

TEAMVORSTELLUNG

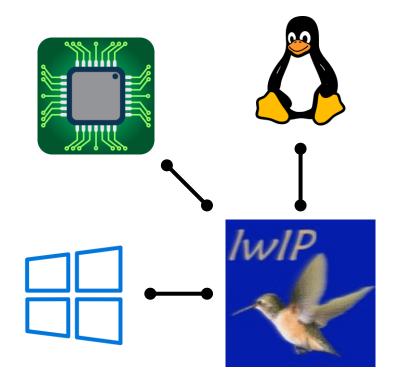
Name	Matrikelnummer	Projektrolle	eMail-Adresse
Lucas Kaczynski	9411266	Techn. Redakteur	inf20147@lehre.dhbw-stuttgart.de
Benjamin Esenwein	5655208	Produktmanager	inf20074@lehre.dhbw-stuttgart.de
Maximilian Meier	6020212	Leitender Entwickler inf20084@lehre.dhbw-stuttgart	
Yannis Plaschko	6645234	Testmanager	inf20093@lehre.dhbw-stuttgart.de
Laura Reeken	3811697	Projektleiterin	inf20051@lehre.dhbw-stuttgart.de
Isabel Schwalm	6038831	Systemarchitektin	inf20085@lehre.dhbw-stuttgart.de

INHALT

- Was ist lwIP?
- Master Use Case
- Systemarchitektur & Produktübersicht
- Funktionale und Nichtfunktionale Anforderungen
- Module
- Entwicklungsprozess
- Vorgehen beim Testen
- Testplan
- Rückblick Prototyp & Live Demo
- Fazit / Ausblick

Was ist lwIP?

- kleine unabhängige Implementierung des TCP/IP-Protokolls
- Zielsetzung
 - Reduzierung des Ressourcenverbrauchs
 - Gleichzeitige Bereitstellung eines vollwertigen TCP/IP-Protokolls
- → Geeignet für Embedded Systems
- → Lauffähig auf mehreren Betriebssystemen



MASTER USECASE



- Ziel: Websocket Support für lwIP implementieren
 - lwIP-Status Seite & REST-API implementieren



Aufgabe: Mängel des vorhandenen Patch #9525 beheben



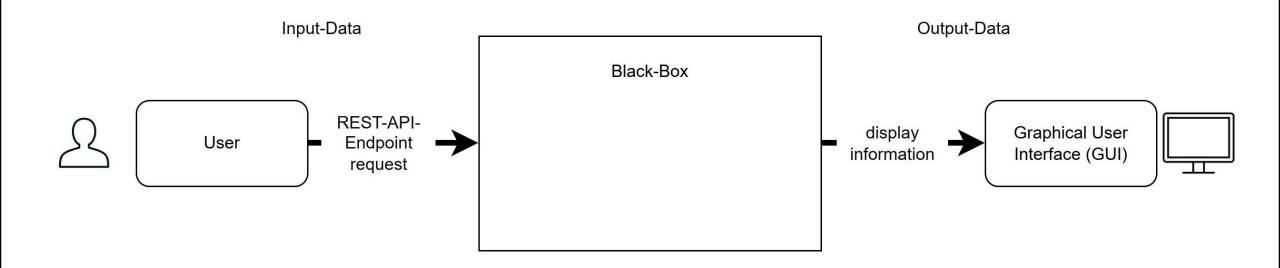
Use Case:

Beteiligte Nutzer: Entwickler von Embedded System

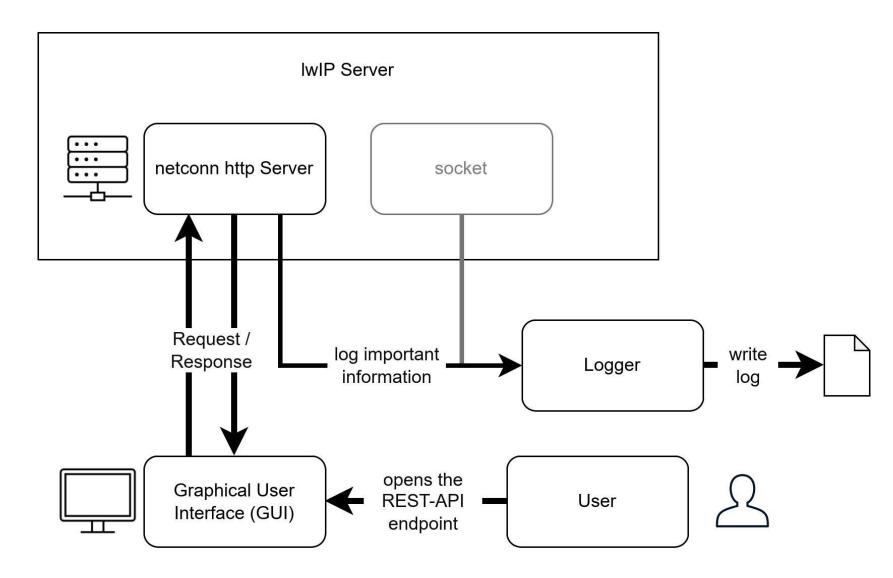
Zielsetzung: • Support für TCP/IP und Websockets

