

**Programm Code** V 1.0

Library

Inhalt

[**Programm Code** V 1.0 1](#_Toc384367302)

[Library 1](#_Toc384367303)

[1 TSR Com 2](#_Toc384367304)

[1.1 Variablen Deklaration 2](#_Toc384367305)

[1.2 Main 3](#_Toc384367306)

[1.3 Aktionen 3](#_Toc384367307)

[2 FB Connect with TSR 7](#_Toc384367308)

[2.1 Variablen Deklaration 7](#_Toc384367309)

[2.2 Main 7](#_Toc384367310)

[2.3 Aktionen 7](#_Toc384367311)

[3 FB Get TSR position 9](#_Toc384367312)

[3.1 Variablen Deklaration 9](#_Toc384367313)

[3.2 Main 9](#_Toc384367314)

[3.3 Aktionen 9](#_Toc384367315)

[4 FN Set TSR Startpoint 11](#_Toc384367316)

[4.1 Variablen Deklaration 11](#_Toc384367317)

[4.2 Main 11](#_Toc384367318)

[5 FB Move TSR xy 11](#_Toc384367319)

[5.1 Variablen Deklaration 11](#_Toc384367320)

[5.2 Main 11](#_Toc384367321)

[5.3 Aktionen 12](#_Toc384367322)

[6 FB Go to TSR xy 14](#_Toc384367323)

[6.1 Variablen Deklaration 14](#_Toc384367324)

[6.2 Main 14](#_Toc384367325)

[6.3 Aktionen 14](#_Toc384367326)

[7 FB TSR Pen up down 17](#_Toc384367327)

[7.1 Variablen Deklaration 17](#_Toc384367328)

[7.2 Main 17](#_Toc384367329)

[7.3 Aktionen 17](#_Toc384367330)

# TSR Com

## Variablen Deklaration

FUNCTION\_BLOCK TSR\_Com

VAR\_INPUT

xEXecute :BOOL; //Startbedingung für den FB.

iCom :INT; //Die Nummer des Com Ports.

stSendData :STRING; //Der String der Gesendet werden soll.

END\_VAR

VAR\_OUTPUT

xBusy :BOOL; //Das Busy bit.

xDone :BOOL; //Ist 1 wenn FB fertig ist.

stReceivedData :STRING; //Der Empfangene Daten String.

wState :WORD; //Der ENUM Wert für die ENUM Tabelle.

xError :BOOL; //Ist 1 wenn es einen Error gab.

END\_VAR

VAR

LTi\_Zustand :INT; //Der Zustand.

LTi\_PosString :INT; //Die Position vom Carisch retain lengfield.

LTocpa\_ComParameter1 :ARRAY[1..20] OF COM.PARAMETER; //Initalisierung der Com Parameter.

LTx\_ComOpen :BOOL; //Gibt an ob ein Com Port schon offen ist.

LTudi\_Handle :UDINT; //Der Interne COM Handle.

LTst\_ReceivedData :STRING; //Received Data zwischenspeicher.

FB\_ComOpen :COM.Open; //FB Open.

FB\_WriteCom :Com.Write; //FB Write.

FB\_ReadCom :Com.Read; //FB Read.

FB\_Timer :TON; //Timer Init

GLRudi\_ComPort :UDINT; //Welcher Com Port ist gemeint.

END\_VAR

VAR CONSTANT

GLRudi\_StopBit :UDINT := 0; //Die Stopbits: 0 = 1stb, 1 = 1.5stb, 2 = 2stb.

GLRudi\_Parity :UDINT := 2; //Parity: 0 = gerade, 1 = ungerade, 2 = keine.

GLRudi\_Baudrate :UDINT := 9600; //Möglich sind 2,5 m 115200 Bit/s, 5 m 57600 Bit/s, 10 m 38400 Bit/s, 15 m 19200 Bit/s, 30 m 9600 Bit/s

GLRudi\_TimeOut :UDINT := 0; //Keine ahnung was das bewirkt.

GLRudi\_BufferSize :UDINT := 2048; //Die Buffer grösse.

GLRudi\_ByteSize :UDINT := 8; //Die Grösse vom daten Byte. kann nur von 4 bis 8 sein

GLRudi\_Binary :UDINT := 0; //Binärmodus, keine EOF-Überprüfung/ keine ahnung wad das ist.

GLRudi\_OutxCTSFlow :UDINT := 0; //CTS Handshake auf Ausgang/ keine ahnung wad das ist.

GLRudi\_OutxDSRFlow :UDINT := 0; //DSR Handshake auf Ausgang/ keine ahnung wad das ist.

GLRudi\_DTRControl :UDINT := 0; //DTR Control. DTR Flusssteuerung/ keine ahnung wad das ist.

GLRudi\_DSRSensitiv :UDINT := 0; //DSR Empfindlichkeit/ keine ahnung wad das ist.

GLRudi\_RTSControl :UDINT := 0; //Rts Flussteuerung/ keine ahnung wad das ist.

GLRudi\_TXContinueOnXoff :UDINT := 0; //XOFF fortfahren Tx / keine ahnung wad das ist.

GLRudi\_OUTX :UDINT := 0; //XON/XOFF Ausgabeflusssteuerung / keine ahnung wad das ist.

GLRudi\_INX :UDINT := 0; //XON/XOFF Eingabeflusssteuerung / keine ahnung wad das ist.

GLRudi\_XOnChar :UDINT := 0; //Tx AND Rx XON Zeichen / keine ahnung wad das ist.

GLRudi\_XOffChar :UDINT := 0; //Tx AND Rx XOFF Zeichen.

GLRudi\_XOnLim :UDINT := 0; //Übertrage XON Schwellenwert

GLRudi\_XOffLim :UDINT := 0; //Übertrage XOFF Schwellenwert

LTt\_WaitTimeout :TIME := T#2S; //der Timout Timer.

LTt\_WaitCom :TIME := T#250MS; //Wait auf Received Data.

