The Smarter Power Strip

- Design and implementation of a smart power strip

Bachelor of Science Thesis

MARCO BAXEMYR
JOHANNES BROBERG
KARL BUCHKA
MATTIAS HENRIKSSON
GUSTAV OLSSON
JOHAN SWETZÉN

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, June 2013

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.
The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

The Smarter Power Strip
Design and implementation of a smart power strip
MARCO BAXEMYR
JOHANNES BROBERG
KARL BUCHKA
MATTIAS HENRIKSSON
GUSTAV OLSSON
JOHAN SWETZÉN

© MARCO BAXEMYR, June 2013.
© JOHANNES BROBERG, June 2013.
© KARL BUCHKA, June 2013.
© MATTIAS HENRIKSSON, June 2013.
© GUSTAV OLSSON, June 2013.
© JOHAN SWETZÉN, June 2013.

Examiner: ARNE LINDE

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover: The final prototype describe in section 5

Department of Computer Science and Engineering
Göteborg, Sweden June 2013

Abstract

It is estimated that up to 20% of U.S. households' power consumption is used by standby equipment (Ross and Meier 2000). In this report, we describe the design and implementation of a *smart power strip*, with the purpose of lowering such useless energy waste. The device puts its users in control of their consumption by providing monitoring of the power drawn by appliances connected to it, and the ability to remotely turn these appliances on and off. Power consumption history and control can be accessed by users through a web application, and the power strip communicates with this web application over the Internet via an embedded wireless interface. Furthermore, through the open design of the device it is easy for third parties to create services that utilize it.

A set of desired features are specified, and a prototype is developed accordingly. To evaluate the prototype empirical tests are performed and results indicate that, with some further development of the software, its goals would be achieved. This opens up for the device to be used as a tool in the important task of lowering power consumption in today's society.

Sammanfattning

Enheter försatta i standbyläge genererar upp till 20% av de amerikanska hushållens totala elförbrukning (Ross and Meier 2000). I denna rapport beskrivs design och implementation av ett *smart grenuttag*, vars syfte är att sänka sådant energislöseri. Produkten sätter sina användare i kontroll över sin elförbrukning genom att ge dem möjlighet att övervaka hur mycket el enheter kopplade till den drar, och att på avstånd slå på och av dess uttag. Övervakning och kontroll erbjuds via en webbapplikation, och grenuttaget kommunicerar med denna webbapplikation över internet via ett inbyggt trådlöst gränssnitt. Vidare har produkten en öppen design som gör det enkelt för tredje parter att uttnyttja dess tjänster.

Ett antal önskade egenskaper specificeras och en prototyp utvecklas utefter dessa. För att utvärdera prototypen utförs ett antal tester och dessa visar att, om viss vidareutveckling av produktens mjukvara genomfördes, skulle den uppfylla sina mål. Detta öppnar upp för att produkten skulle kunna användas som ett verktyg för att sänka elförbrukningen i dagens samhälle.

Acknowledgements

We would like to thank our supervisors, Marina Papatriantafilou, Magnus Almgren, Vincenzo Gulisano and Giorgos Georgiadis for providing excellent help and feedback, making this project possible.

Contents

1 Introduction

In developed countries, households are responsible for more than 20% of the total energy consumption (Streimikiene and Volochovic 2011). Despite a trend for more energy-efficient electrical appliances, an ever-increasing amount of products result in larger net power consumption (Herring 2001). Even though these devices are not always powered on or used, the constant need for availability has resulted in standby modes that provide responsiveness at the cost of passive power drain. It has been estimated that an average continuous consumption of 67 W is used to power standby devices in homes today. This corresponds to 5-26% of the yearly electricity consumption of a U.S. home (Ross and Meier 2000).

In order to mitigate the ever-growing power usage, all methods need to be taken into account. One approach has been to classify electrical appliances and in this way create incentives for manufacturers to lower the power consumption of their products. While reducing the power consumption of electrical appliances is important, it is equally important to minimize their usage. By taking advantage of computer control, it is possible to better customize the use of electrical appliances to fit a particular lifestyle.

The concept of the smart house has emerged to denote more energy efficient, computer controlled buildings. Buildings account for around 73% of the total electricity consumed by a country, of which the total is equally divided between residential and commercial buildings (Bonino, Corno, and Russis 2012). The ability to control and analyze the appliances in a building, either automatically or manually, provides a lot of possibilities, both in terms of new features and in terms of lowering power consumption. The lack of smart features in a building makes it difficult to control the power consumption and finding where bottlenecks exist. However, these control and measurement tools are often meant to be integrated with the building and thus often require a significant investment from the user. It is not realistic to completely replace all housing with new smart buildings. We must instead use what we have in a better way.

We believe that, in order for smart house technologies to be adopted by the public, they need to be easy to use, cheap and require minimum investment from the user. Furthermore, we believe that the lack of feedback on energy consumption in conventional homes is a crucial part of the problem with the rising energy consumption. Without the ability to spot changes in power usage or the ability to attribute changes to specific devices, the user is effectively disconnected from his energy consumption.

We aim to provide a tool for users to make their power consumption visible and to put them in control of it. By empowering them in this way, they will be able to make more informed choices and, ultimately, lower their power consumption.

1.1 Purpose

The purpose of this project is to present a novel approach to a smart power strip, and develop a demonstration prototype. Through automation and usage visualization, it should have the potential to lower power consumption.

For each socket, the power strip should be able to *measure* current draw and *control* the output state. Measurement and control signals should be transmitted to a central server via an embedded wireless interface (see Figure 1). The data should be presented through an easy-to-use web application with the possibility to turn sockets on or off. Socket state should also be able to toggle via momentary buttons on the power strip.

The prototype aims to appeal to a broad target group of possible users. It should be straightforward and easily integrated into a user's living habits. All features provided which require user interaction should be simple to get started with and to use.

The prototype will throughout this report be referred to as *The Smarter Power Strip*.

1.2 Scope

We aim to present a product with the potential of lowering power consumption. However, this report will not provide any research regarding if this product is actually able to do that. Instead, it focuses on describing the development of a prototype with a set of features that *could* help a user lower his power consumption. Thereafter, it evaluates if the designing and implementation of these features were done in a satisfactory way (see Evaluation, chapter 6).

Prototype
While developing the prototype, it is assumed that the environment where it will be deployed is responsible for providing a reliable wireless connection and power supply. Hence, the prototype will not be designed to handle issues such as connection failure and power outage. Furthermore, the focus is to design and implement features with high performance. The cost of components and solutions will not be taken into account when developing the prototype, but will be discussed after it is implemented.

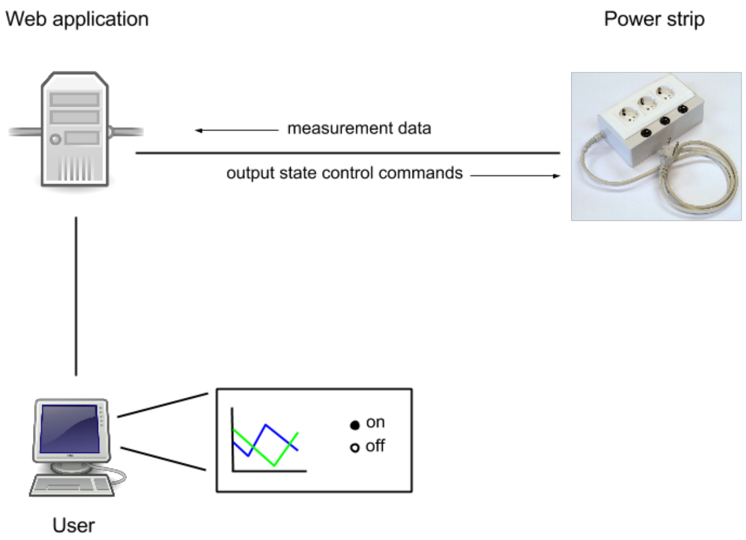


Figure 1: The power strip will be able to measure power consumption, which will be presented for a user by a web application. It will also be possible for a user to, via the web application or the momentary buttons on the power strip, toggle the state of each socket.

1.3 Problem/Task

Below, the purpose of this project is explained in more detail. Problems that need to be solved while developing the prototype are briefly introduced to the reader. These topics are then further analyzed, and different solutions are presented, in chapter 4.

Hardware
Every socket in the power strip will need measuring equipment so that the power consumption can be quantified, as well as a relay for turning it on and off. A microcontroller will be needed to collect the measurement data and control the relays, and a Wifi module for sending the collected data to the web server and receiving commands. There must also be hardware support for momentary buttons.

Communication
The communication between power strip and web application should be conducted over the Internet. The connection to a specific power strip should be owned by a specific user, meaning that one user should not be able to have access to control and measurement data of another user's power strip. Thus, there must be a way to connect a particular user to a particular power strip. Also, the power strip must provide an additional communication channel

other that the Internet through which it can be configured.

Security
Since the Internet is an insecure medium, there is a risk that an attacker might try to access a user's connection, gaining information of and control over the user's appliances. To minimize this risk, data being sent between the power strip and the web application should be encrypted in such a way that only the intended actors can control the device and supply or access power usage data.

Web application
The user and the power strip will communicate through a web application, which will receive the collected power consumption data and present it in an accessible way for the user, by using graphical diagrams. Both real time and historical consumption in different time intervals should be possible to view. Through the web application the user should also be able to send output state control commands to the power strip, such as "turn on socket 3".

In addition, the web application should provide the possibility to label power strips and sockets so that users are able to easily determine what room a specific power strip of theirs is in, or what appliance is plugged into which socket. Also, it should be able to handle different users, and provide identification possibility through log-in functionality so that only the owner of a specific power strip can access data and have control over that device.

Interoperability
To provide interoperability between the power strip and third party entities, a public *Application Programming Interface* (API) should be exposed. This will allow third parties to remotely control a user's power strips and access power measurement data, if authorized by the user. Users should be able to configure and authorize other software, such as the web service *If This Then That* (IFTTT 2012) or the *Android app Tasker* (CraftyApps 2012), to send requests to the web application server when certain events occur.

1.4 Method

In order to realize the purpose of this project, we defined a specification of the prototype which can be viewed in Appendix A.1. This, as well as the features described in the problem/task section above, was used as a template while designing and implementing the prototype. The solutions developed aimed to adhere to the specification in the best possible ways. Further, the theory on which the solutions were based was collected through studies of books, research papers, Internet forums and other relevant literature.

When the prototype had been implemented, it was evaluated regarding how well it fulfilled the purpose of the project. To see if the criteria defined in the specification (see section A.1) were achieved, suitable empirical tests were conducted. We also looked at if all the desired features described in the problem/task section (see chapter 1.3) were actually implemented.

Since this was an extensive project, and we did not have much experience in developing similar devices from before, planning the entire project in an early phase would not be possible. Therefore, we decided to work according to an agile process, were we could discover and solve problems while designing and implementing the prototype. A more detailed definition of the working process can be found in Appendix A.

1.5 Report layout

Following the introduction, this report starts by putting The Smarter Power Strip in perspective by describing some existing research and products with similar functionality in *chapter 2*. In *chapter 3*, a technical background is presented for the reader to understand the underlying technologies used in constructing the prototype. As this project requires knowledge ranging from electrical engineering all the way to computer science and software programming, a wide range of topics have been covered. Readers may find some parts trivial and others very hard to grasp, depending on the level of expertise in different areas, but this is to be expected. The technologies presented are then compared and contrasted in *chapter 4* and on the basis of this, the choices made for the prototype are presented in *chapter 5*. *Chapters 6* and *7* contain an evaluation of the prototype as well as a discussion on what went well and what could have been done differently. It all draws to a close in *chapter 8*, Conclusion.

2 Related work

Prior to The Smarter Power Strip, devices with the same purpose of lowering a user's power consumption, providing the same features of measuring and monitoring power consumption and controlling the states of electronic devices has been developed, both in the academic and the commercial sphere. Below, a selection of these is presented. This is followed by a brief comparison between those and The Smarter Power Strip.

2.1 Academic

Two projects that have been done in the academic context, which are similar to The Smarter Power Strip, are the WPCOM (Lien, Bai, and Lin 2007) and the ACme (Jiang et al. 2009). Even though they provide the same features as The Smarter Power Strip, their design differs a lot. Thus, they give an interesting contrasting view on how the purpose of this project could have been realized in other ways.

WPCOM

The *Wireless Power Controlled Outlet Module* (WPCOM) , like The Smarter Power Strip, is a device that has the physical form of an ordinary power strip, but contains some additional modules providing remote control and monitoring of power consumption data. Lien et al. investigates different communication solutions between user interface and power strip, such as RS 232, power line communication (PLC) and RF. However, in order to achieve a wireless, easy-to-setup, robust device the final WPCOM implementation provides users the possibility to connect to the power strip via Bluetooth, Ethernet (non-wireless) and GSM technology.

PDA software was developed through which a user could monitor and control the WPCOM. The idea is that a PDA could connect to the WPCOM via Bluetooth while a user is "at home", in the vicinity of the power strip. While further away the user is given the options to connect with the PDA via the Internet, or control and monitor the device using a cellular phone and the SMS technology, sending and receiving messages from the WPCOM.

ACme

The *ACme*, developed by Jiang et al, is not a single power strip-like device but a network of single-socket units. The main idea is to create a measurement and controlling system for a whole building without the need of existing network infrastructure. Since adding new devices in a Wifi network, as is done in The Smarter Power Strip, is considered complicated, ACme instead achieve its ability to be accessed wirelessly by so called mesh networking. One of the ACme nodes, i.e. one of the units in the ACme network, is given the responsibility of being end router and is connected with the Internet. The Internet access is shared with all other ACme nodes in the building via a technology called 6LoWPAN in cooperation with 802.15.4 (Montenegro et al. 2007), where all nodes cooperate in routing data between each other.

Each ACme node is given a unique IPv6 address and can therefore be accessed as an endpoint over the Internet. The nodes also expose an API through which any application can control their state or get measurement data. In the ACme project, just like in The Smarter Power Strip, a web application is built for the purpose of presenting the measurement data and through which nodes can be controlled. However, the API also provide possibility for other developers to use the ACme nodes and their functionality by building their own web application, mobile applications or similar.

2.2 Commercial

Several commercial products providing similar features as The Smarter Power Strip have also been presented and made available on the market. Some of these are *Tendril Connect* (Tendril 2013), *Fortum Hemkontroll* (Fortum 2013), Silver Spring's *Smart Energy Platform* (Silver Spring Networks 2013), *EnergyHub* (Energy Hub 2013), *GreenWave Reality* (*GreenWave Reality* 2013), *Belkin WeMo* (*Belkin WeMo* 2013) and *VisibleEnergy* (*Visible Energy* 2013).

The commercial solutions in general provide a large set of sophisticated features, such as defining schedules of when different sockets should be toggled on or off, possibility to design rules controlling states of sockets as a consequence of that specific events occur, and possibility to view not just the consumption of a single socket, but combined consumption of groups of sockets or the total consumption of a household. Also, the solutions are often made of more components than a power strip and a web application, such as meters, mobile applications, and more. In addition, several of the products provide interoperability through APIs.

The price on the products differs quite a lot. Some products comprises a purchase price and an additional monthly fee, and others only the purchase price. For instance, the Fortum Hemkontroll, which is provided on the Swedish market, costs 3000-3300 SEK at purchase and an additional 50 SEK every month. The Belkin WeMo on the contrary only costs 50 USD at purchase.

2.3 The Smarter Power Strip in comparison

As seen, The Smarter Power Strip will not introduce a new technology to the market. Several both academic and commercial devices providing the same functionality have already been presented. As will be presented in the following chapters of this report, what makes The Smarter Power Strip interesting is rather the simplicity of its design.

To the knowledge of the writers, none of the previously presented devices share the simplicity of The Smarter Power Strip. Looking at the WPCOM or the ACme for instance, they on the contrary comprise some rather complex technology, which means designing them required advanced knowledge on the subject. The Smarter Power Strip, however, is developed by a group of bachelor level students with limited domain knowledge under a limited period of time. The project will therefore provide an interesting view on whether it is possible to create a smart power strip with a less complicated design and under simpler circumstances than has been done before. Such information could for example be of interest for companies looking at developing an own smart power strip, evaluating how high level knowledge they will need.

