

JUNE 2017

M.Sc. in Electrical and Electronics Engineering

AHMED ARIF MOHAMMED

**UNIVERSITY OF GAZIANTEP
GRADUATE SCHOOL OF
NATURAL & APPLIED SCIENCES**

**DEVELOPMENT OF EMBEDDED SYSTEM FOR MAKING
PLUGS SMART**



**M. Sc.THESIS
IN
ELECTRICAL AND ELECTRONICS ENGINEERING**

**BY
AHMED ARIF MOHAMMED
JUNE 2017**

Development of Embedded System for Making Plugs smart

M.Sc. Thesis

in

Electrical and Electronics Engineering

University of Gaziantep

Supervisor

Prof. Dr. Ergun ERÇELEBİ

by

Ahmed Arif MOHAMMED

June 2017



© 2017 [Ahmed Arif MOHAMMED]

UNIVERSITY OF GAZIANTEP
GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES
ELECTRICAL AND ELECTRONIC ENGINEERING DEPARTMENT

Name of the thesis: Development of Embedded System for Making Plugs smart

Name of the student: Ahmed Arif MOHAMMED

Exam date: 12. 06. 2017

Approval of the Graduate School of Natural and Applied Sciences



Prof. Dr. A. Necmeddin YAZICI

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.



Prof. Dr. Ergun ERÇELEBİ

Head of Department

This is to certify that we have read this thesis and that in our consensus; it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.



Prof. Dr. Ergun ERÇELEBİ

Supervisor

Examining Committee Members:

Signature

Prof. Dr. Ergun ERÇELEBİ

Assoc. Prof. Dr. Ahmet ALKAN

Assoc. Prof. Dr. Sema KAYHAN



I now declare that all information in this document has been obtained and presented in agreement with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Ahmed Arif MOHAMMED

ABSTRACT

DEVELOPMENT OF EMBEDDED SYSTEM FOR MAKING PLUGS SMART

MOHAMMED, AHMED ARIF

M. Sc. in Electrical and Electronics Engineering

Supervisor: Prof. Dr. Ergun ERÇELEBİ

June 2017

65 Pages

Smart Switch allows switching almost any appliance ON and OFF remotely via local network or via the internet. Such switches are designed to receive start / stop commands and to monitor smart switch applications such as voltage, current, and power.

The study presents a smart power switch system that can be controlled wirelessly or via the internet, which has an ability control power outlet and monitors electrical events such as voltage, current and power consumption. This system embeds the program into microcontroller to control overall process, in addition to voltage and current sensor. The main system features include the remote control of multiple power outlet, control smart plugs, real-time monitoring of the current, and voltage and power consumption. Also, it schedules the devices on/off time with consumption limits, calculates the electrical cost that user can interface it with any operating system. A prototype of the system has been implemented and tested with the local network and the internet in order to monitor its functionalities. The overall results proved that system runs fast and has very close energy monitoring to common monitoring devised. This system can be replaced manual distribution board and old energy meter to have one auto-remoted real time monitoring smart switch.

Keywords: smart switch, smart home, power switch, power meter.

ÖZET
PRİZLERİ AKILLI YAPMAK İÇİN GÖMÜLÜ SİSTEMİN
GELİŞTİRİLMESİ

MOHAMMED, AHMED ARIF

Yuksek Lisans Tezi, Elektrik ve Elektronik Mühendisliği

Danışman: Prof. Dr. Ergun ERÇELEBİ

Haziran 2017

65 Sayfa

Akıllı Anahtar, hemen hemen her cihazın yerel ağ üzerinden veya internet üzerinden uzaktan AÇIK ve KAPALI anahtarlanmasına izin verir. Bu tür anahtarlar başlangıç / durdur komutlarını almak ve voltaj, akım ve güç gibi akıllı anahtar uygulamalarını izlemek üzere tasarlanmıştır. Çalışma, kablosuz olarak kontrol edilebilen veya internet üzerinden kontrol edilebilen, güç kontrollü güç çıkışına sahip ve voltaj, akım ve güç tüketimi gibi elektrik olaylarını izleyebilen akıllı bir güç anahtar sistemini sunar. Bu sistem voltaj ve akım algılayıcısının yanı sıra tüm işlemleri kontrol etmek için mikro denetleyiciye program gömmektedir. Ana sistem özellikleri, çoklu priz in uzaktan kumandası, kontrol akıllı fişleri, akım, gerilim ve güç tüketiminin gerçek zamanlı izlenmesini içerir. Ayrıca, cihazları tüketim sınırlarıyla birlikte açma / kapama zamanlarını ayarlar, kullanıcının herhangi bir işletim sistemi ile ara yüzleyebileceği elektrik maliyetini hesaplar. Sistemin bir prototipi, işlevlerini izlemek için yerel ağ ve internet ile uygulanmış ve test edilmiştir. Genel sonuçlar, sistemin hızlı bir şekilde çalıştığını ve ortak izlemeye çok yakından enerji izlediğini kanıtladı. Bu sistem manüel dağıtım panosu ve eski enerji sayacının yerini alabilir ve otomatik olarak uzaktan izleme gerçek zamanlı akıllı anahtarı seçebilir.

Anahtar Kelimeler: akıllı anahtar, akıllı ev, güç anahtarı, güç ölçer.



Dedicated
To
My Country
And
Beloved Parents

ACKNOWLEDGEMENTS

First and most importantly I must thank the Most Merciful, Almighty ALLAH for making everything possible.

I would like to thank my supervisor: **Prof. Dr. ERGUN ERÇELEBI** Head of Electrical & Electronic Engineering Department for his support and guidance throughout the study, motivation and encouragement throughout the period this work was carried out.

Also, I would like to express my sincere thanks to all member of my family for supporting me in every situation.

I would like to provide special thanks to my father: **Prof. Dr. Arif Mohammed** for providing moral and intellectual support to me in my study.

I do not have sufficient words to thank my first ever teachers and my parents. I would like to offer my sincere thanks to all my friends.

.

TABLE OF CONTENTS

	Page
ABSTRACT.....	v
ÖZET.....	vi
ACKNOWLEDGEMENTS	viii
TABLE OF CONTENTS.....	ix
LIST OF TABLES	xiii
LIST OF FIGURES	xiv
LIST OF SYMBOLS/ABBREVIATIONS	xvi
CHAPTER 1	1
INTRODUCTION	1
1.1 Overview	1
1.2 Literature Survey.....	2
1.3 Aim of Thesis	3
1.4 Organization of Thesis	4
CHAPTER 2	5
THE ORETICAL BACKGROUND.....	5
2.1 General.....	5
2.2 Smart Swithch Hardware Component.....	5
2.2.1 Microcontroller Unit	5
2.2.1.1 Bits.....	6
2.2.1.2 Memory.....	7
2.2.1.3 Instruction Set.....	7

2.2.1.4 Memory Architecture.....	7
2.2.2 Available Types of Microcontroller.....	8
2.2.2.1 8051 Microcontroller.....	8
2.2.2.2 AVR Microcontroller.....	9
2.2.2.3 Programmable interface controller (PIC) Microcontroller.....	9
2.2.2.4 ARM Microcontroller.....	9
2.2.3 Arduino Microcontroller Boards.....	10
2.2.3.1 Arduino Mini / Mini Lite.....	10
2.2.3.2 Arduino Pro Mini 3.3V / Pro Mini 5 V.....	11
2.2.3.3 Arduino Nano / Nano Lite.....	12
2.2.3.4 Arduino Fio.....	13
2.2.3.5 Arduino Lily Pad / Simple Lily Pad.....	13
2.2.3.6 Arduino Leonardo.....	14
2.2.3.7 Arduino Pro 3.3V / Pro 5V.....	15
2.2.3.8 Arduino Diecimilla / Duemilanove / Uno.....	16
2.2.3.9 Arduion Ethernet / Ethernet PoE.....	17
2.2.3.10 Arduino Bluetooth.....	18
2.2.3.11 Arduino Mega 1280 / 2560.....	18
2.2.3.12 Arduino Mega ADK.....	20
2.2.4 Electrical Sensors.....	20
2.2.4.1 Current and Voltage Sensors.....	21
2.2.4.2 Light Sensors.....	22
2.2.4.3 Infrared Sensors.....	22
2.2.4.4 Relay Module.....	23
2.3 Smart Switch Software Component.....	23
2.3.1 Hardware Programming.....	24
2.3.2 User Interface Programming.....	25

CHAPTER 3	27
TECHNIQUES FOR REACTIVE POWER AND VOLTAGE CONTROL.....	27
3.1 Introduction.....	27
3.2 System Design.....	27
3.2.1 Hardware Design.....	28
3.2.2 Software Design.....	31
CHAPTER 4	33
EXPERIMENTAL RESULTS.....	33
4.1 Introduction.....	33
4.2 System Performance Results.....	33
4.2.1. Test the reading accuracy	33
4.2.2. Stability Results	35
4.3 System Analyses.....	39
4.3.1. Reliability	39
4.3.2. Ease of Installation and Management.....	39
4.3.3. Communication.....	39
4.3.4. Cost.....	39
CHAPTER 5	40
CONCLUSION, SUGGESTION FOR FUTURE WORK	40
5.1. Conclusion.....	40
5.2. Suggestion for Future Work	41
REFERENCES.....	42
APPENDIX 1	46
Smart Switch System Components.....	46
APPENDIX 2.....	47
Voltage and current Readings.....	47
APPENDIX 3.....	52
Smart phone application GUI.....	52

