**Customer Requirements Specification**

**(Lastenheft)**

(TINF20C, SWE I Praxisprojekt 2021/2021)

Project: Websockets im LwIP HTTP Server

Customer: Rentschler & Holder

Rotebühlplatz 41

70178 Stuttgart

Supplier: Team 4 (Laura Reeken, Benjamin Esenwein, Yannis Plaschko, Maximilian Meier,

Lucas Kaczynski, Isabel Schwalm)

Rotebühlplatz 41

70178 Stuttgart

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comment** |
| 0.1 | 20.09.2021 | Benjamin Esenwein | Created |
| 0.2 | 27.09.2021 | Team 4 | Formulated the general components of the project using the CRS template |
| 0.3 | 05.10.2021 | Lucas Kaczynski | Cleaned up the goal description. Added a Product Environment description. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Allgemeine Hinweise:

Alles, was in dieser blauen Schriftart gesetzt ist, dient nur zur Erläuterung und sollte im fertigen Lastenheft nicht mehr auftauchen!

Der Umfang dieses Dokuments darf sechs Seiten nicht überschreiten.

Ein Lastenheft enthält eine grobe Beschreibung aller fachlichen Anforderungen, die das zu entwickelnde Produkt erfüllen muss. Die Inhalte des Lastenheftes (CRS) dienen als Grundlage für das Pflichtenheft und können -wenn sinnvoll- im Pflichtenheft (SRS) wieder verwendet werden.

#### Offene Punkte

In diesem Abschnitt sollen alle Probleme und offenen Fragen gesammelt werden. Bei einem fertigen Lastenheft sollte er leer sein, aber bei Zwischenversionen kommt diesem Abschnitt besondere Bedeutung zu!

CONTENTS

*1.* Goal 3

2. Product Environment 4

3. Product Usage 5

*3.1.* Business Processes 5

3.1.1. <BP.001>: <Name> 5

3.2. Use Cases 6

3.2.1. <UC.001> Use Case Name 6

*3.3.* Features 7

3.3.1. /LF10/ ….. 7

3.3.2. /LF20/ ….. 7

*4.* Product Data 8

4.1. /LD10/ ….. 8

4.2. /LD20/ ….. 8

*5.* Other Product Characteristics 9

5.1. /NF10/ ….. 9

5.2. /NF20/ ….. 9

5.3. System Environment 9

6. References 10

# Goal

LwIP fokussiert sich darauf möglichst wenig Ressourcen zu verbrauchen, gleichzeitig jedoch einen voll funktionsfähigen TCP/IP stack bereitzustellen. Wegen seiner hohen Effizienz, wird LwIP zum Beispiel von Intel, Xilinx und Analog Devices in embedded systems verwendet (Q: https://en.wikipedia.org/wiki/LwIP#OS\_implementations).

Das Ziel dieses Projekts ist die Behebung der Architekturmängel des Patches „#9525 (httpd: add Websocket support)“ [1] in Abstimmung mit der Projekt-Community. Diese experimentelle Basis soll verbessert und im Open-Source-Projekt durch den Approval-Prozess gebracht werden.

Des Weiteren soll ein Demo-Server in einer virtuellen Umgebung unter Windows entworfen und implementiert werden.

Zu Demonstrations- und Testzwecken der Features soll ein GUI-basierter Testclient entworfen und implementiert werden.

~~lwIP TCP/IP kann für Systeme verwendet werden, welche eine vollständige TCP Unterstützung benötigen, allerdings eine Beschränkung im Bereich des RAM und ROM haben.~~

TBD:

**Zielgruppe?**

(Google, wer dies benutzt)

