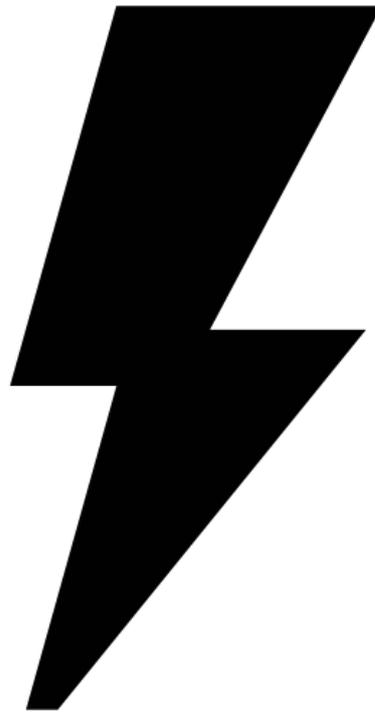


*Gewerblich-Industrielle Berufsschule Bern  
Allgemeinbildender Unterricht*

*Verminderung der Treibhausgase*

# Home Automation – Zeus



*Verfasser:*

*Ramon Marti, Informatiker*

*Agash Thamothersampillai, Informatiker*

*Fabrice König, Informatiker*

*Lehrperson:*

*Pascal Sigg*

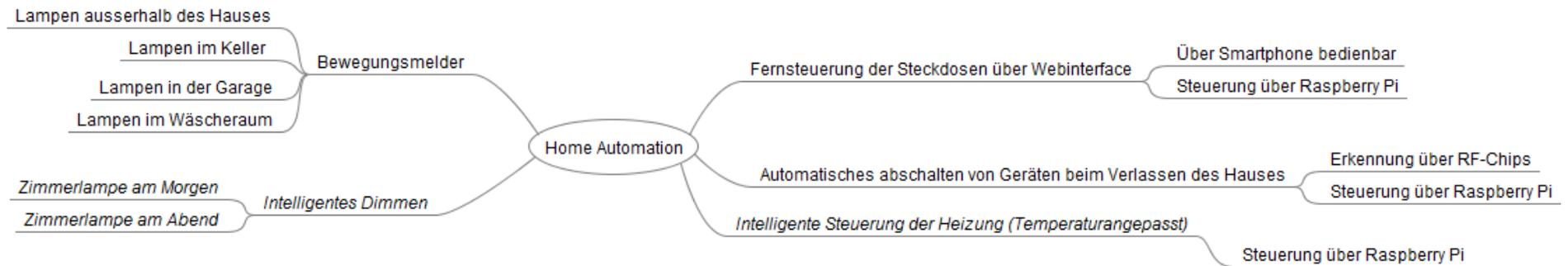
*Abgabetermin:*

*19. März 2014*

## *Inhaltsverzeichnis*

1. Mindmap .....	3
2. Einleitung .....	4
3. Zielformulierungen .....	5
4. Grobplanung .....	6
5. Zeus – Home-Automation-System .....	7
5.1. Einleitung .....	7
5.2. Inbetriebnahme .....	7
5.3. Smartphone-App.....	8
Einleitung .....	8
Testcases .....	9
5.4. Raspberry Pi .....	10
Einleitung .....	10
Funksteckdosen/Fernbedienung .....	10
Webserver / Service.....	12
6. Auswertung.....	13
6.1. Einleitung .....	13
6.2. Messungsergebnisse .....	13
Verbrauch an einem durchschnittlichem Arbeitstag.....	13
Verbrauch pro Jahr .....	13
6.3. Stromersparnisse.....	14
7. Schlusswort .....	15
8. Schlusserklärung .....	15
9. Arbeitsjournal.....	16

# 1. Mindmap



direkt übernommen	kaum bearbeitet	stark bearbeitet	<b>selbst erstellt</b>
Mindmap wurde mithilfe eines Tools komplett selbst erstellt.			

## 2. Einleitung

Unser Klassenthema ist die „Verminderung der Treibhausgase“. Unser Gruppenthema verbindet sich damit, indem wir Energie in Haushalten sparen. Dies erreichen wir damit, dass wir bestimmte definierte Geräte beim Verlassen des Hauses automatisiert komplett ausschalten (Steckdose wird ausgeschaltet).

