Technical University of Košice
Faculty of Electrical Engineering and Informatics

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

Master's Thesis

2016 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:	Computer Modeling
Field of study:	9.2.9 Applied Informatics
Department:	Department of Electronics and Multimedia Communications (DTIEE)
Supervisor:	Ing. Tibor Vince, PhD.
Consultant(s):	

Košice 2016 **Peter Babič**

Errata

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

Peter Babič

Košice 2016

Page	Line	Wrong	Correct

Abstract

This thesis shows the process of designing, building and programming of an inter-connected electronic system. It starts with explaining the fundamentals of the physics underlining the electronic power measurement process, transitioning into describing integrated components/modules used later in the proposed solution, such as ESP8266 Wi-Fi chip, GL.inet router board or OpenWRT - an unix-like operating system. The conceptual design of a final solution, utilising the aforementioned topics, follows. It includes diagrams describing the inner working of the hardware, and later software running on it. The manufactured device is capable of measuring the electric power provided by the electric socket to the appliance and send the measured values over Wi-Fi to the cloud, to be visualised on a custom web server employing a charting library to plot the measured quantities over time.

Keywords

electrical, power, socket, system

Abstrakt

Táto práca opisuje proces navrhovania, realizácie a programovania prepojeného elektronikcého systému. Začína vysvetlením základov fyziky stojacích za meraním elektrického výkonu a postupne prechádza do opisu integrovaných komponentov/modulov použitých neskôr v navrhovanom riešení, ako napríklad ESP8266 Wi-Fi čip, GL.inet dosky alebo OpenWRT - operačný systém na báze unixu. Ďalej nasleduje návrh konceptu finálneho riešenia, používajúceho spomenuté témy. Práca obsahuje diagramy opisujúce fungovanie hardvéru a softvéru, ktorý na ňom beží. Vyrobené zariadenie je schopné merať eletrický výkon dodaný zásuvkou spotrebiču a posielať namerané hodnoty cez Wi-Fi do cloud-u, aby boli následne zobrazené ako graf výkonu a času.

Kľúčové slová

elektrický, výkon, systém, zásuvka

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 28, 2016 .....

Signature

Acknowledgement

I would like to express my sincere thanks to my supervisor Ing. Tibor Vince, PhD., for his constant and constructive guidance throughout all the struggles that have occurred during this work. And to all other who gave a helping hand, I say thank you very much, too.

Preface

This thesis is about constructing a system of co-working electronic devices. The main input of such system is the measured electric power with respect to time. It requires problem solving skills, analytical thinking and decision making. Among core technical skills used are electronics design, firmware programming and product documentation, which are all part of the product life-cycle. Practising and improving named skills was the main motivation to choose this topic. One another motivator worth mentioning is the requirement to understand and use unix-like operating system, which is also desired skill to be honed.

Contents

List of Figures

List of Tables

Acronyms

AC	Alternating Current
ADC	Analog-to-digital converter
AP	Access Point
ASIC	Application-specific integrated circuit
BOM	Bill of the materials
BSD	Berkeley Software Distribution
CLI	Command-line interface
CPU	Central processing unit
DAC	Digital-to-analog converter
DC	Direct Current
DDNS	Dynamic Domain name server
DDR	Double data rate synchronous DRAM
DNS	Domain name server
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
HW	hardware
Hz	Hertz, the SI unit of frequency
I/O	Input/Output
I^2C	Inter-Integrated Circuit
I^2S	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
NAT Network address translation
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)

peripheral a 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

The idea is to invent way of measuring the electrical power and some more related information, preferably in a non-invasive way. The non-invasive way means, that the appliance that is being measured does not require any modifications, for instance in a form of some probe or a man-in-the-middle plug, suggesting an embedded system. When the data are obtained, they are presented to the user, preferably plotted as a quantity over time, not just and actual measurement. Since the solution is going to be multi-purpose, it has to incorporate at least one additional function, than just the measurement. In this case it is going to be the remote power-on/power-off of the appliance. The name of the thesis also suggests, that the final solution has to be compatible with the electrical sockets used in the local region, in this case the European ones. Since the solution is going to be an *embedded system* measuring a *physical quantity*, these two topics are described in following chapters.

There are two fixed components/modules that are going to be included in the work, mostly because if the research purposes, but also because they fit the implementation greatly and are accessible. The components are GL.inet router board and ESP8266 Wi-Fi module. The fact that these two are already chosen beforehand will streamline the design and development and will narrow down the possible implementation paths a little. Topics revolving around these two are also complex on their own, thus they are deserving their own chapters.

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 [23]

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

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 [13]. 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[14], 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 [42]. 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

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

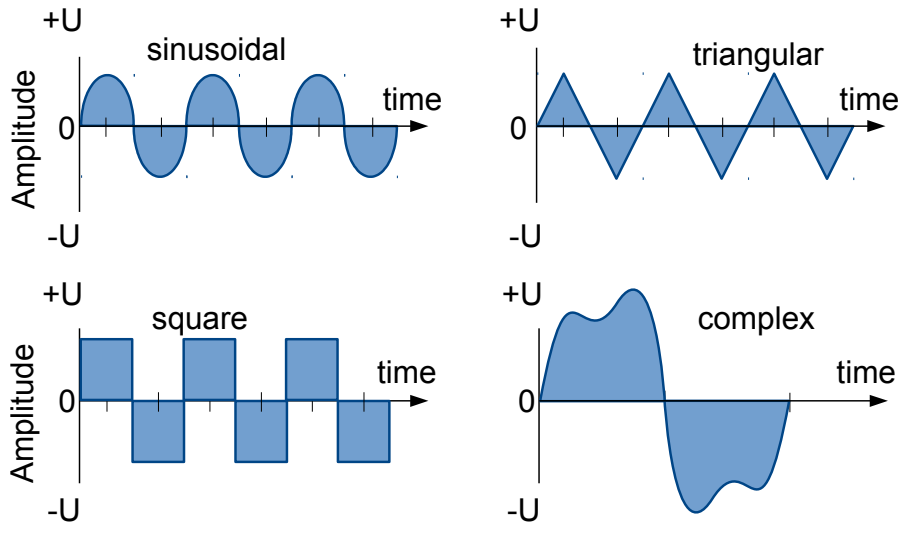

Figure 1 – 1 The common types of waveforms visualised as a function of amplitude

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 [24] 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 [25].

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[32] 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 [22].

