Technical University of Košice
Faculty of Electrical Engineering and Informatics

Multi-purpose system for measuring electrical power supplied by electric sockets

Master's Thesis

2013 Peter Babič

Technical University of Košice
Faculty of Electrical Engineering and Informatics

Multi-purpose system for measuring electrical power supplied by electric sockets

Master's Thesis

Study Programme:	Infoelectronics
Field of study:	5.2.13 Electronics
Department:	Department of Elecronics and Multimedia Communications (DTIEE)
Supervisor:	doc. Ing. Vojtech Čierny, CSc.
Consultant(s):	Ing. Matej Biely, PhD. RNDr. Marián Čierny, DrSc.

Košice 2013 Peter Babič

Errata

Multi-purpose system for measuring electrical power supplied by electric sockets

Peter Babič

Košice 2013

Ak je potrebné, autor na tomto mieste opraví chyby, ktoré našiel po vytlačení práce. Opravy sa uvádzajú takým písmom, akým je napísaná práca. Ak zistíme chyby až po vytlačení a zviazaní práce, napíšeme erráta na samostatný lístok, ktorý vložíme na toto miesto. Najlepšie je lístok prilepi

Forma:

Strana	Riadok	Chybne	Správne
12	6	publikácia	prezentácia
22	23	internet	intranet

Abstract

Text abstraktu v svetovom jazyku je potrebný pre integráciu do medzinárodných informačných systémov. Ak nie je možné cudzojazyčnú verziu abstraktu umiestniť na jednej strane so slovenským abstraktom, je potrebné umiestniť ju na samostatnú stranu (cudzojazyčný abstrakt nemožno deliť a uvádzať na dvoch strabách).

Keywords

Optimization, thesis, writing

Abstrakt

Abstrakt je povinnou súčasťou každej práce. Je výstižnou charakteristikou obsahu dokumentu. Nevyjadruje hodnotiace stanovisko autora. Má byť taký informatívny, ako to povoľuje podstata práce. Text abstraktu sa píše ako jeden odstavec. Abstrakt neobsahuje odkazy na samotný text práce. Mal by mať rozsah 250 až 500 slov. Pri štylizácii sa používajú celé vety, slovesá v činnom rode a tretej osobe. Používa sa odborná terminológia, menej zvyčajné termíny, skratky a symboly sa pri prvom výskyte v texte definujú.

Kľúčové slová

Optimalizácia, záverečná práca, písanie

Assign Thesis

Namiesto tejto strany vložte naskenované zadanie úlohy. Odporúčame skenovať s rozlíšením 200 až 300 dpi, čierno-bielo! V jednej vytlačenej ZP musí byť vložený originál zadávacieho listu!

Declaration

I hereby declare that this thesis is my own work and effort. Where other sources of information have been used, they have been acknowledged.

Košice, April 24, 2013 . . . .

Signature

Acknowledgement

I would like to express my sincere thanks to my supervisor Dr Vojtech Čierny, PhD, the main Supervisor. Special mention should go to Dr Matej Biely, CSc. for his constant, and constructive guidance throughout the study. To all other who gave a hand, I say thank you very much.

Preface

Predhovor (*Preface*) je povinnou náležitosťou záverečnej práce, pozri V predhovore autor uvedie základné charakteristiky svojej záverečnej práce a okolnosti jej vzniku. Vysvetlí dôvody, ktoré ho viedli k voľbe témy, cieľ a účel práce a stručne informuje o hlavných metódach, ktoré pri spracovaní záverečnej práce použil.

Contents

List of Figures

List of Tables

Acronyms

AC	Alternating Current
ADC	Analog-to-digital converter
AP	Access Point
ASIC	Application-specific integrated circuit
BSD	Berkeley Software Distribution
CLI	Command-line interface
CPU	Central processing unit
DAC	Digital-to-analog converter
DC	Direct Current
DDR	Double data rate synchronous DRAM
DRAM	Dynamic random-access memory
DSP	Digital signal processor
EEPROM	Electrically erasable programmable ROM
FPGA	Field-programmable gate array
GPIO	General-purpose I/O
GUI	Graphical user interface
HDD	Hard-disk drive
HW	hardware
Hz	Hertz, the SI unit of frequency
I/O	Input/Output
I^{2}S	Integrated Interchip Sound
IC	integrated circuit
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol
JTAG	Joint test action group
LAN	Local area network
LED	Light emitting diode
LNA	Low-noise amplifier
LPCC	Quad Flat No-leads
MAN	Metropolitan are network
MB	Mega-Byte
MHz	Mega-Hertz

MIPS	Microprocessor without Interlocked Pipeline Stages
OS	Operating system
PA	Power amplifier
PCB	printed circuit board
PCM	Pulse code modulation
PDA	Personal digital assistant
POSIX	Portable operating system interface
PTC	Positive thermal coefficient
PWM	Pulse-width modulation
QFN	Quad Flat No-leads
RAM	Random-access memory
RDBMS	Relational Data-base management system
RF	Radio frequency
RMS	Root-mean square
ROM	Read-Only memory
RTOS	Real-time operating system
S/PDIF	Sony-Philips Digital Interface Format
SDIO	Secure Digital Input Output
SDR	Synchronous dynamic random access memory
SLIC	Subscriber line interface circuit
SMT	Surface-mount technology
SoC	System-on-Chip
SPI	Serial peripheral interface
SSR	Solid-state relay
SW	software
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
THT	Through-hole technology
UART	Universal asynchronous receiver/transmitter
USB	Universal serial bus
VOIP	Voice over IP
WAN	Wide area network
WLAN	Wireless local area network

List of Terms

android a mobile Operating system (OS) based on the Linux kernel and currently developed by Google, designed primarily for touchscreen mobile devices such as smartphones and tablet computers, and for specialized user interfaces like televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear).

application a program, or group of programs, that is designed for the end user

arduino common term for a software company, project, and user community that designs and manufactures computer open-source hardware, open-source software, and microcontroller-based kits for building digital devices and interactive objects that can sense and control physical devices

cloud (computing) is a model for enabling ubiquitous, convenient, on-demand access to a shared pool of configurable computing resources

command a directive to a computer program acting as an interpreter of some kind, in order to perform a specific task, commonly a directive to some kind of Command-line interface (CLI), such as a shell

compiler a computer program (or set of programs) that transforms source code written in a programming language (the source language) into another computer language (the target language, often having a binary form known as object code)

computer a programmable machine, that responds to a specific set of instructions in a well-defined manner and can execute a prerecorded list of instructions (a program).

daemon a computer program running on the multi-tasking OSes in a background, rather than being under the direct control of an interactive user

datasheet a document that summarizes the performance and other technical characteristics of a product, machine, component (e.g., an electronic component)

driver a computer program that operates or controls a particular type of device that is attached to a computer

ethernet family of computer networking technologies for Local area networks (LANs) and Metropolitan are networks (MANs), conforming to standard Institute of Electrical and Electronics Engineers (IEEE) 802.3

firmware the combination of a hardware (HW) device, e.g. an IC, and computer instructions and data that reside as read only software (SW) on that device, it usually cannot be modified during normal operation of the device

flash an electronic non-volatile computer storage medium (memory) that can be electrically erased and reprogrammed, next evolution of Electrically erasable programmable ROM (EEPROM)

interface a shared boundary across which two separate components of a computer system exchange information that can occur between SW, computer HW, peripheral devices, humans and combinations of these

kernel a computer program that manages Input/Output (I/O) requests from software, and translates them into data processing instructions for the central processing unit and other electronic components of a computer, being a fundamental part of a modern computer's OS

library a collection of programs and SW packages made generally available, often loaded and stored on disk for immediate use

linux an Unix-like and mostly POSIX-compliant computer OS assembled under the model of free and open-source SW development and distribution, from the beginning maintained by Linus Torvalds

memory In computing, refers to the computer hardware devices used to store information for immediate use

network a medium that allows computing devices pass data to each other along links (data connections)

peripherala device that is connected to and works with a computer in a some way, but is not essential to a computer's function

program a specific set of ordered operations for a computer to perform

router a networking device that forwards data packets between computer networks, connected to two or more data lines from different networks

shell a user interface for access to an OS's services, using either CLI or Graphical user interface (GUI), depending on a computer's role and particular operation

stack (protocol) is an implementation of a computer networking protocol suite, used interchangeably

system a set of interacting or interdependent components forming an integrated whole, observing properties not obtainable with individual components

unix a family of multi-taskings, multi-user computer OS that derive from the original AT&T Unix, developed in the 1970s at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others

utility is system SW designed to help analyze, configure, optimize or maintain a computer

Introduction

V úvode autor podrobnejšie ako v predhovore, pritom výstižne a krátko charakterizuje stav poznania alebo praxe v špecifickej oblasti, ktorá je predmetom záverečnej práce. Autor presnejšie ako v predhovore vysvetlí ciele práce, jej zameranie, použité metódy a stručne objasní vzťah práce k iným prácam podobného zamerania. V úvode netreba zachádzať hlbšie do teórie. Nie je potrebné podrobne popisovať metódy, experimentálne výsledky, ani opakovať závery prípadne odporúčania, pozri.