Auf die Idee sind wir gekommen, weil wir etwas Intuitives machen wollten, etwas das wir Zuhause auch verwenden könnten und das nicht zu viel vom Benutzer verlangt, zumal wir Menschen meist zu faul sind um solche Dinge konsequent durchzusetzen. Da alle Gruppenmitglieder in diesem Projekt gefordert sind (Embedded-Geräte einrichten, konfigurieren; Android-App entwickeln; iOS-App entwickeln; Web-Services erstellen; etc.) erachten wir das Projekt als sehr gut geeignet. Auch ist es ein interessantes Projekt, das vielen Leuten klar machen könnte, wie viel Strom man eigentlich auf theoretisch einfacher Art und Weise sparen könnte.

Uns ist auch aufgefallen, dass es viele Home-Automation-Systeme gibt. Diese vereinfachen bestimmte Prozesse im Haushalt, keiner der Systeme beabsichtigt jedoch konsequent Strom zu sparen. Wir möchten das Stromsparen mit solchen Systemen fördern. In 5-10 Jahren kann man davon ausgehen, dass die Mehrheit der Haushalte über solche Systeme verfügen wird. Würde man das Prinzip als solches weiterführen, könnte man wahrscheinlich massenhaft Strom sparen, ohne dass wir Menschen direkt etwas machen müssten, einfach so, vollautomatisch.

Unser Ziel ist es also, ein kleines Home-Automation-System zu entwickeln, welches mittels Funksteckdosen diverse Geräte ausschalten kann.

Die Geräte werden dann ausgeschaltet, wenn alle Bewohner die Wohnung verlassen haben. Dies möchten wir entweder mittels Chips an Schlüsselanhängern, oder mittels Smartphone erkennen. Befinden sich also keine Bewohner mehr in der Wohnung, wird nach einem bestimmten Delay das Zeichen zum Ausschalten der Geräte (z.B. Entertainment-Anlagen) gesendet.

In einem weiteren Schritt nach Projektschluss werden wir eventuell den Sourcecode der Schnittstellen des Raspberry Pi's veröffentlichen, damit alle darauf zugreifen können um sich ein solches System selbst einzurichten.

Zusätzlich werden wir die Stromersparnisse bei uns Zuhause messen und diese entsprechend dokumentieren. Dies ist Teil der Mini-VA.

Wir möchten herausfinden, wie viel Strom man damit sparen kann, ohne das man selbst nach dem Einrichten jemals wieder etwas machen müsste.

direkt übernommen	kaum bearbeitet	stark bearbeitet	<b>selbst erstellt</b>
Die Einleitung, wie auch das Titelbild wurden selber erstellt.			

### 3. Zielformulierungen

Unsere Zielformulierung wurde in einem separaten Dokument geschrieben, hier sind die wichtigen zwei Punkte nochmals aufgeführt:

- **Im Rahmen unserer Projektarbeit entwickeln wir ein Home Automation-System, welches dazu beiträgt, Energie zu sparen. Dies möchten wir realisieren, indem wir ein Funksystem entwickeln, das erkennt, ob alle Personen das Büro oder das Zimmer verlassen haben. Ist dies der Fall, soll das System die definierten, an Funksteckdosen angeschlossenen Geräte, über ein Rapsberry Pi ausschalten.**
- **Wir vergleichen den Strombedarf mittels Stromzähler vor und nach dem Installieren des Systems, kommentieren die Ergebnisse und bewerten diese anhand von Grafiken auf ihren realen Nutzwert.**

