Protokoll

Projekt: industriepc_lbulic_msimov_soezturk_gsina

Repo kann man aufrufen unter: https://github.com/lbulic-tgm/industriepc_lbulic_msimov_soezturk_gsina

Schritt 1 – Dokumentation & Hardwareanalyse

Zunächst wurde die vorhandene Dokumentation der Beckhoff-Hardware recherchiert und studiert. Ziel war es, die eingesetzten I/O-Karten zu identifizieren und deren Funktion (Input / Output) festzulegen.

Ergebnis

- **Output-Karten (Ausgänge):**
  - EL2004
  - EL2008
- **Input-Karten (Eingänge):**
  - EL1008
  - EL1104
  - EL1104

Zuordnung:

- Button (Taster) → Input
- Lampe → Output

Diese Zuordnung ist notwendig für das spätere Hardware-Mapping in TwinCAT.

Schritt 2 – Verbindung des Industrie-PCs

Der Industrie-PC wurde über **LAN** mit dem Netzwerk verbunden, da WLAN für industrielle Anwendungen als zu instabil gilt.

Problem

- Das verwendete LAN-Kabel funktionierte nicht korrekt

- Die Netzwerkverbindung konnte nicht hergestellt werden

Maßnahme

- Beim nächsten Versuch wird ein geprüftes, funktionsfähiges LAN-Kabel verwendet

---

Schritt 3 – Ermitteln der IP-Adresse

Nach der Netzwerkverbindung wurde die IP-Adresse des Industrie-PCs ermittelt.

Verwendeter Befehl

```
ip addr show
```

Schritt 4 – Erstellen der Authentifizierungsdatei

Für den Zugriff auf Beckhoff-Paketquellen wurde eine Authentifizierungsdatei angelegt.

Befehl

```
sudo nano /etc/apt/auth.conf.d/bhf.conf
```

Schritt 5 – Installation von TwinCAT (XAR)

Nach der Vorbereitung wurde TwinCAT Runtime unter Linux installiert.

Befehle

```
sudo apt update
sudo apt install tc31-xar-um
```

Verwendete Dokumentation

- Beckhoff_RT_Linux_de.pdf (Embedded PC CX)

Schritt 6 – SPS-Programmierung (TwinCAT)

Ziel

Mit einem **Taster** wird eine **Lampe getoggelt**.
Bei jeder **steigenden Flanke** des Tasters ändert die Lampe ihren Zustand (Ein/Aus).

Funktionsprinzip

- Der Taster wird als digitale Eingangsvariable gemappt
- Die Lampe wird als digitale Ausgangsvariable gemappt
- Die steigende Flanke wird mit dem Funktionsbaustein `R_TRIG` erkannt
- Bei erkannter Flanke wird der Zustand der Lampe invertiert

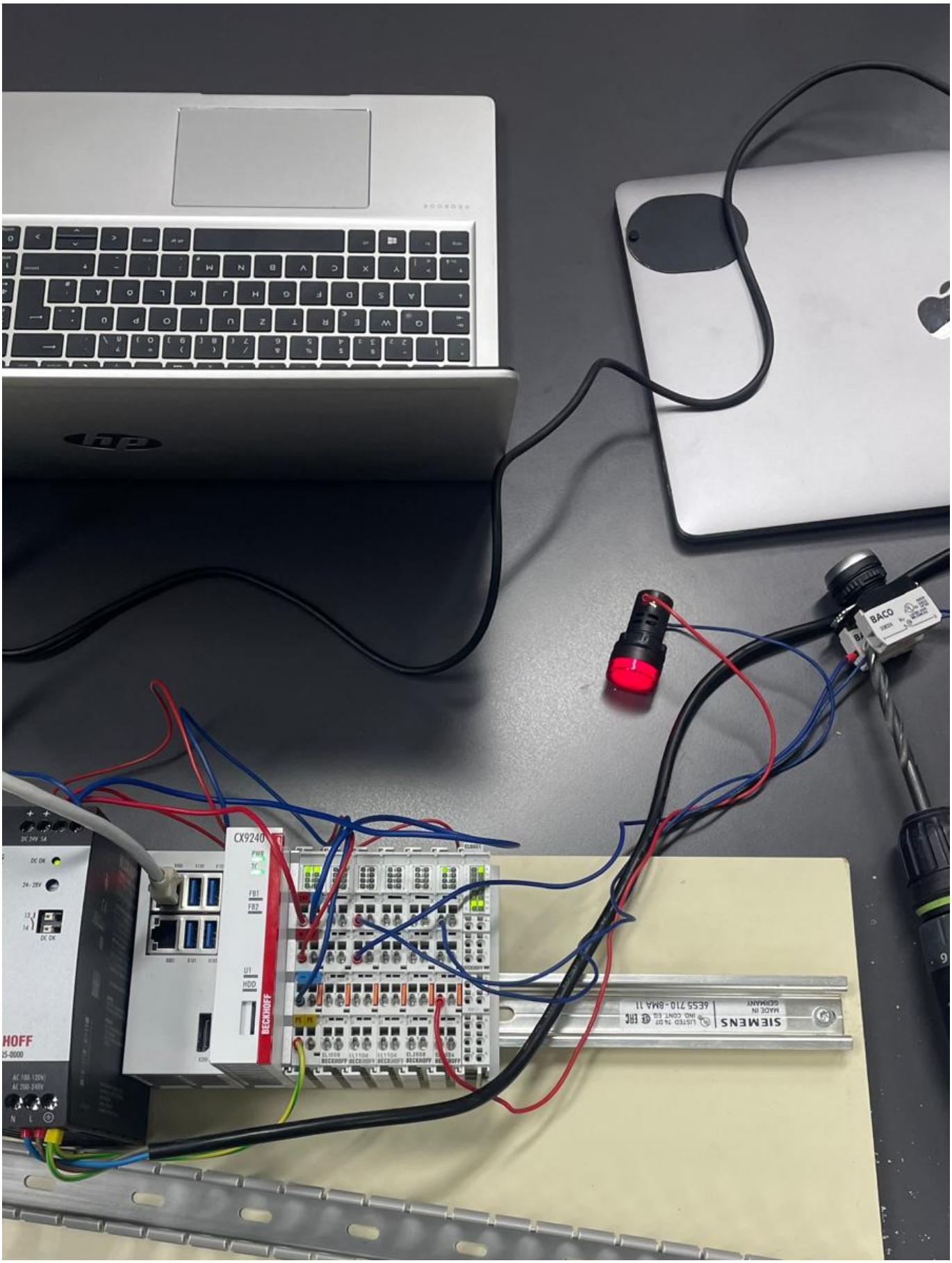
SPS-Code (Structured Text)

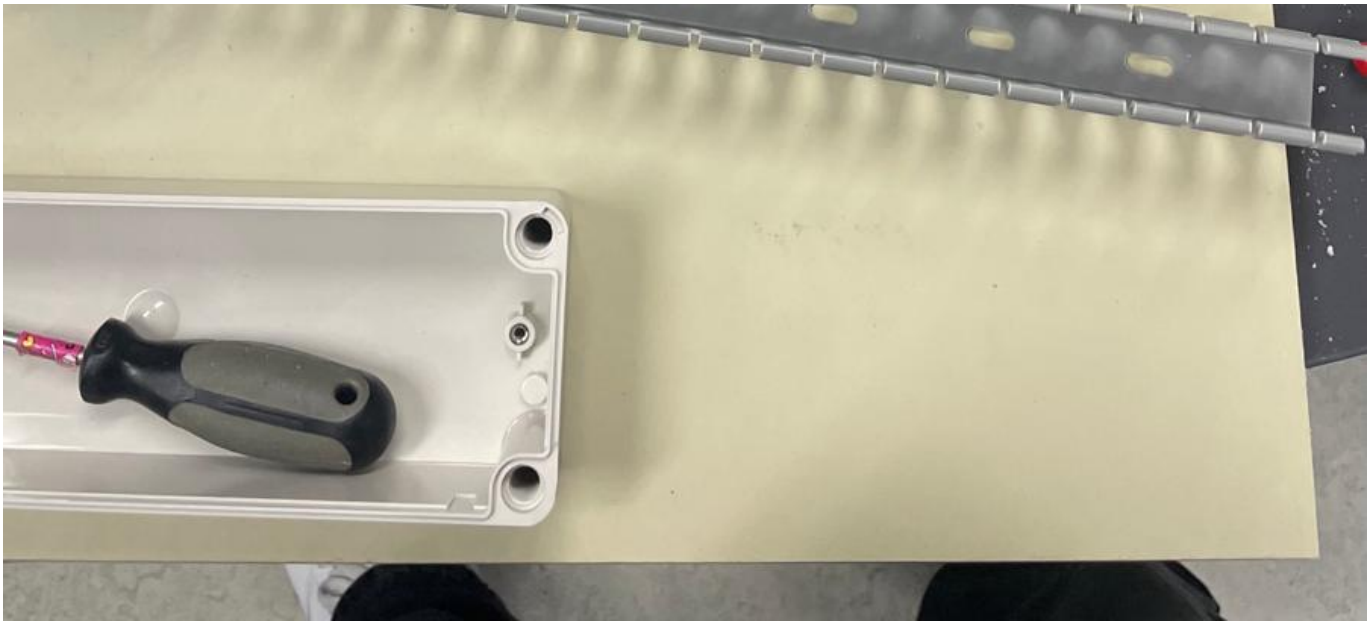
```
PROGRAM MAIN
VAR
    // Hardware-Variablen (Namen bleiben für das Mapping erhalten)
    bKnopf  AT %I* : BOOL;
    bLampe  AT %Q* : BOOL;

    // Instanz zur Erkennung der steigenden Flanke
    fbTasterFlanke : R_TRIG;
END_VAR

// Aktuellen Zustand des Tasters an den Flankenerkennungs-Baustein übergeben
fbTasterFlanke(CLK := bKnopf);

// Wenn eine positive Flanke erkannt wurde
IF fbTasterFlanke.Q THEN
    // Lampenzustand umschalten
    bLampe := NOT bLampe;
END_IF
```

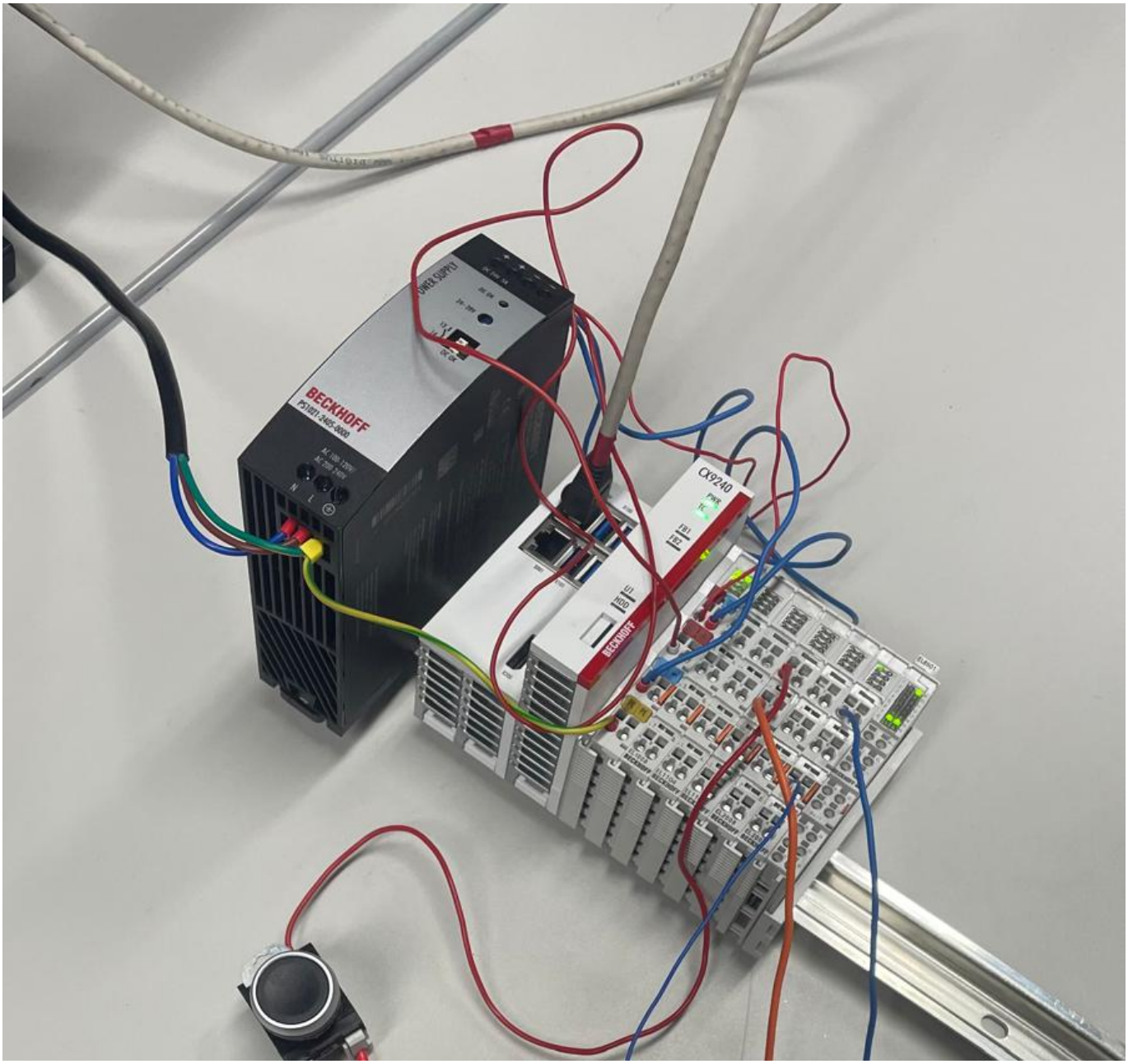
Ergebnis

- Hardware-Zuordnung (Input / Output) erfolgreich analysiert
- TwinCAT Runtime unter Linux installiert

- Funktionsfähiges SPS-Programm zur Taster-Lampen-Steuerung erstellt
- Flankenerkennung verhindert Mehrfachschaltungen beim Gedrückthalten des Tasters






BECKHOFF
CX9240
BECKHOFF