

Testbericht Smart Home

Henrik Trübenbach, Dennis Wilpert

May 2020

Inhaltsverzeichnis

1	Einleitung	3
2	Umgebung	3
3	Funktionale Tests	4
4	Nichtfunktionale Tests	6

1 Einleitung

Die Suche nach einem geeignetem Testframework hat etwas länger gedauert, da wir spezielle Anforderungen hatten. Da wir mit Visual Studio in Verbindung mit Windows wsl programmiert haben, müssen auch die Tests in der Lage sein, für Linux-Plattformen kompiliert zu werden. Da VS jedoch von Haus aus kein solches Framework zu Verfügung stellt, mussten wir was eigenes finden.

Die anderen Voraussetzungen waren, dass es möglichst leicht zu benutzen ist und sich einfach in das Projekt eingliedern lässt.

2 Umgebung

Aufgrund dieser Überlegungen haben wir uns für das Testframework "doctest" [<https://github.com/onqtam/doctest>] entschieden. Es besteht aus einer einzigen header file, orientiert sich stark an Catch, und kann für Windows und Linux kompiliert werden. Da doctest einfach zu nutzen und zu implementieren ist, erfüllt es alle vorher bestimmten Anforderungen.

3 Funktionale Tests

Test 1:

Name:

Kommt die Nachricht an der Zentrale an

Beschreibung:

Mit diesem Test soll getestet werden, ob Nachrichten von dem Client an den Server geschickt werden müssen.

Durchführung:

Zu begin wird ein Objekt der Zentrale initialisiert und dann die Methode initialize() der Klasse UDP_servers aufgerufen um den Nachrichtenaustausch zu ermöglichen. Wichtig dabei ist, dass man vorher über Windows WSL Ubuntu die Zentrale selber startet. Danach wird ein Sensor (Client) erstellt und dann die Nachricht verschickt. Wenn die status Variabel der Zentrale auf 0 steht, so kam die Nachricht an.

Erwartung:

Wir haben erwartet, dass der Test positiv ausgeht.

Ergebnis:

Der Test geht positiv aus, die versendete Nachricht kommt an.

Test 2:

Name:

HTTP REQUEST and RESPONSE

Beschreibung:

Mit diesem Test soll getestet werden, der Client eine erfolgreiche http request an die Zentrale schicken kann und dann eine erfolgreiche respond bekommen kann mit den angeforderten Daten.

Durchführung:

Zu begin wird ein Objekt der Zentrale initialisiert und dann werden die erforderlichen Methoden für den Client gestartet. Wichtig dabei ist, dass man vorher über Windows WSL Ubuntu die Zentrale selber startet.

Erwartung:

Wir haben erwartet, dass der Test positiv ausgeht und eine erfolgreiche response vom Server zurückkommt.

Ergebnis:

Der Test geht positiv aus, der request kommt an.

4 Nichtfunktionale Tests

Test 1:

Name:

Paketverlust 1%

Beschreibung:

Innerhalb dieses Testes soll die Zuverlässigkeit der Datenübertragung getestet werden. Unser Ziel ist dabei weniger oder gleich 1% Paketverlust.

Durchführung:

Wir führen insgesamt 1000 mal das senden einer Nachricht durch, aufgeteilt auf alle 4 arten von Sensoren. Jedesmal wenn die Nachricht erfolgreich ankommt, wird dies mit einem Counter gespeichert. Am Ende des Durchlaufs wird die Anzahl an angekommenen Paketen überprüft.

Erwartung:

Wir erwarten ein sehr niedrigen Paketverlust, da einmal die größe der Pakete sehr klein ist, andererseits die Nachrichten über den localhost gesendet werden. Dadurch sollten der Paketverlust verschwindend gering sein.

Ergebnis:

Es konnte kein Paketverlust von größer als 1% festgestellt werden. Genau genommen kommen von den 1000 Nachrichten alle an.

Test 2:

Name:

Stabiles gleichzeitiges Senden

Beschreibung:

Innerhalb dieses Testes soll die Möglichkeit zur Zeitgleichen Datenübertragung an den Server getestet werden.

Durchführung:

Es werden die nötigen Methoden aufgerufen um jeweils Sensordaten als auch Client http-requests zu schicken. Der Test gilt als erfolgreich wenn alle Nachrichten wie erwartet ankommen.

Erwartung:

Wir erwarten dass alle Nachrichten von den Sensoren und dem Client ankommen, da wir dies mit eigenen Prozesses gelöst haben.

Ergebnis:

Alle Nachrichten kommen gleichzeitig an.