## 4. Grobplanung

Woche	Arbeitsschritte	Verantwortlich	Bemerkung
6-7	<ul style="list-style-type: none"> <li>• Themenfindung</li> <li>• Gruppeneinteilung</li> <li>• Mindmap</li> <li>• Zielformulierung</li> </ul>	Agash Thamo., Ramon Marti, Fabrice König	Bereits angefangen, als dieses Dokument erstellt wurde.
8-10	<ul style="list-style-type: none"> <li>• Grobplanung</li> <li>• Konzeption des Projekts</li> </ul>	Agash Thamo. Agash Thamo., Ramon Marti, Fabrice König	-
10-12	<ul style="list-style-type: none"> <li>• Einkauf der Teile</li> <li>• Erarbeitung der Softwareschnittstelle auf dem Raspberry Pi</li> <li>• Erstellung der einfachen Android-Applikation</li> </ul>	Agash Thamo. Agash Thamo., Fabrice König, Ramon Marti Fabrice König, Ramon Marti	Ein Teil des Einkaufs wurde bereits erledigt. Die Android- Applikation hat Vorzug gegenüber der iOS-App, diese wird in diesem Projekt zwar auch angefangen, gehört jedoch nicht mehr zum Umfang der VA.
13	<ul style="list-style-type: none"> <li>• Testing der Software</li> <li>• Fertigstellung der Dokumente</li> <li>• Abgabe der Dokumente an „Klimawerkstatt“</li> </ul>	Agash Thamo., Ramon Marti, Fabrice König Agash Thamo., Ramon Marti, Fabrice König Agash Thamo.	Inkl. Zum Testing gehört auch die Veröffentlichung eines Teils der Software auf GitHub.
17	<ul style="list-style-type: none"> <li>• Präsentation</li> </ul>	Agash Thamo., Ramon Marti, Fabrice König	

## 5. Zeus – Home-Automation-System

### 5.1. Einleitung

Das Zeus-Home-Automation-System besteht aus 2 Komponenten: einem Raspberry-Pi mit gekoppelten Steckdosen und einer App, die auf einem iPhone und/oder Android-Gerät läuft. Die App trackt den Benutzer und falls dieser den von ihm definierten Bereich betritt oder verlässt werden die gekoppelten Steckdosen ein- respektive ausgeschaltet. Somit können Standby-Geräte automatisch ein und ausgeschaltet werden, ohne manuelles Eingreifen des Nutzers.

### 5.2. Inbetriebnahme

Das Raspberry-Pi wird mitsamt Steckdosen eingesteckt. Das Raspberry-Pi muss danach noch mit einem Netzwerk mit Internet-Anbindung verbunden werden. Der Router dieses Netzwerks muss die Öffentliche IP dem Raspberry-Pi vergeben. Zudem müssen am Raspberry initial die Koordinaten von der Wohnung hinterlegt werden. Die Webapp wird auf dem Raspberry-Pi gestartet.

In der Konfigurationsdatei der App wird die IP-Adresse des Raspberry-Pi eingetragen. Die App wird gestartet. Unter dem Tab Location kann nun manuell die GPS-Koordinaten eintragen oder die aktuelle Position abfragen und speichern. Mit dem Button Start Proximity Alert wird die Überwachung gestartet.

direkt übernommen	kaum bearbeitet	stark bearbeitet	<b>selbst erstellt</b>
Die Einleitung wie auch die Inbetriebnahme zum Produkt wurden selber erstellt, da das Produkt ebenso eine eigene Anfertigung ist.			

## 5.3. Smartphone-App

### Einleitung

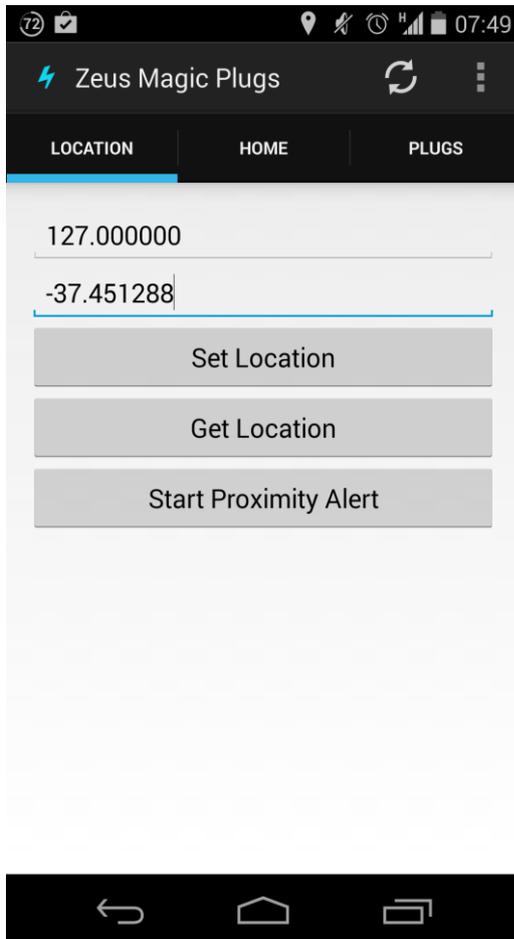


Abbildung 1 Android-App

Durch das Orten des Smartphones per GPS werden die Koordinaten des Benutzers an das Raspberry Pi gesendet. Dieses wertet die Koordinaten aus und vergleicht diese mit den hinterlegten Koordinaten der Wohnung.