END\_VAR

## Main

CASE LTi\_Zustand OF

0:

A0\_Start(); //Prüfe die Startbedingungen und Prüfe auf einen offenen Com Port

1:

A1\_Get\_Settings(); //Holt die Einstellungen für zu öffnenden Com Port.

2:

A2\_Open\_Com(); //Öffnet den Com Port.

3:

A3\_Open\_Wait(); //Wartet bis der Com Port offen ist.

4:

A4\_Send\_Com(); //Den String Senden.

5:

A5\_Send\_Wait(); //Warte bis der String gesendet wurde.

6:

A6\_Receive\_Com(); //Empfange den String.

7:

A7\_Receive\_Wait(); //Warte auf FB String lesen.

8:

A8\_String\_Auswertung(); //Werte den FB String aus.

9:

A9\_End(); //Beende den FB vorgang und warte auf das zurücksetzten.

100:

B100\_Error(); //Nur wenn ein Error gab.

END\_CASE

FB\_ComOpen();

## Aktionen

A0\_Start

IF xExecute THEN //Prüfe Startbedinung.

xBusy := 1; //Setze Busy.

IF LTx\_ComOpen THEN //Ist der Com Port offen ?

LTi\_Zustand := 4; //Gehe direkt zum Schritt wo der String geschrieben wird.

ELSE

LTi\_Zustand := 1; //Gehe direkt zum Schritt wo der Com Port geöffnet wird.

END\_IF

END\_IF

A1\_Get\_Settings

GLRudi\_ComPort := INT\_TO\_UDINT(iCom);

//Port

LTocpa\_ComParameter1[1].udiParameterId := COM.CAA\_Parameter\_Constants.udiPort; //ID von Com Port

LTocpa\_ComParameter1[1].udiValue := GLRudi\_ComPort; //Der COM Port1

//Stopbit

LTocpa\_ComParameter1[2].udiParameterId := COM.CAA\_Parameter\_Constants.udiStopBits; //ID von StopBit

LTocpa\_ComParameter1[2].udiValue := GLRudi\_StopBit; //Der COM Port. 0 = 1stb, 1 = 1.5stb, 2 = 2stb.

//Parität

LTocpa\_ComParameter1[3].udiParameterId := COM.CAA\_Parameter\_Constants.udiParity; //ID von Parity

LTocpa\_ComParameter1[3].udiValue := GLRudi\_Parity; //0 = gerade, 1 = ungerade, 2 = keine.

//Baudrate

LTocpa\_ComParameter1[4].udiParameterId := COM.CAA\_Parameter\_Constants.udiBaudrate; //ID von Baudrate

LTocpa\_ComParameter1[4].udiValue := GLRudi\_Baudrate; //Möglich sind 2,5 m 115200 Bit/s, 5 m 57600 Bit/s, 10 m 38400 Bit/s, 15 m 19200 Bit/s, 30 m 9600 Bit/s

//Timeout

LTocpa\_ComParameter1[5].udiParameterId := COM.CAA\_Parameter\_Constants.udiTimeout; //ID von Timeout

LTocpa\_ComParameter1[5].udiValue := GLRudi\_TimeOut; //Keine ahnung wad das ist.

(\*

//BufferSize

LTocpa\_ComParameter1[6].udiParameterId := COM.CAA\_Parameter\_Constants.udiBufferSize; //ID von Buffersize

LTocpa\_ComParameter1[6].udiValue := GLRudi\_BufferSize; //Die Grösse die der Buffer haben soll

\*)

//ByteSize

LTocpa\_ComParameter1[7].udiParameterId := COM.CAA\_Parameter\_Constants.udiByteSize; //ID von ByteSize

LTocpa\_ComParameter1[7].udiValue := GLRudi\_ByteSize; //Die Grösse vom daten Byte. kann nur von 4 bis 8 sein

//Binary

LTocpa\_ComParameter1[8].udiParameterId := COM.CAA\_Parameter\_Constants.udiByteSize; //ID von Binary

LTocpa\_ComParameter1[8].udiValue := GLRudi\_Binary; //Binärmodus, keine EOF-Überprüfung/ keine ahnung wad das ist.

(\*

//OutxCtsFlow

LTocpa\_ComParameter1[9].udiParameterId := COM.CAA\_Parameter\_Constants.udiOutxCtsFlow; //ID von OutxCtsFlow

LTocpa\_ComParameter1[9].udiValue := GLRudi\_OutxCTSFlow; //CTS Handshake auf Ausgang/ keine ahnung wad das ist.

//OutxDSRFlow

LTocpa\_ComParameter1[10].udiParameterId := COM.CAA\_Parameter\_Constants.udiOutxDsrFlow; //ID von OutxDSRFlow

LTocpa\_ComParameter1[10].udiValue := GLRudi\_OutxDSRFlow; //DSR Handshake auf Ausgang/ keine ahnung wad das ist.

//DTRControl

LTocpa\_ComParameter1[11].udiParameterId := COM.CAA\_Parameter\_Constants.udiDtrControl; //ID von DtrControl

LTocpa\_ComParameter1[11].udiValue := GLRudi\_OutxDSRFlow; //DSR Handshake auf Ausgang/ keine ahnung wad das ist.

//DSR Sensitiv

LTocpa\_ComParameter1[12].udiParameterId := COM.CAA\_Parameter\_Constants.udiDsrSensitivity;//ID von DSR Sensitiv

LTocpa\_ComParameter1[12].udiValue := GLRudi\_DSRSensitiv; //DSR Empfindlichkeit/ keine ahnung wad das ist.

//RTS Control

LTocpa\_ComParameter1[13].udiParameterId := COM.CAA\_Parameter\_Constants.udiRtsControl; //ID von RTS Control

LTocpa\_ComParameter1[13].udiValue := GLRudi\_RTSControl; //Rts Flusssteuerung/ keine ahnung wad das ist.