3 Technical background

In this chapter, we provide information that will give the reader a theoretical background and a basic knowledge of the technology that was used designing and implementing the prototype, which is discussed in the following chapters 4 and 5. We also motivate why the solutions discussed in those chapters were needed in order to realize the purpose of The Smarter Power Strip.

3.1 Outlet toggling

In order to turn the sockets on and off, techniques for output state control need to be implemented. To control the output state, a device is needed that can break strong currents by applying a small control signal to it. The main alternative methods are described below, and section 4.2 compares them against each other.

3.1.1 Electro-mechanical relay, EMR

Conventional electro-mechanical relays uses a solenoid to create a magnetic field, when the control current or voltage applied exceeds a pickup value the switch is pulled. When the control signal is released the natural tension in the switch pushes it back to its default state (Elmore 2003). A variant to the standard EMR is the latching relay. The latching relay picks up of a single pulse from the control signal and keeps this state when the pulls stop affecting the coil. This helps to reduce the power dissipation in the circuit (Gurevich 2005).

3.1.2 Solid state relay, SSR

Solid state relays are a competitive alternative to conventional relays. SSRs use components made of semiconducting materials and no moving parts are therefore needed. Materials are divided into three categories according to their electrical characteristics, conductors, semiconductors and isolators. By manipulating semiconductor materials, it is possible to create components which electrical characteristics which can be controlled by applying a control voltage to them. SSRs operate with low losses and use different techniques to control the output state. SSRs are made for either direct current or alternating current applications. One common method for AC SSRs are optoelectronic relays which use LED-light to control electrical characteristics of the semiconducting material (Gurevich 2005).

3.2 Current measurement

In this section, basic theory regarding techniques for current measurement is presented. This will give a background on section 4.1, which discusses what solution that should be chosen in The Smarter Power Strip. Accurate measurement of AC-current is complicated which is described in section 3.3.2

3.2.1 Series resistance

The use of a shunt resistor is an easy and cheap way to measure current. The voltage drop over a resistor is proportional to the current through the resistor, by connecting it in series with the measurement object the voltage drop over the resistor will be proportional to the current drawn by the measurement object.

To be able to get a usable measurement, the resistor needs to be precisely specified and well dimensioned. The current and resistance will result in a voltage drop and consumed effect in the resistor. The voltage drop needs to be small enough to not affect the measurement object and the consumed effect need to be small enough to not overheat the resistor.

Most energy IC meters are not fully differentiated, which means they will measure the voltage between ground and a measurement point, to be able to measure with a shunt resistor one must establish a virtual ground on one side of the resistor. This means that the measurement circuit is coupled to a very high voltage, meaning the low-voltage circuit has a very high voltage compared to the grid ground or the actual ground we stand on.

3.2.2 Hall effect sensor

A Hall Effect current sensor is generally a monolithic IC that, when placed in series with the load, produces either a digital or analog signal corresponding to the current being passed through it. It is a relatively simple and compact way to provide current sensing while maintaining galvanic isolation between the load and measurement circuit. The main drawback, which relates mainly to currently available current sensor ICs, is relatively low measurement resolution.

3.2.3 Current transformer

The current transformer uses electromagnetic induction to transfer electrical energy from one circuit to another. The transformer consists of two coils which are wound around a common core, the core transfers the magnetic energy between the windings (S. A. Khaparde 2013). Equation 1 shows how the currents in the coils are dependent in a transformer. N_1 is the number of turns on the primary coil, N_2 turns on the secondary coil. I_1 and I_2 represents the current on the primary and secondary side.

The most common type of current transformer is the toroidal transformer. In the toroidal transformer the core is wrapped around the primary winding and the secondary winding is wrapped around the toroidal core. N for the primary winding is 1 in the toroidal transformer. To increase N, it is possible to wind the primary cable around the toroidal core, which decreases the transfer ratio. By connecting a burden resistor to the secondary side it is possible to measure the voltage drop over the resistor which is dependent on the current according to ohms law, $V = R * I$. Since there will be a inductance in the coils, there will be a phase shift requiring phase compensation for accurate phase measurement (*Elteknik* 2011).

$$\frac{I_2}{I_1} = \frac{N_1}{N_2} N_1 = 1 \Rightarrow I_2 = \frac{I_1}{N_2} \tag{1}$$

3.3 Power calculation

In this section, basic theory of AC power is described, and also how power consumption is calculated. The effects of power electronics are also discussed.

3.3.1 Alternating current

The grid provides AC-power, alternating current. In direct-current applications the power is calculated according to $P = V * I$. In a AC-circuit the voltage and current varies like a sine curve. The active effect is defined as the time integral of the instantaneous $V * I$. Since it is a product of the time integral it is not possible to simply use the mean values for voltage and current, one must use the root mean square value, the *RMS*.

However, in an AC-circuit there can be a time shift between the current and voltage they are out of phase, see figure 2. This means that there is a part of the RMS current and voltage which is not affecting the active power. This is the reactive power. Mathematically it is described as a complex value, where the active power is the real part and the reactive power is the imaginary part. The absolute value is the apparent power . Equation 2 shows this relationship. S is the apparent power, P the active power, Q the reactive power and ϕ the phase shift. j is the imaginary unit(Dorf and Svoboda 2011).

$$\underline{S} = P + jQ = V_{rms} * I_{rms} * (cos(\phi) + jsin(\phi)) \tag{2}$$

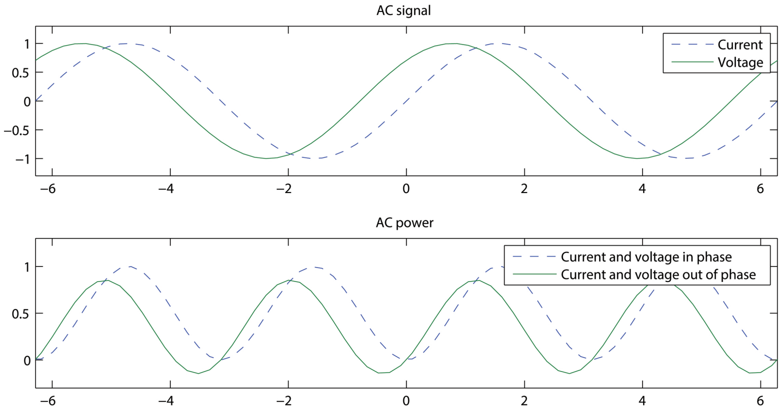


Figure 2: The upper plot shows a signal where current and voltage is out of phase. The lower shows the resulting power compared to the case where voltage and current are in phase.

3.3.2 Frequency components

The voltage in our grid varies as a sine curve with a frequency of 50Hz, as a result the current will also be a sine signal with a frequency of 50Hz for linear continuous loads, for example filament light bulbs and heating elements. Today more and more power electronics are used to regulate voltage for our electrical appliances. This is seen in compact fluorescent lamps, electrical machines and home entertainment systems to mention a few. This results in non-linear currents which contains harmonics. If the use of electrical appliances with non-linear loads increases they will distort the 50Hz voltage, therefore national authorities issues rules for electromagnetic compatibility (EMC). These rules both govern the disturbances allowed from appliances connected to the grid as well as the quality of the voltage on the grid. This ensures that the grid voltage frequency is kept at 50Hz with little noise. Since the voltage frequency is kept at 50Hz the current harmonics will not contain any energy and only contribute to reactive power (Math Bollen 2009). The standard IEC1268 (1996) - *Alternating Current Var-Hour Meters for Reactive Energy* defines the energy measurements for reactive power at the fundamental line-frequency (Calegari 2005).

The *Nyquist sampling theorem* states that any periodic signal can be completely reconstructed if it is sampled at a rate double to the highest frequency in the signal. The recreation of the signal is done with a algorithm called *fast Fourier transform*, FFT. The FFT will show the amplitude of the different frequency components(Alan V. Oppenheim 1997). This means that

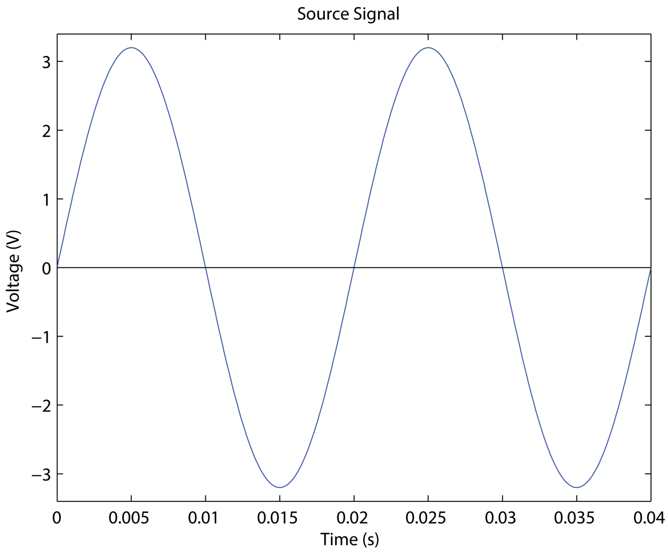


Figure 3: An AC signal

for a 50Hz signal the sampling frequency needs to be 100Hz. The standard *IE1036* for Active energy specifies current measurement to the 20th harmonic which means the sampling frequency needs to be 2kHz (Calegari 2005).

3.4 Digitization of measurement values

To be able to process the measurements, the analog measurement signal needs to be converted to a digital signal. This is done with a Analog-to-digital converter, ADC. The sampling frequency is the number of measurements per second the ADC makes. The sampling frequency is an important aspect of the ADC, it needs to be high enough to accurately represent the measured signal according to *the Nyquist sampling theorem.* The resolution is also critical to give a satisfactory view of the measured signal. The resolution sets the number of discrete levels the measurement interval can be divided into. The resolution is a power of 2. The standard ADC measures the voltage between ground and the input, and the signal needs to be positive.To be able to measure negative signals a bipolar ADC is needed. A bipolar ADC have both a negative and a positive power supply (Bishop 2007).

ADCs are usually sold with two different input types, *single ended* or *differential.* The single ended compares the input signal with ground and gives the result as output. The differential takes two inputs and gives the difference between the signals as output. The differential measurement is usually

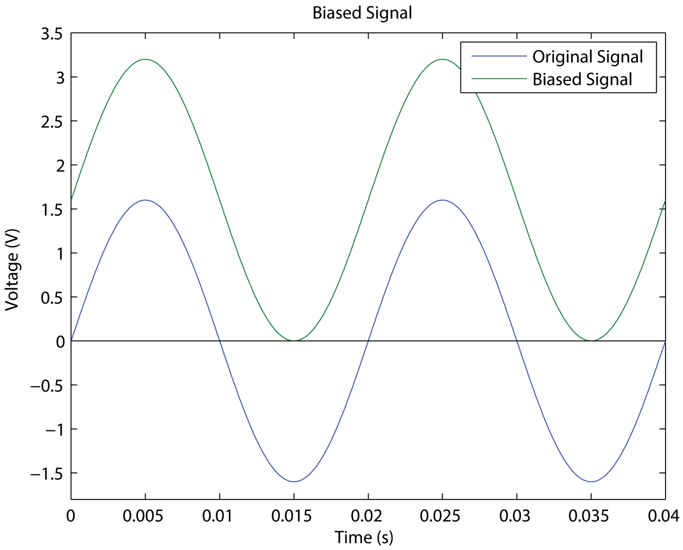


Figure 4: An AC signal which is biased.

done by comparing each signal to ground, the same as for the single ended. Then the difference between the results is used as output.

3.5 Making AC signals measurable by standard ADCs

As stated in section 3.4 the signal to a standard ADC needs to be constantly positive. An AC signal, which can be viewed in Figure 3, has a changing polarity and needs to be transformed into a signal that is constantly positive for the ADC to be able to measure it. There are two common methods for this. The first one is biasing. When biasing, a DC-component is added to the signal. This corresponds to adding a constant to the signal. Biasing is a simple method which is done with a simple circuit with a few resistors. Figure 4 shows a biased signal.

The second method is to use a full wave rectifier. The full wave rectifier inverts the negative parts of the signal as shown in figure 5. For large signals this is easily done with a simple circuit with four diodes. However, for weaker signals where the voltage drop over the diodes can not be neglected, the circuit needs to compensate for this. This is done with operational amplifiers. When a signal is rectified with this method the peak-to-peak value is halved. When measuring this means that the measurement interval can be used more efficiently.

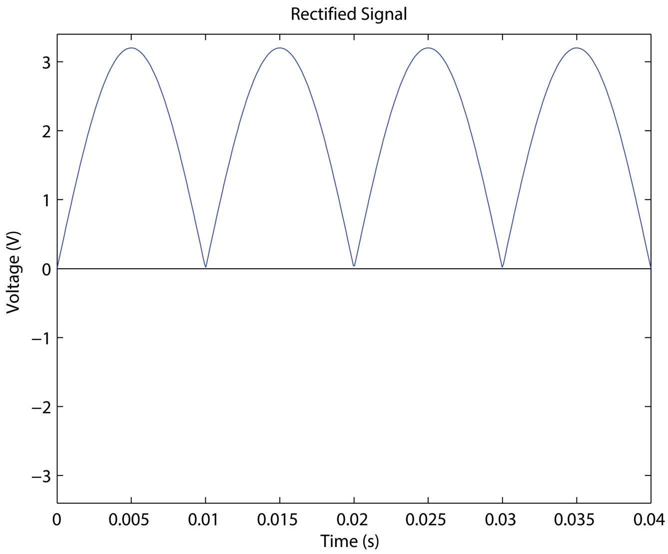


Figure 5: An AC signal which is rectified with a full wave rectifier.

3.6 Communication

The features that The Smarter Power Strip sets out to provide are based upon the fact that two of its main entities, the web application and the power strip, can communicate with each other. The power strip should be able to send measurement data to the web application, and the web application output state control commands to the power strip.

We have, because of its ubiquity and simplicity to get started with, chosen to conduct this communication over the Internet. The web application will be located on a web server, and the power strip will get its connectivity through an embedded wireless interface.

However, this decision opens up a range of problems that will have to be reflected upon and handled. Different possible solutions, and discussions of the suitability of these, regarding the most important problems are presented in chapter 4. Below, some introductory information needed in order to understand these solutions is presented.

3.6.1 Communicating over the Internet

The Internet can be described as a large number of so called *hosts* tied together by a big network (Kurose and Ross 2013). The hosts are all the devices between which communication is conducted, such as computers, servers, smart phones, et cetera. Data that is sent between hosts is passed forward towards their destination by *routers*, and every host has a specific router that it is connected to and through which it gets access to the Internet. This router, who handles the communication that goes in and out from a specific host, is called that host's *gateway*.

Every host and router has a unique address on the Internet called *IP-address*. The address is provided by an *Internet service provider* (ISP), i.e. an organization with access to the Internet backbone that offers customers connectivity, and is used when routing data to a desired destination. However, although a unique address for every host and router on the Internet was the original intent, this is not the entire truth. Since the number of hosts in an office or a home is very dynamic over time, it has been difficult for ISPs to know how many IP-addresses they should allocate for each of their customer (Kurose and Ross 2013). One approach that has become a popular solution on this problem is called *network address translation* (NAT) (Zhang 2007; Srisuresh and Holdrege 1999; Srisuresh and Egevang 2001). With NAT, however, comes new problems, which will have to be dealt with in the context of The Smarter Power Strip (see chapter 4.4.1)

The NAT issue

If a gateway router uses NAT, it is given a globally unique IP-address by some ISP in an ordinary manner. All the hosts that uses that router as a gateway, however, form a *local area network* (LAN) in which they are not given globally unique IP-addresses, but IP-addresses that are only unique in the context of the LAN (see Figure 6). Consequently, ISPs only have to provide a single globally unique IP-address to a customer, which is occupied by the customer's gateway. The customer can then add and remove hosts in the LAN without having to worry about them requiring globally unique IP-addresses.

However, since the hosts in the LAN have not got globally unique IP-addresses, they cannot be reached from what is usually called the *wide area network* (WAN), i.e. the network of hosts on the outside of the gateway. From the WAN's point of view, only the gateway is visible. Hosts on the LAN however can access host on the WAN since they all have global IP-addresses. This means the gateway has to distribute communication from the WAN to the hosts in the LAN, which it handles by keeping a reference table on which specific host in the LAN who sent a message to a specific destination in the WAN. When a message is returned from the WAN, the gateway checks the reference table and forwards the message to the correct host.

