CHALMERS

The Smart Power Strip

Bachelor of Science Thesis in Computer Science and Engineering

JOHAN BREGELL

ERIK HILLBOM

DAVID JOHANSSON

PHILIP KARLSSON

JACOB LARSSON

MARCUS STIGELID

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

The Smart Power Strip

JOHAN BREGELL
ERIK HILLBOM
DAVID JOHANSSON
PHILIP KARLSSON
JACOB LARSSON
MARCUS STIGELID

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

Department of Computer Science and Engineering
Göteborg, Sweden June 2013

Abstract

People are relying on technology to help save the environment. Much difference can be done if people start making small changes in everyday life. This report describes the development of a Smart Power Strip: a modern tool to help households decrease their energy consumption. The Smart Power Strip is a system that gives its users the ability to measure the energy consumption of their home electronics. The system also brings the capability of remotely controlling home electronics in a smart way, to make it easy to shut down devices that are consuming unnecessary energy. This interdisciplinary project has resulted in a working prototype consisting of a power strip that can be controlled through a communication server and a web application. The Smart Power Strip has four power sockets that regularly measure power usage and send this data via WiFi to the remote communication server. Users can then log in to the web application and view graphs of the data and remotely control each power socket.

The report presents the theoretical background of the project's different aspects, before presenting design choices and the evaluation of the chosen designs. The conclusion of the report is that all primary requirements were met as the implementation of the prototype was completed. The evaluation shows that the system is working in a stable manner and that measurements are accurate according to specifications.

Sammanfattning

Vi förlitar oss mer och mer på att tekniken ska lösa dagens miljöproblem. Stora skillnader kan göras genom att många bidrar med små förändringar i vardagen. Denna rapport beskriver utvecklingen av Den Smarta Grendosan: en modern produkt som hjälper hushåll med att sänka sin energiförbrukning. Den Smarta Grendosan är ett system som ger sina användare möjligheten till att mäta sin hemelektroniks energiförbrukning. Systemet ger också möjlighet till att fjärrstyra elektronikprodukter på ett smart sätt, för att enkelt kunna stänga av apparater som drar onödig energi. Detta tvärvetenskapliga projekt har resulterat i en fungerande prototyp bestående av en grendosa som kan kontrolleras via en kommunikationsserver och en webbapplikation. Den Smarta Grendosan har fyra eluttag som regelbundet mäter effektåtgång och sänder denna över WiFi till kommunikationsservern. Användare kan sedan logga in i webbapplikationen för att se grafer över effektdatan och fjärrstyra varje eluttag.

Rapporten presenterar den teoretiska bakgrunden till projektets olika delar, innan den redogör för designval och utvärdering av dessa. Slutsatsen av rapporten är att alla primära mål uppnåtts eftersom implementeringen av prototypsystemet färdigställdes. Utvärderingen visar att systemet fungerar stabilt och att mätningar är noggranna enligt specifikationer.

Acknowledgement

A special thanks to Marina Papatriantafilou, Vincenzo Gulisano, Magnus Almgren and Giorgos Georgiadis for providing constructive feedback throughout the project. Gratitudes to Aleksander Bartnicki for making it possible to use the high voltage laboratory.

Dictionary

E

EEA *The European Economic Area.*

F

Framework *A reusable set of libraries or classes for a software system.*

G

GMT *Greenwich Mean Time refers to the standard time used in the time zone covering England.*

H

Hot swapping *Hot swapping describes the functions of replacing computer system components without shutting down the system.*

HTML *HyperText Markup Language is the main markup language for creating web pages.*

J

JavaScript *JavaScript is an interpreted computer programming language used mainly for interactive web content.*

JQuery *A JavaScript library. See also: Framework.*

M

MCU *Microcontroller Unit is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals.*

N

NAT *Network Address Translation is the process of modifying IP address information in headers while in transit across a traffic routing device.*

NTP *Network Time Protocol is a protocol for synchronizing time in a network with varying response times.*

O

ODBC *Open Database Connectivity is a middleware for accessing database management systems.*

ORM *Object-relational mapping is a technique for converting data between incompatible type systems.*

P

PCB *Printed Circuit Board is used to mechanically support and electrically connect electronic components using conductive pathways.*

R

RMS *Root Mean Square is a statistical measure of the magnitude of a varying quantity.*

S

SPI *Serial Peripheral Interface is a synchronous serial data link between devices.*

T

TCP *Transmission Control Protocol provides reliable, ordered, error-checked delivery of a stream of octets between programs running on computers connected to an intranet.*

U

UART *Universal Asynchronous Receiver / Transmitter is a piece of computer hardware that translates data between parallel and serial forms.*

UDP *User Datagram Protocol is a connectionless transmission protocol.*

UTC *Coordinated Universal Time is the primary time standard by which the world regulates clocks and time.*

V

VAC *Volt Alternating Current is the flow of electric charge which periodically reverses direction.*

Contents

1 Introduction

With a constant increase in population and with finite natural resources, it becomes more important every day to care for the environment. Today, one of the largest environmental problems is the combustion of fossil fuels. In Sweden almost no fossile fuels are used to produce electricity but in Europe significant part of these energy sources is used for producing electricity (Ekonomifakta, 2011). Therefore, minimizing people's electricity usage can spare the environment considerably.

Today all households in Sweden use 20.7 TWh of electricity each year to power their home appliances. In comparison to the total Swedish electricity usage of 130 TWh this accounts for 15.9%. If the energy usage could be lowered by 1% the amout of CO_2 emissions that reach the atmosphere would be reduced by 4563 *tonnes*. This can be translated to a reduction of 4 $kWh/month$ for each household in Sweden. As an individual it is difficult grasp how much electricity one actually uses, and unfortunately this leads to unnecessary consumption. With the right motivation this consumption could be reduced to contribute to a sustainable future for the planet (Energimyndigheten, 2012) (Statistiska centalbyrån, 2013) (Svensk energi, 2010).

Changing people's behavior can be difficult. Many electrical companies are trying to develop products and services to help people reduce their consumption and at the same time make everyday life easier. There are devices connected to the telephone line that can switch a power outlet on and off when one calls the device with a conventional telephone. This technology has evolved to devices connected to the Internet, controlled on a website. So far these devices are mostly designed for companies or very advanced home users. Recently the development of these types of devices has continued to include more sophisticated solutions like washing machines that turn on during the night when power is cheap or electrical cars that use the same principle when charging. Overall, companies in the home electronics business put great effort into reducing their products' energy consumption and lowering stand-by consumption. TVs, DVDs, stereos, gaming consoles and similar products are now often consuming under one Watt of power in stand-by mode (Eon, 2007).

Encouraging people to change their behavior regarding their electrical energy usage can be done in several ways. Sometimes it is as easy as informing people of how they can reduce their energy consumption. They may then gain motivation to save energy through the realization of how they can change to benefit the environment. Another way is to express possible reductions in energy directly as savings; an even more tempting reason to some people. Finally, people may also be driven by competition from comparing their consumption with other people. This project targets all of these motivators by developing a Smart Power Strip system.

1.1 Purpose

The overall purpose of the project is to develop a smart power strip, with the functionalities of measuring power and switching each power socket on and off, and using that functionality through a user-friendly web interface. The Smart Power Strip will connect and upload data to a remote server to which users will be able to log in through their Internet browser or phone app to access power statistics and control the switching functionality. The ability to view energy consumption statistics together with an encouraging web interface should help the users take control over and optimize their power usage. Users will be presented with an easy way of controlling electrical devices with an online accessible tool for scheduling or instantly turning on and off power sockets.

The project has two parallel design tracks: one for hardware and one for software. Regarding hardware design, the aim is to create a safe, dependable, low-cost and energy efficient device that should work as a prototype for construction of commercial products. The software development will first of all focus on the main features including straightforward handling of several hardware units and presentation of data. The amount of available time resources will determine what additional features will be implemented in a later stage.

1.2 Problem description

In total, the problem consists of creating a design of three parts. Figure 1 gives an overview of the system design. First of all, there is a hardware unit distributing electricity to multiple power sockets, while recording, calculating and transmitting consumption data and control signals. This unit, referred to as the Smart Power Strip, is represented by Figure 1 (a). Data from the power strip is sent via a local WiFi router to an intermediary server and stored in a database, represented by Figure 1 (b). The third part of the system is a web server that supplies an interface which the users can access through their Internet browser, represented by Figure 1 (c). Together, the Smart Power Strip, the communication server and the web application constitute the system that is analyzed, designed and evaluated in this project.

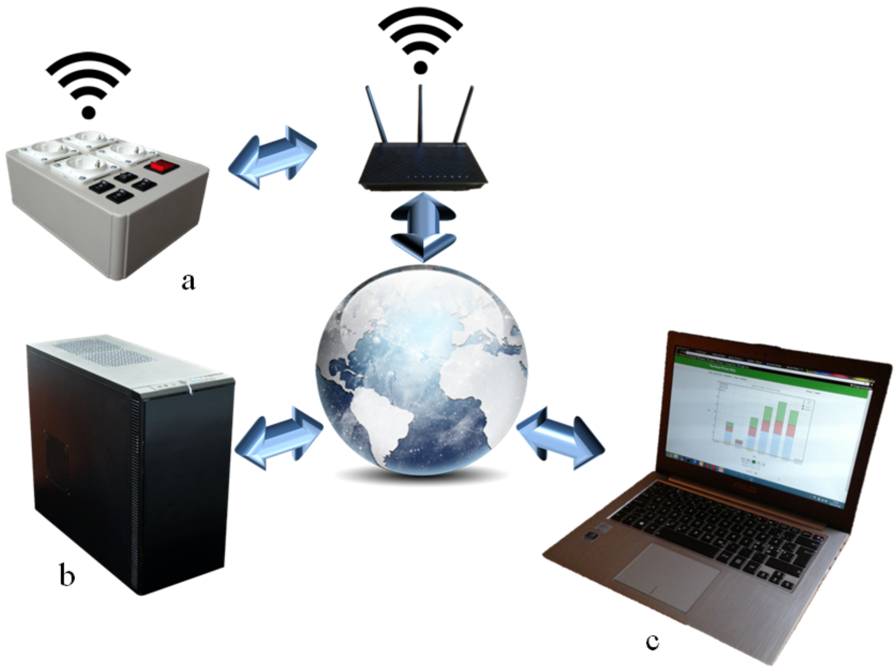


Figure 1: a) The Smart Power Strip that measure energy consumption and remotely switches power sockets. b) The intermediary server with a database and backends for web application and communication. c) The web application that provides an interface for the user in order to control the power strips and view consumption data.

1.2.1 Hardware

In order to implement the features needed for the Smart Power Strip, an microcontroller unit (MCU) is used as a base. It has to be chosen according to specifications such as number of inputs/outputs, processing power, A/D converter resolution and more. The measurement of instantaneous active effect on each of the devices power sockets is one of the projects main problems. To enable this, both current and voltage have to be measured in such a way that RMS (Root-Mean-Square) values and phase differences can be calculated. Overall, components and circuits have to be designed for current- and voltage measuring, power switching and WiFi connectivity.

This includes 230V circuits that require careful high voltage designs. In the initial stage, designing the current measuring circuit is the most complicated task compared to the design of the other parts.

After selecting MCU and designing the circuits, the MCU and WiFi module have to be programmed with algorithms for power calculations and switch controlling, and with protocols for sending and receiving messages over WiFi. Finally, to be able to create a complete prototype of the power strip, a casing has to be built or bought, depending on design choice and availability. The casing has to be appropriate for high voltages and specified after the sizes of the circuits have been determined. In every design step, the fact that dangerous voltages are used in the product has to be taken into account and the end product has to be safe to use according to electrical safety standards.

1.2.2 Software

For the smart features of the Smart Power Strip to be utilized, additional software must be present. This software should provide storage for consumption data and a way for the user to interact with the system. This is done, as explained in Figure 1, by a web application, a database and a communication server.

The communication server acts as a relay between the Smart Power Strip and the web application. It relays data from the power strip to the database and control signals from the web interface to the power strip.

In a scenario where the Smart Power Strip system is produced and sold to the market, the communication server may have to handle a great number of connections to power strips and users. The server software should therefore be designed with the ability to scale up to handle many connections at once in mind. Also, the database should be able to store the large amounts of data associated with many connected power strips. Ensuring scalability of the communication server is one of the main challenges faced when designing the software for this project.

There are a number of features that the web application should provide. The basic features should include the ability of switching on and off each of the power strip's sockets, and to be able to view the energy consumption of home electronics connected to the power strip. An additional feature may be grouping and naming of sockets and power strips. Further, a scheduling tool that switches the power of chosen groups, sockets or power strips can be implemented. The groups can be used to turn on and off more than one socket at a time and to summarize consumption data for multiple sockets.

As with all software connected to the Internet there are security risks that must be considered. One important aspect is to verify the sender of every packet received in the different parts of the software. Other aspects are to keep the consumption data private for the user and to manage user authentication securely.

1.3 Limitations

The main focus of the project is to develop a prototype that implements the most important functionalities. These functionalites are based on the timeframe of the project and the combined level of knowledge in the group. In view of this, the following small set of major features were formulated:

For the power strip (Figure 1, a):

- power measuring
- power switching
- stable communication with the communication server

For the communication server (Figure 1, b):

- stable communication with the database
- stable and reliable software
- ability to relay control packets

For the web application (Figure 1, c):

- easy to use interface
- ability to control sockets/power strips
- data representation in a graphic format

Secondary objectives to be pursued, if time allows, include the design of a more appealing casing for the hardware, the implementation of encrypted communication between the power strip and the communication server and the design of more advanced features in the web application.

1.4 Report layout

In Section 2 the report will give a brief description of the methods that have been used during the development of the Smart Power Strip system. The theory frameworks necessary in order to facilitate the understanding of the design choices made are presented in Section 3. The requirements imposed on the different parts of the system are specified in Section 4. These are followed by Section 5: Design and implementation, where the prototype system is implemented with regard to the previously presented requirements and theory. How well the implementation complied with the requirements is then evaluated in Section 6 and accompanied by Section 7: Discussion, which presents observations and possible improvements. Finally, the conclusion summarizes the project in Section 8.

2 Method

The development of the product was made through an iterative approach, based on ideas from Scrum - the popular agile development method (Schwaber and Sutherland, 2011). Using an agile development method ensured flexibility and provided the ability to make adjustments to the plans along the way if it became necessary.

The project was divided into seven periods, a short planning phase followed by three periods focused on implementation and three periods focused on testing. In the planning phase work was made on developing a plan for the project and to set the scope. In addition, research began on components to be used for the power strip as well as programming languages and techniques for the software. After this phase, the project was divided into six periods. This chapter aims to provide a general overview over the methods used during these six periods in order to create the final product.