Befinden sich alle Bewohner ausserhalb eines bestimmten definierbaren Radius, sendet das Raspberry ein Signal an alle definierten Funksteckdosen, woraufhin diese sich entsprechend ein- oder ausschalten.

Zusätzlich bietet die App die Möglichkeit, die Funksteckdosen (in der App „Plugs“ genannt) fernzusteuern.

## Testcases

Name des Testcases	Soll	Ist
<b>Positionsbestimmung</b>	Beim Klick auf "Get Location" sollte die aktuelle Position abgefragt werden.	Die Position wird abgefragt und eingetragen.
<b>Speichern der Präferenzen</b>	Beim erneuten Aufstarten sollte die App die gespeicherten Koordinaten automatisch ausgelesen werden.	Beim Starten werden die Koordinaten korrekt ausgelesen
<b>Proximity Alert</b>	Beim Betreten oder Verlassen des definierten Bereichs sollte eine Benachrichtigung in der Benachrichtigungsleiste erscheinen mit möglichen Fehlermeldungen bei fehlender Verbindung zum Raspberry-Pi	Die Benachrichtigungen werden angezeigt und die Befehle an den Raspberry-Pi geschickt.
<b>Ausschalten/Einschalten</b>	Mit den Befehlen über Http können die Steckdosen an und ausgeschaltet werden	Die Befehle funktionieren

direkt übernommen	kaum bearbeitet	stark bearbeitet	<b>selbst erstellt</b>
Abbildung 1: Screenshot von einem Teammitglied Der Text wurde ebenfalls selber erstellt, wie auch die Testcases zur App, welche wir auch eigenständig angefertigt haben.			

## 5.4. Raspberry Pi

### Einleitung

Der Kernteil unseres Systems ist das Raspberry Pi. Dieser verfügt über mehrere digitale Ein- und Ausgänge. Diese werden verwendet um die Funksteckdosen zu bedienen, resp. An den sogenannten GPIOs wird ein RF (Radio frequency)-Transmitter angeschlossen. Über dieses kann man anschliessend die Funksteckdosen ansprechen.

Der Vorteil von RF ist, dass man die Übertragung drahtlos durchführen kann. Zudem gibt es mehrere Home-Automation-Systeme, welche ebenso Komponente herstellen, die über RF angesteuert werden können (z.B. Heizkörperregler, Lampenfassungen, etc.).

Die Logik des Raspberry Pis wurde in der Programmiersprache Python geschrieben. Die Funktionalitäten sind über einen Webservice verfügbar. Die Smartphone-App bietet die Oberfläche dazu. Alternativ kann das System auch über den Browser gesteuert werden (Steckdosen, etc. an- und ausschalten).

### Funksteckdosen/Fernbedienung

Für unser Projekt haben wir verschiedene Funksteckdosen evaluiert. Eigentlich kommt es nicht besonders darauf an, welche man nimmt. Hauptsache die Frequenz der Steuerung beträgt 433 MHz, denn solch ein RF-Transmitter haben wir verwendet.

Diese Frequenz ist auch gängig bei anderen Home-Automation-Herstellern.

Folgende Steckdosen wären funktionstüchtig (unsere konnten wir im Internet nicht finden):

<http://www.conrad.ch/ce/de/product/059095/>

Man könnte nun wenn man möchte die Fernbedienung selbst an das Raspberry Pi anlöten und das Pi als logische elektrischen Schalter verwenden. Dies ist jedoch nicht besonders schön und bietet weniger Funktionsumfang, als wenn wir direkt mit dem Raspberry Pi Funksignale senden könnten. Deshalb benötigen wir einen RF-Transmitter, wir haben diesen hier verwendet:

<http://ch.farnell.com/1304024>

Anleitungen zum anschliessen eines solchen Transmitters an das Raspberry gibt es im Internet zur Genüge, deshalb wird das hier nicht weiter erläutert.

