

**Fachwissenschaftliche Projektarbeit**

**ENTER TITLE**

**ENTER PARTICIPANTS**

---

Betreuer : Nils Dorsch, M.Sc.  
Fabian Kurtz, M.Sc.  
Eingereicht : 14. März 2016



# Inhaltsverzeichnis

<b>1</b>	<b>Zur Nutzung dieser Vorlage</b>	<b>1</b>
1.1	Grundlagen . . . . .	1
1.2	Vorraussetzungen . . . . .	1
1.2.1	Texmaker Konfiguration . . . . .	2
1.3	Latex Crashkurs . . . . .	3
1.3.1	Dokumentinterne Referenzen . . . . .	4
1.3.2	Abkürzungen . . . . .	5
1.3.3	Einheiten und Zahlen im Text . . . . .	5
1.3.4	Tabellen . . . . .	6
1.3.5	Abbildungen . . . . .	6
1.3.6	Literaturreferenzen . . . . .	7
1.3.7	Beispielcode: Formeln . . . . .	7
1.3.8	Beispielcode: Komplexe Tabellen . . . . .	8
1.3.9	Beispielcode: Gedrehte Tabelle . . . . .	8
<b>2</b>	<b>Einführung</b>	<b>10</b>
<b>3</b>	<b>Kapitel 1</b>	<b>11</b>
3.1	Kurzfassung . . . . .	11
3.2	Abschnitt 1 . . . . .	11
3.2.1	Unterabschnitt 1 . . . . .	11
<b>4</b>	<b>Kapitel 2</b>	<b>12</b>
4.1	Kurzfassung . . . . .	12
4.2	Abschnitt 1 . . . . .	12
4.2.1	Unterabschnitt 1 . . . . .	12
<b>5</b>	<b>Kapitel 3</b>	<b>13</b>
5.1	Kurzfassung . . . . .	13
5.2	Abschnitt 1 . . . . .	13
5.2.1	Unterabschnitt 1 . . . . .	13
<b>6</b>	<b>Kapitel 4</b>	<b>14</b>
6.1	Kurzfassung . . . . .	14
6.2	Abschnitt 1 . . . . .	14
6.2.1	Unterabschnitt 1 . . . . .	14
<b>7</b>	<b>Kapitel 5</b>	<b>15</b>
7.1	Kurzfassung . . . . .	15
7.2	Abschnitt 1 . . . . .	15
7.2.1	Unterabschnitt 1 . . . . .	15

<b>8 Kapitel 6</b>	<b>16</b>
8.1 Kurzfassung . . . . .	16
8.2 Abschnitt 1 . . . . .	16
8.2.1 Unterabschnitt 1 . . . . .	16
<b>9 Fazit</b>	<b>17</b>
<b>A Abkürzungsverzeichnis</b>	<b>18</b>
<b>Literaturverzeichnis</b>	<b>19</b>

# 1 Zur Nutzung dieser Vorlage

## 1.1 Grundlagen

Dies ist eine Latex Vorlage zur Dokumentation von Projektarbeiten. Dieses How-To ist mit derselben Vorlage generiert die auch für die Projektarbeit selbst genutzt werden sollte. Die **main.tex** enthält die Package-Konfiguration und legt fest welche Dateien eingebunden werden sollen. Weiterhin ist ein kleiner Latex Crashkurs enthalten, der sowohl die Grundlagen von Latex vorstellt als auch einige Grundfunktionen der verwendeten Latex Packages. Sämtlicher Code der zur Generierung dieser Einführung genutzt wurde, ist in der Datei **tex/example.tex** zu finden. Er sollte als Referenz herangezogen werden.

## 1.2 Vorraussetzungen

Um diese Latex Vorlage zu nutzen wird zunächst eine (La)-Tex Distribution benötigt. Tabelle 1.1 zeigt eine Übersicht der bekanntesten kostenlosen Distributionen.