2.1 Implementation

The first three periods were dedicated to the implementation of the product. Based on the research made in the planning phase, components for the power strip were ordered and programming languages and techniques for both the communication server, database and the web application were chosen. Before development on the web application began, a document with specific requirements over the functionality was created (see Appendix C). This provided a clear overview of the important functionality that needed to be implemented, as well as additional features that could be implemented if more time would be available. Regular meetings during the development of all the parts of the system gave an opportunity to discuss design choices and changes were made when new requirements emerged. The functionality of the different parts of the system were manually tested continuously during implementation.

2.2 Testing

When the implementation periods were over, testing of all parts of the product began to ensure that the communication between them was working correctly. While some additional features were added to the web application during these periods, the main focus was to make the product stable. Work was made to ensure that the product met the specified requirements. These requirements are described in more detail in Section 4.

3 Theoretical background

The subsequent sections discuss the underlying theoretical framework that was used when implementing the project's different parts. This includes electrical theory and components, comparison of software environments and cryptography.

3.1 Electrical theory

The following electrical theory was used when deciding on components and design for the Smart Power Strip prototype. The section presents different options for the implementation of the hardware, such as current measurement and power switching methods.

3.1.1 Energy calculation

Electrical energy is theoretically defined as instantaneous power integrated over a specific interval of time.

$$E = \int_{t_0}^{t_1} P(t)\mathrm{d}t \tag{1}$$

E is the energy in $Joule$, $P(t)$ the instantaneous power in $Joule/second$, $[Watt]$, and t denotes the time in seconds. In practice this means that in order to calculate the energy used in an electrical device during a certain amount of time, T, one would have to take note of the instantaneous power of each infinitesimal interval of time, dt, during T. The interval, dt, would have to be so small that the measures would take place constantly without interruption. This would not be possible in reality since measures cannot be executed continuously when making use of digital systems.

However if one can assume the instantaneous power to be constant over a fixed period of time, T_1, equation (1) yields

$$E = \int_{0}^{T_1} C\mathrm{d}t \tag{2}$$

which has the solution

$$E = C \cdot T_1 \tag{3}$$

From equation (3) the total energy can be calculated as the sum of the products of each time interval and the corresponding instantaneous power.

$$E_{tot} = \sum_{i=0}^{n} C_i \cdot T_i \quad (4)$$

In many cases, such as when households pay for their electricity consumption, energy is measured in kiloWatt hours, kWh. A kiloWatt hour is simply a specified amount of energy, more specifically

$$kWh = (3600[seconds] \cdot 1[Joule/second]) \cdot 1000 = 3600 \cdot 10^3[Joule] \quad (5)$$

When discussing electric power, three different types of power has to be taken into consideration: *real*, *reactive and apparent power*. Real power is the power that is used to do the work on the load and it is defined as

$$P = V_{rms} \cdot I_{rms} \cdot \cos\Phi \quad (6)$$

where P is the real power in $[Watt]$, I_{rms} the RMS (Root Mean Sqare) of the current, V_{rms} the RMS of the voltage and Φ the phase difference between voltage and current in time. An example of phase difference is shown in Figure 2.

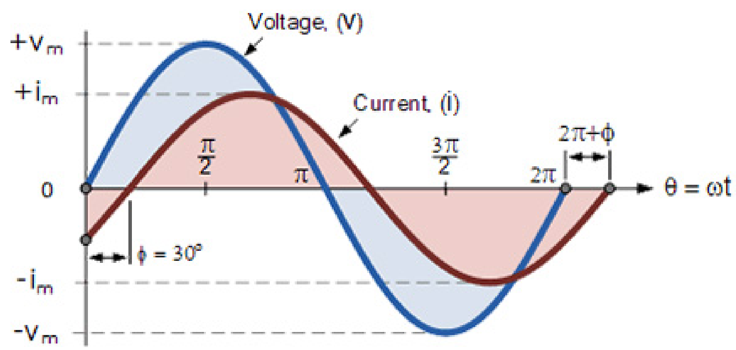


Figure 2: A graph showing the voltage and current of an inductive load. In this example the phase difference is 30 degrees.

The reactive power is defined similar to equation (6) but with the $\cos\Phi$ term replaced by $\sin\Phi$ and it is measured in the unit $[VAr]$. Reactive power does not perform any work at the load instead it dissipates by heating cables, transformers and other equipment used for transporting electricity (Bartnicki, 2012).

Finally, *apparent power* is defined as $V_{rms} \cdot I_{rms}$ without taking the phase difference into consideration. It is used to describe the total power in an AC-circuit, both dissipated and absorbed. From a consumer's perspective, only the real power needs to be taken into consideration since it is the only type that the customer can make use of when driving standard home electronics such as TVs, computers and vacuum cleaners. This is also what the consumer is billed for.

Theoretically, for a continuous wave the root mean square is defined as

$$V_{rms} = \lim_{T \to \infty} \sqrt{\frac{1}{T} \int_0^T [v(t)]^2 \mathrm{d}t} \tag{7}$$

where V_{rms} is the RMS value of the wave, $v(t)$ the wave function through time and T the boundary of the integral.

By using the same argument as applied to equation (1) it is not possible to execute integration continuously by using digital technology. Therefore equation (7) must be approximated for discrete values.

$$V_{rms} = \sqrt{\frac{1}{n} \sum_{i=0}^{n} n_i^2} \tag{8}$$

Here, V_{rms} is the RMS of the wave, n the number of samples and n_i the amplitude value of the wave at sample i.

The phase difference, Φ, can be calculated in different ways. It is possible to mathematically transform a function describing a waveform from the time domain to the frequency domain using Fourier analysis (Oppenheim, 2011). For the case of discretely sampled waveforms, the Discrete Fourier Transform (DFT) is defined as

$$X_k = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{-i2\pi kn/N} \tag{9}$$

where $k = \frac{f*N}{SampleRate}$, N is the number of samples, f is the frequency and n is the index of the vector. By inserting a fixed frequency, f, into equation (9) a DFT can be calculated for a single frequency. If this is executed for two separate sine formed functions $r[n]$ and $i[n]$ the phase difference can be calculated using the following equations.

$$R = \sum_{n=0}^{N} r[n] \cdot C + i[n] \cdot S \tag{10}$$

$$I = \sum_{n=0}^{N} i[n] \cdot C - r[n] \cdot S \tag{11}$$

Here, C is $\cos(2\pi nk/N)$ and S is $\sin(2\pi nk/N)$. Combining equation (10) and (11) the phase difference between $r[n]$ and $i[n]$ is calculated as

$$\Phi = \arctan(\frac{I}{R}) \tag{12}$$

It is also possible to calculate Φ directly by measuring the number of samples between the two neighbouring peak values of $r[n]$ and $i[n]$. This sample difference can then be calculated to radians as

$$\Phi = 2\pi \cdot \frac{Sample\ difference}{Samples/Period} \tag{13}$$

3.1.2 Current measuring

As mentioned, both current and voltage of the power sockets have to be measured to be able to calculate real power. Voltages can be measured directly by using analog to digital converters, but currents must first be converted into voltages before they can be measured. The corresponding currents are then calculated by logical circuits. There are several ways of measuring current. Three of these are resistive current sensing, Hall effect metering and current transforming.

Resistive current sensing

Resistive current sensing uses the fact that the voltage drop over a resistor is proportional to the current flowing through it, as can be seen in Figure 3.

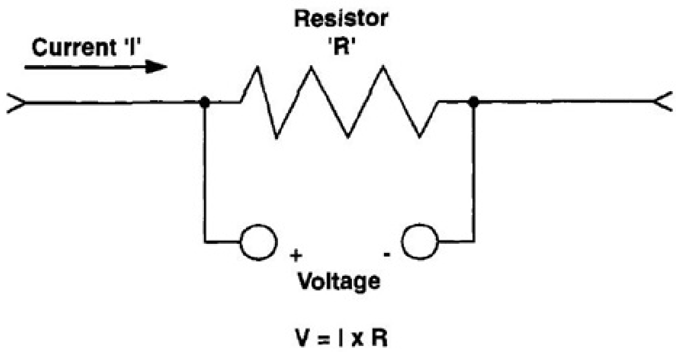


Figure 3: A principle sketch of resistive current sensing (Ramsden, 2006).

The voltage drop over the resistor is given by $V = R{\cdot}I$. It is required that the resistor has a very small and accurate resistance so that the power dissipated in it is small and the measurements are accurate. The type of resistors used in this type of application is called shunt resistors.

Resistive current sensing is an accurate and inexpensive current sensing technology, but it has a few drawbacks. Without additional components, the high voltage mains are not electrically isolated from the low voltage measurement circuits. The small resistance of the shunt ensures that the differential voltage over the resistor is small and thus also the power dissipation. However, the voltage potential that is shared on both sides of the shunt – the common mode voltage – is large in high power applications and requires extra insulation from low voltage circuits (Ramsden, 2006).

As the shunt resistance is chosen to be small to minimize power dissipation, the voltage drop over the resistor also becomes small. This may cause a problem as it can be hard to measure low voltages accurately. In many applications it is therefore necessary to use some sort of amplifier before the measuring circuitry.

Current transformer

There are several ways of measuring current by using magnetism. One of the most common is using a toroidal coil that encircles the current-carrying conductor as in Figure 4. To achieve greater accuracy the coil is winded around a high permeability core that significantly increases magnetic flux inside it. The current generates a circular magnetic flux around the conductor and in the core. This in turn induces a current in the toroidal coil that is proportional to the original current. This can then be measured by resistive current sensing as discussed previously, the difference being that the voltages measured are now low and the circuits electrically isolated from the potentially high voltage of the conductor carrying the current that is measured. The only interference the current transformer has on the measured circuit is the very small inductance of the coil. This only affects AC applications and is in most cases negligible (Ramsden, 2006).

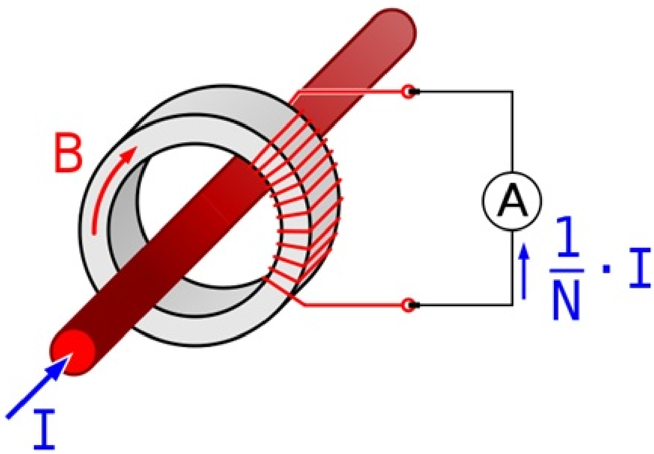


Figure 4: A toroidal current transformer (Wikipedia, 2009).

There are many factors that affect the accuracy of this sensing method. Overall, current transforming equipment are inaccurate when sensing low currents. Another factor is the saturation flux of the toroid core that may decrease with use and cause linearity errors. Similar errors may also be caused by temperature changes that impact the core's permeability (Ramsden, 2006).

Hall effect sensing

When charge-carriers – like electrons in electric currents – pass through a magnetic field they are affected by a force given by the following relation:

$$\boldsymbol{F} = q\boldsymbol{v} \times \boldsymbol{B} \tag{14}$$

The force, F, velocity, v, and magnetic flux, B, are all vectors. Hence the force is right-angled to both v and B according to the cross product. The constant q is the charge of the charge-carrier, and qv could therefore also be written as I, the current. The Hall effect occurs when a current is passing through a magnetic field. Every electron in the current is affected by the force given by equation (14). In Figure 5 it can be seen that this force pushes the electrons to one side of the conductor. This creates an excess concentration of electrons to this side, and depletion on the opposite side. This in turn gives rise to an electric field across the conductor. As more electrons move, the electric field grows and eventually equilibrium is reached when the force of the electric field on the electrons is equal to the force generated by the magnetic field. The electric field causes a voltage to appear, which can then be measured. The voltage is linearly proportional to the magnetic flux and current flowing through the Hall element and is also affected by conductor thickness and by the material's carrier density (Ramsden, 2006).

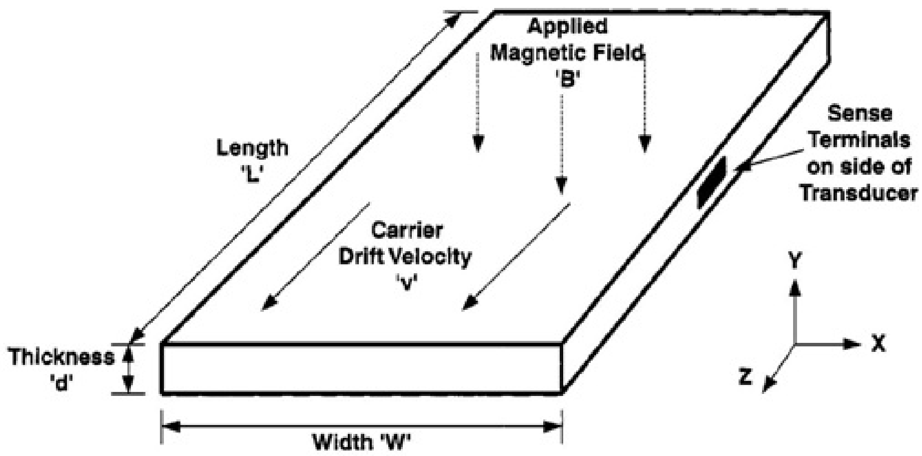


Figure 5: A sketch explaining Hall effect physics (Ramsden, 2006).

Integrated circuits that use the Hall effect phenomenon is available. In these, the magnetic flux around a current-carrying conductor is led through a magnetic core and concentrated over a small Hall effect element. A constant current is generated to flow through this element. The fact that the Hall effect voltage is proportional to the magnetic flux is then used to measure the original current (Popovic, 1991). An illustration of this principle can be seen in Figure 6.

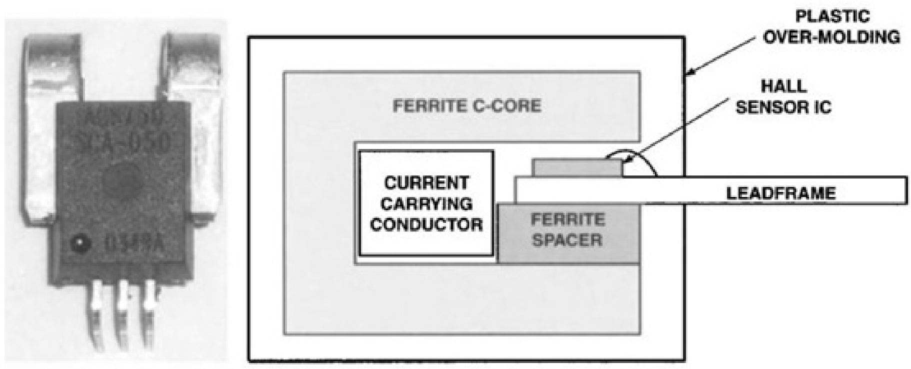


Figure 6: An integrated Hall-effect sensor based on magnetic coupling (Ramsden, 2006).