Nun kommt noch die Steuerung der Funksteckdosen. Für das Raspberry Pi, gibt es schon diverse Module zur Verwendung solcher RF-Transmitter in einem Programm. Wir konnten also ziemlich rasch Daten über das Modul versenden. Dazu haben wir die Codes der Funksteckdosen eingestellt und diese in eine Liste im Programm gespeichert. Diese Liste ist zusätzlich hinterlegt mit bestimmten Informationen, wie z.B. in welchem Zimmer sich diese Steckdose befindet, oder welche Geräte alle an dieser Steckdose dran sind. Diese Informationen könnte man später eventuell verwenden um das ganze System intelligenter zu machen, oder auch einfach um den Überblick zu behalten, wenn man etwas manuell ein- oder ausschalten möchte.

Jede Funksteckdose kann auf einen Haus-Code eingestellt werden. Dieser kann hinter an der Funksteckdose eingerichtet werden. Der Code besteht aus 5 Schaltern, die Kombination der Schalter ergeben den Haus-Code. Dieser sollte bei allen Funksteckdosen gleich sein, kann aber durchaus variieren, da man die Liste des Raspberry Pi's damit erweitern könnte. Die mitgelieferte Fernbedienung kann dann jedoch nicht mehr alle Steckdosen bedienen, da bei ihr auch der selber Haus-Code hinterlegt werden muss.

Ein Haus-Code könnte (logisch betrachtet) also etwa so aussehen:

**10110**

Anschliessend folgt nun die Gerätenummer, welche ebenfalls bei den Steckdosen eingestellt werden kann. Diese besteht meistens aus Buchstaben und ergibt eine Zahl. Der zu sendende Code besteht nun also aus dem Haus-Code und dem Gerätecode (in diesem Beispiel wird „B“ benutzt, also 2):

**10110 2**

Nun müssen wir dem Gerät noch mitteilen, was es tun soll. Die Geräte können entweder ein- oder ausgeschaltet werden. Der Code zum Einschalten wäre 1, der zum Ausschalten wäre 0. Möchten wir unser Gerät also ausschalten, so ergäbe sich folgender Code:

**10110 2 0**

Möchten wir das Gerät hingegen einschalten, so ergäbe sich folgender Code:

**10110 2 1**

Diese Codes können wir nun direkt über das Raspberry Pi an den Transmitter sende. Dieser schickt den Code über Funksignale aus, die Steckdosen erhalten das Signal, werten ihn aus und verhalten sich entsprechend.

Wurde auf dem Raspberry das Modul „Raspberry Remote“ installiert, so lässt sich der Code auf der Kommandozeile wie folgt senden:

**send 10110 2 1**

Dieser Befehl wird auch vom Webdienst aus an die Konsole übergeben, wenn der Dienst ein Gerät ausschalten will.

## Webserver / Service

Um die Funktionalität der Steckdosen überhaupt zu verwenden, haben wir einen Webdienst realisiert. Dieser verfügt über eine RESTful-API und wird wie folgt bedient:  
(folgende Beispiele sind Links und können von einer Smartphone-App oder auch von einem Browser durchgeführt werden. In den geschweiften Klammern sind jeweils die entsprechenden Werte einzufügen)

### Position senden

[http://raspberrypi/location/{mac\\_adresse}/{laengen\\_und\\_breitengrad}/](http://raspberrypi/location/{mac_adresse}/{laengen_und_breitengrad}/)

### Funksteckdosen (alle) mit Status abrufen

<http://raspberrypi/plugs/>

### Funksteckdose (bestimmte) Status abrufen

<http://raspberrypi/plugs/{id}/>

### Funksteckdose einschalten

<http://raspberrypi/plugs/{id}/on/>

### Funksteckdose ausschalten

<http://raspberrypi/plugs/{id}/off/>

### Funksteckdose Status switchen

<http://raspberrypi/plugs/{id}/switch/>

Anhand der Position die gesendet wird, weiss das Raspberry nun, wo sich wer befindet. Die Person, resp. Das Smartphone wird über die MAC-Adresse erkannt. Beim einmaligen einrichten des Raspberry Pis muss natürlich die Position der Wohnung an das Raspberry übergeben werden. Dies geschieht auf dem Server.

direkt übernommen	kaum bearbeitet	stark bearbeitet	<b>selbst erstellt</b>
Der Text wie auch die einzelnen Code-Beispiele wurden selber erstellt. Die Code-Beispiele sind Teilstücke des Programmcodes den wir für den Server geschrieben haben.			