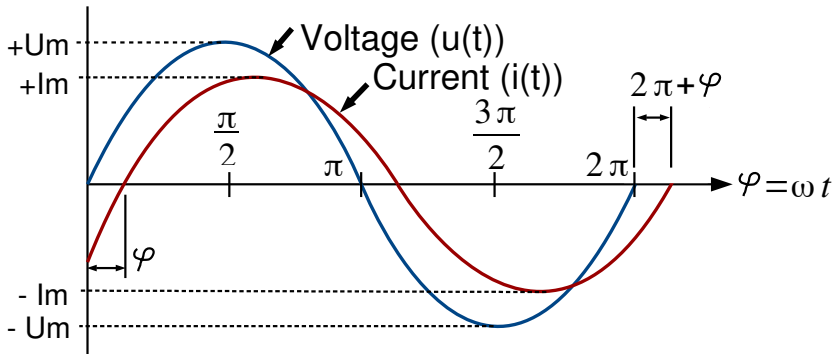

Figure 1 – 2 The phase difference between voltage (blue) and current (red), the origin of phase difference of angle φ

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[9]. 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 [33] (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 [40] 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.

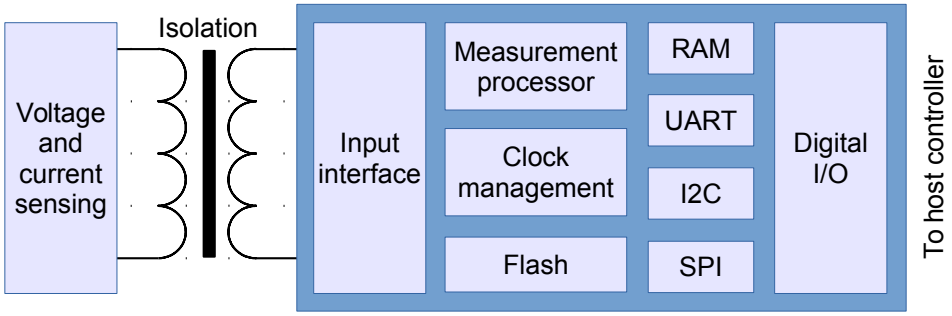

Figure 1 – 3 The simplified block diagram for a power measurement IC

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 [34] 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 [11]. 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 [10]. 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 [15]. 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[20]. 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[12].

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.

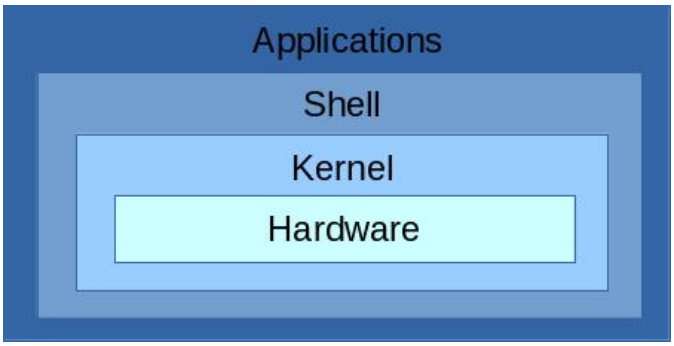

Figure 2 – 1 The simplified view on the Linux system structure

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^2S)/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

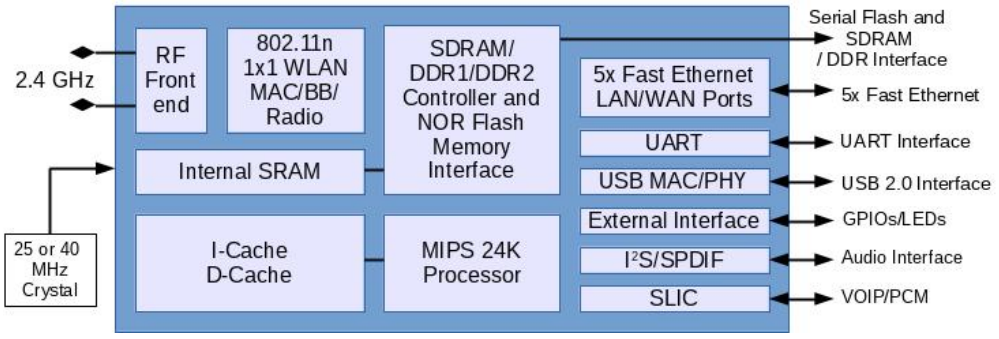

Figure 3 – 1 The block diagram of the Atheros AR9331 SoC used as a main processing unit on GL.inet board

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.

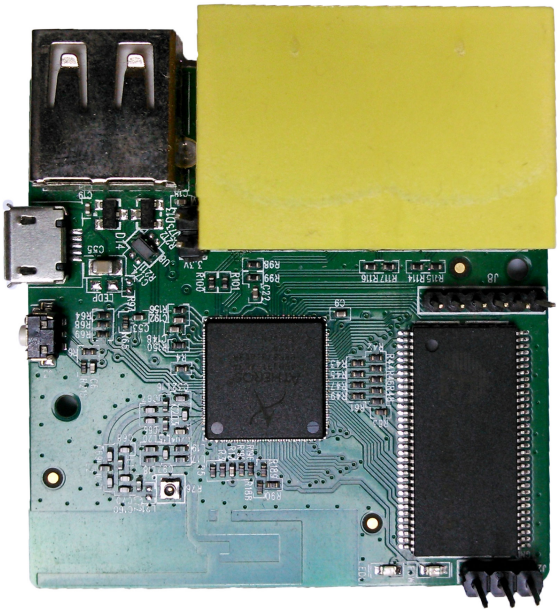

Figure 3 – 2 The front side of the GL.inet board exposing the main Atheros SoC, RAM and interfaces

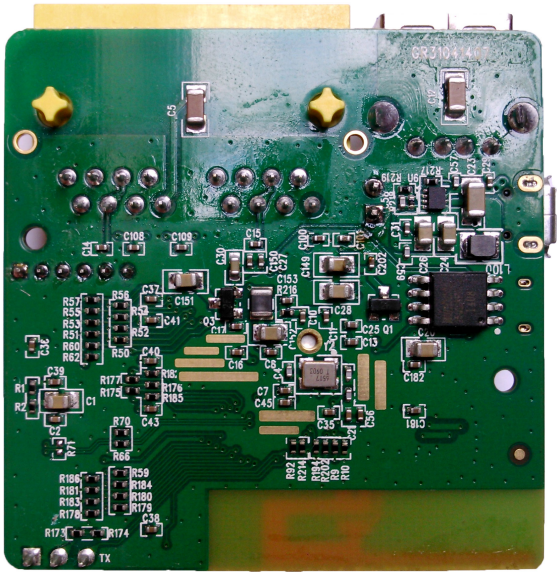

Figure 3 – 3 The back side of the GL.inet board exposing the Flash memory and a main voltage regulator

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.