The current-carrying conductor and the Hall effect element are only coupled magnetically in these types of sensors and are thus electrically isolated. Because of this they can be used with high voltages without endangering other electronic circuits present. As the sensors are built into small packages that can be mounted on circuit boards, they cannot withstand as high currents as current transformers. The sensors are generally not as accurate as resistive sensors, although the influence on the measured conductor is negligible in most cases. As this type of Hall effect sensor uses a high permeability core much like a current transformer does, the same linearity problems may arise because of saturation flux and temperature changes (Ramsden, 2006).

3.1.3 Microcontroller unit

In order to connect multiple electrical modules to each other, a master device is needed. In embedded applications this device is likely to be a microcontroller unit (MCU). A microcontroller is a small computer built on a single integrated circuit. It contains a processor, memory, programmable input/output pins and additional units suited for embedded applications. Sensors are often connected to MCU:s and since there are both analog and digital sensors many microcontrollers are equipped with analog/digital converters (ADC:s). The block diagram of an AVR type microcontroller can be seen in Figure 7.

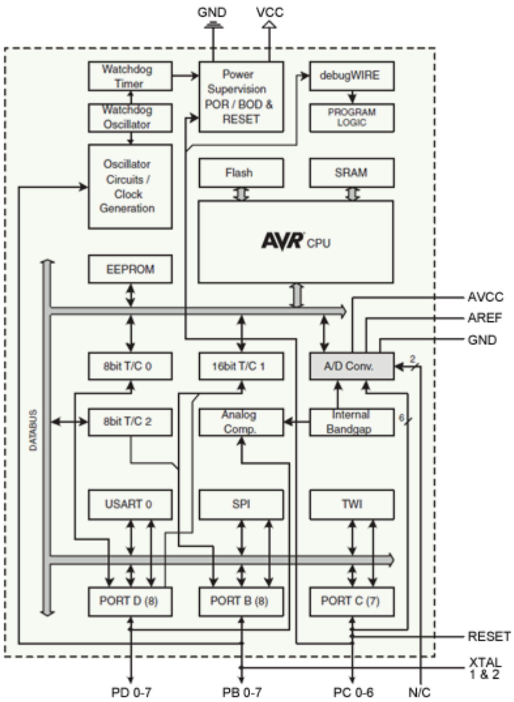


Figure 7: Block diagram of a AVR type microprocessor from the ATMEGA family (Atmel, 2010).

In analogy to the traditional computer, the microcontroller makes use of different types of memory. It uses nonvolatile memory for storing programs and information that is stored at compile time with assistance of separate hardware. This memory is called the program memory and consists of FLASH type memory cells. For storing nonvolatile information during execution of the program, a built in EEPROM is used. The microcontroller uses volatile static random access memory cells (SRAM) for storing runtime data such as variables and the active parts of the program.

There are several different microcontroller architectures of interest when designing embedded systems. The two most interesting architectures for this project are the ARM and AVR architectures. The ARM architecture generally offers more computing power compared to the AVR, and it is suited for applications that run complex operating systems. While the AVR architecture might not be as powerful, there are lighter applications where it can be a better solution due to its simplicity and low price. A few of the ATMEGA AVR microcontrollers can with ease be flashed with the Arduino bootloader[1]

[1]The Arduino bootloader is a software which makes USB-programming possible.

This makes low level programming easier due to the possibility of making use of the Arduino environment and existing libraries. This is also possible to achieve using specific ARM microcontrollers.

3.1.4 Power switching

Controlling high voltages with low voltage electronics can be difficult. This can however be done by using relays, in which a control voltage somehow switches another voltage. The most common types of relays are electro-mechanical and consist of a coil that magnetically switches the main voltage and provides isolation. These types of relays generally require a relatively large current through the control inputs because of the coil, although there are types that switch state permanently and thus only need a short voltage pulse for each on/off. However, these are expensive compared to ordinary, momentary relays (Sinclair, 2000).

Another type of relay is the SSR (Solid State Relay). Solid state basically means that the electronic component has no moving parts; the SSR is therefore not a mechanical device. There are different SSR designs but the ones intended to handle AC on the main load usually has a built-in LED which the input control voltage controls. The LED then lights a semiconductor component that switches the higher main voltage. SSRs require much smaller control currents compared to electromechanical relays because of the low power consumption of the LEDs (Bishop, 1997).

3.1.5 Electrical safety

Electronic products should be designed to meet regulations common throughout the EEA area. Firstly, this requires products to be built so that people, animals and property is not endangered if the products are correctly installed, used and maintained. The manufacturer has responsibility to ensure throughout the manufacturing process that all regulations are met. Similarly, the manufacturer should ensure that the product follows directions regarding fire safety. Products should also be designed with respect to regulations regarding electromagnetic interference. Manufacturers whose products meet the requirements can apply for CE-branding. This is then used as proof that the product is certified in the EEA area (Olofsson, 2013).

3.2 Comparison of programming tools

When starting to develop software the first task is to find a suitable language to program in and decide on which frameworks to use. To do this one must look at what the software is supposed to do, in what environment the software is supposed to run and also which languages the programmers have experience with.

To begin with different frameworks for the web application will be compared, followed by languages for the communication server and lastly a few different data storage solutions.

3.2.1 Web development frameworks

Three of the most popular web development languages are PHP, Python and Ruby. Each of them is a part of one or more web development frameworks. For PHP there is CakePHP, Python has Django and Ruby has Ruby on Rails. They will all assist when developing web applications, but they have different advantages.

CakePHP's programming language PHP is considered to be one of the most popular languages (TIOBE, 2013) and is also said to be easy to learn (Plekhanova, 2009). Ruby is the least popular language out of the three, (TIOBE, 2013) but Ruby on Rails still has a large community and a good manual. Django is considered to have an even better manual – mainly because there are many examples in the documentations. Compared to the others, CakePHP is considered to have a rather poor manual (Plekhanova, 2009). All of the three frameworks come with an Object Relational Mapper (ORM), and all of them support MySQL, PostgreSQL and SQLite (CakePHP, 2013) (Django Software Foundation, 2013) (Ruby on Rails, 2013).

Django and Ruby on Rails provide good testing tools, while CakePHP has fewer options. One large advantage with Django is the automatically generated administrator view that is easy to adjust for different purposes. Additionally, when comparing Django with Ruby on Rails then Ruby on Rails has the disadvantage of a more complex syntax for handling templates (Plekhanova, 2009).

3.2.2 Server development languages

To develop the communication server a stable and fast language is preferred. The language must also have support for Open Database Connectivity (ODBC) to be able to communicate with the Database Management System (DBMS). The following languages all support this and are compared based on their respective strengths.

Java

Java is a general-purpose, concurrent, class-based, object-oriented programming language. It is specifically designed to be simple, robust and safe to run while still being portable, scalable and modular. All Java programs are run in the Java Viritual Machine which is platform independent. This means that Java programs can be executed on almost any platform. However, hot loading – a way to insert new program code without the need of a program restart – is not supported natively which can become a problem if software updates are necessary.

C++

C++ is a language that builds upon the C language. It can run very efficiently on a large number of platforms due to the efficient compiler to native code. This also makes it suitable for high performance server applications. C++ meets the requirement of being run in real-time and is highly reliable. The availability might be a problem due to no hot loading of code i.e. the server needs to be shut down, recompiled and started again to do a program update.

Erlang

Erlang was developed by Ericsson to manage their telephone switch AXD and was released to the public in 1998. It is designed to be distributed, fault-tolerant, real-time and non-stop. The distribution feature makes the written programs easy to scale by adding more nodes to the program cluster. An Erlang server is highly available due to the native support for code hot-swapping. However, the code can not be compiled into an executable file and is dependent on the Erlang Virtual Machine.

3.2.3 Database management systems

In order to find a good storage system every part of the project that needs to communicate with the storage system has to be taken into consideration. In this project this means the web application and the communication server. Most web development frameworks have support for several different Database Management Systems (DBMS). The two most common are MySQL and PostgreSQL. Both of these are also compatible with most programming languages through support for Open Database Connectivity (ODBC). The Erlang language has additionally support for a more advanced solution called Mnesia.

Mnesia
Mnesia is the default distributed database management system (DDBMS) in Erlang. It provides great performance, redundancy and security. It does not act as or work like a normal DBMS but is intended to support Erlang and therefore it is built into the language and has no standalone installer. This would be the fastest solution in the communication server if it is coded in Erlang.

MySQL
MySQL is one of the most popular open source relational database management system (RDBMS). It has great performance and stability and many tools are provided by the community. When developed the main goal was to build the fastest DBMS and therefore it has an advantage in speed over PostgreSQL. The options for secure connections and software backup are also available. One of the main benefits is the MySQL Workbench which has a great graphical database builder.

PostgreSQL
PostgreSQL is according to their own slogan "The world's most advanced open source database". It was developed with the intent to support all features of the SQL language and has many fertures that MySQL does not support. PostgeSQL lacks a good graphical development IDE. The system has some advantages and some disadvantages in regards to speed and is in general slower than MySQL. Furthermore, it has some additional features to help the developer. One of these is the option to get the selected data formatted in JavaScript Object Notation (JSON), which is a standard method to format data between different programming languages.

3.3 Authentication and communication security

Connecting a device to the Internet requires consideration regarding security. Many devices connected to this large network need to communicate and transfer private information. Ciphers have been created for encrypting messages in order to satisfy the need for this private communication. They are categorized into two types: symmetric and asymmetric. These ciphers are explained in the following sections.

Creating accounts in order to use services on the Internet is a common phenomenon. Passwords are often used for authentication of such accounts. Different ways of adding security to this authentication process are discussed in the subsequent sections.

3.3.1 Communication encryption

In this section two types of ciphers will be explained: symmetric and asymmetric. A protocol that combines these types will also be described.

Symmetric cipher
Symmetric encryption means that a secret key is shared before communication is initialized. This key is then used to encrypt the message and also to decrypt the message when it has arrived at its destination, hence the symmetry (Kizza, 2009). Advanced Encryption Standard (AES) is a symmetric cipher in which the key can be 128, 192 or 256 bits long. It is a block cipher which means it splits the message into blocks and each of these blocks contains 128 bits. If the message is longer than 128 bits that means that the message will be divided into more than one encryption block. Each block is then run through a loop performing several predefined operations. The number of iterations depends on the length of the key. But in the loop, bits are shuffled around and changing places. This will make the message unreadable and also extremely difficult to decrypt (Knudsen, 2011).

Every step done during encryption must be easy to perform backwards during decryption for an authorized person with the correct key. Therefore, the operations performed in the loop during encryption also have inverse operations used during decryption (Knudsen, 2011).

The performance of the cipher is good and it is suitable for 8 bit processors, but it is also efficient on 32-bit and 64-bit processor because of opportunities for parallelism. The security of the cipher is good as long as the key is kept a secret. Knudsen and Robshaw claim in their book "the Block cipher companion" (2011) that "no serious weaknesses have been found in the AES".

Asymmetric cipher

Asymmetric encryption is when one private and one public key are used. This will be explained in an example where Bob wants to send an encrypted message to Alice. Bob requests Alice's public key which is shared publicly. The public key is sent to Bob, and he uses it to encrypt his message to Alice. Bob then sends the encrypted message back to Alice and she can then use her private key for decryption. This way – if Bob can be sure he encrypted the message with Alice's public key – he knows that only Alice can decrypt his message, assuming her private key is secret. Asymmetric encryption requires larger key sizes than symmetric encryption, and the algorithms used when generating new keys are not optimized for being executed in hardware. This is why asymmetric encryption is slower than the symmetric encryption in computer systems (Kizza, 2009).

RSA is a commonly used asymmetric cipher, which uses two large prime numbers to produce private and public keys. The cipher utilizes the problem that arises when trying to factorize large numbers into their original primes. This can, as almost all calculation problems, be solved using brute force. The only problem is that the algorithm for finding the primes has exponential complexity. Depending on the security demands the key length can be adjusted between 1024-4096 bits. This key length must be increased with the same pace as computer calculation power increases (Tuzzio, 2012).

Combination of symmetric and asymmetric ciphers

There are methods that combine both symmetric and asymmetric encryption. These methods use asymmetric encryption to share a key which then is used for symmetric encryption. This gives the advantage of the faster symmetric encryption and an easy way of updating the symmetric key. Users however need a way to authenticate public keys. One example of this is SSL, which is a commonly used encryption protocol. SSL is an abbreviation for Secure Socket Layer, and is a protocol for secure communication. For example, it is used by HTTPS which implies that almost all web browsers support it. It uses both symmetric and asymmetric encryption, and has support for several different ciphers including RSA (Oppliger, 2009).

3.3.2 Password protection

When a web application uses login sessions to authenticate users, passwords need to be stored in a database. The passwords can then become available for other people if the server, with the database, is exposed to an attack. Therefore the passwords need some extra protection. Different types of this kind of protection are described in the following section.

Hashing

Hashing a password is a way of making it harder for possible intruders to actually use them. It is intended to be a one way transformation of the password into an unreadable text where changing one bit in the password should result in many changed bits in the hashed text. Instead of storing the password in the database the hashed value is stored and every time a user is authenticated, the input password is hashed and compared with the hashed password in the database (Sullivan, 2012).

Salting

For every password a random text string called a salt can be added. This will give two identical passwords different hash values. Besides concatenating the salt with the password before it is hashed it will also be written in plaintext together with the hash of the password in the database. This is required during authentication because the user input password needs to be hashed with the same salt, or else the input password's hash would never conform to the stored password's hash from the database. If an intruder would gain access to the hashed passwords and they did not have any salt, then it is easy to compare all the passwords with a database with pre-computed hash values for common passwords. If instead all of them have a salt value, then the intruder has to calculate hash values for each password and this is time consuming (Sullivan, 2012).

Key stretching

Key stretching means the hashing is performed with many iterations. This increases the time it takes to calculate each hash value. The number of hash iterations is selected so that the time lag should never be noticeable for the user (Sullivan, 2012). If someone comes across a hashed password, that person would probably have to try millions of password combinations before finding the password corresponding to the hashed value acquired. If the password uses key stretching each password tried will take more time.

For example, if the calculation of the hashed value takes 0.3 seconds more, then testing one million values will take 300 000 seconds. That is the same as three and half day.

3.4 Visualization of measurement data

When displaying data points in the form of graphs there are many aspects that need to be take into consideration. This section describes the underlying problems in form of performance issues when handling large amounts of data and the way a user interacts and interprets the graph itself.

