

SPS-TECHNIK SCHRITTMOTOREN STEUERN MIT B&R

FH CAMPUS 02, Automatisierungstechnik

Anton Holzerbauer
Christian Rinner

Inhalt

- Aufgabe 1
- Aufgabe 2
- Hardware
 - Verdrahtung
- Programmierung
- Schrittmotoransteuerung
 - Steuerwort
 - Statuswort

Aufgabenstellung 1: Schrittmotor ansteuern

■ Grundlagen

- Schrittmotor-Ansteuerung
- Was ist ein Statuswort / Steuerwort?
- Verdrahtung

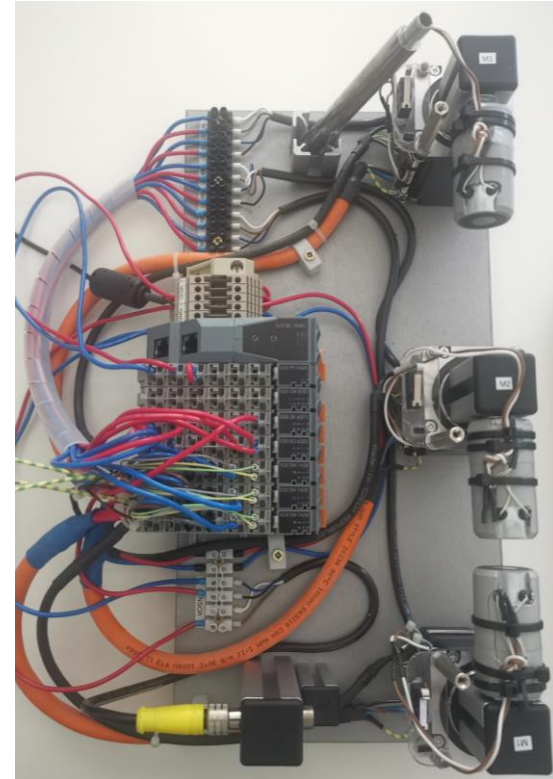
■ Programmierung



Quelle: Eigene Darstellung

Aufgabenstellung 2: Pick- and Place-Anwendung

- 3 Schrittmotoren
- 3 Elektromagnete
- Metallobjekt wird weitergegeben



Quelle: Eigene Darstellung

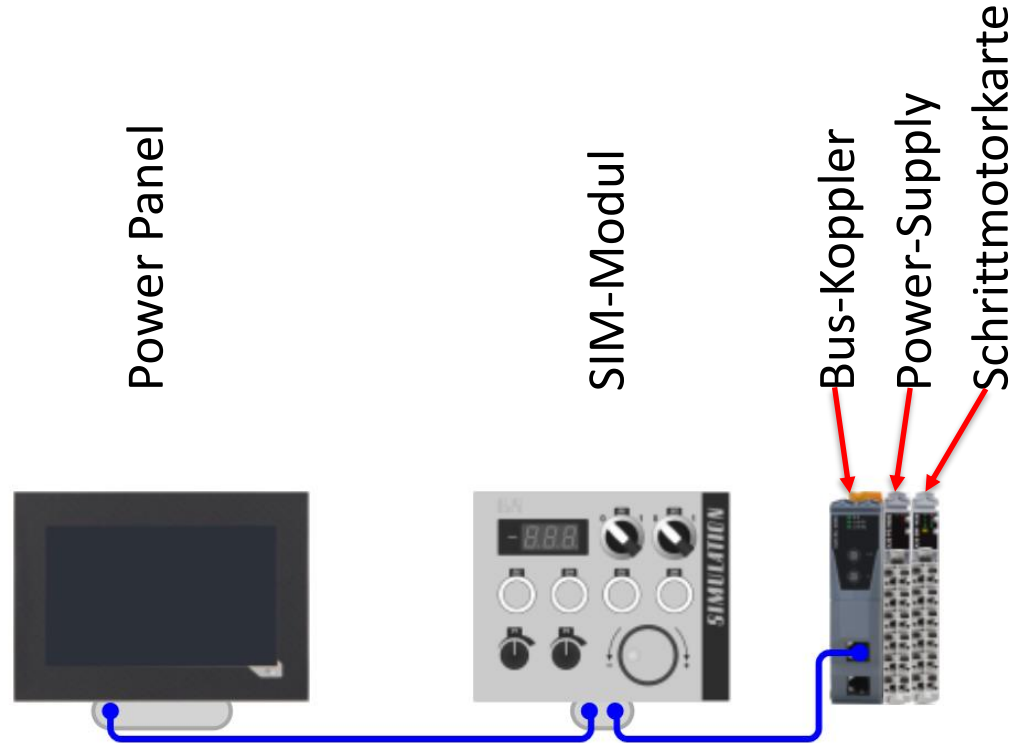
Hardware

- I/O-Simulationspanel (4SIM.10-01)
- Power Panel 70
- Bus Controller (X20BC0083)
- Einspeisemodul (X20PS9400)
- Schrittmotormodule (X20SM1426)
- Schrittmotor NEMA17
- Induktiver Näherungssensor (I1CH005)



