

Pflichtenheft

Testumgebung und Performancevergleich von Zigbee, Thread und Bluetooth Mesh Netzwerken

BACHELOR THESIS - ANKLIN, BOBST, HORATH
14. Mai 2020

Fachcoach:

Matthias Meier
Manuel Di Cerbo

Team:

Raffael Anklin
Robin Bobst
Cyrill Horath

Studiengang:

Elektro- und Informationstechnik

Semester:

Frühlingssemester 2020

Inhaltsverzeichnis

1 Übersicht	1
1.1 Ausgangslage	1
1.2 Ziel der Arbeit	1
2 Lösungskonzept	2
2.1 Punkt zu Punkt Testinfrastruktur	3
2.2 Test Mesh Netzwerke	3
2.2.1 Bluetooth Mesh	4
2.2.2 Thread	4
2.2.3 Zigbee	4
2.3 Steuer und Auswertesoftware	5
3 Projektziele und Lieferobjekte	6
3.1 Punkt zu Punkt Testinfrastruktur	6
3.2 Test Mesh Netzwerke	7
3.3 Steuer- und Auswertesoftware	8
3.4 Zusatzziele	8
3.5 Lieferobjekte	8
4 Projektmanagement	9
4.1 Projektaufteilung	9
4.2 Projektplan	9
4.3 Risikoanalyse	9
4.4 Projektvereinbarung	10
A Aufgabenstellung	11
B Test Kriterien Point to Point auf MAC Ebene	16
C Test Kriterien Mesh Netzwerke	17
D Gesamt Terminplanung	18
E Bluetooth Mesh Terminplanung	19
F Thread Terminplanung	20
G Zigbee Terminplanung	21
H Risikoanalyse	22

1 Übersicht

Das vorliegende Dokument stellt das Pflichtenheft der Bachelorthesen von Raffael Anklin, Robin Bobst und Cyrill Horath an der Fachhochschule Nordwestschweiz Brugg-Windisch im Studiengang Elektro- und Informationstechnik dar. Im kommenden ersten Kapitel soll eine Übersicht über die Ausgangslage sowie das Ziel dieser Arbeit gegeben und somit die Rahmenbedingungen abgesteckt werden. Weiter werden die Lösungskonzepte 2 sowie die Projektziele und Lieferobjekte 3 definiert. Abschliessend soll auch noch das Projektmanagement 4 thematisiert werden.

1.1 Ausgangslage

Unter den standardisierten Low Power Mesh Netzwerk Protokollen im freien GHz ISM-Band konkurrenzieren sich derzeit vorrangig Bluetooth Mesh, Zigbee sowie Thread. Bezüglich MAC und Physical Layer basieren Zigbee und Thread auf IEEE 802.15.4 wogegen Bluetooth Mesh auf Bluetooth Low Energy (BLE) basiert. Jedes dieser Netzwerkprotokolle hat gewisse Vorzüge: Bluetooth Mesh, dass BLE mittlerweile von jedem Smartphone und Notebook unterstützt wird, Thread aufgrund seiner IPv6 Basis und damit einfachem Übergang ins Internet sowie Zigbee aufgrund seiner etablierten Verbreitung im Smart-Lampenbereich durch Philips, IKEA und Osram. Hauptproblem aller drei Mesh Netzwerkprotokolle ist nebst physikalisch und distanzbedingter Absorption und Reflexion die Störbeeinflussung durch WLAN (WiFi) und andere Netzwerke im GHz Frequenzbereich.

Im Rahmen des P5 mit dem Namen *Bluetooth-Mesh Plattform für IoT Anwendungen*, wurde das Bluetooth-Mesh Protokoll bereits vertieft betrachtet und dessen Vor- und Nachteile aufgezeigt. Basierend auf diesen Erkenntnissen und Erfahrungen und der oben beschriebenen Thematik soll das Bluetooth-Mesh Protokoll mit den Alternativen Thread sowie Zigbee verglichen werden.

1.2 Ziel der Arbeit

In der vorliegenden Arbeit soll zuerst ein praxistaugliches, einheitliches Testframework für alle drei Mesh Netzwerke erstellt werden, wonach die Tauglichkeit aller drei Mesh Netzwerke unter realitätsnahen Bedingungen ermittelt und verglichen werden soll. Zwecks besserer Vergleichbarkeit sollen alle drei Testnetze das gleiche Radio-Interface als Grundlage verwenden. Aufgrund der guten Unterstützung aller drei Mesh Protokolle als auch dem im vergangenen P5 gesammelten Wissens, sollen hierfür die nRF52840 SoCs der Firma Nordic eingesetzt werden. Die zu erstellende Testinfrastruktur soll aus den drei folgenden Teilen bestehen:

- Punkt-Punkt Testinfrastrukturen auf MAC-Ebene
- Test Mesh Netzwerke für BT Mesh, Zigbee und Thread
- Steuer- und Auswertesoftware

Die genauen Anforderungen an die Testumgebung sind einerseits in der Aufgabenstellung im Anhang A aufgeführt und andererseits werden sie anhand der Projektziele 3 definiert.

2 Lösungskonzept

Zur Messung und Auswertung der Mesh-Netzwerke dient ein Testframework wie es in Abbildung 2.1 schematisch dargestellt ist.

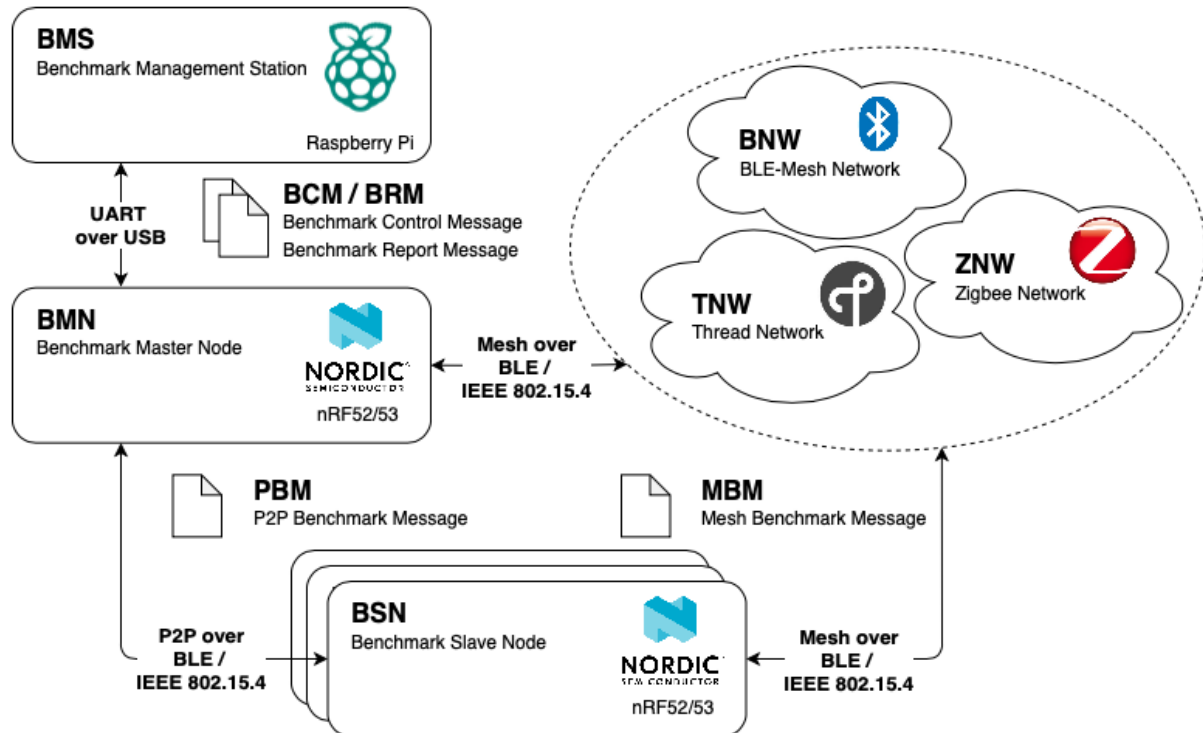


Abbildung 2.1: Konzeptschema Testframework

Das Testframework besteht aus folgenden physikalisch getrennten Teilsystemen:

- **BMS** Benchmark Management Station
Dient zur Verwaltung und Konfiguration des Testframeworks. Beinhaltet einen Webserver um dem Endanwender die Bedienung zu ermöglichen. Realisiert wird die BMS durch einen *Raspberry Pi 4 Model B*. Als Webserver wird das Python-Framework *Django* eingesetzt.
- **BMN** Benchmark Master Node
Dient als Zugangspunkt der BMS für die im Testframework gefahrenen Tests und lässt sich über eine serielle Schnittstelle ansprechen. In der Aufgabenstellung A wird der BMN als Master bezeichnet. Realisiert wird der BMN über einen nRF52840 von *Nordic*.
- **BSN** Benchmark Slave Node
Dient als Zugriffspunkt der im Testframework gefahrenen Tests und kann frei in der Testumgebung platziert werden, daher muss die Energieversorgung über einen Akku oder Batterie erfolgen. In der Aufgabenstellung A wird von einem Slave gesprochen. Realisiert wird der BSN über einen nRF52840 oder nRF5340 von *Nordic*. In einem Test Netzwerk werden unterschiedliche Arten von BSN eingesetzt wie zum Beispiel Mesh Router oder sogenannte End Devices. Letztere werden häufig auch als Low Power Nodes (LPN) bezeichnet.