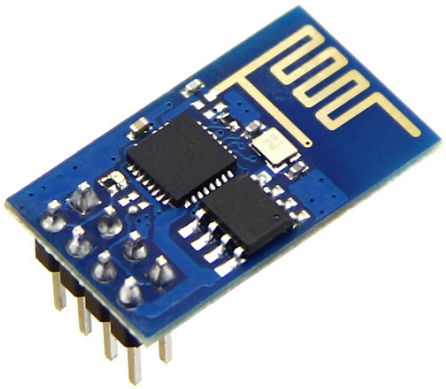

Figure 4 – 1 The first commercial iteration of the ESP8266 module, the ESP-1, exposing two GPIOs

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

4.2 The boot-up process

The Espressif code can boot in different modes, selected on power-up based on GPIO pin levels. Details are listed in the truth table 4 – 1.

Table 4 – 1 The ESP8266 boot-up process, based on pin levels

GPIO15	GPIO0	GPIO2	Mode	Description
L	L	H	UART	Download code from UART
L	H	H	Flash	Boot from SPI Flash
H	x	x	SDIO	Boot from SD-card

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 (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

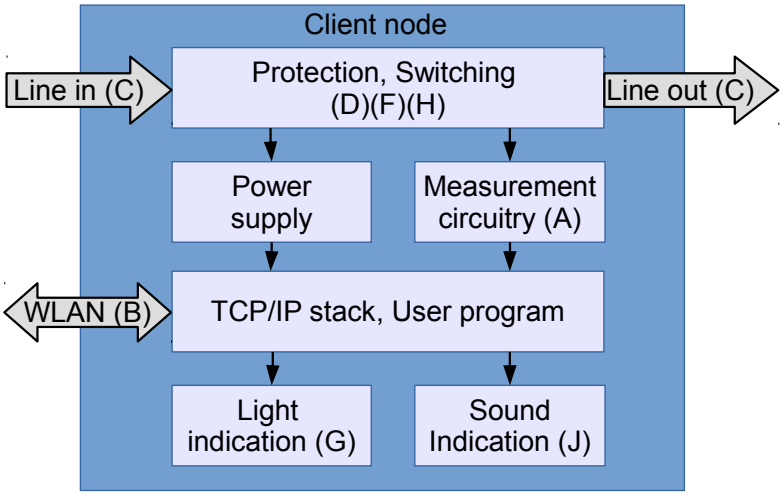

Figure 5 – 1 The proposed block diagram of a *client node*, including HW requirements

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

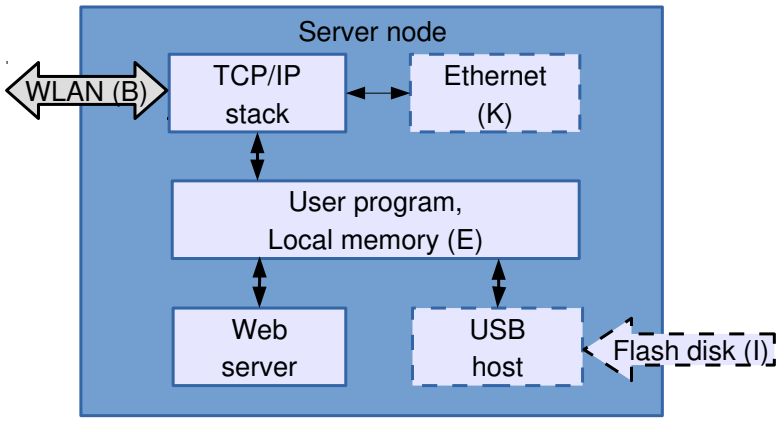

Figure 5 – 2 The proposed block diagram of a *server node*, including HW requirements

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[39] 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 node (displayed as a simplified schematic in 6 – 1) is already integrated as an 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 ESP-

12E has been chosen as an actual module, because of the available certification[31], which allows it to be introduced on the market later. It was already shown in the figure 4 – 2. The Pulse-width modulation (PWM) is present there too, so sound indication requires just a sound emitting device.

Talking about the measurement circuitry, the viable candidate is MAX78615 [16] with the companion IC MAX78700 [17]. 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.

