

Moduldokumentation

(MOD)

(TINF18C, SWE I Praxisprojekt 2019/2020)

Modul

NetworkScanner

Project: Profinet DCP Client als WEB-Applikation

Customer: Rentschler & Ewertz
Rotebühlplatz 41
70178 Stuttgart

Supplier: Team 2 (Sinan Yurrtadur, Jannik Schwarz, Noah Broß, Marvin Sonntag,
Nicholas Breuninger, Rene Scholz)
Rotebühlplatz 41
70178 Stuttgart

Version	Date	Author	Comment
0.1	28.04.2020	Noah Broß	Started filling out the basics
1.0	30.04.2020	Noah Broß	Added all the information

1. Content

1.	Content	2
2.	History	3
3.	Scope	3
4.	Definitions	Error! Bookmark not defined.
5.	Module Requirements	4
5.1.	User View	4
5.2.	Requirements	4
5.3.	Module Context	4
6.	Analysis	5
7.	Design	5
7.1.	Risks	5
8.	Implementation	6
9.	Module Test	7
9.1.	Component Testplan	7
9.2.	Component Testreport	7
10.	Summary	8
11.	Appendix	8
11.1.	References	8
11.2.	Interface Definitions	8
11.3.	Code	8
11.4.	Module Test Cases	8

2. History

Version	Datum	Autor(en)	Kommentare
0.1	28.04.2020	Noah Broß	Started filling out the basics
1.0	30.04.2020	Noah Broß	Added all the information

3. Scope

The Module Documentation (MOD) describes the architecture, the interfaces and the main features of the module. It also describes the module/component test including the results. It can also serve as a programming or integration manual for the module. If there are some risks related to the module itself, they shall be noted and commented within this document.

Die Moduldokumentation beschreibt die Architektur, die Schnittstellen und die Hauptmerkmale des Moduls. Außerdem werden die Modul bzw. Komponententests einschließlich der Ergebnisse beschrieben und dokumentiert. Die MOD dient bei Bedarf auch als Programmier- oder Integrationshandbuch für das Modul. Wenn bestimmte Risiken direkt mit der Verwendung des Moduls verknüpft sind, so sind sie in diesem Dokument zu benennen und zu kommentieren.

4. Module Requirements

4.1. User View

Wenn die Datenhaltung aktualisiert werden soll. Hier kann im Frontend ein Befehl gesendet werden, der dann nach neuen Profinet Geräten sucht

4.2. Requirements

Anforderungen welche durch dieses Modul (teilweise) abgedeckt werden:

- /LF10/ Searching Devices
- /LF30/ Save changes to device configuration

4.3. Module Context

Das Modul kommuniziert mit der Ausgabe. Die Ausgabe erhält vom Scanner die Dateien und liefert diese dann bei Bedarf aus.

5. Analysis

In order to configure Profinet-Devices you first have to find each device in the network. The Network Scanner is responsible for finding the Profinet-Devices in the connected network. This module is dependent on nmap (<https://nmap.org/>). It will be used by other modules to connect to the local network and give a response with the detected devices.

The nmap script must be in the same directory as this module for it to work.

6. Design

Solution:

The issue is finding devices in a network. By scanning the network for devices, it is possible to show each device.

Module-architecture:

When the module receives a request, it will scan the network with the help of nmap. The response is a list of devices.

APIs:

This module will receive REST-requests in order to communicate with the other modules. Currently only a GET-request will be handled.

Additionally, the command line can be used to communicate with the module.



6.1. Risks

If this module fails it won't be possible to find any new devices. Though you cannot find new devices, already found devices will still be able to be configured.

A risk lies within nmap and the required script for it to work. As the functionality of nmap isn't developed or maintained by us there is a risk to using it as 3rd party software.

This module starts a new process with a new Process-ID. When this module gets started it starts the new process and kills it after it has retrieved the network data.

Measures to reduce the risk have been taken. Nmap is a widespread tool for network analysis. Thus, it is a tool built upon years of expertise. So, the choice of the software has a solid foundation.

7. Implementation

The module was implemented with Node.js and the framework Node-red. Node-red allows a graphical development of programs. This was done to quickly create simple code and to save time. The more complex parts of the application were developed in JavaScript.

Code documentation can be found in the source-files.

8. Module Test

< Wie wird die Komponente getestet? White-Box und Black-Box-View! Dokumentation von Vorgehen und Ergebnissen. Bei Bedarf entsprechend erweitern.>

8.1. Component Testplan

Test-ID	Feature ID	Test Specification (Description or TCS)

8.2. Component Testreport

Test-ID	Pass/Fail	If failed: Test Observation	Date	Tester

9. Summary

Strengths:

This module works with nmap. Thus an operating system who can run nmap must be chosen.

Because of the graphical development method, the program is understood easily. It uses APIs which have been used for decades. So, it has a well understood foundation.

Weaknesses:

It might have to be configured.

The script for nmap must be in the same directory as nmap or else it won't work.

To be made improvements:

Another script could be used to retrieve more information from the profinet-devices.

Preparations for improvements:

To change the script, you can simply use the Exec-Node to load another script.

10. Appendix

10.1. References

10.2. Interface Definitions

10.3. Code

< Code nur, wenn es dem Verständnis dienlich ist >

10.4. Module Test Cases

Test-ID	Name	Description	Test Steps		
			Step	Action	Expected Result