//TXContinueOnXoff

LTocpa\_ComParameter1[14].udiParameterId := COM.CAA\_Parameter\_Constants.udiTXContinueOnXoff; //ID von XOFF fortfahren Tx

LTocpa\_ComParameter1[14].udiValue := GLRudi\_TXContinueOnXoff; //XOFF fortfahren Tx / keine ahnung wad das ist.

//OUT X

LTocpa\_ComParameter1[15].udiParameterId := COM.CAA\_Parameter\_Constants.udiOutX; //ID von OUT X

LTocpa\_ComParameter1[15].udiValue := GLRudi\_OUTX; //XON/XOFF Ausgabeflusssteuerung/ keine ahnung wad das ist.

//IN X

LTocpa\_ComParameter1[16].udiParameterId := COM.CAA\_Parameter\_Constants.udiInX; //ID von IN X

LTocpa\_ComParameter1[16].udiValue := GLRudi\_INX; //Eingabeflusssteuerung/ keine ahnung wad das ist.

//XOnChar

LTocpa\_ComParameter1[17].udiParameterId := COM.CAA\_Parameter\_Constants.udiXonChar; //ID von x on char

LTocpa\_ComParameter1[17].udiValue := GLRudi\_XOnChar; //Tx AND Rx XON Zeichen/ keine ahnung wad das ist.

//XOffChar

LTocpa\_ComParameter1[18].udiParameterId := COM.CAA\_Parameter\_Constants.udiXoffChar; //ID von x off char

LTocpa\_ComParameter1[18].udiValue := GLRudi\_XOffChar; //Tx AND Rx XOFF Zeichen/ keine ahnung wad das ist.

//XonLim

LTocpa\_ComParameter1[19].udiParameterId := COM.CAA\_Parameter\_Constants.udiXonLim; //ID von X on Lim

LTocpa\_ComParameter1[19].udiValue := GLRudi\_XOnLim; //Übertrage XON Schwellenwert/ keine ahnung wad das ist.

//XoffLim

LTocpa\_ComParameter1[20].udiParameterId := COM.CAA\_Parameter\_Constants.udiXonLim; //ID von X off Lim

LTocpa\_ComParameter1[20].udiValue := GLRudi\_XOnLim; //Übertrage XOFF Schwellenwert/ keine ahnung wad das ist.

\*)

LTi\_Zustand := 2;

A2\_Open\_Com

//Com Port öffnen

FB\_ComOpen.usiListLength := SIZEOF(LTocpa\_ComParameter1); //Die länge der Parameter Liste

FB\_ComOpen.pParameterList := ADR(LTocpa\_ComParameter1); //Die Adresse der Parameter Liste

FB\_ComOpen.xExecute := 1; //Start Flag setzten.

LTi\_Zustand := 3;

FB\_Timer(IN := 1, PT:= LTt\_WaitTimeout); //Rufe Timer auf

A3\_Open\_Wait

IF FB\_ComOpen.xDone THEN //Wenn der Port offen

FB\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 4; //Gehe in den Sende Schritt

LTudi\_Handle := FB\_ComOpen.hCom; //Speichere den Handle

LTx\_ComOpen := 1; //Der Offene Com Port setzten.

ELSE

IF FB\_ComOpen.xError THEN //Wenn es einen Error bei Com Port öffnen gab.

FB\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := 1; //Den ENUM Wert Error angeben.

ELSE

IF FB\_Timer.Q THEN //Wenn der Timer Abgelaufen ist.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := 4; //Den ENUM Wert Error angeben.

FB\_Timer.IN := 0;

END\_IF

END\_IF

END\_IF

FB\_Timer(); //Rufe Timer auf

FB\_ComOpen(); //Rufe den FB auf.

A4\_Send\_Com

FB\_WriteCom(

hCom := LTudi\_Handle, //Ubergebe Handle.

pBuffer := ADR(stSendData),

szSize := INT\_TO\_UDINT(SIZEOF(stSendData)),

xExecute := 1 //Starte Sende Vorgang.

);

LTi\_Zustand := 5;

FB\_Timer(IN := 1, PT:= LTt\_WaitTimeout); //Rufe Timer auf

A5\_Send\_Wait

IF FB\_WriteCom.xDone THEN //Wenn der String gesendet wurde.

FB\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 6; //Gehe in den Sende Schritt

ELSE

IF FB\_WriteCom.xError THEN //Wenn es einen Error bei String senden gab.

FB\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := 2; //Den ENUM Wert Error angeben.

ELSE

IF FB\_Timer.Q THEN //Wenn der Timer Abgelaufen ist.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := 5; //Den ENUM Wert Error angeben.

FB\_Timer.IN := 0;

END\_IF

END\_IF

END\_IF

FB\_Timer(); //Rufe Timer auf

FB\_WriteCom(); //Rufe den FB auf.

A6\_Receive\_Com

FB\_Timer(IN := 1, PT:= LTt\_WaitCom); //Rufe Timer auf

IF FB\_TIMER.Q THEN //Wenn der Timer fertig ist

FB\_Timer.IN := 0; //Timer zurücksetzten

FB\_Timer(); //FB Timer zurückstzen.

FB\_ReadCom //Rufe ReadCom auf.

(

xExecute := 1, //Starte Lese STRING

hCom := LTudi\_Handle, //übergib den Handle.

pBuffer := ADR(LTst\_ReceivedData),

szBuffer := SIZEOF(LTst\_ReceivedData),

);

LTi\_Zustand := 7;

END\_IF

A7\_Receive\_Wait

IF FB\_ReadCom.xDone THEN //Wenn der String gesendet wurde.

FB\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 8; //Gehe in den Sende Schritt

ELSE

IF FB\_ReadCom.xError THEN //Wenn es einen Error bei String senden gab.

FB\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := 3; //Den ENUM Wert Error angeben.

ELSE

IF FB\_Timer.Q THEN //Wenn der Timer Abgelaufen ist.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := 6; //Den ENUM Wert Error angeben.

FB\_Timer.IN := 0;

END\_IF

END\_IF

FB\_Timer(IN := 1, PT:= LTt\_WaitTimeout); //Schalte Timer ein.