3.4.1 Usability

One of the biggest challenges when it comes to graph visualization is to present the data in a useful way (Chen, 2005). When the amount of data increases, users will need to interact with the graphs and explore the data to gain the knowledge needed to make decisions (Huang, Liang & Nguyen, 2008). This means that the tools used for visualization need to provide good ways to interact with the graph.

3.4.2 Style

The style of a graph can affect how a user perceives the information that is presented (Moere et al, 2012). It could be differentiated through the design of the graph but also through factors such as colors and contrast. If a good color combination is used, then these factors could help facilitate the understanding of the information, but a bad color combination could make the user interpret the information incorrectly (Kim, 2010).

3.4.3 Scalability

Scalability is one of the key issues of graph visualization (Herman, Melancon & Marshall, 2000). Many of the existing techniques for small and medium sized datasets do not have good performance when it comes to large datasets. While there is currently no single solution that can handle large datasets well, researchers in the graph visualization area have come up with different solutions that handle the issue with varying success. One of the most effective solutions is to divide the graph hierarchically into a clustered graph (Huang

& Nguyen, 2007). By clustering the graph and creating smaller sub-graphs the complexity can be reduced and users can then more easily interpret and navigate the displayed graph.

Improvement in performance for large datasets could also be made by using distributed computing. By distributing the work between several computers performance of the computing of the data could drastically improve. This can be implemented by using e.g. the open-source distributed computing framework Hadoop. Another solution is the use of hardware to compute the visualizations by using ordinary graphics cards with graphical processing units (GPUs) (McDonnel & Elmqvist, 2009). A GPU is capable of processing large amounts of data quickly and in the future it will likely become more frequently used for computation of graphs as datasets continue to grow.

4 Requirements

Expanding the problems described in the introduction, this section will further explain the requirements imposed on the different parts of the project. These requirements are the base for the project evaluation presented in Section 6.

4.1 Power strip

In addition to the basic functionalities mentioned in the introduction, the Smart Power Strip should meet the following requirements:

Measurment accuracy
The power strip should preferably be able to measure active power with an accuracy of approximately 2 W, with currents down to 10 mA.

Unit power consumption
The electrical components of the Smart Power Strip should have a small energy consumption compared to other home electronics. For example it should not have a greater consumption than a modern low power lamp.

Total unit cost
As the unit built during the project is a prototype, the cost will be significantly higher than a manufactured version. Because of this the final cost of the prototype is not the main focus of the development process. The prototype should, however, be designed with the ambition to minimize component costs.

Ability to buffer data locally
In case of connection failure between the power strip and the server, a buffer in the power strip should have the capacity to store measured data for no less than one month.

Electrical safety
A vital requirement for the power strip is that it follows regulations regarding electrical safety for home electronics. This includes protection between users

and high voltages, and that the device should be able to withstand falls or other normal scenarios for home electronics.

Operational reliability
The Smart Power Strip should be able to recover from a lost server connection. It should also be able to take measurements every 10 seconds and still have a respons time of maximum one second for incoming switch commands.

Communication security
The major part of the communication with the server consists of sending consumption data. Adding security to the communication is necessary to protect the user's energy consumption data and also for verifying the sender of all the received packets. This is a low priority requirement.

4.2 Communication server

The communication server is the instance between the power strip (Figure 1, a) and the web application (Figure 1, c), therefore the server must have a very high uptime, also know as availability. To achieve this when developing the server the following three cornerstones has to permeate the implementation.

Reliability
This focuses on how often something in the server breaks down. This is also referred to as the mean time between failure (MTBF).

Scalability
The server should be able to increase the performance in a scenario where the load on the server increases; either by adding more powerful harware to the server machine or by using serveral physical server machines.

Real-time
Every packet with data or control signals has to be processed with an as low delay as possible.

Features
With these cornerstones in mind the features that are required to be implemented into the server are:

- receive data packets from the power strips and insert them into the database.
- relay control signals from the web application to the power strips.
- manage security for the communication with the power strips and the web server.
- be able to analyze the energy consumption on each socket on every connected power strip and turn them on if connected devices are in standby.
- fetch data containing scheduling information from the database and control the sockets on the power strips according to that information.

4.3 Database management system

The web application (Figure 1, c) and the communication server (Figure 1, b) needs a place to store and retrieve data in an easy, secure and dependable way. To achieve this a database management system will be used. In a scenario with one hundred power strips the database will need to be able to handle several hundred connections at once as it serves both the web application, which can be accessed by all users, and the communication server. In this case the database has to be able to make several hundred new data inserts per second and up to five times as many updates and selects – possibly even more depending on how many users that are acessing the web application. The main usage of the data storage system will be when the server stores new measurement data and states of power strip sockets, but the visualization of the consumption data in the web application will also impact on the performance.

The connections to the database do not have to be encrypted assuming all the sensitive data stored will be encrypted, although the passwords for login will have to be adequately strong.

Prevention of data loss will have to be implemented on hardware level by regular backups and redundant drives, and if the data storage system supports software level data duplication, this is also a desired feature.

4.4 Web application

All web application requirements can be found in Appendix C. The following paragraphs explains the most important of these requirements.

Primary requirements
The web application is the user's tool for controlling the power strips. In order to provide that service several requirements have to be fulfilled. Firstly, the user must be able to create and edit an account. The account includes a password which needs to be used and stored in a secure way. This also means that the web application must provide a login session to keep track of the users that are logged in and the ones that are not. Once the user is authenticated the web application should let the user add and remove power strips to the account. Then the user should be able to name a power strip and also name each individual socket on the power strip. The sockets should have an on and an off switch that is executed with a short response time. This means that when a user executes the command for switching between on and off, the socket should change its state within a few seconds. All of these sockets will generate energy consumption data. The user should be able to view this data through graphs in the web application. All the requirements mentioned so far are considered to be very important because without them there would not be a product.

Secondary requirements
There are also requirements that are less important, but still necessary for making the product usable. Two of these requirements are grouping and scheduling of sockets. For making it easier for the user to control many power strips and their sockets, it should be possible to create groups. These groups can contain power strips, other groups and individual sockets. The groups will be used when viewing consumption data from multiple sockets, power strips or groups, and also for switching all of their respective sockets on and off. The scheduling requirement aims to enable scheduled switching events for sockets, power strips and groups.

5 Design and implementation

This section describes the implementation of the Smart Power Strip and the software behind both the web interface and the communication server. Design choices are explained based on the theory explained in Section 3 and the requirements specified in Section 4.

5.1 Power strip

The hardware of the Smart Power Strip is designed as an embedded system centred around a microcontroller unit. This unit connects and controls the components needed for WiFi connectivity, current measurments, non volatile storage and power switching. A power transformer connected to voltage regulators is also included in the design so that the power strip can supply its components with power without the need of an external low voltage power supply. The following sections presents the design choices made when constructing the Smart Power Strip prototype. By using theory and the requirements specified in previous chapters a microcontroller unit, WiFi module, power switching device and current sensor was chosen.

5.1.1 Microcontroller unit

When deciding on which microcontroller unit (MCU) that would be best suited for controlling the power strip a few aspects were taken into consideration. The MCU needed to have enough analog in- and outputs to be able to measure the current on each socket as well as the voltage from the main power supply. The analog digital converter also needed to have enough precision in order to represent the analog values as accurate digital values. Another requirement was enough RAM so that voltage and current could be sampled accurately. Since the MCU has to control external modules such as the WiFi and

modules, the microcontroller must support both SPI and UART for communicating with these modules. SPI and UART are two standardized serial communication protocols used in embedded systems. Several other components were ordered in addition to the MCU, so price had to be weighed against performance to get the most out of the microcontroller unit.

Taking these factors as well as previous experience into consideration the AVR type microcontroller unit Atmega328p was chosen. It features 6 analog as well as 14 configurable digital I/O:s. The Atmega supports communication over both UART and SPI. It has a low price compared to an ARM type microcontroller and is widely available. With a clock speed of 16 MHz, 32kB of FLASH memory, 2kB of SRAM and 10 bits precision in the ADC it had enough computer power for the Smart Power Strip. The Atmega328p MCU is compatible with the Arduino boot loader. This allows for use of the Arduino programming environment and libraries. This made the process of lower level programming significantly easier as existing libraries for example the wifi module and SPI communication could be used.

5.1.2 WiFi

A module that provides WiFi connectivity was built into the Smart Power Strip to enable the device to connect to the communication server over the Internet. The WiFi module had to provide a UART or an SPI connection for it to be able to communicate with the Atmega328p microcontroller. It also had to implement all layers of the network stack – excluding the application layer – as these are not implemented in the MCU. The program running on the MCU had to be supplied with a TCP/UDP socket that it could read from and write to.

As size and cost of the electronic components of the power strip was to be minimized, the WiFly GSX RN131 C was chosen as it met all requirements. It also has an onboard antenna which was convenient as this saves space. The module also implements an AD-HOC networking mode which the Smart Power Strip uses for network setup. At power up, the device tries to connect to a WiFi network with a stored SSID and passphrase. If this connection fails, the device sets up an open AD-HOC network. The user can then join this network with a WiFi-enabled computer or smartphone. Through a web brower, the user is then presented with an HTML page. The page shows a list of available WiFi networks that the user can connect to by filling in a passphrase field. The Smart Power Strip will then save the chosen network so that the setup only has to be done once.

5.1.3 Power measurement

Since the sockets in power strips are connected in parallel with the main power supply, each socket share the same voltage as the supply. Therefore the voltage only had to be measured once each sampling session. The current on the other hand had to be read individually for each of the sockets.

The analog digital converters in the MCU can sample positive voltages from 0 to 5 volts to digital values. Since 230 VAC was to be measured, a solution for an intermediary circuit was needed. This was accomplished by approximating the voltage level on the secondary side of the power supply transformer to be proportional to the primary side where 230 V main voltage was present. By measuring the voltage level on the secondary side of the transformer, the MCU can calculate the corresponding main supply voltage. However, at the secondary side of the transformer is a voltage of 9 VAC. This is AC with no offset, meaning that the voltage shifts between being positiv and negative. This was a problem since the MCU could only measure positive voltages. This was solved by using a voltage divider and adding a DC offset from the microcontroller's supply voltage according to Figure 8. This enabled the voltage to be measured by the analog input of the MCU and an RMS value could be calculated using equation (8).

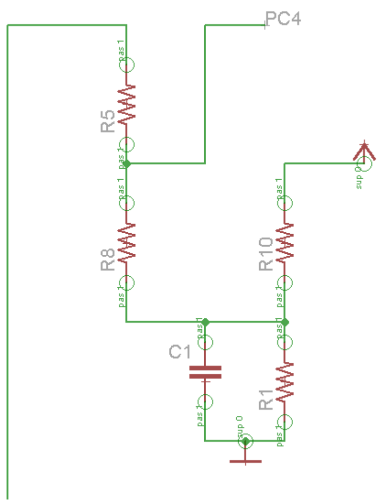


Figure 8: Analog sensor connected to PC4, output from the secondary side of the power transformer to the left.

Since the electricity that flows through the power strip consists of a mixture of high voltages and small electrical signals, it was desired that high voltages were galvanically isolated from signal voltages. This ruled out the option of measuring currents using a shunt resistor without an intermediary isolation amplifier. A solution with an isolation amplifier was dissmissed out as these were large and expensive. Due to that our vision was to fit all of the components in a standard power strip, it was decided that using a current transformer for each socket would take up too much space. This left us with the Hall effect sensor. The Hall effect sensor Allegro ACS712 was chosen as it is well suited for applications using microcontrollers since it outputs a voltage level between 0-5V that corresponds to the flow of current through the sensor. This voltage is then sampled by the microcontroller and an RMS value is calculated using equation (8).

In order to calculate the phase difference the method of measuring samples between two peaks where chosen over using Fourier analysis. A method of using Fast Fourier Transform (FFT) for a single frequency was tested, but did not show any increased accuracy over the zero crossing algorithm. The algorithm was programmed into the micro controller using the theory described in Section 3.1.1.

5.1.4 Power switching

Both electro-mechanical and Solid State Relays (SSR) provide isolation from high voltages and can be used with 230 V. As the circuitry of the Smart Power Strip is intended to be as energy efficient as possible the SSRs are generally more suiting than electro-mechanical relays for the project. SSRs with the specifications required for the Smart Power Strip have proved relatively hard to come by, but in the end the decision was made to use the Sharp S216S02F SSR. This relay can be controlled directly with the MCU and is designed for switching 230 VAC. It is also of a relatively small size compared to many electro-mechanical relays (Sharp, 2004).

5.1.5 Circuit design

To optimize the Smart Power Strip for size, all electrical components were fitted onto one circuit board. The design for this board was made with the Printed Circuit Board (PCB) layout design software CadSoft EAGLE. After all components were placed and routed, the circuit board was etched out and the components were mounted. The decision was made not to buy the board externally to save time and money, and because if an unexpected error should have occurred with the circuit design it would be significantly more economic and time saving to be able to build another board with internal resources.

The etching process begins with the PCB layout being printed on a transparent sheet. A fiberglass laminate covered in a photo resist layer is then exposed to UV light, with the printed sheet as a mask. When the laminate then is immersed in sodium hydroxide ($NaOH$), the areas of the photo resist layer that were exposed to the UV light dissolves, while on the other parts the layer remains to protect the copper layer underneath. The next step is to etch the exposed copper by placing the board in a tank filled with heated sodium persulfate ($Na_2S_2O_8$). When all exposed copper has dissolved, the PCB is ready to be drilled and mounted with components (Varteresian, 2002).

5.1.6 Prototype design

When planning the external casing for the Smart Power Strip prototype, using a conventional power strip to embed the circuits in was discussed. This was, however, not feasible as there would not be enough space for the circuits. Instead a decision was made to use a standard plastic electronics box. An IP-54 classed box was chosen to ensure the users' safety. The IP-54 standard certifies that the box resists dust and splashing water. Conventional power sockets were mounted on the box together with fuse holders and a power plug for the supply cable. The prototype also features an illuminated switch that switches the incoming power line, as well as one three-way switch for each of the power sockets. These switches bring the ability to control the power outlets manually – in case of Internet connection loss – or to choose the remote controlled mode.

5.2 Communication server

The purpose of the communication server is to handle and relay all communication to and from all power strips. To keep the workload of the power strips as low as possible, the communication server takes over the responsibility of inserting data into the database and implement the power saving functionality. When a power strip measures consumption values, it sends them to the communication server without caring for what to do with them next. The communication server in turn makes sure the data is analysed and inserted into the database. For example, if newly received data shows that the consumption of a socket on a power strip is below a given threshold, the communication server will make a decision to to turn it off. This is all done by analysing previously collected data. When a decision is made to turn a socket off, the communication server sends a control signal back to the power strip. This way, the power strip does not need to analyse the data on its own.