Die logischen Komponenten des Testframeworks lassen sich wie folgt aufteilen:

- **BCM** Benchmark Control Message
Beschreibt Nachrichten welche zur Steuerung eines Benchmarks dienen. Dies sind zum Bei-

spiel Konfigurations-, Start- oder Stop-Befehle. Werden von der BMS initiiert und gelangen über eine USB-UART Verbindung zum BMN.

- **BRM** Benchmark Report Message
Beschreibt Nachrichten welche den Status oder die Ergebnisse eines Benchmarks zurück-melden. Werden vom BMN initiiert und gelangen über eine USB-UART Verbindung zur BMS.
- **PBM** P2P Benchmark Message
Nachrichten welche während der Durchführung eines Punkt zu Punkt Benchmarks versendet werden. Dies sind zum Beispiel Ping-Anfragen zur Latenzzeitmessung. Sie ermöglichen den Datenaustausch zwischen zwei Teilnehmern auf MAC-Ebene.
- **MBM** Mesh Benchmark Message
Nachrichten welche während der Durchführung eines Mesh Benchmarks versendet werden. Dies sind zum Beispiel Ping-Anfragen zur Latenzzeitmessung. Sie ermöglichen den Datenaustausch über ein Mesh-Netzwerk auf Applikations-Ebene.

2.1 Punkt zu Punkt Testinfrastruktur

Die Punkt zu Punkt Testinfrastruktur (P2P) ermöglicht es ein Benchmark auf physikalischer Ebene durchzuführen. Dies soll unabhängig vom Mesh-Protokoll und basierend auf den beiden MAC-Ebenen *BLE* und *IEEE802.15.4* möglich sein. Somit ist ein Vergleich der beiden MAC-Ebenen machbar. Weiter soll diese Infrastruktur einem Endanwender die Möglichkeit bieten die Sende- und Empfangsbedingungen an gegebenen Örtlichkeiten auf physikalischer Ebene auszumessen um geeignete Standorte für die Mesh Router zu bestimmen. Die Datenerfassung und grafische Aufbereitung erfolgt dabei auf der BMS. Eine Erfassung der Momentanwerte soll ebenso möglich sein wie die Aufzeichnung der Bedingungen in Form eines Langzeittests über mindestens 24 Stunden. Schliesslich soll also ein Messinstrument zur Bestimmung der Signalqualitäten im Feld entstehen. Im Anhang B werden die Testkriterien für die P2P Infrastruktur aufgeführt und beschrieben. Diese Tests werden mit Hilfe bereits bestehenden Beispiel Firmware (Radio-Example) aus der nRF Connect SDK auf dem nRF52840 durchgeführt. Optional wird dies auch noch auf den nRF5340 portiert welcher leistungsfähiger ist. Weiter soll diese P2P Testinfrastruktur dazu genutzt werden können, gezielte Störungen auf die Test Mesh Netzwerke zu richten die nachfolgend unter 2.2 beschrieben werden.

2.2 Test Mesh Netzwerke

Der Mesh-Benchmark soll die verschiedenen Mesh-Netzwerke möglichst identisch ausmessen um damit deren Protokoll Stacks untereinander zu vergleichen. Dazu dient bei allen Mesh-Netzwerken die Applikations-Schicht. Ein Mesh Netzwerk wird zwischen dem BMN und den BSN aufgebaut. Dazu werden die einzelnen Nodes über die BMS mit der entsprechenden Firmware geladen und anschliessend im Raum verteilt. Das Laden ist über eine Kabelverbindung (UART) vorgesehen. Allenfalls könnte dies zu einem späteren Zeitpunkt drahtlos mithilfe eines Bootloaders möglich gemacht werden. Dabei handelt es sich jedoch um ein Zusatzziel (siehe 3.4). Die Test Mesh Netzwerke sollen anders als die P2P Testinfrastruktur jedoch nur zu Testzwecken innerhalb dieser Arbeit eingerichtet werden und nicht als Messinstrument für Feldmessungen dienen. Im Anhang C sind die Testkriterien für den Mesh-Benchmark aufgeführt. Anhand dieser sollen die Messungen durchgeführt und analysiert werden. Zusätzlich sollen die Messungen auch unter verschiedenen Bedingungen durchgeführt werden. Beispielsweise ist zu Erwarten, dass die Messresultate aufgrund von sich verändernder Störbelastung durch andere Geräte in unmittelbarer Nähe unterschiedlich ausfallen, abhängig von Örtlichkeit und Anordnung der Nodes. Alle drei Mesh Netzwerke verfügen über unterschiedliche Nodetypen mit entsprechend differenzierten

Funktionen wie beispielweise Router oder End Devices. Die Mesh Netzwerke sollen möglichst realitätsnahe aufgebaut werden und somit mindestens diese beiden Typen beinhalten. Realitätsnah bedeutet in diesem Kontext dass eine mögliche Anwendung, als Beispiel in der Heimautomation, nachgebaut wird in welcher die Nodetypen und auch die Kommunikationswege klar definiert sind.

2.2.1 Bluetooth Mesh

Die BLE-Mesh Firmware der Nodes werden aus Beispielen der nRF Connect SDK und Zephyr abgeleitet. Das Mesh-Demo Beispiel erlaubt es die essentiellen Netzwerk Parameter fix vorzugeben. Dadurch müssen die Nodes nicht mehr Provisioniert werden und sind sofort einsatzbereit.

2.2.2 Thread

Die OpenThread Firmware wird mit Hilfe der API und den Tutorials von der offiziellen OpenThread Webseite erstellt. Die offizielle Seite von Google beschreibt das Netzwerk und alle Informationen die benötigt werden, um eine Firmware zu schreiben.

Die Abbildung 2.2 zeigt das Framework des Openthread Netzwerkes auf. Die Kommunikation von der BMS zum BMN findet Seriell mit UART over USB statt. Der Serielle Kanal wird mit Hilfe eines Python-Skripts ausgeführt.

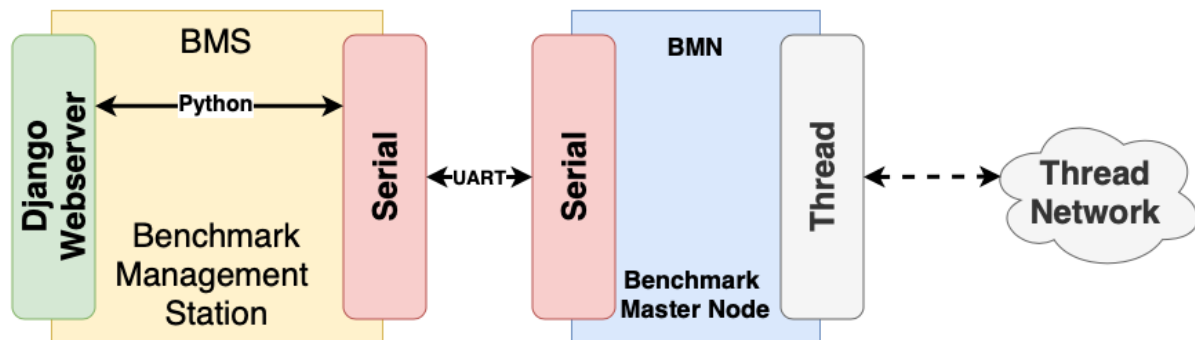


Abbildung 2.2: Konzept OpenThread Framework

2.2.3 Zigbee

Das Zigbee Test Mesh Netzwerk wird mithilfe der *nRF SDK for Thread and Zigbee* eingerichtet und für die Messungen vorbereitet. Der BMN wird dabei als Zigbee Coordinator und gleichzeitig als Zigbee Router eingesetzt. Die BSN können wieder als Router oder aber als Zigbee End Device betrieben werden.

