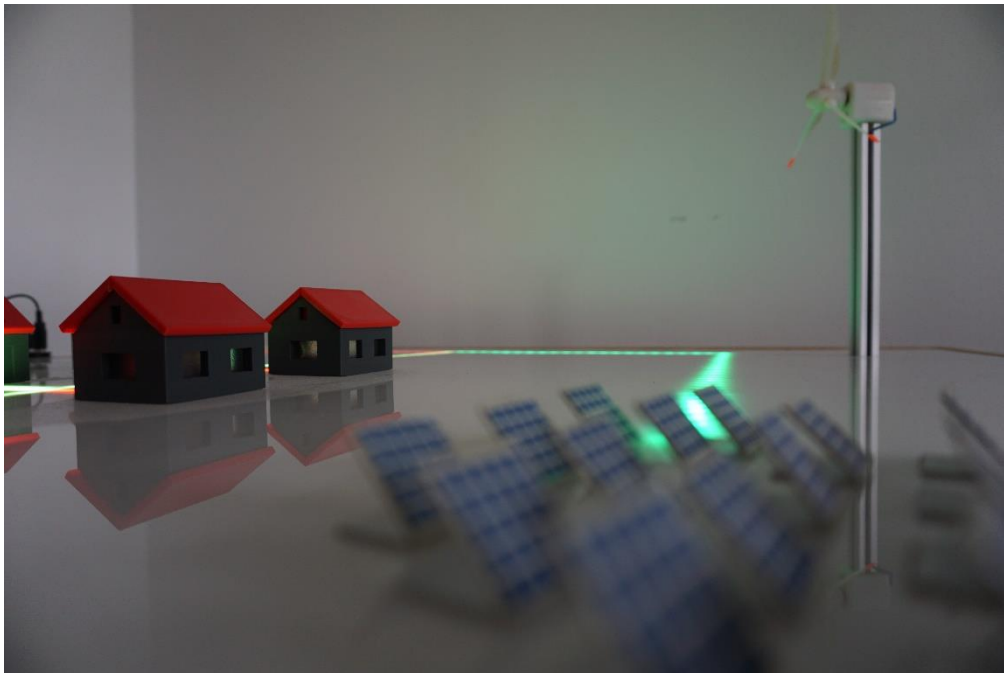


Dokumentation zu dem Modul „Automatisierung und Leittechnik“ mit dem Projekt „Erneute Inbetriebnahme des Smart Grid Modells“



Name, Vorname: Thiel, Lena

Matrikelnummer: 013200420

Name, Vorname: von Kuczkowski, Nicolas

Matrikelnummer: 013200193

Abgabedatum: 18.09.2019

Datum, Unterschrift

Datum, Unterschrift

Inhalt

Abkürzungsverzeichnis	3
Einleitung	4
Smart Grid Modell	4
Windrad	4
Häuser	5
Solaranlage.....	5
Kraftwerk	5
Bestandsaufnahme.....	6
Funktionsprüfung	7
Software	7
Hardware.....	8
Betriebsanleitung	9
Bedienung des Smart Grid Modells	9
Demontage des Smart Grid Modells.....	11
Montage des Smart Grid Modells.....	11
Datenaustausch durch OPC Server Client Cloud Architektur	12
Prototypische Implementierung.....	13
Zusammenfassung und Ausblick	16

Abkürzungsverzeichnis

SGM	Smart Grid Modell
HDMI	High Definition Multimedia Interface
GUI	Graphics User Interface
OPC	Open Platform Communication
APP	Anwendungssoftware
URL	Uniform Resource Locator

Einleitung

Die vorliegende Dokumentation beschreibt das Vorgehen zur Inbetriebnahme des „Smart Grid Modells“, das sich, zum Zeitpunkt des Sommersemesters 2019, in Projektraum Automatisierung 1.08 am Campus Velbert/Heiligenhaus der Hochschule Bochum befindet. Das Ziel ist die erfolgreiche Inbetriebnahme des Modells sowie die Wiederherstellung seiner Funktionalität. Des Weiteren soll erreicht werden, dass Daten aus dem Modell über einen OPC Server an eine Cloud übergeben werden können. Dabei ist das Aufsetzen des OPC Servers und der Cloud sowie die Schnittstellendefinition zu dem Modell Aufgabenstellung eines anderen Projekts. Die vorliegende Arbeit liefert das Konzept für die OPC Server-Client-Cloud Architektur.

Smart Grid Modell

Das „Smart Grid Modell“, folgend SGM, wurde als Prüfungsleistung in dem Modul „Vertiefung Mechatronik“ in dem Bachelorstudiengang „KIS Mechatronik & Informationstechnologie“ entwickelt im Wintersemester 2017/2018 fertig gestellt.

Das SGM ist ein Modell einer Stadt, welches Strom aus konventionellen Kraftwerken verwendet, wenn der Strom aus den erneuerbaren Energien nicht ausreicht, um die Stadt zu versorgen.

Die folgenden Abschnitte erläutern die einzelnen Komponenten des SGMs. Im Rahmen der vorliegenden Arbeit wurden das Projektteam kontaktiert, dass im Wintersemester 2017/2018 das SMG fertig gestellt hat. Grund hierfür ist, dass am Campus Velbert/ Heiligenhaus keine Dokumentation zu dem SMG vorliegt. Das damalige Projektteam hatte seiner Zeit angefangen eine Dokumentation zu schreiben, diese aber nicht fertig gestellt. Teile der Dokumentation aus dem Wintersemester 2017/2018 sind in die vorliegende Dokumentation integriert. Die Stellen, an denen diese Dokumentation integriert ist, sind kenntlich dargestellt, da Sie nicht geistiges Eigentum des Projektteams sind, welches die vorliegende Dokumentation verfasst.

Windrad

„Das Windrad ist in diesem Sinne kein richtiges Windrad, da es von einem Motor angetrieben wird und somit keine Energie produziert. Es soll lediglich als Anschauungsobjekt dienen und die aus Windkraft erzeugte Energie simulieren.



Abbildung 1: Windrad CAD
(Schrinkel, et al., 2017)

Da es schwer war ein geeignetes motorgesteuertes Windrad für diesen Anwendungsfall zu finden, entschied man sich dazu ein individuelles Windrad zu konstruieren. Dazu wurde die CAD-Software CimatronE verwendet. Das Design des Windrads ist ein Unikat und gibt es so kein zweites Mal. Vor Beginn der Konstruktionsarbeiten wurde ein geeigneter kleiner DC-Motor gesucht, welcher dann ausschlaggebend für den Aufbau des Windrades war. Das Datenblatt des Motors ist im Anhang zu finden. Das Besondere an dem Windrad ist, dass jede einzelne Komponente (Gehäuse, Nabe und Rotorblätter) mit einem 3D Drucker gedruckt wurden. Auch das musste bei der Gestaltung des Windrads berücksichtigt werden. [...]. Als Standfuß dient ein 20x20 Profil.“ (Schrinkel, et al., 2017) Der DC-Motor kann stufenlos in verschiedenen Geschwindigkeiten angetrieben werden.