Tabelle 1.1: Die bekanntesten Tex Distributionen

Betriebssystem	(La-)Tex Distribution	Homepage
Windows	MikTex	<a href="http://www.miktex.org">www.miktex.org</a>
Mac OSX	MacTex	<a href="http://www.tug.org/mactex">www.tug.org/mactex</a>
Linux	TeX Live	<a href="http://www.tug.org/texlive">www.tug.org/texlive</a>

Die Distribution kümmert sich normalerweise um das Herunterladen von Latex-Plugins (Packages) und die Bereitstellung der Grundfunktionen. Generelles Arbeiten an den Tex Dokumenten ist zwar auch ohne Distribution möglich, aber aus folgendem Grund nicht sinnvoll wenn Änderungen vorgenommen werden.

Zwar sind .tex Dateien, die als Standardformat für Latex Quellcode genutzt werden, simple Textdateien, die auch mit Notepad oder jedem anderen Texteditor bearbeitet werden können. Um allerdings ein fertiges (PDF) Dokument zu erhalten muss der Latex Quellcode kompiliert werden. Zu diesem Zwecke werden meist Latex Integrated Development Environments (IDEs) verwendet.

Eine gute, kostenlose und einfach zu bedienende Tex IDE ist Texmaker<sup>1</sup>, die für Linux, Windows und Mac erhältlich ist. Texmaker hat einige Funktionen, die gerade Einsteigern den Umgang mit LaTeX erleichtern.

Da der Einstieg in LaTeX recht schwer ist sind im Internet einige Tutorials zu finden, die allerdings nicht alle eins zu eins auf diese Vorlage und die konkrete Konstellation an verwendeten Packages angewendet werden können.

### 1.2.1 Texmaker Konfiguration

Bevor Texmaker erfolgreich genutzt werden kann um PDF Dateien zu erzeugen müssen die nachfolgenden Einstellungen gemacht werden.

1. build als Outputordner
2. Bibtex Einstellen
3. main.tex als Masterdatei

Die nachfolgenden Grafiken (Abbildung 1.1 zeigen wichtige Einstellungen, die grundsätzlich in Texmaker eingestellt werden sollten.

Das gezeigte Menü findet sich unter "Optionen -> Texmaker konfigurieren".

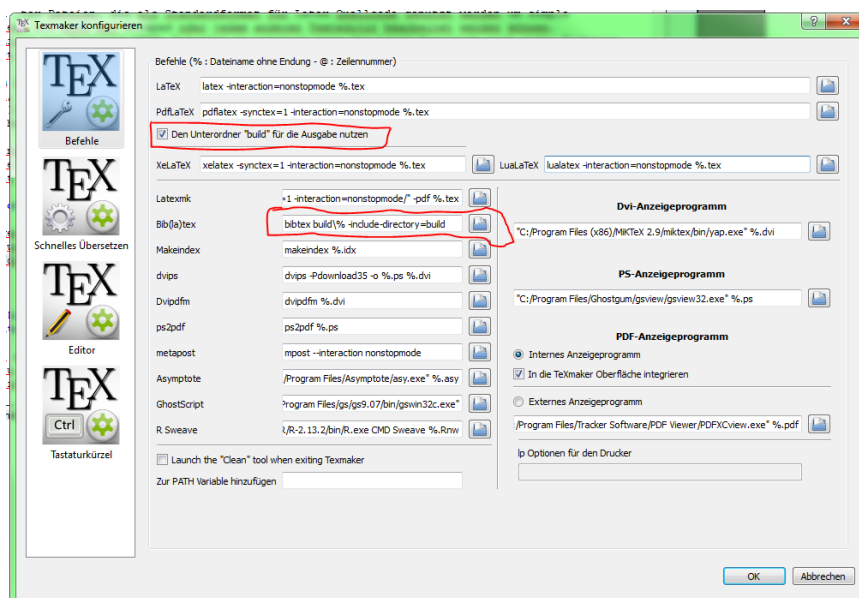


Abbildung 1.1: Texmaker Konfiguration Seite 1: Build Ordner einstellen

Die Konfiguration in Abbildung 1.1 (rot umkreist) sorgt dafür, dass die gebaute PDF-Datei im Ordner **build** erstellt wird, dies ist hilfreich beim Einsatz des Git Versionsverwaltungssystems, da