**Wo sollen wir starten?**

* TCP/IP findet man überall (z.B. kann man den localhost mit ping 127.0.0.1 anpingen)
* IP-Stack unterstützt Protokolle (z.B. ping Protokoll)
* Aufgaben:
  1. Google „lwIP Windows“ liefert Hinweise wie man lwIP auf Windows installieren kann. Recherchiere, ob es eine Referenzumgebung gibt, in der man lwIP unter Windows testen kann.
     + Wenn man nichts finden kann, dann die Leute anschreiben. Vielleicht können die uns die Windowsumgebung rüberschicken von früher.
     + Hardcodierte IP-Adresse am Anfang vergeben, um es einfacher zu bekommen.
  2. Den Maintainer des Projekts (Simon Goldschmidt) anfragen, dass wir den Patch von Sakari in den Mainbranch einarbeiten wollen. Vielleicht kann uns der Projectmaintainer dabei helfen.
     + Wir wollen http Websockets implementieren. Das würde lwIP um ein spannendes neues Feature ergänzen und es zu einem robusteren Projekt zu machen.
     + Dafür müssen wir das http Protocol verstehen, um das zu implementieren.
  3. Wenn lwIP unter Windows läuft (man es anpingen kann) kann man den Patch testen und schauen, was daran falsch ist.
  4. Den Patch nach Simon Goldschmidts Anweisungen umschreiben, dass er in den Main Branch übernommen werden kann.

**Warum/was ist das zugrunde liegende Problem bei dem Patch?**

* Patch wurde nicht sauber ausprogrammiert -> soll in Mainstream des Projekts übernommen werden

**Welche Version des Projekts sollen wir benutzen, wo der Patch rein soll? (STABLE-2\_0\_3\_RELEASE)?**

* Immer die neueste Version des lwIP Stacks verwenden

**Was muss am Patch gemacht werden, damit er durch den Approval Process kommt?**

* Patch soll von Goldschmidt übernommen werden können

**Wie einfach muss es sein den Demo-Server auf einem PC neu zu installieren? Muss überhaupt ein Demo-Server woanders aufgesetzt werden können? Zum Beispiel für Tests von Ihrer Seite aus?**

* Nach fertigem http Server im Internet suchen (haben schon x andere gemacht) oder alternativ selbst aufsetzen
* In C ein kleines Testprogramm programmieren schreiben, damit der lwIP-Stack als Windows Programm ausgeführt werden kann. Dieses Testprogramm muss auf anderen PCs lauffähig sein.

**Wie weit sind wir bei der Programmiersprache für den Client eingeschränkt? Muss es C sein?**

* C oder C++
* C ist am einfachsten

**Namenskonvention unklar beim Repository: Auf S. 3 steht „TINF20C/TeamNr/ProjektName“ und auf S. 6 „Team\_x\_Projektname\...\“**

* Formatierung von S. 6 nehmen

**Ist der Business Process richtig angelegt?**

* Da wir Open Source sind wäre Business Process Geld Ersparnis für andere Menschen, die unsere lwIP Lösung verwenden wollen.
* Was bisher drin steht ist nicht Business Process

The goal of this project is to develop a software which supports..

…..

Dieser Abschnitt hat die Aufgabe als Einleitung zu dienen. Beschrieben wird die Hauptaufgabe des Systems. Meist kann man von der Aufgabenstellung bzw. Auftragsanfrage abschreiben. Wichtig ist es, den Grund für die Systementwicklung (Probleme oder Geschäftsideen) und damit ihre Ziele herauszuarbeiten.