1 Electric power fundamentals

In general physics terms, power is defined as the rate at which energy is transferred (or transformed). Electric energy in particular, begins as electric potential energy – what we commonly refer to as voltage. When electrons flow through that potential energy, it turns into electric energy. In most useful circuits, that electric energy transforms into some other form of energy. Electric power is measured by combining both how much electric energy is transferred, and how fast that transfer happens.

The electric power P is equal to the energy consumption E divided by the consumption time t [20]

$$P = \frac{E}{t}$$

where P is the electric power in watt [W], E is the energy consumption in joule [J] and t is the time in seconds [s].

Electrical Power, in a circuit is the amount of energy that is absorbed or produced within the circuit. A source of energy such as a voltage will produce or deliver power while the connected load absorbs it. Light bulbs and heaters for example, absorb electrical power and convert it into heat or light. The higher their value or rating in watts the more power they will consume.

1.1 Ohm's law

Ohm's Law deals with the relationship between voltage and current in an ideal conductor. This relationship states that: The potential difference (voltage) across an ideal conductor is proportional to the current through it [11]. The constant of proportionality is called the *resistance*.

$$I = \frac{U}{R}$$

where I is the current expressed in Amperes [A], U is the voltage, bearing the Volt units [V] and R is the electrical resistance in ohms [Ω].

The Ohms's law can be further expanded [3], to get these three quantities in relationship with **power**, such as

$$P = I \cdot U = I^2 \cdot R = \frac{U^2}{R}$$

1.2 Direct current (DC) circuits

Generally, Ohm's law is used on Direct Current (DC) circuits. A DC voltage or current has a fixed magnitude (amplitude) and a definite direction associated with it. Both DC currents and voltages are produced by power supplies, batteries, dynamos and solar cells to name a few.

We also know that DC power supplies do not change their value with regards to time[12], they are a constant value flowing in a continuous steady state direction. In other words, DC maintains the same value for all times and a constant uni-directional DC supply never changes or becomes negative unless its connections are physically reversed.

1.3 Waveforms and alternating current (AC) circuits

An alternating function or Alternating Current (AC) waveform on the other hand is defined as one that varies in both magnitude and direction in more or less an even manner with respect to time making it a "bi-directional" waveform [35]. An AC function can represent either a power source or a signal source with the shape of an AC waveform generally following that of a mathematical sinusoid as defined by

$$A(t) = A_{max} \cdot sin(2\pi ft)$$

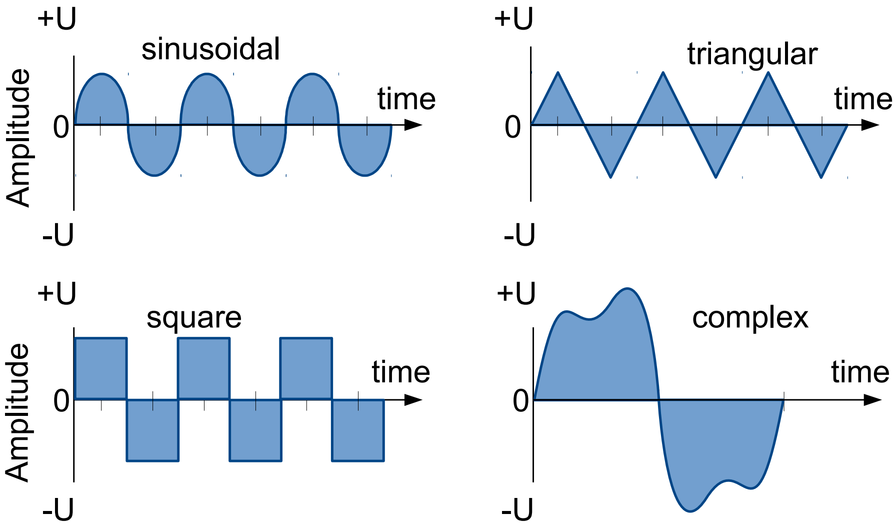

Figure 1 – 1 The common types of waveforms visualised as a function of amplitude

The term AC or to give it its full description of Alternating Current, generally refers to a time-varying waveform with the most common of all being called a **Sinusoid** better known as a **Sinusoidal Waveform**. Sinusoidal waveforms are more generally called by their short description as **Sine Waves**. Sine waves are by far one of the most important types of AC waveform used in electrical engineering.

This means then that the AC waveform is a "time-dependent signal" with the most common type of time-dependant signal being that of the Periodic Waveform. The periodic or AC waveform is the resulting product of a rotating electrical generator. Generally, the shape of any periodic waveform can be generated using a fundamental frequency and superimposing it with harmonic signals of varying frequencies and amplitudes but that is out of the waveform fundamentals theory.

Alternating voltages and currents can not be stored in batteries or cells like direct current (DC) can, it is much easier and cheaper to generate these quantities using alternators or waveform generators when they are needed. The type and shape of an AC waveform depends upon the generator or device producing them, but all AC waveforms consist of a zero voltage line that divides the waveform into two symmetrical halves. The main characteristics of an AC waveform [21] are defined as:

- the **period (T)** is the length of time in seconds that the waveform takes to repeat itself from start to finish. This can also be called the Periodic Time of the waveform for sine waves, or the Pulse Width for square waves
- the **frequency** is the number of times the waveform repeats itself within a one second time period. Frequency is the reciprocal of the time period, defined as $f = \frac{1}{T}$, with the unit of frequency being the Hertz [Hz]
- the **amplitude** is the magnitude or intensity of the signal waveform

1.4 Power in AC circuits

When a reactance (either inductive or capacitive) is present in an AC circuit, the Ohm's law formula does not apply and different approach must be taken to express and calculate power [22].

Real power (or true power) is the power that is used to do the work on the load:

$$P = U_{RMS} \cdot I_{RMS} \cdot cos\,\varphi$$

where P is the real power in watts, U_{RMS} is the Root-mean square (RMS) voltage, defined as $U_{peak}/\sqrt{2}$ in volts, I_{RMS} is the RMS current, defined as $I_{peak}/\sqrt{2}$ in amperes and φ is the impedance phase angle - phase difference between voltage and current.

Reactive power on the other hand, is the power that is wasted and not used to do work on the load. Curiously, it is defined as

$$Q = U_{RMS} \cdot I_{RMS} \cdot sin\,\varphi$$

with Q being the reactive power in volt-ampere-reactive [var].

Apparent power is the power that is supplied to the circuit. Definition:

$$S = U_{RMS} \cdot I_{RMS}$$

where the unit of apparent power S is volt-ampere [VA]. It can be seen that it is not phase-angle dependent.

The relation all these three quantities are in is defined as

$$P^2 + Q^2 = S^2$$

however, again, nothing in the real world is perfect, and this relation only applies for a perfectly **sinusoidal waveforms**!

1.5 Phasor and phase shift

A phasor[26] is a constant complex number representing the complex amplitude (magnitude and phase) of a sinusoidal function of time. It is usually expressed in exponential form. Phasors are used in engineering to simplify computations involving sinusoids, where they can often reduce a differential equation problem to an algebraic one. The origin of the word phasor comes from phase + vector.

Phasor is a vector that represents a sinusoidally varying quantity, as a current or voltage, by means of a line rotating about a point in a plane, the magnitude of the quantity being proportional to the length of the line and the phase of the quantity being equal to the angle between the line and a reference line.

Considering the figure $1-2$, the voltage waveform above starts at zero along the horizontal reference axis, but at that same instant of time the current waveform is still negative in value and does not cross this reference axis until 30°later. Then there exists a Phase difference between the two waveforms as the current cross the horizontal reference axis reaching its maximum peak and zero values after the voltage waveform.

As the two waveforms are no longer *in-phase*, they must therefore be *out-of-phase* by an amount determined by phi, φ. The waveform of the current can also be said to be *lagging* behind the voltage waveform by the phase angle φ. This angle represents the phase shift (also called phase difference) between two sinusoids [19].

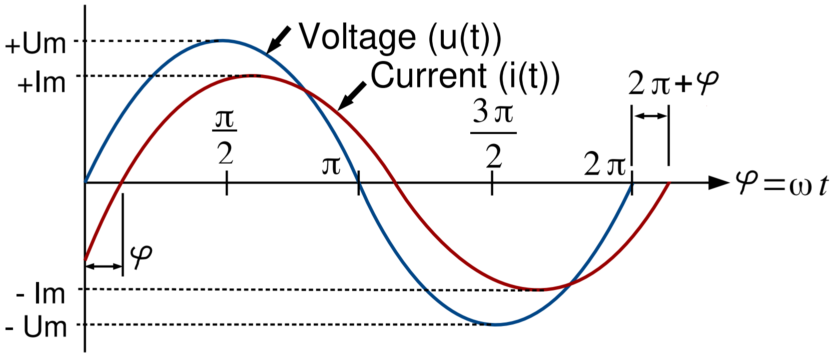

Figure 1 − 2 The phase difference between voltage (blue) and current (red), the origin of phase difference of angle φ

1.6 Power factor and power factor correction

The power factor is just a specific name for a phase shift between the sinusoids of a current and voltage. So the figure 1 − 2 in fact shows the power factor. However, it is not expressed in a plane angle, but rather as a dimensionless number between -1 and 1.