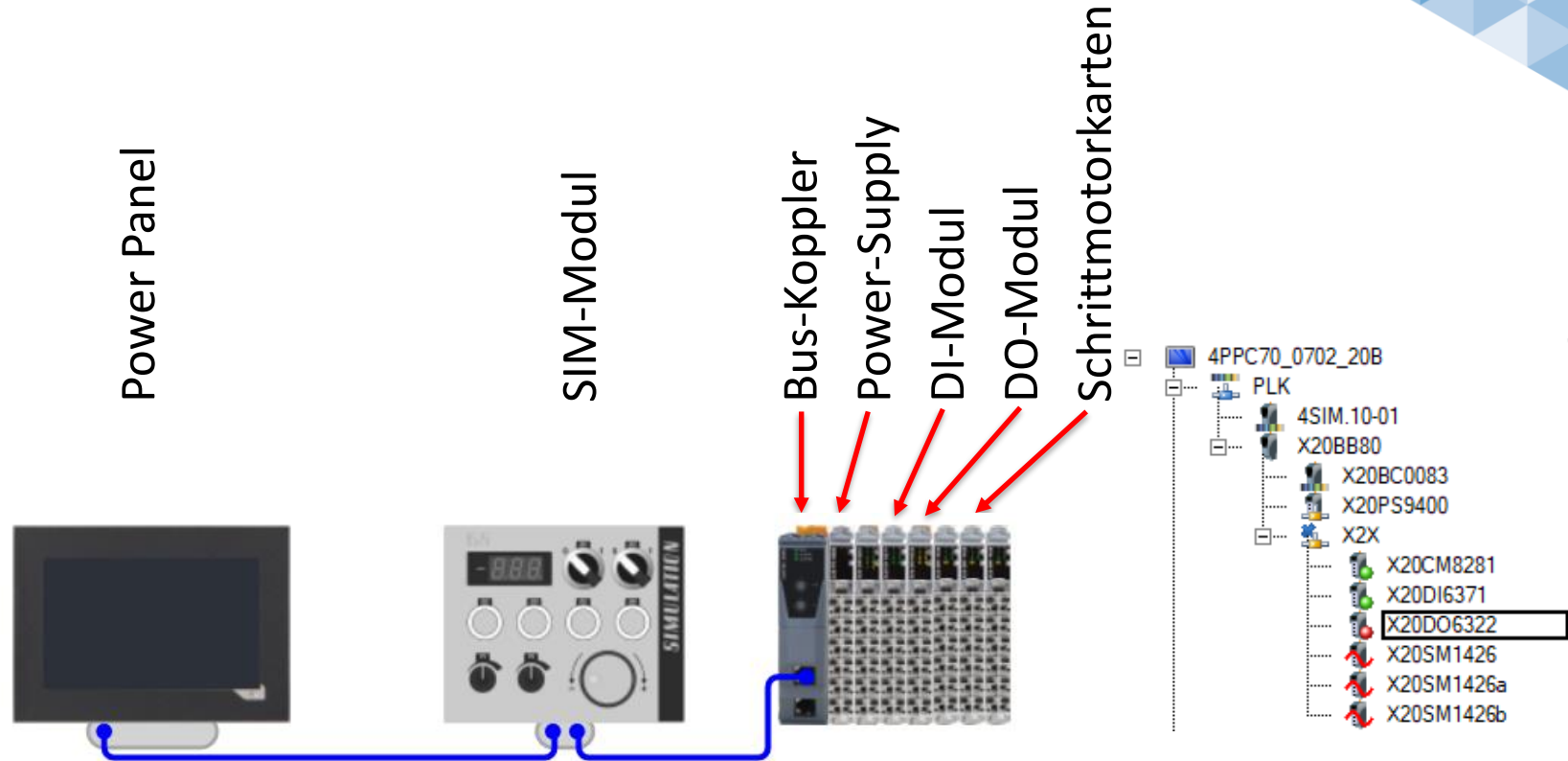
Quelle: Datenblätter

Hardwarekonfiguration Aufgabenstellung 1



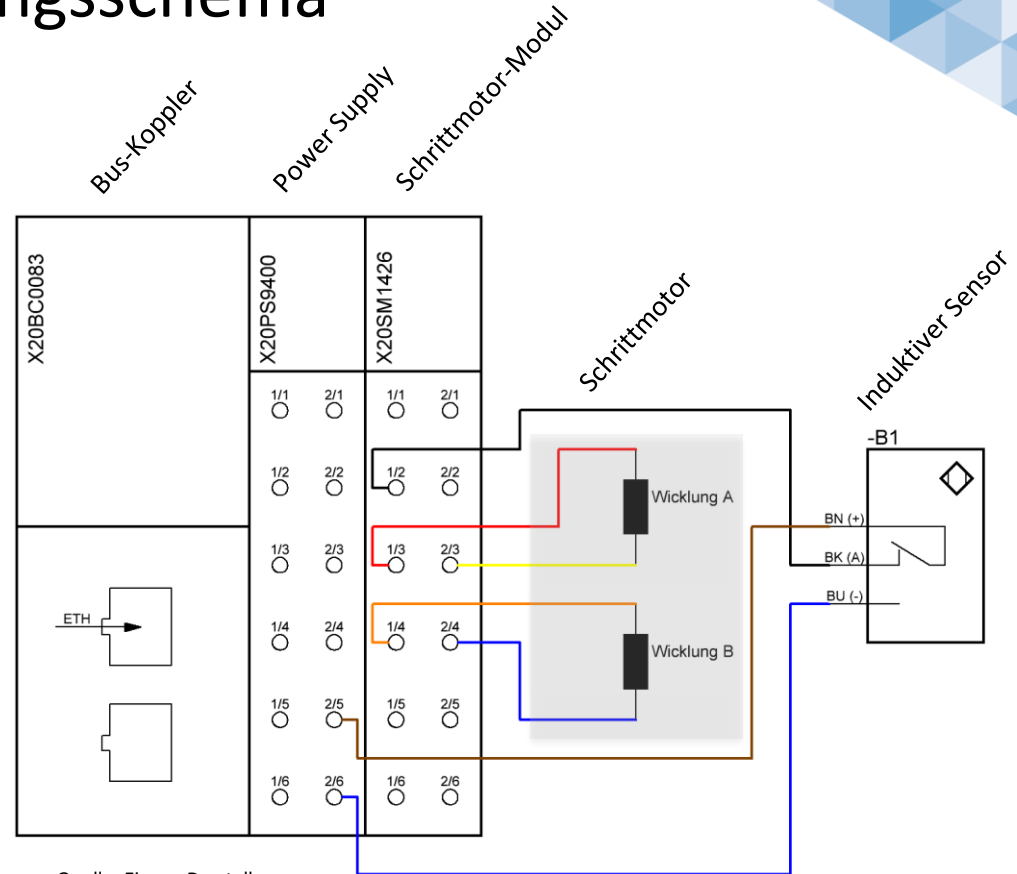
Quelle: Eigene Darstellung

Hardwarekonfiguration Aufgabenstellung 2



Quelle: Eigene Darstellung

Schrittmotor Verdrahtungsschema



Quelle: Datenblatt

Quelle: Eigene Darstellung

Berechnung der Schrittmotorkennwerte

$$n_{\text{Schritte}} := 200$$

$$\alpha_{\text{Schritt}} := \frac{360^\circ}{n_{\text{Schritte}}} = 1.8^\circ$$

$$U := 60 \quad 60 \text{ U/min} \quad 60 \frac{1}{\text{min}} = 1 \frac{1}{\text{s}} \quad 60 \text{ U/min sind } 1 \text{ U/s}$$

$$t := 60 \text{ s}$$

$$t_{\text{Zyklus}} := 0.025 \text{ s}$$

$$n_{\text{Schritte_Umdrehungen}} := U \cdot n_{\text{Schritte}} = 12000$$

Vollschritte bei 60 Umdrehungen