This situation will however still have a problem: even though hosts on the

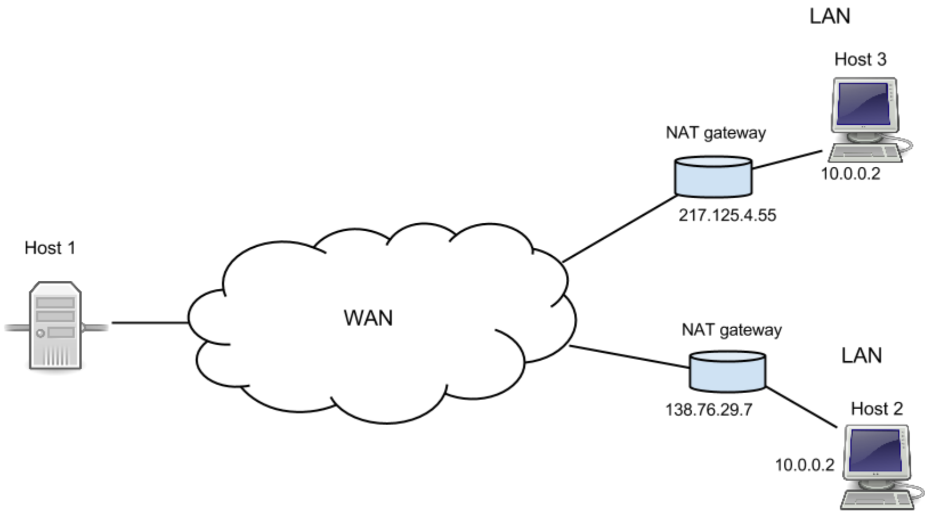


Figure 6: Host 1 cannot contact Host 2 or Host 3 since they are behind NAT gateways, and therefore does not have globally unique IP-addresses.

LAN can contact hosts on the WAN, hosts on the WAN will not be able to initiate contact with hosts on the LAN. This, since no reference table entry could have been created and, hence, the gateway will not know to which host it should forward the message.

One solution that has been proposed for this problem is called *connection reversal* (Ford, Srisuresh, and Kegel 2005). Connection reversal is only possible when one of the parties in the communication, the host on the WAN, is not behind a NAT-router and there is a server to which the host on the LAN has an ongoing connection. If that is the case, and the host on the WAN wants to connect to the host on the LAN, it can send a request to the server, who in its turn can send a request to the host on the LAN to contact the host on the WAN. At this point a connection will be established. Connection reversal is a commonly used technique for so called peer-to-peer (P2P) applications (Kurose and Ross 2013).

Another solution is a technique called *port forwarding* (Kurose and Ross 2013). Port forwarding uses an address number called *port number* which all messages on the Internet contain. A gateway can be configured so that if a message with a specific port number should arrive from the WAN, it will automatically be forwarded to a certain application process at a certain host in the LAN. The port forwarding configuration can be done manually, but also automatically by protocols such as the *Universal Plug and Play protocol* (UPnP; UPnP-Forum 2013).

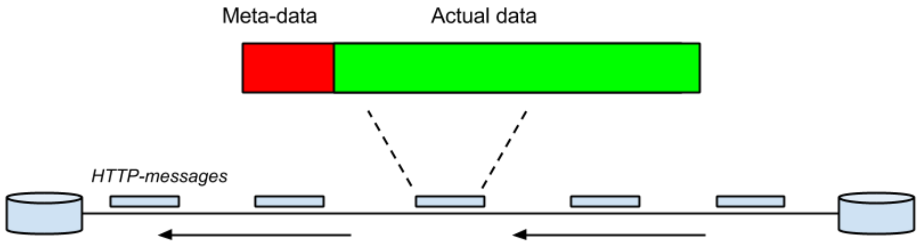


Figure 7: Apart from the data that should actually be communicated, HTTP-messages contain a portion of meta-data.

3.6.2 HTTP

In order for communication between entities over the Internet to work, a common *protocol* is needed. A protocol is an agreement between the entities on how the communication will work, what rules it will follow and how data sent will be structured (Kurose and Ross 2013). The protocol that handles communication for web clients accessing websites on web servers, but also can be used in other context such as in The Smarter Power Strip, (see chapter 4.4.2) is the *Hypertext Transfer Protocol* (HTTP; Berners-Lee, Fielding, and Frystyk 1996; Fielding et al. 1999).

While communicating over HTTP, so called *HTTP-request* and *HTTP-responses* are sent. The HTTP-requests coming from the client consists of a path to a web page on a web server it wants to access. The response sent back from the web server contains the content of the web page requested. Although web page content is what a HTTP-response is originally meant to contain, it could contain any arbitrary data defined by the sender. Also the HTTP-request could have any data attached to it.

As any other protocol, HTTP-messages do not only consist of the data that should be communicated, but also a set of Meta data (see Figure 7) which contain details about the data sent relating to the specific communication situation. Since HTTP is defined for handling web client-server communication, the Meta data in HTTP-messages is used to make such communication function. The Meta data in a HTTP-request contains information specifying for example which method, i.e. if the data sent should change the content of the website or just read website data, the message is of, what web browser the client uses, in which preferred language the response content should be given and more, while the HTTP-responses contain information such as which server software and version the web server uses, when the content on the web page requested was last modified, et cetera.

3.7 Security

The system in which The Smarter Power Strip will function is a distributed system where all power strips are controlled and have their power consumption data analyzed remotely by the web application over the Internet. Furthermore, the web application is accessed remotely by users. Since the smart power strip can be used to control an individual's appliances and access her power consumption history, it is important that the communication channels used are robust and secure.

To achieve secure communication, the following four properties need to be fulfilled (Kurose and Ross 2013):

- *Confidentiality* - The confidentiality of a communication channel is the property that only the sender and the intended receiver should be able to understand the data being transmitted.

  Andrés Molina-Markham et. al., in their paper Private Memoirs of a Smart Meter, illustrate how easy it is to infer sensitive information by merely having access to a log of a households' power consumption every second. Using off-the-shelf statistical methods, a range of questions like how many people are in the home as well as eating and sleeping habits can be answered. Considering that the smart power strip not only reveals aggregate household power consumption, but also which device uses what and when, the privacy implications would be even greater than the smart meter considered in the paper. To protect against this, confidentiality is needed. (Molina-Markham et al. 2010)

  An open communication channel would pose a great risk to the privacy of users of The Smarter Power Strip and it is important that data sent and received is encrypted.

- *Message integrity* - Message integrity is the property that the data in transit over a communication channel must not be altered before it reaches its intended destination.

  If message integrity is not enforced on the communication channel, it will be possible for a third party to have some degree of control over a user's appliances. Even if only partial or limited control is exposed, it could be very dangerous.

- *End-point authentication* - End-point authentication is needed to verify the identity of the parties involved when communicating over a communication channel.

  For The Smarter Power Strip, it is important that the hardware only acknowledges instructions from the web application and that the web application only accepts consumption and state data from the hardware. If no end-point authentication is used on the communication channel, a third party will be able to have full control over a user's appliances. As with message integrity, this could be very dangerous and end-point authentication should be enforced.

- *Operational security* - Having the tools to detect and act on malicious activity is important when an attack on a distributed system occurs.

  A power strip should be recoverable in the sense that any stored data that is used to facilitate the secure communication should be exchangeable without replacing the hardware.

3.7.1 Encryption

To ensure that communication conducted over the Internet is confidential, data sent should be encrypted. Encrypting is the process of substituting a plaintext message with a so called *ciphertext*, which is an unreadable modification of the original message (Kurose and Ross 2013). The ciphertext can be securely sent over the Internet without there being any risk of anybody else than the intended receiver understanding it. However, for encryption to be meaningful, the receiver must be able to decrypt the message as it arrives to him. This process can be done in some different ways.

Symmetric key encryption

One widely used encryption method is called *symmetric key encryption*. The method's main idea is that both of the parties in the communication have access to the same secret key, which is used for both encryption and decryption (Kurose and Ross 2013).

The strengths of symmetric key encryption are that it is both secure and relatively fast. Using a key size of k bits, there are 2^k different keys with which the data could be encrypted, therefore making it hard to brute-force for sufficiently large values of k. Also, compared to public key encryption, it consumes a small amount of computational resources. However, in order for the method to work, both parties in the communication must have access to a shared and secret key. Symmetric key encryption provides no solution for sharing this key securely between the parties. (Kurose and Ross 2013)

The widely used *Advanced Encryption Standard* (AES) is a symmetric-key algorithm.

Public key encryption

Another encryption method is so called *public key encryption*. Here, communicating parties do not share a secret key. Instead there are two different keys involved; a *public key* and a corresponding *private key*. Both parties have their individual public and private keys, where the public key is known to everyone, but the private key is kept secret. If party A wants to send a confidential message to party B he encrypts the message using B's public key, turning it into an unreadable ciphertext. The only way to decrypt this ciphertext is by using B's private key, which accordingly only can be done by B.

Consequently, no shared secret keys are needed, which is also the biggest strength of public key encryption. However, using the private and public keys requires more sophisticated mathematics than in the case of symmetric key encryption, making the method computationally expensive.

3.7.2 HTTPS

In order to provide a solution for secure communication on the Web, the *Hypertext Transfer Protocol Secure* (HTTPS) has been presented. HTTPS has become widely adapted and comprise, among other features, a combination of public key encryption and symmetric key encryption, where public key encryption is used to exchange secret keys. HTTPS is not an individually defined protocol, but is rather a collaboration between the HTTP protocol, which provides the features of HTTP communication (see chapter 3.6.2), and the *Transport Layer Security* protocol (TLS; Dierks and Rescorla 2008), which provides the security features.

4 Discussion on design alternatives

As we saw in chapter 1.3, there are several problems that needs to be solved in order to realize the purpose of The Smarter Power Strip. This chapter will describe different solutions for doing this. The solutions will then be compared against each other in the context of how well they serve the project's purpose.

4.1 Current measurement

When choosing a current measurement method it is important to consider size, accuracy, heating, and noise of measurement technique. To reduce noise it also desirable to have galvanic isolation between the measurement circuit and the load circuit. This is one of the drawbacks with the shunt resistor method. The shunt resistor will also require that the measurement circuit is grounded at the shunt resistor resulting in a high ground potential. When it comes to the current transformer and the Hall-effect sensor, both techniques offer galvanic isolation. The prototype is aimed to measure low effects with high accuracy and the available Hall-effect sensors do not provide the accuracy needed.

When choosing the ADC sampling frequency, resolution and the input type is important. As the Nyquist sampling theorem states the sampling frequency needs to be twice of the highest frequency that should be measured. The Smarter Power Strip focuses on active power and a stiff grid is assumed. There is therefore no need to sample at a frequency higher than 100Hz. Most ADCs only handle positive signals and therefore, the signal needs to be rectified. The simplest way to rectify a signal is by a simple circuit with four diodes. These diodes will result in a voltage drop and when working with weak signals, this would seriously wreck the signal. Instead an "Ideal rectifier" is constructed. The ideal rectifier uses operation amplifiers to invert the negative signals.

4.2 Outlet toggling

Since the Solid State Relay (SSR) made its appearance some decades ago a discussion has been going on whether which is better: SSR or electromechanical relay (EMR). The answer depends on the application. The SSR, with no moving parts, will usually outlast the equipment, run quietly and produce little interference. They also usually employ "zero-voltage crossover" and there is therefore no need to monitor the crossover to avoid current surges. The low power consumption is a great advantage. When SSRs fail, which they seldom to, they usually fail shortened which in some applications may cause danger. SSRs do not physically separate the conductors and they are therefore never completely off, and when on they are resistive which can lead to significant heat (Mahaffey 2002). The third alternative copes with the drawback of the high consumption of EMRs to the cost of a higher price.

4.3 Resolution

To get usable data for standby equipment the power strip needs to be able to measure accurately for small effects. $1W$ corresponds to a $4.34mA$ current and the maximum current that the power strip is dimensioned for is $10A$. With current ratio of 1000:1 this will result in currents from $1.43\mu A$ to $100mA$. To be able to get a $2W$ resolution according to the specification (see section A.1) the ADC needs to be at least 11 bit. The ADC works in an interval from 0 to $3.2V$, which means that $2W$ is equal to $2.78mV$. This is a very low voltage and it will be very hard to separate from noise in the circuit. Since the ADC only works with positive signals one must either rectify the signal or bias it.

A normal rectifier is a circuit with four diodes. However such a circuit will cause voltage drops and require the measured signal to be large enough, larger than the small standby voltages that the power-strip is needed to measure. A alternative is to construct an ideal rectifier. Very high precision components will be needed to cope with the small signals. One advantage with the rectified signal is the fact that the voltage drop over the burden resistor can be doubled.

4.4 Communication

Having communication over the Internet is a task that causes several different issues that need to be solved. For instance, there needs to be a pattern for the communicating parties to follow, and communicating entities must be layered in some way. Below, different solutions on these questions are analyzed and compared against each other.

4.4.1 Two-way-messaging versus polling

One of the aspects that need to be considered is according to which pattern the communication should be conducted, i.e. what roles and responsibilities each party of the communication should have, what are the rules of the communication, et cetera. Developing The Smarter Power Strip, two different patterns are of particular interest: *two-way-messaging* and *polling*.

The main idea with two-way-messaging, which is illustrated in Figure 8, is that both parties should be able to contact the other party at any given time. As soon as one party has a message for the other it can send this,

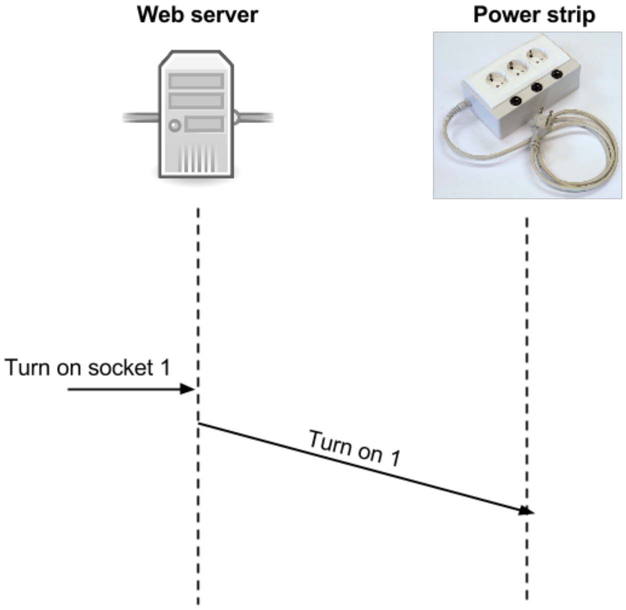


Figure 8: Two-way-messaging. The web server can notify the power strip instantly when a state change command is issued.

knowing that the other party will be ready to receive it. Such a relationship can also be referred to as *peer-to-peer* (P2P) communication (Kurose and Ross 2013). In polling on the other hand only one of the parties, *the client*, has the right to contact the other, *the server*. This means that if the client wants information from the server he has to ask, or *poll*, for it (see Figure 9). Since the server cannot know when the client has generated new information, he has to send polls at regular intervals controlling this. This is a relationship that also can be referred to as *client-server* communication.

These methods both have their pros and cons. In the context of The Smarter Power Strip there will be two entities – the web application and the power strip – who both will be able to generate data – commands and measurement data – that they want to send to the other. One could argue that this speak for two-way-messaging as the communication pattern of choice, since two-way-messaging allows both parties to send messages to the other and polling does not.

There are also incentives for the communication to happen instantly, which is another circumstance that implies two-way-messaging is the most suitable communication pattern in this situation. For example, if a user wants to change a state on a socket on his power strip to turn on a light connected to

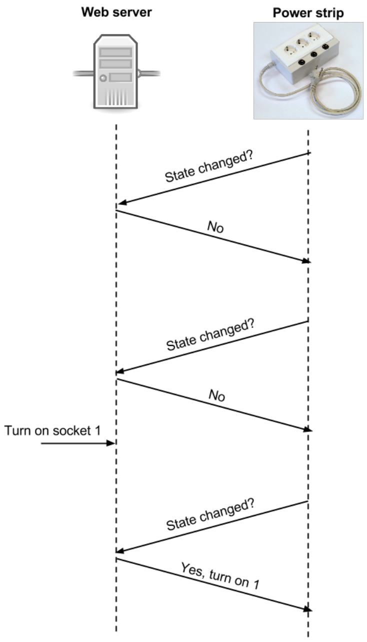


Figure 9: Polling. The power strip asks the web server at regular intervals if there has been any state change commands issued.

that socket, the state change must happen rather instantly. If not, the user may feel that he might as well move to the light and turn it on in an ordinary manner. In that case, the remote controlling feature of The Smarter Power Strip would have lost much of its value.