APPENDIX 4.....	53
Smart Switch System Arduino C Code.....	53



LIST OF TABLES

	Page
Table 4.1 Smart switch reading for multiple devices.....	34
Table 4.2 Clamp meter reading for multiple devices.....	35



LIST OF FIGURES

	Page
Figure 2.1 Arduino Mini/Mini Lite board.....	10
Figure 2.2 Arduino Pro Mini board	11
Figure 2.3 Arduino Nano / Nano Lite board.....	12
Figure 2.4 Arduino Fio board	13
Figure 2.5 Arduino LilyPad board	14
Figure 2.6 Arduino Leonardo board	14
Figure 2.7 Arduino Pro 3.3V / Pro 5V board	15
Figure 2.8 Arduino Duemilanove board (left), Arduino Uno (right)	16
Figure 2.9 Arduino Ethernet / Ethernet PoE board.....	17
Figure 2.10 Arduino Bluetooth board.....	18
Figure 2.11 Arduino Mega 1280 / 2560 board	19
Figure 2.12 Arduino Mega ADK 1280 / 2560 board.....	20
Figure 2.13 Arduino current sensors.....	21
Figure 2.14 Arduino light sensor.....	22
Figure 2.15 Arduino light sensor.....	23
Figure 2.16 Arduino relay module	23
Figure 2.17 Arduino IDE Program	25
Figure 2.18 “Mit inverter” software	26
Figure 3.1 Architecture of smart switch communication.....	28
Figure 3.2 Arduino mega 2560.....	29
Figure 3.3 Circuit Diagram of a prototype voltage sensor.....	30
Figure 3.4 Ethernet shield	30
Figure 3.5 system circuit scheme of the smart switch.	31
Figure 3.6 Smart switch Android platform structure	32
Figure 4.1 Smart switch test setup	34
Figure 4.2 Iron reading records	36
Figure 4.3 Oven reading records	37

Figure 4.4 Microwave reading records..... 38



LIST OF SYMBOLS/ABBREVIATIONS

VAC	Alternative voltage
Vdc	Direct voltage
f	converting factor
AC	Alternative current
ADC	Analog to digital converter
CCD	charged coupled device
CISC	Complex instruction set computer
CPU	Central Processing Unit
DAC	digital to analog converter
DC	Direct Current
DIP	dual inline package
EEPROM	electrically erasable programmable read-only memory
EPROM	erasable programmable read-only memory
GSM	Global System for Mobile communication
GUI	graphical user interface
I2C	pronounced I-squared-C
IDE	integrated development environment
I/O	input/output
IR	Infrared
LCD	liquid crystal display
MHz	Mega Hertz
PIC	Peripheral Interface Controller
PIR	passive infrared sensor
PLC	Power Line Communication
PWM	Pulse Width Modulation
PoE	power over Ethernet
RAM	Random-access memory
RISC	Reduced Instruction Set Computers

SRAM	Static random-access memory
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
SCL	serial clock pin
SDA	serial data pin
SFRs	special function registers
SMS	Short Message Service
SPI	Serial Peripheral Interface
SPRs	special purpose registers
UI	user interface



CHAPTER 1

INTRODUCTION

1.1 Overview

Over the past decade, power saving has become a top topic over the world. Peoples usually find it hard to program and take care of their energy consumption in their houses. Distribution board that includes circuit breaker belong with Single Pole and Single Throw switch is a common type of conversion switch that usually used in the house. The circuit breaker and single pole/throw switch are press manually to switch on / off electrical devices including main house or building outlet, electrical devices for example fans, lights and etc. coming from a specific place in which the switch is installed on the wall. This would mean, we have to walk towards the switch's place to switch it on or off. However, smart switches can be operating in an intelligent way to allow it to become smart. Thus, we require a smart controller that could control and monitor all operation the of the switch. The benefit of using a smart switch is; it could be controlled from a far distance either from inside the room or outside the room through Wi-Fi or even from worldwide by the Internet [1, 2].

Smart switch allows peoples to monitor the energy consumption with time as well as turn on/off devices automatically or manually in order decrease power consumption (like as standby power for some smart devices), in order to avoid electrical power overload within a smart grid as well as to remotely manage overall switches of home wirelessly. The Smart switch system can be placed in-between the power outlet and any electrical device in order to switch on/off and measure the energy of device that been connected that also can be controlled by the user through the local Wi-Fi or via web by using many devices like mobile phone, tablet, PC, etc. and it allowing user to schedule power on/off depending on several parameters, such as room temperature,

Current electricity price or another factor in order to help to facilitate an additional safe and effective energy lifestyle [3].

In order to get all these processes, several approaches have been studying developing a unique system and one of the most useful strategies who based on the utilized microcontroller to manage these switches. Nowadays, there are common applications for microcontroller technique in the field of electricity, one of the most important application are Smart Home, which is typically remote control or via some portable device (smartphone, etc.) as a controller for electrical switches. The smart switches connected in between electric outlet and plug switch so that it can get manual and remote switch [4, 5].

Nowadays the mobile phones are widely used and it carried by people all the time there for controlling smart switch with mobile has highly recommended by people. Android is an open source operating system and considered more affordable. The development of an app on the Android operating system for the smart home switch will enable the switch can be controlled and monitored from inside the house or anywhere in the world. The Wi-Fi and internet connection is required for the Android smartphone to communicate with the smart switch device. This project is purposely done to design a smart switch that using an infrared sensor to turn on and off without touching the switch and remotely control the switch from a smartphone [6].

1.2 Literatures Survey

During the last year's, studies have improved power outlets by adding various smart functions. In 2006 [7], Lien et al presented a power outlet system which can be managed remotely by utilizing GSM and Bluetooth wireless interfaces. The system is able to switch each individual outlet on and or remotely via sent a Short Message Service (SMS), which then improved by almost the same authors in [8], these improvements involve used Ethernet support along with the capability of determining current consumption. Moreover, these authors provide another similar electric outlet system with PLC (Power Line Communication) module in [9, 10]. They also improved their work in [11, 12] by making both power outlet systems have the ability to turn on/off electric outlets and to monitor current consumption,

however, the latter work is very interesting: it can use different sensors to monitored living activities then improve the power savings of a home.

In 2008, Bai et al [13], presented one of the first wireless sensor networks examples that used on a power outlet system. This system suggested utilizes ZigBee to switch remote outlets on/off and also to determine their current consumption.

In 2012, Lentine et al [14], described the electrical outlet that included an Ethernet unit for a smart power grid to be able to manage the loads autonomously. Their unique system is able to turn on and off the electric outlets, it can monitor current consumption, and has clearly suitable for working in a network whenever a massive amount the power gets from renewable sources.

Shajahan et al 2012 [15], developed a smart plug based on used Arduino microcontroller and Android platform to remotely monitor the power consumption. This smart plug offers cost benefit because it takes advantage of using free software's, offers the simplicity of monitoring several devices but lacks in control. In the same year, Horvat et al [16] present an electric outlet system that based upon PLC and ZigBee. This system is able to manage the state of the outlets and monitor current consumption and it able to Issuance of warnings regarding probable power overloads.

Yan and Shi in 2013 [17], proposed home automation system by using Bluetooth which saves a lot of power and time. The system has been effectively developed and prototyped to control and monitor the lighting condition by Android phone via Bluetooth. The BF10-A microchip has been used to aid the lighting gathering status and gives user interface to manage the lighting. Their Smart Living system can be gradually developing into reality, in which consumer can control their house wirelessly and remotely.

1.3 Aim of Thesis

This study is to aim to design a smart switch distribution board that controls turning on and off of power outlet remotely and monitoring of the current, voltage and power consumption, schedule the time of switching on/off devices that lead to reduced power consumption, in addition, to calculating the electrical cost. This

system can be managed from the user by used any devices like smartphone, tablet, PC, etc. via Wi-Fi local network or via internet smartphone. This study utilized android application to design a user interface platform that controls and monitor smart

Switches. The Wi-Fi and internet connection is necessary to make the Android smartphone communicate and control the smart switch.

1.4 Organization of Thesis

The thesis is divided into five chapters as follows:

- **Chapter One:** This chapter provides a comprehensive introduction to the detection of explosives including an overview, fundamentals, and development based on a survey of these techniques in the recent few years. The literature survey followed by the aims and objectives of the present work.
- **Chapter Two:** provides a general review of smart switch techniques and the fundamental of these approaches. Additionally, this chapter explains the electronic component that could be used for designing smart switches.
- **Chapter Three:** illustrates the steps of designed the proposed smart switch systems.
- **Chapter Four:** gives the results and summarizes a brief decision about this work presented.
- **Chapter Five:** presents the conclusions, and suggestions for the future work that may improve the performance of the project.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 General

The most popular smart switch system includes two most important units: one is the smart switch circuit that controlling the electric power, that contained many electronics and electrical component like: microcontroller unit, electrical sensors, communication units, etc., the other unit is the user controller interaction that making it possible for peoples to manage smart switch like remote controller which interface with Bluetooth, IR, Wi-Fi, cell phone devices, etc. This chapter elucidates the Theoretical Background and Hardware Overview of the smart switch system.