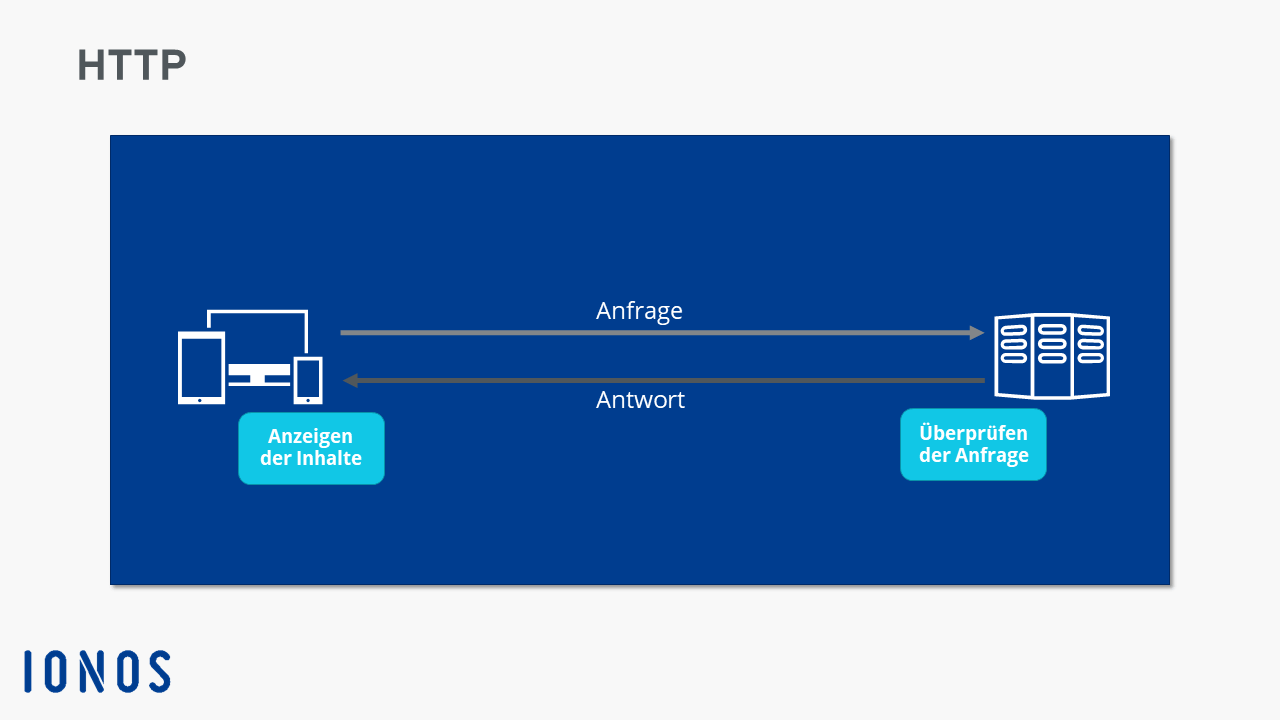
Benennen Sie auch die Zielgruppe, die später mit dem System arbeiten soll.

# Product Environment

TCP/IP ist ein Set an Protokollen, durch welches die Kommunikation zwischen unterschiedlichen internetfähigen Geräten möglich ist. TCP/IP legt fest, wie Informationen packetized, addressed, transmitted, routed and received werden. TCP/IP wird in vier Schichten unterteilt. Der link layer liegt am tiefsten. In ihm werden alle hosts hinterlegt, die ein Gerät innerhalb seines Netzwerks, ohne über einen Router zu gehen, ansteuern kann. Datenpakete können über den link layer innerhalb dieses lokalen oder virtuellen Netzwerks ausgetauscht werden. Eine Schicht darüber liegt der internet layer. Mit ihm ist es möglich, das Datenpakete aus einem Netzwerk in ein anderes Netzwerk übertragen werden können. Dieser Prozess wird routing genannt. Über dem internet layer befindet sich der transport layer. Hier finden die host-to-host Verbindungen statt. Diese end-to-end message transfer services sind unabhängig von ihrem zugrundeliegenden Netzwerk. Eine Schicht darüber befindet sich der application layer. Hier findet process-to-process data exchange for applications statt. Applications können Informationen, über die bereits von den tieferen Schichten etablierten Verbindungen austauschen. Über den application layer laufen z.B. das Hypertext Transfer Protocol (HTTP) oder das File Transfer Protocol (FTP).

(Q: <https://en.wikipedia.org/wiki/Internet_protocol_suite#Link_layer>)

TCP/IP stellt eine Verbindung zwischen zwei Kommunikationsendpunkten her. Diese Kommunikationsendpunkte werden auch Sockets genannt. Daten können von beiden Parteien gesendet und empfangen werden. Bei diesem Vorgehen kommt es jedoch zu Verzögerungen, weil der Client erst eine Anfrage an den Server senden muss, bevor ihm die Webseite zugesendet wird.