However, two-way-messaging requires from both the parties that they at any time are able to receive a message from the other. This is not a problem for a web application at a web server, but on the power strip side it leads to some issues. For one thing, it implies that the power strip's Wifi module must be active, draining power, constantly. It also implies that the power strip must be accessible via the Internet from the web application's location constantly. If the power strip is connected to the Internet through a user's home router, and that router uses NAT, it will block messages from the web application that the power strip has not explicitly asked for unless it has been configured not to (see chapter 3.6.1).

If the method of choice instead would be polling, these problems would be solved. The power strip could sleep, saving power, while not sending messages to the web application since it would not have to be ready to receive anything. Also, there would not be any accessibility issues due to blocking NAT routers since messages would not need to be sent to the power

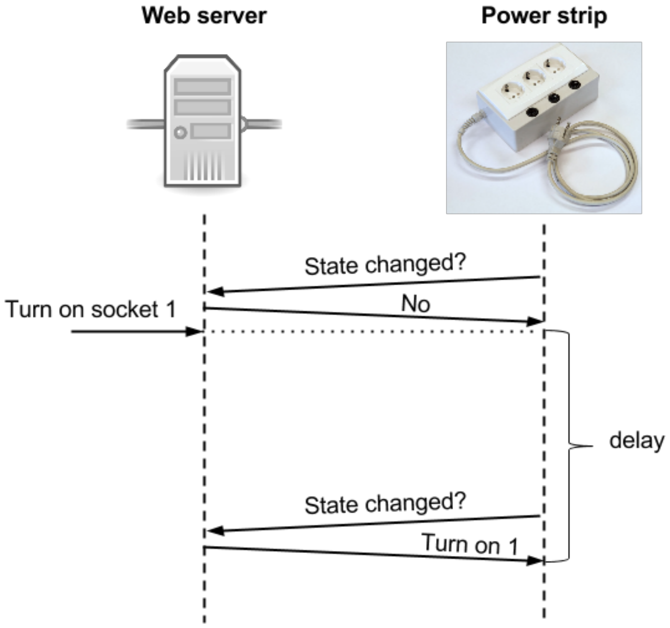


Figure 10: The consequence of using polling is a possible delay between the command being issued in the web application and the actual change happening at the power strip. The length of the delay depends on the polling frequency.

strip. The drawback, though, would be that instant communication would no longer be possible. The power strip would poll the web application, checking if something had changed at regular time intervals, and the delay, i.e. the time it would take from that a command was issued on the web application to when a change is made on the power strip, would be at most the poll interval length. This problem is depicted in Figure 10.

Making the interval very short would therefore solve the problem with delay between commands being generated in the web application and the actual change happening in the power strip. However, short intervals would mean that the Wifi module could not sleep much, and also that a lot of unnecessary data would be sent. Unnecessary, in the meaning that most of the requests would not generate any changes in the power strip which was the communications initial purpose, therefore not adding any value to the product.

One way to solve this issue is to regulate the frequency in which polls are sent depending on if the web application is likely to generate new data. In order to change a state of a socket in the power strip, a user must log on to

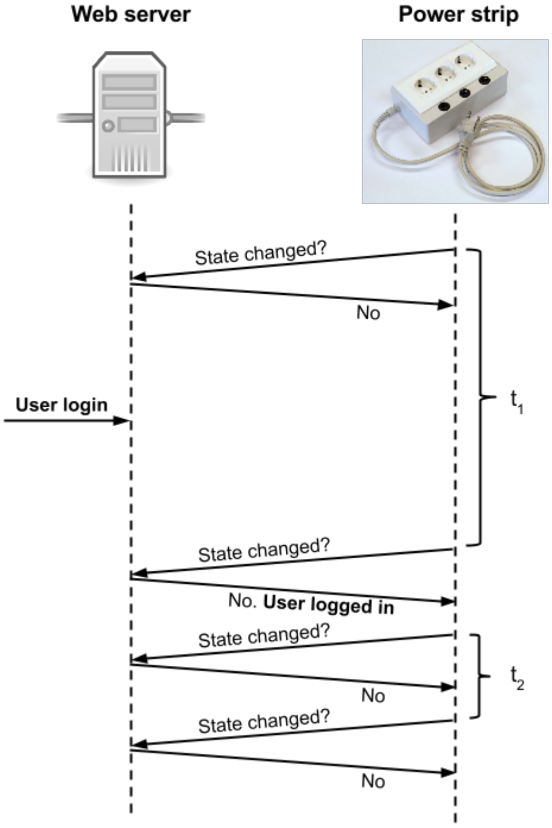


Figure 11: While the user is not logged in, the power strip uses a polling frequency interval of t_1. When a user logs in the interval is increased to t_2.

the website. Thus, when a user is logged in, one can expect commands to be generated. During this state we want the communication to react faster, which we achieve by increasing the polling frequency. Therefore, when a user logs on to the website the power strip could be noticed of this and the polling frequency could be increased (see Figure 11). When the user logs out it can be lowered again, giving the power strip the possibility to sleep more consuming less power.

However, for the power strip to register that the user is logged in, a message has to be sent from the web application. This will not be done until the power strip send its next poll. Therefore, the longest time it could take for the power strip to register that a user is logged in would be the poll interval time. This can be an issue, but only if the user can log in and generate a command in a shorter time than it takes for the power strip to register that the user is logged in, since in this case there would be a delay on the command to be executed. To prevent this from happening, the maximum interval of the lower polling frequency should be the equal to the shortest time interval it could take for a user to log in and generate a command on the website.

4.4.2 Layering between communicating entities

In the The Smarter Power Strip, communication will occur between the power strip and the web application. The power strip should be able to send measurement data to the web application and the web application commands for changing states on sockets to the power strip. However, there is also a third entity that needs to be included in the communication: a *database*. This, since the socket states of a power strip must be stored somewhere (see chapter 4.6), and since it should be possible to view historical measurement data. From the database, the web application should be able to fetch data when a user wants to view a power consumption graph on the website, and the power strip should be able to get information on what state each of its sockets should be in.

This situation, when a web application which is located on a web server, a device interacting with the web application and a database which is also located on a server, shall communicate over the Internet can be formed according to different structures. Structures, as in who will communicate with who, how will the communication be layered, et cetera. One way is to have the database server in the middle of the communication (see Figure 12, alternative 1). Both the web application and the device will then read and write data to the database and will not have any direct communication between each other. From this follows that the database server must be able to communicate with both the web application and the power strip, and thus must be equipped with software for handling both these communication channels. There must also be a common protocol, both between the database server and the web application, and between the database server and the power strip.

Another approach on the communication structure would be to have the web application in the middle of the communication (see Figure 12, alternative 2). Consequently, the web application would handle the reading and writing of data to the database, and the device would have to work through the web application in order to access the data. Since many web frameworks, for example Django which will be used in The Smarter Power Strip (see chapter 5.5), contains embedded support for reading and writing data to a database, this alternative would be easy to get started with.

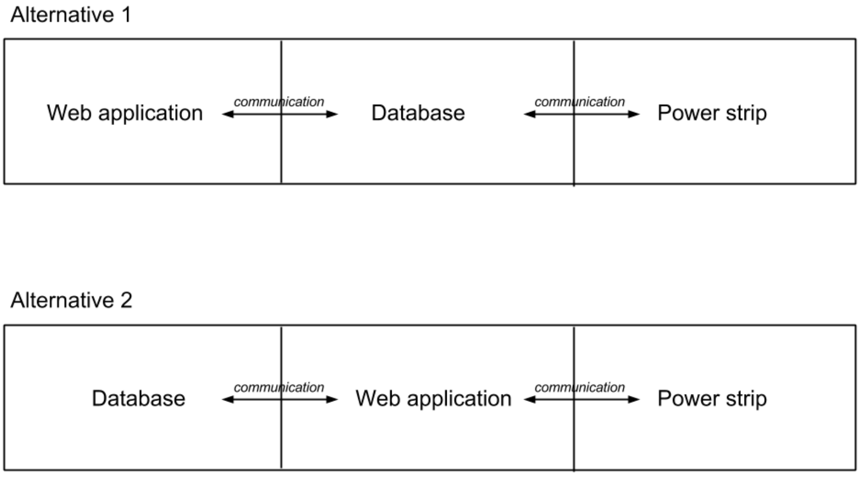


Figure 12: Two alternatives on how the layering could look like. Either the database (Alternative 1) or the web application (Alternative 2) could be in the middle of the communication.

4.5 Symmetric- versus public key encryption

In The Smarter Power Strip, secure communication between power strip and web application is a feature of significant importance (see chapter 3.7). Therefore, a robust security solution should be implemented. However, in the reality of the prototype there are several constraints limiting which solutions are appropriate and which are not.

First of all, there is the *computational resource* constraint. Since the main goal of the product is to help users lower their power consumption, we want the device itself to draw as little power as possible. The more memory and computational power, i.e. computational resources, a security solution would require from the microcontroller on the power strip, the more power it would draw. Thus, the less computational resources needed by the security solution chosen, the better.

There is also a constraint on how long time that can be spent researching and implementing different alternatives. Since we do not have a lot of experience in the security field, we need a solution that is simple and easy to implement.

One of the features needed in a security solution is encryption. The *Advanced Encryption Standard* (AES) is an encryption method that can do this in a simple and low-power consuming, but still reliable way. AES is a *symmetric key encryption* algorithm, which means that both communicating parties need a common and secret key that they can use to encrypt and decrypt messages. In order for the method to be secure however, the exchange of the key must also be secure, and therefore it cannot be sent in plaintext over the Internet. In the context of The Smarter Power Strip though, this could be easily avoided. During production a key could be put in the power strip, and on the other end stored in the central database. When a user plugs in the power strip at home, AES would be set up and ready to be used.

Unfortunately, there are other ways to obtain an AES key than intercepting a plaintext version of it. For example, an attacker could try decrypting the message with every possible AES key (brute-force). This problem is usually mitigated by regularly changing the key. Accordingly, this is something that should be possible in The Smarter Power Strip. However, this means that new keys have to be exchanged between power strip and web application when they are separated. If an attacker has obtained the key and listens to the communication between web application and power strip, then he will be able to intercept new keys sent and keep on listening for as long as he wishes. This is usually solved using *public key encryption*. Public key encryption can accomplish confidential communication without the need of a shared secret key. However, the method does consume more computational resources than symmetric key encryption. Therefore, symmetric key encryption is often used to encrypt all communication between two parties except when a new common key needs to be shared. At that time public key encryption is used. This is also the method that is used by the HTTPS protocol.

Another way of solving the problem would be to distribute the new key on another media than over the connection between the web application and the power strip. One way of doing that would for example be to periodically inform the user that it is time to change AES key via the website. When a user logs on to the site he can be provided a key that can be inserted manually, through the configuration interface, into the power strip. This way, the new key would not need to be transported anywhere where it could be intercepted by someone unintended.

4.6 Where to store the sockets' state

The idea of being able to control sockets' state on a power strip with software relies on the fact that somewhere, a software model describing if a socket is on or off is stored. Accessing the model must be possible both for parties trying to read the state of a socket and for parties trying to modify it. In the context of The Smarter Power Strip, the parties that would want to read the state of a socket is the hardware providing load to it, and the

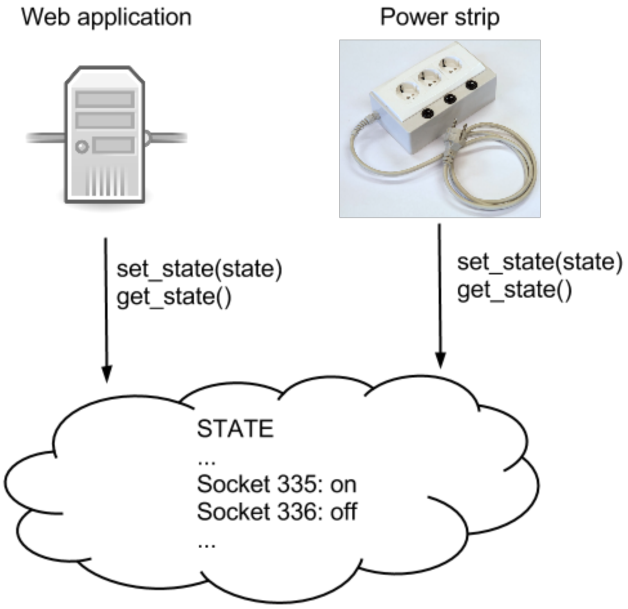


Figure 13: The state of a socket should be able to be read or modified from both the web application and the power strip.

website presenting if a socket is currently switched on or off (see Figure 13). The parties that would want to modify the state are the web application from which a user should be able to toggle socket output state, but also the momentary buttons on the power strip. The fact that the state should be able to modify and read from two remote locations raises the question were the software model representing the state should be stored; at the power strip close to the momentary button hardware, or at the web application.

If the state would be stored at the power strip, it would be easy to access and modify it from the momentary buttons. However, changes issued on the web application would have to be sent to the power strip. As a consequence of power strips being located behind routers with NAT, this solution would unfortunately cause other problems (see chapter 4.4.1). Placing the state on the web application instead, changes issued there would not have to be sent, but could be stored directly at place. The toggling hardware on the power strip could access the state by polling the web application.

This solution however raises another problem. Having the state located on the web application means that changes made by the momentary buttons would have to be sent there. The hardware toggling the load supply on the power strip would not be notified of the state change until a poll

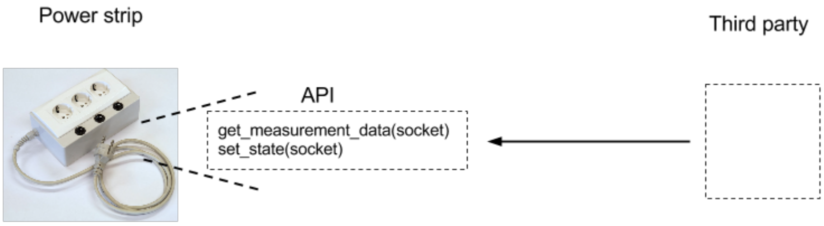


Figure 14: Power strip level API. A third party can access the power strip's services by using the exposed methods.

with state information would be answered. Consequently, there would be a delay between when a button was pushed and when the actual state change would happen. In the specification of the prototype (see section A.1) it is stated that response time should not be higher than five seconds, which is something that might be risked in a setup like this.

To solve this problem as well, there is an alternative to keep the state both at the power strip and the web application. Changes issued in the web application could modify the state located at the web application and changes issued by the momentary buttons could modify the state located at the power strip. The hardware performing the toggling of sockets could quickly access the state on the power strip and issues made on the web application could be stored at location, not having to deal with routers with NAT.

4.7 Power strip- versus web application level API

The Smarter Power Strip is a product that would gain from providing interoperability with its functions to third parties. The combined possibility of measuring power consumption and controlling state on sockets provides a foundation for seemingly endless possibilities. Even though The Smarter Power Strip offers some services through its web application, there are many potential features that it does not offer, and providing third parties the ability to access the measurement and controlling function of power strips therefore opens up for additional services to be added, making the product more attractive.

Additionally, the situation of The Smarter Power Strip is quite unique given that while the prototype will be planned and developed, a similar project will be conducted in parallel, developing a similar prototype. The ability to offer our services to the other project, and vice versa, would improve the outcome of both projects.

