

Bachelorarbeit

Amalia Böhm Correcher

Studiengang Informatik

Hochschule Kempten

Thema

Integration eines Terminal-
Emulators in die Webapp CODESYS
go! mit serverseitiger Shell-
Anbindung und Dateiübertragung

Kurzbeschreibung des Themas

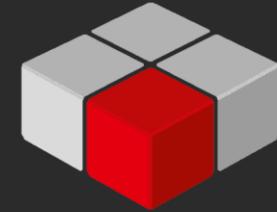
- > Ziel ist es, eine serverseitige Shell-Anbindung
- > zur Ausführung von Befehlen bereitzustellen.
- > Dazu soll eine Analyse und Integration eines
- > Terminal-Emulators in die Webanwendung CODESYS
- > go! unter Einbindung einer sicheren
- > serverseitigen Shell-Prozesssteuerung
- > durchgeführt werden.
- > Zusätzlich wird eine Dateiübertragungsfunktion
- > implementiert, um den bidirektionalen Austausch
- > von Dateien zwischen Client und Server zu
- > ermöglichen.

Personen

- > Betreuer (Unternehmen): Markus Schaber
m.schaber@codesys.com
- > Betreuer (HS Kempten): Prof. Dr. Arnulf Deinzer
Fakultät Informatik
arnulf.deinzer@hs-kempten.de
- > Bachelorandin: Amalia Böhm Correcher
amalia.m.correcher@stud.hs-kempten.de
a.boehm@codesys.com

CODESYS Development GmbH

A member of the CODESYS Group



- > führende herstellerunabhängige IEC 61131-3 Automatisierungssoftware
- > Eckdaten: Hauptsitz: Kempten
 Gründung: 1994
 Mitarbeiter: 230
- > Kontakt: E-Mail: info@codesys.com
 Tel.: +49 831 540310
 Web: www.codesys.com
- > Anschrift: CODESYS Development GmbH
 Tobias-Dannheimer-Str. 5
 87439 Kempten

Projekt CODESYS go!

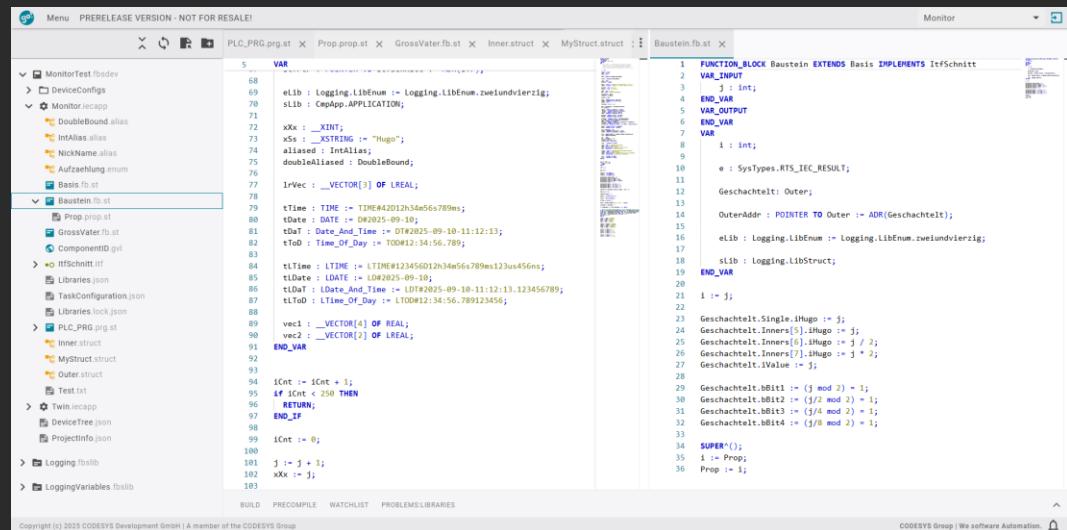


> Webbasierte Entwicklungsumgebung

> Plattformunabhängig

> Bereitstellungsformen:
Windows Installer
Debian Package
Docker Container

> Technologien:
Frontend: JavaScript/TypeScript, Angular, Tailwind
Backend: C#
Tests: Playwright, xUnit



The screenshot shows the CODESYS go! development interface. The left sidebar displays the project structure with files like MonitorTest.libdev, DeviceConfig, Monitor.tecapp, and Baustein.fb.st. The main area shows a code editor with C# code for a function block named 'Baustein'. The code includes declarations for variables like 'elLib', 'VAR_OUTPUT', and 'VAR', and logic involving 'LTIME', 'DATE', and 'TIME_OF_DAY'. The bottom of the interface shows standard development tools like 'BUILD', 'PRECOMPILE', 'WATCHLIST', 'PROBLEMS', and 'LIBRARIES'.

```
1 FUNCTION_BLOCK Baustein EXTENDS Basis IMPLEMENTS ItfSchnitt
2 VAR_OUTPUT
3  i : int;
4 END_VAR
5 VAR_OUTPUT
6 VAR
7 VAR
8  i : int;
9  e : SysTypes.RTS_IEC_RESULT;
10 Geschachtelt: Outer;
11 OuterAddr : POINTER TO Outer := ADR(Geschachtelt);
12 elLib : Logging.LibEnum :: Logging.LibEnum.zweiundvierzig;
13 sLib : Logging.LibStruct;
14 END_VAR
15
16
17
18
19
20
21 1 := j;
22
23 Geschachtelt.Single.iHugo := j;
24 Geschachtelt.Inner1.iHugo := j;
25 Geschachtelt.Inner2.iHugo := j / 2;
26 Geschachtelt.Inner3.iHugo := j * 2;
27 Geschachtelt.Value := j;
28
29 Geschachtelt.bHil1 := (j mod 1);
30 Geschachtelt.bHil2 := (j/2 mod 2) = 1;
31 Geschachtelt.bHil3 := (j/4 mod 2) = 1;
32 Geschachtelt.bHil4 := (j/8 mod 2) = 1;
33
34 SUPER();
35
36 Prop := j;
37 Prop := j;
```