2.3 Steuer und Auswertesoftware

Die Steuerung des Testframeworks erfolgt über eine Weboberfläche. Diese wird von der BMS mittels WLAN auf den Benutzergeräten zur Anzeige gebracht. Als Webserver dient das Python-Framework *Django*. Zur Steuerung der Benchmarks dienen Schrittketten, welche mit der Firmware auf dem BMN kommunizieren. Als letzter Schritt eines Benchmarks werden die Ergebnisse geloggt, nachbearbeitet und wiederum zur Anzeige gebracht.

Die Abbildung 2.3 stellt ein erstes Konzept dar, wie der Webserver für das P2P Test Framework aussehen kann. Die vom BMN ersichtlichen BSN werden aufgelistet und es können verschiedene Aktionen durchgeführt werden. Mit Reitern soll auf verschiedene Seiten gewechselt werden auf welchen wiederum Resultate oder Logs der Kommunikation ersichtlich sein sollen.

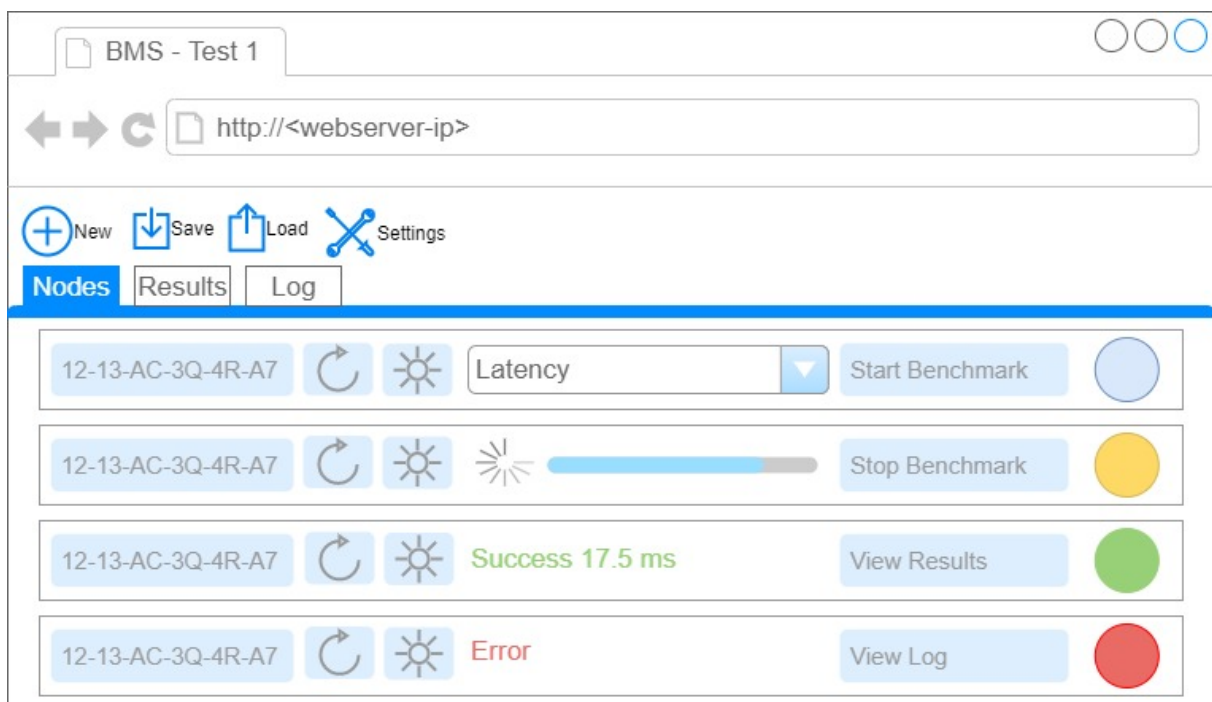


Abbildung 2.3: Entwurf Webserver

3 Projektziele und Lieferobjekte

Nachfolgend sind alle Projektziele aufgelistet. Die Ziele wurden in vier Teile unterteilt. Die ersten drei Teile entsprechen den Hauptzielen die in Kapitel 1.2 bereits erwähnt wurden. Die Tabelle 3.4 zeigt zusätzliche Ziele die als Wunschziele betrachtet werden und somit nicht im Fokus der Arbeit stehen. Wie bereits unter Kapitel 2.1 sowie 2.2 erwähnt, sind ergänzende Beschreibungen von Testkriterien in den Anhängen B und C zu finden.

3.1 Punkt zu Punkt Testinfrastruktur

Projektziele		
Nr.	Ziel	Beschrieb
P1	Kommunikation mit BMS	Der BMN kommuniziert via USB oder UART mit dem BMS.
P2	Senden von MAC-Frames	Das BMN sendet gemäss den Vorgaben der Testkriterien und gesteuert durch das BMS MAC-Frames an einen oder mehrere BSN.
P3	Rückbestätigung der MAC-Frames	Der oder die batteriebetriebenen BSN, bestätigen die MAC-Frames mit entsprechender Payload zurück.
P4	Konfiguration der Anzahl BSN	Die Anzahl der BSN ist über eine Steuer- und Auswertesoftware konfigurierbar.
P5	Adressierung der BSN	Die Adressierung der BSN ist über eine Steuer- und Auswertesoftware konfigurierbar.
P6	Konfiguration der Kanäle	Die BLE resp. 802.15.4 Kanäle können über eine Steuer- und Auswertesoftware ausgewählt werden.
P7	Einstellbare Framelänge	Die Framelänge der MAC-Frames ist über eine Steuer- und Auswertesoftware konfigurierbar.
P8	Einstellbare Frame- und Kanalwechselrate	Die Frame- und Kanalwechselrate ist über eine Steuer- und Auswertesoftware konfigurierbar.
P9	Einstellbare Sendeleistung	Die Sendeleistung der BSN ist über eine Steuer- und Auswertesoftware konfigurierbar.
P10	Anpassung der Modulationsart	In BLE soll die Modulationsart über eine Steuer- und Auswertesoftware konfigurierbar sein, um die Datenrate von 125kbps auf 2Mbps und die Long Range Funktion einzustellen.
P11	Ein- und Ausschalten der Collision Avoidance (CSMA/CA)	Beim 802.15.4 Protokoll soll die Collision Avoidance über eine Steuer- und Auswertesoftware ein- und ausgeschaltete werden können.
P12	Erfassen der Verbindungsqualität	Sowohl master- wie auch slaveseitige Erfassung der Verbindungsqualität (RSSI, Package Loss, Collisions, Noise Level, ...). Die BSN senden hierzu die erfassten Werte im Rückantwortframe dem BMN zurück. (Siehe auch Anhang B)
P13	Tool für Feldtests	Es soll ein Tool entstehen, welches dem Anwender die Möglichkeit gibt Messungen durchzuführen und somit sein Mesh Netzwerk zu planen.