Häuser

„Da es sich um ein mechatronisches Projekt handelt, wurde sich überlegt die modellhaften Häuser selber zu konzipieren, sodass diese genau auf den Anwendungsfall abgestimmt sind. Auch hier wurde auf das 3D-Druckverfahren zurückgegriffen und die Häuser mit CimatronE entworfen. Bei der Konstruktion wurde darauf geachtet, dass das Drucken möglichst wenig Stützmaterial verbraucht, um Kosten zu sparen. Daher sind die im Anhang beigefügten STL-Daten nicht ein komplettes Haus in einem, sondern aufgeteilt in das Grundgestell und zwei einzelne Dachhälften, welche später mit Zweikomponentenkleber zusammengeklebt werden müssen. So ist das Dach auch abnehmbar, um an die Schrauben zur Befestigung mit der Plexiglasscheibe zu gelangen.“ (Schrinkel, et al., 2017)

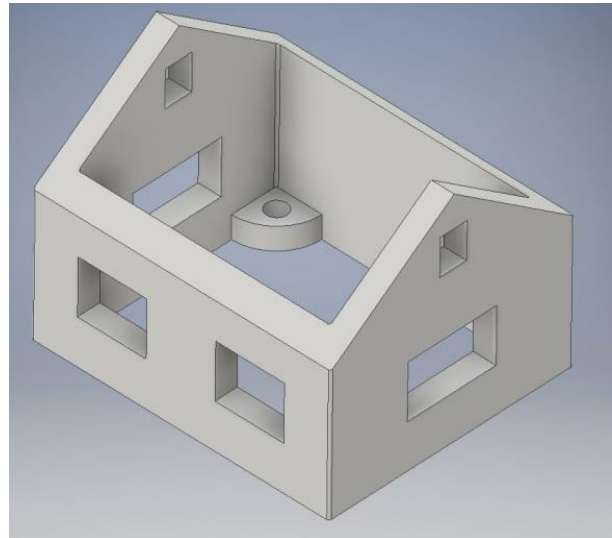


Abbildung 2: Grundgestell Haus (Schrinkel, et al., 2017)

Solaranlage

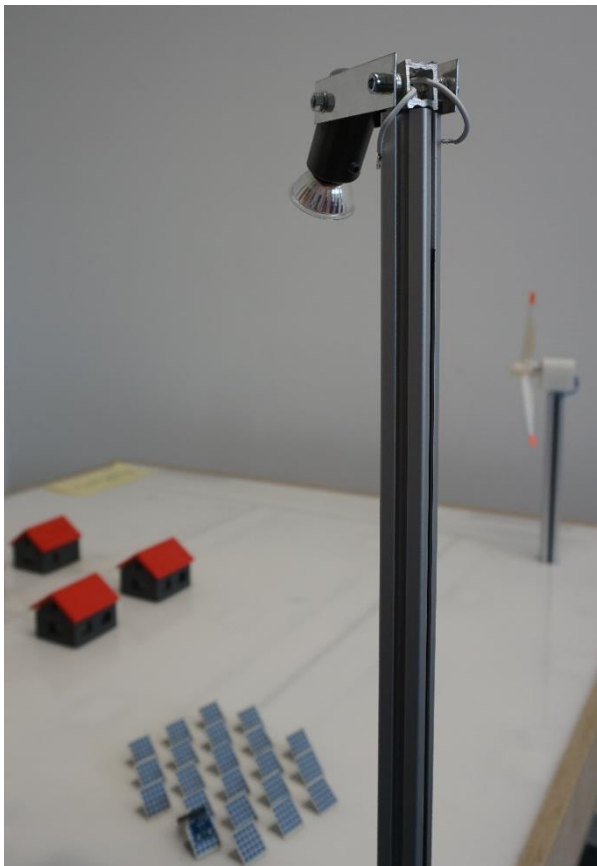


Abbildung 3: Halogenlampe des Solaranlagenmodells auf dem SMG

Das Model der Solaranlage besteht aus zweiundzwanzig metallenen Winkeln, welche Solarpaneele darstellen sollen. An einem dieser Winkel ist ein Helligkeitssensor (BH1750, Datenblatt befindet sich im Anhang) angebracht. Dieser ist über sein I²C-Interface (SDA und SLC) mit sowohl mit dem Arduino als auch mit dem Raspberry Pi verbunden sowie mit 3,3V Spannung versorgt.

Desweiteren ist an SGM eine Halogenlampe auf einem 20x20 Profil montiert, jedoch nicht angeschlossen. Diese kann durch eine Minimierung der Spannungsversorgung gedimmt werden.

Mit der Halogenlampe und dem Sensor kann in einem Zukünftigen Projekt die Simulation der Fotovoltaikanlage zusätzlich zur Software auf dem SMG widergegeben werden.

Kraftwerk

Um die konventionelle Energieerzeugung auf dem SGM darzustellen wurde ein Model gebastelt welches zukünftig gegen einen 3D-Druck ausgetauscht werden könnte.

LEDs

Auf dem SGM wird die Verbrauchsverteilung mittels LEDs dargestellt. Bei diesen LEDs handelt es sich um RGB-LEDs, damit diese so wohl grün als auch rot leuchten können, da bei symbolisieren die rotleuchtenden LEDs die Abdeckung der Verbraucher durch herkömmliche Stromerzeugung und die grünleuchtenden die Abdeckung über die erneuerbaren Energien.

Daher sind die LED-Leisten, die von der Solaranlage und dem Windrad kommen immer entweder grün oder aus und die Leiste, die vom Kraftwerk kommt, entweder rot oder ausgeschaltet. Die gemeinsame Zuleitung zu den Häusern hat drei mögliche Zustände rot, grün oder gelb.