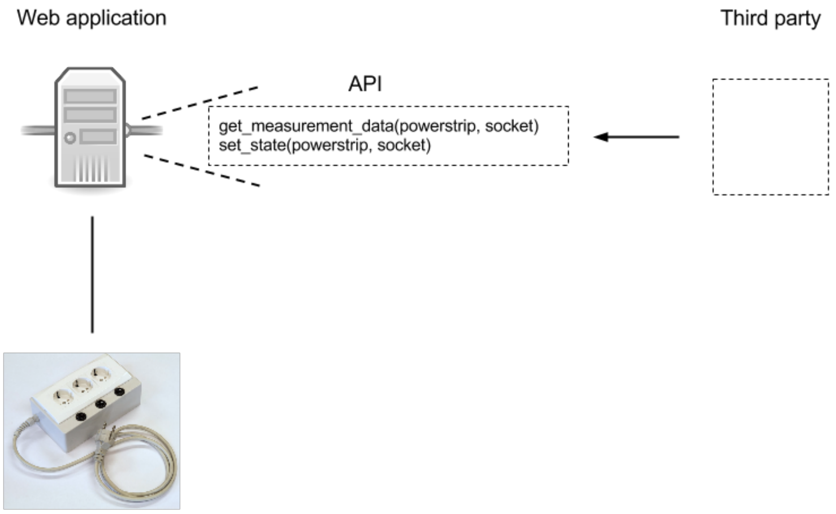


Figure 15: Web application level API. A third party can access the power strip's services through the web application logic.

Software interoperability is commonly provided by a so called *application programming interface* (API). There are different levels at which an API could be located. One way would be to expose the functionality embedded in the power strip, providing a set of methods implemented on the microcontroller's software for reading measurement data and controlling states (see Figure 14). This has been done in prior projects similar to The Smarter Power Strip with good results (Jiang et al. 2009). A power strip level API would give third parties great freedom and flexibility in how to use The Smarter Power Strip. For example, in this case anyone could create an own web application or possibly mobile application that could access the power strips.

An API could also be provided on the web application level (see Figure 15). This would give a third party access to the logic defined in the web application. An API on this level would mean less freedom for a third party since it would be forced to communicate with the power strips in the way that is defined in the web application. However, in the case of The Smarter Power Strip, this would in many ways be simpler. This, since there are many basic features, such as connecting a certain power strip to a certain user, that most third parties are not interested in implementing. A web application level API is also something that, with time constraints in mind, will be easier for us to provide since there are web framework libraries that offers this service (Django Packages 2013).

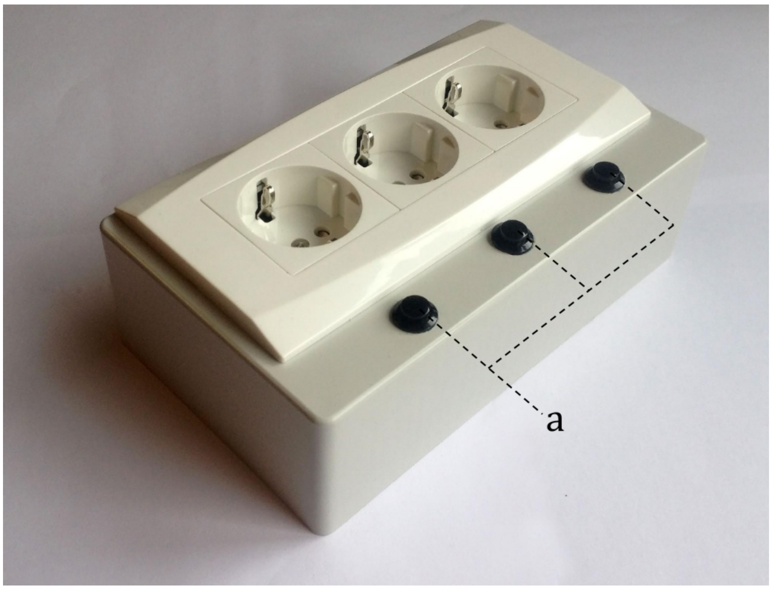


Figure 16: The prototype. a) shows momentary buttons.

5 Prototype design

In this chapter, we present the design of The Smarter Power Strip. We describe what solutions, of which some were introduced in chapter 4, were chosen to realize the purpose of the project, and why. The prototype can be viewed in Figure 16.

5.1 Hardware design

Below, the hardware solutions for measurement and output toggling are presented. The design of circuit board can be viewed in Figure 19 and a block diagram of the low voltage circuit is shown in figure 17.

5.1.1 Measurement

The voltage was measured with a transformer which was connected in parallel with the loads and the embedded circuit. By doing this instead of measuring the voltage from the transformer which supplied the embedded circuit, the measured voltage was affected by the current drawn from the embedded circuit. The setup of the measurement circuit was based on a current transformer. The current in the cable to the load induced a current in the secondary coil on the transformer. By connecting a high precision resistor and measuring the voltage drop, the current into the load could be calculated. To achieve a robust system the burden resistor current in the secondary circuit needed to be large to avoid induced noise. Therefore, we

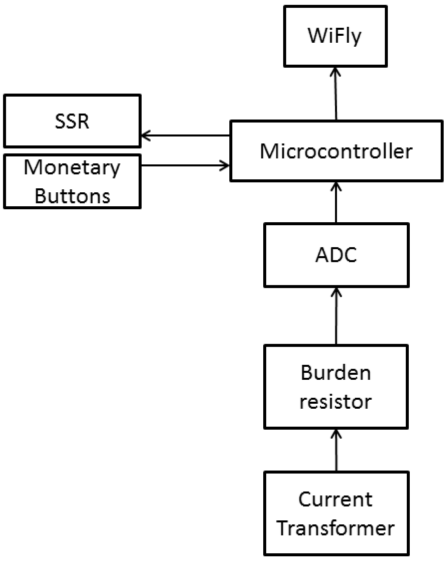


Figure 17: A block diagram of the low voltage circuit

dimensioned the circuit to operate with as high current as the burden resistor allowed.

To be able to measure effects on 1 W the rectifying circuit had to be able to handle signals as weak as $2.8mV_{p-p}$ peak-to-peak, which after the rectifier only was $1.4mV_{p-p}$. This was very weak signals, and to come as close as possible to this, high precision operational amplifiers were used. Even though the performance was not enough to satisfy the conditions in the specification (see section A.1), it was not possible to measure signals weaker than $20mV_{p-p}$, and they had to be above $50mV_{p-p}$ to have acceptable shape. This problem was solved with the switchable measurement interval described below.

The ADC measured voltages between 0 and $3.2V$. Since a rectifier was used, the maximum peak-to-peak value over the burden resistor was $6.4V$. The standard through hole resistors which was used allowed a maximum effect of $250mW$(*Metal Film Resistors, Precision Type [MFP Series]* 2013). The current ratio was 1:250. With a load current of $10A$ the secondary current was $40mA$, and to give a $6.4V$ voltage drop the resistance had to be 160Ω.

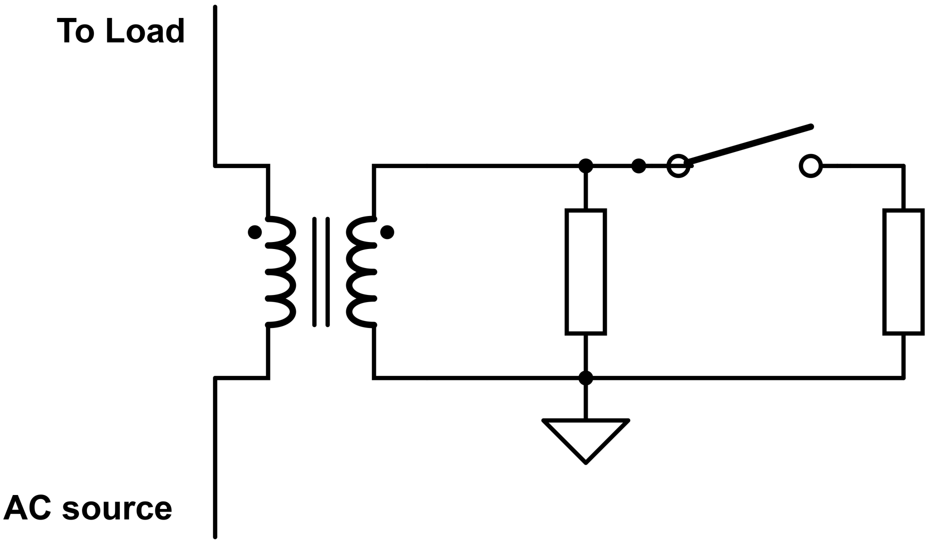


Figure 18: Measurement circuit with switchable burden resistance

To be able to both measure low and high currents accurately, a switchable resistance was implemented. By letting the microprocessor control a switch, which connected a resistor in parallel, resulting in a lower total resistance. The current to voltage ratio could then be changed making weak currents resulting in a higher voltage. This was important since the "Ideal rectifier" was not ideal. In the setup a $4.7k\Omega$ was connected in parallel with a switchable 150Ω resistance. The circuit is shown in figure 18

The microprocessor was equipped with internal ADCs with a 10bit resolution. This could be enough if a switchable measurement was implemented and, also when worse accuracy is tolerated in the higher measurement intervals. However to better evaluate the overall performance of the system external 16bit ADCs were used. To accurately measure the 50Hz signal it was possible to sample with 100Hz and do a FFT of the signal. A simpler way when only the active power is needed is to in every measurement instant multiplying voltage and current. To get better accuracy with this method a higher sampling rate is needed compared to the FFT method.

5.1.2 Output toggling

Durability and low power consumption were the main priorities when it came to techniques for output toggling. SSRs offered a price which was much lower compared to the latching relay, and even though they cost more than conventional EMRs, the EMRs have higher power consumption. The compact design of SSRs is also a advantage making it easier to keep the

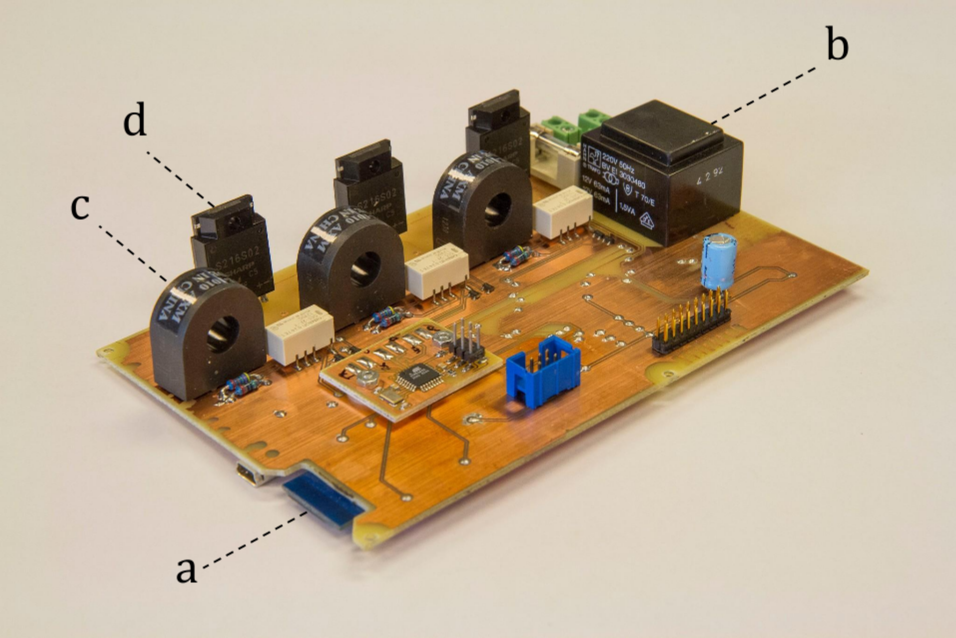


Figure 19: The circuit board of the power strip. a) shows the Wifi module, b) the transformer, c) a current transformer and d) a SSR. ADCs and micro processor are located on the other side of the board.

design compact. Since of this, we chose to us SSR relays.

5.2 Communication

Deciding on the communication solution in The Smarter Power Strip was an important process, since so many of the other solutions depended upon this. What had to be provided was Internet based communication between power strip and web application, which comprised solutions regarding what *communication pattern* should be used, how the *layering* between entities should be structured and how the *content and structure* of messages should look like. The communication should also be able to connect certain power strips to certain users. In addition, the prototype had to provide an alternative communication channel through which it could be configured.

5.2.1 Polling

The communication pattern chosen for The Smarter Power Strip was polling. Polls were to be sent once every second from the power strip to the web application. Polling was mainly chosen as a consequence of the presence of NAT routers on the Internet (see chapter 4.4.1). An assumption made while making this decision was that most of our potential users obtain their Internet access via a NAT router. If this was not the case, two way messaging would have been a stronger candidate. However, since routers with NAT block unknown traffic, messages originating from the web application would never be able to reach the power strip, unless the user explicitly configured his home router not to do so.

Thus, if we wanted it to be possible for the web application to send messages to the power strip, as is necessary in the case of two-way-messaging, we would have to require from the users of our product to configure their routers before starting to use it. However, one of the purposes of the product was that it should be easy to use and target a broad group of possible users. Since configuring routers is something that many people would find complicated, this was something we wanted to avoid.

The decision was also based on the fact that if two-way-messaging would have been chosen, the power strip would always have to have its Wifi module active, ready to receive messages from the web application. An active Wifi module means a higher power draw in the power strip and since the main purpose of The Smarter Power Strip is to lower a user's power consumption this was a consequence we found undesirable. Polling on the contrary made it possible for the Wifi module to sleep while not sending messages to the web application, saving power.

Choosing this solution meant that messages generated in the web application, i.e. commands for changing the output state of the power strip's sockets, did not reach the power strip instantly. This was an undesirable effect and made the remote controlling feature of The Smarter Power Strip perform worse. However, in comparison with higher power draw and making users configure their routers, it was a drawback we were willing to accept.

Choosing polling also led to that a lot of the messages sent from the power strip to the web application did not ultimately result in a change of a sockets' state. Sending messages cost power and bandwidth, and should therefore, if possible, be avoided. Since the messages sent were so small and few, on the user side, the power consumption and network stress were not significant. However, on the server side, if The Smarter Power Strip would gain a lot of users, an extensive amount of small messages could be an issue. Since we chose to communicate via HTTP (see chapter 5.2.2), though, which is a protocol that has proven to be able to handle the scalability issue, this could be handled.

5.2.2 Communicating via HTTP

The communication structure between entities in The Smarter Power Strip eventually took the following form: the web application and the power strip communicated directly over the HTTP protocol. Reading and writing of data generated by these two entities to the database was handled by the web application, which was in direct contact with it through the web framework. Thus, when the power strip wanted to poll for changes in its state or upload measurement data it sent an HTTP-request with that data attached to the web application which forwarded the data to the database.

The reason this method was chosen was because it provided a possibility to get started easily. Since the Wifi module in the power strip supported HTTP, and since Django had built in methods for handling HTTP-request and reading and writing data to a database, there was already a communication channel established. Hence, since of this choice we did not have to find or write own software and protocols that could handle communication and reading and writing of data between web application and database server, and power strip and database server.

There was also an incentive to use HTTP instead of defining own protocols since HTTP has been used by many other websites. The protocol has been tested thoroughly in different situations and has been proven to work. The web application and the communication structure as a whole had to be scalable; it will start by handling a small volume of power strips, but it should have the possibility to handle a large number communicating concurrently. We know that this feature was provided by HTTP. If we would have defined our own software and protocol, though, there would have been a risk that it would not work properly when volumes of concurrently communicating power strips were to be handled.

However, a problem that followed from communicating over HTTP was that it put overhead on the communication. HTTP is a protocol that was originally designed for communication between web client and web server. To handle this kind of communication, every HTTP-message contains meta data describing for example which browser the client uses, what language the client would preferred the answer was sent in, and more (Berners-Lee, Fielding, and Frystyk 1996; Fielding et al. 1999). This meta data is needed when the purpose of the main data sent is to be presented at the web client in a web browser. Since the power strip, which had the client role in the communication conducted in The Smarter Power Strip, did not present the data it requested in a browser, but rather processed it in other ways, the HTTP-message meta data had no purpose (overhead). If we had instead defined an own protocol, the overhead could have been decreased.

As a consequence of the overhead, more data than what was needed for the communication to function was sent between the web application and the power strip. This meant that the Wifi module had to work unnecessarily much, draining power, and the communication was slower than it had to be since the messages sent were bigger than they needed to. There was also an unnecessarily high pressure on the network since it had to handle a lot of data that did not really contribute to the communications purpose: transporting output control commands and measurement data.