One problem that arose from the start was how to be able to send control signals back to a power strip. Most home routers are for example protected with firewalls and NAT. This means that trying to connect to a power strip from the outside would in most cases fail. The workaround for this was to always keep the TCP socket alive from the moment the power strip is turned on. To make sure that a connection to a power strip never goes down, the power strip sends data packets to the communication server with small gaps in between. Not only does this mean that the communication server can send control signals back uninterrupted to a power strip, it also gives the benefit of many data points in the database.

The functional language Erlang became the choice for the implementation of the communication server. This was because of the straightforward way to handle a large number of connections, which is a well founded property of the language itself since it was developed for telecommunications. Other benefits of Erlang include scalability, i.e. the ability to split the workload across multiple machines, which is important on a large scale implementation.

Figure 9 illustrates of the underlying structure of the communication server, modules and functions are represented in red and yellow respectively. The parallelograms represent the data flow trough the system, and the green colored are especially for the incoming and outgoing packets. Also, the blue colored represent the two types of packets that the listener can handle, the data packet that consists of serial id, consumption data and socket status, and the control package that consists of serial id and a control message.

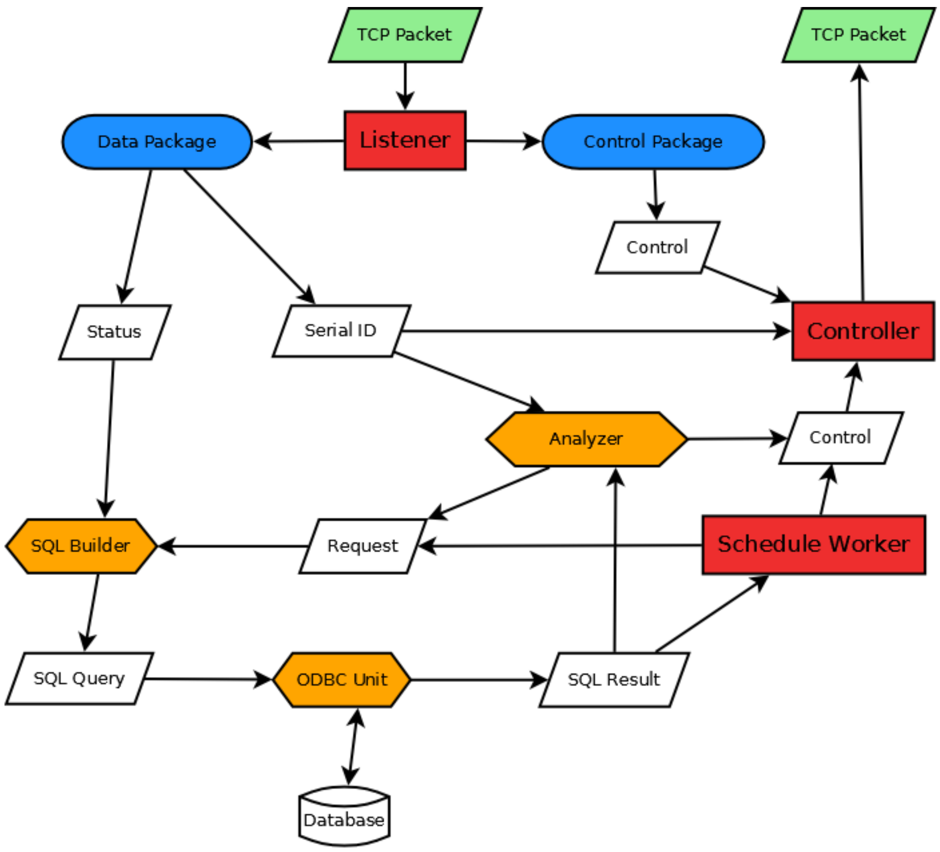


Figure 9: The underlying structure of the communication server.

5.2.1 Modules

In order to optimise scalability and concurrency, the server software is divided into three modules running continuously, which in turn spawns sub-functions in order to handle the data. That way, tasks can easily be divided between different resources. These modules are listed below with a more detailed description and they are represented in red in Figure 9.

Listener
The most top-layered module is the listener. This module has the very simple task of just listening for incoming packets, such as data packets from power strips or a control packet from the web application. When a packet is received, the module determines what type of packet it is and handles it accordingly.

Controller
The controller keeps track of which TCP socket is connected to which power strip by updating a internal list every time a data packet is received. Each message recieved from other modules are matched against this list to retrieve the correct TCP socket and then forwarded to that TCP socket.

Schedule Worker
The schedule worker implements the scheduling functions in the web application. It is activated every five minutes and sends a request to the ODBC Unit to gain the information of which sockets to switch on or off. This list of control messages are then sent to the Controller to be realyed out to the correct power strips.

5.2.2 Functions

In difference to the modules, the functions are only spawned when needed. This is to minimize the amount of processes and therefore minimize system load. These sub-tasks are represented in yellow in Figure 9 with a more detailed described below.

ODBC Unit
The ODBC unit opens a connection to the database, this could be any Database Management System (DBMS) with Open DataBase Connectivity (ODBC) support, and then executes the provided SQL query. The answer is then sent back to the requesting module or function.

SQL Builder
When data needs to be selected or inserted in the database, the SQL Builder is spawned to create the correct type of database query. This SQL query is then forwarded to the ODBC Unit.

Analyser
When a data packaet is recieved the analyzer function is spawned to decide if any socket has a connected device that are in standby mode. In order to do this the last ten minutes of data are requested from the database. This data is then processed and matched against a threshold to determine if any socket on the power strip should be turned off. If this is the case a control message is sent to the Controller.

5.3 Web application and database

This section explains which web application framework and which database that were used, and why they were chosen. The modeling of the database and the implementation of the required features is explained. Additionally, other tools used for creating the web application are mentioned.

5.3.1 Development framework

Creating a web application can be done in many ways, but the most important design choice is what development framework to use. Alternatives for this project were CakePHP, Django and Ruby on Rails. CakePHP's language PHP was the first alternative encountered because of its position as one of the most used languages and its association with web development. Django and Ruby on Rails were encountered when investigating further. When these three alternatives were examined, CakePHP was excluded since Django and Ruby On Rails provided more development tools. Django and Ruby On Rails was similar in benefits for the project, but Django was finally chosen because of the advantage of being able to automatically generate an administrator view and especially for a having documentations that include many examples.

5.3.2 Database Management System and Object Relational Mapper

Since the server uses Erlang an option for a database was Mnesia. However, Django's Object Relational Mapper (ORM) does not support this database and thus it was not a feasible alternative. The framework supports both MySQL and PostgreSQL. Since these databases are considered to be suitable for the project and the ORM is desirable to use, no other databases have been investigated. Deciding between these databases was difficult since the both of them are similar. MySQL is considered to be a little bit faster, while PostgreSQL has a wider support for the Structured Query Language (SQL). PostgreSQL was finally chosen because of the wider support. If the extra functionality in PostgreSQL is not used and there is a need for better performance, a switch to MySQL can be done.

It is the ORM in Django's framework that makes this switch between databases easy. The database is not modeled in the SQL language, which has only small differences between different databases. Instead it is modeled in a python file that is interpreted by Django's ORM and translated into SQL code for the chosen database. Translation of the text strings in the database to objects in the web application is an even greater advantage that the ORM provides. This allows the developers to spend valuable time on other tasks than integrating the web application with the database.

The database created for this project is illustrated in Figure 10. It consists of 5 main tables: user, powerstrip, socket, consumption and group. All tables have an id attribute as their primary key. Tables for the scheduling feature has also been created, but are not shown in Figure 10. Instead they are shown in an ER diagram in Appendix D. More tables are used in the database than the ER diagrams show, but these are created and managed by the Django framework. For example, there are tables managing user sessions and user permissions.

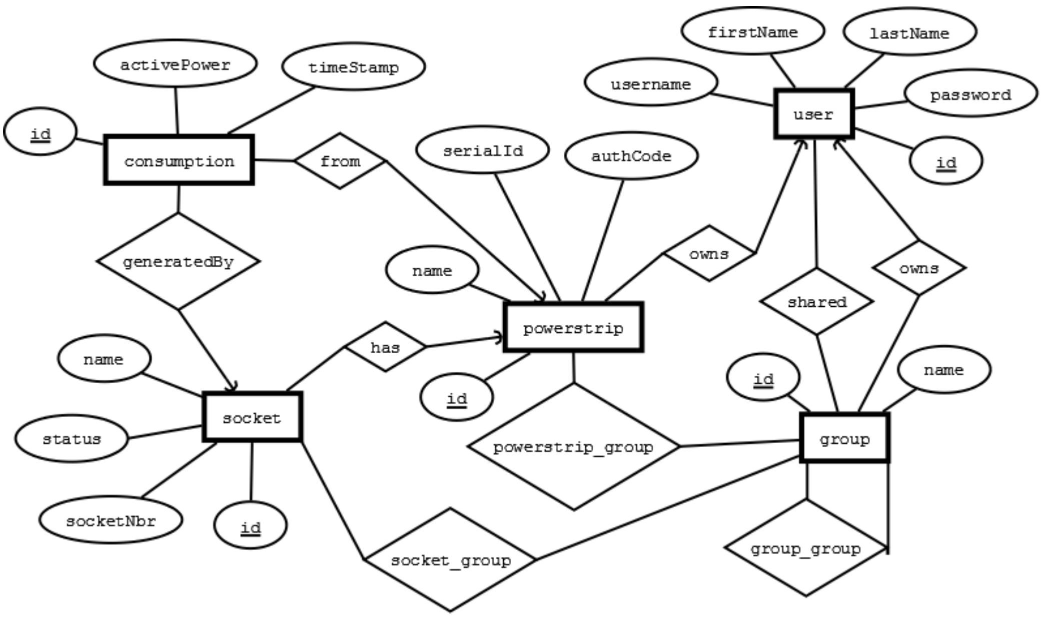


Figure 10: ER diagram for the database without tables for scheduling functionality.

5.3.3 Implemented features

Programming the web application has been about implementing the most important requirements, which can be found in Appendix C and further explained in section 4.4.

User account
The requirements for creating an account and authentication were the first to be implemented. The application is designed so that having an account and being logged in is necessary in order to use it. Users can create accounts for the web application through a registration form. The form contains fields for username, email address and password, the minimal information required to register. The authentication of a user is performed using the built-in authentication feature of Django which facilitated the implementation.

Power strip
Turning on and off sockets is done through the main page of the application. A table of all existing power strips and sockets is displayed, together with a status field and a button for switching on and off. Pressing the button sends a command packet to the communication server, which relays the signal to the power strip.

The addition of a power strip to an account is done through a form in the web application, where users input a serial id and an authorization code. The authorization code is used to verify that a user owns a specific power strip. For a public release all manufactured power strips would already be in the database, so without an authorization code anyone could claim ownership of a power strip if they know the serial id. In order to facilitate the development no power strips are created in advance, instead the power strip is created together with four sockets when a user adds a power strip to their account. These power strips initially have an empty user field that is updated when the function is used.

Renaming a socket or a power strip are done in similar ways. After navigating to a specific power strip or a socket there is a button that takes the user to a page where they can input a new name.

Erasing a power strip is done by entering a specific power strips page and pressing the remove button. This triggers a new page where the user has to confirm that they want to erase the power strip from the account, as an extra precautionary step. When this is done, all consumption data that belonged to that power strip is deleted from the database.

Statistics
Energy consumption statistics are displayed with graphs. When a user enters the consumption page the application begins to fetch consumption data from the database. The graph is not shown until the application has finished

gathering all data. In the meantime an animated loading icon is displayed in the area where the graph will appear.

When all data has been fetched, a graph displaying the consumption for the past 7 days is shown. From this page users can change the time period for the graph, by specifying a start date and an end date through two text fields or with a JavaScript widget that allows more specific choices. A zooming functionality is implemented which allows users to get a more detailed look on the consumption. Initially, the consumption of all power strips combined is displayed, but this can be changed by selecting a specific power strip from a drop-down list. In order to view updated information about the consumption, a manual refresh of the web page needs to be made.

Groups
Users can create groups that contain individual sockets, power strips or other groups through a form. JavaScript and AJAX are used to fetch existing power strips, sockets and groups and then populate drop-down lists with these items as options.

Schedule
A scheduled event can either be a timer or a repeating event. To create a timer a date and a time needs to be defined and the type of action, on or off, need to be chosen. A repeating event requires a date from when it is active, start and end times, a type of action, intervals and possible end dates. In addition, a socket, power strip or a group needs to be chosen in order to create a scheduled event.

5.3.4 Web browser and web server interaction

Navigation of the web application is done by using a web browser. Each time a URL (Uniform Resource Locator) is entered, a request is sent to the web server. The request is processed and an HTML (HyperText Markup Language) page associated with the URL is returned. If the web page uses JavaScript, several JavaScript files could also be returned. JavaScript is code that is executed in the user's web browser, and can be used to change the behavior of the web page. For example it is possible to create a button in HTML that is linked to a JavaScript function. Each time the user presses the button, the JavaScript function is executed.

There are times when new data has to be retrieved from the web server without requesting an update of the whole web page. An example of this is when plotting a graph of energy consumption data. If the user chooses another time interval new data has to be retrieved from the web server. For retrieval of new information without a reload of the page, Asynchronous JavaScript and XML (AJAX) is used in the web browser. AJAX can be used to create a request to be sent to the web server asynchronously, which means that the web browser can continue with other processes while waiting for an answer from the web server. All data sent between the web browser and the web server is in text format, but when programming in JavaScript and Python it is common to use other data structures like arrays and lists. These data structures have to be parsed into ordinary text strings before being transferred. This is done according to a standard format called JavaScript Object Notation (JSON). Since this is a standard format, there already exist parsers in JavaScript and in Python for converting the standard object types into text strings.

5.3.5 User Interface

While Django is used in the web server, the structure and design of the web application is implemented using HTML and Cascading Style Sheets (CSS). Django has a powerful template system that is capable of generating different types of documents, including HTML. With the template system it is easy to implement dynamic pages, which are needed to display user specific information in the interface.

In order to create graphs over the consumption a decision was made early on to use the JavaScript library JQuery. There are already a number of JavaScript libraries that uses JQuery in order to create graphs so there was no need to create a new solution from scratch. After looking around and reviewing different libraries and plugins the choice fell on a library called Flot since it has all the features needed and an easy to use Application Programming Interface (API). Figure 11 illustrates a graph created in the web application. Each staple represents one day's consumption for one power strip, and the four different colors on the staples represent each socket's consumption.