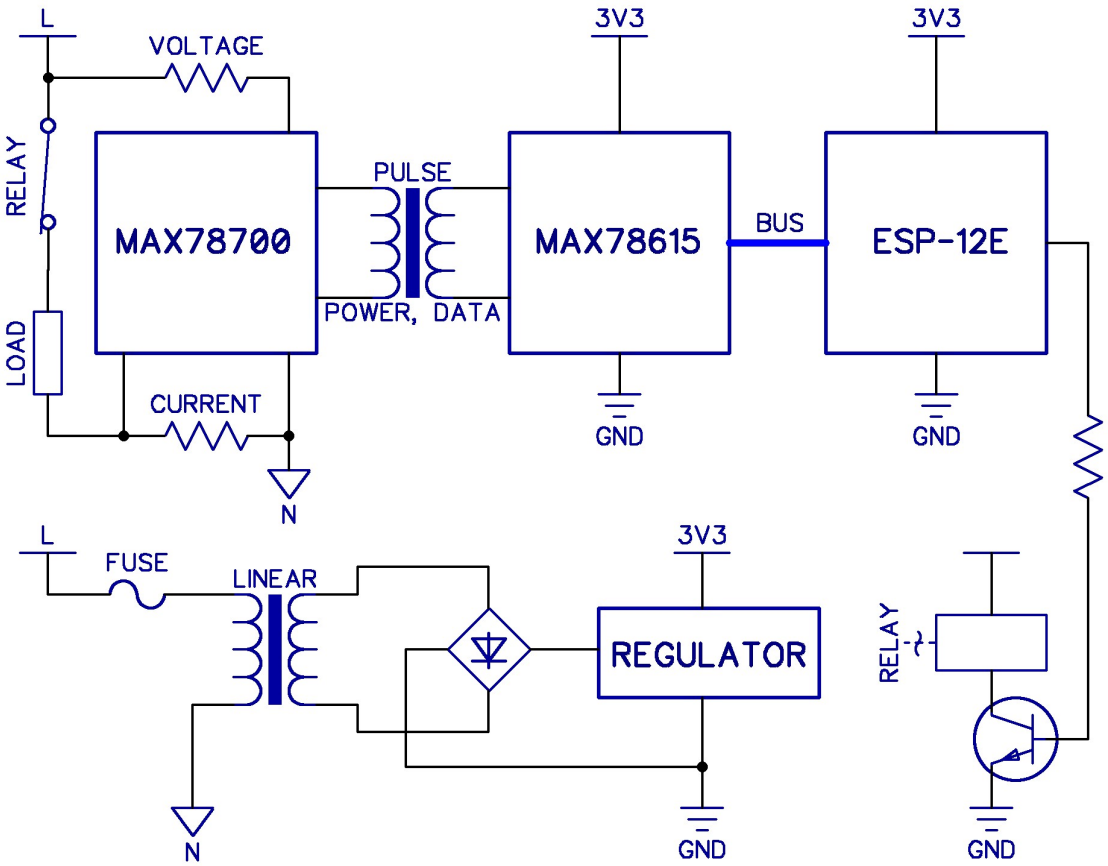

Figure 6 – 1 Greatly simplified schematic of a client node sketching the inner working

For the protection against fire a standard glass fuse or a resettable Positive thermal coefficient (PTC) fuse[43] should be used. Because of the variable nature of most used 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 a linear transformer coupled with a linear voltage regulator[21].

Since 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 [16], working on nominal 3.3V level, along with the companion IC MAX78700 [17]. The couple has been chosen, because it provides multiple ways of communication with the processor (buses/serial interfaces), galvanic isolation via a 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[37] 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

The detailed schematic can be seen in the figure 6 – 2, along with the Bill of the materials (BOM), shown in the table 6 – 3. Some parts of it are quite straightforward, but some require less or more description, to avoid confusion.

In the top section, there is a power supply circuit, which consists of the linear transformer T1, stepping the mains voltage down to the comfortable level of less than 10V AC. The BR1 bridge rectifier MYS80[29] is a compact unit, capable of providing up to 500mA@~10V continuous current, which is split between a relay K1 and a U1 linear voltage regulator LD1117S33[35]. Regulator can provide up to 800mA@3V3 of current, which is more than needed, but was picked for its price. Most of the current it provides (around 250mA) is consumed by U4, an ESP-12E module. All the other connected devices, including the relay, have at least an order of magnitude lower current consumption, thus the current provided should suffice.

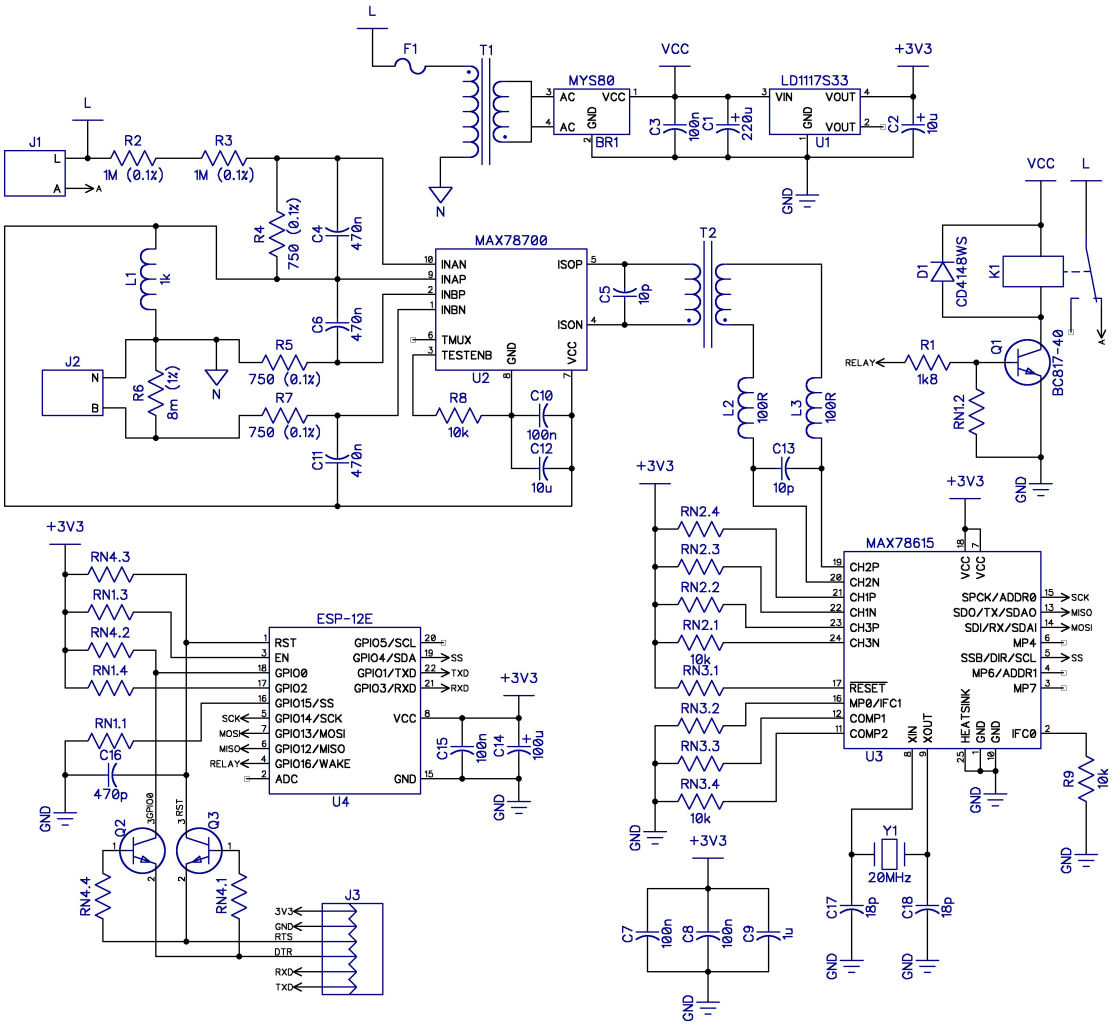

Figure 6 – 2 Full client node schematic, providing all the details

Surrounding capacitors are for smoothing and for preventing the regulator from oscillating.