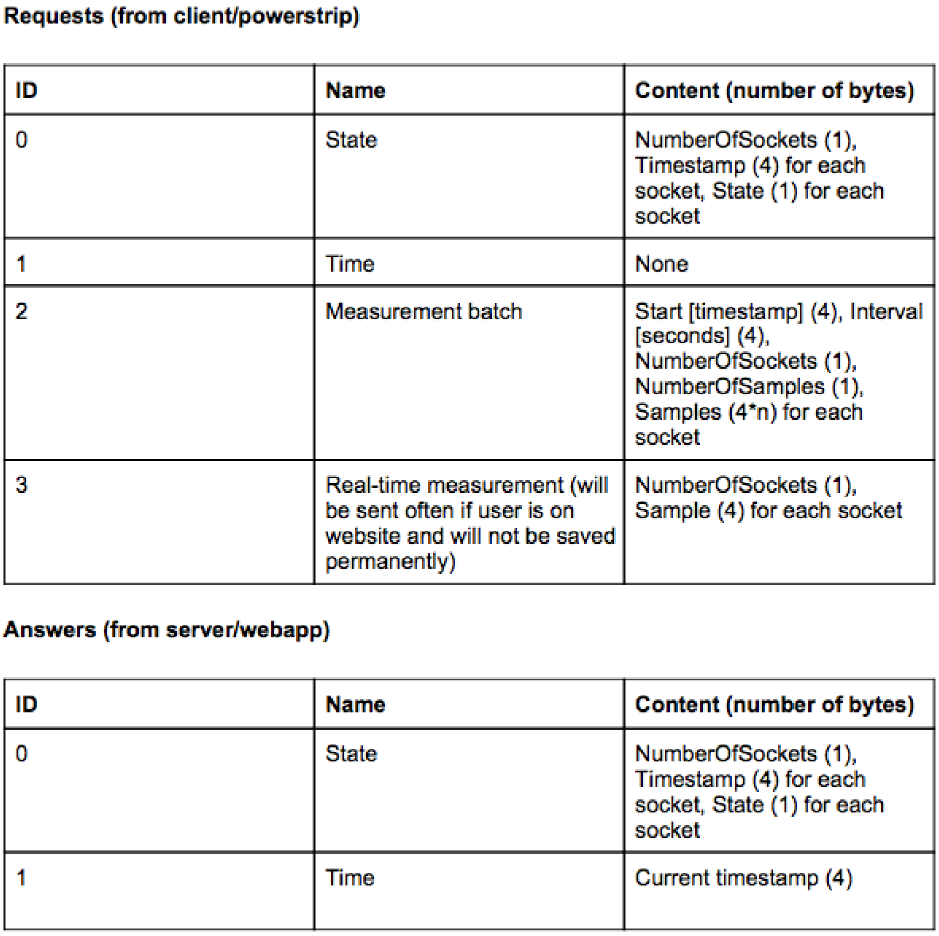
However, since there was not very much communication conducted in The Smarter Power Strip, the overhead did not affect the overall performance on a very high degree. The extra pressure on the network and the server was not so severe that it should be considered as a significant issue. The extra power drain on the Wifi module in a way worked against our goal that the product should help a user lower his power consumption, but as a consequence of choosing HTTP instead of defining an own protocol, it was something we were willing to accept.

5.2.3 Message content and structure

In The Smarter Power Strip, the idea was that the power strip should send measurement data to the web application and the web application output state control commands to the power strip. In order for the software at the two entities to be able to process the data received, there had to be a predefined structure on it. For instance, receiving a message with measurement data, the web application had to know what segment of bits in the message that described measurement data from a specific socket.

Content
As a first step in defining this message structure, it had to be specified in detail what content the messages should contain. That the power strip communicates measurement data and the web application output state control commands is actually a simplification, and in order for the communication to provide the required functionality, messages sent between the two entities needed to contain more information. A detailed description of the message content that was needed is viewed in Figure 20.

We chose to call the messages sent from the power strip *requests* and the messages from the web application *answers*. Several requests and answers with different content and responsibilities were then defined. A request could either be a *state request*, a *time request*, a *standard measurement batch* or a *real time measurement batch*. Their respective content and responsibility were:



Requests (from client/powerstrip)

ID	Name	Content (number of bytes)
0	State	NumberOfSockets (1), Timestamp (4) for each socket, State (1) for each socket
1	Time	None
2	Measurement batch	Start [timestamp] (4), Interval [seconds] (4), NumberOfSockets (1), NumberOfSamples (1), Samples (4*n) for each socket
3	Real-time measurement (will be sent often if user is on website and will not be saved permanently)	NumberOfSockets (1), Sample (4) for each socket

Answers (from server/webapp)

ID	Name	Content (number of bytes)
0	State	NumberOfSockets (1), Timestamp (4) for each socket, State (1) for each socket
1	Time	Current timestamp (4)

Figure 20: The different message types and their content.

- State request - Contained information on the number of sockets of the sending power strip, and what states these were in. For each socket there was also a time stamp on when the socket was changed. This content was needed to synchronize the state stored on the power strip with the state stored on the web application (see chapter 5.4).

- Time request - In order to keep time on the power strip and the web application synchronized, a time request had to be sent at an even interval. This would be answered by a message containing the time on the web application.

- Standard measurement batch - Contained the actual measurement data, as well as data about the time when the measurements were done, the number of samples that were used while measuring and the number of sockets that were measured upon.

- Real time measurement batch - Contained measurement data and the number of sockets measured upon. This request had the purpose of delivering data that should not be stored at the web application side, why it did not need time stamps.

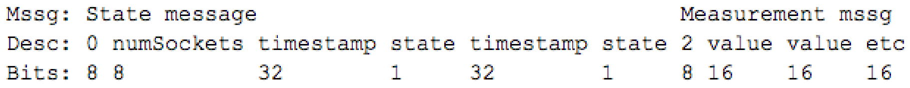
An answer on the other hand could be either a *state answer* or a *time answer*. Their respective content and responsibility were:

- State answer - Contained the state of sockets' as it was stored on the web application, as well as time stamps on when these states were changed.

- Time answer - Contained the current time according to the web applications clock.

Structure

Since an HTTP-message could carry several requests or answers at the time, the next step in specifying the messages was to organize different requests and answers into messages in an effective way. The fact that polling was used meant that we needed to send state requests at even intervals. We also wanted to update the web application with new measurement data ongoing. To achieve this, these two requests could be concatenated into one message. A schema over the result is presented in Figure 21. This message could be answered from the web application with a state answer. The time request and answer, however, did not have to be sent in as high frequency as state and measurement message. Thus, these were not concatenated with the other messages.



```
Mssg: State message                                  Measurement mssg
Desc: 0 numSockets timestamp state timestamp state 2 value value etc
Bits: 8 8          32        1     32        1     8 16    16    16
```

Figure 21: The structure of a concatenated state and measurement request.

5.2.4 Providing an alternative communication channel

In order for the power strip and the web application to be able to communicate with each other, the power strip needed Internet connectivity. To achieve this, it had to be connected to a user's wireless network. However, to be able to access this network, credentials regarding the user's home router were required. Consequently, the power strip needed to provide an interface through which it could be given this information, before it had gained Internet connectivity.

In The Smart Power Strip, this was solved by giving the power strip the possibility to act like a web server, offering connection to a web application located on it. The power strip would, if a certain button on it was pushed, create a wireless ad-hoc network. This would allow another wireless device to set up a connection between them. Then, if sending an HTTP-request to the power strip from this device, the power strip would return a website through which it could be configured. The user would then be able to provide the power strip with the credentials for the home router.

5.2.5 Associating specific users with specific power strips

The Smart Power Strip was organized with a single web application accessed by multiple users, all with the intention of interacting with their own power strips. Hence, the web application had to make sure that different users could access different power strips concurrently, and that only the user who owned a certain power strip could access it. For this to be solved, user and power strip had to in some way be able to identify themselves to the web application.

Identifying a user interacting with the web application is a common task with a simple solution: when the user wanted to use the web application he had to log in with a unique user name. This gave him access to a selected set of power strips. The task of identifying the power strips, however, did not have an equally obvious solution. We chose a way that had been used in prior projects to The Smarter Power Strips (Jiang et al. 2009), that is, labeling every power strip with a key. When a user started using a power strip, he would have to provide the key on the label to the web application, which could then tie together that particular power strip with that user.

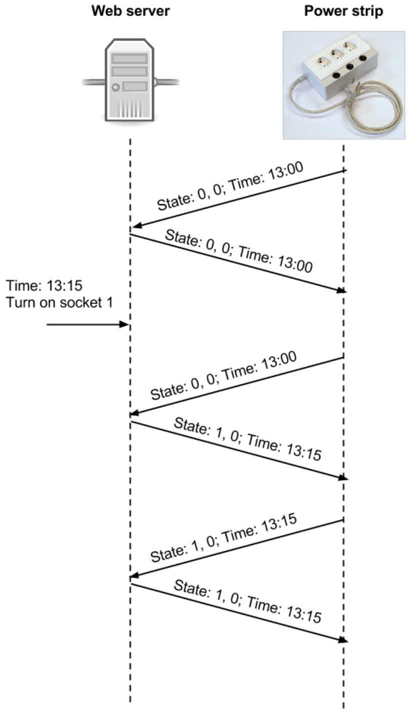


Figure 22: The web application receives a state from the power strip that differs from its own. Since the time stamp on the received state is older than the one on the own, the own state *should not* be changed.

5.3 Security

If an unauthorized person would get access to a user's power strip connection, he would be able to control that user's appliances and view his consumption habits. Hence, the security issue was of great concern in The Smarter Power Strip. However, due to the time constraints of the project, no solution was eventually implemented. What could and should have been done if more time was given is discussed in chapter 7.3.

5.4 Socket state storage

The software model describing the state of a power strip's sockets was in The Smarter Power Strip placed both on the web application and on the power strip. This choice was made so that the communication pattern polling could be used, and so that momentary button changes would not have a slow response time (see chapter 4.6).

Having the state at two different places meant something had to be done to ensure that both states were synchronized with each other. To solve this,

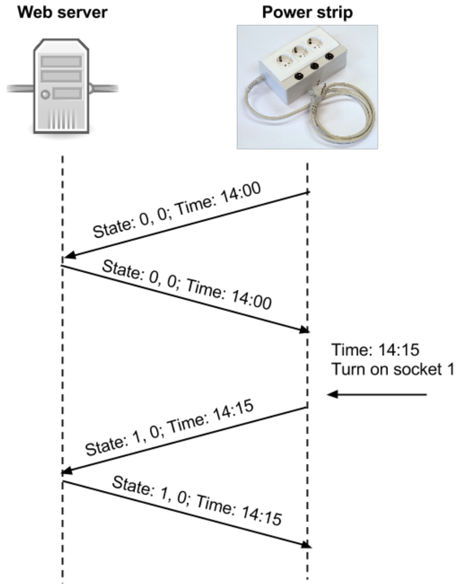


Figure 23: The web application receives a state from the power strip that differs from its own. Since the time stamp on the received state is newer than the one on the own, the own state *should* be changed.

the states of sockets were sent to the web application at even intervals. The web application answered with its version of the sockets' state.

It was not always the case that states received in a message differing from the own state was newer than the own. This had to be possible to derive, since if that was the case, the own state should not be modified. For instance, when a user issued a state change on the web application, the next poll received from the power strip would contain a different, older state than the web application's (see Figure 22). In this case, the web application should not change its state, but reply to the power strip with the state containing the change of the user.

To solve this issue, when a state was changed a time stamp containing what time the change was made also had to be stored. In this way, if a state that differed from the own arrived at the web application, it could compare the own state's time stamp with the one in the message. If the own state was changed earlier than the one arriving, it would know that there had been a state change issued by the momentary buttons, and that it should update the own state. If the state from the power strip was older, however, it would know that the difference was due to a change issued by it. Therefore, it should not change the own state.

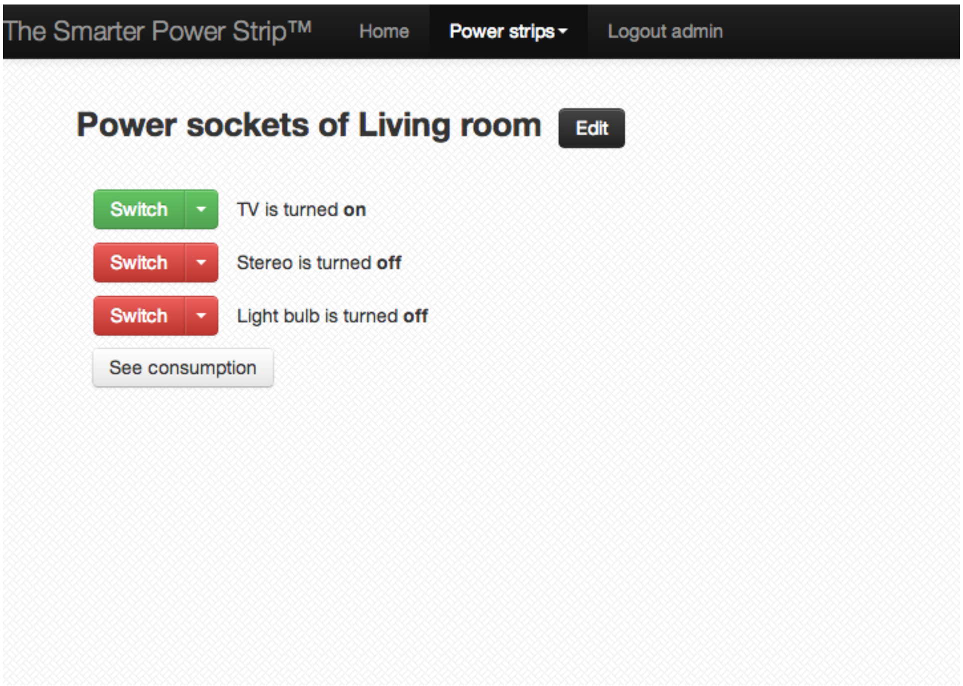


Figure 24: A screenshot depicting the "Power strip detail" view. Through toggling the buttons the user could control the output state of a socket on one of his power strips.

At the power strip side, the issue was not as severe. It only occurred if a momentary button change was made in the time between when a poll was sent and when the answer from the web application was received, which was a time window on at most five seconds (see response time, section A.1). As of that, we decided to not implement the time stamp check on the power strip software.

5.5 Web Application

The task of storing data and providing an interface through which users could access their power strips was in The Smarter Power Strip handled by a central web application. We chose to develop the application with the Python based web framework *Django* (Django 2013). The choice was made essentially upon the fact that Django provides functionality for interacting with a database without having to write any SQL code, as would have been the case if using for example PHP. This made developing the software a less complicated task.