END\_IF

FB\_Timer(); //Rufe Timer auf

FB\_ReadCom(); //Rufe den FB auf.

A8\_String\_Auswertung

LTi\_PosString := FIND(STR2:='$R$N' , STR1:=LTst\_ReceivedData); //Das caretsch Retain lengfield

LTst\_ReceivedData[LTi\_PosString - 1] := 0; //DAs caretain legas löschen mit allem was dahinter ist.

stReceivedData := LTst\_ReceivedData; //Ausgeben

LTi\_Zustand := 9; //Zum Zustand 9 gehen.

A9\_End

xBusy := 0; //Alles zurücksetzten

xDone := 1;

IF xExecute = 0 THEN //Warten bis der User Zurückgesetzt hat.

xDone := 0;

LTi\_Zustand := 0;

stReceivedData := '';

FB\_WriteCom.xExecute := 0;

FB\_ReadCom.xExecute := 0;

LTst\_ReceivedData := '';

FB\_ComOpen.xExecute := 0;

FB\_WriteCom();

FB\_ReadCom();

FB\_ComOpen();

END\_IF

B100\_Error

xBusy := 0; //Alles zurücksetzten

xError := 1;

IF xExecute = 0 THEN //Wartem bis der User das xExecute zurückgesetzt hat.

xError := 0;

LTi\_Zustand := 0;

wState := 0;

stReceivedData := '';

LTst\_ReceivedData := '';

FB\_WriteCom.xExecute := 0;

FB\_ReadCom.xExecute := 0;

FB\_ComOpen.xExecute := 0;

FB\_WriteCom();

FB\_ReadCom();

FB\_ComOpen();

END\_IF

# FB Connect with TSR

## Variablen Deklaration

FUNCTION\_BLOCK Connect\_with\_TSR

VAR\_INPUT

xExecute :BOOL; //Startbedingung für den FB.

iCom :INT; //Die Nummer des Com Ports.

END\_VAR

VAR\_OUTPUT

xBusy :BOOL; //Das Busy bit.

xDone :BOOL; //Ist 1 wenn FB fertig ist.

wState :WORD; //Der ENUM Wert für die ENUM Tabelle.

xError :BOOL; //Ist 1 wenn es einen Error gab.

END\_VAR

VAR

LTi\_Zustand :INT; //Der Zustand.

//TIMER

FB\_Timer :TON; //Timer Init

END\_VAR

VAR CONSTANT

LTt\_WaitTimeout :TIME := T#2S; //der Timout Timer.

END\_VAR

## Main

CASE LTi\_Zustand OF

0:

A0\_Start(); //Prüfe die Startbedingungen

1:

A1\_Progress(); //Starte den Prozess.

2:

A2\_Progress\_Wait(); //Warte auf den FB TSR Com

3:

A3\_Auswerten(); //Werte das Egebniss aus.

4:

A4\_End(); //Warte auf Execute

100:

B100\_Error(); //Nur wenn ein Error gab.

END\_CASE

FB\_TSRCom();

## Aktionen

A0\_Start

IF xExecute THEN //Prüfe Startbedinung.

xBusy := 1; //Setze Busy.

LTi\_Zustand := 1; //Gehe direkt zum Schritt wo der Com Port geöffnet wird.

END\_IF

A1\_Progress

FB\_TSRCom( //TSR Com Aufrufen und Daten übergeben.

xEXecute := 1 ,

iCom := iCom,

stSendData := '?', );

LTi\_Zustand := 2; //In den nächsten Zustand Wechseln.

A2\_Progress\_Wait

IF FB\_TSRCom.xDone THEN //Wenn der Port offen

FB\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 3; //Gehe in den Sende Schritt

ELSE

IF FB\_TSRCom.xError THEN //Wenn es einen Error bei Com Port öffnen gab.

FB\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := FB\_TSRCom.wState; //Den ENUM Wert Error angeben.

ELSE

IF FB\_Timer.Q THEN //Wenn der Timer Abgelaufen ist.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := 7; //Den ENUM Wert Error angeben.

FB\_Timer.IN := 0;

END\_IF

END\_IF

FB\_Timer(IN := 1, PT:= LTt\_WaitTimeout);

END\_IF

FB\_Timer(); //Rufe Timer auf

A3\_Auswerten

IF FB\_TSRCom.stReceivedData = '!' THEN //Wenn der TSR reagiert.

LTi\_Zustand := 4; //Gehe zum letzten Zustand.

ELSE

LTi\_Zustand := 100;

wState := 8;

END\_IF

A4\_End

xBusy := 0; //Alles zurücksetzten

xDone := 1;

IF xExecute = 0 THEN //Warten bis der User Zurückgesetzt hat.

xDone := 0;

LTi\_Zustand := 0;

FB\_TSRCom.xEXecute := 0;

END\_IF

B100\_Error

xBusy := 0; //Alles zurücksetzten

xError := 1;

IF xExecute = 0 THEN //Wartem bis der User das xExecute zurückgesetzt hat.

xError := 0;

LTi\_Zustand := 0;

wState := 0;

FB\_TSRCom.xEXecute := 0;

END\_IF

# FB Get TSR position

## Variablen Deklaration

FUNCTION\_BLOCK FB\_Get\_TSR\_position

VAR\_INPUT

//FB EINGANGS VAR

xExecute :BOOL; //Startbedingung für den FB.

END\_VAR

VAR\_OUTPUT

//FB AUSGANGS VAR

xBusy :BOOL; //Das Busy bit.

xDone :BOOL; //Ist 1 wenn FB fertig ist.

wState :WORD; //Der ENUM Wert für die ENUM Tabelle.

xError :BOOL; //Ist 1 wenn es einen Error gab.

iPosx :INT; //Die Position vom TSR.

iPosy :INT; //Die Position vom TSR.

END\_VAR

VAR

//INTERNE VAR

LTi\_Zustand :INT; //Der Zustand.

LTi\_PosMid :INT; //Diese Variable soll angeben bei welcher Position sich das : vom rückgabe wert befindet.