2.2 Smart Switch Hardware Component

The smart switch circuit consists of an electronic and electrical element that join directly to electric power in order to be controlling automatically or by people. In follows most of the important components that enter the creation of a smart switch circuit.

2.2.1 Microcontroller Unit

A microcontroller is a small computer with the majority of the important assist chips onboard. Similar to computers, microcontroller has lots of points in common, such as (CPU) that 'executes' programs. A number of (RAM) that can store data which is variable. A ROM, in which applications that executed can be saved. (I/O) devices which allow communication being arranged with the outside world i.e. link to hardware such as- LCD's, printers, etc. Serial Ports, that are plug-ins give serial connections in between microcontroller & many other add-ons such as parallel port. Counters and timers, that control all timing and counting functions in just a microcontroller (microcontroller could be inbuilt with a number of counters or

timer). Timers are utilized to count outer pulses. The typical functions done by timers are frequency measuring, making oscillations, clock functions, modulations, pulse generations, etc. also there is "Analog to digital converter" called (ADC), that used to transform signals from analog to digital ones. The input signals should be analog towards ADC. The digital signal generation can be used for several digital applications (like measurement gadgets). Also, there is digital to analog converter called (DAC), that runs reverse functions that ADC perform. The ADC system is usually used to supervise analog appliances such as DC motors, etc. Interpret Control, which is utilized for providing delayed control for a working process. The interpret could be external or internal. Special Functioning Block, in which some kind of special microcontrollers constructed for specific appliances such as robots, space systems, etc., include this unique function block. This unique block has extra ports to be able to carry out some specific operations [18].

Microcontrollers could be embedded' within some other type of device (usually a customer product) in order to control the functions or actions of the products. An additional term for a microcontroller is thus an "embedded controller". Specialized in one task and can run one specific program. The program is usually saved in ROM and usually will not modify. A low-power system. A battery-operated microcontroller could possibly consume as low as 50 milli-watts. A microcontroller could take an input coming from the device and controls the device by transmitting signals to various pieces in the device. A microcontroller is frequently low cost and small. The parts could be chosen to minimize size as well as be as low-cost as possible. Typically, the processor utilized to carry out a microcontroller may vary widely. In several systems, like microwaves, the need on the CPU is somewhat low [18].

Microcontrollers can be categories based on its: instruction sets, memory, bits, and architecture that described in details as in follows [19]:

2.2.1.1 Bits

- 8-bits microcontroller: it can execute arithmetic and logic operations. An example of such microcontroller is Intel 8031 and 8051.

- 16-bits microcontroller: It can execute with higher accuracy and performance than an 8-bit microcontroller. An example of this type of microcontroller is Intel 8096.
- 32-bits microcontroller is: it is employed commonly in appliances that automatically controlled such as (implantable medical appliances, office machines, etc). It needs 32-bit instructions to run any arithmetic or logical function.

2.2.1.2 Memory

- **External Memory Microcontroller:** As an embedded structure is constructed having a microcontroller that doesn't contain all the functioning blocks pre-existing within a chip it is called as external memory microcontroller. An example, the 8031 microcontroller doesn't have program memory in the chip.
- **Embedded Memory Microcontroller:** As an embedded structure is constructed with a microcontroller that contains all of the functioning blocks present on the chip it is called "embedded memory microcontroller". an example, the 8051 microcontroller comes with all I/O ports, interrupts, counters and timers, program and data memory, and so it's classified as embedded memory microcontroller.

2.2.1.3 Instruction Set

- Complex instruction set computer (**CISC**): it enables the operator to use 1 instruction instead of many basic instructions.
- Reduced Instruction Set Computers (**RISC**): it lowers the operations time by decreasing the clock cycle for every instruction.

2.2.1.4 Memory Architecture

two types of Architecture for microcontroller

- Princeton Memory Architecture
- Harvard Memory Architecture

2.2.2 Available Types of Microcontroller

A variety of microcontroller families is presented such as 8051, ARM, AVR, PIC, etc. In follows a number of typical microcontroller families [20].

2.2.2.1 8051 Microcontroller

It is an 8-bit microcontroller produced in 1985 by Intel Corporation and the majority of microcontrollers originate from the 8051 families. The 8051 Microcontrollers remain to be a perfect option for an enormous group of experts and hobbyists. It's found in 40 pins dual inline package (DIP). It includes ROM of (4kb) (within-chip programmable space) and RAM space of (128-bytes) of that is built-in, if required 64KB the external memory could be interfaced with the microcontroller. A four 8-bits parallel ports that are easily programmable in addition to addressable. 8051 microcontroller speed is A million instructions per second. The on-chip crystal oscillator is built-in within the microcontroller that has 12MHz crystal frequency. Within the microcontroller, there's a serial I/O port that has 2-pins. Two(16-bits) timers are also integrated into it; these kinds of timers can be used as a timer for interior functioning in addition as a counter for the external functioning. The microcontroller includes five interrupt sources: External Interrupt 0, External Interrupt 1, Timer Interrupt 1, Timer Interrupt 0, Serial Port Interrupt. 8051 microcontroller has programming mode that includes: the special purpose registers (SPRs), the special function registers(SFRs) and the general purpose registers (GPRs). 8051 comes with highly effective instruction set; it provides commands that execute extra complex calculations [20].

An original 8051 microcontrollers have been originally designed by Intel. The 2 fellow members of this 8051 family are [21]:

- **8052:** it has 256bytes RAM, 3timers, belong with all the features that are found in traditional 8051 microcontrollers. The 8051 microcontroller is a part of 8052 microcontrollers.
- **8031:** it is ROM less microcontroller along with other features of a traditional 8051 microcontroller. An external ROM of 64K bytes can be added to chip for execution, 8051 microcontroller provides in to perform two different memory sorts like Flash, UV-EPROM, and NV-RAM.

2.2.2.2 AVR Microcontroller

AVR is a microcontroller of RISC architecture. AVR comes from the developer's names (Alf-Egil Bogen and Vegard Wollan). It has two types 16-bit and 32-bit microcontroller in the same family. It includes 140 instructions all of which are 1 cycle based instructions. AVR microcontrollers normally work with the 1 MHz clock cycle. The AVR speed is 12 million instructions per second. AVR family has an on-chip bootloader. With that, we are able to program microcontroller effortlessly with no external programmer. AVR has a wide variety of interrupts, PWM channels, I/O ports, USART, on-chip analog comparators, A/D converters, timers/counters, and I2C interfaces. It includes Flash, EEPROM, and SRAM which are integrated into a single chip, therefore reducing needing additional external memory. Various appliances include parallel external bus alternative, to add additional data memory gadgets. Almost all appliances, except for Tiny-AVR chips include a serial interface, that is utilized to connect EEPROMs chips and large serial Flash [22].

2.2.2.3 Programmable interface controller (PIC) Microcontroller

PIC available in 8-bit, 16-bit and 32-bit architecture. It has about 40 instructions that all are need 4 clock cycles for executing. The PIC speed is 3 million instructions per second. The PIC programming part is very hard. Thus, it's not recommended for people that getting into embedded world newly. It includes on-chip peripherals like in-system programmability, UART, analog comparator, ADC, internal RC oscillator, I2C, SPI, etc. [23].

2.2.2.4 ARM Microcontroller

ARM comes from company name that designs microprocessors architecture. It had been first created in 1980 by Acorn Computers Ltd. This microprocessor doesn't have flash memory onboard. It is mainly made for microcontroller hardware, it is easy to learn and uses it, even so, it's powerful for the many complicated embedded devices. The architecture of ARM is based on 32-bit RISC processor manufactured by ARM Ltd. Because of the attributes of power-saving, ARM CPU are prevailing in the portable electronics that needs less power consumption [24].

2.2.3 Arduino Microcontroller Boards

Arduino is rapidly the most preferred microcontrollers utilized in various applications. There are various models of Arduino microcontrollers that are different not just in design and characteristics, but also in processing abilities and it's in size. There are several characteristics which are usual to all Arduino boards, which make them very flexible. Almost all Arduino boards are based on ATMEGA AVR series microcontrollers provided by ATMEL that characteristic both of those digital and analog pins. Furthermore, Arduino developed a software program which can be suitable for almost all Arduino microcontrollers. The software program, also known as "Arduino", is used to program all of the Arduino microcontrollers. The Arduino software is free and based upon C programming language. Furthermore, Arduino users aren't limited by this software, they can use different alternative software to program the microcontrollers. There are some other companies using the Arduino open-source schematics to build their unique boards (either like the original or with modifications to enhance the functionality). As an example, the most famous board (Uno) has a large number of resemble boards from alternative suppliers that are different slightly from original Uno board in color, USB port, etc. In follows the most popular Arduino board existing till know [25].