<sup>1</sup>[www.xm1math.net/texmaker](http://www.xm1math.net/texmaker)

Änderungen in diesem Ordner ignoriert werden können. Weiterhin ist es dann deutlich ersichtlich welche Dateien zum Dokument gehören und welche durch den PDF-Bauprozess erzeugt werden. Weiterhin muss das Literaturverzeichnis aus der Datei in **build** erstellt (untere Einstellung). Die Einstellung Abbildung 1.2 sorgt dafür, dass beim "Schnellen Übersetzen" (Hotkey: F1), sowohl die PDF als auch das Literaturverzeichnis generiert werden. Da dies vergleichsweise lange dauert, kann man nun F6 (PDF Latex bauen) verwenden falls es nur darum geht schnelle Layout-/Formatchecks zu machen.

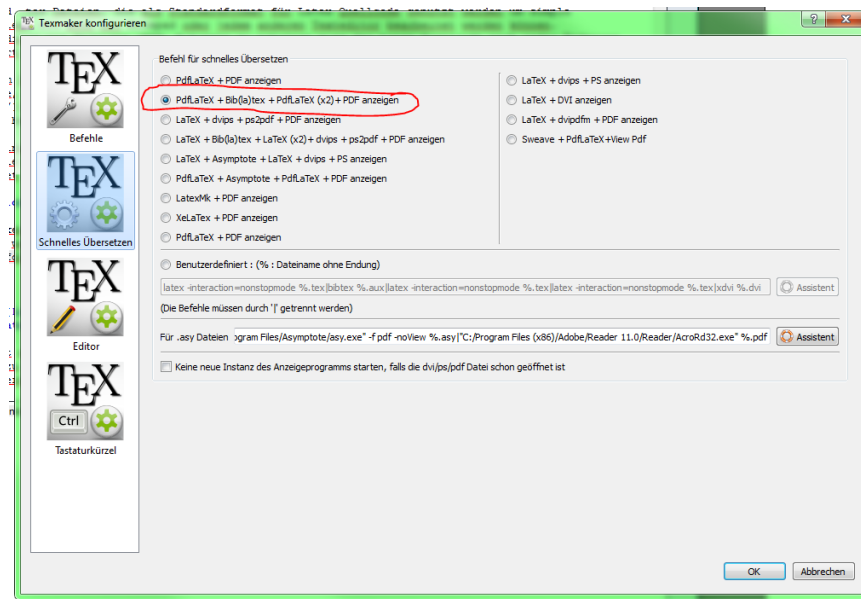


Abbildung 1.2: Texmaker Konfiguration Seite 2: Schnelles Übersetzen

Der letzte Schritt zur Einrichtung ist das Einstellen der Masterdatei. Da diese Vorlage aus mehreren Dateien besteht, ist es nötig das Hauptdokument (**main.tex**) zur Masterdatei zu erklären. Dann ist es auch möglich beim editieren einer anderen Datei als **main.tex** den Bauprozess zu starten. Dazu muss nach jedem Neustarten von Texmaker, die main.tex geöffnet werden und "Optionen -> Aktuelle Datei zur Masterdatei erklären" gewählt werden.

## 1.3 Latex Crashkurs

Latex ist ein Textsatzsystem, dass es dem Nutzer (zumindest theoretisch) ermöglicht sich auf die Inhalte zu konzentrieren und explizit zu formatieren, was gegenüber Textbearbeitungsprogrammen (wie Word) den Vorteil hat, dass Gründe für gebrochene Formatierung klarer ersichtlich sind.

Befehle beginnen mit \, gefolgt vom Befehl. Weiterhin gibt es Befehle, wie `\begin{equation} ... \end{equation}`, die eine Umgebung definieren, welche explizit wieder geschlossen werden muss

(wie z.B. Tabellen, Formeln).

Zusätzliche Zeilenumbrüche sind ebenso explizit und werden mit `\verzwungen`.

**Fettgedruckter Text** lässt sich mit `\textbf{fettgedruckender Text}` erreichen (alternativ: in Texmaker den Text markieren per Strg+B drücken).