Tabelle 3.1: Projektziele der Punkt zu Punkt Testinfrastruktur

3.2 Test Mesh Netzwerke

Projektziele		
Nr.	Ziel	Beschrieb
P1	Kommunikation mit BMS	Das BMN kommuniziert via USB oder UART mit dem BMS.
P2	Konfiguration BSN	Die BSN lassen sich frei zu einem Routing-Knoten, End-Knoten oder einem Low-Power Knoten konfigurieren.
P3	Mesh-Netzwerk	Alle drei Technologien Bluetooth, Thread und Zigbee müssen als Mesh-Netzwerk mit mindestens 10 BSN aufgebaut werden.
P4	Simulation Sensorwerte	Die BSN sollen in einem vom BMS vorgegebenen parametrisierbaren Intervall Sensorwerte simulieren
P5	Sensordaten	Als Sensordaten sollen die Netzzustandsdaten übermittelt werden: Paketnummer, Anzahl Retries, Paketverluste, RSSI, Strombedarf und aktive CPU- und Radio-Zeiten.
P6	Datenauswertung	Die Auswertung der gemessenen Daten soll entweder direkt auf dem BMS oder alternativ auf einem Client Rechner erfolgen. Eine Gegenüberstellung der Daten der drei Mesh Protokolle ist dabei ebenfalls gewünscht.
P7	Störimmunität	Um die Störimmunität der Netzwerke zu ermitteln sollen gezielt Fremdstörungen mit definierbarer Tastung und Störfrequenz eingebracht werden. Hierfür soll die Punkt zu Punkt Testinfrastruktur auf MAC-Ebene eingesetzt werden.
P8	Unterschiedliche Test Bedingungen	Die Messungen und Tests an den Mesh Netzwerken sollen unter unterschiedlichen Bedingungen bezüglich Testumgebung durchgeführt werden. Einerseits soll dies in einem Gebäude der FHNW sein und andererseits in einer Umgebung im Heimbereich.
P9	Test und Validierung	Umfassende Gegenüberstellung und Validierung der Messresultate aller drei Netzwerke. Insbesondere Durchsatz, Antwortzeit, Zuverlässigkeit, Einfachheit der Konfiguration (inkl. Routing), Einfachheit der Ermittlung geeigneter Router-Standorte, Sicherheit und Energieverbrauch.

Tabelle 3.2: Projektziele der Test Mesh Netzwerke

3.3 Steuer- und Auswertesoftware

Projektziele		
Nr.	Ziel	Beschrieb
P1	Ansteuerung Funkmodul	Das BMS steuert via USB oder UART ein BMN an.
P2	Visualisierung Parameter	Die Parameter der Ziele von Kapitel 3.2 und 3.1 sollen vom BMS visualisiert und eingestellt werden können.
P3	User Interface (UI)	Die Testinfrastruktur beinhaltet ein benutzerfreundliches UI.
P4	Konfiguration Mesh-Netzwerk	Das BMS verwaltet und konfiguriert über einen BMN das Mesh-Netzwerk.
P5	Einheitliche Kommunikation von BMS	Das Protokoll und Interface zum BMS soll für alle drei Mesh-Netzwerke einheitlich sein.

Tabelle 3.3: Projektziele der Steuer- und Auswertesoftware

3.4 Zusatzziele

Projektziele		
Nr.	Ziel	Beschrieb
W1	Hardware Testmodul BMN/BSN	Entwicklung einer eigenen Hardware für das Testmodul mit unabhängiger Stromversorgung und dezidierter Strommessung um den Stromverbrauch aufzuzeichnen.
W2	Vergleich SOC	Vergleichen zwischen nRF52840, nRF5340 und weiterer kompatibler SOC's.
W3	Drahtlos Konfiguration	Drahtlose Konfiguration der BSN / BMN Firmware. Somit könnte ein Wechsel zwischen BLE Mesh, Thread und Zigbee während der Runtime möglich werden.
W4	UI für Mesh Test	Für die Mesh Tests soll analog zu den P2P Tests ein User Interface implementiert werden.

Tabelle 3.4: Zusatzziele des Gesamtprojektes

3.5 Lieferobjekte

Zusätzlich zu den Projektzielen, folgen in diesem Kapitel die Lieferobjekte mit dem jeweiligen Fälligkeitsdatum. In der Tabelle 3.5 sind diese aufgelistet.

Nr.	Datum	Lieferobjekt
1	02.03.2020	Abgabe Pflichtenheft, 1. Version
2	08.03.2020	Abgabe Pflichtenheft, definitive Version
3	14.08.2020	Abgabe Fachbericht
4	14.08.2020	Abgabe Paper
5	14.08.2020	Abgabe Testaufbau
6	14.08.2020	Abgabe Factsheet
7	14.08.2020	Abgabe Poster
8	01.09.2020	Projektpräsentation

Tabelle 3.5: Lieferobjekte

4 Projektmanagement

Nachfolgend werden die wichtigsten Punkte bezüglich Projektmanagement behandelt. Im Zentrum stehen dabei vor allem die Aufteilung der Zuständigkeiten 4.1 sowie die Projektplanung 4.2.

Um den Projektfortschritt zu überwachen und allfällige Probleme zu besprechen sind zweiwöchentliche Sitzungen mit den Fachcoaches vorgesehen. Ausserordentliche Termine für Besprechungen werden bei Bedarf zusätzlich definiert.

4.1 Projektaufteilung

Wie bereits im Kapitel 1 erwähnt, stellt die vorliegende Projektarbeit die Bachelorthesen von Raffael Anklin, Robin Bobst und Cyrill Horath dar. Die einzelnen Thesen werden grundsätzlich als separate Projekte umgesetzt welche jedoch einen gemeinsamen Teil beinhalten. Innerhalb dieses gemeinsamen Teils welcher als Framework bezeichnet wird, werden die Arbeitspakete dynamisch an die 3 Mitarbeiter verteilt. In der Tabelle 4.1 ist die genaue Zuweisung der Zuständigkeiten ersichtlich.

Bezeichnung	Inhalt	Zuständigkeit	Kennung
Bluetooth Mesh	Aufbau, Messung und Analyse eines Bluetooth Mesh Netzwerks.	Raffael Anklin	EIT-P-20FS-030
Thread	Aufbau, Messung und Analyse eines Thread Mesh Netzwerks.	Robin Bobst	EIT-P-20FS-031
Zigbee	Aufbau, Messung und Analyse eines Zigbee Mesh Netzwerks.	Cyrill Horath	EIT-P-20FS-032
Framework	Bereitstellung der Test- und Messinfrastruktur. Dazu gehört der Punkt zu Punkt Testaufbau sowie die Steuer- und Auswertesoftware.	Alle	-

Tabelle 4.1: Zuweisung der Zuständigkeiten

4.2 Projektplan

Die Terminpläne zum Projekt *Wireless Controller for Smart Systems* sind im Anhang zu finden. Sie sind in die 4 Teile unterteilt die in der Projektaufteilung 4.1 behandelt wurden. Unter Anhang D ist der Gesamt Terminplan ersichtlich, welcher unter anderem das Framework sowie Projektmanagement Aufgaben beinhaltet. Die Anhänge E, F und G sind die Terminpläne für die 3 persönlichen Bachelorthesen.

4.3 Risikoanalyse

Für die Projektarbeit wurde ausserdem eine schlanke Risikoanalyse erstellt. Diese ist in Anhang H ersichtlich und soll dem Team bei der Beseitigung von Probleme hilfreich sein.

4.4 Projektvereinbarung

Projektcoach und Auftraggeber

Di Cerbo Manuel

Ort, Datum:

Meier Matthias

Ort, Datum:

Unterschrift:

Unterschrift:

Projekt: EIT-P-20FS-030

Anklin Raffael

Ort, Datum:

Unterschrift:

Projekt: EIT-P-20FS-031

Bobst Robin

Ort, Datum:

Unterschrift:

Projekt: EIT-P-20FS-032

Horath Cyrill

Ort, Datum:

Unterschrift:

A Aufgabenstellung

Aufgabenstellung P6 / Thesearbeit FS20

Wireless Controller for Smart Systems

Test und Vergleich von GHz Low Power Mesh Netzwerken

Studierende	Raffael Anklin
	Robin Bobst
	Cyrell Horath
Betreuende Dozenten	Manuel Di Cerbo FHNW Studiengang EIT manuel.dicerbo@fhnw.ch
	Matthias Meier FHNW Studiengang EIT matthias.meier@fhnw.ch
Ausgangslage	<p>Unter den standardisierten Low Power Mesh Netzwerk Protokollen im freien GHz ISM-Band konkurrenzieren sich derzeit vorrangig Bluetooth Mesh, Zigbee sowie Thread.</p> <p>Bezüglich MAC und Physical Layer basieren Zigbee und Thread auf IEEE 802.15.4 wogegen Bluetooth Mesh auf Bluetooth Low Energy (BLE) basiert.</p> <p>Jedes dieser Netzwerkprotokolle hat gewisse Vorzüge: Bluetooth Mesh, dass BLE mittlerweile von jedem Smartphone und Notebook unterstützt wird, Thread aufgrund seiner IPv6 Basis und damit einfachem Übergang ins Internet sowie Zigbee aufgrund seiner etablierten Verbreitung im Smart-Lampenbereich durch Philips, IKEA und Osram.</p> <p>Hauptproblem aller drei Mesh Netzwerkprotokolle ist nebst physikalisch und distanzbedingter Absorption und Reflexion die Störbeeinflussung durch WLAN (WiFi) und andere Netzwerke im GHz Frequenzbereich.</p>

Ziel der Arbeit

In der vorliegenden Arbeit soll zuerst ein praxistaugliches, einheitliches Testframework für alle drei Mesh Netzwerke erstellt werden, wonach die Tauglichkeit aller drei Mesh Netzwerke unter realitätsnahen Bedingungen ermittelt und verglichen werden soll.