2.2.3.1 Arduino Mini / Mini Lite

The tiniest Arduino unit is the Arduino Mini Light that is a 24-pin microcontroller with no connectors soldered. The product characteristics 8 analog pins and 14 digital pins. The unit is based on ATMEGA168 processor. Just one different in between Mini Light and Mini, that's the Mini comes with pre-soldered pin headers. Arduino Mini has been being developed and it will be including the new 32U4 processor. The Mini/Mini Lite are usually used with breadboards. For programming these microcontroller boards, there is a need of separated USB to serial adapter [25].



Figure 2.1: Arduino Mini/Mini Lite board [25]

The specification of Arduino Mini and Mini Lite are:

- ATmega328Microcontroller
- 5V Operating Voltage
- 7-9 V Input Voltage
- Digital I/O Pins 14 (6 provide PWM output)
- 8 Pins Analog Input
- 40 mA DC Current per I/O Pin
- 32 KB Flash Memory (2 KB used by bootloader)
- 2 KB SRAM
- 1KB EEPROM
- 16 MHz Clock Speed

2.2.3.2 Arduino Pro Mini 3.3V / Pro Mini 5V

The Arduino Pro is considered the one of smallest and fastest Arduino boards available till now. It is also breadboard mountable like as Arduino Mini but it a bit longer, and available in two types: 8-MHz and16-MHz. The 8-MHz Pro Mini operates at 3.3 volts, while the 16-Mhz Pro Mini operates at 5 volts. Both of has features of 14 digital and 6 analogs I/O [25].

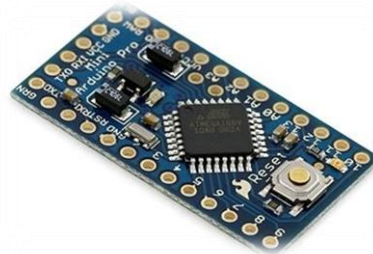


Figure 2.2: Arduino Pro Mini board [25]

The specification of Arduino Mini and Mini Lite are:

- ATmega328 Microcontroller
- It is 3.3V or 5V Operating Voltage (depending on model)
- Input Voltage 3.35-12 V for (3.3V model) or 5-12 V (5V model)
- 14 Pins Digital I/O (of which 6 provide PWM output)
- 6 Pins Analog Input
- 40 mA DC Current per I/O Pin
- 16 KB Flash Memory (2KB used by bootloader)

- 2 KB SRAM
- 1 KB EEPROM
- 8 MHz Clock Speed for (3.3V model) and 16 MHz for (5V model)

2.2.3.3 Arduino Nano / Nano Lite

Arduino Nano is the last Arduino mountable breadboard. This microcontroller different from another breadboard by having connector onboard and USB to serial chip. It has eight analog pins and fourteen digital pins. There are the ISCP headers to reflash the AT Mega chip. There is another type of Arduino Nano which is Arduino Nano Lite that doesn't have downward facing pin headers [25].



Figure 2.3: Arduino Nano / Nano Lite board [25]

The specification of Arduino Mini and Mini Lite are:

- Microcontroller Atmel ATmega328
- Operating Voltage (logic level) 5 V
- Input Voltage (recommended) 7-12 V
- Input Voltage (limits) 6-20 V
- Digital I/O Pins 14 (of which 6 provide PWM output)
- Analog Input Pins 8
- DC Current per I/O Pin 40 mA
- Flash Memory 32 KB (2KB used by bootloader)
- SRAM 2 KB
- EEPROM 1 KB

2.2.3.4 Arduino Fio

The Arduino Fio is a bit of a one-off board and is essentially an Arduino Mini with a built-in LiPo charger and XBee headers [25].



Figure 2.4: Arduino Fio board [25]

The specification of Arduino Mini and Mini Lite are:

- ATmega328P Microcontroller
- 3.3 V Operating Voltage
- 3.35-12 V Input Voltage
- 14 Pins Digital I/O
- 8 Pins Analog Input (10-bit resolution)
- 40 mA DC Current per I/O Pin
- 32 KB Flash Memory (2KB used by bootloader)
- 3.3 KB SRAM
- 1024 bytes EEPROM
- 8 MHz Clock Speed

2.2.3.5 Arduino LilyPad / Simple LilyPad

The LilyPad is different from all other Arduino microcontrollers due to its circular, pink PCB. The LilyPad had been intended to be used with clothing and fabric-related projects, however, users have discovered various other applications for it. LilyPad can be washed together with the clothing. The LilyPad needs as low as 2.7 volts to operate. There are several LilyPad shields such as sensors, buzzers, LEDs, etc.) in the same format that be able to connect by using conductive fabric [25].



Figure 2.5: Arduino LilyPad board [25]

The specification of Arduino Mini and Mini Lite are:

- ATmega168V or 328V Microcontroller
- 2.7-5.5 V Operating Voltage
- 2.7-5.5 V Input Voltage
- 14 Pins Digital I/O
- 6 Pins Analog Input
- 40 mA DC Current per I/O Pin
- 16 KB Flash Memory (2KB used by bootloader)
- 1 KB SRAM
- 512 bytes EEPROM
- 8 MHz Clock Speed

2.2.3.6 Arduino Leonardo

Arduino Leonardo has the typical Arduino board shape and cannot be attached to breadboards. Leonardo can be available without or with shield stacking headers. Currently, Arduino plans to use the 32U4 chip and reducing the price of Arduino boards. [25].



Figure 2.6: Arduino Leonardo board [25]

The specification of Arduino Mini and Mini Lite are:

- ATmega32U4Microcontroller (onboard USB Transceiver)
- 5 V Operating Voltage
- 2.7-5.5 V Input Voltage
- 14 Pins Digital I/O
- 12 Pins Analog Input
- 40 mA DC Current per I/O Pin
- 32 KB Flash Memory (2KB used by bootloader)
- 3.3 KB SRAM
- 1024 bytes EEPROM
- 16 MHz Clock Speed

2.2.3.7 Arduino Pro 3.3V / Pro 5V

Arduino Pro is the most similar Arduino board to the Leonardo. A number of the benefits to this board are lighter weight, smaller footprint, and the range of operating voltage (from 3.3 to 12V). Arduino Pro doesn't include pin headers and even though it's small size than other Arduino boards. Arduino Pro are compatible with most Arduino shields [25].



Figure 2.7: Arduino Pro 3.3V / Pro 5V board [25]

The specification of Arduino Mini and Mini Lite are:

- ATmega168Microcontroller
- 3.3V Operating Voltage
- 3.35 -12 V Input Voltage
- 14 Pins Digital I/O
- 6 Pins Analog Input
- 40 mA DC Current per I/O Pin

- 16 KB Flash Memory (2KB used by bootloader)
- 1 KB SRAM
- 512 bytes EEPROM
- 8 MHz Clock Speed

2.2.3.8 Arduino Diecimilla/Duemilanove/Uno

It's the most common Arduino microcontrollers board. Duemilanove same as the Diecimilla but have a less powerful ATmega168 chip. Uno has nearly the same as Duemilanove, but Uno uses ATmega8 for USB to serial conversion. These boards come pre-assembled and ready to use. The Duemilanove is based around the ATMEGA328 chip while the Diecimilla used the ATMEGA128. Uno board can feed with power by any one of three ways: power connector, USB port, or the Vin and ground pins. ATMEGA chip has been detachable from the board. This is mainly useful for those who have fried the processor and want to replace it, in addition, the board can be used alone as a USB to serial interface. In the Uno R3 board, there are two new pins over the digital side: SCL and SDA [25].

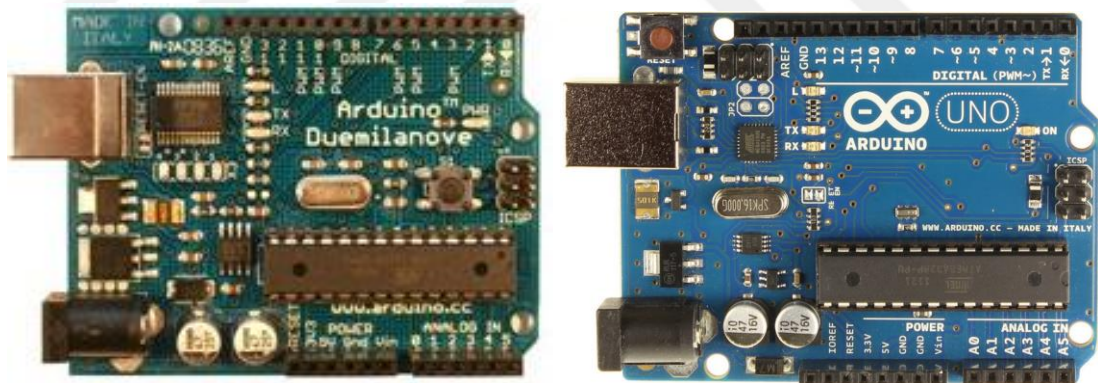


Figure 2.8: Arduino Duemilanove board (left), Arduino Uno (right) [25]

The specification of Arduino Mini and Mini Lite are:

- ATmega168 or 328 Microcontroller
- 5V Operating Voltage
- 7-12V Input Voltage (recommended)
- 6-20V Input Voltage (limits)
- 14 Pins Digital I/O
- 6 Pins Analog Input

- 40 mA DC Current per I/O Pin
- 50 mA DC Current for 3.3V Pin
- 16 KB Flash Memory for (ATmega168) or 32 KB Flash Memory for (ATmega328) (2KB used by bootloader).