Fast alle Befehle und Strukturen, die in einer wissenschaftlichen Arbeit genutzt werden, werden diesem Beispieldokument verwendet und der verwendete Quellcode kann einfach als Vorlage übernommen werden. Der Quellcode ist in **tex/example.tex** zu finden.

**Tipp:** Auch wenn Tex Zeilenumbrüche von selbst einfügt, ist es sinnvoll jeden Satz in eine einzelne Zeile zu schreiben, dies erleichtert einpflegen von Korrekturen und das Finden von Kompilierfehlern.

Latex Quellcode lässt sich mit `%` (aus-)kommentieren was die Fehlersuche erleichtert. Texmaker hat dafür eine Tastenkombination, sodass sich auch mehrere Zeilen komfortabel auskommentieren lassen.

### 1.3.1 Dokumentinterne Referenzen

Interne Referenzen werden mit `\Fref{präfix:...}` eingefügt, hierzu wird das **fancyref** Package verwendet. Um sich später auf einen Abschnitt zu berufen zu können (und damit die Referenzen zu Links werden), müssen zunächst Elemente wie Tabellen, Überschriften und Grafiken mit einem Label versehen werden. Labels werden mit `\label{präfix:...}` gesetzt.

Tabelle 1.2 gibt einen Überblick über die üblichen Präfixe und die resultierenden Referenzen.

Tabelle 1.2: Die wichtigsten Fancyref Präfixe

Element	Label-Präfix	Resultat
Tabelle	tab:	Tabelle ??
Abbildung	fig:	Abbildung ??
Formel	eq:	Gleichung (??)
Quellcode	lst:	Quelltext Ausschnitt ??
Kapitel	chap:	Kapitel ??
Abschnitt	sec:	Abschnitt ??
Unterabschnitt	sub:	Unterabschnitt ??

**Tipp:** Es spart Zeit den Labels sinnvolle Namen zu geben. Es bietet sich bei Kapiteln, Abschnitten etc. an, einfach die Überschrift mit `_` statt den Leerzeichen zu übernehmen und bei



Grafiken den Dateinamen (ohne Endung) zu verwenden. So ist es deutlich leichter Text "herunterzuschreiben" ohne hin- und herspringen zu müssen um Labels nachzuschauen.

### 1.3.2 Abkürzungen

Latex hat die Möglichkeit eure genutzten Abkürzungen für euch zu sammeln und diese dynamisch einzubinden (z.B. ausschreiben bei der ersten Nutzung). Hierzu wird das **acronym** Package genutzt.

Vor der Nutzung im Text müssen die Abkürzungen oder "Akronyme" in die **acronyms.tex** eingetragen werden.

Wenn sie eingetragen sind können sie wie folgt verwendet werden.

- `\ac{abkürzung}`: falls die Abkürzung noch nicht verwendet wurde, wird das Wort ausgeschrieben und das Akronym in Klammern dahinter gedruckt.
- `\acs{abkürzung}`: nur die Abkürzung.
- `\acs{abkürzung}`: nur das ausgeschriebene Wort.
- `\acp{abkürzung}`: plural der Abkürzung.

### 1.3.3 Einheiten und Zahlen im Text

Die um die Formatierung von Einheiten im Text zu gewährleisten wird das **siunitx** Package verwendet.

Zahlen mit Einheiten wie  $9,81 \text{ m s}^{-2}$  oder 100 Mbps in Texten und Tabellen lassen sich komfortabel durch ausschreiben der Einheiten einbinden. So ergibt beispielsweise `\SI{9,81}{\meter \per \second \squared}` den Text  $9,81 \text{ m s}^{-2}$ .

Außerdem sollten einheitenlose Zahlen mit `\si{Zahl}` geschrieben werden um eine einheitliche Formatierung zu erhalten.