The K1 relay RM96-1011-35-1009[26] circuit in the top right corner shows relay coil connected as a low-side switch over a transistor Q1. The coil is consuming around 30mA and the transistor BC817-40 [28] can handle far more, but it was picked, because of its availability. Base of the Q1 is biased over a resistor R1 to be fully open at 3.3V, provided by U4 pin high level. Resistor RN1.2 is weakly pulling the base down until U4 boots up and starts maintaining pin states, to prevent bouncing. The relay itself was picked up, because of low operational voltage, low price and, current consumption and small dimensions. It is cofigured as NC (normally connected), so in case of failure at U4, the appliance will still function as expected. Another less obvious treat of NC is that pulse transformer T2 will not be saturated by coil's magnetic field (which would prevent its function), more details later. The flywheel diode D1 is anti-parallel to the coil, providing a

way for a current to flow, when Q1 is cut-off (the coil's collapsing magnetic field is discharged trough D1 to prevent dangerous negative voltage to build up at the collector of Q1). Practically any semiconductor diode would suffice, provided it can handle the peak current. In this case, the common 1N4148[7] was picked, again, because of it's availability and price.

Table 6 – 1 The truth table of logic implemented by the transistor Q2 and Q3 providing automatic programming and communication interface for ESP-12E

DTR	**RTS**	**RST**	**GPIO0**
H	H	H	H
L	L	H	H
H	L	L	H
L	H	H	L

The application processor U4, an already mentioned ESP-12E module, is connected in a configuration, that allows it to be programmed **and** to communicate over a serial link without hassle. This functionality is achieved by two standard NPN transistors Q2 and Q3, utilising the DTR and RTS pins of a serial link, switching GPIO0 into required level. The logic implemented by the transistors is described in the table 6 – 1. The required level is, on the other hand, based on the boot-up state described previously in the table 4 – 1. The U4 is decoupled by a high-value capacitor for current surges, when using Wi-Fi. The low value capacitor is added for a greater decoupling frequency response.

Table 6 – 2 The configuration pins selecting the serial interface of the MAX78615

Interface mode	**IFC0**	**IFC1**
SPI	L	X (don't care)
UART	H	L
I^2C	H	H

The U3, the data processing unit MAX78615[16], has two configuration pins, selecting the serial interface (communication bus) used to talk to the application processor U4. The truth table for the pins can be seen in the table 6 – 2. The SPI has been chosen, because unlike UART and Inter-Integrated Circuit (I^2C), it is able to communicate faster, than the sampling frequency of the ADC (U2), thus getting instantaneous measurements, in case the data are needed. SPI requires two more pins from U4, but there are some spare ones, so this is not a problem. Also, the UART cannot be used anyway, because U4 uses it for programming and

debugging. The unit is utilising external 20MHz crystal and both its power supply pins are decoupled by small value capacitors.

The last part of the schematic is the actual measuring circuitry around the U2, the MAX78700[17] ADC. The whole circuit is obtained from reference design[18] provided by the chip manufacturer. It communicates and obtains the power from U3 via the pulse transformer T2. Care has to be taken when surrounding components emit strong magnetic field, because T2 core could get saturated by it, preventing correct function. This is also the reason, that the relay K1 is in NC configuration - either the relay coil is conducting, or the pulse transformer (in case they are physically situated near each other).

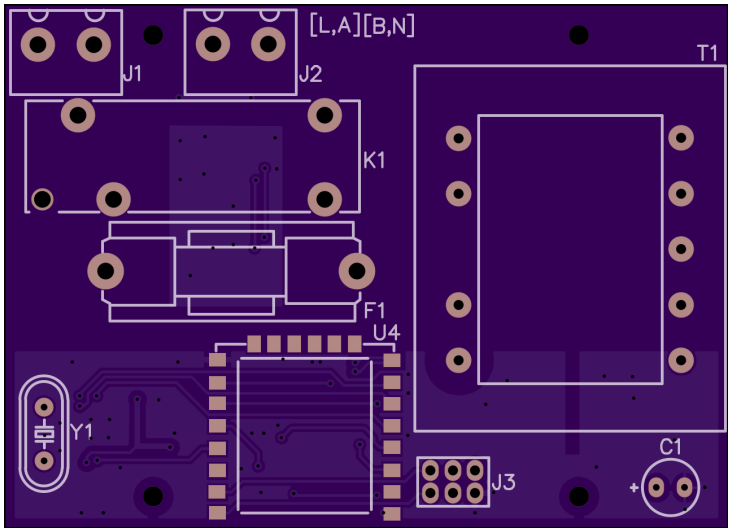

Figure 6 – 3 The top layer of the designed PCB (client node), exposing mainy THT components

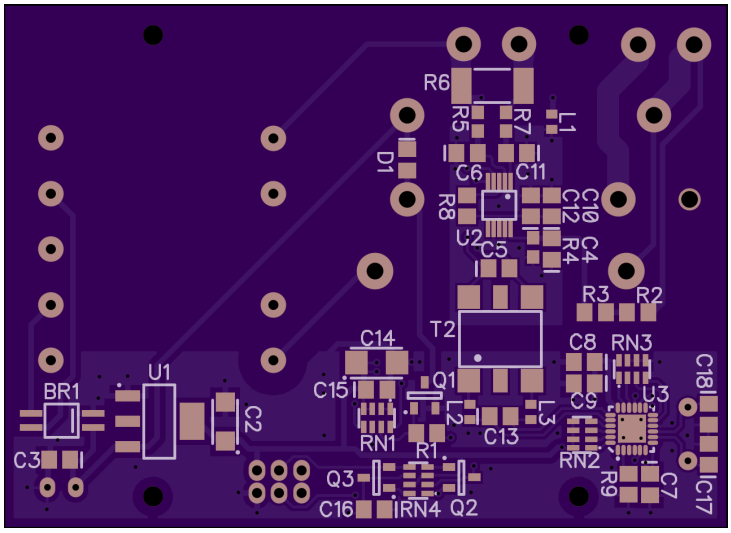

Figure 6 – 4 The bottom layer of the designed PCB (client node), exposing mainly SMT components