2.2.3.9 Arduino Ethernet / Ethernet PoE

Arduino Ethernet is basically a standard Arduino Uno in which the ATmega8 chip and USB plug are replaced for an Ethernet port. The (power over Ethernet (PoE) model means that there are no needs to a separate power source, even though, the router should also be PoE compatible. The same setup can be achieved employing a normal shield-compatible Arduino with an Ethernet shield [25].



Figure 2.9: Arduino Ethernet / Ethernet PoE board [25]

The specification of Arduino Mini and Mini Lite are:

- ATmega328 Microcontroller
- 5 V Operating Voltage
- 7-12 V Input Voltage (36 to 57 volt PoE)
- 10* Pins Digital I/O
- 6 Pins Analog Input (10-bit resolution)
- 40 mA DC Current per I/O Pin
- 32 KB Flash Memory (2KB used by bootloader)
- 3.3 KB SRAM
- 1024 bytes EEPROM
- 16 MHz Clock Speed

2.2.3.10 Arduino Bluetooth

The board layout of Arduino Bluetooth is similar for that of Duemilanove, but it has just one big difference, that the Arduino Bluetooth board Bluetooth module instead of USB plug, which means a person has the ability to program it remotely. The Arduino Bluetooth board comes with different power specifications compared to Duemilanove and does not have a 3.3 V output pin. The 9-volt output pin pointed out board is not really functional [25].



Figure 2.10: Arduino Bluetooth board [25]

The specification of Arduino Mini and Mini Lite are:

- ATmega328Microcontroller
- 5V Operating Voltage
- 1.2-5.5 V Input Voltage
- 14 Pins Digital I/O
- 8 Pins Analog Input
- 40 mA DC Current per I/O Pin
- 16 KB Flash Memory (2KB used by bootloader)
- 2 KB SRAM
- 51 KB EEPROM
- 16 MHz Clock Speed

2.2.3.11 Arduino Mega 1280 / 2560

Arduino MEGA is the modern Arduino boards. It is physically bigger than all of the other boards and provides a lot more analog and digital pins. The MEGA utilizes a unique processor making it possible for larger program size [25].



Figure 2.11: Arduino Mega 1280 / 2560 board [25]

The specification of Arduino Mini and Mini Lite are:

- ATmega1280 or 2560 Microcontroller
- 5V Operating Voltage
- 7-12V Input Voltage (recommended)
- 6-20V Input Voltage (limits)
- 54 Pins Digital I/O (14 provide PWM output)
- 16 Pins Analog Input
- 40 mA DC Current per I/O Pin
- 50 mA DC Current for 3.3V Pin
- 128 KB or 256KB Flash Memory
- 8 KB SRAM
- 4 KB EEPROM
- 16 MHz Clock Speed

2.2.3.12 Arduino Mega ADK

It's same as Arduino Mega but the Arduino ADK is designed to communicate with Android OS based devices. [25].



Figure 2.12: Arduino Mega ADK 1280 / 2560 board [25]

The specification of Arduino Mini and Mini Lite are:

- ATmega1280 or 2560 Microcontroller
- 5V Operating Voltage
- 7-12V Input Voltage (recommended)
- 6-20V Input Voltage (limits)
- 54 Pins Digital I/O (of which 14 provide PWM output)
- 16 Pins Analog Input
- 40 mA DC Current per I/O Pin
- 50 mA DC Current for 3.3V Pin
- 128 KB or 256KB Flash Memory
- 8 KB SRAM
- 4 KB EEPROM
- 16 MHz Clock Speed

2.2.4 Electrical Sensors

There are wide sensors that can serve in electric switches such as current sensors, voltage sensors light sensors, etc. In follows the description of some selected sensors that is useful for smart switch system.

2.2.4.1 Current and Voltage Sensors

A current sensor is an instrument that detects and converts current to a simply recorded output voltage that is proportional to the current throughout the measuring path. As a current move via a wire or in a circuit, the voltage drop will be occurring. In addition, a magnetic field is produced around the current carrying conductor. These two phenomena are intended to use of in the design of the current sensor. Therefore, there are two different types of current sensing: indirect and direct. Indirect sensing is dependent on Ampere's and Faraday's law, while direct sensing is dependent on Ohm's law. Direct Sensing requires calculating the voltage drop belonging to the current transferring through passive electrical hardware. Indirect Sensing requires measuring the magnetic field surrounding the conductor during which current passes. This magnetic field that generated will be utilized to produce proportional current or voltage which can be converted to a form appropriate for measuring and/or control system. Voltage sensor, on another hand, has just one sensing mode which is a direct sensor that used to measuring voltage variation of the circuit [27, 28]. To measure AC current by using Arduino microcontroller, there is a large range of the current sensor in both indirect and direct types. Figure 2.13 shows some of the current sensors types. For AC voltage measurement, there is no AC voltage sensor available in the market so it should be designed in order to get real reading which is so important in case of smart switch to allow real power reading which led to calculate close power cost in addition it can serve for devices safety via cut off power when voltage power up or drop down the allowable range.

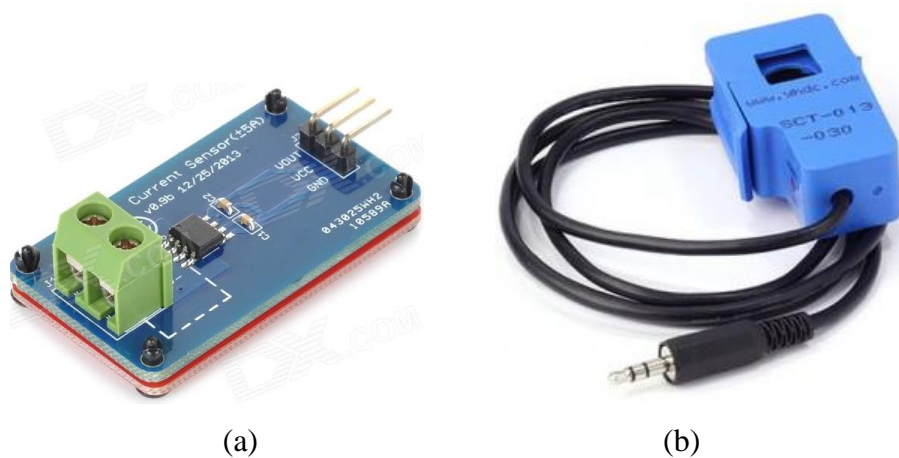


Figure 2.13: Arduino current sensors. (a) direct current sensor, (b) indirect current sensor [29]

2.2.4.2 Light Sensors

Light sensor, is a unit which is used to recognize light. There are various different types of light sensors, each has been worked in a slightly different way. A photo resistor or photocell as an example is a compact sensor that varies its resistance as light shines over it; they are utilized in several consumer systems to determine the light intensity. The second type is the charged coupled device (CCD) which transfers electrically charged signals, which is utilized as a light sensor in night-vision and digital cameras devices. Many present-day electronic devices, like wireless phones, televisions, and computers, make use of ambient light sensors to robotically regulate the brightness of a screen, especially in highlight or low-light or situations, also the most important application of light sensors now days is to controlling the buildings light in order to save energy. The Light sensor output can be analog or digital as needed. Figure 2.14 shows a sample of Arduino light sensor [30].

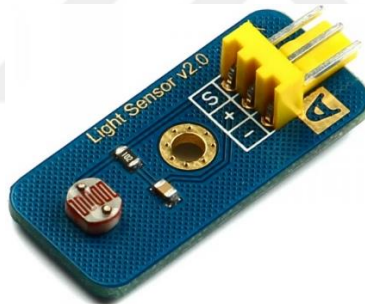


Figure 2.14: Arduino light sensor [30]

2.2.4.3 Infrared Sensors

Infrared (IR) sensor is utilized to recognize the motion. The passive infrared (PIR) sensor indicates IR light released from objects that produce heat and as a result IR radiation in their field of view. As motion is detected, PIR sensor produces a high signal on its output pin, which could be either read by an MCU or drive a transistor to switch a greater current load [31].

The advantage of utilizing IR with a smart switch that it could employ to recognize motion in places and switch the light off or on automatically based upon human activity. Figure 2.15 shows a sample of Arduino PIR sensor [32, 33].



Figure 2.15: Arduino light sensor [34]

2.2.4.4 Relay Module

The relay unit was utilized to control the “ON” or “OFF” applications. This relay unit used 5V to operate and previously supplied by the microcontroller. The relay unit had 1 pole and 6 pins. There are various relays in the market and it depends upon current needs. The transistor is utilized to control turn “ON” and “OFF” the relay. The diode is set up in reverse direction across the relay to guard it against a voltage spike once the transistor is switched “OFF”. Figure 2.15 shows a sample of Arduino relay module [35].