Eine komplette Liste der Einheiten lässt sich in der Dokumentation des Packages finden. Weiterhin sind in dieser Vorlage folgende Einheiten definiert:

Tabelle 1.3: Zusätzliche Einheiten für siunitx

<code>\bitf</code>	bit
<code>\bytef</code>	Byte
<code>\bps</code>	(bps) bit per second
<code>\kbps</code>	kbps
<code>\Mbps</code>	Mbps
<code>\Gbps</code>	Gbps

### 1.3.4 Tabellen

Das Einfügen von Tabellen kann auf zweierlei Wegen geschehen, zum einen durch Textmaker (unter Assistent->Tabellen Assistent) oder durch direkte Bearbeitung des Quellcodes, wobei letzteres direkte Kontrolle ermöglicht.

Da die Texmaker-Variante relativ simpel ist wird hier näher auf die Quellcode-Variante eingegangen. Zunächst werden Tabellen in einer **table** Umgebung erzeugt, die die Tabellenunterschrift, das Label und die eigentliche Tabelle enthält. Die Tabelle ist als **tabular** Umgebung angelegt, die als parameter die Spaltenkonfiguration erhält.

Zum Beispiel: `\begin{tabular}{|L{5cm}|C{3cm}|C{3cm}|}`

- | Zeichen zeigen wo vertikale Linien zwischen den Spalten eingefügt werden sollen um die Tabelle zu erhalten.
- L/C/R{x cm} erzeugt jeweils eine linksbündige/zentrierte/rechtsbündige Spalte mit Breite x cm
- Zeilen werden manuell durch `\hline` getrennt
- Die Trennung von Inhalten in jeder Zelle einer Reihe wird durch das reservierte & Zeichen angezeigt, während `\zum` beenden der Reihe genutzt wird.
- Um einen Zeilenumbruch innerhalb einer Tabellenzelle zu erzwingen kann `\newline` verwendet werden.

Der Quellcode von Tabelle 1.3.3 und Tabelle 1.3.8 geben einen guten Überblick über das Erstellen von Tabellen.

**Tipp:** Auch in Tabellen gilt, dass Leerzeilen von Latex ignoriert werden, sodass Tabellen durch gezieltes auseinanderschreiben jeder Zeile/Reihe übersichtlich gegliedert werden können, das macht den Quellcode lesbarer und erleichtert die Fehlersuche.

Ein gutes Beispiel hierfür ist Tabelle 1.3.9.

### 1.3.5 Abbildungen

Um einen qualitativ hochwertigen Druck zu erreichen sollten die verwendeten Grafiken entweder als Vektorgrafik oder eine Rastergrafik mit mindestens 300 Dots per Inch (DPI) sein.

Die beste Variante ist der Export von Grafiken als PDF-Datei. Um Grafiken nun einzubinden muss der Name der Datei **ohne Dateiendung** angegeben werden. Die Grafiken sollten im **images** Ordner abgelegt werden, sie können dann mit **images/namedergrafik** referenziert werden. Abbildung 1.1 ist eine derartig eingebundene (JPEG)-Datei.

Um den Quellcode zum Einbinden der oben genannten Grafik zu verstehen hier ein kurzer Walkthrough:

Tabelle 1.4: Erklärung für den Code zum Einbinden von Grafiken

Code	Erklärung
<code>\begin{figure}[h]</code>	beginne Grafikumgebung; h <sup>2</sup> : dieses Objekt wird grob in der Nähe seines Quelltextes erscheinen
<code>\centering</code>	zentriert
<code>\vspace{12pt}</code>	12 pt vertikaler Abstand
<code>\caption {Texmaker Konfiguration....}</code>	die Bildunterschrift
<code>\label{fig:example_texmaker_conf1}</code>	das Label
<code>\end{figure}</code>	Bildumgebung beenden

Dieser Beispielcode lässt sich in fast allen Fällen zum Einbinden von Grafiken verwenden.

### 1.3.6 Literaturreferenzen

Diese Vorlage verwendet die IEEE Zitierweise, so referenziert [1] den ersten Eintrag des Literaturverzeichnisses. Um Literatur referenzieren zu können muss zunächst eine Bibliotheksdatei (Bibtex-Datei; Endung .bib) erzeugt werden, die Angaben zur Literatur enthält.