The power factor is defined as $\frac{P}{S}$, as a ratio of the real power over the apparent power[7]. If φ is the phase angle between the current and voltage, then the power factor is equal to the cosine of the angle, $cos\,\varphi$.

If the power factor is 1, it means that all the supplied power is completely consumed by purely resistive load. A positive power factor that is lower than 1 indicates that some power is not consumed by the load and is returned back. The lower the factor, the more power is returned. When power factor is equal to 0, the energy flow is entirely reactive, and stored energy in the load returns to the source on each cycle. A negative power factor means that the device, considered to be power load is in fact a power source (produces more power than consumes).

How can this information be useful? Every load with a power factor other than 1 returns some power back to the transmission line. Since the transmission lines does have some resistance, this returned power translates to some wasted power in a form of heat. Energetic companies want to minimise the power wasted in the transmission lines to increase their profit, so numerous laws are coming into effect to correct [27] (increase) the power factor.

1.7 Electric power measurement

Measuring the electric power makes most sense on the customer appliances. The first reason is, that they generally consume power that is purchased on contract. The energetic company measures all the power used up by the end customer, but customer has no easy way to see how much and how *effectively* is power used by the appliances. The second important reason is that the appliances has a standardised connector (plug) that is guaranteed to fit in all the area using it, which is not a case for example on battery powered devices (batteries has different sizes, connectors and general properties.

When it comes to measuring the electrical power, the first and the most important thing to discuss is safety. Only after all the safety precautions had been made clear, the theory can be clarified and subsequently, the practice can be applied.

If not handled with care, operating or manipulating with voltage can cause permanent damage to appliance or health, or can cause fire or even death. Thus, respect, increased care and knowledge is necessary in all further practical steps involved.

1.8 Power measuring integrated circuits

Although it is possible to construct a circuit out of discrete components that would measure [33] the required physical quantities, and such a solution would probably be the cheapest solution out there, it would be highly impractical due to multiple reasons.

The most importantly, the obtained accuracy of the measurements would be dependent on the implementation and used components. It is safe to assume, that without multiple design iterations, the accuracy may be too low to be used in practice.

Another point is that, there is no definitive guide, ready to follow, about how to design such circuit. The reason of this is the vast amount of components available on the market and a lot of design considerations to take into account, depending on the requirements.

A special purpose integrated circuits (ICs) are being developed for the exact purpose of measuring the real, apparent and reactive power, the power factor, and in most cases, gathering some other relevant information.

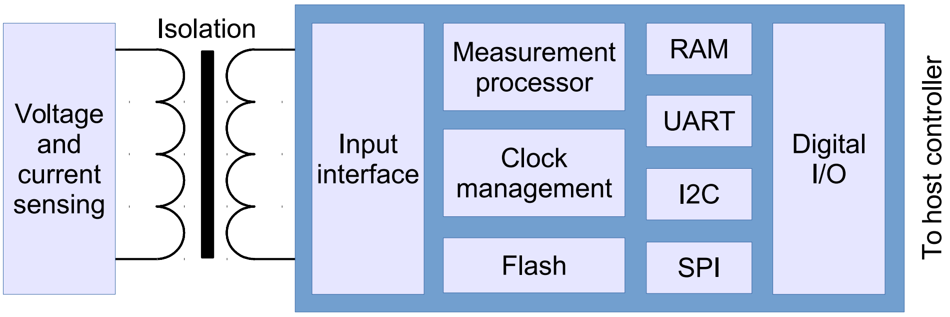

Figure 1 – 3 The simplified block diagram for a power measurement IC

From the block diagram 1 – 3, it can be seen that the power measuring IC is just a specialised microcontroller. It takes the data from the sensing circuitry, which in case of voltage can be measured *directly*, provided that the galvanic isolation is included, for the sake safety. The current however, must be measured *indirectly*. There are three common ways [28] of doing so:

1. **shunt resistor** - a resistor with a very small but precise value, that causes a voltage drop with a current passing through it due to the Ohm's law, regardless of frequency. The actual voltage drop is so small, that it can be assumed insignificant. However, the voltage drop is still present and may cause some issues, if not taken into account. The advantage is really low price. External galvanic isolation must be provided.
2. **current transformer** - a current passing wire inside a current sensing coil. Since it is a magnetic induction based transformer, the galvanic isolation is naturally present. The disadvantage is, that the transformer has a cut-off after which it's effect diminishes rapidly. External magnetic fields can cause problems too. Suitable for measuring current of a fixed (or non-decreasing) frequency.
3. **Hall-effect sensor** - a sensor measuring absolute electromagnetic field in a conductor. In contrast to the current transformer, this sensor is able to measure low frequency currents, down to DC, which is a feat that the shunt resistor possesses too. Can be placed anywhere near the current path and doesn't require physical connection, thus providing galvanic isolation too. The price increases with operating currents range and precision. Prone to external magnetic fields too.

Using dedicated power measuring IC has another advantage apart from being more accurate. In fact, the part datasheet can be consulted and if all application notes and advices are abided, the specified accuracy can be guaranteed.

2 Embedded system

An embedded system is some combination of computer hardware (HW) and software (SW), either fixed in capability or programmable, that is specifically designed for a particular function [9]. Industrial machines, automobiles, medical equipment, cameras, household appliances, airplanes, vending machines and toys (as well as the more obvious cellular phone and Personal digital assistant (PDA)) are among the myriad possible hosts of an embedded system. Embedded systems that are programmable are provided with programming interfaces, and embedded systems programming is a specialized occupation.

2.1 Processing units

The term embedded system is quite broad, so there is no surprise that the spectrum of used processing units is also wide. Since the general purpose microprocessors require external components, namely memories and peripherals, they tend to consume extra power and a board space. Since the design limitations of an embedded systems are most of the time low physical size, low power consumption and/or long uptime and ruggedness (more components mean more parts could fail), microprocessors are seldom used. However, most of the commonly used architectures and word lengths are covered. Due to aforementioned reasons, microcontrollers are favored over microprocessors.

2.2 System-on-chip

Today's state of chip integration allows production costs of a complex system on chip devices to be relatively low, thus making System-on-Chip (SoC) attractive choice for embedded systems. SoCs could be described as an IC that integrates all components of a computer or other electronic system into a single chip. It may contain digital, analog, mixed-signal, and often Radio frequency (RF) functions - all on a single chip substrate [8]. SoCs are very common in the mobile electronics market because of their low power consumption. A typical SoC consists of (specific block diagram can be seen on $3-1$):

- a microcontroller, microprocessor or Digital signal processor (DSP) core(s)
- memory blocks including a selection of Read-Only memory (ROM), Random-access memory (RAM), Electrically erasable programmable ROM (EEPROM) and Flash
- timing sources including oscillators and phase-locked loops

- peripherals including counter-timers, real-time timers and power-on reset generators
- external interfaces, including industry standards such as Universal serial bus (USB), FireWire, Ethernet, Universal asynchronous receiver/transmitter (UART), Serial peripheral interface (SPI)
- analog interfaces including Analog-to-digital converters (ADCs) and Digital-to-analog converters (DACs)
- voltage regulators and power management circuits
- a bus connecting these blocks

SoCs can be implemented as an Application-specific integrated circuit (ASIC) or using a Field-programmable gate array (FPGA).

2.3 Operating system

An Operating system (OS) is a computer program that supports a computer's basic functions, and provides services to other programs (or applications) that run on the computer. The applications provide the functionality that the user of the computer wants or needs. The services provided by the operating system make writing the applications faster, simpler, and more maintainable.

Over time, a lot of embedded OSes suited for embedded systems were developed. An embedded OS is a type of OS that is embedded and specifically configured for a certain HW configuration. HW that uses embedded OS is designed to be lightweight and compact, forsaking many other functions found in non-embedded (i.e. desktop) computer systems in exchange for efficiency at resource usage [13]. This means that they are made to do specific tasks and do them efficiently. Notable embedded OSes currently in use by consumers include:

- **Embedded Linux** - used in many other devices like printers, routers or smart TVs; Android is a derivative of embedded Linux
- **iOS** - subset of Mac OS X, used in Apple's mobile devices Palm OS
- **Windows Mobile** - Microsoft's OS for mobile devices

2.4 Real-time operating system

A Real-time operating system (RTOS) is just a special purpose OS. The real time part of the name does not mean that the system responds quickly, it just means that there are rigid time requirements that must be met. If these time requirements are not met, the results can become inaccurate or unreliable[17]. Embedded systems frequently posses the real time requirement. There are two kinds of RTOSes:

Hard Real Time - system delays are known or at least bounded. Said to be operating correctly if the system can return results within any time constraints.