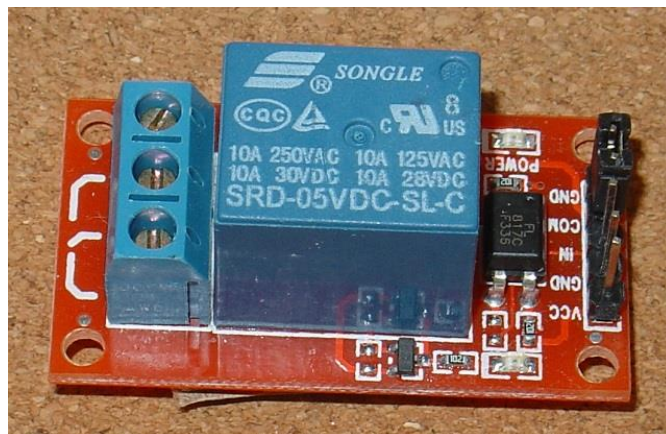


Figure 2.16: Arduino relay module [36]

2.3 Smart Switch Software Component

The smart switch requires designing programs on both sides: one to programming hardware operation and second to create a graphical user interface (GUI) that provide user-friendly dialog allowing him to control and monitor smart switch from every were [1].

2.3.1 Hardware Programming

The hardware is required to be programmed by using certain software in order to make embedded system work. There are many computer languages that can be utilized for hardware programming like Python, C++, etc. as well as a number of a unique software platform that provided for many microcontrollers like Arduino software that dependent on C++ language and provides an easy solution to programming Arduino board in order to perform wide range application. In this study, we have been used Arduino software that has been helped in programming microcontrollers in order to manage smart switch by the simple and effective way [1].

A program is used for Arduino microcontroller can be written in any programming language for any compiler that creates binary machine code for that target processor. Atmel offers a development environment that is AVR Studio and the latest Atmel Studio with regard to its microcontrollers. Arduino project offers the Arduino integrated development environment (IDE), which is actually a cross-platform application designed by Java programming language. It came from the IDE for the languages wiring and processing. It contains an editor for code with some options like syntax highlighting, text cutting and pasting, brace matching searching and replacing text, and, in addition to offering easy one-click mechanisms to upload and compile programs to an Arduino board. Furthermore, it has a text console, automatic indenting, a message area, hierarchy of operation menus and a toolbar with buttons for common functions. The program that is written using the Arduino IDE is termed a sketch, which can be saved on the development computer in form of text files using the file extension “.ino”. The Arduino Software (IDE) pre-1.0 stored sketches using the extension “.pde” [37].

The IDE of Arduino support C and C++ languages by using specific code structuring rules. The IDE of Arduino offers a programs library wiring project, which gives many common input/output procedures. The User who has written code just needs two basic functions in order to start the sketch as well as the main program loop, which is compiled and linked to a program stub main () towards an executable cyclic executive program using the GNU toolchain, as well integrated with the IDE distribution. The IDE of Arduino is using the program air dude to transform the

executable code towards a hexadecimal encoding text file which can be loaded to the Arduino board by a loader program in the firmware of the board's. Figure 2.17 shows the Arduino IDE Program [38].

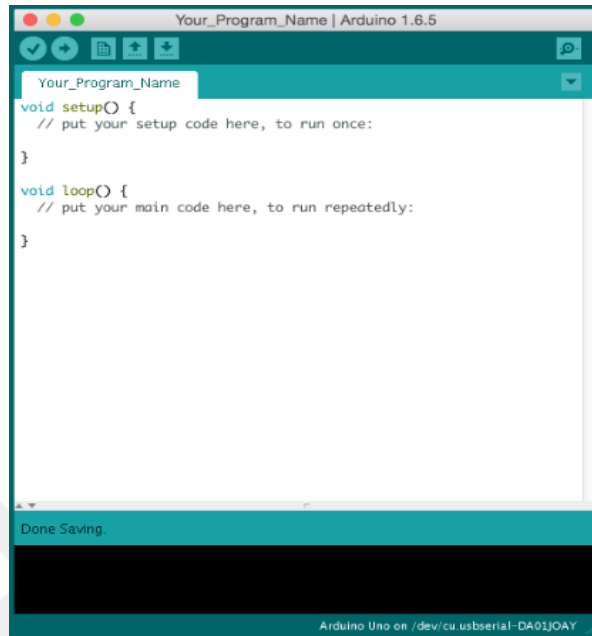


Figure 2.17: Arduino IDE Program [38]

2.3.2 User Interface Programming

The aim of user interface (UI) programming is to build an interactive dialog which allows people to monitor smart switch including current, voltage, power in addition to managing it like switch on or off the devices or arranged some process via date and time, etc. [1].

The UI software that needs to design to be dependent on device type and OS operating system have used. For smartphone device, there's wide free platform which could assist for design a UI for example "Mit inverter" software which can be use to develop Android mobile application in order to control Arduino boards, which is able to design an easy UI for smart switch application to allow users to read the AC Current, AC Voltage, Aparent power in addition to control switch on/off devices. Figure 2.17 shows the GUI of Mit inverter software [39].



Figure 2.18: Mit inventor software [38]

CHAPTER 3

TECHNIQUES FOR REACTIVE POWER AND VOLTAGE CONTROL

3.1 Introduction

This chapter shows the practical side of this work aiming to test the performance of smart switch reading system in a real work circumstances. In addition, the practical tests are used to show the smart switch abilities in controlling the switch via a network and worldwide internet.

3.2 System Design

The smart switch is designed to be installed at the house electrical entry point. Each of the smart switch outlet controls a desired room. The switch is connected to a Wi-Fi Router via Ethernet then to the internet in order to manage it remotely and to allow users to control and monitor the electrical outlets and smart plugs. The user can interact with the switch via Android devices, iPhone, PC, etc. to control loads remotely. The smart switch communication architecture is illustrated in Figure 3.1.

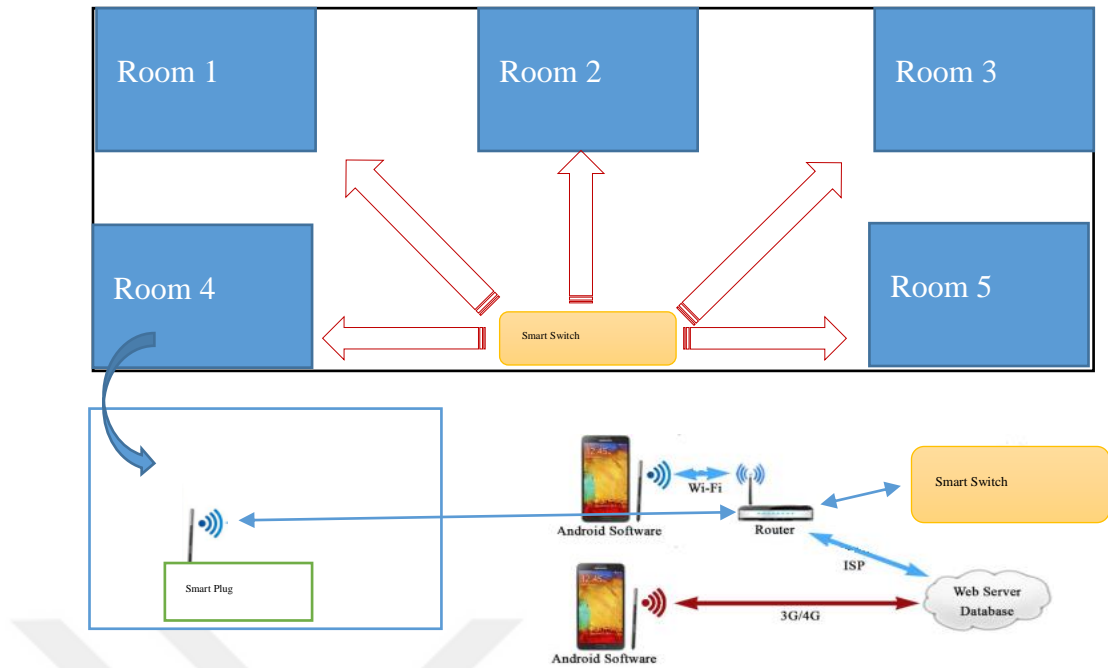


Figure 3.1: Architecture of smart switch communication

This system is designed and implemented by two parts: hardware design and software design.

3.2.1 Hardware Design

The hardware circuit is designed to achieve a good accuracy, best functional operation with reliability and safety. The hardware includes the following components:

1. Microcontroller: The system uses Arduino Mega 2560 in order to control plug switches and to receive the signal from AC voltage and a current sensor. For monitoring, a current, voltage and energy provide real time update of the energy consumption at the device level. The microcontroller Arduino Mega 2560 executes the received user commands. It has Internet connectivity through an Ethernet shield mounted on it. The user side can use a mobile device provides an interface with the smart switch as a whole through a user-friendly application. The mobile device can be either wired to the central controller through USB cable for instance or communicates with it wirelessly.