LTi\_Len :INT; //Gibt die String länge an für den Progress.

//TIMER

FN\_Timer :TON; //Timer Init.

END\_VAR

//Parameters

VAR CONSTANT

LTt\_WaitTimeout :TIME := T#2S; //der Timout Timer.

LTt\_WaitCom :TIME := T#200MS; //Wait auf Received Data.

END\_VAR

## Main

CASE LTi\_Zustand OF

0:

A0\_Start(); //Prüfe Startbedingunen

1:

A1\_Progress(); //Starte den Prozess. und sende den Befehl an das TSR.

2:

A2\_Progress\_Wait(); //Warte auf den FB Com.

3:

A3\_Auswerten(); //Beende den FB und warte auf den User.

4:

A4\_End(); //Das ende der Funktion.

100:

B100\_Error(); //Die Error Aktion.

END\_CASE

FB\_TSRCom(); //Aufrufen.

## Aktionen

A0\_Start

IF xExecute THEN //Prüfe Startbedinung.

xBusy := 1; //Setze Busy.

LTi\_Zustand := 1; //nächster Schritt

END\_IF

A1\_Progress

//TSR Com Aufrufen und Daten übergeben.

FB\_TSRCom(

iCom := 1,

xEXecute := 1,

stSendData := 'getpos',);

//Auf zum nächsten schritt.

LTi\_Zustand := 2;

A2\_Progress\_Wait

IF FB\_TSRCom.xDone THEN //Wenn der Port offen

FN\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 3; //Gehe in den Sende Schritt

ELSE

IF FB\_TSRCom.xError THEN //Wenn es einen Error bei Com Port öffnen gab.

FN\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := FB\_TSRCom.wState; //Den ENUM Wert Error angeben.

ELSE

IF FN\_Timer.Q THEN //Wenn der Timer Abgelaufen ist.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := 7; //Den ENUM Wert Error angeben.

FN\_Timer.IN := 0;

END\_IF

END\_IF

FN\_Timer(IN := 1, PT:= LTt\_WaitTimeout); //Setzte den Timer.

END\_IF

FN\_Timer(); //Timer Aufrufen. //Rufe Timer auf

A3\_Auswerten

LTi\_PosMid := FIND(FB\_TSRCom.stReceivedData,':'); //Speichert die Pos wo sich das : befindet.

LTi\_Len := LEN(FB\_TSRCom.stReceivedData);

iPosx := REAL\_TO\_INT(INT\_TO\_REAL((STRING\_TO\_INT(LEFT(STR:=FB\_TSRCom.stReceivedData , SIZE:=LTi\_PosMid))) - iStartPosx) / GTr\_Faktor);

iPosy := REAL\_TO\_INT(INT\_TO\_REAL((STRING\_TO\_INT(RIGHT(STR:=FB\_TSRCom.stReceivedData , SIZE:=LTi\_Len - LTi\_PosMid))) - iStartPosy) / GTr\_Faktor);

LTi\_Zustand := 4;

A4\_End

xBusy := 0; //Alles zurücksetzten

xDone := 1;

IF xExecute = 0 THEN //Warten bis der User Zurückgesetzt hat.

xDone := 0;

LTi\_Zustand := 0;

FB\_TSRCom.xEXecute := 0;

iPosx := 0;

iPosy := 0;

END\_IF

B100\_Error

xBusy := 0; //Alles zurücksetzten

xError := 1;

IF xExecute = 0 THEN //Wartem bis der User das xExecute zurückgesetzt hat.

xError := 0;

LTi\_Zustand := 0;

wState := 0;

FB\_TSRCom.xEXecute := 0; //FB Com zurücksetzetn

iPosx := 0;

iPosy := 0;

END\_IF

# FN Set TSR Startpoint

## Variablen Deklaration

FUNCTION FN\_Set\_TSR\_startpoint : BOOL

VAR\_INPUT

//FB EINGANGS VAR

iStartpointx :INT; //Der Virtuelle x Nullpunkt.

iStartpointy :INT; //Der Virtuelle y Nullpunkt.

END\_VAR

## Main

iStartPosx := REAL\_TO\_INT(INT\_TO\_REAL(iStartpointx) \* GTr\_Faktor); //Speicher eingang in Global Var

iStartPosy := REAL\_TO\_INT(INT\_TO\_REAL(iStartpointy) \* GTr\_Faktor); //Speicher eingang in Global Var

FN\_Set\_TSR\_startpoint := 1; // Der Rückgabe wert der Funktion

# FB Move TSR xy

## Variablen Deklaration

FUNCTION\_BLOCK FB\_Move\_TSR\_xy

VAR\_INPUT

xExecute :BOOL; //Startbedingung für den FB.

iMovx :INT; //Die relative x bewegung des TSR in richtung x

iMovy :INT; //Die relative y bewegung des TSR in richtung y

END\_VAR

VAR\_OUTPUT

xBusy :BOOL; //Das Busy bit.

xDone :BOOL; //Ist 1 wenn FB fertig ist.

wState :WORD; //Der ENUM Wert für die ENUM Tabelle.

xError :BOOL; //Ist 1 wenn es einen Error gab.

END\_VAR

VAR

LTi\_Zustand :INT; //Der Zustand.

LTst\_Zwischenspeicher :STRING; //Ein Zwischenspeicher für den Progress.

LTi\_Len :INT; //Gibt die String länge an für den Progress.

LTi\_x :INT; //Die x Schritte die der TSR machen soll.

LTi\_y :INT; //Die y Schritte die der TSR machen soll.

//TIMER

FN\_Timer :TON; //Timer Init.

END\_VAR

//Parameters

VAR CONSTANT

LTt\_WaitTimeout :TIME := T#2S; //der Timout Timer.

LTt\_WaitCom :TIME := T#200MS; //Wait auf Received Data.

END\_VAR

## Main

CASE LTi\_Zustand OF

0:

A0\_Start(); //Prüfe Startbedingunen

1:

A1\_Progress(); //Starte den Prozess. und sende den Befehl an das TSR.