Zwecks besserer Vergleichbarkeit sollen alle drei Testnetze das gleiche Radio-Interface als Grundlage verwenden. Aufgrund der guten Unterstützung aller drei Mesh Protokolle als auch dem im vergangenen P5 gesammelten Wissens, sollen hierfür die nRF52840 SoCs der Firma Nordic eingesetzt werden.

Die zu erstellende Testinfrastruktur soll jeweils aus folgenden Teilen bestehen:

1. Zweier **Punkt-Punkt Testinfrastrukturen auf MAC-Ebene** (für BLE und 802.15.4)
Diese sollen es ermöglichen, möglichst einfach zwischen zwei beliebigen Standorten kanalweise die Übertragungsbedingungen zu ermitteln. Diese Testinfrastruktur kann somit einerseits in der Planung eines BT Mesh, Zigbee oder Thread Mesh-Netzwerks als Messinstrument dienen, andererseits auch als Stör-Infrastruktur um gezielt Störungen zu generieren, wie sie von konkurrenzierenden resp. interferierenden BLE und/oder 802.15.4 Netzwerken hervorzurufen werden.
2. Dreier **Test Mesh Netzwerke** (für BT Mesh, Zigbee und Thread)
Mit diesen Test-Netzwerken soll die Robustheit und Mesh-Funktionalität der drei Protokoll-Stacks unter realitätsnahen und nachvollziehbaren Bedingungen ermittelt werden, d.h. bei verschiedenen Netzbelastungen, Netztopologien und gezielt eingebrachter Störungen mit unterschiedlichen Störmustern.
3. Einem Leitrechner, auf welchem eine Python basierte **Steuer- und Auswertesoftware** läuft. Für kürzere Tests und während der Entwicklung kann somit hierfür ein Notebook eingesetzt werden, für Langzeittests hingegen ein Raspberry Pi.
Die Steuer- und Auswertesoftware soll möglichst modular aufgebaut und ein einfaches Commandline-Interface haben, mit Ausgabe der verarbeiteten Daten auf Standard Output.

Zwecks einfacher Bedienung und grafischer Anzeige soll darauf aufbauend ein einfaches Python basiertes Web-Interface realisiert werden.

**Eckpunkte der
Punkt-Punkt
Testinfrastrukturen
auf MAC-Ebene
BLE und 802.15.4**

Beide Varianten sollen konzeptionell wie folgt realisiert werden:

- Der Leitrechner steuert via USB oder UART ein Master-Funkmodul an.
- Das Master-Funkmodul sendet gemäss den Vorgaben des Leitrechners regelmässig MAC-Frames an einen oder mehrere Slave-Funkknoten.
- Der oder die batteriebetriebenen Slave-Funkknoten, bestätigen die MAC-Frames zurück.

Folgende Parameter sollen von der übergeordneten Steuer- und Auswertsoftware konfigurierbar sein:

- Anzahl und ID der Slaves
- Welche BLE resp. 802.15.4 Kanäle zyklisch getestet werden
- Einstellbare Framelänge
- Einstellbare Framerate und Kanalwechselrate
- Einstellbare Sendeleistung
- Nur für BLE: Modulationsart resp. Datenrate (2Mbps ... 125kbps d.h. auch Long Range)
- Nur für 802.15.4: mit/ohne Collision Avoidance (CSMA/CA)

Sowohl master- wie auch slaveseitige Erfassung der Verbindungsqualität (RSSI, Package Loss, Collisions, Noise Level, ...). Die Slaves senden hierzu die erfassten Werte im Rückantwortframe dem Master zurück.

Einfaches und für beide Protokolle (BT und 802.15.4) taugliches Protokoll zwischen Master-Knoten und Auswerterechner.

**Eckpunkte der drei
Mesh Testnetzwerke
(BT Mesh, Zigbee,
Thread)**

Erstellen der drei Mesh-Testnetzwerke mit jeweils ca. 10 Netzknoten wovon:

- Ein Master-Node mit wahlweise USB- oder UART Anschluss zum Auswerterechner
- die restlichen Knoten konfigurierbar z.B. 4 Routing und 5 Low Power Sensor Knoten

Die Sensor-Knoten sollen in einem vom Auswerterechner vorgegebenen parametrisierbaren Intervall Sensorwerte simulieren.

Als "Sensordaten" sollen die Netz-Zustandsdaten übermittelt werden (z.B. Paketnummer zwecks erkennen von verlorenen Datenpaketen, Anzahl Retries, Paketverluste, RSSI, Strombedarf resp. aktive CPU- und Radio-Zeiten, ...)

Das Protokoll und Interface zum Leitrechner soll wenn möglich für alle drei Mesh Netze einheitlich sein.

Um die Störimmunität der Netze zu ermitteln sollen auch gezielt "Fremdstörungen" im eingebracht werden, mit definierbarer Tastung und Störframelänge. Hierfür sollen die "Punkt-Punkt Testinfrastrukturen auf MAC-Ebene" eingesetzt werden.

Umfassende Gegenüberstellung und Validierung aller drei Netzwerke in einem FHNW Gebäude, insbesondere Durchsatz, Antwortzeit, Zuverlässigkeit, Einfachheit der Konfiguration (inkl. Routing), Einfachheit der Ermittlung geeigneter Router-Standorte, Sicherheit und Energieverbrauch, etc.

**Teamaufteilung,
Fachberichte und
Bewertung**

- Jedes Teammitglied realisiert eines der drei Test Mesh Netze.
- Bezüglich Aufteilung der restlichen Arbeiten einigen sich die Teammitglieder untereinander.
- Die Thesisnote setzt sich jeweils hälftig aus einer Individualnote und einer Teamnote zusammen.
- Die Fachberichte resp. Dokumentation setzt sich zusammen aus:
 - Dokumentation der **individuell erstellten Teile**
 - Dokumentation der **gemeinsam erstellten Teile**
 - Einem **Paper** mit Dokumentation und Interpretation der durchgeführten Tests
- Alles Sourcen sollen als Open Source auf GitHub oder GitLab veröffentlicht werden.

Pflichtenheft und Projektvereinbarung

Nach Projektstart soll innert ca. 1 Monat ein technisches Pflichtenheft erarbeitet werden beinhaltend:

- Lösungskonzept und Spezifikation des zu erstellenden Systems,
- Formulierung von Arbeitspaketen (typisch 5-15) und Meilensteinen,
- Zeitplan in Form eines Gantt-Diagrammes.

Projektmanagement, Kommunikation, Abgabetermine, Bewertung:

Das Projekt soll von einem schlanken, ergebnisorientierten Projektmanagement begleitet werden.

Arbeitgeber und betreuender Dozent sollen periodisch (mind. alle 3 Wochen) über den Stand der Arbeiten sowie allfälliger Abweichungen zum Pflichtenheft und Projektplan informiert werden.

Es finden mindestens folgende Meetings statt:

- Kickoffmeeting
- Besprechung Pflichtenheft/Projektvereinbarung
- Schlusspräsentation

Bei Bedarf können mehr Meetings durchgeführt werden.

Betreffend Fachbericht, Ausstellung der Thesearbeit (inkl. Erstellen eines Factsheets und Posters) sowie Verteidigung und Bewertung gelten die Vorgaben und Richtlinien der FHNW, Hochschule für Technik.