## 6. Auswertung

### 6.1. Einleitung

Gemäss unserer Zielformulierung haben wir mehrere Messungen mittels Stromzähler durchgeführt. Die Messungen wurden jeweils vor und nach dem Installieren des Systems vorgenommen.

Die Zahlen der nachfolgenden Kapiteln und Grafiken basieren auf diesen Zahlen.

### 6.2. Messungsergebnisse

Mit den Messungen simulieren wir einen typischen Arbeitstag, an dem die Bewohner beim Verlassen des Hauses vergessen Geräte in ihrem Haushalt auszuschalten (komplett, d.h. kein Standby-Modus). Hierbei gehen wir von einer Abwesenheit von ca. 10h aus.

Die Messungen wurden mit einem einfachen Stromzähler an jeweiligen Steckdosen vorgenommen.

#### Verbrauch an einem durchschnittlichem Arbeitstag

Gerät	Vorher (10h)	Nachher (0h)	Bemerkung
Fernseher	0.05 kWh	0 kWh	Messungen im Standby-Modus
HiFi-Anlage	0.1 kWh	0 kWh	Messungen im Standby-Modus
DVB-T-Receiver	0.1 kWh	0 kWh	Messungen im Standby-Modus
Stehlampe	0.8 kWh	0 kWh	
Spielkonsole	0.03 kWh	0 kWh	Messungen im Standby-Modus
Kaffevollautomat	0.03 kWh	0 kWh	Messungen im Standby-Modus
<b>Total</b>	<b>1.11 kWh</b>	<b>0 kWh</b>	<b>Pro 10h (Tag)</b>

#### Verbrauch pro Jahr

Wir gehen davon aus, dass der Durchschnittserwachsene täglich während 10 Stunden das Haus verlässt, um Arbeiten zu gehen. Bei 45 Arbeitswochen (abzüglich Feiertage) mit Arbeitswochen à 5 Tage kommen wir deshalb auf eine Anzahl Arbeitstage pro Jahr von 225 Tagen. Multiplizieren wir nun den durchschnittlichen Strombedarf von Geräten im Standby-Modus während eines typischen Arbeitstages, erhalten wir einen Stromverbrauch von 249.75 kWh pro Jahr.

$1.11 \text{ kWh pro } 10\text{h} \times 225 \text{ Arbeitstage} = 249.75 \text{ kWh pro Jahr.}$

### 6.3. Stromersparnisse

Die Auswertung unserer Messergebnisse zeigt wie erwartet, dass durch das Ausschalten von elektrischen Geräten beim Verlassen des Hauses, eine grosse Menge an Strom und schlussendlich auch Geld gespart werden kann.

Durch den Einsatz unseres Home-Automation-System können jährlich ca. 219.09 kWh Strom gespart werden.

Dieser Wert ergibt sich aus dem zuvor gerechneten Verbrauch der Geräte im Standby-Modus pro Jahr abzüglich des Strombedarfs des Raspberrys, das wir einsetzen.

Stromverbrauch des Raspberrys an einem Tag:  $3.5W * 24h = 84 Wh = 0.084 kWh$

Stromverbrauch des Raspberrys in einem Jahr:  $0.084 kWh * 365 Tage = 30.66 kWh$

Stromersparnisse durch das Zeus – Home-Automation-System in einem durchschnittlichen Haushalt pro Jahr:

$$249.75 kWh (1) - 30.66 kWh (2) = 219.09 kWh (3)$$

1) Ersparnisse durch Zeus

2) Verbrauch des Raspberrys

3) Gesamte Stromersparnisse durch das Zeus – Home-Automation-System

Bei einem Strompreis von 25 Rp./kWh ergibt das einen Betrag von 54.70.- / Jahr.