Table 6 – 3 The bill of materials used in a client node design

RefDes	Qty	Value	Name	Pattern
BR1	1		MYS80	microDIL
C1	1	220u	CAP100RP	2.54/5.08
C2	1	10u	CAP_TC3216	TC_3216
C3,C7,C8,C10,C15	5	100n	CAP_0805	CAP_0805
C4, C6, C11	3	470n	CAP_0805	CAP_0805
C5, C13	2	10p	CAP_0805	CAP_0805
C9	1	1u	CAP_0805	CAP_0805
C12	1	10u	CAP_0805	CAP_0805
C14	1	100u	CAP_TC3528	TC_3528
C16	1	470p	CAP_0805	CAP_0805
C17, C18	2	18p	CAP_0805	CAP_0805
D1	1	1k8	CD4148WS	DIO_0805
F1	1		PTF/15	PTF/15
J1, J2	2		DG301-5.0	02P-12
J3	1		87758-06	2x3T/2x2/6x4
K1	1		RM96-1011	35-1009
L1	1	1k	MMZ1608S102A	IND_0603
L2, L3	2	100	MPZ1608S101A	IND_0603
Q1	1	1k8	BC817-40	SOT23
Q2, Q3	2		BC817-40	SOT23
R1	1	1k8	RES_0805	RES_0805
R2, R3	2	1M .1%	RES_0805	RES_0805
R4, R5, R7	3	750 .1%	RES_0603	RES_0603
R6	1	8m 1%	RES_2512	RES_2512
R8, R9	2	10k	RES_0805	RES_0805
RN1-RN4	4	10k	4D03WGJ0103T	8/1.6x3.2x0.8
T1	1		TEZ1_5_D_6V	TEZ1_5_D_6V
T2	1		Midcom	750110056
U1	1		LD1117S33	SOT223-4
U2	1		MAX78700	uMax-10
U3	1		MAX78615	QFN-24/4x4x0.5
U4	1		ESP-12E	ESP-12E
Y1	1	20MHz	HC-49US	HC-49US

6.3 Software architecture and components

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[27] 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`[41], while `NodeMCU`[19] is a `Lua` eco-system based on ESP-8266. Both solutions 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)[36]. 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 for a data stream named `phant.io` 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[38], 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[30] 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.

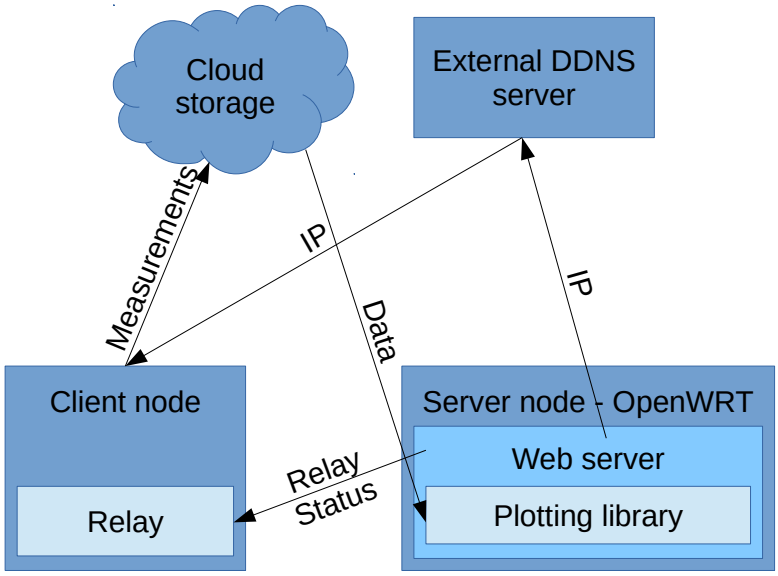

Figure 6 – 5 The block diagram of a data flow in the proposed software architecture, including a client node, a server node, the cloud storage for measurements and an external DDNS server for providing the IP of the server to the client nodes

One of the SW requirements is to provide the ability to add another measuring nodes without modifications of the existing infra-structure. As it is common in a SW world, there are again multiple solution to do this. Since there are so many ways, only the two most viable ones are described. The first is to use automatic configuration sequence, that would be initialized from the server node to communicate to freshly booted up client node. This would work without the Internet connection, but would require additional actions on the side of the user. Another way is to use external DDNS server to provide the IP of the server to the clients. Since it is a centralised resource, all clients get updated simultaneously. It also provides outside of Network address translation (NAT) (local network) access to the server node, which is one of the optional requirements. Since the cloud storage has been already chosen as a primary storage, which requires the Internet con-

nection, external DDNS server has been chosen for this task. Additional positive side-effect of this choice is, that entire process is invisible to the user, thus simpler to use.

All the mentioned SW components are included in a block diagram 6 – 5. The arrows help indicate which way the data are flowing around the the system, to get the better idea.

7 Realisation

The manufactured client node has been inserted into the enclosure containing an European mains socket (female) on one side and an European mains plug (male) on the other side, forming a man-in-the-middle adaptor, that can be non-invasively put between wall socket and an appliance. The result can be observed in figure 7 – 1.

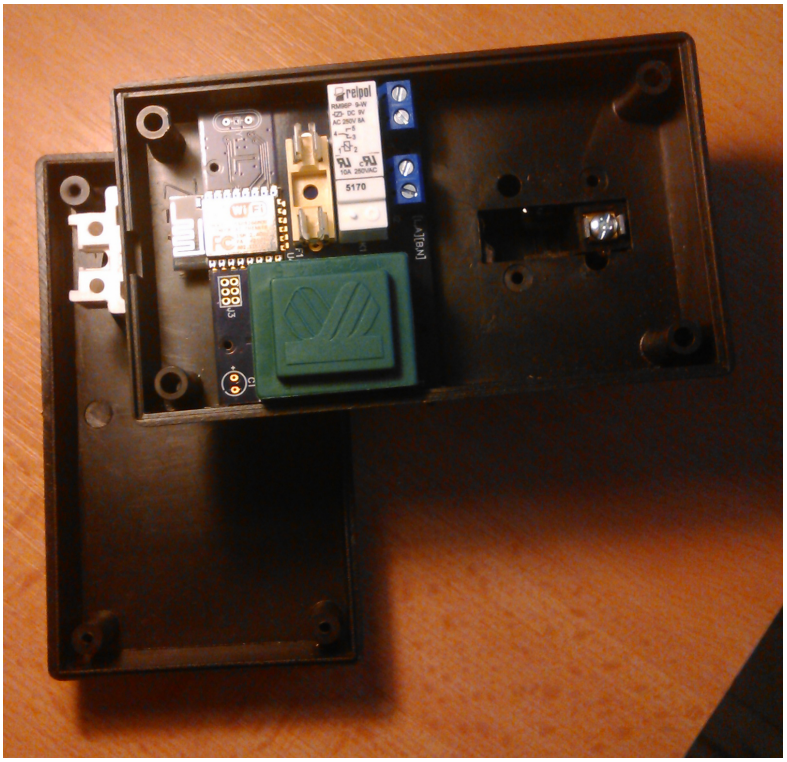

Figure 7 – 1 The view into the client node's enclosure, before the final assembly, exposing top side of the board containing linear transformer T1 (green), mains connectors J1 and J2 (blue), a fuse holder for F1 (yellow-ish), a relay K1 (white) and an ESP-12E module