lwIP ist auf CPU lauffähig

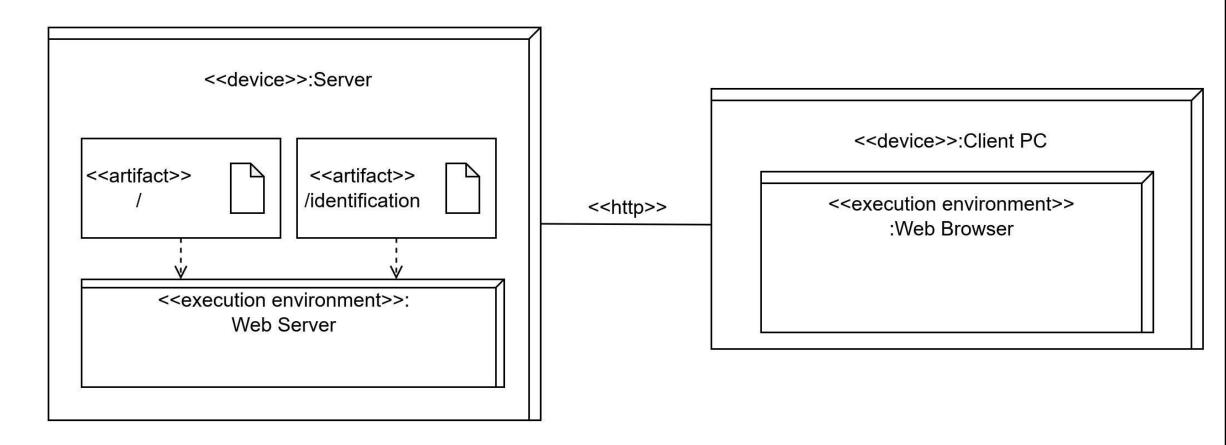
SYSTEMARCHITEKTUR



Produktübersicht



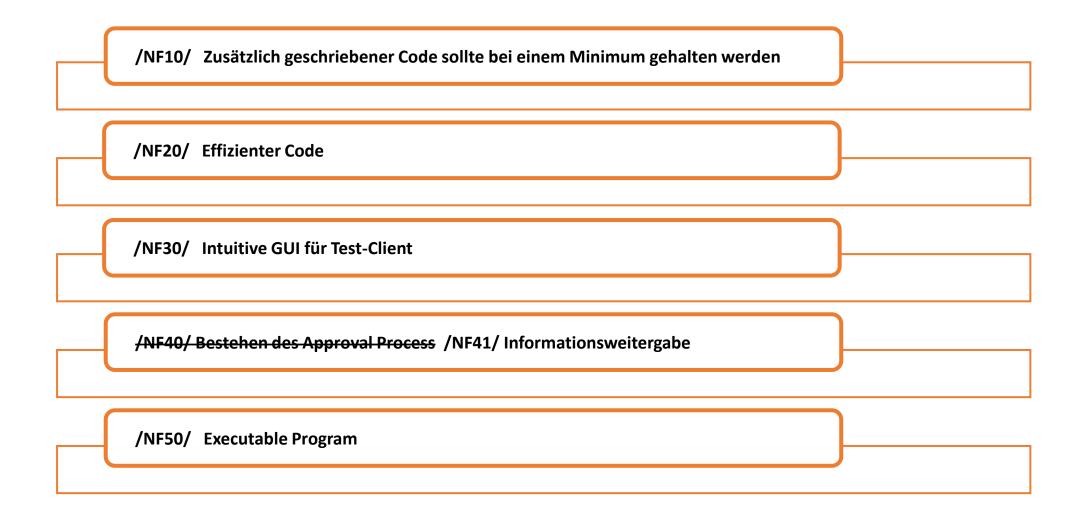
Produktübersicht



FUNKTIONALE ANFORDERUNGEN

/F10/ Extra Funktion für base64 encoding /F20/ Implementation der WebSocket API /F30/ Schreiben eines GUI-basierten Test-Client /F40/ http Webserver für Testzwecke

NICHTFUNKTIONALE ANFORDERUNGEN



Module

```
/MOD.001/ Graphical User Interface

/MOD.002/ HTTP-Server

/MOD.003/ WebSocket Server
```

Entwicklungsprozess

- Visual Studio 2022
- WIN SDK 10.0
- Analyse & Einarbeitung Patch#9525
- Debugging
- Alternative WebSocket Server

libwebsockets.org









Bildquellen:

https://logos.fandom.com/wiki/Microsoft_Visual_Studio#2021.E2.80.93present

https://cmake.org/

https://www.winpcap.org/docs/docs 412/html/

Npcap: Windows Packet Capture Library & Driver

libwebsockets.org lightweight and flexible C networking library

VORGEHEN BEIM TESTEN

- LwIP ist sehr umfangreich und komplex
 - Kein White-Box-Testing möglich
- Black-Box-Testing
- GUI als POC daher kaum Testdaten
- POO aus Sicht des Users
- Testplanung
 - Verifikation der Lauffähigkeit
 - Erzeugen und Zerstören von Verbindungen

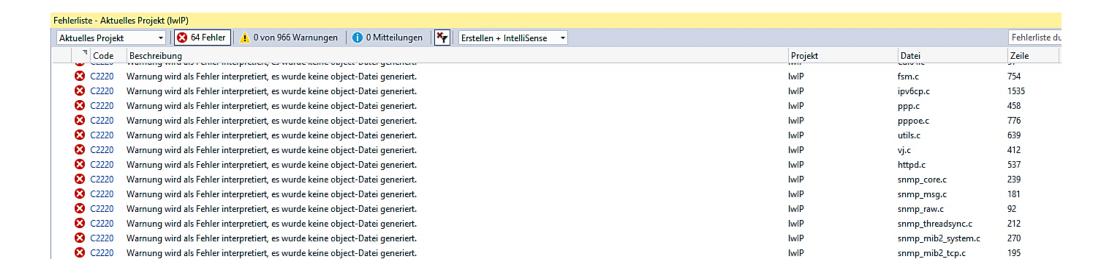
Testplan

Test- ID	Req ID	Functionality
1	LF60: Call API root	This test verifies that calling the Root Endpoint returns a sample website
2	LF70: Call API /something	This test case verifies that calling an undefined Endpoint results in an error Message

11.2.1 (TC 002.001) (C-II ADI D4))							
11.2.1 <tc-002-001> (Call API Root)></tc-002-001>							
Testcase ID:		TC-002-001					
Testcase Name:		Call API Root					
ReqID:		LF60					
Description	This test case verifies th sample website.			nat calling the Root Endpoint returns a			
				,			
Test Steps:							
Step	Action	Action		Expected result			
1	Run th	Run the LwIP HttpServer		Application starts without problems.			
2	the IP	Open the Browser and type the IP of the Server in the Search bar and hit enter		A Site should become visible with the Headline "LwIP Test Application", a Text block and a link to "/identification";			
Testdata:	TD-0	002-001					
Dataset	Inpu	ıt File	Output File				
1	Х		Х				

RÜCKBLICK AUF PROTOTYP

- Rückmeldung der lwIP-Community stand aus
- Code-Abhängigkeits-Fehler



LIVE-DEMO

Anwendung öffnen

JSON Validator

• FAZIT