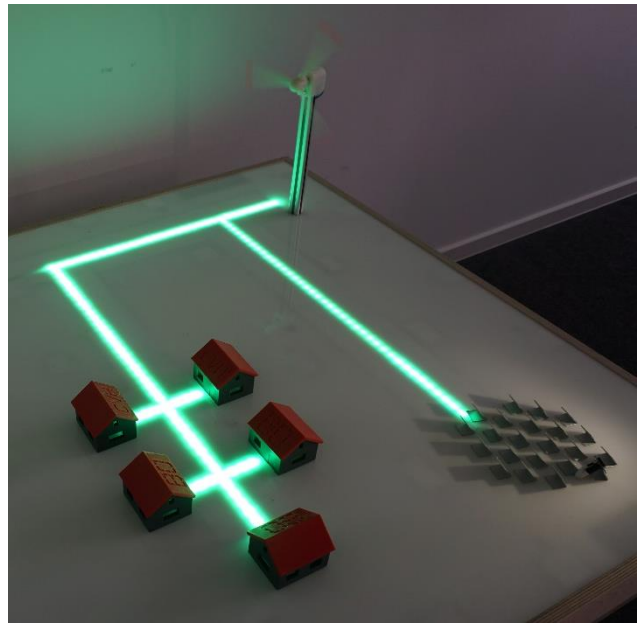
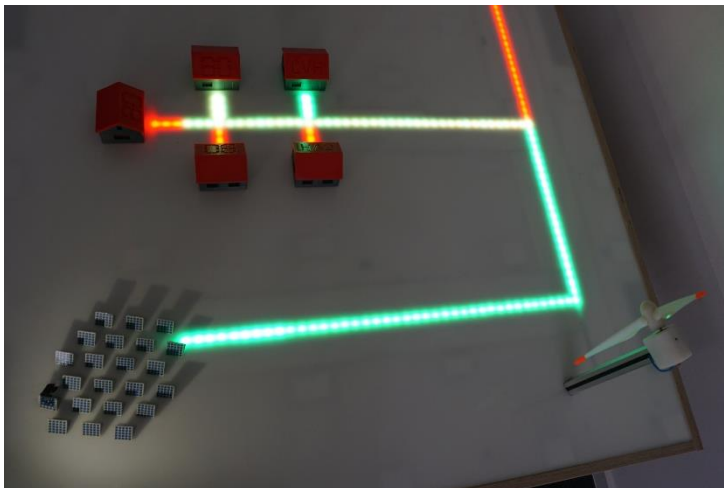


Abbildung 4: Verbraucherabdeckung vollständig über erneuerbare Energien



Die jeweiligen Zuleitungen der einzelnen Häuser leuchten entweder grün oder rot. Hier bei hängt die Anzahl der jeweiligen Farbe vom Grad der Verbraucherabdeckung durch die jeweilige Art der erzeugten Energie ab.

Abbildung 5:
Verbraucherabdeckung durch
Solar- und konventionelle Energie

Bestandsaufnahme

Zu Beginn des Sommersemesters 2019 führte das Projektteam eine Bestandsaufnahme durch, um zu evaluieren, welche Arbeitspakete abzuarbeiten sind, damit das SGM wieder funktionsfähig ist. Die Bestandsaufnahme enthielt eine Funktionsprüfung, sowie eine Sichtung der Software und Hardware.

Funktionsprüfung

Die Funktionsprüfung begann mit dem Anschluss aller Stecker in Steckdosen, damit das SGM mit Spannung versorgt wird. Des Weiteren wurde ein Bildschirm an den Raspberry Pi via HDMI angeschlossen, um die grafische Benutzeroberfläche, folgend GUI, des Raspberry's zu sehen. Da die Dokumentation aus dem Wintersemester 2017 nicht komplettvorhanden war, musste das Programm zum Betrieb des SMGs gesucht werden. Nachdem es gefunden wurde, wurde eine Desktop-Verknüpfung zu dem Speicherort des Programms erstellt, damit von nun an das Programm schnell zu erreichen ist.

Anschließend wurden verschiedene Einstellungen in der GUI des SGMs vorgenommen. Das Ergebnis war, dass sich keine LEDs durch Veränderungen der Parameter im GUI einschalten ließen. Somit konnte die Aussage von Prof Dr. C. Faller bestätigt werden, dass das SGM derzeit keine Funktion aufweist.

Als ein weiteres Problem stellte sich heraus, dass die USB-Anschlüsse des Raspberry Pis nur erreichbar sind, wenn das SGM teilweise demontiert ist. Hierfür erfolgt im Sommersemester 2019 die Montage eines USB-Armaturenbrettpkabels, damit die USB-Anschlüsse im Zusammenbauzustand des SGMs erreichbar sind.



Abbildung 6: Anschlüsse des SGM (zweimal USB, Netzwerk, HDMI und Stromanschluss)

Software

Wie im vorherigen Abschnitt erwähnt, befindet sich zum Abgabezeitpunkt der vorliegenden Arbeit auf dem Desktop des Raspberry Pis eine Verknüpfung zum Speicherort der verschiedenen .c und .h Dateien des SGMs. Folgend ist der grundsätzliche Aufbau der Softwarearchitektur beschrieben:

„Der gesamte Programmcode für das SmartGrid ist in C geschrieben. Dabei wurde das Programm nach dem Prinzip der Kooperativen Echtzeitfähigkeit aufgebaut. Die visuelle Darstellung der Stromverteilung wird im Echtzeit-Task alle 2 Sekunden aktualisiert. Die restliche Zeit wird für Berechnungen und die Bearbeitung von Userinput in der GUI verwendet. Dabei wurde darauf geachtet, dass alle Funktionen die Durchführung nicht unterbrechen können und somit die Echtzeitfähigkeit nicht negativ beeinflussen.

Da der Pi nicht genügend IOs besitzt, um das gesamte SmartGrid ansteuern zu können, wurde dieser Teil auf einem Arduino Mega realisiert, der über I2C mit dem PI verbunden ist. Es gibt zwei Modi, mit denen das SmartGrid betrieben werden kann. Zum einen gibt es einen Spielmodus, in dem sämtliche Parameter über die GUI veränderbar sind. Zum anderen wurde zusätzlich ein Simulationsmodus erstellt, in dem die Tageszeit, die Windstärke und der Bewölkungsgrad fest vorgegeben werden.“ (Schrinkel, et al., 2017)

Hardware

Um zu verstehen, warum das SGM keine Funktion aufweist, wurde das SGM erst einmal demontiert, um eine Sichtprüfung der verbauten Komponenten durchzuführen. Parallel erfolgte die Durchsicht des Stromlaufplans, um die Hardwarearchitektur zu verstehen. Bei der Demontage des SGMs, als Teil der Bestandsaufnahme, und einer Analyse des Stromlaufplans wurde festgestellt, dass sich die Lötverbindung der Spannungsversorgung zur Hauptplatine gelöst hat (rot umrandet im folgenden Bild). Um sicher zu gehen, dass damit auch die Versorgung der LED-Ansteuerung nicht mehr gegeben war, wurde zu dem noch eine Durchgangsprüfung mit Hilfe eines Multimeter gemacht. Diese Prüfung wurde auch noch an weiteren Stellen durchgeführt um eventuelle kalte Lötstellen zu finden, welche jedoch nicht vorhanden waren.