Das komplette Zeus- Home-Automation-System wäre also bereits nach ca. 2.5 Jahren amortisiert.

direkt übernommen	kaum bearbeitet	stark bearbeitet	<b>selbst erstellt</b>
Die Auswertung wurde ebenso komplett von uns erstellt. Die Zahlen haben wir nachgemessen und anschliessend hochgerechnet.			

## 7. Schlusswort

In den vergangenen Wochen hatten wir die Möglichkeit, im Rahmen der Schule Gibb in Zusammenarbeit mit klimawerkstatt.ch ein Projekt umzusetzen, das zum Ziel hat, Energie zu sparen respektive den Treibhausgasausstoss zu verringern.

Als Naturfreunde ist uns das Thema Klimaschutz sehr wichtig, denn wir sind alle der Meinung, dass etwas passieren muss. Umso stolzer macht es uns, wenn wir an dieser Stelle auf die Messergebnisse bzw. Stromersparnisse unseres Home-Automation-Systems blicken, mit dem jährlich ca. 220 kWh Strom gespart werden können. Für uns ist dies ein klarer Erfolg – unser Ziel ist erreicht. Wir spielen derzeit bereits mit dem Gedanken, unser System zukünftig auf dem Markt anzubieten.

Dies alles wäre nicht möglich gewesen, wenn wir als Team nicht so gut zusammengearbeitet hätten. Leider wurden wir aber von diversen, negativen Ereignissen verfolgt. Sei es der Diebstahl der ganzen Hardware von einem unserer Projektmitarbeitenden oder die vielen Krankheitsausfälle. Eine grosse Arbeitsauslastung in den Lehrbetrieben sowie Militärabwesenheiten erschwerten uns die Organisation zusätzlich.

An dieser Stellen möchten wir gerne Herrn Sigg von der GIBB sowie Herrn Gantenbein von myclimate danken, ohne deren Unterstützung das Projekt nicht zu Stande gekommen wäre.

## 8. Schlusserklärung

„Hiermit versichern wir, dass die vorliegende Arbeit komplett selbstständig erarbeitet wurde. Wir haben keine Hilfe von Drittpersonen beansprucht. Quellen gibt es keine, da alle Daten selbstständig erarbeitet wurden. Die Erarbeitungsgrade in den Kapiteln entsprechen der Wahrheit.“

Agash Thamocharampillai

Fabrice König

Ramon Marti

---

---

---

## 9. Arbeitsjournal

KW	Zeit	Wer	Tätigkeit	Reflexion
9	2h	Alle	Grobplanung angefangen, Mindmap fertiggestellt.	Wir haben durch das Mindmap unsere Anforderungen an das Projekt selber, besser definieren können. Nun haben wir einen groben Plan, was wir alles machen möchten.
10	2h	Alle	Grobplanung fertiggestellt	Diese Woche haben wir die Grobplanung fertiggestellt. Es gab diverse Komplikationen da wir uns nicht ganz sicher waren, wann z.B. jemand abwesend sein könnte, da wir alle wussten, dass in der letzten Zeit ziemlich viele an die Rekrutierung mussten.
11	2h	Agash Thamo.	Benötigte Hardware evaluiert.	Wir haben noch mit Basil von myclimate telefoniert um nachzufragen, ob er uns das Projekt finanzieren könne. Er hat dies glücklicherweise bestätigt.
11	3h	Ramon Marti Fabrice König	Angefangen mit der Android-Applikation	Der GPS-Empfänger von Ramon machte nicht ganz mit, dieses war defekt. Wir mussten die App entwickeln und erst gegen Schluss auf einem anderen Gerät testen ob die App funktioniert.
12	5h	Agash Thamo.	Angefangen den Server aufzusetzen und die benötigten Funksteckdosen besorgt.	Wir hatten mehrere Funksteckdosen zur Auswahl. Wir haben uns schlussendlich für Outdoor-Funksteckdosen entschieden. Die sind Wasserdicht und haben eine grössere Reichweite.
12	5h	Ramon Marti	An der Android-App weiter gearbeitet.	
12	0h	Fabrice König	Abwesend: Rekrutierung	
13	5h	Alle	Dokumente abgeschlossen	