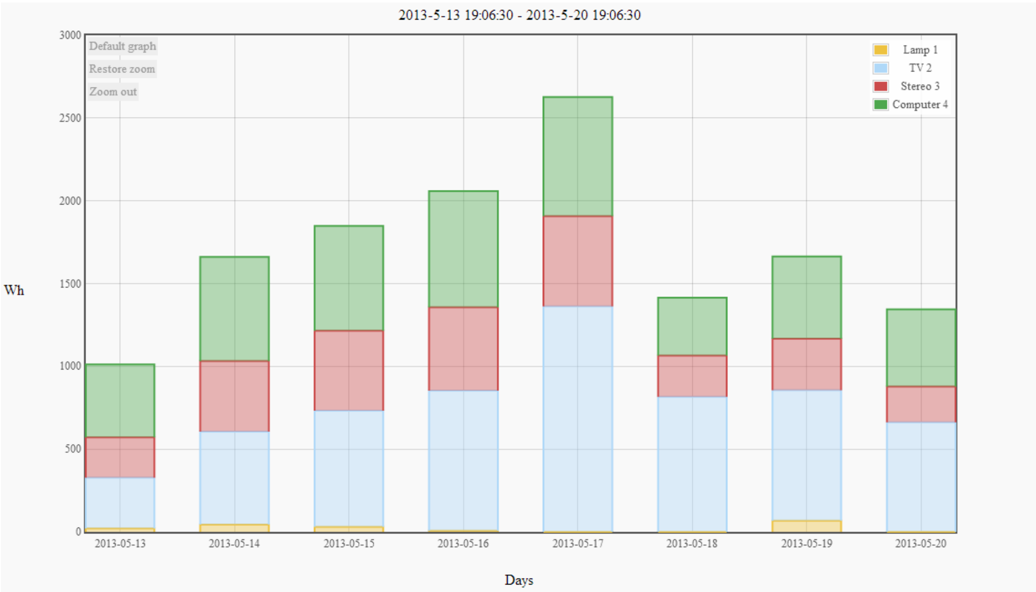


Figure 11: A graph displaying the energy consumption from a few days.

5.3.6 Time zone offset management

All energy consumption data requires a timestamp and therefore the Smart Power Strip needs to keep track of time. Since there is no battery in the power strip there is no way for it to know the time when it is powered on. This is solved by retrieving the correct time from a Network Time Protocol (NTP) server. This time is specified in Coordinated Universal Time (UTC) +00:00, or also called Greenwich Mean Time (GMT). With every consumption data a timestamp is appended before transmission to the server. This data is inserted into the database and the timestamps remain in the same format through the whole system all the way to the web application backend. There it is converted to the user's time zone right before transmission to the web browser for visualization. This means that the Smart Power Strip can be used all over the world and still display the correct time.

5.4 Authentication and communication security

The Smart Power Strip utilizes the Internet for communicating between different parts of the system (see Figure 1 in section 1.2). Since the Internet is an open network used by millions of people there are several security threats.

This section introduces different aspects of security in the Smart Power Strip system.

5.4.1 User authentication

In order to use the web application an account has to be created which contains information about the user. Authentication of the account requires a password to be stored in a database. Saving a password in plain text is considered to be a security risk and therefore each password is sent through hashing, salting and key stretching. The web application uses the built in user authentication that is provided by the Django framework. The hashing algorithm used is SHA-256, and for the key stretching PBKDF2 is used with 10,000 iterations. This is the default settings for user authentication in Django. These methods are recommended by the National Institute of Standards and Technology (NIST) and requires "massive amounts of computing time to break" (Django Software Foundation, 2013). When the user is authenticated a session id will be shared between the user's web browser and the back-end of the web application. This id is then used to keep track of all the users that are logged in.

5.4.2 Communication security

There are data signals used for communication between the different parts of the system and several of these signals require additional security. Between the Smart Power Strip, the web application and the server there are two types of transmissions that are important to secure. The first type is the control signals which are used for turning sockets on and off. The second type is the consumption signals, which are used for transmitting energy consumption data. If no security is added to these signals, it is possible for outsiders to fake them. This would mean that the outsider can control the sockets on the power strip and send false consumption data to the server. It would also be possible for the outsider to collect the data and to summarize it in order to calculate other people's energy consumption. In research conducted by Molina-Markham (2010) it is shown what information can be revealed from energy consumption data. Examples of what can be analyzed are: how many people that live in the household, when people are home, sleeping routines and eating routines. In this research the total energy consumption for the entire household was measured. Depending on what devices that are connected to the Smart Power Strip, both more or less information could potentially be aquired. To avoid these situations the senders must be verified and the consumption signals need to be encrypted.

From the start the security was a secondary requirement because the implementations of features that support a presentation of the prototype were higher prioritized. Since security is still considered to be an important aspect of the project a short research has been conducted on the subject.

It is easy to find a solution for adding encryption to the communication between the web server and the communication server. This is because the programming languages used for communication are Erlang and Python, which have many libraries for encryption. Calculation power in the computers that run the web server and the communication server is not a problem, but the Smart Power Strip's processor limits the use of more demanding encryption algorithms. The best alternative would be to use a combination of an asymmetric cipher and a symmetric cipher. The asymmetric cipher would only be used to share a key that can be used by the symmetric cipher. This would give the advantage of the faster symmetric cipher, but there would still be no need for sharing a key before communication. A suitable encryption method would be the Secure Socket Layer (SSL) protocol, which utilizes both symmetric ciphers and asymmetric ciphers. There is however no SSL library available for the microcontroller's Arduino environment. In fact there are no asymmetric cipher libraries available for the microcontroller at all. But there is a library for the symmetric cipher Advanced Encryption Standard (AES). If the power strips are programmed with a key during manufacture, AES can be used. This key must also be stored in a database connected to the server. If an outsider gets access to any encryption keys, then these keys have to be changed in the power strips and in the database.

6 Evaluation

In this section the project is evaluated based on the purpose, the problem description and the more specific requirements listed in Section 4.

6.1 Power strip

The main goals of the Smart Power Strip prototype were to be able to measure instantaneous power and to remotely switch each power socket. These features are fully implemented in the prototype and are working reliably. The following sections evaluates the more detailed requirements listed in section 4.

6.1.1 Measurement accuracy

The power strip has been calibrated with the instruments available at Chalmers high voltage laboratory. While these instruments are not made for high accuracy, they are accurate enough for calibration of the power strip. Several measurements ensures that the Smart Power Strip is able to measure power down to approximately 11.5 W. This limitation is due to that the current sensing becomes noisier as current magnitudes get low. The increasing noise makes calculations of both RMS and phase difference harder, resulting in increasingly unreliable values for low currents.

Since the voltage are measured after the power transformer the waveform becomes slightly distorted due to the nature of the transformer. This distortion affects both the RMS of the voltage as well as the phase difference calculation. From testing it proved that the RMS value is stable enough to not contribute to spike errors and that the error correction could be linearly approximated in the interval between 210-230 VDC.

The phase difference calculations have proven to be sensitive to noise according to the tests. For higher current magnitudes the algorithm provides stable values for 9 out of 10 calculations. The misread values differ about $10 - 15$ % from the correct values. For low current magnitudes in the region of $0.04 - 0.06A$ an increase in faults could be noticed. This is due to that for small currents the algorithm is faced with comparing two noisy wave forms since the current wave gets noisier with decreasing amplitude. However there are scenarios where the current wave gets distorted for other reasons like driving reactive loads. This also makes the phase difference calculation less reliable. There is a small delay between the sampling of current and voltage. This has been measured and compensated for in the calculations of the power usage.

In total, with currents higher than the mentioned background noise the Smart Power Strip is measured to have an accuracy of around 3 W – assuming the phase difference calculation is stable.

Since the power strip updates time from an NTP server (Network Time Protocol) the time stamps that are attached to each of the power readings are accurate.

6.1.2 Unit power consumption

One of the project's secondary goals was to minimize the Smart Power Strip's energy consumption. While most of the components were chosen in favour of energy efficiency there are exceptions. Due to limited options the power transformer is rated for slightly more demanding tasks than this application, which results in additional energy usage. This applies to the standby as well as load energy usage since transformers consume energy even without any load connected to their secondary side.

By using the high voltage laboratory at Chalmers the average energy consumption of the power strip could be measured to 7.6 W. This value could have been decreased by implementing various improvements. For instance the power transformer could be replaced with a switching power supply similar to the ones used in cell phone chargers. This would be more energy efficient and at the same save physical space. The energy consumption could also be decreased by optimising the program. For instance the WiFi module has a sleep mode that it can wake up from in less than 100 milliseconds. This could be used in between each data send/read from the power strip. The sleep mode is not utilized in the current build as quick responses to switching commands are prioritized.

6.1.3 Total unit cost

The most economic or otherwise suitable parts could not always be chosen as the number of available suppliers through the institution was limited. All prices and parts would be much more optimized in a scenario where the Smart Power Strip was redesigned for manufacturing. With the available resources the total cost was optimized as much as possible. The final cost for one prototype was calculated to 1390.53 SEK excluding VAT. A major contributor to the price was the casing and power sockets. These costs would be much lower if a custom case with integrated power sockets and switches was built in an extension of the project. Another significant contributor to the price was the WiFly module.

6.1.4 Memory usage

As the Smart Power Strip features a 16 MB FLASH memory, it can buffer data with timestamps for over two months in case of connection loss to the communication server. As the microcontroller and the WiFi module cannot keep time when not powered, the power strip must always update its time from the Internet after power off, before it can start buffering data. The buffer, on the other hand, is maintained when the device is powered off. In case of there being data in the buffer, the device will try to connect to the server to send the data on power on and in regular intervals. To make the device able to start buffering data without first having updated time from the Internet, a battery driven clock would be needed. This is not implemented in the built prototype.

6.1.5 Electrical safety

The Smart Power Strip prototype is made from an IP-54 classed electronics box. This means that the box interior is protected against dust and splashing water. However, as the box is fitted with power sockets the IP classification is void. The power strip should still be resistant against dust and water splashing as any other conventional power strip. As the unit is a prototype and does not have a custom made sealed casing, it would not be allowed as a final design. In its current state, it should still not be a danger to users when used in a correct way as it is solidly built to withstand short falls and as the electronics are well isolated. The device also meets the requirements regarding electromagnetic interference. In total, the project requirements for safety are met as the device is safe to use under normal scenarios.

6.1.6 Operational reliability

The Smart Power Strip's normal operation was tested for around two weeks without any reliability issues. If the device loses its WiFi, Internet or server connection it starts to buffer data, which is sent when the connection can be reestablished. Remote switching of the sockets is executed quickly; sometimes with delays up to one second if the microprocessor is doing power calculations when the switching command is sent.

6.1.7 Communication encryption

As encryption of the communication between the Smart Power Strip and the communication server was a low priority requirement, this was not implemented as no extra time was available for this purpose. A feasible encryption method would however have been the AES protocol, as described in Section 3.3.

6.2 Communication server

All of the features listed in the requirements section 4.2 have been implemented and can be used when deployed with the web application and the power strip.

The data packets from the power strip are inserted with timestamps into the database trough an ODBC connection, and the control packets from the web application is relayed to the correct power strip.

The server includes a function that runs every time a data packet is recieved. This function analyzes the data history off that power strip and turns off sockets if the connected device is in stand by.

Another process is executed every five minutes. This process checks if any scheduled activity is to be executed at the time and sends control packets to the relevant power strips.

6.2.1 Reliability

The main server process monitors the child processes (analyzer, listener, controller and schedule). This means that if one of the child processes goes down, the main process will quickly restart this process to minimize downtime. To the end user this may result in having to do a request one more time or in the loss of a data value, but in comparison to a whole server restart this is very quick and effective.

6.2.2 Scalability

As there has been no need to scale up the server in the prototype stage, no scaling has been done. The server can handle up to a hundred connections without any problem and thus the workload does not need to be divided. One scaling option is hardware scaling which requires the purchase of a more powerful dedicated server machine with more RAM and CPU cores. This is a simple fix and quite expensive because of the need to buy a more powerful machine every time an upgrade from the old server is needed. The best solution would be to combine Erlang's built in node distribution, which enables the ability to run processes or functions on different physical machines with hardware scaling as in Figure 12. This basically means that a less powerful machine can be bought to handle a part of the workload, thus reducing the cost of each upgrade and minimizing the amount of discarded hardware.

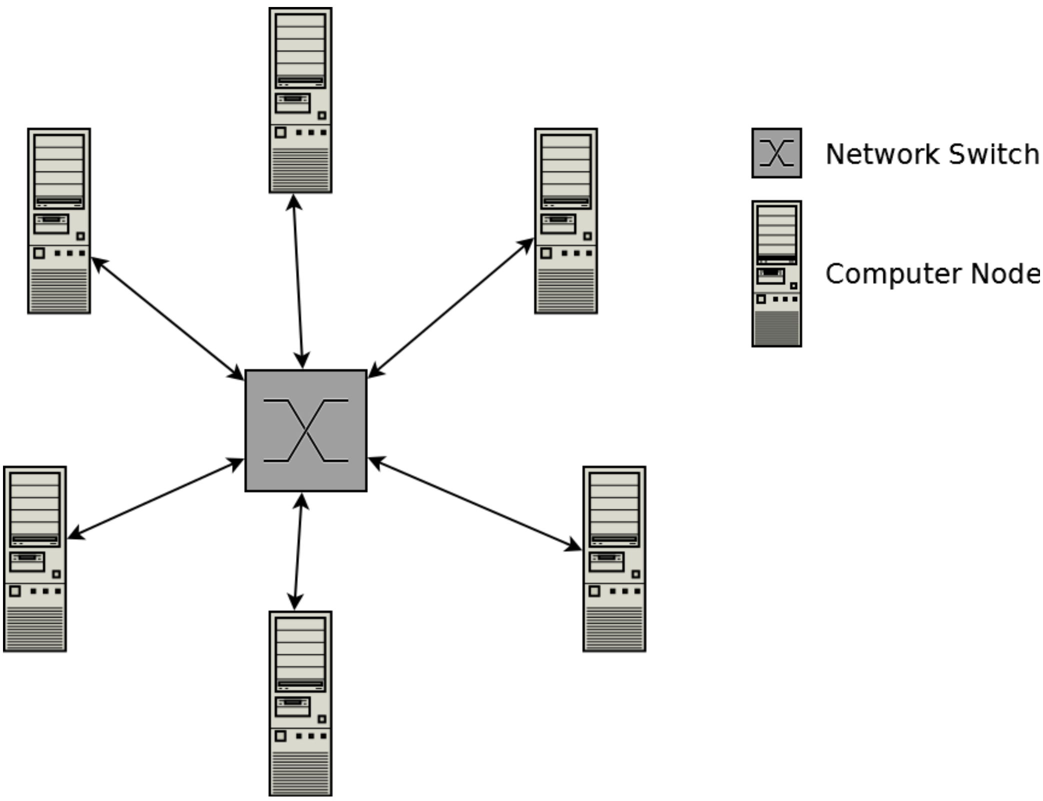


Figure 12: An overview of a node network where each computer node can handle a part of the total software load

6.2.3 Real-Time

It is hard to measure if a program runs in real-time because of the many aspects that are involved. However, for this kind of software the end user experience is the most important. The delay when sending a control signal from the web application to the power strip is so small that it feels instantaneous. This also works for the scheduling part of the server as the server can pre-load the next control packets from the database and therfore send them out with no delay.