7.1 Discovered problems

After a few test runs performed on an assembled client node, the first problem became obvious: the node's application processor (contained inside of the ESP-12E module) starts erratically, when the node is plugged into socket. Investigations of the supply voltage under the oscilloscope shows no difference in voltage surges during the boot-up, either if the processor starts or not, suggesting a firmware problem too. The first 220ms of power line inspection can be seen in the figure 7 – 3, showing a voltage fluctuations caused by current surges from the processor starting. However, this problem does not occur, when the processor is powered from external source via J3, but otherwise makes the node unreliable to use.

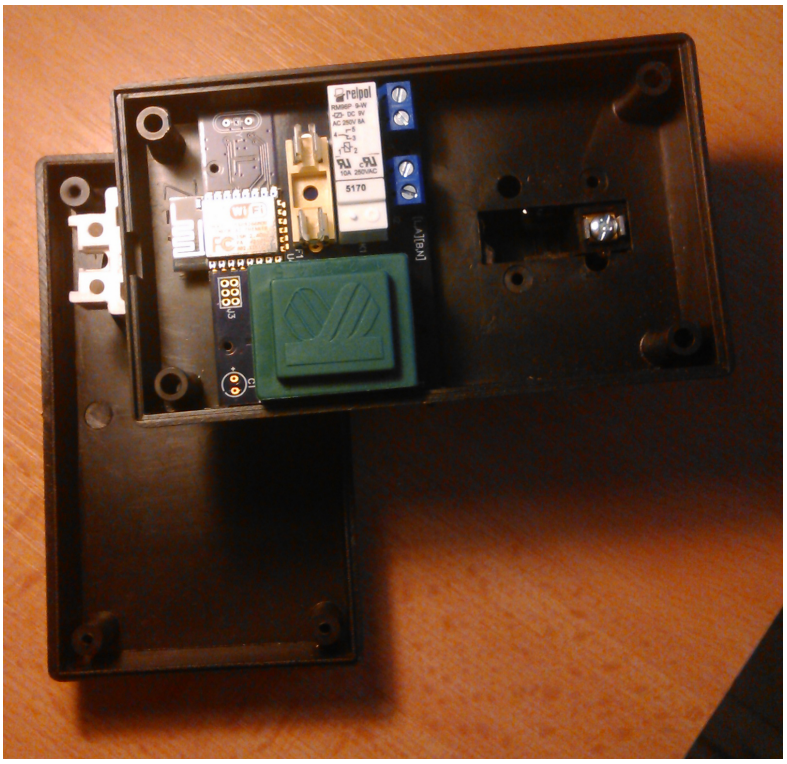

Figure 7 – 2 The view onto a fully assembled client node

Figure 7 – 3 The power line of the ESP-12E inspected during the boot-up by the oscilloscope - it looks the same either if the processors boots, or it doesn't, suggesting the possible occurence of the ESP8266 firmware problem

Ignoring the boot-up problem, the client node is sort of working as indented, apart from one huge problem with the MAX78615 IC, that was not apparent during the design stage: the SPI protocol only allows for 6 bit long memory addressing,

enabling only the first 64 words of the memory to be accessed, leaving the 104 words out of 186 completely inaccessible. As a result, from the required data, only RMS Voltage and RMS Current can be obtained. All the data depending on phase shift, namely real power, reactive power and power factor are not accessible.

The SPI limitation obviously cripples the node's functionality. The protocol was chosen, because the data-sheet for the MAX78615 suggested it as the only way to obtain the instantaneous measurements, as discussed back in the sub-chapter 6.2. Although the instantaneous measurements are not required, there was no reason to not enable this feature in the design stage. The fact, that such a limitation exists was observed too late.

Requesting help from the technical support of the IC manufacturer, the Maxim Integrated, did not help resolve or at least minimize the damage. They responded, that they are no longer supporting the part in question, because whole power measurement department was sold to another manufacturer based in China, Silergy Corp in March 2016.

There are multiple, rather cosmetic issues, that does not affect the functionality of the board, but are imposing small physical troubles. They include the following:

- The J3 header is wrong pitch (2mm instead of 2,54mm)
- The mounting holes are too small diameter
- The longer side of the board is 1mm wider than desired, it doesn't fit easily into the enclosure

7.2 Data representation within the web interface

Since there exists a possibility to read the RMS Current drawn by the appliance at measured RMS Voltage, at least the apparent power can be obtained by multiplying it, described in deeper detail back in a sub-chapter 1.4.

Having something to work with gives opportunity to continue the work, despite the fact that the desired real power will not be obtained. Client node makes statistical median of the measured current and voltage and every 10 seconds sends them to the could database, creating a data stream. The sample of the stream can be seen in the table 7 – 1. The timestamp is in the internal format, allowing transformations to any other desired format easily.

Table 7 – 1 Sample of the data of the measurements stored in the cloud database, showing 10 successive samples

current	voltage	timestamp
1.0821	237.1633	2016-04-19T16:26:53.834Z
1.2323	237.0149	2016-04-19T16:26:41.787Z
1.0335	234.3102	2016-04-19T16:26:31.168Z
1.6476	233.5277	2016-04-19T16:26:19.157Z
1.0879	235.6132	2016-04-19T16:26:08.485Z
1.2149	238.0802	2016-04-19T16:25:57.706Z
1.1474	237.1673	2016-04-19T16:25:45.881Z
0.9841	238.3774	2016-04-19T16:25:35.115Z
0.9922	238.8315	2016-04-19T16:25:23.355Z
0.9598	236.3844	2016-04-19T16:25:12.715Z

The data from the could data stream are then visualised on the server node via plotting library, provided by GoogleAPIs. The web interface can be accessed via `http://vameter.ddns.net`, a DDNS redirect service provided by `http://no-ip.com`. It traslates the Domain name server (DNS) to the IP of the server node on the local network. If there were multiple client nodes, and the IP of the server node should change, this way all of them would be updated. The illustrative results can be seen in the figure 7 – 4. The mentioned apparent power is obtained by multiplying current and voltage, as has been already stated, so these values are not stored or displayed explicitly.