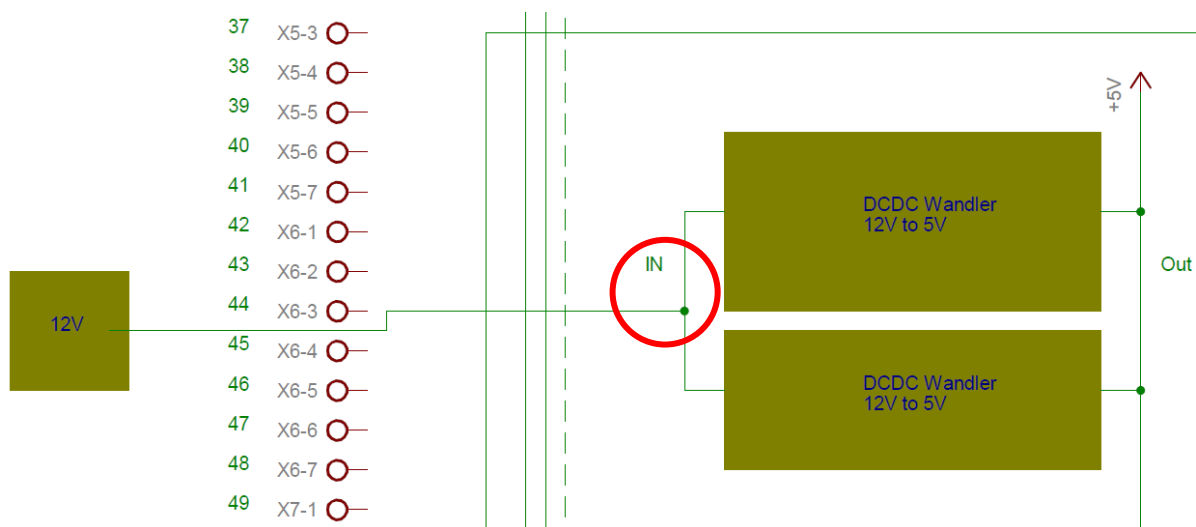


Abbildung 7: Ausschnitt aus dem Stromlaufplan - Schnittstelle der Spannungsversorgung des SGMs (Schrinkel, et al., 2017)

Dies ist der Grund, warum das SGM keine Funktion aufweist. Obwohl der Raspberry Pi mit Spannung versorgt ist, stehen dem SGM keine 5 V für die LEDs zur Verfügung. Eine Schwierigkeit bei der Behebung dieses Problems ist die Bauteilbeschriftung im SGM bzw. die Beschriftung der Bauteile im Stromlaufplan. Bei den Beschriftungen ist nicht für alle Bauteile eine konsistente Beschriftung vorhanden. Nach der Wiederherstellung der Lötverbindung und einer erneuten Funktionsprüfung zeigt sich, dass die Funktion des SGMs zu großen Teilen wiederhergestellt ist. Um die Lötverbindung zu erneuern war eine komplette Demontage des SGMs und die Entnahme der Platine, welche die Spannungsversorgung regelt, nötig.

Betriebsanleitung

Dieser Abschnitt setzt sich aus den Unterabschnitten „Bedienung des SmartGrids“ sowie „Demontage des SGMs“ und „Montage des SGMs“ zusammen. In „Bedienung des SmartGrids“ ist eine detaillierte Beschreibung zur Nutzung des GUIs enthalten. In den Unterabschnitten „Demontage des SGMs“ und „Montage des SGMs“ sind die Reihenfolgen der notwendigen Arbeitsschritte zur De- und Montage des SGMs erläutert.

Bedienung des Smart Grid Modells

Um das Modell zu starten ist es notwendig, dass zuerst ein Monitor und eine Maus an den Raspberry Pi angeschlossen sind. Desweiteren muss dieser an mit einem Netzwerk verbunden sein, beispielsweise über ein Netzkabel verbunden mit einem Laptop, welcher sich im Hochschulnetzwerk befindet. Dann können die Stromversorgung des SMG und des Rasperrys angeschaltet werden.

Nach dem der Raspberry Pi hochgefahren ist, kann über die Verknüpfung auf dem Desktop der Ordner „Smart Grid“ geöffnet werden. Wenn das Programm „Smart“ gestartet wird, erscheint eine Abfrage bezüglich der Ausführungsoption, hier muss die Option „ausführen“ ausgewählt werden um die GUI zu starten.

„Die Benutzeroberfläche für das SmartGrid kann über einen am Raspberry PI angeschlossenen Monitor angezeigt werden. Zur Bedienung kann eine Funkmaus verwendet werden.“



Abbildung 8: Einstellungsfenster SmartGrid (Schrinkel, et al., 2017)

Über die Slider kann die Einwohnerzahl, der Industrieanteil, die Anzahl der Windräder sowie die Gesamtfläche der Solarpanels eingestellt werden.

Der/die Bediener/-in hat unten rechts die Möglichkeit zwischen „Spiel-„ und „Simulationsmodus“ auswählen (in Abbildung 4 rot umrahmt).