Soft Real Time - critical tasks get priority over other tasks and will retain priority until the task is completed. This is another way of saying that real time tasks cannot be kept waiting indefinitely. Soft real time makes it easier to mix the system with other systems.

2.5 Embedded Linux

Linux itself is a kernel, but Linux in day to day terms rarely means so. Embedded Linux generally refers to a complete Linux distribution targeted at embedded devices. There is no Linux kernel specifically targeted at embedded devices, the same Linux kernel source code can be built for a wide range of devices, workstations, embedded systems, and desktops though it allows the configuration of a variety of optional features in the kernel itself. In the embedded development context, there can be an embedded Linux system which uses the Linux kernel and other software or an embedded Linux distribution which is a prepackaged set of applications meant for embedded systems and is accompanied by development tools to build the system[10].

With the availability of consumer embedded devices, communities of users and developers were formed around theses devices: Replacement or enhancements of the Linux distribution shipped on the device has often been made possible thanks to availability of the source code and to the communities surrounding the devices. Due to the high number of devices, standardized build systems have appeared, namely OpenWRT.

2.6 Kernel

The kernel is the essential center of a computer OS, the core that provides basic services for all other parts of the OS [5]. It has complete control over what happens in the system. A kernel can be contrasted with a shell, the outermost part of an OS that interacts with user commands. Kernel and shell are terms used more frequently in Unix or Unix-like OSes than in IBM mainframe or Microsoft Windows systems.

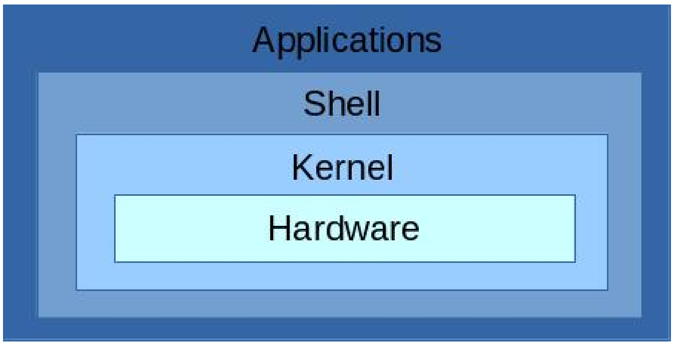

Figure 2 − 1 The simplified view on the Linux system structure

The simplified view on the Linux system structure can be seen on 2 − 1. It does not include device driver, compilers, daemons, utilities, commands, library files and such, but should be enough for a demonstration.

2.7 OpenWRT

OpenWrt is an OS (in particular, an embedded OS) based on the Linux kernel, primarily used on embedded devices to route network traffic. It has been optimized for size, to be small enough for fitting into the limited storage and memory available in home routers.

OpenWrt is configured using a command-line interface (ash shell), or a web interface (LuCI). There are about 3500 optional SW packages available for installation via the `opkg` package management system.

2.8 Components of the OpenWRT

The main components are the Linux kernel, `util-linux-ng`, `uClibc` and `BusyBox`. The Linux kernel was already mentioned. `util-linux-ng` is self explanatory - it is a set of linux utilities.

`BusyBox` is a software that provides several stripped-down Unix tools in a single executable file. It runs in a variety of Portable operating system interface (POSIX) environments such as Linux, Android, Berkeley Software Distribution (BSD) family and others, such as proprietary kernels, although many of the tools it provides are designed to work with interfaces provided by the Linux kernel.

`uClibc` is a small C standard library intended for Linux kernel-based operating systems for embedded system and mobile devices.

3 GL.inet board

GL.inet Smart Router is a remake of a common TP-Link router TL-WR703N. The board changes include, but are not limited to, increased RAM and Flash memory, custom firmware and what is the most important - 5 usable General-purpose I/O (GPIO) pins exposed to the 2cm pin header for utility. Whole thesis is revolving around taking advantage of this fact. The frequency of Central processing unit (CPU) is 400 Mega-Hertz (MHz) and it is suited for running Linux distributions for embedded devices, preferably OpenWrt or DD-Wrt. The board provides Local area network (LAN) and Wide area network (WAN) connection, as well as other interfaces defined in Institute of Electrical and Electronics Engineers (IEEE). The information about the board are summed up in the table 3 – 1.

Table 3 – 1 The basic characteristics of the GL.inet board

Model	GL-iNet 6408A / 6416A
CPU	Atheros 9331, 400 MHz
RAM	DDR 64 Mega-Byte (MB)
ROM	Flash 8MB (6408A) / 16MB (6416A)
Interface	1 WAN, 1 LAN, 1 USB2.0, 1 Micro USB(Power), 5 GPIO
Wireless	IEEE802.11n/g/b, IEEE 802.3, IEEE 802.3u

3.1 Atheros AR9331 Wi-Fi System-on-Chip

The Atheros AR9331 is a highly integrated and cost effective IEEE 802.11n 1x1 2.4 GHertz, the SI unit of frequency (Hz) SoC for Wireless local area network (WLAN) Access Point (AP) and router platforms. The block diagram of the chip can be seen in figure 3 – 1. Features of this SoC are following:

- Complete IEEE 802.11n 1x1 AP or router in a single chip
- Microprocessor without Interlocked Pipeline Stages (MIPS) 24K processor operating at up to 400 MHz
- External 16-bit Double data rate synchronous DRAM (DDR) or Synchronous dynamic random access memory (SDR) memory interface
- SPI NOR Flash memory support

- No external EEPROM needed
- 4 LAN ports and 1 WAN port IEEE 802.3 Fast Ethernet switch with auto-crossover, auto polarity
- Fully integrated RF front-end including Power amplifier (PA) and Low-noise amplifier (LNA)
- Optional external LNA/PA
- Switched antenna diversity
- High-speed UART for console support
- Integrated Interchip Sound (I^{2}S)/Sony-Philips Digital Interface Format (S/PDIF) audio interfaces
- Subscriber line interface circuit (SLIC) for Voice over IP (VOIP)/Pulse code modulation (PCM)
- USB 2.0 host/device mode support
- GPIO/Light emitting diode (LED) support
- Joint test action group (JTAG)-based processor debugging supported
- 25 MHz or 40 MHz reference clock input
- 148-pin, 12 mm x 12 mm dual-row Quad Flat No-leads (LPCC) package

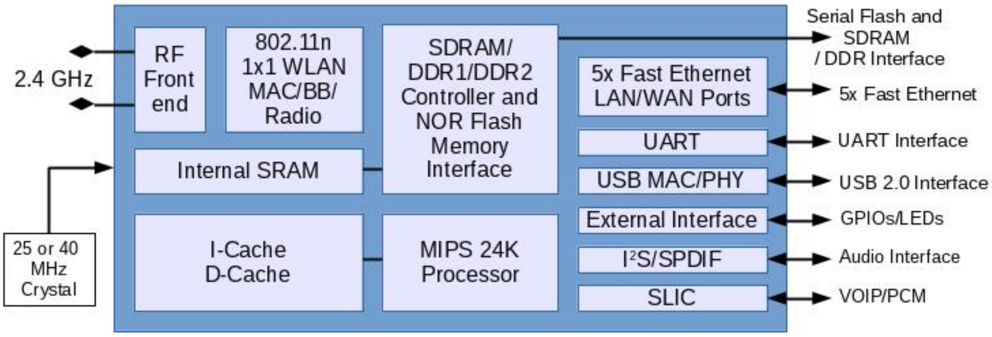

Figure 3 – 1 The block diagram of the Atheros AR9331 SoC used as a main processing unit on GL.inet board

3.2 From TL-WR703N to GL.inet

TP-Link TL-WR703N router is a popular choice among HW customisation community because of it's cheap price tag compared to processing power and usage of a full-grown Linux distribution. People have figured out how to upgrade RAM / Flash memories or to make use of not used GPIO / UART ports for their own needs. These solutions however were mostly crude and expensive to replicate. The

GL.inet team saw an opportunity to grasp this public knowledge and rolled out their own improved board clone to the marked.

Whole printed circuit board of TL-WR703N was remade by the GL.inet team to expose the unused GPIO pins on the SoC, utilize two Ethernet port instead of one and utilize the USB 2.0 port. Memory chips were replaced by their higher capacity alternatives.