$$n_{\text{Mikroschritte}} := n_{\text{Schritte}} \cdot 256 = 51200$$

Mikroschritte pro Vollschritt

$$n_{\text{Mikroschritte_Umdrehungen}} := U \cdot n_{\text{Mikroschritte}} = 3072000$$

$$n_{\text{Zyklen_pro_Minute}} := \frac{t}{t_{\text{Zyklus}}} = 2400$$

Anzahl der Zyklen pro Minute

$$n_{\text{Mikroschritte_25ms}} := \frac{n_{\text{Mikroschritte_Umdrehungen}}}{n_{\text{Zyklen_pro_Minute}}} = 1280$$

Anzahl der Mikroschritte pro Minute

Quelle: Eigene Darstellung

I/O-Mapping

StepperMotor			
StateWord	UINT		
ControlWord	UINT		
Speed	DINT		
Mode	SINT		
MotorOn	BOOL		
State	DS402_State		
Control	DS402_Control		
Position_actual	DINT		

➔ AbsPos01	::Program_An:Motor1.Speed	DINT	Automatic	☐	☐	\\4PPC70_0702_20B\IoMap.iom	Absolute position 01
➔ MpGenControl01	::Program_An:Motor1.ControlWord	UINT	Automatic	☐	☐	\\4PPC70_0702_20B\IoMap.iom	MpGen control 01
➔ MpGenMode01	::Program_An:Motor1.Mode	SINT	Automatic	☐	☐	\\4PPC70_0702_20B\IoMap.iom	MpGen mode 01
➔ AbsPos01ActVal	::Program_An:Motor1.Position_actual	DINT	Automatic	☐	☐	\\4PPC70_0702_20B\IoMap.iom	Absolute position 01 actual value
➔ MpGenStatus01	::Program_An:Motor1.StateWord	UINT	Automatic	☐	☐	\\4PPC70_0702_20B\IoMap.iom	MpGen status 01

Programmcode

```
PROGRAM _CYCLIC
    //Motor Status einlesen
    Motor1.State := GetMotorState(Motor1.StateWord);

    //Motor Status setzen
    Motor1.Mode := 2;

    //Motor Drehzahl setzen
    Motor1.Speed := Drehzahl/5;

    //Motor einschalten wenn Schalter2 "1"
    Motor1.MotorOn := Schalter2;

    //Motor Status aktualisieren
    MotorOnOff(Motor1);

END_PROGRAM
```