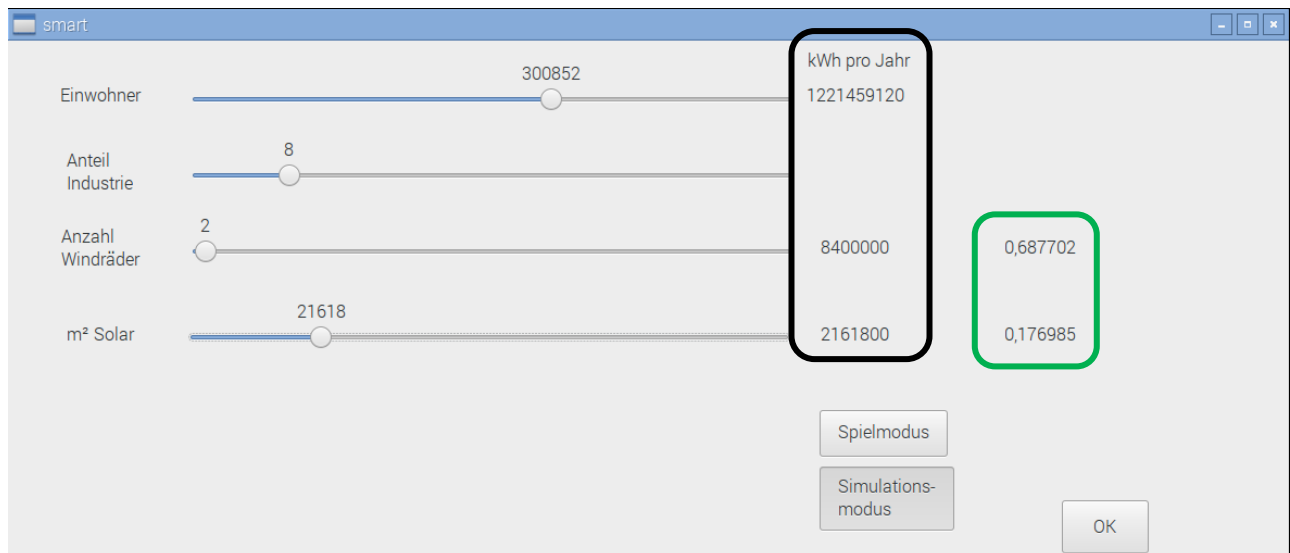


Abbildung 9: Einstellungsfenster mit Werten (Schrinkel, et al., 2017)

Rechts neben den Slidern werden der Jahresverbrauch bzw. Jahresertrag der Windräder und Solarpanels angezeigt (schwarz umrahmt).

Zusätzlich werden die prozentualen Anteile des Ökostroms am Jahresverbrauch angezeigt (grün).

Nach Einstellung der Werte gelangt man durch betätigen der Ok-Taste zum Hauptfenster, dass in Abbildung 6 dargestellt ist.

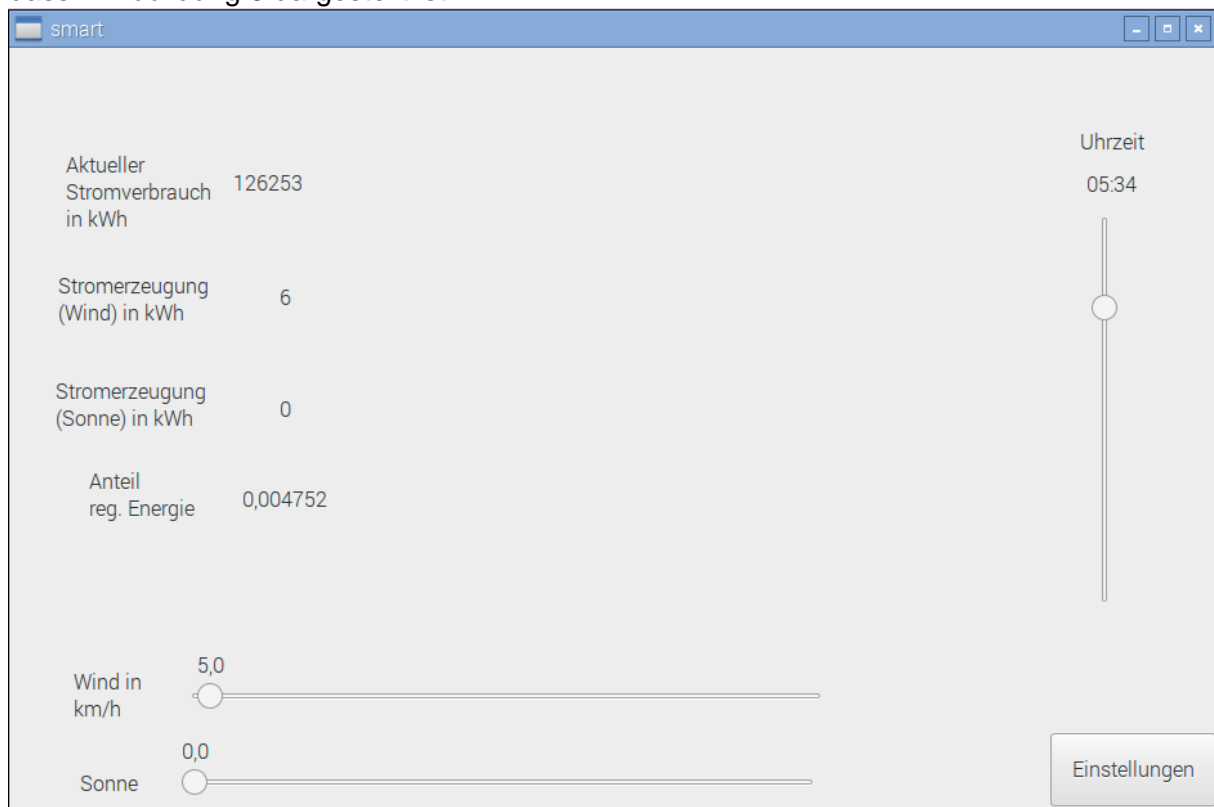


Abbildung 10: Hauptfenster Smart Grid (Schrinkel, et al., 2017)

Wenn der Simulationsmodus gewählt wurde, werden die Slider für Windgeschwindigkeit, Sonneneinstrahlung und Uhrzeit automatisch durch das Programm angepasst. Im Spielmodus kann der Benutzer diese manuell betätigen.

Über einen Klick auf den Button „Einstellungen“ gelangt man zurück zum Einstellungsfenster. Dabei wird die Simulation pausiert.“ (Schrinkel, et al., 2017) In diesem Modus kann es vorkommen, dass sich die Simulation aufhängt, die kann entweder durch eine Beendigung des Programms oder einen Neustart des SMGs beendet werden.

Allgemein ist zu der GUI noch zu sagen, dass sich mit der Uhrzeit auch der Verbrauch der Häuser ändert. Dies soll ein möglich genau Abbildung von realen Haushalten widerspiegeln. Desweiteren ist noch zu beachten, dass ein kleiner Delay von ca. 2 Sekunden zwischen Werteänderung in der GUI und der Reaktion des SGM vorliegt.