The web application contained logic for receiving, storing and presenting

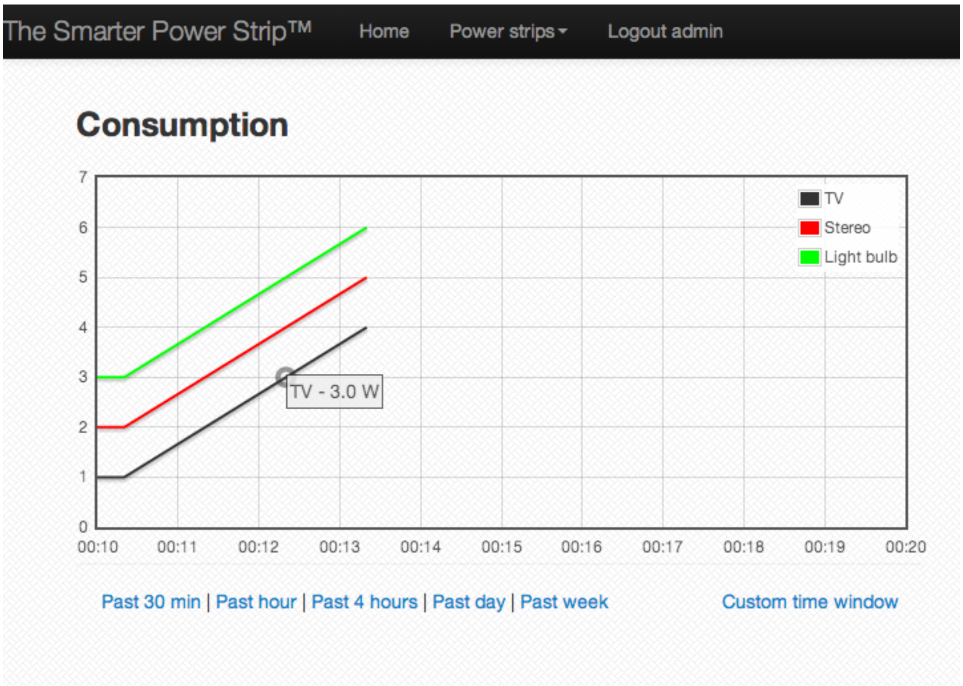


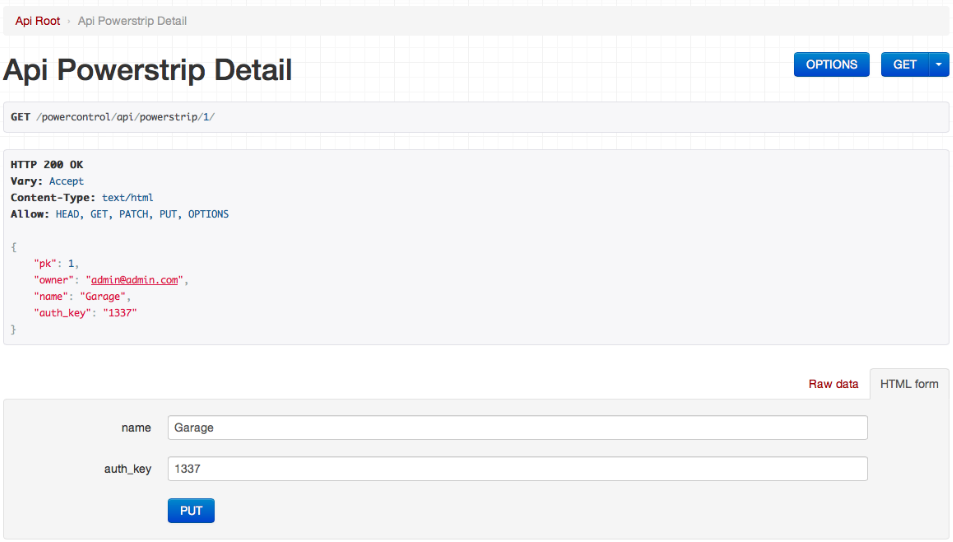
Figure 25: A screenshot depicting the "Power strip consumption" view. The user had the possibility to view the consumption for different periods.

measurement data. It could also generate and send output state control commands to the power strip. Users could access and control these functionalities through a website. An API through which third parties could access the functionalities was also provided.

5.5.1 Website

The Smarter Power Strip provided a website through which users could control their power strips and view power consumption data. How the website presented these two main features graphically can be viewed in Figure 24 and 25. Additional features that were also provided were the possibility to label power strips and sockets in order for the user to remember for instance what power strip was in what room and what appliance was connected to what socket, and the ability to log in, in order to connect an owner with his particular power strips.

The purpose of The Smarter Power Strip states that the prototype should be easy to use and target a broad group of possible users. This was the main goal aimed at when developing the website. Hence, it was given a graphical design that was as simple and unambiguous as possible. A focus on simplicity however, did not mean that aesthetic factors were totally disregarded.



Api Powerstrip Detail

```
GET /powercontrol/api/powerstrip/1/
```

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html
Allow: HEAD, GET, PATCH, PUT, OPTIONS