Aufbauend darauf erweitert Websocket diese Verbindung und macht sie bidirektional. Daten können zwischen den Sockets gleichzeitig gesendet und empfangen werden. Ein Client bekommt die Webseite sofort angezeigt, sobald er eine Verbindung zum Server öffnet.

Die Verbindung zwischen Server und Client bleibt offen. Der Server kann also Informationen an den Clienten senden, ohne dass er diese anfordert. Dieser Prozess kann in figure 1 eingesehen werden. (Q: https://www.ionos.de/digitalguide/websites/web-entwicklung/was-ist-websocket/)

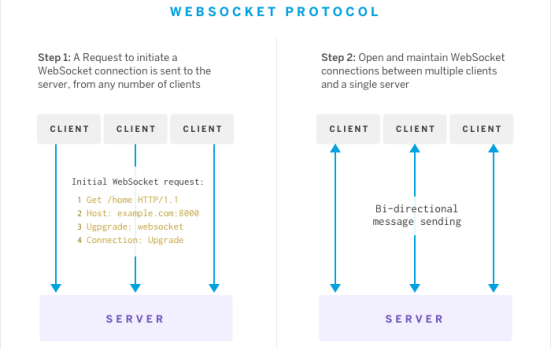


Figure . Websocket Protocol.

TCP/IP lief früher über http Version 1.0. Mittlerweile existiert jedoch http Version 2.0, welche schnellere Verbindungen ermöglicht. Ein Vergleich der beiden Versionen kann Tabelle 1 entnommen werden.

|  |  |  |
| --- | --- | --- |
| Feature | http 1.0 | http 2.0 |
| Datastream | Für unterschiedliche Seitenelemente (JS-, CSS-, Bildateien) werden mehrere TCP-Verbindungen geöffnet. | Es wird eine TCP-Verbindung geöffnet, über die mehrere Seitenelemente parallel übertragen werden können. |
| Komprimierung | Daten werden unkomprimiert übertragen. | Daten werden in Binärcode komprimiert und dann übertragen. |
| Priorisierung | Datenpakete werden nicht priorisiert. | Datenpakete werden priorisiert. |

Table 1. Comparisson between http version 1.0 and 2.0.

(Q:<https://www.seonative.de/http2-und-seo-welche-vorteile-hat-das-neue-protokoll/>, <http://www.http2demo.io/>)

The usage environment of the software is characterized by...

…..

Dieser Abschnitt hat die Aufgabe den Einsatzbereich des zu entwickelnden Systems klarzustellen. Dazu gehören Erläuterungen der notwendigen Fachbegriffe und deren Zusammenhänge ebenso wie die Darstellung der systemrelevanten Abläufe im Einsatzbereich.

Unter dem Produkteinsatz versteht man sowohl den direkten Problembereich, in dem das zu entwickelnde System eingesetzt werden soll, als auch die umgebenden Geschäftsprozesse.

Hier also den Problembereich des Projektes benennen und erläutern, ob es zu unterstützende Abläufe im Einsatzbereich (Geschäftsprozesse) gibt und wo sie zu finden sind.

Dieser Abschnitt muss so geschrieben sein, dass er, den Laien mit der Terminologie und den Zusammenhängen im Problembereich vertraut macht. Daher muss die Beschreibung möglichst allgemein sein. Außerdem sollte der Text gut strukturiert sein. Auch der Einsatz von erläuternden Graphiken ist manchmal sinnvoll.



Figure x: Product Environment

Wichtig ist es auch noch, gemachte Annahmen sauber von den oben beschriebenen Fakten getrennt aufzulisten. Dies erleichtert eine spätere Fehlersuche, wenn das System nicht die Erwartungen erfüllt.

# Product Usage

The following business processes, use cases and features shall be supported by the system.

Dieser Abschnitt hat die Aufgabe, die Anwendung des zu entwickelnden Systems sowohl überblicksartig als auch detaillierter aus Benutzersicht zu beschreiben. Abschließend sollen die vom Produkt erwarteten Features beschrieben werden.