Demontage des Smart Grid Modells

Folgend sind die Arbeitsschritte beschrieben, die notwendig sind, um das SMG zu demontieren:

1. „Stromversorgung abschalten!
2. Die Schrauben für das Windrad und der Zwischenplatte auf der Unterseite lösen. Anschließend das Windrad herausnehmen. Achtung! Beim Herausziehen die Steckverbindung lösen!
3. Die Steckverbindung am Lichtsensor lösen.
4. Plexiglasplatte abnehmen.
5. Auf der Unterseite die Zwischenplatte losschrauben.
6. Die Zwischenplatte anheben. Stromversorgung der Steuerplatte und Verbindung zwischen Raspberry PI und Arduino trennen. Anschließend kann die Zwischenplatte abgenommen werden.“
(Schrinkel, et al., 2017)

Montage des Smart Grid Modells

Folgend sind die Arbeitsschritte beschrieben, die notwendig sind, um das SMG zu montieren:

1. „Die Zwischenplatte in korrekter Orientierung oberhalb des Kastens positionieren.
2. Die Stromversorgung der Steuerplatte und Verbindung zwischen Raspberry PI und Arduino anschließen.
3. Die Steckverbindungen des Lichtsensors und Windrads durch die jeweiligen Bohrungen nach Außen führen
4. Zwischenplatte einlegen
5. Plexiglasplatte einlegen
6. Den Lichtsensor anschließen.
7. Das Windrad anschließen. Anschließend die Zwischenplatte und das Windrad auf der Unterseite festschrauben.“
(Schrinkel, et al., 2017)

Datenaustausch durch OPC Server Client Cloud Architektur

Eine Anforderung von Prof. Dr. C. Faller ist, dass Daten vom SGM über einen Open Plattform Communication Server, folgend OPC, in eine Cloud speicherbar sind. Wie eingangs erläutert, ist das Aufsetzen des OPC Servers die Aufgabe eines anderen Projektteams. Die vorliegende Arbeit liefert ihren Teil zur Erfüllung dieser Anforderung indem sie ein Konzept für die OPC Server-Client-Cloud Architektur erläutert.

Um das grundsätzliche Konzept bzw. die Architektur von SGM/ OPC Server/ Cloud festzulegen, hat das Projektteam des Sommersemesters 2019 ein Konzept entwickelt, dass die Anforderungen von Prof. Dr. C. Faller erfüllt. Das Konzept ist folgend in Abbildung 7 dargestellt.

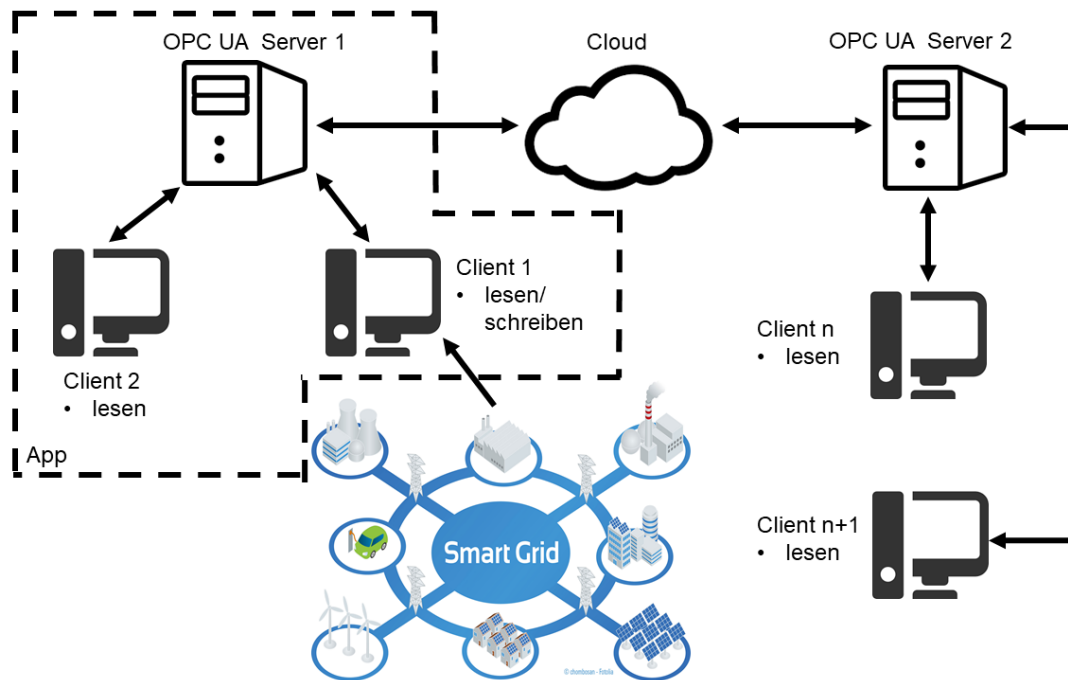


Abbildung 11: Konzept für die OPC Server-Client-Cloud Architektur nach (Pethig, 2017)

Das Konzept sieht vor, dass die Daten des SGMs (Client 1) an den OPC Server übertragen werden. Bei den Daten handelt es sich um die Energien, die durch das Windrad, die Solarzellen und durch das Kraftwerk einstellbar sind. Aus Abbildung 7 ist zu entnehmen, dass der Client 1 Lese- und Schreibrechte auf den OPC Server 1 hat. Dies ist notwendig, damit Client 1 seine Daten übertragen kann. Über den OPC Server 1 werden die Daten anschließend Client 2 zur Verfügung gestellt. Die Rechte dieses Clients sind auf das Lesen beschränkt, damit die ursprünglichen Daten des SGMs nicht von Client 2 überschreibbar sind.

Das Konzept sieht weiterhin vor, dass der OPC Server eine Cloudanbindung bekommt, um eine weitere Anforderung von Prof. Dr. C. Faller zu erfüllen. In der Cloud werden die Daten gespeichert, in diesem Fall Energie aus Wind- und Sonne sowie fossilen Brennstoffen, die im Betrieb des SGMs entstehen. Mit der Cloud als zentrale Schnittstelle kann diese Architektur um OPC Server 2 erweitert werden. Wiederum an diesen Server können weitere Clients (n und n+1) angebunden werden. Somit sind die Daten des SGMs auch von den Clients n und n+1 abrufbar. Für den Fall einer Koppelung des SGMs mit einem anderen Modell oder Gerät ist es auch möglich, dass die Clients n und n+1 die Aufgabe von Client 1 übernehmen und der Cloud die Daten von dem gekoppelten Modell oder Gerät zur Verfügung stellen. In diesem Anwendungsfall müssen die Zugriffsrechte angepasst werden.

Prototypische Implementierung

Der vorliegende Abschnitt zeigt die prototypische Implementierung des Konzepts aus Abbildung 7 aus dem vorherigen Kapitel. Das eigentliche Ziel, die Implementierung des Konzepts aus Abbildung 7 zu erreichen, konnte nicht erreicht werden. Um das Teilziel „Konzeptimplementierung“ zu erreichen, erfolgt die prototypische Implementierung des Konzepts aus Abbildung 7. In dem Konzept aus Abbildung 7 sind der OPC Server 1 und die Clients 1 und 2 von einer gestrichelten Linie umgeben. Der von der gestrichelten Linie umschlossene Bereich ist in einer App prototypisch implementiert. Die Entwicklung der App erfolgte in Matlab/Simulink.