Programmcode Zusatz 1

```
FUNCTION MotorOnOff
```

```
    IF Motor.MotorOn = TRUE THEN
```

```
        CASE Motor.State OF
```

```
            SwitchOnDisabled: SetMotorCommand(Shutdown, Motor);
```

```
            ReadyToSwitchOn: SetMotorCommand(SwitchOn, Motor);
```

```
            SwitchedOn: SetMotorCommand(EnableOperation, Motor);
```

```
        END_CASE;
```

```
    ELSE
```

```
        SetMotorCommand(DisableVoltage, Motor);
```

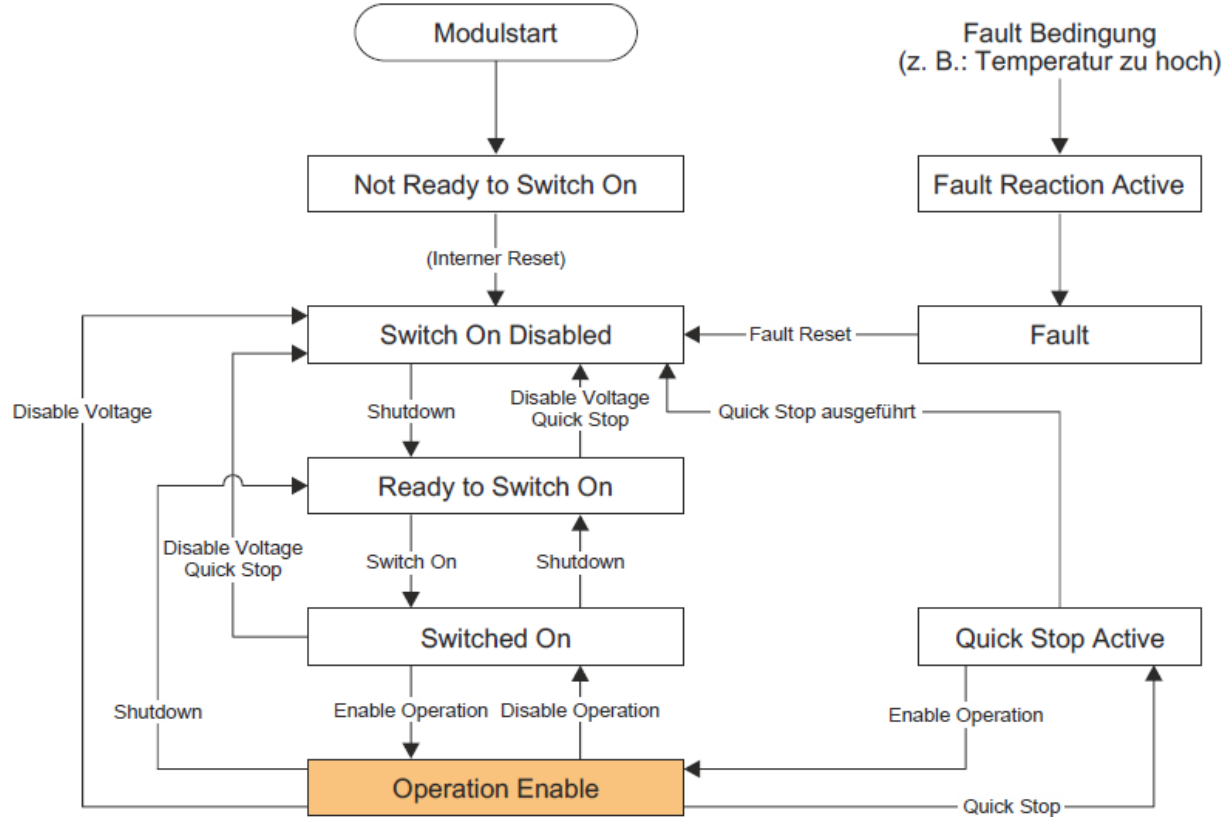
```
    END_IF;
```

```
END_FUNCTION
```