Figure 3.2: Arduino Mega 2560

- Sensors:** The system uses AC current sensor 30A SCT013 and shield card designed to increase the sensitivity of the readings. Because there is no AC voltage sensor for Arduino microcontroller and to calculate exact power consumption values, the work designed a prototype of AC sensor that can measure AC electric voltage with high accuracy. The AC voltage sensor is prototype voltage sensor which designed with the transformer, full wave bridge, polar capacitor, and variable 10K ohm resistor. This circuit includes AC voltage sensor that converts the 220V AC to 3-volt DC. Any change in the AC voltage leads to a change in the DC voltage. The microcontroller does not read AC voltage, therefore, the AC voltage value is converted into DC value. The Arduino's microcontroller reads the analog DC voltage between 0V to 5V. Figure 3.3 shows the circuit diagram of a prototype AC voltage sensor.

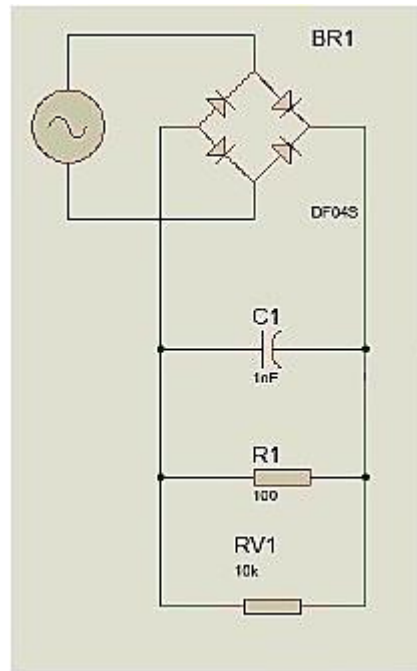


Figure 3.3: Circuit diagram of a prototype voltage sensor

The microcontroller calculates AC voltage based on multiplying the real time DC voltage by a converting factor as in following formula:

$$V_{AC} = V_{dc} * f \quad (1)$$

where V_{ac} is the AC voltage, V_{dc} is the DC voltage from AC voltage sensor that varies from 0 to 3 volt, f is converting factor which is 75.

3. Ethernet shield: The system uses this part to connect the microcontroller to the internet. The Ethernet module acts as a bridge to connect the switch Gateway to the local proxy.



Figure 3.4: Ethernet shield

4. **Relays card:** The system uses eight relays operate at 5 volts DC signal to derive a 220 VAC, 10A, to control eight devices in the house. The hardware of the smart switch is shown in Figure 3.5.

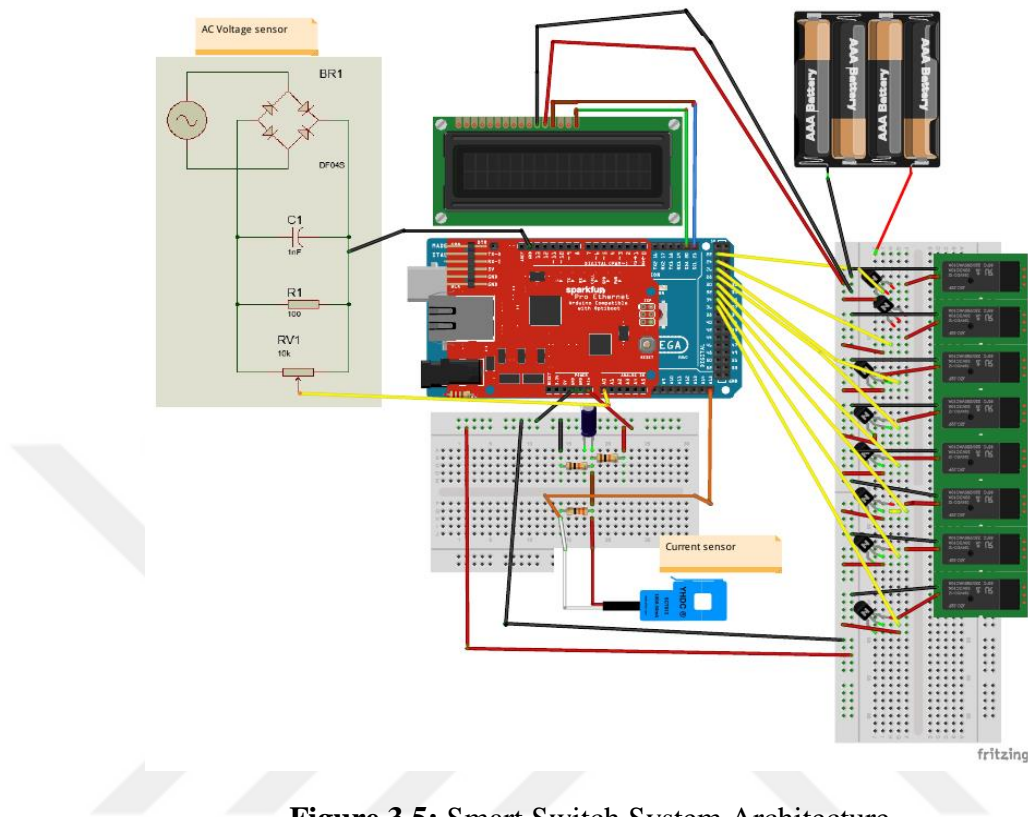


Figure 3.5: Smart Switch System Architecture

In addition, the previous main components, there are other components such as: DC power supply, that supplies smart switch system with 5 volt DC power that is important to operate microcontroller and other electronic components, a wifi router that is connected to Arduino Mega 2560 via Ethernet shield which used to implement the micro Web-server for the switch gateway that gives ability to the user to control switches by connected to wifi router directly or via internet. The smart switch component is put inside an electric plastic box that gives safety and protection to user from contact to the circuit.

3.2.2 Software Design

The Android application is used to allow human to interact with the smart switch device. The application can monitor and control the smart switch device. Mit inverter2 software is used to design an Android mobile application to read the power

consumption, AC Voltage, AC Current, and also control eight devices by turning them on & off individually. Figure 3.6 shows the application layers for design application software and the Android platform structure.

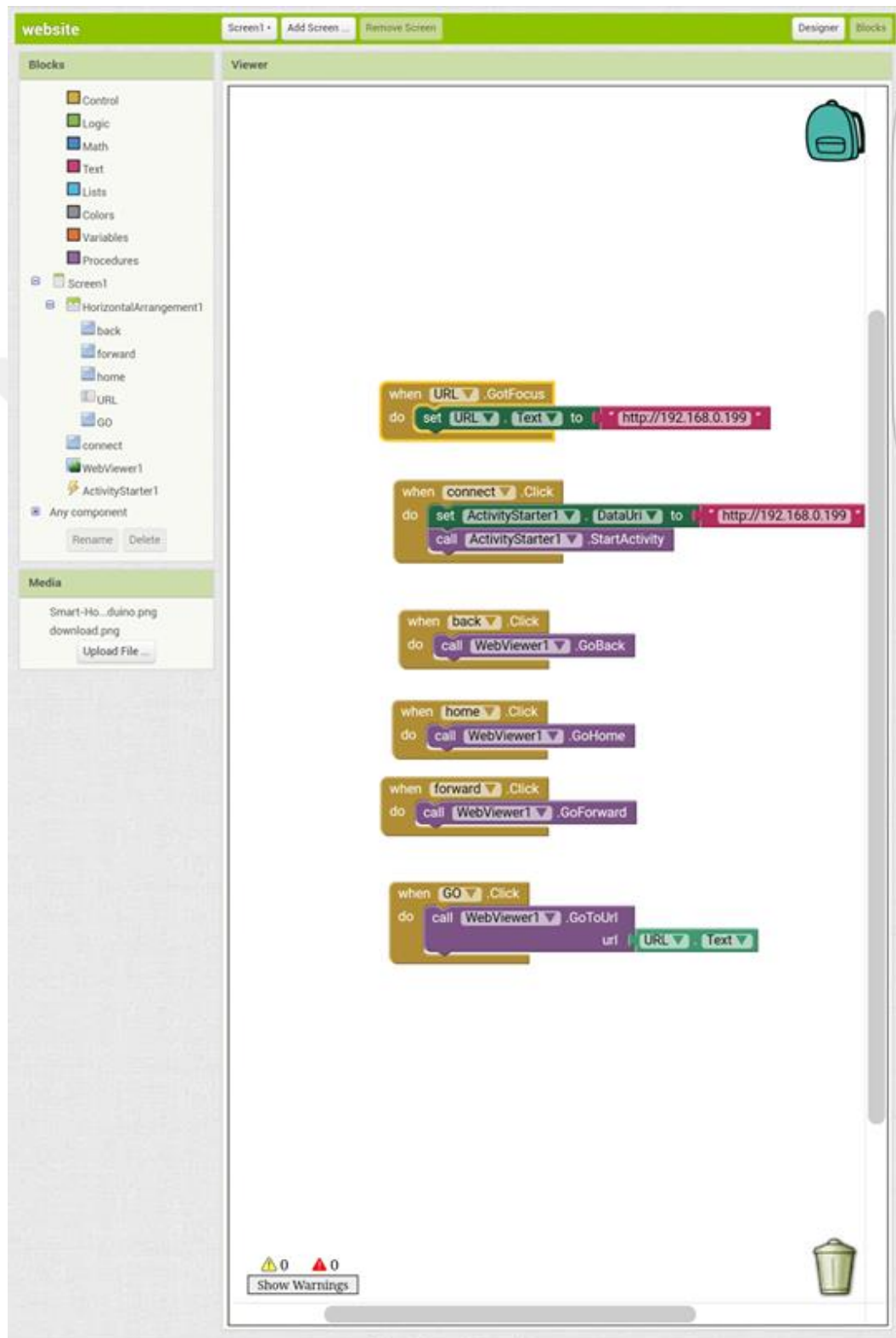


Figure 3.6: Smart Switch Android platform structure

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Introduction

This chapter presents the experimental results, which shows a practical side of the work. First, the system performance is tested by using multiple loads in order to check the smart switch reading in real work and compared it with authorized power measuring instrument. In addition, it is also tested to check system stability at a time. Second, the proposed system is analyzed for some important factors of the reliability, ease of installation, communication, and cost.