6.3 Database

The priority when designing the database has been on creating a working database for the demonstration of the Smart Power Strip prototype. The database has been tested for a short period and has been working correctly during these tests. However, it is already obvious that the database has to be improved in case of a scenario when the product is released for commercial use. The area in greatest need of improvement is the consumption table which contains active power, timestamps, sockets and power strips. The power strip sends the active power for each socket approximately every 10 seconds and the communication server will insert this data into the database. Equation 15 shows how many table rows that will be inserted for one power strip with four sockets. Even if only a small number of Smart Power Strips are used, the consumption table would soon become unmanageable.

$$\begin{aligned} 4 \cdot 6 \cdot 60 \cdot 24 &= 34,560 \; rows/day \\ 4 \cdot 6 \cdot 60 \cdot 24 \cdot 31 &= 1,071,360 \; rows/month \\ 4 \cdot 6 \cdot 60 \cdot 24 \cdot 365 &= 12,614,400 \; rows/year \end{aligned} \tag{15}$$

The first indication of low performance will be noticeable when plotting graphs in the web application. When a user requests a summary of the past six months' consumption, it will take a long time before an answer is delivered. This is because there are many rows to retrieve and iterate over. If a request from one user requires many calculations by the server, requests from many users simultaneously would easily cause the server to overload.

6.4 Web application

All of the features that were considered to be high priority have been implemented and can be used through the web application. Although they are implemented, that does not imply that they are optimized and working perfectly. For example:

- The format of the inputted data in forms is not controlled. This means for example that a user can input an invalid email address.
- Zooming in the graph is possible, but not panning. This forces the user to manually adjust the time interval for the graph.
- If a wide time interval is entered when plotting a graph, a large number of datapoints have to be transferred and displayed which results in a delay and an indistinct graph.
- Grouping power strips, groups and sockets is possible. However, it is not possible to use the groups when plotting graphs or switching sockets on and off.
- Scheduling power strips and sockets is possible, but no view for displaying scheduled events have been created. This means that if you have added a scheduled event it cannot be removed.

While no detailed requirements for the design of the web application was created, effort has been made to make the interface as clear and as user-friendly as possible. The menu provides a clear overview of the structure and where the different content is located. The first thing users encounter when they log into the web application are all the power strips and the status of all the sockets on them. From here users can turn sockets on and off and since this is one of the most important features having it easily available is important. Then the only way of knowing that something has been added or removed is to see if the tables that lists these things are updated. This could be improved by displaying information messages on the page that the users are redirected to after performing these actions. In order to fully evaluate the interface user studies need to be made, but this was not within the scope of this project.

6.5 User authentication

The implementation of the authentication was easy because of a library provided by the Django framework. It was one of the first functionalities implemented, and during the development of all other features no problems have occurred because of the authentication. Still, all other features require the user to be authenticated. Tests conducted until now have only included a few users at the same time and in the event of a release further testing has to be conducted. One factor that increases the likelihood for an overload is the key stretching used for password protection, which extends the time each password comparison takes. However, the increase in security compensates for the decrease in performance.

7 Discussion

The discussion section begins by discussing the Smart Power Strip system solution and the overall personal experience of the project outcome. Following this are sections that in more detail discusses the system parts.

The system structure that is the basis of this project was agreed on in the earliest stage of the planning. This is the system that includes the Smart Power Strip, the communication server and the web application. All research have then been done under the assumption of this system structure. However, a few other options were discussed during the planning. Instead of having the Smart Power Strip connect to a remote server over the Internet, a server could be placed in the user's residence. The power strip would then connect to either a computer where a server has been installed, or to a separate server unit shipped with the Smart Power Strip. Using the user's personal computer as a server would require the user to have the computer on constantly. As the aim of the project was to encourage the user to save energy, a solution which forced the consumer to have a computer always running was not feasible. As a result, it was decided that the Smart Power Strip should connect to a remote server. Maintaining a central server can however be costly and may require a higher product price or a subscription fee.

One of the purposes behind the Smart Power Strip system is to encourage users to lower their energy consumption. To evaluate this, a large study with live tests would have to be done. This is naturally outside of this project's time frame. Even so, our personal experiences gained through testing the system are that the information of how much energy different home electronics consume can be quite enlightening. Usually one might not think much of what products consume the most energy. With this information given through the web application it becomes easy to see where energy can be saved. With graphs that summarize over different time periods one can also realize which products that consumes the most over long-term. This increases awareness and may result in more energy efficient choices when buying new products.

7.1 Power strip

When designing the hardware of the Smart Power Strip various obstacles were overcome. High voltage circuits of $230VAC$ were integrated into the same circuit board as small signal voltage components. Multiple modules such as

WiFi, FLASH memory, relays and Hall effect sensors have been successfully integrated into a single unit centered around a microcontroller. In order for the different modules of The Smart Power Strip to work together there were also programming related issues that had to been solved. One particularly daunting task was to program the microcontroller to send data packets to the communication server via the WiFi module in a correct way.

While most of the components proved to be well suited for building the Smart Power Strip some choices could have been made differently. The most evident improvement in order to lower the energy consumption of the Smart Power Strip would be to exchange the power supply transformer for a switched alternative similar to those used in modern cellphone chargers. This would also decrease the physical space needed inside the power strip.

The microcontroller proved to be capable of connecting the different modules and at the same time sampling the electrical waveforms. However, there could be performance gained by choosing a more powerful alternative. A microcontroller with more computational power would provide more RAM and a faster CPU. This would give the opportunity to sample more values of the voltage and current. With more samples and a faster CPU the algorithm for calculating the phase difference would be more accurate and provide more stable readings.

For power switching the SSR, $Sharp S216S02$, is a good choice. Compared to other alternatives it is small and easily connected to a microcontroller. However, it proved to get hot when currents of around 5 A flowed through it. In future implementations this would have to be taken care of by attaching heat sinks to the back of the relays.

The Hall effect element, $Allegro ACS712$ benefits from its small size compared to current transformers and the way it is connected to the MCU compared to a shunt resistor. From tests it provided stable readings from $0.01A$. This limitation would probably not be lowered by choosing a current transformer which tends to be even more unstable for low magnitude waveforms.

The voltage reading suffers from getting distorted in the power transformer which affects the entire energy calculation. A way to increase the accuracy of this process would be to use another alternative for attenuating the voltage from $230V$ to a measurable magnitude. This could be done by using a resistive power attenuator similar to those used for lowering the output of a guitar amplifier. Another option might be to use a voltage divider. This would however not provide galvanic isolation from the $230V$ mains. If a switched power supply is chosen over a power transformer one of these alternatives would have to be chosen since the output of a switched power supply is perceived as DC.

7.2 Communication server

To make the underlying structure of the communication server even more scalable, the on-need spawned tasks, such as the ODBC-Unit, analyzer and SQL-builder, could be converted into always running modules. This would be a good improvement in a very large scale scenario when the communication server have to be able to handle millions of simultaneous connections. However, this would require an advanced message passing solution throughout the whole program.

Furthermore, some testing in larger scale would be preferred. Both to examine what kind of hardware that would be required to run the server when utilized in a thorough manner and to examine how good the scaling would perform when tested in a node network configuration as in Figure 12.

7.3 Database

To satisfy the requirements of the database is not easy. It would require a large number of tests and optimizations before the database would be ready for commercial use. As described in the evaluation (Section 6.3), the amount of data in the consumption table (Figure 10) is a problem. One solution could be to increase the time between each measurement, but less data gives a lower resolution when plotting graphs, which is undesirable. This would also result in the web application becoming less responsive as updates of present power usage would arrive with more delay. Instead of lowering the resolution from the beginning, this can be done later when the resolution becomes less important. Already after a couple of days the need for viewing data with minute precision becomes low. The data could instead be summarized to one row for each hour, which would mean 96 rows/day (Equation 16) instead of 34,560 rows/day (Equation 15).

$$
\begin{aligned}
4 \cdot 24 &= 96\ rows/day \\
4 \cdot 24 \cdot 31 &= 2,976\ rows/month \\
4 \cdot 24 \cdot 365 &= 35,040\ rows/year
\end{aligned}
\tag{16}
$$

After a couple of months it might even be possible to summarize for each day without lowering the usability. This would result in 1,460 rows/year (Equation 17) compared to the 12,614,400 rows/year (Equation 15) generated in the current implementation.

$$
\begin{aligned}
4 \cdot 1 &= 4\ rows/day \\
4 \cdot 31 &= 124\ rows/month \\
4 \cdot 365 &= 1,460\ rows/year
\end{aligned}
\tag{17}
$$

This would mean a large increase in performance and probably not make any substantial difference in usability. However, if it does the data would have to be summarized without erasure of the high resolution data. This is because the web application cannot summarize that large amount of data fast enough when a request is received. Both these options would require a daemon that summarizes and organizes the data. Iteratively organizing the data in this manner would be more difficult to implement, but the increase in performance is necessary for the Smart Power Strip system.

7.4 Web application

A great advantage with the Smart Power Strip system is that given the basic functionalities of consumption measurements and power socket switching, it is possible to utilize the software to expand the functionalities. There are several possible improvements and extensions that could be implemented either in house or by external developers by using an API, see Section 7.6. One extension would be to enable users to compete and compare consumption data with each other, which may motivate people to a more energy saving behavior. This can be accomplished by adding this functionality to the web application and also by integrating the application with Facebook and other social media services.

Gaming platforms such as Playstation 3, Xbox 360 and PC have achievements or trophies, rewards that the user receives for completing different objectives. The purpose is to increase the competition between users and create a community around each game. A feature such as this could also be integrated into the web application to enforce the motivation to save energy.

Consumption in the form of graphs are good for visualization and for the comprehension of the user, but this may not be applicable on everyday living.

Therefore, an improvement would be to let the user see how much money they can save by being environmentally friendly. Translating the consumption into money could further enforce an environmentally friendly behavior. To do this either the user specifies their electricity price or the current price is automatically downloaded from the Internet.

Power strips with a master/slave relation already exist on the market, but the technology is in the hardware and therefore not customizable for the user. This could be software implemented in the web application, which enables the user to select one socket as master and add several sockets as slaves. The slave sockets will automatically be turned off when the master socket is turned off. This functionality could be extended with an advanced option where computer logic could determine which sockets that should be on or off depending on the state of other sockets. This means that the status of one socket could be determined by several other sockets independently of which power strip the sockets belongs to.

Sometimes users might want lights or other electronic devices to be regulated by sunrise and sunset. This could be implemented by retrieving time for sunrise and sunset from a weather service and then allow the user to schedule on and off according to these times.

7.5 Communication security

The microcontroller in the power strip has too little processing power to be able to manage asymmetric encryption. One alternative would be to investigate if there are any hardware modules that could be added to make this possible. If no such hardware could be found, the next option would be to upgrade the microcontroller itself. However, this will probably also mean a higher price and higher energy consumption. One of the goals with the Smart Power Strip is to make households more aware of their energy usage and help them reduce their consumption. This is why any change to the product that makes it more energy consuming has to be carefully considered.

If no hardware changes are possible or not considered to be appropriate, then AES could be used. It runs on 8-bit processors and there is an existing library for the microcontroller. Using a symmetric encryption would mean that the Smart Power Strip must be programmed with a secret key during manufacture. This key must also be stored in a database that only the communication server can access. But if someone managed to perform an attack against the server and got access to the encryption keys, then all power strips' encryption keys would have to be changed. If no way to change the key is implemented in the software, then the only way to change it is to send all power strips back to the manufacturer. This would require large resources to resolve. However, this can be considered as a worst case scenario. If the database is used with caution and security is prioritized, the chance for this scenario to occur would be small. Adding a GUI for the user to interact with the Smart Power Strip in order to change the key would also prevent this situation.

7.6 API

When developing a product such as the Smart Power Strip system it is important to take into consideration how other developers and users may want to use the product. To help with further development and improvement an Application Programming Interface (API) could be released. The API is an interface and a library of functions that are open for others to use. For example, the Smart Power Strip system could be extended to support more types of hardware devices. These could use the system's data input possibilities with sensors for temperature, light, motion or similar, or they could also take advantage of the ability to control devices with the web application. An example of the latter could be to control a lamp that changes colour depending on the users energy consumption. If the web application implemented a way to add different types of devices manually, users could even add their own sensors provided they could communicate with the communication server.

The following subsections discuss the possibilities of using an API with the separate modules of the system.

7.6.1 Power strip

The power strip uses TCP/IP to communicate with the server. It sends packets with data to a specified server and can receive packets with control commands. As of now the server address is hardcoded into the power strip, but the device does not care from where the control commands are sent. This means that a separate application to control the power strip can be constructed if the correct IP address is known, although this will probably be prohibited when encryption is implemented and when all traffic have to go trough the communication server.

7.6.2 Communication server

As of now all communication to and from the power strip have to pass through the communication server. It has an interface that handles communication from both the web interface and the power strip. Control commands are relayed from the web application to the correct power strip and from the power strip data packets are received and put into the database. The development of an interface for Java programming has started in cooperation with another project. This lets Android or Java developers create applications that can use most features available in the web application. In the future an interface towards iPhone application development might be considered.

7.6.3 Web application

At this time no API has been developed in order to allow third-party developers to connect to the web application. It is possible to build one manually in Django, however there are third party frameworks such as Tastypie (Tastypie, n.d.) and Django REST Framework (Django REST Framework, n.d.) that make the process easier and faster.

8 Conclusion

The purpose of the project was to develop a smart power strip that would help people reduce their energy consumption. The report presents the development of the Smart Power Strip system and shows a successful implementation of all primary features. The built prototype is able to measure energy consumption and send that data to the communication server. The web application presents clear graphs over the users' energy consumption during different time periods, and provides the ability to remotely control the sockets of the power strips. Communication of both data packets and control signals are stable and quick despite the low processing power of the Smart Power Strip. This is because of the fast and reliable software of the communication server and the power strip.

The communication server has been stress tested and is able to handle up to 100 simultaneous connections without experiencing any decrease in performance. This test was conducted without utilizing the server's scalability features, which would have increased the performance and enabled the server to handle a vast amount of connections.