Termine

Es gelten die offiziellen Termine der FHNW, Hochschule für Technik

- Die Thesearbeit startet KW 8/2020 und umfasst 360 Arbeitsstunden für jeden Probanden (inkl. Erstellung Fachbericht).
- Der Abgabetermin der Arbeit (Fachbericht) ist am Tag der Bachelor-Ausstellung 14.8.2020.
- Die Verteidigung (d.h. Präsentation und Diskussion der Thesis vor den Betreuern und externem Experten) findet gemäss Semesterplanung der FHNW voraussichtlich in der KW36 oder KW37 statt. Der genaue Termin wird noch bekannt gegeben.

B Test Kriterien Point to Point auf MAC Ebene

Index Messung	MAC-1	MAC-2	MAC-3	MAC-4	MAC-5	MAC-6	MAC-7	MAC-8	MAC-9
Bezeichnung	Latency Time	Response Time	Data Transmission Rate	Noise Level Detection	RSSI	Packet-loss	Active radio-time	Active CPU-time	Theoretical power consumption
Beschreibung	Bestimmung der Latenzzeit eines Node	Bestimmung der Antwortzeit eines Node	Bestimmen der Datenübertragungsrate	Bestimmung des Noise Level in dbm in Verschiedenen Kanälen	Bestimmung des Received Signal Strength Indicator	Bestimmung der Anzahl verlorenen Pakete	Bestimmung der Aktiven Radio Zeiten	Bestimmung der Aktiven CPU Zeit	Bestimmung der theoretischen Leistungsaufnahme.
Messgrösse	Latenzzeit	Antwortzeit	Datenübertragungsrate	Störsignalleistung	Empfangsleistung	Paketverlust	Zeit	Zeit	Leistung
Einheit	Millisekunden (ms)	Millisekunden (ms)	kBit/s	dBm	dBm	Verhältnis gesendete Pakete zu velorene Pakete in %	ms	s	mW
Vorgehen	Zu Beginn findet eine Zeit sowie Kanal-Synchronisation zwischen Master und den Slaves statt. Anschliessend sendet der Master einzeln Testpakete mit der Sendezeit T1 an die Nodes. Jeder Node vergleicht die Empfangszeit T2 mit der Sendezeit T1 und schickt die Differenz als Latenzzeit dem Master zurück.	Zu Beginn findet eine Zeit sowie Kanal-Synchronisation zwischen Master und den Slaves statt. Anschliessend sendet der Master einzeln Testpakete mit der Sendezeit T1 an die Nodes. Jeder Node sendet eine Empfangsbestätigung an den Master zurück. Dieser vergleicht die Empfangszeit T2 mit der Sendezeit T1 und generiert daraus die Antwortzeit.	Zu Beginn findet eine Zeit sowie Kanal-Synchronisation zwischen Master und den Slaves statt. Anschliessend sendet der Master einzeln Testpakete mit der Sendezeit T1 und Zufallsdaten in verschiedener Längen an die Nodes. Jeder Node speichert die Empfangszeit T2 nach vollständig erhaltenem Paket ab und bildet mit der Sendezeit T1 die Differenz (Latenzzeit). Aus dem Quotient der Datenmenge und Latenzzeit wird die Datenübertragungsrate gebildet. Diese meldet der Node dem Master zurück.	Die Nodes messen das Noise Level auf allen Kanälen auf Anfrage des Masters und senden dieses an den Master zurück. Dabei ist jeweils nur ein Node aktiv womit nur Störsignale erfasst werden.	Der RSSI Wert wird von den verschiedenen Nodes erfasst und als Payload den Nachrichten mitgegeben und dem Master zugeschickt.	Die Paketnummer vom empfangen Signal wird ausgelesen und mit der Tatsächlichen Paketnummer, die in der Payload mit geliefert wird verglichen. Das Verhältnis zwischen den Werten stellt den Paketverlust dar.	Beim Einschalten und Ausschalten der Rx- / Tx-Schnittstelle wird ein Timer gestartet bzw. gestopt, so wird die aktive Radio Zeit ermittelt.	Beim Ein- und Ausschalten der CPU soll ein timer gestartet bzw. gestopt werden, so wird die aktive CPU Zeit gemessen.	Anhand der gemessenen Radio und CPU Zeiten wird die Leistung berechnet.
Störfaktoren	Umliegende Kommunikationsgeräte, welche das 2.45GHz ISM Band benutzen. Gezielte Störung des Testsetups sind nicht vorgesehen.								
Anzahl Wiederholungen	1-10	1-10	1-10	1-10	1-10	1-10	1-10	1-10	1-10
Einstellbare Parameter	Modulationsarten (Ble1Mbits, Ble2Mbits, BleLR, IEEE802.15.4), Kanäle, Ziel Node(s), Anzahl Wiederholungen	Modulationsarten (Ble1Mbits, Ble2Mbits, BleLR, IEEE802.15.4), Kanäle, Ziel Node(s), Anzahl Wiederholungen	Modulationsarten (Ble1Mbits, Ble2Mbits, BleLR, IEEE802.15.4), Kanäle, Ziel Node(s), Anzahl Wiederholungen	Kanäle, Ziel Node(s), Anzahl Wiederholungen	Kanäle, Ziel Node(s), Anzahl Wiederholungen	Modulationsarten (Ble1Mbits, Ble2Mbits, BleLR, IEEE802.15.4), Kanäle, Ziel Node(s), Anzahl Wiederholungen	Ziel Node(s), Anzahl Wiederholungen	Ziel Node(s), Anzahl Wiederholungen	Ziel Node(s), Anzahl Wiederholungen

C Test Kriterien Mesh Netzwerke

Index Messung	Mesh-1	Mesh-2	Mesh-3	Mesh-4	Mesh-5	Mesh-6	Mesh-7	Mesh-8	Mesh-9	Mesh-10
Bezeichnung	Latency Time	Number of hops	Data Transmission Rate Unacknowledged	Data Transmission Rate Acknowledged	RSSI	Packet-loss	Active radio-time	Active CPU-time	Theoretical power consumption	Number of retries
Beschreibung	Bestimmung der Latenzzeit von Aktor zu Sensor über Anzahl Hops.	Bestimmung der Anzahl Hops, die eine Nachricht nehmen musste.	Bestimmen der Datenübertragungsrate (Unbestätigt)	Bestimmen der Datenübertragungsrate (Bestätigt)	Bestimmung des RSSI von verschiedenen Nodes	Bestimmung der Anzahl verlorenen Pakete	Bestimmung der Aktiven Radio Zeiten	Bestimmung der Aktiven CPU Zeit	Bestimmung der theoretischen Leistungsaufnahme.	Anzahl Retries
Messgrösse	Latenzzeit	n = Anzahl Hops	Datenübertragungsrate	Datenübertragungsrate	Empfangsleistung	Paketverlust	Zeit	Zeit	Leistung	n = Anzahl Retries
Einheit	Millisekunden (ms)	-	kBit/s	kBit/s	dBm	Verhältnis gesendete Pakete zu verlorene Pakete in %	Millisekunden (ms)	Sekunden (s)	Milliwatt (mW)	-
Vorgehen	Die Latenzzeit wird immer von einem Sensor zu einem Aktor gemessen, z.B. von einem Lichtschalter zum Licht. Wenn die Nachricht vom Sensor gesendet wird, wird ein Timestamp als Payload der Nachricht hinzugefügt. Beim Aktor werden weitere Timestamps zum Payload hinzugefügt und dem Sensor als Acknowledge zurückgeschickt. Im Sensor wird danach die Latenzzeit anhand der Timestamps berechnet.	Auf einem Node werden die Next Hop Informationen lokal gespeichert. Diese Information wird der Nachricht als Payload mitgegeben, um am Ziel Node auszuwerten wie viele Hops die Nachricht genommen hat.	Es werden Datenpakete verschiedener Länge [1Byte - ca. 1MByte] zufällig generiert. Anschliessend wird wie bei Mesh 1 eine Zeitsynchronisation durchgeführt, dabei wird zusätzlich die Grösse der Datenmenge angegeben. Nach Bestätigung der Bereitschaft beginnt der Sensor mit der Übertragung der Datenpakete. Wurde das erste Datenpaket erhalten, so wird dies gepuffert und die Empfangszeit T2 gespeichert. Ist die vollständige Datenmenge beim Node angekommen wird die Differenz aus der aktuellen Zeit und T2 gebildet. Diese bestimmt die Übertragungszeit. Anschliessend wird diese dem Sensor zurückgesendet, welcher die Datenrate aus dem Quotient der Datenmenge und Übertragungszeit bildet.	Der Ablauf ist mit T3 identisch, ausser dass der Erhalt von jedem Datenpaket (ebenfalls segmentiert) bestätigt werden muss. Die Zeitmessung ist mit der letzten Bestätigung an den Sensor abgeschlossen.	Der RSSI Wert wird von den verschiedenen Nodes erfasst und als Payload den Nachrichten mitgegeben und dem Master zugeschickt.	Die Paketnummer vom empfangen Signal wird ausgelesen und mit der Tatsächlichen Paketnummer, die in der Payload mit geliefert wird verglichen. Das Verhältnis zwischen den Werten stellt den Paketverlust dar.	Beim Einschalten und Ausschalten der Rx- / Tx-Schnittstelle wird ein Timer gestartet bzw. gestoppt, so wird die aktive Radio Zeit ermittelt.	Beim Ein- und Ausschalten der CPU soll ein Timer gestartet bzw. gestoppt werden, so wird die aktive CPU Zeit gemessen.	Anhand der gemessenen Radio und CPU Zeiten wird die Leistung berechnet.	Wird das Acknowledge nicht quittiert, wird die Nachricht erneut gesendet. Diese Anzahl Retries werden ermittelt und der Payload mitgegeben.
Störfaktoren	Umliegende Kommunikationsgeräte, welche das 2.45GHz ISM Band benutzen.									
Anzahl Wiederholungen	Periodisch									
Einstellbare Parameter	-	Anzahl Hops kann begrenzt werden	Packetsize	Packetsize	-	-	-	-	-	-
Voraussetzungen	Node muss bereit und konfiguriert sein. Zeit der Nodes muss synchronisiert.	Node muss bereit und konfiguriert sein.								
Allgemeine Bedingungen	Die Tests werden unter belastetem und unbelastetem Mesh-Netzwerk durchgeführt									