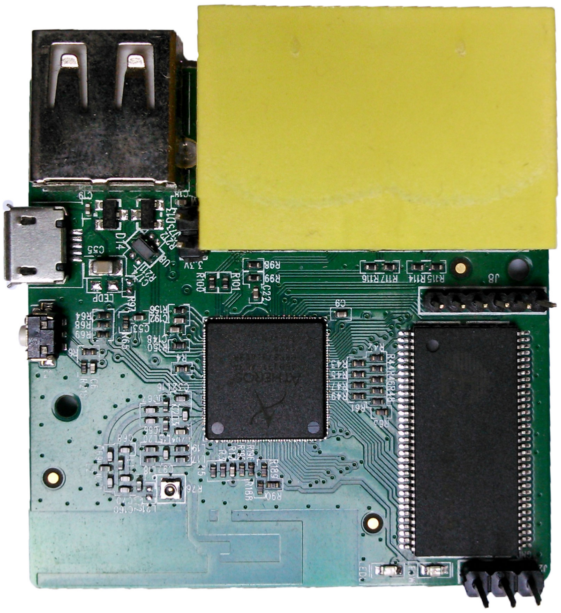

Figure 3 – 2 The front side of the GL.inet board exposing the main Atheros SoC, RAM and interfaces

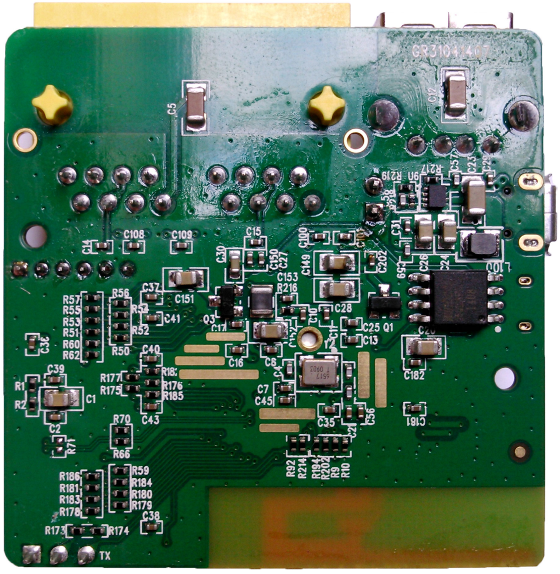

Figure 3 – 3 The back side of the GL.inet board exposing the Flash memory and a main voltage regulator

4 ESP8266 Wi-Fi module

The ESP8266 Wi-Fi module is a self contained SoC with integrated Transmission Control Protocol/Internet Protocol (TCP/IP) protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. The ESP8266 module is an extremely cost effective solution, with a huge code-base and community, making it a preferable option for many modern projects, mainly the ones that follow the Internet of Things (IoT) trend.

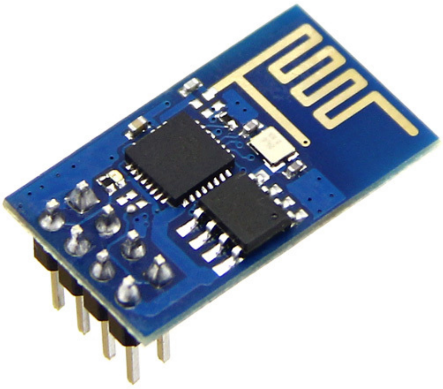

Figure 4 – 1 The first commercial iteration of the ESP8266 module, the ESP-1, exposing two GPIOs

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal printed circuit board (PCB) area.

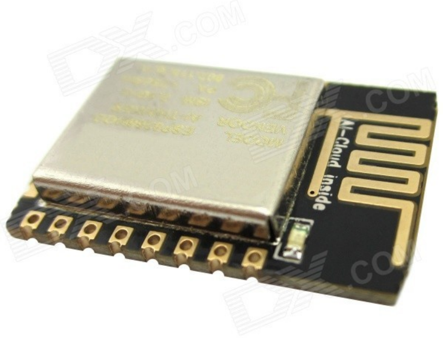

Figure 4 – 2 The certified ESP-12E module exposing all GPIOs

4.1 Features of a ESP8266 chip

The list of features contained in a 24 pin plastic Quad Flat No-leads (QFN) package are listed (but not limited to) in the following list:

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated LNA, power amplifier, matching network and power management
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 512kB - 4MB Flash Memory (depending on manufacturer)
- Integrated low power 32-bit CPU could be used as application processor
- Secure Digital Input Output (SDIO) 1.1 / 2.0, SPI, UART
- Standby power consumption of < 1.0mW

5 Requirements

The device under test will be referred to as **appliance**. The requirements for the final **measuring device** are grouped to the three categories. Mandatory requirements are bound to be met at any cost. Some of the high importance requirements can be skipped or slightly modified, if unpredictable obstacles are found. However, they are all assumed to be completed for well being of the project. Optional requirements will be completed only if the resources allow it.

They are also divided to a hardware part and software part. Software is easier to change than hardware and requires hardware to be run on. Software is also limited by the resources provided by the hardware. Therefore, hardware needs to be logically completed first and are also highlighted in figures $5-1$ and $5-2$.

5.1 Hardware requirements

Mandatory:

A. Measure current, voltage and phase angle simultaneously to calculate the real power and the power factor

B. More measuring devices can be added to the system by user without HW or SW modifications

C. Devices under test run at nominal 230 V, 50 Hz that use two-way or three-way EU plug

D. Protection against the electrical shock, fire hazard and damage caused by power surges

High importance:

E. Store the measured data in server's node available non-volatile local memory

F. Completely shut the appliance off or back on

G. Indicate that the measuring device is active (configured and working) with a LED

H. Handle maximum of 8 A currents drawn by the appliance

Optional:

I. Store the measured data on an USB flash disk

J. Provide HW support for some crossed support signalisation (i.e. by sound)

K. Internet and Ethernet connection on server node

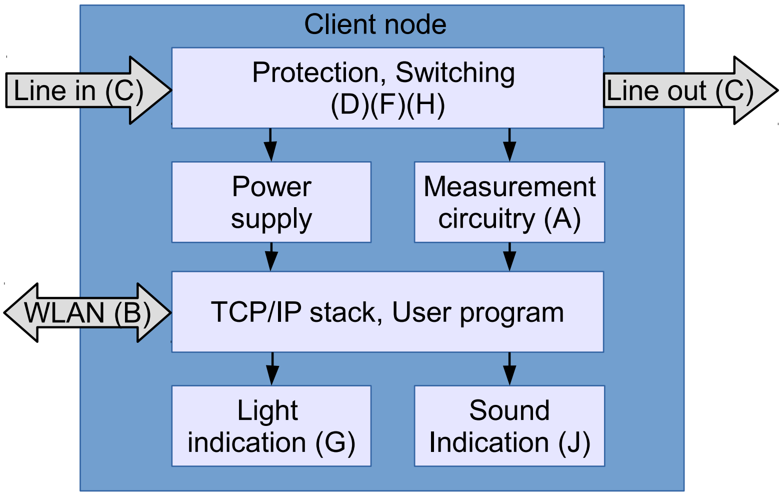

Figure 5 – 1 The proposed block diagram of a *client node*, including HW requirements

5.2 Software requirements

Mandatory:

- The Graphical user interface (GUI) running on the web-server
- Add, edit (configure) and remove measuring devices to/from the system
- Present the instantaneous (real) power consumed for each appliance

High importance:

- Graphs of all measured quantities over time
- Authentication mechanism
- Automatic configuration of the new connected measuring devices

Optional:

- Access to GUI outside of local network
- Control Wi-Fi repeater mode to strengthen the signal for client nodes
- Send measured data to the cloud storage
- Separate administrator (view and change) and user (view only) privileges
- Ability to set thresholds for measured data and notify user about crossing them via text based message

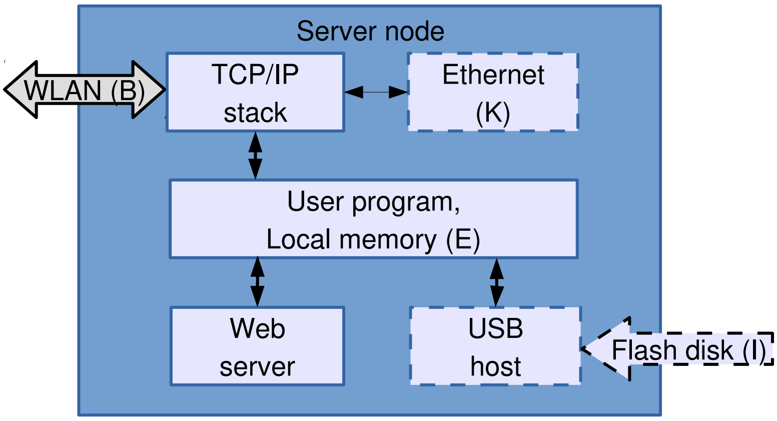

Figure 5 – 2 The proposed block diagram of a *server node*, including HW requirements

6 System design

The first mandatory software requirement asks for a web server. It is entirely possible for every measuring device to contain its own web server. However, multiple points are requiring separate parts to work as a **system**. Two common system structures are *centralised* and *decentralised*. Decentralised (peer-to-peer) systems[32] are harder to build but are more fail-proof. Since fail-proofness is not mentioned in the requirements, centralised system might suffice.