2:

A2\_Progress\_Wait(); //Warte auf den FB Com.

3:

A3\_Auswerten(); //Beende den FB und warte auf den User.

4:

A4\_Wait\_for\_TSR(); //Init warten bis der TSR bei der Position angekommen ist

5:

A5\_Wait\_for\_TSR\_Wait(); //Warte auf FB Com.

6:

A6\_Wait\_for\_TSR\_Auswerten(); //Werte aus.

7:

A7\_End(); //Das ende der Funktion.

100:

B100\_Error(); //Die Error Aktion.

END\_CASE

FB\_TSRCom(); //Aufrufen.

## Aktionen

A0\_Start

IF xExecute THEN //Prüfe Startbedinung.

xBusy := 1; //Setze Busy.

LTi\_Zustand := 1; //nächster Schritt

END\_IF

A1\_Progress

//Hier wird der Befehl für den TSR zusammengesetzt.

LTst\_Zwischenspeicher := 'mov '; //Speicher

//Die Fakto miteinbeziehen so das die mm umgerechnet werden, in Schritte.

LTi\_x := REAL\_TO\_INT(INT\_TO\_REAL(iMovx) \* GTr\_Faktor);

//Schreibe Die X koordinate hinein.

LTst\_Zwischenspeicher := INSERT(LTst\_Zwischenspeicher,INT\_TO\_STRING(LTi\_x),4);

//Finde die länge des String heraus

LTi\_Len := LEN(STR := LTst\_Zwischenspeicher);

//Schreibe am Schluss ein :

LTst\_Zwischenspeicher := INSERT(LTst\_Zwischenspeicher,':',LTi\_Len);

//Die Fakto miteinbeziehen so das die mm umgerechnet werden, in Schritte.

LTi\_y := REAL\_TO\_INT(INT\_TO\_REAL(iMovy) \* GTr\_Faktor);

//Nun schreibe hinter dem : die y Koordiante.

LTst\_Zwischenspeicher := INSERT(LTst\_Zwischenspeicher,INT\_TO\_STRING(LTi\_y),(LTi\_Len + 1));

//TSR Com Aufrufen und Daten übergeben.

FB\_TSRCom(

iCom := 1,

xEXecute := 1 ,

stSendData := LTst\_Zwischenspeicher, );

//Auf zum nächsten schritt.

LTi\_Zustand := 2;

A2\_Progress\_Wait

IF FB\_TSRCom.xDone THEN //Wenn der Port offen

FN\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 3; //Gehe in den Sende Schritt

ELSE

IF FB\_TSRCom.xError THEN //Wenn es einen Error bei Com Port öffnen gab.

FN\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := FB\_TSRCom.wState; //Den ENUM Wert Error angeben.

ELSE

IF FN\_Timer.Q THEN //Wenn der Timer Abgelaufen ist.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := 7; //Den ENUM Wert Error angeben.

FN\_Timer.IN := 0;

END\_IF

END\_IF

FN\_Timer(IN := 1, PT:= LTt\_WaitTimeout); //Setzte den Timer.

END\_IF

FN\_Timer(); //Timer Aufrufen.

A3\_Auswerten

IF FB\_TSRCom.stReceivedData = 'moving' THEN //Wenn der TSR reagiert.

LTi\_Zustand := 4; //Gehe zum letzten Zustand.

FB\_TSRCom.xEXecute := 0; //Setzte Com Communication zurück.

ELSE

IF FB\_TSRCom.stReceivedData = 'out of range' THEN

wState := 9; //out of range state

LTi\_zustand := 100;

ELSE

LTi\_Zustand := 100;

wState := 8; //Gehe zum letzten Zustand.

END\_IF

END\_IF

A4\_Wait\_for\_TSR

//TSR Com Aufrufen und Daten übergeben.

FB\_TSRCom(

iCom := 1,

xEXecute := 1 ,

stSendData := 'state', ); //frage den Status ab.

//Auf zum nächsten schritt.

LTi\_Zustand := 5;

A5\_Wait\_for\_TSR\_Wait

IF FB\_TSRCom.xDone THEN //Wenn der Port offen

FN\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 6; //Gehe in den Sende Schritt

ELSE

IF FB\_TSRCom.xError THEN //Wenn es einen Error bei Com Port öffnen gab.

FN\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := FB\_TSRCom.wState; //Den ENUM Wert Error angeben.

ELSE

IF FN\_Timer.Q THEN //Wenn der Timer Abgelaufen ist.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := 7; //Den ENUM Wert Error angeben.

FN\_Timer.IN := 0;

END\_IF

END\_IF

FN\_Timer(IN := 1, PT:= LTt\_WaitTimeout); //Setzte den Timer.

END\_IF

FN\_Timer(); //Timer Aufrufen.

A6\_Wait\_for\_TSR\_Auswerten

IF FB\_TSRCom.stReceivedData = 'ready' THEN //Wenn der TSR reagiert.

LTi\_Zustand := 7; //Gehe zum letzten Zustand.

FB\_TSRCom.xEXecute := 0; //Setzte Com Communication zurück.

ELSE

IF FB\_TSRCom.stReceivedData = 'busy' THEN //Wenn der TSR noch busy ist frage nochmal.

FB\_TSRCom.xEXecute := 0;

LTi\_zustand := 4;

ELSE

IF FB\_TSRCom.stReceivedData = '' THEN

FB\_TSRCom.xEXecute := 0;

LTi\_zustand := 4;

ELSE

LTi\_Zustand := 100;

wState := 8; //Gehe zum letzten Zustand.