{
    "pk": 1,
    "owner": "admin@admin.com",
    "name": "Garage",
    "auth_key": "1337"
}
```

name: Garage

auth_key: 1337

Figure 26: A view of the API's browsable interface.

These factors are important in order to provide users a good user-experience, i.e. making it attractive and easy to use the prototype (Tidwell 2011). Some efforts on making the website look good were therefore made. For instance, the Twitter Bootstrap (Twitter Bootstrap 2013) framework was integrated, giving the website a pleasant look.

5.5.2 API

In order to provide the possibility for third parties to interact with The Smarter Power Strip, and to use power strips and their functionalities for own applications, we chose to implement a web application level API (see Figure 26). The choice was web application level and not power strip level since this was considered simpler; both for the third parties interacting with our product, since it would give them access to the basic communication functionalities implemented in the web application, but also for us, since there were already implemented Django-libraries available on the Internet for creating a web application API. Developing an own power strip level API would have had to be done from scratch. With the time constraints in mind, it would have been a bad choice.

The API creation library chosen was the *Django REST framework* (Django REST framework 2013). This choice was made upon the fact that it was one of the most popular libraries for API creation (Django Packages 2013), which implies high quality, and that it unlike other popular alternatives, such as *Piston* and *Django-Tastypie*, provided out-of-the-box support for

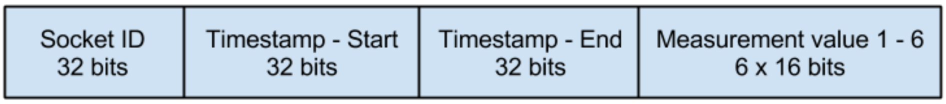


Figure 27: The measurement data batch. A batch contained six measurement values, the time of when the first and the last measurement were done, and an id specifying to which socket the measurement values applied.

OAuth2, which was the framework we chose for API authorization.

5.5.3 Storing and presenting data

In The Smarter Power Strip, measurement data was sent from the power strip to the web application where it was presented in graphical diagrams. The specification of the prototype (see section A.1) stated that it should have the possibility to present a measurement value of the consumption every tenth second.

It should also be possible to present historical data, which meant all measurement data had to be stored in a database. Thus, how stored data should be structured had to be defined. When a user wanted to view measurement data, the web application needed to retrieve the data regarding the time specified by the user from the database. Hence, all data stored could not only contain the actual consumption values, but also the time at which the values were measured. It should also be possible for users to see which socket was measured upon. Accordingly, every consumption measurement value had to be accompanied by information identifying which socket the measurement applied to.

A measurement value required 16 bits of memory, a time stamp 32 bits, and a socket identification number 32 bits. Given that a measurement value for every tenth second should be stored, the memory required for storing a user owning a three-socket power strips' consumption over a year would be 754 megabytes. Applying this to a large number of users owning multiple power strips resulted in quite a large amount of data, which had to be reduced.

Also, every measurement value should be stored in a database, becoming its own database entry. If the user wanted to present one week's power consumption for one socket, 60480 entries would have to be read from the database. Database lookups take time, and since we wanted the presentation of data to be as fast as possible, this was also a problem.

In order to solve these two issues, we decided to structure the measurement information in what we called *batches*. Every batch included several measurement values, and instead of storing socket identification and time stamp for every measurement value, every batch only needed to store information regarding which socket the entire batch applied to, and the time stamp for the first and last measurement. This lead to a specification on a batch structure that is presented in Figure 27. The batch included six measurement values, i.e. power consumed over a minute. This batch structure both lead to less memory and less database entries used.

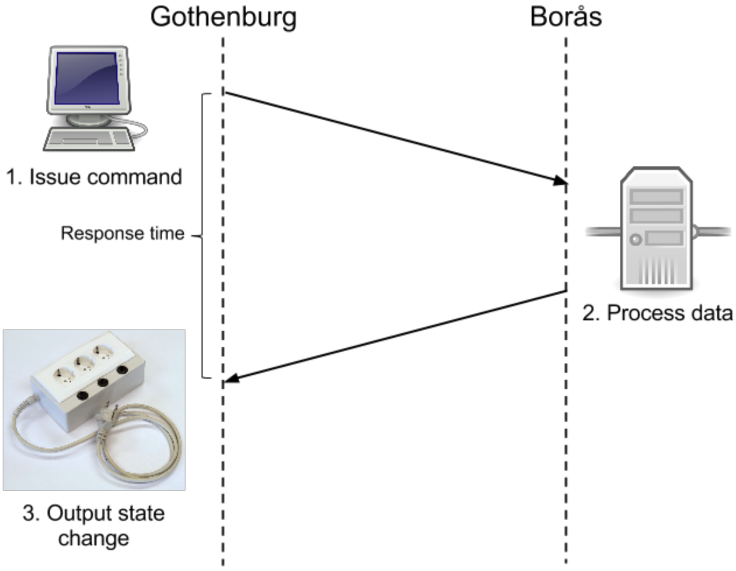


Figure 28: The response time was tested by issuing state change commands on a host in the town of Gothenburg to the prototype, which was located in the same town, and with the web application server located in the town of Borås, 60 kilometers away.

6 Evaluation

In this chapter, we evaluate how well the solutions implemented, which were described in chapter 5, actually performed in order to realize the purpose of The Smart Power Strip. We present empirical tests where it is analyzed whether the prototype fulfills the goals set up in the specification (see section A.1).

When evaluating the hardware of The Smarter Power Strip (see section 6.2 and 6.3), an uncalibrated *Fluke 8808A* multimeter was used. When doing the measurements a stable net voltage was assumed, and the voltage was measured to 234V on the connecting pins to the power strip.

6.1 Response time

In the context of The Smarter Power Strip, the change of a socket's output state could not be conducted instantly on the power strip as such a control command was issued on the web application. A delay on the power strip's response time was inevitable, since the command needed to be sent over the Internet. However, it was important to keep this response time as low as possible. The goal for The Smarter Power Strip was that it should not be greater than five seconds (see section A.1).

In order to evaluate if this goal was fulfilled, we designed a test where the time from issuing a command on the website, until the actual output state change happening on the power strip was measured(see Figure 28). During the test, the command was issued on a host in the town of Gothenburg. The prototype was also located in Gothenburg, while the server with the web application was placed in the town of Borås, 60 kilometers away. Issuing ten commands, the test resulted in a mean value for the response time of 1.3 seconds. From the observed response times seen on the server the worst case would be approximately 4 seconds.

Regarding the polling frequency, a test showed that the prototype did 100 polls in 97 seconds, resulting in a mean time of one second before the database state was propagated to the prototype. Consequently, the goals were fulfilled.

6.2 Power measurement

Measuring power consumption was one of the main features provided by the prototype. Thus, it was very important that the properties of the specification regarding measurement (see section A.1) were fulfilled. If the prototype could not provide relevant measurement data, much of its value and meaning would be lost. To be able to calculate the power consumption correctly, accurate measurement values of both the voltage and the current were needed.

6.2.1 Voltage measurement

The measured voltage the power strip used resulted in approximately $210V_{RMS}$ for all the measurements. This was mainly due to a calibration error where the voltage peak-to-peak values where used to set the correct offset scaling. However, since the sampling frequency was very low, the peak value was not measured and therefore not used for the calibration. The measured RMS should be compared to the RMS measured with the multimeter.

6.2.2 Current measurement

To evaluate the current measurement, three tests were done. One without load to measure the noise floor, one with a $2W$ load and one with a $110W$ water boiler connected to the power strip. The noise floor measurement test was done with the SSRs closed. The noise floor was $0.45\mu A$, which resulted in a noise floor effect of $89.7mW$. This was well below the specified minimum of $2W$. Even though the low sampling frequency resulted in an inaccurate value, it was still low enough to guarantee that the specified minimum load was measurable. When evaluating the minimum load, two high effect resistors connected in series with a measured total resistance of $25.7k\Omega$ were used. The current measured by the power strip were $2.89mA_{RMS}$, which was below the actual $9.1A_{RMS}$ drawn by the resistors. The resulting effect was $480mW$. Lastly, the test with the water boiler showed a drawn current of $4.21A_{RMS}$. The resulting effect was 875W. The current measured with the external multimeter showed a $4.38A_{RMS}$ current and the consumed effect was 1025W.

These results show that the specifications regarding minimum load were met. The power strip had no problem measuring large loads, even though no load of $2300W$ was tested. The test of the water boiler shows that The Smarter Power Strip can measure large loads and there is no indications that there would be any problems with loads up to $2300W$. The resolution of $2W$ should also be okay, but the measured values were inaccurate. This inaccuracy mainly derives from two sources. The voltage was calibrated on the peak-to-peak values, however the low sampling interval resulted in measured peak values which were much lower than the real peak values. This should have been calibrated with the use of RMS values. The RMS values will also be inaccurate, although not as much as the peak-to-peak values, due to the low sampling interval.

6.3 Standby effect

The idea with The Smarter Power Strip is to help users lower their power consumption. It was therefor very important that the power strip itself consumed little energy. The goal was set on no more than $3W$ (see section A.1). To evaluate this, a multimeter was connected in series with the power strip on the high-voltage side of the transformer. The power strip was connected to a Wifi network and was polling the web server for state changes. No measurements were being done by the power strip during this test. However the increased power consumption due to measurement is negligible since the Wifi module is the dominant power consumer. A current of $11.1mA_{RMS}$ where measured which resulted in a standby consumption of $2.59W$, which is under the specified $3W$.

7 Discussion

This project set out to develop a prototype that should provide the ability to measure power consumption data and to remotely toggle sockets on and off. As is presented in chapter 5, the final prototype succeeded in achieving both of these goals. There was also a specification (see chapter A.1) in which properties that should hold for the prototype were defined. Empirical tests described in chapter 6 showed that most of these were achieved, but that there were some issues regarding the power measurement accuracy (see section 6.2). Also, some design solutions used developing the prototype contained flaws, and some important features were, due to time constraints, not implemented. Hence, in this chapter we will discuss what could have been done better in the prototype's design.

7.1 Power measurement

The evaluation tests indicated a few problems. The most severe was the low sampling rate. By using other ADCs with sampling frequency of at least 8 kHz, or even better by performing an FFT on the signal, the signal could be correctly measured or correctly recreated. The calibration error of the voltage could be corrected with more measurements. The evaluation showed a distinct elevation of the measurement signal for the $2W$ load, and even if this is not accurately measured, calibration and signal processing should make it possible to measure it accurately. The same applies to the larger loads from the water boiler. It is therefore assumed that with more development of the software on the microprocessor the power strip would meet its specifications.

7.2 Hardware improvements

The hardware in the final prototype worked well but there are still room for improvements. First of all, the full wave rectifying circuit used was complicated and expensive. Even though it would be possible to use the same circuit with cheaper components thanks to the switchable measurement interval this is still a complicated solution. A simpler way would instead be to bias the signal, that is, add a DC-component to the signal. In this way the resolution for large loads would be coarser but it would still be possible to keep a resolution below 1W for weaker loads, and it is for the weaker loads the resolution is important. Since it is possible to use the switchable measurement to get better resolution for weak loads we do not see the need for external high resolution analog-to-digital converters either. The integrated 10bit ADC are good enough.

7.3 Security

In the beginning of this project, it was decided that a robust security solution was needed, since access to a user's communication channel between the web application and his power strip could give a third party control over that user's electrical appliances, and information about his living habits. However, due to time constraints, no security solution were eventually implemented in the prototype. This is something that must be implemented before The Smarter Power Strip can become a commercial product.

A dominating web security solution for common desktop computers is HTTPS (see section 3.7.2). However, this solution was not suitable for our power strip because of its resource-intensity. While researching, the most suitable HTTPS-library we found needed up to 36 kilobytes of Random Access Memory (RAM) and 100 kilobytes of flash memory (CyaSSL 2013), while the microprocessor in the power strip only had 4 kilobytes of RAM and 64 kilobytes of flash memory (Atmel 2013). Clearly, it would not have worked on the prototype.

To implement a security solution, there are three possible alternatives. The first would be to upgrade the hardware in the prototype so that it supports HTTPS. This upgrade could either be done by using a microprocessor with more RAM and flash memory, allowing the HTTPS-library to be used, or by using a different Wifi module that has built-in support for HTTPS. The drawback of upgrading the hardware would be slightly higher power consumption and cost.

The second alternative would be to implement a basic security solution that used symmetric key encryption, such as AES. In that case, less memory would be needed (see section 3.7.1) and the current hardware could be used. The biggest problem with only using AES, however, is that it is strictly inferior to HTTPS. This, since AES only provides confidentiality, whereas HTTPS also provides message integrity and end-point authentication, which protects against other kinds of attacks.

The third alternative would be to do a custom solution by using AES as a foundation and expanding upon it, adding Message Authentication Code (MAC) to provide message integrity and adding public key encryption to provide end-point authentication as well as allowing automatic updating of AES keys every day. This would, however, be a significant undertaking and it would be easy to make a mistake that would make the communication vulnerable. This alternative would not provide sufficient security to protect users of The Smarter Power Strip.

7.4 Response time

A choice that was made designing The Smarter Power Strip was to use the communication pattern polling instead of two-way-messaging. Since instant communication is not possible using polling (see chapter 4.4.1), this decision resulted in a higher response time, i.e. time needed for the power strip to react after issuing a command on the web application, than what had been the case using two-way-messaging.

To improve the response time, two alternative solutions could therefore have been chosen. One would be to implement two-way-messaging instead of polling, which, however, would mean that other problems, such as getting messages to bypass routers with NAT (see chapter 3.6.1), would have to be solved. The other alternative would be to implement increased polling frequency as a user logs in on the web page. These solutions are presented in detail in chapter 4.4.1.

7.5 Initial configuration

A problem that occurred while performing the initial configuration (see chapter 5.2.4) on the power strip was that it took an undesirably long time for the device and the accessing host from which configuration was carried out to set up a connection between each other. When powered on for the first time, the power strip set up a wireless network through which hosts in the vicinity could contact it. Hosts that connected to that network would need an IP-address in order to communicate with the power strip. IP-addresses are often distributed automatically, a task that is handled by the *Dynamic Host Configuration Protocol* (DHCP; Droms 1997) which the power strip however did not support.

The host, which was set up for DHCP, followed the standard procedure, and sent a request for an IP-address to the power strip. Since the power strip did not act as a DHCP server, nothing happened. This was the source of the time delay in the connection setup. What eventually happened was that a timeout occurred on the host, whereafter it set an IP-address on own initiative. After this, the host was able to send messages to the power strip, and the power strip was able to answer.

The solution to this problem would therefore be to implement a DHCP server on the power strip. At the time of writing, a firmware update just released for the Wifi module actually enables the DHCP server functionality so that this problem can be addressed in the future.

7.6 Sending measurement data

One of the main features of The Smarter Power Strip was that it should be able to present a user's power consumption data. As is described in chapter 5, several sub parts of this functionality, such as the measurement on the power strip (section 5.1.1), the ability to receive, store and present measurement data on the web application (section 5.5), and the definition of the structure of the messages (section 5.2.3), were implemented. However, due to time constraints some of the software on the power strip responsible for sending the measurement data was not implemented. Consequently, an important link in this communication was missing. What would have to be done to solve this problem is simply to spend some more time on developing the power strip's software.

7.7 Cost

The total cost of components used in The Smarter Power Strip were approximately 1300 SEK, which is very high, and given that additional costs, such as for assembly and development, would have to be added, the price definitely would not be competitive compared to what is on the market today (see chapter 2.2). The idea with The Smarter Power Strip was to cover almost all appliances in the home, and to be able to offer a competitive solution, cost is the key. When working with this prototype, though, we focused on which techniques to use, and tried to see how well the device performed. To evaluate this properly components with high performance and precision were used. However, the fact is, as stated in 7.2 several components could either be exchanged to cheaper alternatives or even completely removed. There are also cheaper Wifi modules coming on the market.

These changes would make the cost for components drop to approximately 450 SEK. This is still a bit high, and to lower the cost even more, other techniques could be implemented. For example, the use of shunt resistors would make it possible to reduce the cost by 150 SEK. This would compromise the galvanic isolation the system has today and probably introduce more noise. It would also couple the ground to a very high potential. It would also require expensive relays that could switch the total current of the load for the switchable burden resistors. An alternative could be to use variable amplifiers to create a switchable or dynamic measurement interval. This would also amplify the noise, and the impact of the noise needs to be evaluated to see if it is an acceptable option.

8 Conclusion

This project set out to design and implement a smart power strip which, through providing the ability to measure and present power consumption data and to remotely toggle its sockets on and off, could help a user lower his power consumption. A prototype was developed, comprising a power strip with wireless Internet connectivity and measurement and control equipment, an easy-to-use web application and a communication solution between these two entities.

When developed, the prototype was tested and evaluated to see if it provided the features it set out to provide, and if it fulfilled the goals defined in its specification. Results indicated that, with some further development of the power strip's software, this would be the case. This positive outcome showed that developing devices such as The Smarter Power Strip can be done relatively simple, and that smart home technology can play an important role in the task of lowering power consumption.

8.1 Future work

The most relevant future work in The Smarter Power Strip would be to implement a security solution. This would increase the value of the prototype significantly, since it would then basically have all the features needed in order to be used in reality. Further, it would be interesting to evaluate the device more thoroughly, for instance regarding the usability aspect, and regarding whether it would actually help a user lower his power consumption. Statistical surveys and interviews could provide an important view on the quality of the device.

There are also many ways in which new features could be developed for the device. The combination of providing ability to measure power consumption and toggling sockets on and off makes The Smarter Power Strip a device that could provide many different services. As we started this project, we had grand visions of what features the prototype should include. However, due to the time constraints, only the most basic functionality was implemented. Yet, below is presented some of the concepts that the prototype could be extended with in the future.

- Master-Slave outlets: By measuring the traversal of a programmable threshold value on the power consumed by a certain outlet, the power strip could either switch the remaining outlets on or off. This would be useful for reducing standby-related current draw of larger interoperating systems, such as a home theater.

- Real time cost: By connecting to an online database of regional electricity rates, it would be possible to provide the user with a real time estimate of their usage cost.

- Outlet scheduling: A programmable calendar system would allow the user to set times when each outlet should be active. This would allow the system to replace conventional in-line timer units, along with providing enhanced functionality, such as separate scheduling dependent on day of the week, daylight savings compensation, or, more generally, conditional statement programming.

- LED indicators: Lights on one or more outlets could provide visual feedback as to the current state of the system. A light could provide varying pulsation frequency and intensity to indicate real time current consumption.

References

Alan V. Oppenheim, Alan S. Willsky (1997). *Signals Systems.* Prentice Hall, Inc. ISBN: 0-13-651175-9.

Atmel (May 2013). *ATmega644PA.* URL: `http://www.atmel.com/devices/atmega644pa.aspx`.

Belkin WeMo (May 2013). URL: `http://www.belkin.com/us/wemo`.

Berners-Lee, T., R. Fielding, and H. Frystyk (1996). *RFC 1945 - Hypertext Transfer Protocol – HTTP/1.0.* Tech. rep. IETF.

Bishop, Robert H. (2007). *Mechatronic System Control, Logic, and Data Acquisition.* CRC Press. ISBN: 978-1-4200-0902-6. URL: `http://www.crcnetbase.com/doi/pdf/10.1201/9781420009026.ch31`.

Bonino, Dario, Fulvio Corno, and Luigi De Russis (2012). "Home energy consumption feedback: A user survey". In: *Energy and Buildings* 47.0, pp. 383 –393. ISSN: 0378-7788. DOI: `10.1016/j.enbuild.2011.12.017`. URL: `http://www.sciencedirect.com/science/article/pii/S037877881100627X`.

Calegari, F. (2005). "Electric Power/Energy Measurements for Residential Single-Phase Networks". In: *ISIE 2005 - IEEE International Symposium on Industrial Electronics 2005.* Vol. 3, pp. 1087–1092.

CraftyApps (Feb. 2012). *Tasker.* URL: `http://play.google.com/store/apps/details?id=net.dinglisch.android.taskerm`.

CyaSSL (May 2013). *CyaSSL.* URL: `http://www.yassl.com/yaSSL/Products-cyassl.html`.

Dierks, T. and E Rescorla (2008). *RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2.* Tech. rep. IETF.

Django (May 2013). *Django.* URL: `http://www.djangoproject.com`.

Django Packages (Apr. 2013). *Django Packages.* URL: `https://www.djangopackages.com/grids/g/api/`.

Django REST framework (Apr. 2013). *Django REST framework.* URL: `http://django-rest-framework.org`.

Dorf, Richard C and James A. Svoboda (2011). *Introduction to Electric Circuits.* 8th ed. Wiley. ISBN: 978-0-470-55302-2.

Droms, R. (1997). *RFC 2131 - Dynamic Host Configuration Protocol.* Tech. rep. IETF.

Elmore, Walter A. (2003). *Protective Relaying Theory and Applications.* CRC Press. ISBN: 978-0-203-91285-0. URL: `http://www.crcnetbase.com/doi/pdf/10.1201/9780203912850.ch3`.

Elteknik (2011). Institutionen för Energi och miljö, Chalmers tekniska högskola.

Energy Hub (Apr. 2013). *Energy Hub.* URL: `http://www.energyhub.com/individuals/`.

Fielding, R. et al. (1999). *RFC 2616 - Hypertext Transfer Protocol – HTTP/1.1.* Tech. rep. IETF.

Ford, B., P. Srisuresh, and D. Kegel (2005). “Peer-to-Peer Communication Across Network Address Translators”. In: *USENIX Annual Technical Conference, April 2005 (USENIX ’05).*

Fortum (Apr. 2013). *Fortum Hemkontroll.* URL: `http://www.fortum.com/countries/se/privat/energismart-hemma/smarta-produkter/hemkontroll/pages/default.aspx?gclid=CJW2xPPOrLYCFa-XcAodgB8Asw`.

GreenWave Reality (May 2013). URL: `http://www.greenwavereality.com`.

Gurevich, Vladimir. (2005). *Electric Relays principles and Applications.* CRC Press. ISBN: 978-1-4200-2641-2. URL: `http://www.crcnetbase.com/doi/pdf/10.1201/9781420026412.ch16`.

Herring, H. (2001). “The rebound effect, sustainable consumption and electronic appliances”. English. In: vol. vol.2. Marburg, Germany, pp. 870 –6.

IFTTT (Feb. 2012). *If This Then That.* URL: `http://ifttt.com`.

Jiang, X. et al. (2009). “Design and Implementation of a High-Fidelity AC Metering Network”. In: *IPSN ’09 Proceedings of the 2009 International Conference on Information Processing in Sensor Networks.*

Kurose, J. and K. Ross (2013). *Computer Networking - A top-down approach, 6th edition.* Pearson Education. ISBN: 978-0-13-285620-1. URL: `http://books.google.se/books/about/Computer_Networking.html?id=2hv3PgAACAAJ&redir_esc=y`.

Lien, C. H., Y. W. Bai, and M. B. Lin (2007). “Remote-Controllable Power Outlet System for Home Power Management”. In: *IEEE Transactions on Consumer Electronics, Vol. 53, No. 4.*

Mahaffey, Thomas R. (2002). “Electromechanical relays versus solid-state: Each has its place”. In: 50.19, p. 44. ISSN: 00134872. URL: `http://search.proquest.com/docview/221050756?accountid=10041`.

Math Bollen Martin Lundmark, Christer Boije John Åkerlund (2009). *Kort genomgång av effektberäkningar för lågenergilampor och andra olinjära laster.* Tech. rep. Luleå tekniska universitet, Elkraft.

Metal Film Resistors, Precision Type [MFP Series] (May 2013). URL: `http://www.yageo.com/documents/recent/Leaded-R_MFP_2011.pdf`.

Molina-Markham, A. et al. (2010). “Private memoirs of a smart meter”. In: *BuildSys ’10 Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building.*

Montenegro, G. et al. (2007). *RFC 4944 - Transmission of IPv6 Packets over IEEE 802.15.4 Networks.* Tech. rep. IETF.

Ross, J. P. and A. Meier (2000). “Whole-House Measurements of Standby Power Consumption”. In: *Proceedings of the Second International Conference on Energy Efficiency in Household Appliances.*

S. A. Khaparde, S. V. Kulkarni (2013). *Transformer Engineering Design, Technology, and Diagnostics, Second Edition.* 2nd ed. CRC Press. ISBN: 978-1-4398-5418-1. URL: `http://www.crcnetbase.com/doi/book/10.1201/b13011`.

Schwaber, Ken, Books24x7 - ITPro, and BusinessPro (e-book collection) (2004). *Agile project management with Scrum*. Redmond, Wash: Microsoft Press.

Silver Spring Networks (Apr. 2013). *Smart Energy Platform*. URL: `http://www.silverspringnet.com/products/?nav=p-335669&clickpath=products&wildcard=/index.html/#.UVskNqWGZfg`.

Srisuresh, P. and K. Egevang (2001). *RFC 3022 - Traditional IP Network Address Translator (Traditional NAT)*. Tech. rep. IETF.

Srisuresh, P. and M. Holdrege (1999). *RFC 2663 - IP Network Address Translator (NAT) Terminology and Considerations*. Tech. rep. IETF.

Streimikiene, D. and A. Volochovic (2011). "The impact of household behavioral changes on GHG emission reduction in Lithuania". English. In: *Renewable and Sustainable Energy Reviews* 15.8, pp. 4118 –24. ISSN: 1364-0321. URL: `http://dx.doi.org/10.1016/j.rser.2011.07.027`.

Tendril (Apr. 2013). *Tendril Connect*. URL: `http://www.tendrilinc.com/connect/`.

Tidwell, J. (2011). *Designing interfaces, 2nd edition*. O'Reilly Media, Inc. ISBN: 978-1-449-37970-4. URL: `http://designinginterfaces.com`.

Twitter Bootstrap (May 2013). *Twitter Bootstrap*. URL: `http://twitter.github.io/bootstrap/`.

UPnP-Forum (Apr. 2013). *UPnP*. URL: `http://www.upnp.org`.

Visible Energy (May 2013). URL: `http://www.visiblenergy.com`.

Zhang, L. (2007). "A retrospective view of NAT". English. In: *The IETF Journal* 3.2.

Minimal load	2W
Maximum load	2300W
Resolution	2W
Response time	5 sec
Measurement frequency	0.1 Hz
Power draw	3 W

Table 1: Specification

Appendix A Method

A.1 Specification

The power strip is intended to measure drawn active power over time (more commonly known as watt-hour or kilowatt-hour) for electrical appliances in homes. In addition it should also be able to toggle its outlets. The goal is to create a prototype which works with high precision for small loads but also has the ability to measure large loads. To appeal to a broader audience, low user commitment is prioritized. The device should have a wireless connection and all initial configuration of the power strip should be done wirelessly. The back-end system is a web server, and the power strip will push all measurement data to the server which will store and present data. Outlet toggling should be possible both from the web application and directly on the power strip.

These are the properties that should hold for the prototype (see also Table 1):

- Load - The prototype should be able to measure consumption on a load of at minimum *2W* and at maximum *2300W*
- Resolution - The prototype should be able to measure the amount of power drawn with *2W* accuracy.
- Frequency - The prototype should be able to present a measurement of the size of the consumption every tenth second.
- Response time - The prototype should be able to toggle a socket's output state in less than *5 seconds* after such a command has been issued from the web application.
- Power draw - The prototype should not draw more than *3 W*.

A.2 An agile way of working

In a project of this size, planning everything beforehand is difficult and can cause problems because it is impossible to foresee all problems that might occur. Therefore, we believe that taking an iterative approach to both making the product and writing the report was an advantage. The process framework that was used is called Scrum, the widely used agile method, Schwaber, Books24x7 - ITPro, and BusinessPro (e-book collection) [2004]. A product owner as well as a scrum master was therefore assigned to facilitate the process. Both the product and the report was broken down into manageable chunks in a prioritized backlog that could then be used when planning sprints, periods of two or three weeks that lead to a new, usable version of the product and report that had been set as the sprint goal. As an aid for deciding what sprint goals to plan for, a few milestones needed to be produced as working prototypes before the final product.

A.3 Sharing competences

The members of the project group had backgrounds in electrical engineering, computer science and information technology, all of which were necessary to complete the task. It was therefore important to make use of all the different competences so that they were shared throughout the group. The plan was to split the work so that everyone could work mostly on what they did best, but through weekly meetings the team was kept up to date with what had happened and what the obstacles were. Even if a difficult problem with the server software was encountered, the people working on the hardware could be able to contribute with a different perspective and valuable insights that opened up for a solution that the others could not have come up with on their own. The same was true for the hardware parts, and therefore the work was organized so that in the end, everyone was able to explain almost everything about the product.

A.4 From intra- to inter-team cooperation

In addition to our group, there was another team of six people who were working on the same idea, but producing a product of their own. This opened up a lot of possibilities to learn about cooperation not only within our team but also between the teams, so that both teams could learn from each other's mistakes and discuss important design decisions to expose potential flaws. After a discussion with the other group we decided to implement some form of interoperability between the two final products, and found two main ways in which that could be achieved. One was a tight coupling through power strip-to-server integration (so that our power strip was compatible with their server and vice versa) and the other to communicate more loosely between servers through an API. It was important that we would not end up with two completely identical products, therefore the latter seemed more reasonable.

Index