It is important to remember that when carrying out an interdisciplinary project such as this, several factors have to be taken into consideration. Compared to a project that focuses on one specific field of research, this type of project requires a significant amount of communication and time for assembling the different system parts into one system. The project included many parts of the disciplines of electrical engineering and computer science, and thus required a great amount of work for the creation of a complete system. This is time that otherwise could have been used for implementing additional features. The highest prioritized feature that would have been implemented in an extension of the project is encryption of data and control packets to and from the Smart Power Strip. It is clear that intruders can learn much about the users' behaviour and habits from acquiring consumption data.

Although no large scale user tests were conducted, it was clear from in-house testing that viewing one's energy consumption from just a few days led to the realization of which home electronics that consumed the most energy. With this information the scheduling functionalities have a great potential of saving energy by shutting off electronics with high stand-by consumptions. Overall, the Smart Power Strip system has proven to be working well by providing users with a tool for controlling their energy consumptions.

References

Bartnicki, A. (2012) *ELTEKNIK*. Gothenburg: The institution of Energy and Environment Research, Chalmers University of Technology

Bishop, A. (1997) *Solid-State Relay Handbook*. Pearson Education

CakePHP (2013) *DataSources* `http://book.cakephp.org/2.0/en/models/datasources.html` (2013-04-10)

Chen, C. (2005) Top 10 Unsolved Information Visualization Problems. *Computer Graphics and Applications, IEEE*, vol. 25, 4, pp. 12-16. DOI: 10.1109/MCG.2005.91

Django REST Framework (no date) *Django REST Framework - APIs made easy*. `http://django-rest-framework.org/` (2013-05-18)

Django Software Foundation (2013) *Django documentation* `https://docs.djangoproject.com/en/1.5/` (2013-04-10)

Ekonomifakta (2011) *Elproduktion med fossila bränslen i EU* `http://www.ekonomifakta.se/sv/Fakta/Energi/Energibalans-internationellt/Elproduktion-med-fossila-branslen/` (2013-02-21)

Energimyndigheten (2012) *Energiläget i Siffror* `http://energimyndigheten.se/Global/Statistik/Energil%C3%A4get/Energil%C3%A4get%20i%20siffror%202012.pdf`

Eon (2007) *Lilla energisparboken* `http://www.eon.se/upload/eon.se/dokument/privatkund/energiradgivning/2/Energisparboken_slutlig.pdf`

Herman, I, Melancon, G and Marshall, M S. (2000) Graph Visualization and Navigation in Information Visualization: a Survey. *IEEE Transactions on Visualization and Computer Graphics*, vol. 6.

Huang, M.L. Liang, J. and Nguyen, Q.V. (2007) A Fast Algorithm for Balanced Graph Clustering. *Information Visualization, 2007. IV '07. 11th International Conference*, pp. 46-52. DOI: 10.1109/IV.2007.10

Huang, M.L. Liang, J. and Nguyen, Q.V. (2008) A Usability Study on the Use of Multi-Context Visualization. In *Fifth International Conference on Computer Graphics, Imaging and Visualization*, pp. 311-316 DOI: 10.1109/CGIV.2008.33

Kizza, J.M. (2009) *A Guide to Computer Network Security*. [Electronic] London: Springer

Knudsen, L.R and Robshaw, M.J.B. (2011) *The Block Cipher Companion*. [Electronic] Berlin, Heidelberg: Springer

McDonnel, B. and Elmqvist, N. (2009) Towards Utilizing GPUs in Information Visualization: A Model and Implementation of Image-Space Operations. *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, 6, pp. 1105-1112. DOI: 10.1109/TVCG.2009.191

Moere, A V., Tomitsch, M., Wimmer, C., Christoph, B. and Grechenig, T. (2012) Evaluating the Effect of Style in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, 12, pp. 2739-2748. DOI: 10.1109/TVCG.2012.221

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*, A. Ruzzeli, November 03 - 05, 2010, pp. 61-66. New York: ACM. DOI: 10.1145/1878431.1878446

Olofsson, K. (2013) *Elsäkerhetsverket: Elsäkerhet för produkter.* `http://www.elsakerhetsverket.se/sv/Produktsakerhet/Produkter/Elsakerhet_for_produkter/` (2013-04-25)

Oppenheim, A.V. Willsky, A.S. (2011) *Signals and Systems Second Edition* Cambridge, Massachusetts

Oppliger, R. (2009) *SSL and TSL - Theory and practice.* [Electronic] Boston, London: Artech house

Plekhanova, J. (2009) *Evaluating web development frameworks: Django, Ruby on Rails and CakePHP.* Philadelphia: Institute for Business and Information Technology, Fox School of Business, Temple University. (10 Apr. 2013)

Popovic, R. S. (1991) *Hall effect devices : magnetic sensors and characterization of semiconductors.* Bristol: Hilger

Purer, K. red. (2009) *PHP vs. Python vs. Ruby – The web scripting language shootout.* July 18, 2009, Vienna

Ramsden, E. (2006) *Hall-Effect Sensors - Theory and Application.* 2nd Edition. Elsevier.

Ruby on Rails (2013) `http://guides.rubyonrails.org/getting_started.html` (2013-04-10)

Schwaber, K. and Sutherland, J (2011) *The Scrum Guide.* `http://scrum.org/Portals/0/Documents/Scrum` (2013-02-14)

Sharp (2004) *S216S02F Datasheet.* `http://sharp-world.com/products/device/lineup/data/pdf/datasheet/s116s02_e.pdf` (2013-03-02)

Sinclair, I. (2000) *Passive Components for Circuit Design.* Boston: Newnes

Statistiska centalbyrån (2013) *Antal hushåll i Sverige med olika beräkningsmetoder* http://www.scb.se/Pages/TableAndChart____146283.aspx

Sullivan, B. and Liu, V. (2012) *Web Application Security - A beginner's Guide.* [Electronic] New York: McGraw-Hill

Svenskenergi (2010) *Den Svenska Elens Miljöpåverkan* http://www.svenskenergi.se/Global/Dokument/publikationer/Bild-Den-svenska-elens-miljopaverkan.pdf

Tastypie (no date) emphTastypie - RESTful APIs for Django http://tastypieapi.org/ (2013-05-18)

TIOBE Programming Community Index (2013) http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html (2013-05-13)

Tuzzio, N. and Tehranipoor, M. (2012) *RSA: Implementation and Security.* [Electronic] New York: Springer

Varteresian, J. (2002) *Fabricating Printed Circuit Boards.* New York: Newnes

A Development tools

Aptana Open source web application development tools
`http://www.aptana.com/`

Arduino IDE Open-source Arduino development environment
`http://arduino.cc/en/main/software`

CadSoft EAGLE Electronic design automation software
`http://www.cadsoftusa.com/`

Google Chrome DevTools Debugging tool bundled up within the Google Chrome web browser
`http://developers.google.com/chrome-developer-tools/`

Eclipse IDE for Java EE Developers Source code editor with integrated development environment for Java
`http://www.eclipse.org/`

Git Open source distributed version control system
`http://git-scm.com/`

GNU Emacs Multi purpose source code and text editor
`http://www.gnu.org/software/emacs/`

pgAdmin Open Source management, development and administration tool for PostgreSQL
`http://www.pgadmin.org/`

Sublime Text Source code editor with Python API
`http://www.sublimetext.com/`

B Price/Unit calculation

Product Description	Price	Quantity	Total
ATMEGA Microcontroller	34,04 kr	1	34,04 kr
Crystal 16.0MHz	8,66 kr	1	8,66 kr
Voltage Regulator 5.0V	5,89 kr	1	5,89 kr
Voltage Regulator 3.3V	13,97 kr	1	13,97 kr
Switch	3,01 kr	1	3,01 kr
Hall Effect Sensor	38,97 kr	4	155,88 kr
Transformer	53,75 kr	1	53,75 kr
SSR Relay	43,98 kr	4	175,92 kr
Terminal Block	10,39 kr	1	10,39 kr
Terminal Block	5,25 kr	1	5,25 kr
Pin Header	2,04 kr	1	2,04 kr
Flash Memory	31,78 kr	1	31,78 kr
WiFi Module	270,05 kr	1	270,05 kr
3-Way Switch	28,68 kr	4	114,72 kr
Plastic Box	103,62 kr	1	103,62 kr
Schuko	89,63 kr	4	358,52 kr
Illuminated Switch	33,66 kr	1	33,66 kr
Capacitor 470 uF	9,38 kr	1	9,38 kr
		Cost/Unit	**1390,53 kr**
		With VAT	**1738,16 kr**

C Web application requirements

Below we have chosen to show the priority in the interval high to low meaning;

High Things we must have or else it wouldn't be a product.

Medium Things we need for it to be usable.

Low Things we want to have because it would make the product better.

C.1 Functional

C.1.1 User account

1.00 Create account
Priority: High
Description: The user should be able to create an account.

1.01 Edit account
Priority: Medium
Description: The user should be able to edit information for their account.

1.03 Authenticate
Priority: High
Description: The user should be able to authenticate in order to access their account.

1.04 Add a residence
Priority: Medium
Description: The user should be able to create one or more residences and add power strips to each residence.

1.05 Move power strip between residences
Priority: Medium
Description: The user should be able to move a power strip from a residence to another.

1.06 Share power strips
Priority: Medium
Description: There might be several users with different accounts living in the same residence. This should be able to share power strips.

1.07 Share residence
Priority: Medium
Description: There might be several users with different accounts living in the same residence. They should be able to share residence.

C.1.2 Power strip

2.00 Add a power strip
Priority: High
Description: The user should be able to add a power strip to their account in order to control it.

2.01 Power on and off
Priority: High
Description: The user should be able to power on and off all the devices that are connected to their own power strips.

2.02 Name a socket
Priority: High
Description: The user should be able to give a name for each socket in order to keep track of them.

2.03 Erase a power strip from account
Priority: High
Description: If the user wants to erase a power strip from his account, he should be able to do that. All the consumption data generated for that power strip will be erased.

2.04 Remove a power strip from account
Priority: Medium
Description: If the user wants to remove a power strip from one account, then the consumption data generated by the power strip should not be removed.

2.05 Name a power strip
Priority: High
Description: The user should be able to give a name for each power strip in order to keep track of them.

C.1.3 Statistics

3.00 See consumption
Priority: High
Description: The user should be able to view their consumption generated by their own power strips.

3.01 See consumption for groups
Priority: Medium
Description: The user should be able to see their consumption for their own groups.

3.02 Display graphs
Priority: High
Description: The user should have the functionality to display some of their data in graphs.

3.03 See summary over all data
Priority: Medium
Description: The user should be able to view a summary over the data collected,

3.04 Compare data over time
Priority: Low
Description: The user should be able to compare their data with their older data.

3.05 Compare data with friends and neighbourhood
Priority: Low
Description: The user should be able to compare their data with other users from the same neighbourhood, if they have selected to share their data.

3.06 Get price information
Priority: Low
Description: The user should be able to display an approximately price for their consumption.

C.1.4 Groups

4.00 Create groups
Priority: High
Description: The user should be able to create groups that contain power strips, sockets or other groups.

4.01 Turn on and off groups
Priority: Medium
Description: The user should be able to turn on and off all of their sockets in one group, at the same time.

4.02 Edit groups
Priority: Medium
Description: The user should be able to edit their own groups.

C.1.5 Schedule

5.00 Schedule
Priority: Medium
Description: The user should be able to schedule on and off for their power strips, sockets and groups.

5.01 Auto on and off
Priority: Low
Description: The user should be able to tell the system to turn off sockets according to the consumption in the socket and according to what time it is.

5.02 Schedule according to sunset/sunrise
Priority: Low
Description: The user should be able to schedule on and off according to when the sun rises or sets.

5.03 Time interval for random scheduling
Priority: Low
Description: The user should be able to set, instead of an exact time, a time interval. The socket will then be turned on or off in a random time within the interval.

5.04 Master and slave
Priority: Low
Description: The user should be able to set a socket as a master and tell which of his own sockets that are slave. When the master socket consumption goes up, turning on the device, the power in the slaves will be turn on. And when the master socket consumptions goes down to standby the slaves will be turned off.

C.1.6 Communication

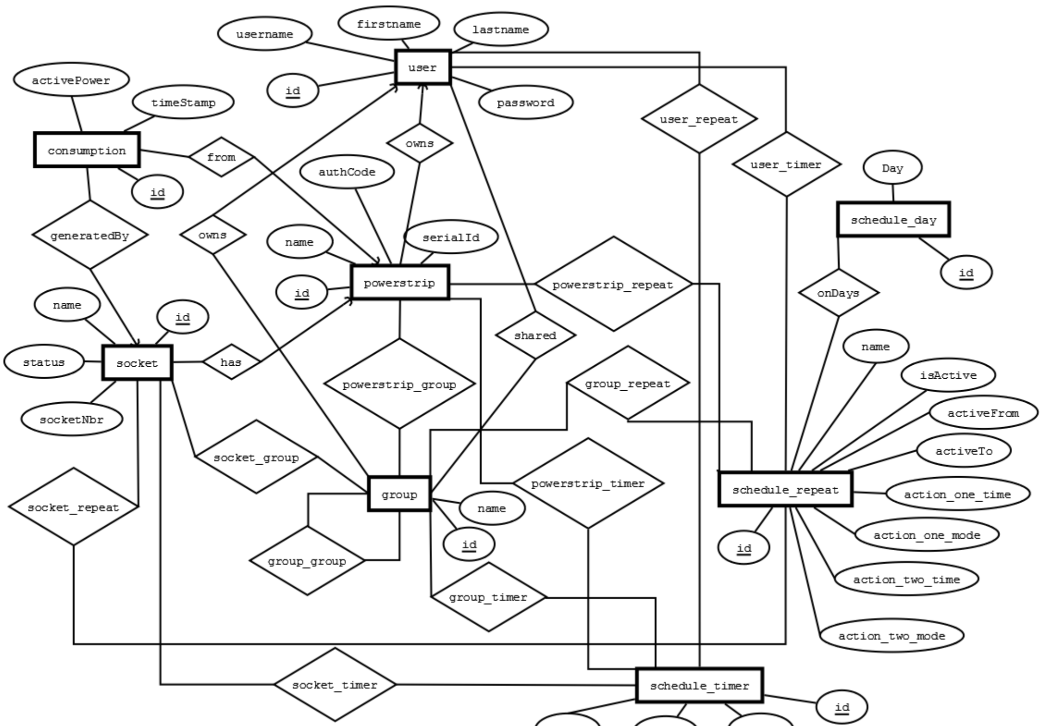
6.00 Security
Priority: Medium
Description: The communication with the server should be done securely

C.2 Non functional

C.2.1 Power strip

6.00 Turn on and off instantly
Priority: High
Description: When the user sends on or off signals to the power strip's sockets they should, on average, have switch power mode (on/off) within 3 seconds.

D Database ER-diagram



An ER diagram for the database with the scheduling functionality.