END\_IF

END\_IF

END\_IF

A7\_End

xBusy := 0; //Alles zurücksetzten

xDone := 1;

IF xExecute = 0 THEN //Warten bis der User Zurückgesetzt hat.

xDone := 0;

LTi\_Zustand := 0;

FB\_TSRCom.xEXecute := 0;

END\_IF

B100\_Error

xBusy := 0; //Alles zurücksetzten

xError := 1;

IF xExecute = 0 THEN //Wartem bis der User das xExecute zurückgesetzt hat.

xError := 0;

LTi\_Zustand := 0;

wState := 0;

FB\_TSRCom.xEXecute := 0; //FB Com zurücksetzetn

END\_IF

# FB Go to TSR xy

## Variablen Deklaration

FUNCTION\_BLOCK FB\_Go\_to\_TSR\_xy

VAR\_INPUT

xExecute :BOOL; //Startbedingung für den FB.

iGotox :INT; //Die relative x bewegung des TSR in richtung x

iGotoy :INT; //Die relative y bewegung des TSR in richtung y

END\_VAR

VAR\_OUTPUT

xBusy :BOOL; //Das Busy bit.

xDone :BOOL; //Ist 1 wenn FB fertig ist.

wState :WORD; //Der ENUM Wert für die ENUM Tabelle.

xError :BOOL; //Ist 1 wenn es einen Error gab.

END\_VAR

VAR

LTi\_Zustand :INT; //Der Zustand.

LTst\_Zwischenspeicher :STRING; //Ein Zwischenspeicher für den Progress.

LTi\_Len :INT; //Gibt die String länge an für den Progress.

LTi\_x :INT; //Die x Schritte die der TSR machen soll.

LTi\_y :INT; //Die y Schritte die der TSR machen soll.

//TIMER

FN\_Timer :TON; //Timer Init.

END\_VAR

//Parameters

VAR CONSTANT

LTt\_WaitTimeout :TIME := T#2S; //der Timout Timer.

LTt\_WaitCom :TIME := T#200MS; //Wait auf Received Data.

END\_VAR

## Main

CASE LTi\_Zustand OF

0:

A0\_Start(); //Prüfe Startbedingunen

1:

A1\_Progress(); //Starte den Prozess. und sende den Befehl an das TSR.

2:

A2\_Progress\_Wait(); //Warte auf den FB Com.

3:

A3\_Auswerten(); //Beende den FB und warte auf den User.

4:

A4\_Wait\_for\_TSR(); //Init warten bis der TSR bei der Position angekommen ist

5:

A5\_Wait\_for\_TSR\_Wait(); //Warte auf FB Com.

6:

A6\_Wait\_for\_TSR\_Auswerten(); //Werte aus.

7:

A7\_End(); //Das ende der Funktion.

100:

B100\_Error(); //Die Error Aktion.

END\_CASE

FB\_TSRCom(); //Aufrufen. //Aufrufen.

## Aktionen

A0\_Start

IF xExecute THEN //Prüfe Startbedinung.

xBusy := 1; //Setze Busy.

LTi\_Zustand := 1; //nächster Schritt

END\_IF

A1\_Progress

//Hier wird der Befehl für den TSR zusammengesetzt.

LTst\_Zwischenspeicher := 'movto '; //Speicher

//Die Fakto miteinbeziehen so das die mm umgerechnet werden, in Schritte.

LTi\_x := iStartPosx + (REAL\_TO\_INT(INT\_TO\_REAL(iGotox) \* GTr\_Faktor));

//Schreibe Die X koordinate hinein.

LTst\_Zwischenspeicher := INSERT(LTst\_Zwischenspeicher,INT\_TO\_STRING(LTi\_x),6);

//Finde die länge des String heraus

LTi\_Len := LEN(STR := LTst\_Zwischenspeicher);

//Schreibe am Schluss ein :

LTst\_Zwischenspeicher := INSERT(LTst\_Zwischenspeicher,':',LTi\_Len);

//Die Fakto miteinbeziehen so das die mm umgerechnet werden, in Schritte.

LTi\_y := iStartPosy + (REAL\_TO\_INT(INT\_TO\_REAL(iGotoy) \* GTr\_Faktor));

//Nun schreibe hinter dem : die y Koordiante.

LTst\_Zwischenspeicher := INSERT(LTst\_Zwischenspeicher,INT\_TO\_STRING(LTi\_y),(LTi\_Len + 1));

//TSR Com Aufrufen und Daten übergeben.

FB\_TSRCom(

iCom := 1,

xEXecute := 1 ,

stSendData := LTst\_Zwischenspeicher, );

//Auf zum nächsten schritt.

LTi\_Zustand := 2;

A2\_Progress\_Wait

IF FB\_TSRCom.xDone THEN //Wenn der Port offen

FN\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 3; //Gehe in den Sende Schritt

ELSE

IF FB\_TSRCom.xError THEN //Wenn es einen Error bei Com Port öffnen gab.

FN\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := FB\_TSRCom.wState; //Den ENUM Wert Error angeben.

ELSE

IF FN\_Timer.Q THEN //Wenn der Timer Abgelaufen ist.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := 7; //Den ENUM Wert Error angeben.

FN\_Timer.IN := 0;

END\_IF

END\_IF

FN\_Timer(IN := 1, PT:= LTt\_WaitTimeout); //Setzte den Timer.

END\_IF

FN\_Timer();

//Timer Aufrufen.

A3\_Auswerten

IF FB\_TSRCom.stReceivedData = 'moving' THEN //Wenn der TSR reagiert.

LTi\_Zustand := 4; //Gehe zum letzten Zustand.

FB\_TSRCom.xEXecute := 0; //Setzte Com Communication zurück.

ELSE

IF FB\_TSRCom.stReceivedData = 'out of range' THEN

wState := 9; //out of range state

LTi\_zustand := 100;

ELSE

LTi\_Zustand := 100;

wState := 8; //Gehe zum letzten Zustand.

END\_IF

END\_IF

A4\_Wait\_for\_TSR

//TSR Com Aufrufen und Daten übergeben.

FB\_TSRCom(

iCom := 1,

xEXecute := 1 ,

stSendData := 'state', ); //frage den Status ab.

//Auf zum nächsten schritt.