Die App besteht aus einem GUI, das drei verschiedene Tabulatoren enthält. Der erste Tabulator (OPC Server 1) stellt die Standardansicht dar, da der Tabulator bei der Ausführung der App als erste Ansicht erscheint. Der Tabulator besteht aus einem Textfeld und drei Checkboxes. Beim Eintragen des Uniform Resource Locators, folgend URL, wird die angegebene URL an die URL Textfelder der Tabulatoren Client 1 und Client 2 übergeben. Die Checkboxes stellen die Freigabe der SGM Daten auf dem Serverlevel dar. Daher müssen alle Checkboxes gecheckt sein, damit die Daten des Clients 1 über den Server an Client 2 übertragen werden können.

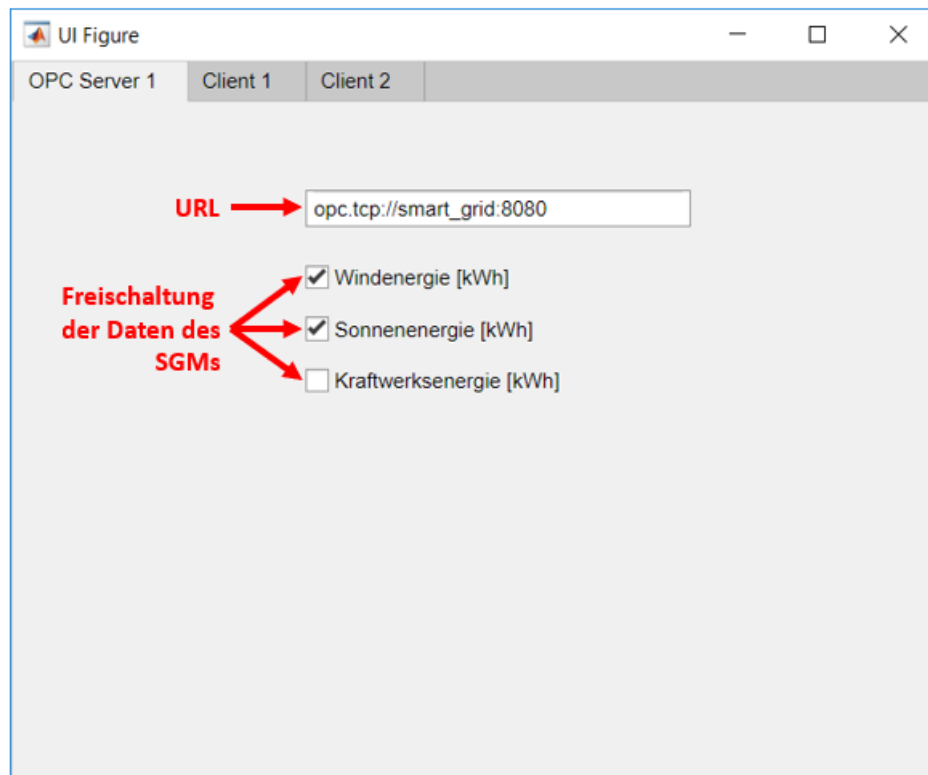


Abbildung 12: SGM App - Ansicht des OPC Server Tabulators

Für den Fall das die Checkboxes nicht mehr gecheckt sind, ist das Prinzip hold-last-value implementiert. Somit zeigt Client 2 die letzten Daten an, die Client 1 geschickt hat, wenn die Checkboxes aus Tabulator OPC Server 1 nicht mehr gecheckt sind.

Aktuell sieht das Konzept vor, dass Client 1 (SGM) die aktuellen Betriebsdaten Wind-, Sonnen- und Kraftwerksenergie in Kilowattstunden an den OPC Server übermittelt. Die URL aus dem Textfeld aus Abbildung 9 ist die URL des OPC Servers und stellt somit das Ziel dar, an den der Client 1 seine Daten übertragen soll. Um dies offline simulieren zu können, enthält der Tabulator Client 1 sogenannte Spinner. Dies sind Elemente, die es ermöglichen Zahlen

einzugeben, zu erhöhen und zu verringern. In Abbildung 8 ist zu sehen, dass die Checkbox der Kraftwerksenergie nicht gecheckt ist. Dies hat zur Folge, dass es nicht möglich ist, in dem Tabulator Client 1 aus Abbildung 9 eine Zahl für die Kraftwerksenergie einzugeben. Dies bestätigt die simulierte Wirksamkeit der Datenfreigabe auf dem Serverlevel. Die Checkbox der Kraftwerksenergie ist nicht gecheckt und somit ist es Client 1 nicht möglich Daten des SGMs zu übertragen.

Grundsätzlich besteht die Möglichkeit weitere Daten des SGMs dem OPC Server zur Verfügung zu stellen. Da die vorliegende App eine mögliche Implementierung darstellt, erfolgte die Beschränkung auf die verwendeten Energien aus Wind, Sonne und Kraftwerk.