Dieses Dokument benutzt die Bibtex-Datei **bibliography/literature.bib**, die direkt erweitert werden kann. Da Bibtex-Markup relativ Zeitaufwendig zu schreiben und zu durchsuchen ist, bieten sich Referenzmanager an. Ein einfaches und kostenloses Literaturmanagement-Tool ist das Java-basierte JabRef [1].

Um nun Literatur zu referenzieren wird der `\cite{schlüssel}` verwendet, wobei schlüssel hier für den eindeutigen Identifier (Bibtexkey) der betroffenen Referenz steht.

**Tipp:** Seiten wie z.B. IEEE Xplore<sup>3</sup> bieten auch direkt die Möglichkeit Bibtex Zitate zu angebotenen Veröffentlichungen herunterzuladen, sodass es nicht nötig ist die Felder selbst auszufüllen. Damit wird das Referenzieren stark vereinfacht.

**JabRef Tipp:** Mit einem Rechtsklick auf eine Referenz hat man die Möglichkeit direkt den Schlüssel in die Zwischenablage zu kopieren.

### 1.3.7 Beispielcode: Formeln

Die folgenden Formeln zeigen beispielhaft wie Gleichungen in Latex eingebunden werden können. Das Einbinden von Variablen oder kurzen Formeln in Fließtext ist durch Nutzung der in-text Math-

<sup>2</sup>H: wäre definitiv exakt hier (kann zu sehr unschönen Umbrüchen führen)

<sup>3</sup><http://ieeexplore.ieee.org>

Umgebung möglich. Diese Umgebung lässt sich durch  $\backslash(\text{Formel})$  definieren.

Zum Beispiel so:  $n_{receiver} = 5$ .

$$L_{receiver,avg}[\frac{bit}{s}] = n_{receiver} \cdot 8 \cdot l_{po} \cdot \frac{1000}{ITT_{avg}} \quad (1.1)$$

$$L_{controllink,avg}[\frac{bit}{s}] = L_{sender,avg}(n_{sender}) + L_{receiver,avg}(n_{receiver}) \quad (1.2)$$

$$L_{Controller,avg}[\frac{bit}{s}] = \sum_{i=0}^m L_{controllink,avg,i} \quad (1.3)$$

$$L_{Controller,avg}(n, ITT_{avg})[\frac{bit}{s}] = \frac{n \cdot l_{po} + n \cdot l_{pi}}{ITT_{avg}} \quad (1.4)$$

$$n_{Saturation}(r_{ControlNetwork}, ITT) = \frac{r_{ControlNetwork} \cdot ITT_{avg}}{l_{pi}} \quad (1.5)$$

### 1.3.8 Beispielcode: Komplexe Tabellen

Diese Tabelle enthält verbundene Zellen, die durch Nutzung des **multicol** Packages erreicht wurden.

Tabelle 1.5: Maximum Number of Monitored Links in Relation to ITT and Control-Network Data Rate

ITT [ms]	Control Network Data Rate [Mbit/s]		
	10	100	1000
1	7	74	744
10	74	744	7440
25	186	1860	18601
100	744	7440	74404

### 1.3.9 Beispielcode: Gedrehte Tabelle

Manche Tabellen sind zu groß um sie hochkant darzustellen, das folgende Beispiel nutzt mehrere Packages um eine querformatige Tabellen darzustellen.