LTi\_Zustand := 5;

A5\_Wait\_for\_TSR\_Wait

IF FB\_TSRCom.xDone THEN //Wenn der Port offen

FN\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 6; //Gehe in den Sende Schritt

ELSE

IF FB\_TSRCom.xError THEN //Wenn es einen Error bei Com Port öffnen gab.

FN\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := FB\_TSRCom.wState; //Den ENUM Wert Error angeben.

ELSE

IF FN\_Timer.Q THEN //Wenn der Timer Abgelaufen ist.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := 7; //Den ENUM Wert Error angeben.

FN\_Timer.IN := 0;

END\_IF

END\_IF

FN\_Timer(IN := 1, PT:= LTt\_WaitTimeout); //Setzte den Timer.

END\_IF

FN\_Timer(); //Timer Aufrufen.

A6\_Wait\_for\_TSR\_Auswerten

IF FB\_TSRCom.stReceivedData = 'ready' THEN //Wenn der TSR reagiert.

LTi\_Zustand := 7; //Gehe zum letzten Zustand.

FB\_TSRCom.xEXecute := 0; //Setzte Com Communication zurück.

ELSE

IF FB\_TSRCom.stReceivedData = 'busy' THEN //Wenn der TSR noch busy ist frage nochmal.

FB\_TSRCom.xEXecute := 0;

LTi\_zustand := 4;

ELSE

IF FB\_TSRCom.stReceivedData = '' THEN

FB\_TSRCom.xEXecute := 0;

LTi\_zustand := 4;

ELSE

LTi\_Zustand := 100;

wState := 8; //Gehe zum letzten Zustand.

END\_IF

END\_IF

END\_IF

A7\_End

xBusy := 0; //Alles zurücksetzten

xDone := 1;

IF xExecute = 0 THEN //Warten bis der User Zurückgesetzt hat.

xDone := 0;

LTi\_Zustand := 0;

FB\_TSRCom.xEXecute := 0;

END\_IF

B100\_Error

xBusy := 0; //Alles zurücksetzten

xError := 1;

IF xExecute = 0 THEN //Wartem bis der User das xExecute zurückgesetzt hat.

xError := 0;

LTi\_Zustand := 0;

wState := 0;

FB\_TSRCom.xEXecute := 0; //FB Com zurücksetzetn

END\_IF

# FB TSR Pen up down

## Variablen Deklaration

FUNCTION\_BLOCK TSR\_Pen\_up\_down

VAR\_INPUT

xExecute :BOOL; //Startbedingung für den FB.

xPen :BOOL; //Wenn null = oben. und 1 = unten.

END\_VAR

VAR\_OUTPUT

xBusy :BOOL; //Das Busy bit.

xDone :BOOL; //Ist 1 wenn FB fertig ist.

wState :WORD; //Der ENUM Wert für die ENUM Tabelle.

xError :BOOL; //Ist 1 wenn es einen Error gab.

END\_VAR

VAR

LTi\_Zustand :INT; //Der Zustand.

LTst\_String :STRING; //Ein Zwischenspeicher.

//TIMER

FB\_Timer :TON; //Timer Init

END\_VAR

VAR CONSTANT

LTt\_WaitTimeout :TIME := T#2S; //der Timout Timer.

END\_VAR

## Main

CASE LTi\_Zustand OF

0:

A0\_Start(); //Prüfe die Startbedingungen

1:

A1\_Progress(); //Starte den Prozess.

2:

A2\_Progress\_Wait(); //Warte auf den FB TSR Com

3:

A3\_Auswerten(); //Werte das Egebniss aus.

4:

A4\_End(); //Warte auf Execute

100:

B100\_Error(); //Nur wenn ein Error gab.

END\_CASE

FB\_TSRCom();

## Aktionen

A0\_Start

IF xExecute THEN //Prüfe Startbedinung.

xBusy := 1; //Setze Busy.

LTi\_Zustand := 1; //Gehe direkt zum Schritt wo der Com Port geöffnet wird.

END\_IF

A1\_Progress

IF xPen THEN

LTst\_String := 'touchdown';

ELSE

LTst\_String := 'touchup';

END\_IF

FB\_TSRCom( //TSR Com Aufrufen und Daten übergeben.

xEXecute := 1 ,

stSendData := LTst\_String, );

LTi\_Zustand := 2; //In den nächsten Zustand Wechseln.

A2\_Progress\_Wait

IF FB\_TSRCom.xDone THEN //Wenn der Port offen

FB\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 3; //Gehe in den Sende Schritt

ELSE

IF FB\_TSRCom.xError THEN //Wenn es einen Error bei Com Port öffnen gab.

FB\_Timer.IN := 0; //Timer Zurücksetzten.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := FB\_TSRCom.wState; //Den ENUM Wert Error angeben.

ELSE

IF FB\_Timer.Q THEN //Wenn der Timer Abgelaufen ist.

LTi\_Zustand := 100; //In den Error Zustand gehen.

wState := 7; //Den ENUM Wert Error angeben.

FB\_Timer.IN := 0;

END\_IF

END\_IF

FB\_Timer(IN := 1, PT:= LTt\_WaitTimeout);

END\_IF

FB\_Timer(); //Rufe Timer auf //Timer Aufrufen.

A3\_Auswerten

IF FB\_TSRCom.stReceivedData = 'ok' THEN //Wenn der TSR reagiert.

LTi\_Zustand := 4; //Gehe zum letzten Zustand.

ELSE

LTi\_Zustand := 100;

wState := 8;

END\_IF

A7\_End

xBusy := 0; //Alles zurücksetzten

xDone := 1;

IF xExecute = 0 THEN //Warten bis der User Zurückgesetzt hat.

xDone := 0;

LTi\_Zustand := 0;

FB\_TSRCom.xEXecute := 0;

END\_IF

B100\_Error

xBusy := 0; //Alles zurücksetzten

xError := 1;

IF xExecute = 0 THEN //Wartem bis der User das xExecute zurückgesetzt hat.

xError := 0;

LTi\_Zustand := 0;

wState := 0;

FB\_TSRCom.xEXecute := 0;

END\_IF