Dem Nutzer wird die Möglichkeit geschaffen, innerhalb von lwIP Websockets zu verwenden. Dazu wird die Dokumentation von lwIP um eine Anleitung über die Verwendung von Websockets und um Codebeispiele ergänzt. Der Nutzer kann einen Demo-Server aufsetzten, auf dem ein rudimentärer HTTP-Server läuft, dessen Kommunikation mittels eines Websockets realisiert wird. Der HTTP-Server kann über eine einfache GUI angesprochen werden und ermöglicht es dem Nutzer die Features des HTTP-Servers zu prüfen.

*✔️*

## Business Processes

Bei diesem Projekt handelt es sich um open source Projekt. Dementsprechend liegt primär kein Gewinninteresse vor. Das Projekt wird mit dem Ziel entwickelt eine Zeit- und Kostenersparnis für Firmen und Entwickler darzustellen.

Falls notwendig, sind hier die identifizierten Geschäftsprozesse näher zu beschreiben. Jeder von ihnen erhält einen eigenen Unterabschnitt gemäß dem Template. In diesem Abschnitt wird der Ablauf der Geschäftsprozesse des vorigen Abschnittes genauer beschrieben. Diese Abläufe sind es, die das zu entwickelnde System ausschnittsweise unterstützen soll.

### <BP.001>: <Name>

|  |  |
| --- | --- |
| Triggering Event: | <Handlung oder Zeitpunkt, die Geschäftsprozess auslöst bzw. zu dem er beginnt> |
| Result: | <Was im Falle einer erfolgreichen Ausführung des Geschäftsprozesses erreicht werden soll> |
| Involved Roles: | <Rollenname derjenigen, die an der Durchführung des Geschäftsprozesses beteiligt sind. Das können auch existierende Systeme sein.> |

***Durch eigenes Aktivitätsdiagramm ersetzen.***



Figure 2.2: <BP.001> Activity Diagram

## Use Cases

Aufgabe dieses Abschnittes ist es, einen Überblick über die Produktfunktionen zu geben. Dazu wird ein Use Case Diagramm eingesetzt, das eine abstrakte Sicht auf die Produktfunktionen und die externen Beteiligten an diesen Funktionen gibt.

***Durch eigenes Use Case Diagramm ersetzen.***



Figure x: Use Case Overview Diagram

### <UC.001> Use Case Name

Dieser Abschnitt muss für jeden Use Case wiederholt werden. Hier soll der Use Case mit einer ID versehen und kurz textuell sowie ggf. mit einem Aktivitätendiagramm erläutert werden.

|  |  |
| --- | --- |
| **Related Business Process:** | Prozess-ID: <elementarer Geschäftsprozess > |
| **Use Cases Objective:** | Ausführliche Beschreibung des Zieles des Use Cases |
| **System Boundary:** | Systemgrenze, die betrachtet wird (aus Diagramm des vorigen Abschnittes) |
| **Precondition:** | Was muss garantiert werden, damit der Use Case durchgeführt werden kann? |
| **Postcondition on success:** | Was muss sichergestellt werden für eine erfolgreiche Ausführung des Use Case |
| **Beteiligte Nutzer:** | Rollenname: Beschreibung des Nutzers, der mit dem System interagiert. Nutzer können auch andere Systeme sein. |
| **Triggering Event:** | Handlung oder Zeitpunkt, die Use Case auslöst bzw. zu dem er beginnt |

## Features

In diesem Abschnitt sollen die bereits definierbaren funktionalen Features in „Balzert-Notation“ aufgelistet werden und nach den Regeln der Anforderungsschablone ausformuliert werden. à (<https://www.sophist.de/fileadmin/SOPHIST/Puplikationen/Broschueren/SOPHIST_Broschuere_MASTeR.pdf>)

### /LF10/ …..

The [Component] shall {[adverb]} check if [Condition] and, {in case of violation trigger the appropriate reaction as specified in the fault database} {perform the following reactions:[List of Reactions]}

### /LF20/ …..

The [Component] shall {[adverb]} check if [Condition] and, {in case of violation trigger the appropriate reaction as specified in the fault database} {perform the following reactions:[List of Reactions]}

# Product Data

In diesem Abschnitt werden die Hauptdaten und Datenschnittstellen beschrieben, mit denen das Softwareprodukt arbeiten soll und die bereits identifizierbar sind (siehe Abb. 1). Im Allgemeinen werden diese Hauptdaten eines Programms auch nonvolatil gespeichert.