Name	Developer	License / Code	Built-in Capabilities	APIs	Language	OpenFlow Version	Comments
<b>Beacon</b>	David Erickson, Rob Sherwood	GPL v2 <sup>4</sup> / Free	Topology, Routing, WebUI	OpenFlow, OSGi	Java	1.0	Platform agnostic, deprecated, OpenFlow only
<b>Floodlight</b>	Project Floodlight / Big Switch Networks Inc	Apache 2.0 / Free	Topology, Routing, WebUI	OpenFlow, REST, CLI	Java	1.0, 1.3 <sup>5</sup>	Platform: Linux and Mac OS X, forked from Beacon
<b>Helium</b>	Open Daylight Project	EPL 1.0 / Free	Topology, Routing, WebUI	OpenFlow, REST, OVSDB, Netconf, SNMP, OSGi, ...	Java	1.0, 1.3	General SDN controller, forked from Beacon
<b>NOX</b>	Nicira / NOXrepo Community	GPL v3 <sup>6</sup> / Free	Topology, Routing, GUI	OpenFlow	C++	1.0	Platform: Linux, deprecated, OpenFlow only, first OpenFlow Controller
<b>OpenIRIS</b>	ETRI / OpenIris Community <sup>7</sup>	Apache 2.0 / Free	Topology, Routing, WebUI, Fast Failover, Multiple Controllers, Firewall	OpenFlow, REST	Java	1.0, 1.3 <sup>40</sup>	Floodlight compatible northbound API, forked from Floodlight
<b>OpenMUL</b>	OpenMUL Community <sup>8</sup>	GPL v2 / Free	Topology, Routing, NAT, Controller Failover	OpenFlow, REST, CLI, Netconf, OVSDB	C / Python	1.0, 1.3-4	Hierarchical distributed controller aimed at carrier networks
<b>POX</b>	NOXrepo Community	Apache 2.0 / Free	Topology, Routing, WebUI, GUI	OpenFlow, OVSDB (experimental)	Python	1.0	Python port of NOX, designed for Ease of Use and Research Purposes
<b>RunOS</b>	ARCCN	Apache 2.0 / Free <sup>9</sup>	Topology, Routing	OpenFlow, REST	C / C++	1.0	Experimental Controller deployed as Linux Kernel Module
<b>Ryu</b>	Open Source Software Computing Group	Apache 2.0 / Free	Topology, Routing, WebUI, Firewall, Switch Monitoring, OpenStack Adapter	OpenFlow, REST, Netconf, OVSDB	Python	1.3-4	Well documented code, large packet parser suite

<sup>4</sup> initially released under BSD, final version released under GPL v2 <http://www.gnu.org/licenses/gpl-2.0.en.html>

<sup>5</sup> OpenFlow 1.0-1.4 support via Loxigen <http://github.com/floodlight/loxigen>

<sup>6</sup> <http://www.gnu.org/licenses/gpl-3.0.en.html>

<sup>7</sup> <http://openiris.etri.re.kr>

<sup>8</sup> <http://www.openmul.org>

<sup>9</sup> Kernel Module not released

## 2 Einführung

## **3 Kapitel 1**

### **3.1 Kurzfassung**

### **3.2 Abschnitt 1**

#### **3.2.1 Unterabschnitt 1**

## **4 Kapitel 2**

### **4.1 Kurzfassung**

### **4.2 Abschnitt 1**

#### **4.2.1 Unterabschnitt 1**



## **5 Kapitel 3**

### **5.1 Kurzfassung**

### **5.2 Abschnitt 1**

#### **5.2.1 Unterabschnitt 1**

## **6 Kapitel 4**

### **6.1 Kurzfassung**

### **6.2 Abschnitt 1**

#### **6.2.1 Unterabschnitt 1**

## **7 Kapitel 5**

### **7.1 Kurzfassung**

### **7.2 Abschnitt 1**

#### **7.2.1 Unterabschnitt 1**

## **8 Kapitel 6**

### **8.1 Kurzfassung**

### **8.2 Abschnitt 1**

#### **8.2.1 Unterabschnitt 1**

## 9 Fazit

# A Abkürzungsverzeichnis

**DPI** Dots per Inch

**IDE** Integrated Development Environment

# Literaturverzeichnis

[1] JabRef Development Team, *JabRef*, 2016. [Online]. Available: <http://www.jabref.org>

## Abbildungsverzeichnis

1.1	Texmaker Konfiguration Seite 1: Build Ordner einstellen . . . . .	2
1.2	Texmaker Konfiguration Seite 2: Schnelles Übersetzen . . . . .	3



## Tabellenverzeichnis

1.1	Die bekanntesten Tex Distributionen . . . . .	1
1.2	Die wichtigsten Fancyref Präfixe . . . . .	4
1.3	Zusätzliche Einheiten für siunitx . . . . .	5
1.4	Erklärung für den Code zum Einbinden von Grafiken . . . . .	7
1.5	Maximum Number of Monitored Links in Relation to ITT and Control-Network Data Rate . . . . .	8