D Gesamt Terminplanung

Aufgabe	Arbeitspaket	Anfang	Ende	Wer	Review	17. Feb 20	24. Feb 20	2. Mär 20	9. Mär 20	16. Mär 20	23. Mär 20	30. Mär 20	6. Apr 20	13. Apr 20	20. Apr 20	27. Apr 20	4. Mai 20	11. Mai 20	18. Mai 20	25. Mai 20	1. Jun 20	8. Jun 20	15. Jun 20	22. Jun 20	10. Aug 20	17. Aug 20
PM	Gesamt Zeitplan erstellen	17.02.2020	02.03.2020	Alle																						
PM	Arbeitspakete definieren	24.02.2020	02.03.2020	Alle																						
PM	Arbeitspakete zuweisen	09.03.2020	09.03.2020	Alle																						
Pflichtenheft	Pflichtenheft verfassen	17.02.2020	01.03.2020	Alle																						
Pflichtenheft	Ziele definieren	18.02.2020	02.03.2020	Alle																						
Pflichtenheft	Messgrössen definieren	19.02.2020	02.03.2020	Alle																						
Pflichtenheft	Abgabe Pflichtenheft	02.03.2020	02.03.2020	Alle																						
Konzept	Lösungsvarianten Recherchieren	17.02.2020	02.03.2020	Alle																						
Konzept	Gesamt-Konzept erstellen	24.02.2020	02.03.2020	Alle																						
Konzept	Messkonzept erstellen	24.02.2020	03.03.2020	Alle																						
Konzept	Konzepte fixen	03.03.2020	09.03.2020	Alle																						
Hardware	Material Bestellen	09.03.2020	06.04.2020	Alle																						
Software	Recherche und Einarbeitung	24.02.2020	09.03.2020	Alle																						
Software	Software-Konzept erstellen	09.03.2020	23.03.2020	Alle																						
Software	Bedienungs-Konzept erstellen	09.03.2020	23.03.2020	Alle																						
Software	Schnittstellen der Teilsysteme definieren	16.03.2020	23.03.2020	Alle																						
Software	Programmierung	30.03.2020	19.04.2020	Alle																						
Software	Teilsysteme Testing und Debugging	13.04.2020	19.04.2020	Alle																						
Software	Gesamtsystemtest und Debugging	13.04.2020	19.04.2020	Alle																						
Messungen	Messungen vorbereiten	13.04.2020	19.04.2020	Alle																						
Messungen	Messungen durchführen	20.04.2020	11.05.2020	Alle																						
Messungen	Resultate dokumentieren	23.04.2020	14.05.2020	Alle																						
Messungen	Resultate validieren	01.05.2020	18.05.2020	Alle																						
Fachbericht	Strukturierung / Disposition	30.03.2020	13.04.2020	Alle																						
Fachbericht	Schreiben	13.04.2020	14.08.2020	Alle																						
Fachbericht	Einleitung	13.04.2020	30.06.2020	Alle																						
Fachbericht	Technische Grundlagen	13.04.2020	30.06.2020	Alle																						
Fachbericht	Hardware	13.04.2020	31.07.2020	Alle																						
Fachbericht	Software	13.04.2020	31.07.2020	Alle																						
Fachbericht	Korrigieren / Gegenlesen	31.07.2020	10.08.2020	Alle																						
Fachbericht	Drucken	10.08.2020	10.08.2020	Alle																						
Fachbericht	Abgabe Fachbericht, Factsheet und Plakat	14.08.2020	14.08.2020	Alle																						
Präsentation	Präsentation erstellen	15.08.2020	31.08.2020	Alle																						
Präsentation	Präsentation halten	KW 36/37		Alle																						

Personen:

RAN Raffel Anklin
RBO Robin Bobst
CHO Cyrill Horath

E Bluetooth Mesh Terminplanung

Aufgabe	Arbeitspaket	Anfang	Ende	Wer	Review	17. Feb 20	24. Feb 20	2. Mär 20	9. Mär 20	16. Mär 20	23. Mär 20	30. Mär 20	6. Apr 20	13. Apr 20	20. Apr 20	27. Apr 20	4. Mai 20	11. Mai 20	18. Mai 20	25. Mai 20	1. Jun 20	8. Jun 20	15. Jun 20	22. Jun 20	10. Aug 20	17. Aug 20
PM	Zeitplan erstellen	17.02.2020	02.03.2020	RAN																						
PM	Arbeitspakete definieren	24.02.2020	02.03.2020	RAN																						
Pflichtenheft	Pflichtenheft verfassen	17.02.2020	02.03.2020	RAN																						
Pflichtenheft	Ziele definieren	18.02.2020	02.03.2020	RAN																						
Pflichtenheft	Abgabe Pflichtenheft	02.03.2020	02.03.2020	RAN																						
Konzept	Lösungsvarianten Recherchieren	17.02.2020	02.03.2020	RAN																						
Konzept	Konzept erstellen	24.02.2020	02.03.2020	RAN																						
Konzept	Konzept fixen	03.03.2020	09.03.2020	RAN																						
Hardware	Material Bestellen	09.03.2020	06.04.2020	RAN																						
Software	Recherche und Einarbeitung	24.02.2020	09.03.2020	RAN																						
Software	Software-Konzept erstellen	09.03.2020	23.03.2020	RAN																						
Software	Programmierung	30.03.2020	19.04.2020	RAN																						
Software	Testing und Debugging	13.04.2020	19.04.2020	RAN																						
Messungen	Messungen vorbereiten	13.04.2020	19.04.2020	RAN																						
Messungen	Messungen durchführen	20.04.2020	11.05.2020	RAN																						
Messungen	Resultate dokumentieren	23.04.2020	14.05.2020	RAN																						
Messungen	Resultate validieren	01.05.2020	18.05.2020	RAN																						

Personen:

RAN Raffel Anklin
 RBO Robin Bobst
 CHO Cyrill Horath

F Thread Terminplanung

Aufgabe	Arbeitspaket	Anfang	Ende	Wer	Review	17. Feb 20	24. Feb 20	2. Mär 20	9. Mär 20	16. Mär 20	23. Mär 20	30. Mär 20	6. Apr 20	13. Apr 20	20. Apr 20	27. Apr 20	4. Mai 20	11. Mai 20	18. Mai 20	25. Mai 20	1. Jun 20	8. Jun 20	15. Jun 20	22. Jun 20	10. Aug 20	17. Aug 20
PM	Zeitplan erstellen	17.02.2020	02.03.2020	ROB																						
PM	Arbeitspakete definieren	24.02.2020	02.03.2020	ROB																						
Pflichtenheft	Pflichtenheft verfassen	17.02.2020	02.03.2020	ROB																						
Pflichtenheft	Ziele definieren	18.02.2020	02.03.2020	ROB																						
Pflichtenheft	Abgabe Pflichtenheft	02.03.2020	02.03.2020	ROB																						
Konzept	Lösungsvarianten Recherchieren	17.02.2020	02.03.2020	ROB																						
Konzept	Konzept erstellen	24.02.2020	02.03.2020	ROB																						
Konzept	Konzept fixen	03.03.2020	09.03.2020	ROB																						
Hardware	Material Bestellen	09.03.2020	06.04.2020	ROB																						
Software	Recherche und Einarbeitung	24.02.2020	09.03.2020	ROB																						
Software	Software-Konzept erstellen	09.03.2020	23.03.2020	ROB																						
Software	Programmierung	30.03.2020	19.04.2020	ROB																						
Software	Testing und Debugging	13.04.2020	19.04.2020	ROB																						
Messungen	Messungen vorbereiten	13.04.2020	19.04.2020	ROB																						
Messungen	Messungen durchführen	20.04.2020	11.05.2020	ROB																						
Messungen	Resultate dokumentieren	23.04.2020	14.05.2020	ROB																						
Messungen	Resultate validieren	01.05.2020	18.05.2020	ROB																						

Personen:

RAN Raffel Anklin
 RBO Robin Bobst
 CHO Cyrill Horath

G Zigbee Terminplanung

Aufgabe	Arbeitspaket	Anfang	Ende	Wer	Review	17. Feb 20	24. Feb 20	2. Mär 20	9. Mär 20	16. Mär 20	23. Mär 20	30. Mär 20	6. Apr 20	13. Apr 20	20. Apr 20	27. Apr 20	4. Mai 20	11. Mai 20	18. Mai 20	25. Mai 20	1. Jun 20	8. Jun 20	15. Jun 20	22. Jun 20	10. Aug 20	17. Aug 20
PM	Zeitplan erstellen	17.02.2020	02.03.2020	CHO																						
PM	Arbeitspakete definieren	24.02.2020	02.03.2020	CHO																						
Pflichtenheft	Pflichtenheft verfassen	17.02.2020	02.03.2020	CHO																						
Pflichtenheft	Ziele definieren	18.02.2020	02.03.2020	CHO																						
Pflichtenheft	Abgabe Pflichtenheft	02.03.2020	02.03.2020	CHO																						
Konzept	Lösungsvarianten Recherchieren	17.02.2020	09.03.2020	CHO																						
Konzept	Konzept erstellen	24.02.2020	09.03.2020	CHO																						
Konzept	Konzept fixen	03.03.2020	16.03.2020	CHO																						
Hardware	Material Bestellen	09.03.2020	06.04.2020	CHO																						
Software	Recherche und Einarbeitung	24.02.2020	09.03.2020	CHO																						
Software	Software-Konzept erstellen	09.03.2020	23.03.2020	CHO																						
Software	Programmierung	30.03.2020	19.04.2020	CHO																						
Software	Testing und Debugging	13.04.2020	19.04.2020	CHO																						
Messungen	Messungen vorbereiten	13.04.2020	19.04.2020	CHO																						
Messungen	Messungen durchführen	20.04.2020	11.05.2020	CHO																						
Messungen	Resultate dokumentieren	23.04.2020	14.05.2020	CHO																						
Messungen	Resultate validieren	01.05.2020	18.05.2020	CHO																						

Personen:

RAN	Raffel Anklin
RBO	Robin Bobst
CHO	Cyrill Horath

Ereignis				Risiko ohne Massnahmen			Prävention	Risiko mit Massnahmen			Verantwortlich	Indikator
Nr.	Risiko	Ursachen	Konsequenzen	si	pi	w		si	pi	w		
A	Teammitglied fällt kurzfristig aus	Unvorhergesehener Termin, leichte Krankheit, leichter Unfall	Weniger Personalressourcen, kleiner Mehraufwand	3	2	6	Reservezeit einplanen, Transparenter Informationsfluss im Team	1	2	2	CHO	Abwesenheit
B	Teammitglied fällt längerfristig aus	Militärdienst, schwere Krankheit, Studienabbruch, schwerer Unfall	Grössere Umplanung, Neuverteilung der Arbeiten	3	2	6	Strukturierte Datenablage, guter Kommunikationsfluss	1	2	2	CHO	Abwesenheit
C	Datenverlust oder Zugriffsprobleme	Löschung der Projektdaten, Unzugänglichkeit von Onedrive, kein Internetverbindung	Zugriff auf Daten nicht möglich, Sämtliche Projektdaten nicht mehr vorhanden	2	1	2	Regelmässige Backups, Dokumente zusätzlich lokal abspeichern	1	1	1	RAN	Arbeiten auf dem Stand des letzten Backups
D	Software kann nicht mehr ausgeführt werden	Datenverlust, Softwareupdate	Schlimmstenfalls Verlust der gesamten Arbeit, vorübergehende Arbeitspause bis Update komplett	2	3	6	Fertige Softwareteile werden zusätzlich im Onedrive gespeichert (Revisionsverwaltung) / Github	2	1	2	RAN	Fehlermeldung
E	Softwarekonzept nicht ausführbar	Mangelnde Vorkenntnisse, schlechte Planung	Überdenken der Arbeit, Verzug der Arbeiten	2	2	4	Mit Software-Fachcoach besprechen	1	1	1	RAN	Nicht funktionierendes Skript
F	Softwareprojekt von Funk Masternode und Leitnehmer nicht verknüpfbar	Schnittstelle wurde nicht korrekt eingehalten	Verzögerung der Arbeit, Mehraufwand	2	2	4	Kommunikation zwischen Softwareteam	1	1	1	RAN	Softwareteam können Vorhaben nicht weiterführen
G	Zu komplizierter Sachverhalt	Inhalt kann nicht umgesetzt werden	Stillstand der Arbeit, Projekt nicht durchführbar	1	3	3	Früzeitige Besprechung mit Fachcoaches	1	1	1	RAN	Kein Weiterkommen
H	Soziale Spannungen im Team	Meinungsverschiedenheiten, schlechte Arbeitsaufteilung, keine Kompromissbereitschaft	Motivation sinkt, Arbeitsmoral sinkt, schlechte Projektarbeit, unzufriedener Arbeitgeber	2	2	4	Gegenseitige Kontrolle, Fehler offen im Team besprechen, Konstruktive Kritik	2	1	2	BOB	Schlechte Arbeitsmoral
I	Mangelnde Kommunikation	zu wenig Sitzungen, Angst vor Demütigung	Schlechtes Zusammenspiel, schlechtere Arbeit	2	1	2	Häufigere Sitzungen, höhere Wertschätzung der einzelnen Teammitglieder	1	1	1	BOB	Zurückhaltung
J	Nicht Termingerechte Abgabe der Arbeiten	Faulheit, mangelnder Einsatz, falsche Prioritäten, schlechte Projektführung	Terminplan kann nicht eingehalten werden	2	2	4	striktere Projektführung, gegenseitige Kontrolle, frühzeitiges Melden	2	1	2	BOB	Schlechte Arbeitsmoral
K	Qualitativ minderwertige Arbeit	Faulheit, mangelnder Einsatz, schlechter Teamgeist	Mehraufwand, Qualitativ ungenügende Arbeit, Zeitliche Probleme	2	2	4	Gegenkontrolle der Arbeiten	2	1	2	CHO	Schlechte Arbeitsmoral
L	Schlechte Terminplanung	Aufwand unterschätzt, keine Reserve eingeplant	Mehraufwand, Überarbeitung des Terminplans, Engpässe	3	2	6	Genug Reservezeit einplanen	2	1	2	CHO	Terminverzug

si= Eintrittswahrscheinlichkeit

pi= Auswirkung

H Risikoanalyse