Using centralised system means, that the measuring devices will use one separate accessory, from now called the **server node**, to do most of the work on the software side. The work includes, but is not limited to, receiving the measured data, storing them, hosting the web server with the GUI containing all necessary options and information, handling the USB or communication with a cloud and so on. The block diagram for a server node, depicting required blocks can be seen in the figure 5 – 2)

Where there are at least two nodes in a system, they have to communicate together in a particular way, known to both of them. The web server naturally operates over TCP/IP. Therefore, same networking stack, that is used for communication between the server node and user can be used to communicate to client nodes as well. TCP/IP hardware is ready to be used and is supporting a full-blown networking stack, powering communication over today's networks.

The measuring devices, from now on called **client nodes**, will consist of blocks of the remaining hardware requirements. The resulting block diagram can be seen in the figure 5 – 1)

6.1 Hardware components breakdown

For the **server node**, a complete working solution already exists, ready to be employed. The **GL.inet board**, described in more detail in the chapter 3, is greatly sufficient in all required aspects, and thus is used for this purpose.

Luckily, a particular part of the required functionality for the client is already integrated as a **ESP-8266 module**, described in more detail in the chapter 4. The module contains the TCP/IP stack, micro-controller (application processor) running the user program, WLAN and light indication, all in one piece, so this greatly simplifies the design process and allows for more focus on the actual measurement circuitry. The actual ESP-12E module should be used, because of the available certification[25], which allows it to be introduced on the market later. It was shown in the figure 4 − 2. The Pulse-width modulation (PWM) is present there too, so sound indication requires just an additional sound emitting device.

Talking about the measurement circuitry, the viable candidate is MAX78615 [14] with the companion IC MAX78700 [15]. The couple 6 − 1 should be used, because it provides multiple ways of same voltage level communication with the processor, galvanic isolation via the pulse transformer for improved circuitry protection, great precision, accuracy and utility. The shunt resistor is utilised as a way of obtaining measurements, described in the sub-chapter 1.8.

Luckily, a particular part of the required functionality for the client node is already integrated as a **ESP-8266 module**, described in more detail in the chapter 4. The module contains the TCP/IP stack, micro-controller (application processor) running the user program, WLAN and light indication, all in one piece, so this greatly simplifies the design process and allows for more focus on the actual measurement circuitry. The actual ESP-12E module has been chosen, because of the available certification[25], which allows it to be introduced on the market later. It was already shown in the figure 4 − 2. The PWM is present there too, so sound indication requires just a sound emitting device.

Luckily, a particular part of the required functionality for the client node (displayed as a simplified schematic in 6 − 1) is already integrated as a **ESP-8266 module**, described in more detail in the chapter 4. The module contains the TCP/IP stack, micro-controller (application processor) running the user program, WLAN and light indication, all in one piece, so this greatly simplifies the design process and allows for more focus on the actual measurement circuitry. The actual ESP-12E module has been chosen, because of the available certification[25], which allows it to be introduced on the market later. It was already shown in the figure 4 − 2. The PWM is present there too, so sound indication requires just a sound emitting device.

For the protection against fire a standard glass fuse or a resettable Positive thermal coefficient (PTC) fuse[36] should be used. Because of the variable nature of most devices, it is hard to calculate the current consumption of the circuit. It can be measured after the first iteration is manufactured. Thus, the easily replaceable standard glass fuse has been chosen because of its versatility. The circuit protection against high voltage should be solved with an isolated DC-to-DC converter[6] or with the linear transformer coupled with the linear voltage regulator[18]. Since

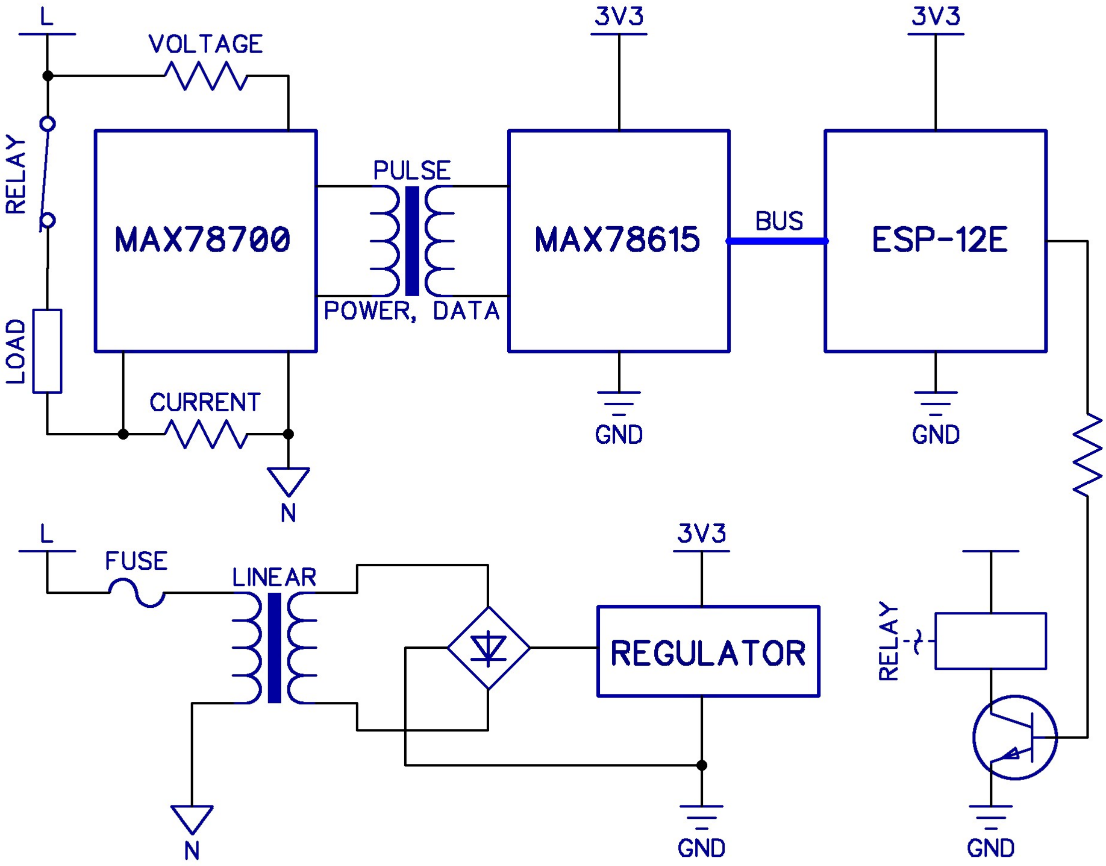

Figure 6 – 1 Greatly simplified schematic of a *client node* sketching the inner working

the former one is either expensive or hard to design, and this work does not want to focus on more complexities, the latter option has been chosen.

Choosing the voltage level for the digital electronics (the output voltage of the linear regulator) is straightforward. Since the ESP-12E works on nominal 3.3V, this is the level that has been chosen. Having ICs using the same voltage level removes the need to level-shift the communication between them, thus increasing the simplicity of the design.

Talking about the measurement circuitry, the candidate is MAX78615 [14], working on nominal 3.3V level, along with the companion IC MAX78700 [15]. The couple has been chosen, because it provides multiple ways of communication with the processor (buses), galvanic isolation via the pulse transformer for improved circuitry protection, great precision, accuracy and utility. The resistor network, including the shunt resistor is utilised as a way of obtaining measurements. The shunt resistor is also briefly described in the sub-chapter 1.8.

The remaining part of the client node block diagram 5 – 1 not yet mentioned is switching. Either a mechanical relay or a semiconductor device, such as a thyristor or a Solid-state relay (SSR) isolated by an opto-coupler[30] will do. Mechanical relays tend to be larger and produce sound noise, have slow response time, but have inbuilt separate isolation and are capable of switching higher currents without additional thermal issues than their semiconductor counterparts[4]. The disadvantages of the mechanical relay are not relevant here, thus it has been chosen.