Betriebsmodi - Schrittmotoransteuerung

- Modus 1: Positionierung
- Modus 2: Konstante Geschwindigkeit
- (+ diverse Referenziermodi)

Schrittmotoransteuerung



Quelle: X20SM1426 Datenblatt V 1.32, Hrsg. Bernecker + Rainer Industrie-Elektronik Ges.m.b.H., S 48

Statuswort

Die einzelnen Bits dieses Registers und deren Zustände sind abhängig vom gerade aktiven Zustand der State Machine:

Status	Reserved / MotorLoadBit 2 ¹⁾	Reserved / MotorLoadBit 1 ¹⁾	Reserved / MotorLoadBit 0 ¹⁾	Reserved	Int. Limit Active	Target Reached	Remote	Reserved	Warning	Switch On Disabled	Quick Stop	Voltage Enabled	Fault	Operation Enabled	Switched on	Ready to switch on
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Not ready to switch on	x	x	x	x	x	x	1	0	x	0	x	0	0	0	0	0
Switch On Disabled	x	x	x	x	x	x	1	0	x	1	x	0	0	0	0	0
Ready to switch on	x	x	x	x	x	x	1	0	x	0	1	0	0	0	0	1
Switched on	x	x	x	x	x	x	1	0	x	0	1	1	0	0	1	1
Operation Enable	x	x	x	x	x	x	1	0	x	0	1	1	0	1	1	1
Quick Stop active	x	x	x	x	x	x	1	0	x	0	0	1	0	1	1	1
Fault Reaction active	x	x	x	x	x	x	1	0	x	0	x	0	1	1	1	1
Fault	x	x	x	x	x	x	1	0	x	0	x	0	1	0	0	0

- 1) Wenn Bit 7 im Register "Mixed Decay / Stall Detection" auf Seite 40 Konfiguration auf 1 gesetzt wird, wird in Bit 13 bis 15 von Statuswort der Motor-Load-Wert zurückgegeben, ansonsten sind diese Bits immer 0.

Quelle: X20SM1426 Datenblatt V 1.32, Hrsg. Bernecker + Rainer Industrie-Elektronik Ges.m.b.H., S 46

Funktion - GetMotorState

```
//Not Ready Switch On
```

```
IF NOT statusWord.0 AND      //Abfrage Bit 0 "0"
    NOT statusWord.1 AND    //Abfrage Bit 1 "0"
    NOT statusWord.2 AND    //Abfrage Bit 2 "0"
    NOT statusWord.3 AND    //Abfrage Bit 3 "0"
    NOT statusWord.4 AND    //Abfrage Bit 4 "0"
    NOT statusWord.6 THEN   //Abfrage Bit 6 "0"
    GetMotorState := NotReadySwitchOn;
END_IF;
```

Status	Switch On Disabled	Quick Stop	Voltage Enabled	Fault	Operation Enabled	Switched on	Ready to switch on
Bit	6	5	4	3	2	1	0
Not ready to switch on	0	x	0	0	0	0	0

Steuerwort

Die Bits des Steuerworts und deren Zustand für die Kommandos der State Machine:

Kommando	Stall Detection	Encoder position sync/async	Current error detection	Warning Reset	Motor ID trigger	Reserved	Reserved	Stop	Fault Reset	Mode Specific	Mode Specific	Mode Specific	Enable Operation	Quick Stop	Enable Voltage	Switch On
Bit ¹⁾	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Shutdown	x	x	x	x	x	0	0	x	0	x	x	x	x	1	1	0
Switch On	x	x	x	x	x	0	0	x	0	x	x	x	0	1	1	1
Disable Voltage	x	x	x	x	x	0	0	x	0	x	x	x	x	x	0	x
Quick Stop	x	x	x	x	x	0	0	x	0	x	x	x	x	0	1	x
Disable Operation	x	x	x	x	x	0	0	x	0	x	x	x	0	1	1	1
Enable Operation	x	x	x	x	x	0	0	x	0	x	x	x	1	1	1	1
Fault Reset	x	x	x	x	x	0	0	x	↑	x	x	x	x	x	x	x