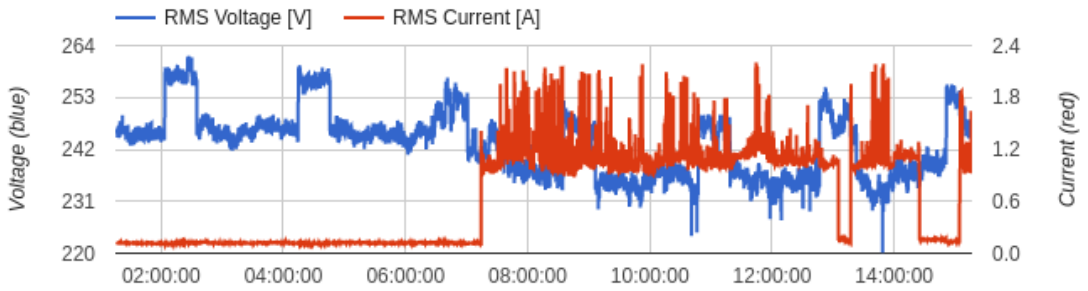

Figure 7 – 4 The preview of the visualised data from the data stream displaying 14 hours of data; it can be seen when the appliance was in standby mode (red) or the periodical fluctuations of the mains voltage (blue)

The web interface also contains a button for shutting the appliance on or off (by controlling the electromechanical relay K1). It consists of simple ON/OFF button that changes its value accordingly to the relay's state. It also uses the mentioned DDNS service to contact the server node.

8 Conclusion

The main requirement of the work was to measure the real, reactive and apparent electric power, as well as power factor, and to show the values to the user, preferably, plotted as a quantity over time. It had to utilise the ESP8266 chip and the GL.inet board in the process. The work's requirements were, however, met only partially. The problem is caused by the inability of obtaining most of the sought values from the power measuring integrated circuit, the MAX78615 over the SPI communication protocol. As a result, only the apparent power could be displayed, subsequently preventing the requirements relying on the other required data from completion. All the attempts to resolve the problem with Maxim Integrated technical support resulted in dead end. The stated reason was, that the part in the question is no longer supported by Maxim. It was sold with all the accompanying parts to the Silergy Corp. in the critical part of the firmware development. Despite of the problems, all the remaining requirements that could be rationally implemented were met.

8.1 Possible future improvements

The most crucial thing to improve is to make the client node's start reliable. The ESP-12E module sometimes won't start. After this problem has been resolved, one can move on to some other issues, but not before.

It is possible to fulfil all the failed requirements due to inability of accessing all the data over SPI by switching to I^2C. The only downside of such a design is inability to obtain all the instantaneous measurements. This is not a problem at all, because a lot of them still lies in the region inaccessible by SPI (SPI is the only protocol fast enough to be capable of reading instantaneous measurements before they are being over-written at a sampling frequency - this should be fixed by the IC manufacturer). The frequency at which I^2C is capable of obtaining measurements is perfectly sufficient anyway. Also, I^2C requires one less pin than SPI, thus leaving more pins on the ESP8266 for additional features. Such a transition would however definitely require another iteration of the PCB, which is not instant and requires a bit of resources. If a project is to be maintained in the future, this is definitely a vital change.

Speaking of a PCB iteration, another opportunity to improve the PCB design lies in utilising the MAX78615 in-built relay controlling mechanism. Fundamental part of responsibilities of the IC is to track the zero-crossing. Relay tied to the

zero-crossing allows for graceful start of appliances. This is helpful in preventing damage to some sensitive appliaces when started at the point of mains line voltage peak, such as incandescent light bulbs[8]. Relay is also not essential for actual measurements, so it could be removed from the system altogether to save some cost and space, in applications where the relay is not needed.

There is always room for improvement, but the last bigger proposed change is to completely remove the GL.inet board from the system. It was included mainly for research and practice reasons, not so much to provide some necessary functionality. If we wanted strictly cloud-connected client devices, given there is a trend in increasing count of such a devices, this would reduce cost and complexity. Such a change would require the client devices to be connected to the Internet all the time and to move the web server from the GL.inet (server node) to some other place, preferably where the could storage is hosted. It would also eliminated the need for an external DDNS server.

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] DC Components. *1N4148 Surface-mount switching diode*. (Accessed on 19/04/2016). URL: `http://www.tme.eu/sk/Document/56f18f971963fb7c3e40973a6fd3d82c/CD4148W_WT.pdf`.

[8] C. DiLouie. *Lighting Controls Handbook*. Fairmont Press, 2008. ISBN: 9780881735741.

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

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

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

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

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

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

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

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

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

[18] Maxim Integrated. *Sonoma (MAXREFDES14): Isolated Energy Measurement Subsystem Reference Design*. (Accessed on 19/04/2016). URL: https://www.maximintegrated.com/en/app-notes/index.mvp/id/5723.

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

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

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

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

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

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

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

[26] Relpol. *RM96 Miniature Relays*. (Accessed on 19/04/2016). URL: http://www.tme.eu/en/Document/f3d8c9df965fd60743222f30152cd5e7/e_RM96.pdf.

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

[28] Diotec Semiconductor. *BC817 Surface Mount General Purpose NPN Transistor*. (Accessed on 19/04/2016). URL: http://www.tme.eu/sk/Document/734370d4141e61893b10180687485177/bc817.pdf.

[29] Diotec Semiconductor. *MYS80 Bridge Rectifier*. (Accessed on 19/04/2016). URL: http://diotec.com/tl_files/diotec/files/pdf/datasheets/mys40.

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

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

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

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

[34] 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.

[35] STMicroelectronics. *LD1117 Linear Regulator*. (Accessed on 19/04/2016). URL: `https://www.sparkfun.com/datasheets/Components/LD1117V33.pdf`.

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

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

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

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

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

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

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

[43] 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 CD with electronic form of thesis, schematics source files, firmware source code and all the software needed to open and compile the all the provided source files