6.2 Schematic and PCB design

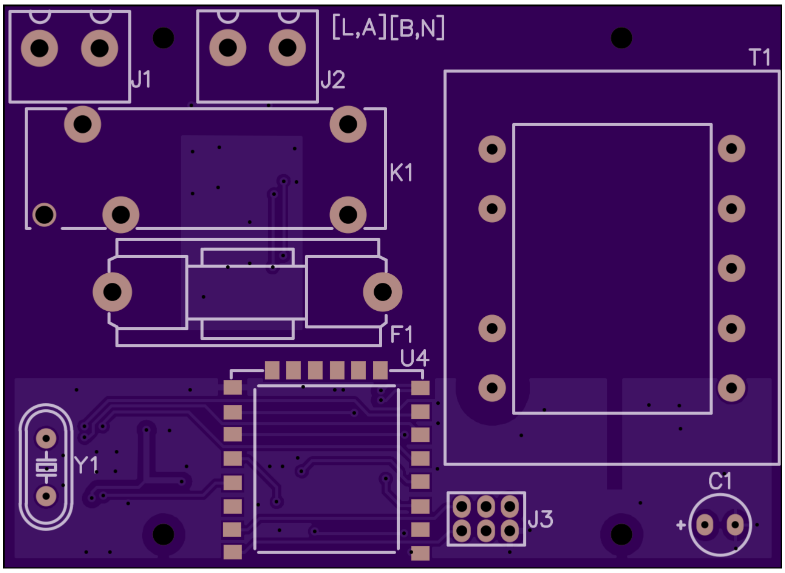

Figure 6 – 2 The top layer of the designed PCB (client node), exposing maintly THT components

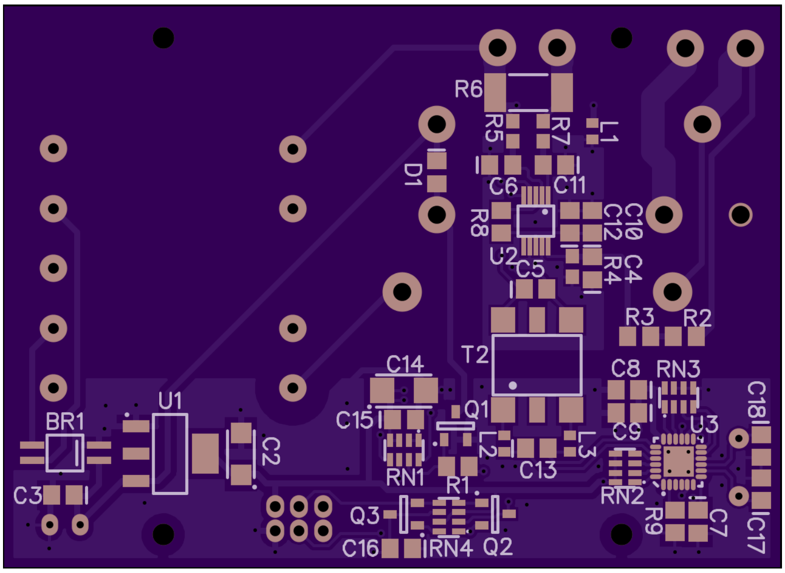

Figure 6 – 3 The bottom layer of the designed PCB (client node), exposing mainly SMT components

6.3 Software components breakdown

As with the hardware components, again, starting with the server node, the board should work alongside the **OpenWRT** OS, again, described in a deeper detail in the sub-chapter 2.7. It allows for relatively easily configurable network connections[23] and the web-server.

The web-server should be handled by some package already available for installation into the OpenWRT distribution. There are various choices. Ordered in an ascending order by their complexity, the most common ones are: `uhttpd`, `lighttpd`, `Apache` and `nginx`. The resource requirements are proportional to the complexity, and since GL.inet is not an extremely powerful machine, the choice should fall on the more lightweight one from the beginning of the list, with just as many features as absolutely necessary[1]. The `lighttpd` was chosen for this task after some research, as it is the best fit for the given criteria.

As far as a choice for the web interface programming/scripting language goes, there are three rather good choices: `Python`, `PHP` and `Lua`. There is no exact way to choose - it all depends on the confidence and the efficiency of the implementer, which one fits the best. Any available language will do. However, it should be noted, that `Lua`, is somewhat superior choice, since two influential software solutions for the proposed system are written in it. On the side of the OpenWRT, we are speaking about the web configuration interface called `LuCi`[34], while `NodeMCU`[16] is a `Lua` eco-system based on ESP-8266. Both solution are relatively mature and open-source and can be referenced easily. However, the `PHP` language was chosen, because it is the easiest way to develop the required functionality in it.

Another choice lies in the way of storing the measured data on the server. For a local storage, one can either use plain text format for very simple data, or a data-base of some sort, preferably a Relational Data-base management system (RDBMS)[29]. If a data-base is to be chosen, the `SQLite`[2] is a good choice - it was already noted, that the computational resources are at premium. Everything marketed as lightweight has an edge here. For a remote storage, there is a plethora of cloud-based solutions, ready to employ and use, so this is a very good choice, if an internet connection is available. Because there is nothing preventing us from using the Internet connection in the design, the cloud storage is chosen as a primary one and `SQLite` in stored in a local memory is used only as a backup, in case the connection cannot be established.

On a client side of a proposed system, there are again three main language options. The first one is the already mentioned `Lua`. This language is simple, yet not terribly popular. The availability of the design resources might be limited. Second one is the ruler of the micro-controller world[31], plain `C`, suited for more experienced programmer, but providing the highest flexibility. The third choice is to use `C++` based eco-system, derived from Arduino. This choice is far more superior, because it provides access to enormous resources and libraries[24] readily available on the Internet with and addition to (mostly) cross-compatible code, which is a nice bonus. This option has been chosen, because it is, again, the easiest way to accomplish the task.

6.4 External components

One entity, that was already mentioned, but still not depicted in the design considerations is the USB storage. It requires both HW and SW to work, and is externally connected to the server node.

Should the USB storage be applied for a local data storage capability, in case when the server node cannot connect to the internet for a cloud upload and available non-volatile local memory is depleted, either a Flash stick or an external Hard-disk drive (HDD) should be used. Both require additional kernel modules to work, in addition to the daemon, that will handle the mounting process. In case of the HDD, the capable power supply is needed to power both GL.inet board and the HDD at the same time.

7 Realisation

8 Conclusion

Táto časť záverečnej práce je povinná. Autor uvedie zhodnotenie riešenia. Uvedie výhody, nevýhody riešenia, použitie výsledkov, ďalšie možnosti a pod., prípadne načrtne iný spôsob riešenia úloh, resp. uvedie, prečo postupoval uvedeným spôsobom.

References

[1] T. Adelstein and B. Lubanovic. *Linux System Administration.* O'Reilly Series. O'Reilly Media, 2007, p. 162. ISBN: 9780596009526.

[2] G. Allen and M. Owens. *The Definitive Guide to SQLite.* Books for professionals by professionals. Apress, 2011. ISBN: 9781430232261.

[3] H.W. Beaty. *Electric Power Distribution Systems: A Nontechnical Guide.* PennWell nontechnical series. PennWell, 1998, p. 12. ISBN: 9780878147311.

[4] S.W. Blume. *Electric Power System Basics for the Nonelectrical Professional.* IEEE Press Series on Power Engineering. Wiley, 2008, p. 163. ISBN: 9780470185803.

[5] D.P. Bovet and M. Cesati. *Understanding the Linux Kernel.* O'Reilly Media, 2005. ISBN: 9780596554910.

[6] J. Carr and J. Carr. *Linear Integrated Circuits.* Elsevier Science, 1996, p. 182. ISBN: 9780080938455.

[7] J.B. Dixit and A. Yadav. *Electrical Power Quality.* Laxmi Publications Pvt Limited, 2010, pp. 80–81. ISBN: 9789380386744.

[8] M.J. Flynn and W. Luk. *Computer System Design: System-on-Chip.* Wiley, 2011. ISBN: 9781118009918.

[9] J.G. Ganssle and S.R. Ball. *Embedded Systems.* Computer languages, systems & structures. Elsevier/Newnes, 2008. ISBN: 9780750686259.

[10] C. Hallinan. *Embedded Linux Primer: A Practical Real-World Approach.* Prentice Hall Open Source Software Development Series. Pearson Education, 2010. ISBN: 9780137061105.

[11] T. Henry. *Ohm's Law, Electrical Math and Voltage Drop Calculations.* Henry Publications, 2008.

[12] S.L. Herman. *Direct Current Fundamentals.* Cengage Learning, 2012. ISBN: 9781111127466.

[13] A. Holt and C.Y. Huang. *Embedded Operating Systems: A Practical Approach.* Undergraduate Topics in Computer Science. Springer London, 2014. ISBN: 9781447166030.

[14] Maxim Integrated. *MAX78615+LMU Data Sheet.* (Accessed on 13/04/2016). URL: `https://datasheets.maximintegrated.com/en/ds/MAX78615+LMU.pdf`.

[15] Maxim Integrated. *MAX78700 Data Sheet.* (Accessed on 13/04/2016). URL: `https://datasheets.maximintegrated.com/en/ds/MAX78700.pdf`.

[16] A. Kurniawan. *NodeMCU Development Workshop:* PE Press.

[17] Jean J. Labrosse. *MicroC/OS-II: The Real Time Kernel.* Meets requirements for safety-critical systems. Taylor & Francis, 2002. ISBN: 9781578201037.

[18] *Linear Integrated Circuits.* McGraw-Hill Education (India) Pvt Limited, 2008, p. 561. ISBN: 9780070648180.

[19] C. Maxfield et al. *Electrical Engineering: Know It All: Know It All.* Newnes Know It All. Elsevier Science, 2011, pp. 230–233. ISBN: 9780080949666.