1) x ... beliebig; ↑ ... Steigende Flanke

Quelle: X20SM1426 Datenblatt V 1.32, Hrsg. Bernecker + Rainer Industrie-Elektronik Ges.m.b.H., S 46

Funktion - SetMotorCommand

```

FUNCTION SetMotorCommand
    controlWord := Motor.ControlWord;

    // Command bearbeiten
    CASE command OF
        Shutdown:    controlWord.0 := FALSE; //Bit 0 "0" setzen
                     controlWord.1 := TRUE;  //Bit 1 "1" setzen
                     controlWord.2 := TRUE;  //Bit 2 "1" setzen
                     controlWord.7 := FALSE; //Bit 7 "0" setzen

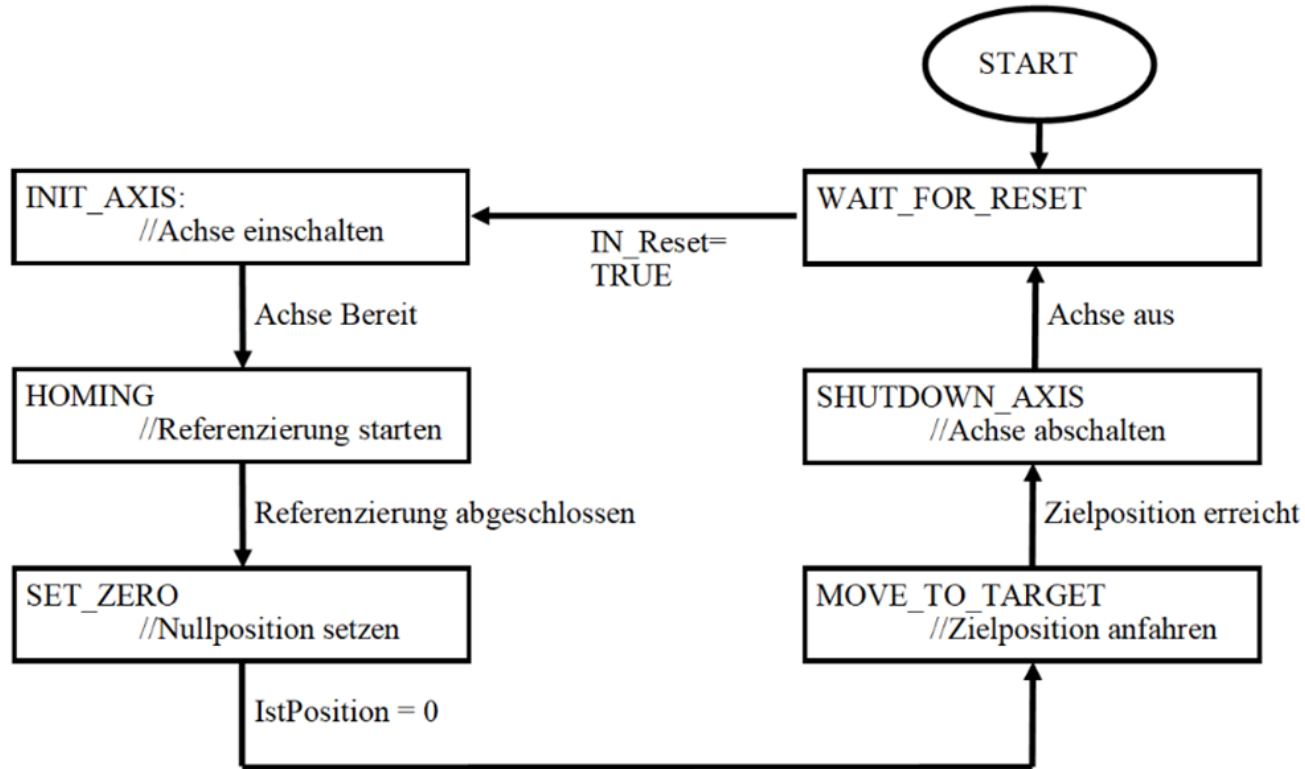
    END_CASE;

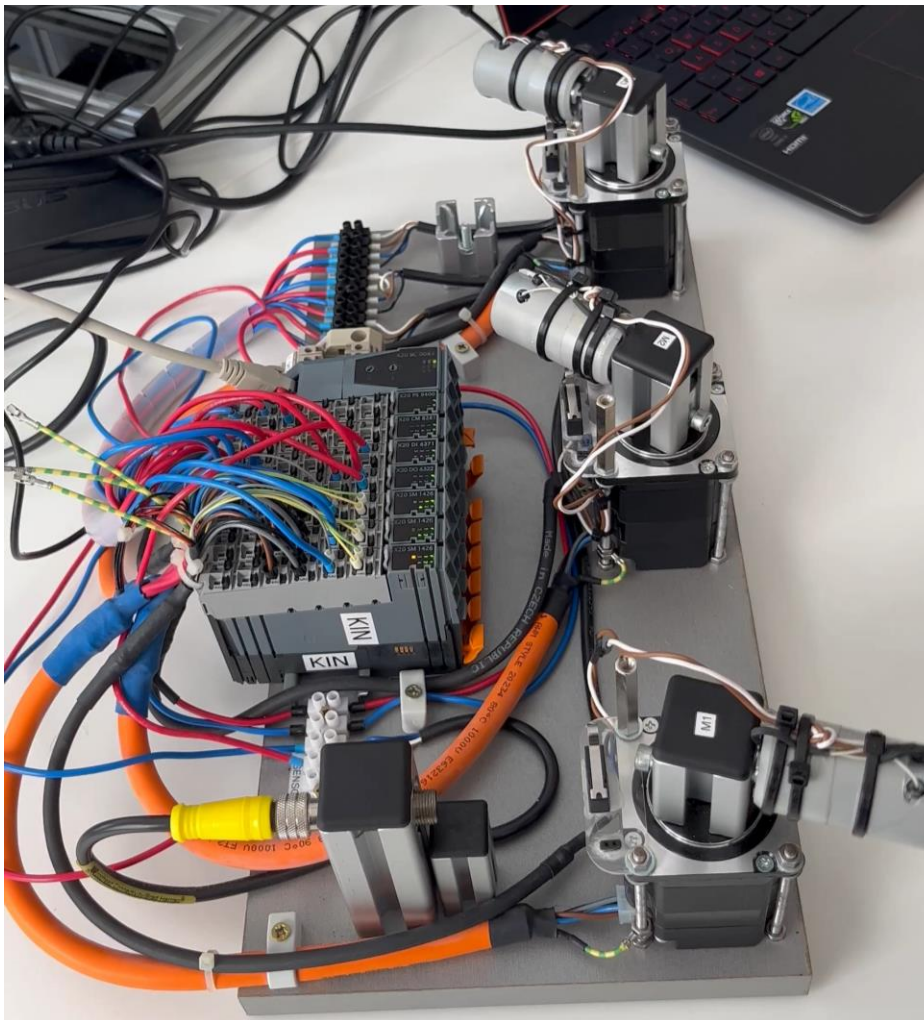
    //Rückgabe des Control Word
    Motor.ControlWord := controlWord;

END_FUNCTION
    
```

Kommando	S	E	O	N	M	R	R	S	E	M	M	M	E	O	E	S
Bit ¹⁾	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Shutdown	x	x	x	x	x	0	0	x	0	x	x	x	x	1	1	0
Switch On	x	x	x	x	x	0	0	x	0	x	x	x	0	1	1	1

Ablaufdiagramm





Danke für Ihre Aufmerksamkeit!

FH CAMPUS 02, Automatisierungstechnik

Anton Holzerbauer

Christian Rinner