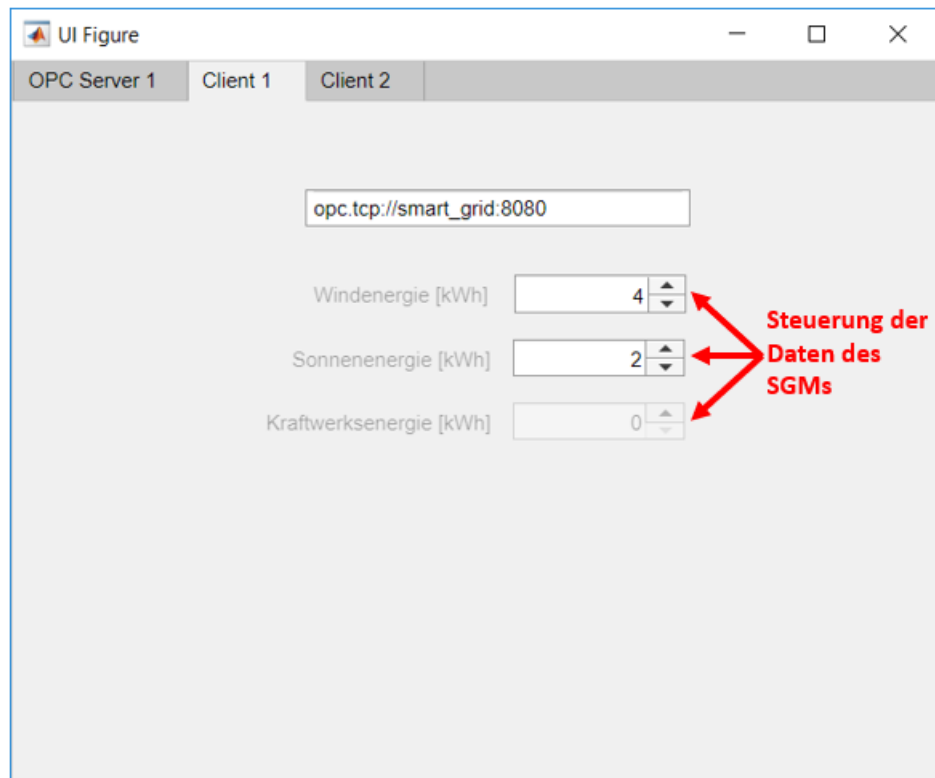


Abbildung 13: SMG App - Ansicht des 1. Client Tabulators

Der Tabulator Client 2 aus Abbildung 10 stellt den Client mit Lesezugriff dar. Auch er hat die URL des OPC Servers übergeben bekommen. Da dieser Client nur Lesezugriff auf die Daten des SGMs hat, ist es nicht möglich die Daten zu verändern. Aus diesem Grund enthält Client 2 nur Felder die Zahlen anzeigen können. Anders als der Tabulator von Client 1 enthält

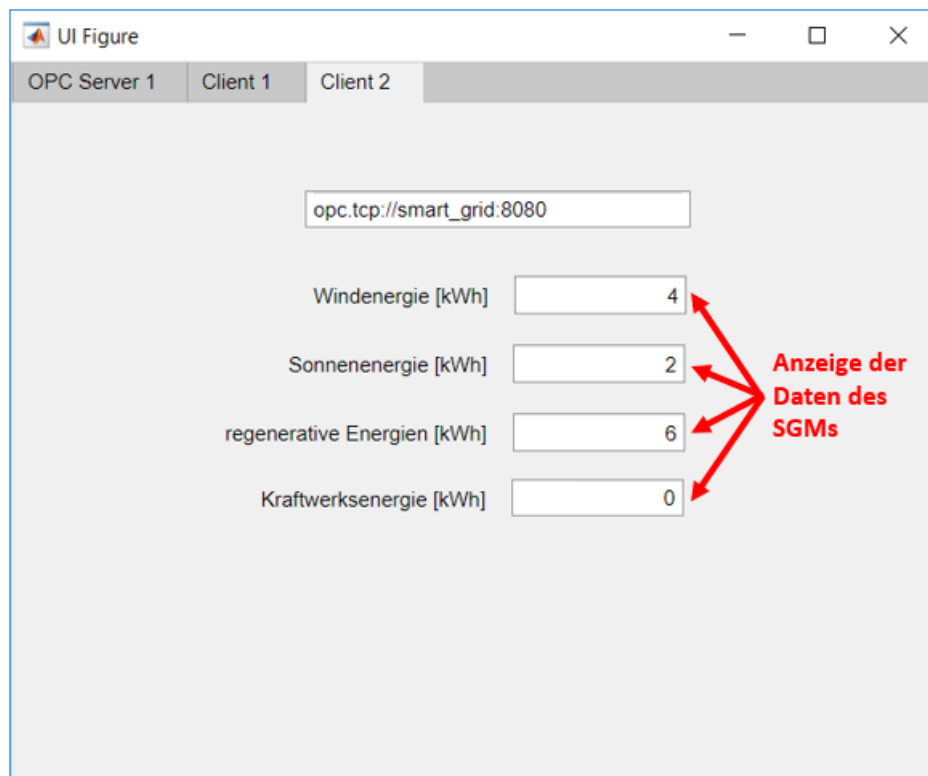


Abbildung 14: SMG App - Ansicht des 1. Client Tabulators
der Tabulator von Client 2 das Feld „regenerative Energien“ dieses Feld zeigt die Summe aus den Feldern Wind- und Sonnenenergie an und dient zu Anzeigezwecken. Weiterhin besteht die Möglichkeit, dass der Tabulator von Client 2 durch Anzeigeinstrumente erweitert wird. Da dies ein rein visueller Effekt ist, wurde an dieser Stelle darauf verzichtet.

Die ausführbare App im mlapp Format befindet sich bei der Abgabe der vorliegenden Arbeit auf dem Raspberry Pi des SGMs.

Zusammenfassung und Ausblick

Die vorliegende Arbeit dokumentiert die am SGM durchgeführten Arbeiten aus dem Wintersemester 2017/2018 und dem Sommersemester 2019. Die beiden Hauptziele sind zum einen die Wiederherstellung der Funktion des SGMs sowie die ordnungsgemäße Dokumentation der bereits erfolgten Arbeiten. Diese beiden Ziele wurden erfolgreich erfüllt. Des Weiteren wurden im Sommersemester 2019 die elektrischen Raspberry Pi Schnittstellen mittels eines Adapters von außerhalb des SGMs zugänglich gemacht. Somit ist für einen Raspberry Pi Zugriff die Demontage des SGMs nicht mehr notwendig.

Die Transferleistung der vorliegenden Arbeit stellt die erfolgreiche Erarbeitung eines Konzepts für eine OPC Server-Client-Cloud Architektur dar. Des Weiteren ist das Konzept prototypisch in einer App in Matlab implementiert, um Möglichkeiten dieser Architektur offline visualisieren zu können.

Für die Umsetzung des erarbeiteten Konzepts für eine OPC Server-Client-Cloud Architektur, gilt es die Architektur bestehend aus einem OPC Server (ggf. mit Cloudanbindung) und mehreren Clients zu errichten.

Hierbei gilt es das bestehende Konzept durch eine Schnittstellendefinition zu erweitern. Dies ist unerlässlich, da entschieden werden muss, welche Daten von dem Raspberry Pi des SGMs an den OPC Server geschickt werden sollen. Dieses sogenannte Framework stellt eine der Hauptherausforderungen bei der Umsetzung des bereits erarbeiteten Konzepts dar, dass mit OPC UA aber bewerkstelligt bar sein wird.

Eine weitere Möglichkeit ist die Entwicklung einer Smartphone App, die die Informationen der aktuellen Energiebilanzen im Betrieb des SGMs visualisiert. Die Möglichkeiten zur Erweiterung sind nahezu unbegrenzt. Grundlage hierfür ist allerdings die funktionierende OPC Server-Client-Cloud Architektur.