[20] R.L. Meade. *Foundations of Electronics.* Foundations of Electronics, Circuits and Devices. Thomson/Delmar Learning, 2002, p. 85. ISBN: 9780766840270.

[21] A. Nicolaides. *Electrical and Electronic Principles II.* P.A.S.S, 1996, p. 104. ISBN: 9781872684345.

[22] C. Rawlins. *Basic AC Circuits.* Elsevier Science, 2000. ISBN: 9780080493985.

[23] R. Rosen. *Linux Kernel Networking: Implementation and Theory.* Books for professionals by professionals. Apress, 2013. ISBN: 9781430261964.

[24] P. Seneviratne. *Internet of Things with Arduino Blueprints.* Packt Publishing, 2015, p. 42. ISBN: 9781785281914.

[25] LTD - 2ADUI Shenzhen Anxinke technology co. *FCC ID 2ADUIESP-12.* (Accessed on 14/04/2016). URL: `https://fccid.io/2ADUIESP-12`.

[26] R.R. Singh. *Electrical Networks.* McGraw-Hill Education (India) Pvt Limited, 2009, pp. 4–13. ISBN: 9780070260962.

[27] S.N. Singh. *Electric Power Generation, Transmission and Distribution.* PHI Learning, 2008, p. 53. ISBN: 9788120335608.

[28] G. Srinivasan, S. Priya, and N. Sun. *Composite Magnetoelectrics: Materials, Structures, and Applications.* Woodhead Publishing Series in Electronic and Optical Materials. Elsevier Science, 2015, p. 209. ISBN: 9781782422648.

[29] S. Sumathi and S. Esakkirajan. *Fundamentals of Relational Database Management Systems.* Studies in Computational Intelligence. Springer Berlin Heidelberg, 2007. ISBN: 9783540483977.

[30] A.M. Trzynadlowski. *Introduction to Modern Power Electronics.* Wiley, 2015, p. 85. ISBN: 9781119003229.

[31] T. Van Sickle. *Programming Microcontrollers in C.* Electronics & Electrical. LLH Technology Pub., 2001. ISBN: 9781878707574.

[32] Q.H. Vu, M. Lupu, and B.C. Ooi. *Peer-to-Peer Computing: Principles and Applications.* Springer Berlin Heidelberg, 2009. ISBN: 9783642035142.

[33] J.G. Webster. *Electrical Measurement, Signal Processing, and Displays.* Principles and Applications in Engineering. CRC Press, 2003. ISBN: 9780203009406.

[34] M. Weidner. *Linux headless with PC Engines ALIX:* LULU Press, 2013, p. 86. ISBN: 9781291599619.

[35] J.C. Whitaker. *AC Power Systems Handbook, Third Edition*. Electronics Handbook Series. CRC Press, 2006. ISBN: 9781420005813.

[36] A. Wright, P.G. Newbery, and Institution of Electrical Engineers. *Electric Fuses, 3rd Edition*. Energy Engineering Series. Institution of Engineering and Technology, 2004, p. 15. ISBN: 9780863413995.

Appendices

Appendix A Prílohy

Appendix B Bibliografické odkazy

Appendix C Vytvorenie zoznamu skratiek a symbolov

Appendix D

Appendix A

Prílohy (Appendices)

Táto časť záverečnej práce je povinná a obsahuje zoznam všetkých príloh vrátane elektronických nosičov. Názvy príloh v zozname musia byť zhodné s názvami uvedenými na príslušných prílohách. Tlačené prílohy majú na prvej strane identifikačné údaje – informácie zhodné s titulnou stranou záverečnej práce doplnené o názov príslušnej prílohy. Identifikačné údaje sú aj na priložených diskoch alebo disketách. Ak je médií viac, sú označené aj číselne v tvare I/N, kde I je poradové číslo a N je celkový počet daných médií. Zoznam príloh má nasledujúci tvar:

Appendix A CD médium – záverečná práca v elektronickej podobe, prílohy v elektronickej podobe.

Appendix B Používateľská príručka

Appendix C Systémová príručka

Prílohová časť je samostatnou časťou kvalifikačnej práce. Každá príloha začína na novej strane a je označená samostatným písmenom (Appendix A, Appendix B, . . .). Číslovanie strán príloh nadväzuje na číslovanie strán v hlavnom texte. Pri každej prílohe sa má uviesť prameň, z ktorého sme príslušný materiál získali.

Appendix B

Bibliografické odkazy

Táto časť záverečnej práce je povinná. V zozname použitej literatúry sa uvádzajú odkazy podľa normy STN ISO 690–2 (01 0197) (Informácie a dokumentácia. Bibliografické citácie. Časť 2: Elektronické dokumenty alebo ich časti, dátum vydania 1. 12. 2001, ICS: 01.140.20). Odkazy sa môžu týkať knižných, časopiseckých a iných zdrojov informácií (zborníky z konferencií, patentové dokumenty, normy, odporúčania, kvalifikačné práce, osobná korešpondencia a rukopisy, odkazy cez sprostredkujúci zdroj, elektronické publikácie), ktoré boli v záverečnej práci použité.

Existujú dva hlavné spôsoby citovania v texte.

- Citovanie podľa mena a dátumu.
- Citovanie podľa odkazového čísla.

Preferovanou metódou citovania v texte vysokoškolskej a kvalifikačnej práce je podľa normy ISO 7144 citovanie podľa mena a dátumu . V tomto prípade sa zoznam použitej literatúry upraví tak, že za meno sa pridá rok vydania. Na uľahčenie vyhľadávania citácií sa zoznam vytvára v abecednom poradí autorov.

Príklad: . . . podľa je táto metóda dostatočne rozpracovaná na to, aby mohla byť všeobecne používaná v . . .

Druhý spôsob uvedenia odkazu na použitú literatúru je uvedenie len čísla tohto zdroja v hranatých zátvorkách bez mena autora (autorov) najčastejšie na konci príslušnej vety alebo odstavca.

Príklad: . . . podľa [13] je táto metóda dostatočne rozpracovaná na to, aby mohla byť všeobecne používaná v . . . ako je uvedené v [14].

Citácie sú spojené s bibliografickým odkazom poradovým číslom v tvare indexu alebo čísla v hranatých zátvorkách. Odkazy v zozname na konci práce budú usporiadané podľa týchto poradových čísel. Viacero citácií toho istého diela bude mať rovnaké číslo. Odporúča sa usporiadať jednotlivé položky v poradí citovania alebo podľa abecedy.

Rôzne spôsoby odkazov je možné dosiahnuť zmenou voľby v balíku `natbib`:

```
% Citovanie podla mena autora a roku
\usepackage[]{natbib}\citestyle{chicago}
% Možnosť rôznych štýlov citácií. Príklady sú uvedené
% v preambule súboru natbib.sty.
% Napr. štýly chicago, egs, pass, anngeo, nlinproc produkujú
% odkaz v tvare (Jones, 1961; Baker, 1952). V prípade, keď
% neuvedieme štýl citácie (vynecháme \citestyle{}) v "options"
% balíka natbib zapíšeme voľbu "colon".
```

Keď zapneme voľbu `numbers`, prepneme sa do režimu citovania podľa odkazového čísla.

```
% Metoda ciselnych citacii
\usepackage[numbers]{natbib}
```

Pri zápise odkazov sa používajú nasledujúce pravidlá:

Pri využívaní informácií z elektronických dokumentov treba dodržiavať tieto zásady:

- upredností ujeme autorizované súbory solídnych služieb a systémov,
- zaznamenáme dostatok informácií o súbore tak, aby ho bolo opäť možné vyhľadať,
- urobíme si kópiu použitého prameňa v elektronickej alebo papierovej forme,
- za verifikovateľnosť informácií zodpovedá autor, ktorý sa na ne odvoláva.

Pre zápis elektronických dokumentov platia tie isté pravidlá, ako pre zápis „klasických“. Navyše treba uviesť tieto údaje:

- druh nosiča [online], [CD-ROM], [disketa], [magnetická páska]
- dátum citovania (len pre online dokumenty)
- dostupnosť (len pre online dokumenty)

Poradie prvkov odkazu je nasledovné: Autor. Názov. In Názov primárneho zdroja: Podnázov. [Druh nosiča]. Editor. Vydanie alebo verzia. Miesto vydania : Vydavateľ, dátum vydania. [Dátum citovania]. Poznámky. Dostupnosť. ISBN alebo ISSN.

Appendix C

Vytvorenie zoznamu skratiek a symbolov

Ak sú v práci skratky a symboly, vytvára sa *Zoznam skratiek a symbolov* (a ich dešifrovanie). V prostredí LaTeXu sa takýto zoznam ľahko vytvorí pomocou balíka `nomencl`. Postup je nasledovný:

Curriculum vitae

Táto časť je nepovinná. Autor tu môže uviesť svoje biografické údaje, údaje o záujmoch, účasti na projektoch, účasti na súťažiach, získané ocenenia, zahraničné pobyty na praxi, domácu prax, publikácie a pod.