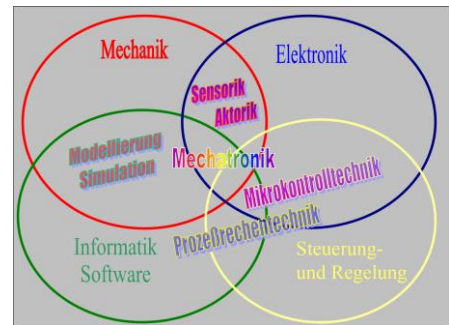


Labor Mechatronik 2



Studiengang: Ingenieurinformatik
Semester: Sommersemester 2019
Lehrgebiet: Mechatronik 2

Versuchsprotokoll

Versuch 3

Zweipunktregler

Datum: 24.06.2019
Praktikumsgruppe: A
Praktikumsversuchsleiter: Prof. Dr. Nils Siebel
Versuchsbetreuer: Prof. Dr. Nils Siebel

Unterschrift:

Student*n: Nsikam Onla Brice

Versuchsziel & Theoretische Grundlagen

Ziel des Versuches ist es, ein Programm in der Sprache ST (Anzeigen und Regler) und FUP zu konstruieren, lesen wir den Istwert vom Analogeingang, an dem der Temperaturmessumformer mit dem Pt100-Widerstand hängt. Der Sollwert ist eingeben bei dem Benutzer (Visualisierung), die Beiden werte „Istwert“ und Sollwert sind vergleicht.

Ist der Istwert größer als der Sollwert, leuchtet die hellblaue SPS auf, und wenn der Benutzer die Taste + in der Visualisierung drückt, addieren Sie 0,5 Kelvin zum Sollwert, und analog zum Drücken der Taste - reduziert sich der Sollwert auf 0,5 Kelvin. Unabhängig vom Sollwert wird der Wert des Istwertes auf dem 7-Segment angezeigt aber der Sollwert kann genauso bei Eingeben Feld in der Visualisierung eingeben.

Verwendete Geräte und Schnittstellen:

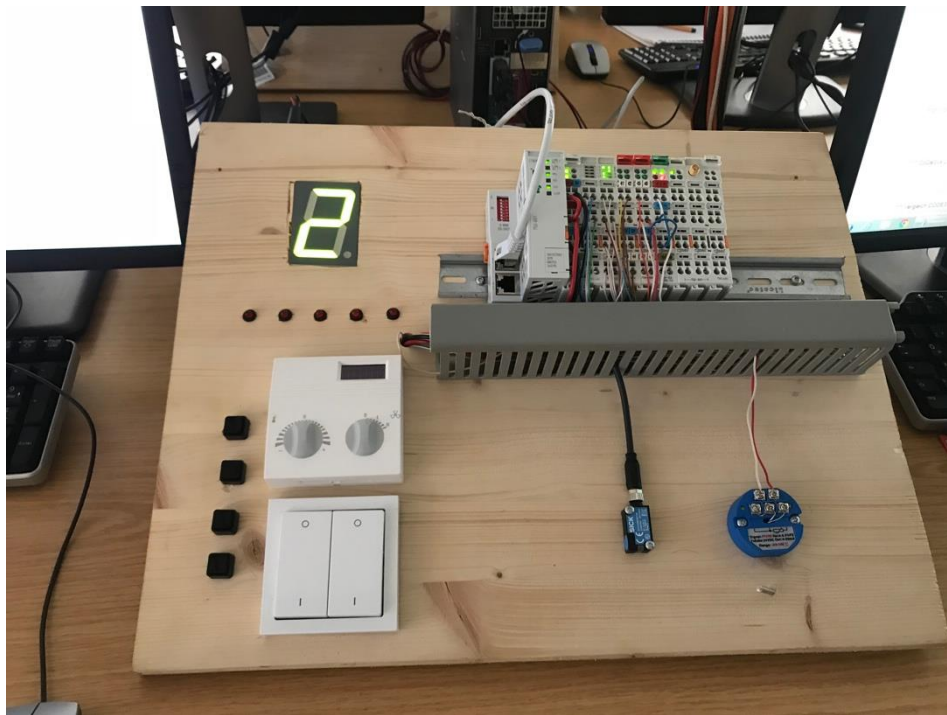


Fig1. Setup der Versuch

Flussdiagramm der Versuch:

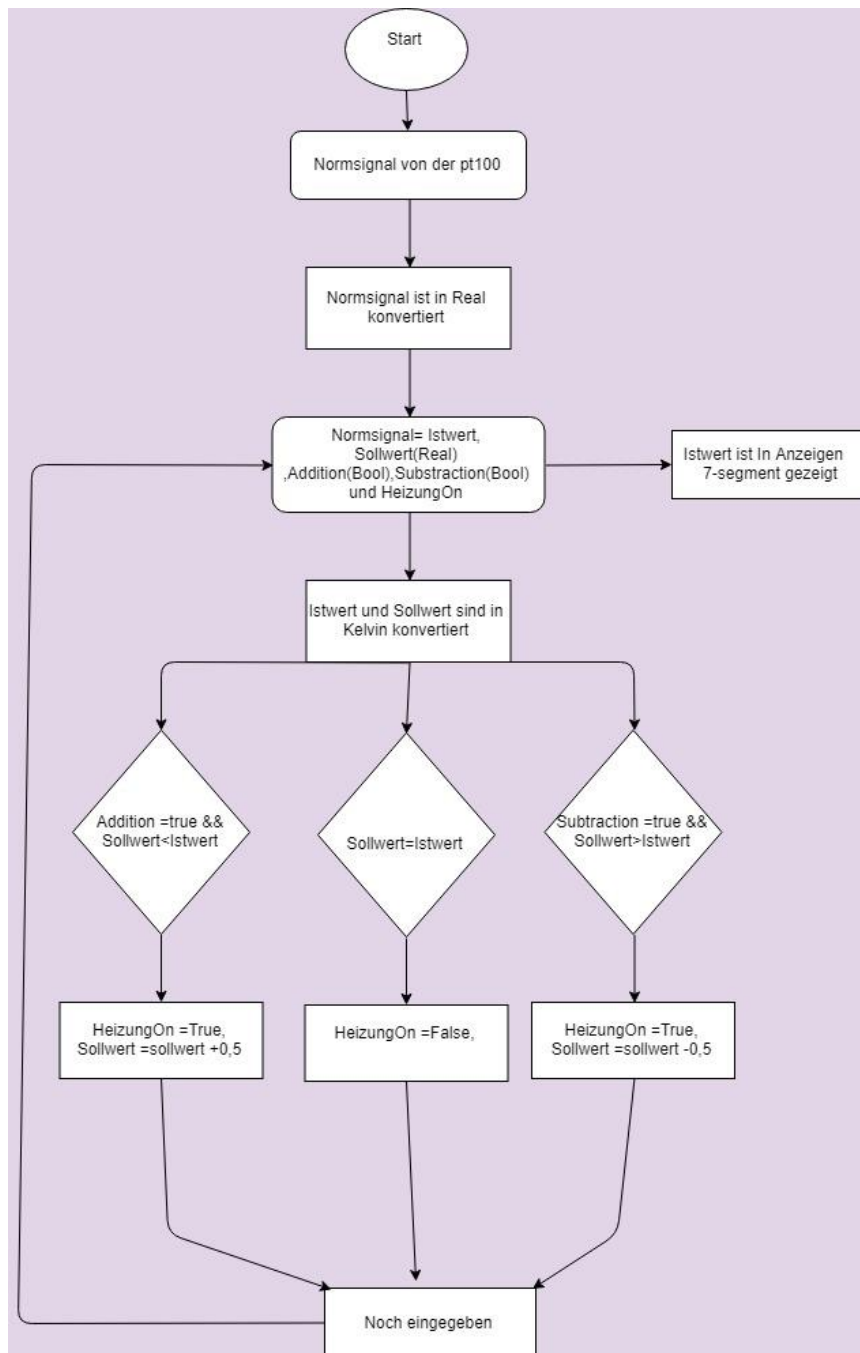
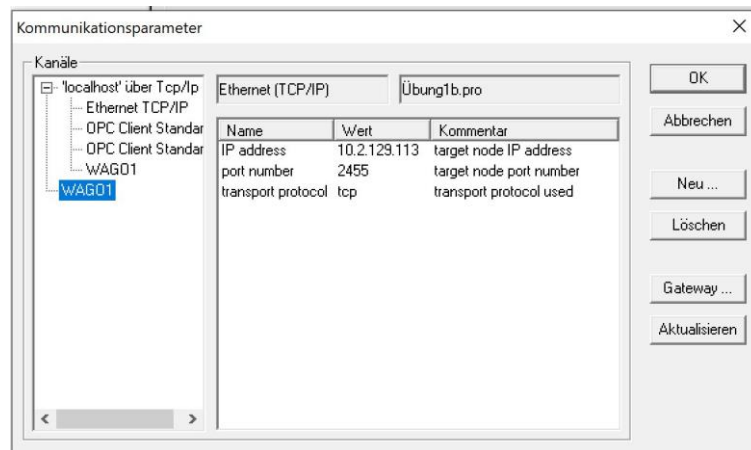


Fig1. Flussdiagramm



Wago 750-881, Mac-Adresse : **00:30:DE:0A:1B:51**

Fig.2.2 Information Wago und Mac

K-Bus Konfiguration mit Systemvariablen

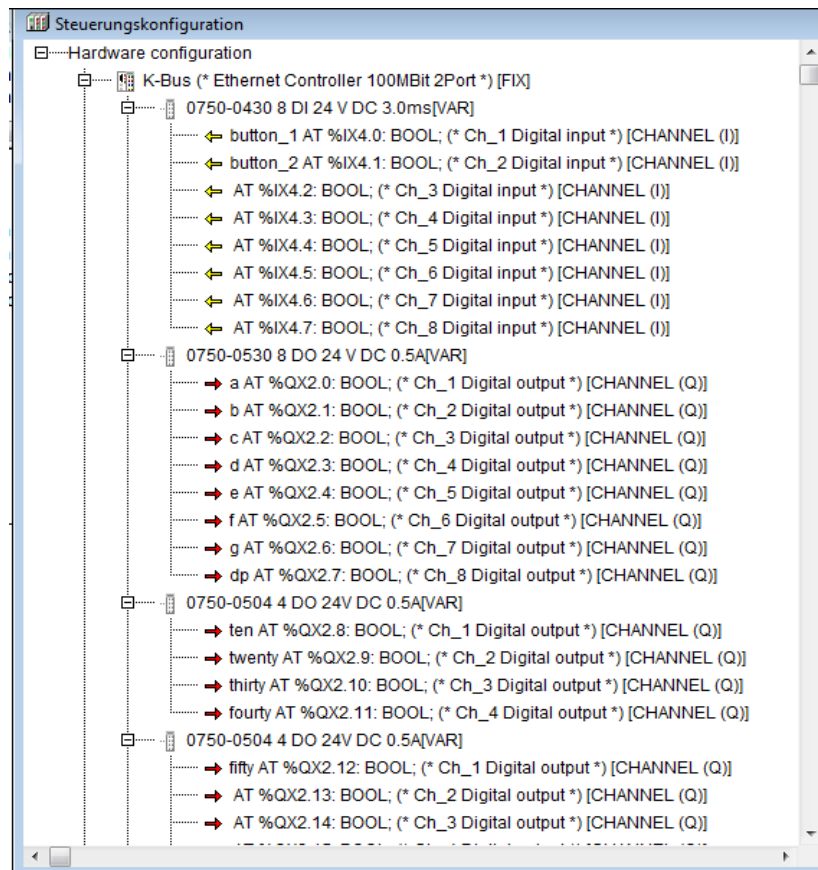


Fig3. Konfiguration Bus Protokolle 1

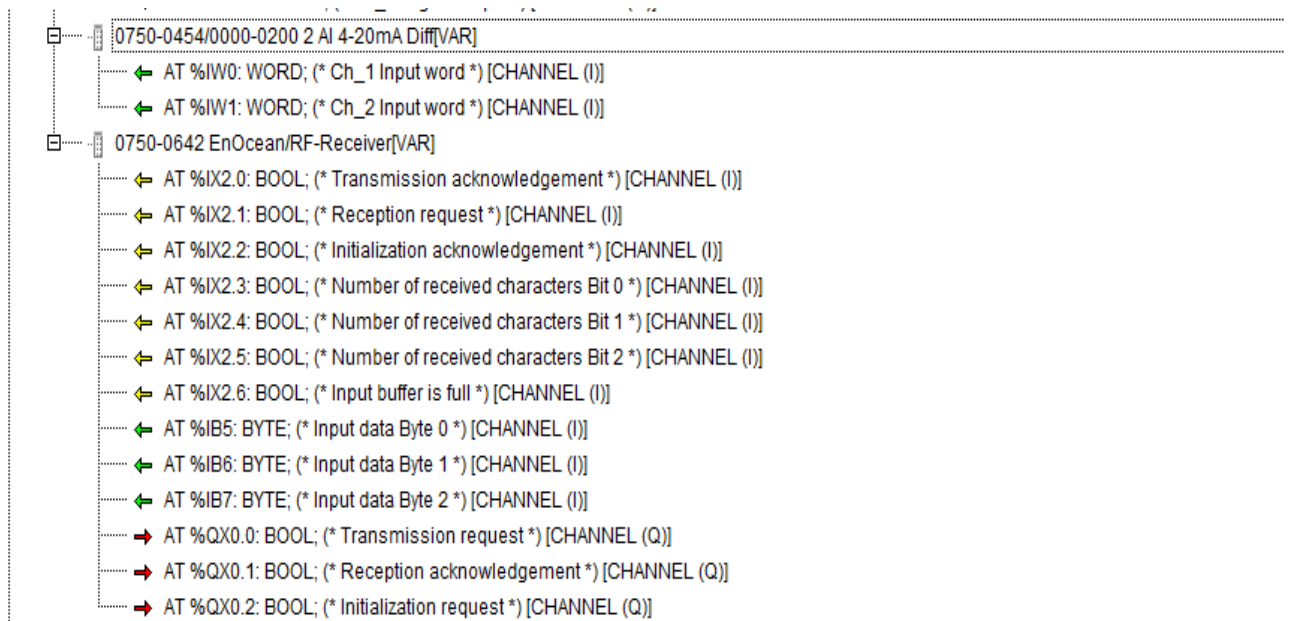


Fig4. Konfiguration Bus Protokolle 2

Programmcode

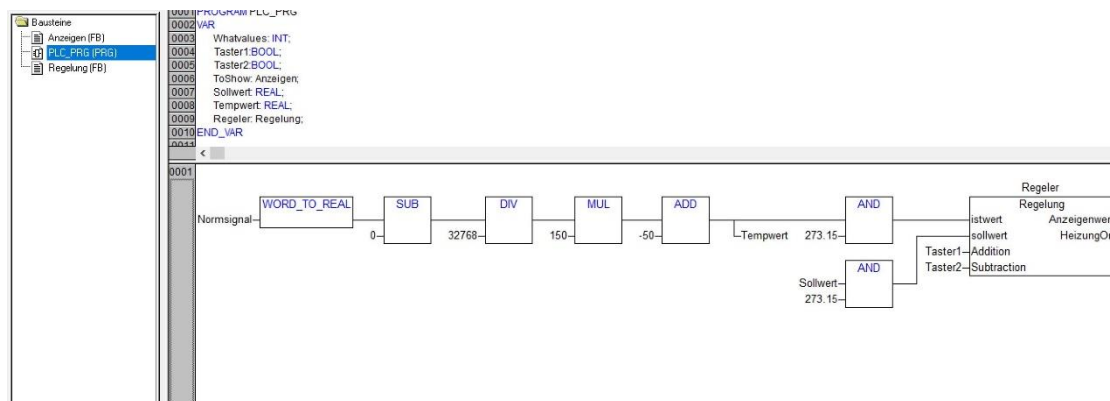


Fig5. Modul des Hauptprogramm mit Variablendeklaration

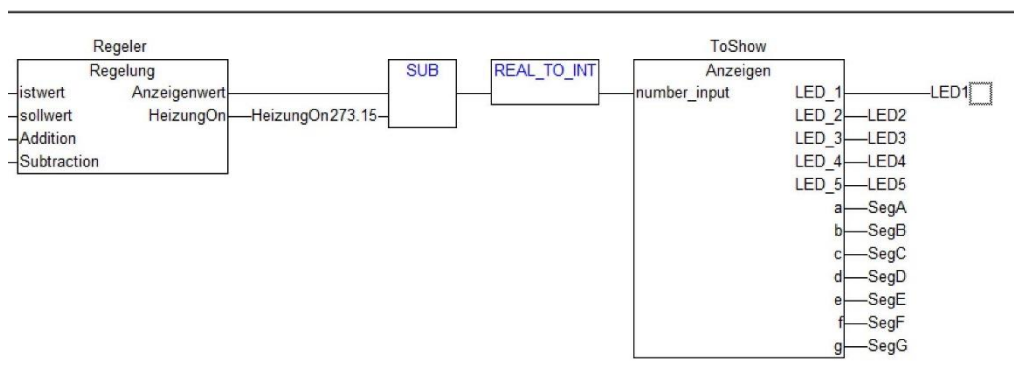


Fig6. Weiter Modul des Hauptprogramm mit Variablendeklaration

Bausteine

- Anzeigen (FB)
- PLC_PRG (PRG)
- Rechnenzahl (FB)

```

0001 FUNCTION_BLOCK Anzeigen
0002 VAR_INPUT
0003 <
0004 values := number_input / 10; (*Show a number*)
0005 restofvalues := number_input MOD 10;
0006
0007 LED_1:=LED_2:=LED_3:=LED_4:=LED_5:=FALSE;
0008
0009 CASE values OF
0010 1: LED_1:= TRUE;
0011 2: LED_1:= TRUE;
0012 LED_2:= TRUE;
0013 3: LED_1:= TRUE;
0014 LED_2:= TRUE;
0015 LED_3:= TRUE;
0016 4: LED_1:= TRUE;
0017 LED_2:= TRUE;
0018 LED_3:= TRUE;
0019 5: LED_1:= TRUE;
0020 LED_2:= TRUE;
0021 LED_3:= TRUE;
0022 LED_4:= TRUE;
0023 END_CASE;
0024
0025 a:=b:=c:=d:=e:=f:=g:=FALSE;
0026
0027 CASE restofvalues OF
0028 0: a:= b:= c:= e:= d:= f:= TRUE;
0029 1: b:= c:= TRUE;
0030 2: a:= b:=g:=e:=d:= TRUE;
0031 3: a:=b:=g:=c:=d:= TRUE;
0032 4: f:=b:=g:=c:= TRUE;
0033 5: a:= f:=g:=c:=d:= TRUE;
0034 6: a:=f:=g:=c:=d:=e:= TRUE;
0035 7: a:=b:=c:= TRUE;
0036 8: a:=b:= c:=d:=e:= f:=g:= TRUE;
0037 9: a:=b:= g:=f:=c:=d:= TRUE;
0038 END_CASE;
0039
0040

```

Fig7. Programmcode des „Anzeigen“-Moduls

Bausteine

- Anzeigen (FB)
- PLC_PRG (PRG)
- Regelung (FB)

```

0001 FUNCTION_BLOCK Regelung
0002 VAR_INPUT
0003 istwert: REAL;
0004 sollwert: REAL;
0005 Addition:BOOL;
0006 Subtraction:BOOL;
0007 END_VAR
0008 VAR_OUTPUT
0009 Anzeigenwert:REAL;
0010 HeizungOn: BOOL;
0011 END_VAR
0012 VAR
0013 END_VAR
0014 <
0015
0016 Anzeigenwert:=istwert;(* Anzeigenwert Take the Istwert *)
0017
0018 IF (sollwert > istwert-1.2) THEN
0019 HeizungOn:= TRUE;
0020 END_IF
0021
0022 IF (sollwert < istwert+1.2) THEN
0023 HeizungOn:= FALSE;
0024
0025
0026 IF (Addition=TRUE) THEN
0027 HeizungOn:= TRUE;
0028 sollwert:= sollwert+0.5;
0029 END_IF
0030
0031 IF (Subtraction=TRUE) THEN
0032 HeizungOn:= TRUE;
0033 sollwert:= sollwert-0.5;
0034 END_IF
0035
0036
0037
0038
0039
0040

```

Fig8. Modul „Reglung“ Variablendeklaration & Programmcode

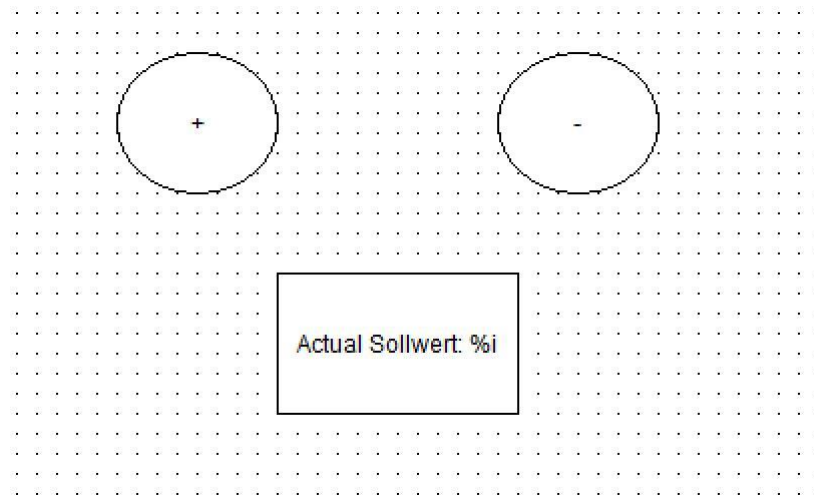


Fig9. Visualisierung des Prorammmcodes