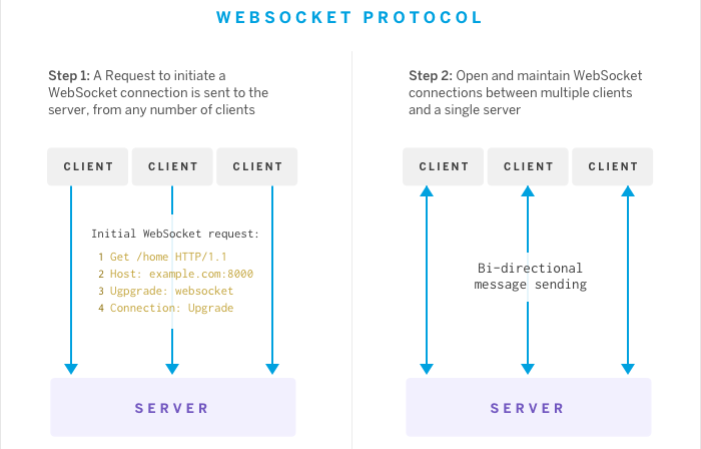


Figure 2: Websocket Protocoll (source: M. Rentschler)

## /LD10/ …..

…

## /LD20/ …..

….

# Other Product Characteristics

Der verringerte Ressourcenverbrauch bei gleichzeitiger Bereitstellung eines vollwertigen TCP/IP-Protokolls charakterisiert den lwIP-TCP/IP-Stack. Dadurch eignet sich lwIP für den Einsatz in Embedded-Systemen mit minimaler Verfügbarkeit von RAM und ROM. Die Attraktivität dieses Produktes liegt klar in der OpenSource-Verfügbarkeit. Damit wird der Verbreitungsgrad und die Weiterentwicklung des Produktes gefördert.

IwIP is a freely software written in the C programming language.

This section describes the already known non-functional requirements for the product.

…

Die Aufgabe dieses Abschnittes ist die Beschreibung der nicht-funktionalen Anforderungen. Dabei handelt es sich um Charakteristiken oder Qualitäten, die das Produkt attraktiv machen und es von vergleichbaren Produkten unterscheiden.

In diesem Abschnitt werden die wesentlichen Eigenschaften des zu entwickelnden Produktes beschrieben, die nicht direkt die zu leistende Funktionalität betreffen.

TBD:

In diesem Abschnitt sollen diese bereits definierbaren Anforderungen in „Balzert-Notation“ aufgelistet werden und nach den Regeln der Anforderungsschablone ([*https://www.sophist.de/fileadmin/SOPHIST/Puplikationen/Broschueren/SOPHIST\_Broschuere\_MASTeR.pdf*](https://www.sophist.de/fileadmin/SOPHIST/Puplikationen/Broschueren/SOPHIST_Broschuere_MASTeR.pdf)) ausformuliert werden.

## /NF10/ …..

The software/system shall support …

## /NF20/ …..

….

## System Environment

This section describes the system environment required to operate the product.

…

Hier sollten alle wesentlichen und notwendigen Parameter der Systemumgebung (Hardware, Software) beschrieben werden, soweit diese bereits festlegbar ist.

Siehe 4. Product Data

* Demo-Server: Virtuelle Windows Umgebung (Orientierung an ESP8266-Hardware und -Umgebung)
* Client-Maschine: Virtuelle Windows-Umgebung

# References

[1] https://savannah.nongnu.org/patch/?9525

[2] <https://savannah.nongnu.org/projects/lwip/>

[3] <https://www.nongnu.org/lwip/2_1_x/group__httpd.html>

[4] <https://lujji.github.io/blog/esp-httpd/>

[5] https://lwip.fandom.com/wiki/Compiling\_lwIP