbSimpleNciSequence.oAxisOvr,  
    Acceleration:= ,  
    Deceleration:= ,  
    Jerk:= ,  
    BufferMode:= ,  
    Options:= ,  
    Done=> ,  
    Busy=> ,  
    Active=> ,  
    CommandAborted=> ,  
    Error=> ,  
    ErrorID=> );
```

```
fb_MC_MoveAbsolute3(  
    Axis:= gvl.ax3 ,  
    Velocity:= fbSimpleNciSequence.oAxisOvr,  
    Acceleration:= ,  
    Deceleration:= ,  
    Jerk:= ,  
    BufferMode:= ,  
    Options:= ,  
    Done=> ,  
    Busy=> ,  
    Active=> ,  
  
    CommandAborted=> ,  
    Error=> ,  
    ErrorID=> );
```

```
//ItpSetOverridePercent(fOverridePercent:=fUserOverride , sPlcToNci:=out_stPlcToItp);
```

```
fbSimpleNciSequence(  
  bReset := reset,  
  bExecute:= ZpracujKod,  
  bBusy=> ,  
  bError=> ,  
  bDone=> ,  
  stX:=gvl.ax1,  
  stY:=gvl.ax2,  
  stZ:=gvl.ax3,  
  stItpToPlc:=in_stItpToPlc ,  
  stPlcToItp:= out_stPlcToItp);
```

```
fb_MC_Reset1(  
  Axis:=gvl.ax1 ,  
  Done=> ,  
  Busy=> ,  
  Error=> ,  
  ErrorID=> );
```

```
fb_MC_Reset2(  
  Axis:=gvl.ax2 ,  
  Done=> ,  
  Busy=> ,  
  Error=> ,  
  ErrorID=> );
```

```
fb_MC_Reset3(  
  Axis:=gvl.ax3 ,
```

Done=> ,
Busy=> ,
Error=> ,
ErrorID=>);

Proměnné FB_SimpleNCISequence:

FUNCTION_BLOCK FB_SimpleNciSequence

VAR_INPUT

bExecute : BOOL;
bReset : BOOL;
iAxisOvr : UDINT;

END_VAR

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
bDone : BOOL;
oAxisOvr : UDINT;

END_VAR

VAR_IN_OUT

stX : AXIS_REF;
stY : AXIS_REF;
stZ : AXIS_REF;
stItpToPlc : NCTOPLC_NCCHANNEL_REF;
stPlcToItp : PLCTONC_NCCHANNEL_REF;

END_VAR

VAR

nState : UDINT := 0;
nErrorState : UDINT;

fbBuildGroup : CfgBuild3DGroup;
fbLoadGCode : ItpLoadProgEx;
fbStartGCode : ItpStartStopEx;
fbClearGroup : CfgReconfigGroup;
fbConfirmHsk : ItpConfirmHsk;

nInterpreterState : UDINT := 0;
bConfirmHsk : BOOL := FALSE;

```
sPrgName          : STRING(255) := 'first.nc';  
END_VAR
```

Program FB_SimpleNciSequunce:

```
stPlcToItp.ChnAxesOvr := iAxisOvr;  
oAxisOvr := stPlcToItp.ChnAxesOvr;  
CASE nState OF  
0:  
    IF bExecute THEN  
        bBusy := TRUE;  
        bDone:= FALSE;  
        bError := FALSE;  
        nErrorState := nState;  
        nState:= 10;  
    END_IF  
    IF NOT bReset THEN  
        fbClearGroup(bExecute:=FALSE);  
    END_IF  
  
10:  
    // vytvoření kinematické skupiny  
    fbBuildGroup(  
        bExecute:=TRUE,  
        nGroupId:=ItpGetGroupId(sNciToPlc:=stItpToPlc) ,  
        nXAxisId:=stX.NcToPlc.AxisId,  
        nYAxisId:=stY.NcToPlc.AxisId,  
        nZAxisId:=stZ.NcToPlc.AxisId,  
        tTimeOut:= ,
```

```

    bBusy=> ,
    bErr=> ,
    nErrId=> );
IF NOT fbBuildGroup.bBusy THEN
    IF NOT fbBuildGroup.bErr THEN
        nState:= 20;
    ELSE
        (* add error handling *)
        nErrorState := nState;
        bBusy := FALSE;
        bError := TRUE;
        nState:= 9999;
    END_IF
    fbBuildGroup(bExecute:=FALSE);
END_IF

```

20:

```

//nahraje soubor s G-kódem
// je třeba se aby byl požadovaný NC program ve složce: C: \TwinCAT\MC\NCI-
folder

```

```

fbLoadGCode(
    sNciToPlc:=stltpToPlc,
    bExecute:=TRUE,
    sPrg:= sPrgName,
    nLength:= INT_TO_UDINT(LEN(sPrgName)),
    tTimeOut:= ,
    bBusy=> ,
    bErr=> ,
    nErrId=> );
IF NOT fbLoadGCode.bBusy THEN
    IF NOT fbLoadGCode.bErr THEN

```



```

        nState:= 30;
    ELSE
        (* add error handling *)
        nErrorState := nState;
        bBusy := FALSE;
        bError := TRUE;
        nState:= 9999;
    END_IF
    fbLoadGCode(sNciToPlc:=stltpToPlc, bExecute:=FALSE);
END_IF

```

30:

// je třeba zkontrolovat stav interpretu, který umožňuje správné vykonávání příkazů

G-kódu

```

nInterpreterState := ltpGetStateInterpreter(sNciToPlc:=stltpToPlc);
IF nInterpreterState = Tc2_NCI.NCI_INTERPRETER_READY THEN
    nState:= 40;
ELSE
    (* add error handling *)
    nErrorState := nState;
    bBusy := FALSE;
    bError := TRUE;
    nState:= 9999;
END_IF

```

40:

// Start načítání G-kódu

```

fbStartGCode(
    bStart:=TRUE,
    bStop:=FALSE,
    tTimeOut:= ,

```

```

        sNciToPlc:= stltpToPlc,
        bBusy=> ,
        bErr=> ,
        nErrId=> );
IF NOT fbStartGCode.bBusy THEN
    IF NOT fbStartGCode.bErr THEN
        nState:= 50;
    ELSE
        (* add error handling *)
        nErrorState := nState;
        bBusy := FALSE;
        bError := TRUE;
        nState:= 9999;
    END_IF
    fbStartGCode( bStart:=FALSE, sNciToPlc:= stltpToPlc );
END_IF

```

50:

```

nInterpreterState := ltpGetStateInterpreter(sNciToPlc:=stltpToPlc);
IF nInterpreterState <> Tc2_NCI.NCI_INTERPRETER_READY THEN
    // nci is running
    nState:= 60;
END_IF

```

60:

```

nInterpreterState := ltpGetStateInterpreter(sNciToPlc:=stltpToPlc);
IF nInterpreterState = Tc2_NCI.NCI_INTERPRETER_READY THEN
    // program has finished
    nState:= 70;
ELSIF nInterpreterState = Tc2_NCI.NCI_INTERPRETER_ABORTED THEN
    // a run-time error occured - this could be a lag error or something else...
    // add error handling

```

```

        ;
    END_IF
70:
    // Program skončil
    // Uknočit spojení kinematické skupiny
    fbClearGroup(
        bExecute:=TRUE,
        nGroupId:=ltpGetGroupId(sNciToPlc:=stltpToPlc) ,
        tTimeOut:= ,
        bBusy=> ,
        bErr=> ,
        nErrId=> );
    IF NOT fbClearGroup.bBusy THEN
        IF NOT fbClearGroup.bErr THEN
            nState:= 80;
        ELSE
            (* add error handling *)
            nErrorState := nState;
            bBusy := FALSE;
            bError := TRUE;
            nState:= 9999;
        END_IF
        fbClearGroup(bExecute:=FALSE);
    END_IF
80:
    bDone:= TRUE;
    bBusy := FALSE;
    IF NOT bExecute THEN
        nState:= 0;
    END_IF
9999:

```

```
// error state
IF NOT bExecute THEN
    nState:= 0;
    bError := FALSE;
END_IF
END_CASE
```

```
fbConfirmHsk(
    bExecute:=bConfirmHsk ,
    sNciToPlc:=stItpToPlc ,
    sPlcToNci:=stPlcToItp ,
    bBusy=> ,
    bErr=> ,
    nErrId=> );
```

```
IF bReset THEN
    fbClearGroup(
        bExecute:=TRUE,
        nGroupId:=ItpGetGroupId(sNciToPlc:=stItpToPlc) ,
        tTimeOut:= ,
        bBusy=> ,
        bErr=> ,
        nErrId=> );

    nState:= 0;
END_IF
```

Stavové diagramy:



