

Praktikum von 03.01.2023 – 27.01.2023

Contents

Einarbeitung in TwinCAT 3 SPS Programmierung	1
Installation von TwinCAT 3 Engineering	1
Programmierung eines Blinkers	1
Einarbeitung in OOP	2
Einarbeitung in TwinCAT 3 HMI Programmierung	2
Installation TwinCAT 3 HMI Engineering	2
Visualisierung des Blinkers	2
Visualisierung der Lichtbausteine durch UserControls	2
Einarbeitung in HTML/CSS/TypeScript Programmierung	3
Einarbeitung in HTML	3
Einarbeitung in CSS	3
Einarbeitung in JavaScript/TypeScript	3
Einarbeitung in TcHmi Framework-Controls	3
Programmierung des Blinkers als Framework-Control	3
Programmierung der Licht Controls als Framework-Control	3
Programmierung DeviceState Control	3

Einarbeitung in TwinCAT 3 SPS Programmierung

In diesem Teil soll sich mit der TwinCAT SPS Programmierung befasst werden.

Ziel ist es einen Überblick darüber zu bekommen, wie eine SPS mit TwinCAT programmiert wird und was es für verschiedene Möglichkeiten gibt.

[Hier](#) gibt es ein komplettes Youtube-Tutorial, das sich mit TwinCAT 3 SPS Programmierung auseinandersetzt.

Installation von TwinCAT 3 Engineering

Bevor mit dem Einarbeiten begonnen wird, muss das [TwinCAT 3 Engineering](#) heruntergeladen und installiert werden.

Programmierung eines Blinkers

Es soll ein einfacher Blinker programmiert werden, der über eine Variable an- und ausgeschaltet werden kann.

Hilfen:

- [Hier](#) gibt es ein Youtube-Tutorial
- Folgende Beispiele sollten durchgearbeitet werden:
 - o https://infosys.beckhoff.com/content/1031/tc3_sampleprogram1/9007206902055819.html?id=4942866403849214565
 - o https://infosys.beckhoff.com/content/1031/tc3_sampleprogram1/9007206902056971.html?id=3841487983518146203

Einarbeitung in OOP

Es soll ein Überblick über die Möglichkeiten der objekt-orientierte Programmierung in TwinCAT geschaffen werden.

Aufgabe ist ein Basis Licht Funktionsblock zu implementieren, der lediglich das Ein- bzw. Ausschalten des Lichtes realisiert. Von diesem Baustein soll ein erweiterte Licht Funktionsblock erben, der neben dem einfach Ein- bzw. Ausschalten auch einen bestimmten Helligkeitswert annehmen kann.

Für beide Bausteine sollen zunächst Interfaces implementiert werden.

Hilfen:

- [Beispiel](#) für OOP
- Aus der Tutorial-Serie von oben könnte [dieses](#) Tutorial helfen

Einarbeitung in TwinCAT 3 HMI Programmierung

Nachdem ein Überblick über die SPS Programmierung mit TwinCAT 3 vorhanden ist, soll nun ein Überblick über die TwinCAT 3 HMI Programmierung geschaffen werden.

Installation TwinCAT 3 HMI Engineering

Das TwinCAT 3 HMI (TcHmi) Engineering [herunterladen](#) und installieren.

[Hier](#) gibt es Hilfen zu Installation des TcHmi Engineerings.

Visualisierung des Blinkers

Es soll der Blinker von oben visualisiert werden. Neben dem Blinker soll es einen Button geben, der den Blinker ein- bzw. ausschaltet.

[Hier](#) gibt es eine Tutorial-Serie über die TcHmi.

Visualisierung der Lichtbausteine durch UserControls

Die Lichtbausteine von oben sollen durch zwei [UserControls](#) visualisiert werden.

Im UserControl soll ein einfaches Rechteck dazu verwendet werden, die Helligkeit des Lichtes anzuzeigen. Daneben soll es auch eine textuelle Anzeige der Helligkeit geben.

Steuern lassen soll sich das Licht über einen ToggleButton bzw. einen Slider.

Einarbeitung in HTML/CSS/TypeScript Programmierung

Als nächstes sollen die UserControls von oben als Framework-Controls implementiert werden. Dazu ist es notwendig sich in die HTML, CSS und TypeScript Programmierung einzuarbeiten.

Installation von Visual Studio Code

Einarbeitung in HTML

Framework-Controls bzw. alle Webanwendungen bestehen aus HTML-Elementen. Auch die Komponenten in unseren Framework-Controls sind HTML-Elemente. Daher ist es wichtig einen Überblick über HTML zu bekommen. [Hier](#) gibt es ein Tutorial.

Einarbeitung in CSS

Mit TypeScript wird die Logik von einem Framework-Control implementiert. Mit CSS wird die visuelle Darstellung von HTML-Elementen beeinflusst. [Hier](#) gibt es ein Tutorial.

Einarbeitung in JavaScript/TypeScript

Tutorials:

- [JavaScript](#)
- [TypeScript](#)

Einarbeitung in TcHmi Framework-Controls

Nun soll ein Überblick über die Framework-Control-Entwicklung mit TcHmi geschaffen werden.

Installation Visual Studio Community

[Hier](#) gibt es ein Youtube-Video, dass den Unterschied zwischen UserControls und FrameworkControls erklärt.

[Hier](#) ist die Dokumentation bzgl. FrameworkControls.

Programmierung des Blinkers als Framework-Control

Das Blinker UserControl von oben soll als Framework-Control implementiert werden

Programmierung der Licht Controls als Framework-Control

Die Licht Controls von oben sollen als Framework-Control implementiert werden.

Hier soll ebenfalls mit OOP gearbeitet werden.

Programmierung DeviceState Control

Für unser Produkt wird ein Control benötigt, welches den Status von allen verbundenen Steuerungen anzeigt.

Das Control soll sich als Button darstellen, der je nach Status der Controls verschiedene Farben animmt.

- Grün → alles in Ordnung
- Gelb → Verbindung kurzweilig verloren
- Rot → Verbindung komplett verloren

Wird auf den Button geklickt, soll sich das Diagnose-Fenster der BaSiteExtension öffnen.