4.2 System Performance Results

To investigate the operation of the smart switch in real operation, the smart switch is tested by connecting it to a building AC power source. In addition, the Ethernet shield is connected to the Wi-Fi router via UTP cable. The router is connected to the internet to allow users to connect the smart switch via the internet.

4.2.1 Test the reading accuracy

In this part, the work tests the accuracy of smart switch current, voltage and power readings. This part is important to investigate the stability of the system with different loads and through a period of time that gives an idea about the accuracy in calculating power consumption as well as the cost. For this purpose, the UT 202 clamp meter is used to take a real reading of voltage and current. These readings are then compared with the smart switch readings. Figure 4.1 shows the smart switch test setup.

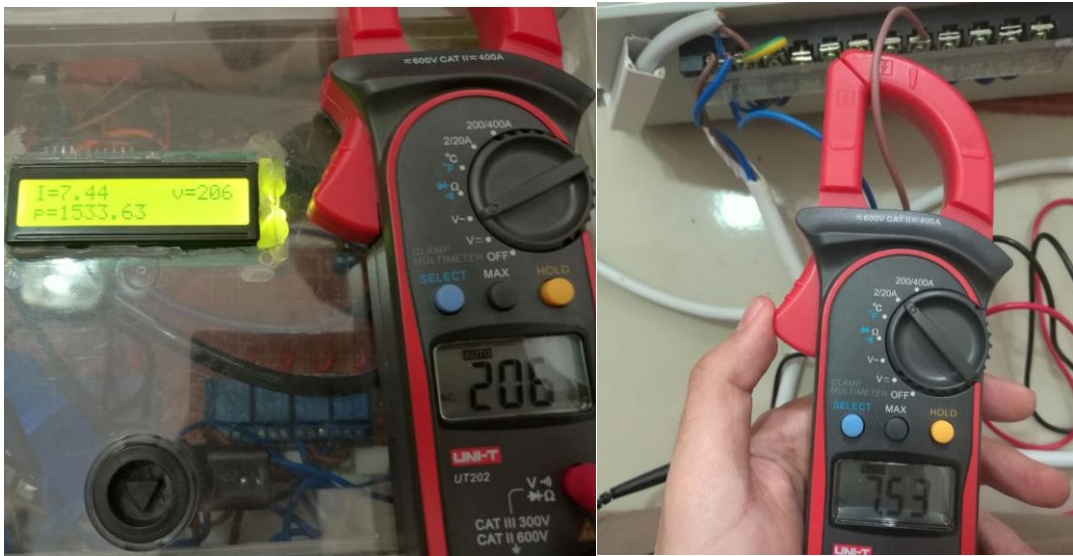


Figure 4.1: Smart switch test setup

As described before, different loads as shown in Table 4.1 is used in order to test the reading accuracy of the smart switch. First, the loads are connected to the AC terminal of the smart switch. The clamp meter is also connected to the same loads by putting its clamps around the loads' wires to get the drawn current readings. The clamp meter probe is used to record the AC input voltage readings. As shown in Table 4.2, the readings are recorded for both smart switch and clamp meter at the same time.

Table 4.1 Smart switch reading for multiple devices

Devices	Current (A)	Voltage(V)	Power (W)
Television	0.35	215	75.430
Vacuum Cleaner	2.53	216	546.01
Iron	7.51	207	1555.20
Computer	0.31	213	66.80
Hair iron	0.27	216	58.80
Hair dryer	3.95	212	836.59

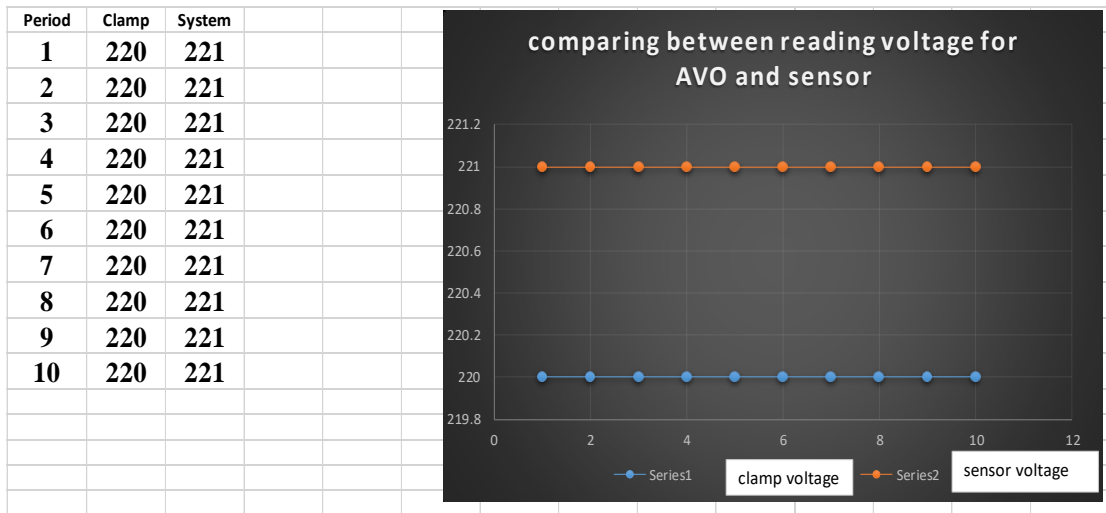
Table 4.2: Clamp meter reading for multiple devices

Devices	Current (A)	Voltage(V)
Television	0.30	216
Vacuum Cleaner	2.51	217
Iron	7.53	208
Computer	0.37	214
Hair iron	0.30	216
Hair dryer	3.97	212

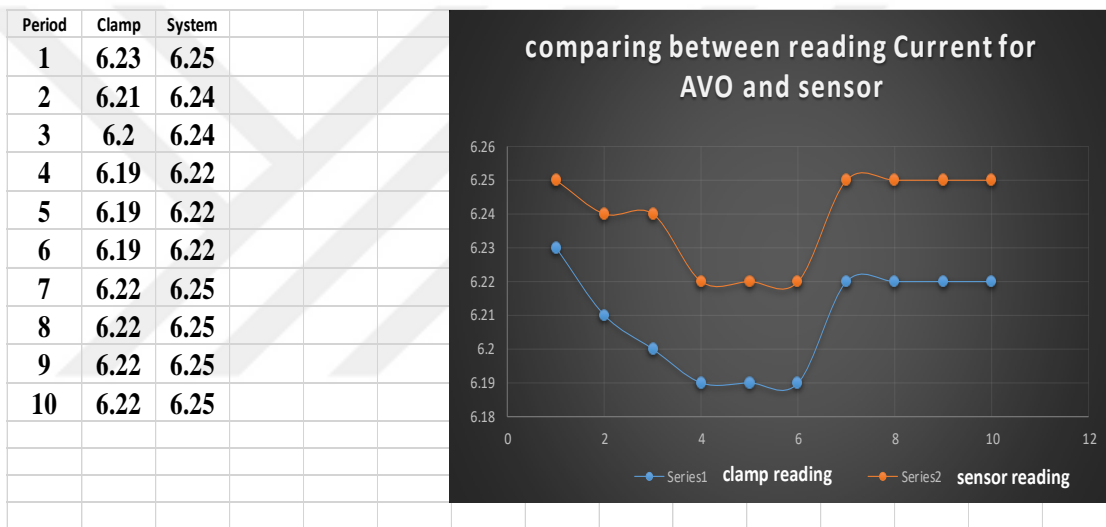
A comparison is made between the results obtained from the smart switch reading and clamp meter reading. It shows that the smart switch readings are close to that obtained from the clamp meter and percentage error does not exceed 0.5% for the current and 1% for the voltage. which indicates that the smart switch reading system has a high accurate reading capabilities.

4.2.2 Stability Results

In order to test the stability of the smart switch reading for real operation through a period of time, the system is connected to a building outlet and then connect the different load to it. Then, loads are operated in a period of 50 minutes. The currents and voltages are recorded every 5 minutes in 10 periods. These reading are recorded from both the smart switch and a clamp meter. Figures 4.2, 4.3, and 4.4 show the results of the recorded readings versus time for both the smart switch and the clamp meter reading.

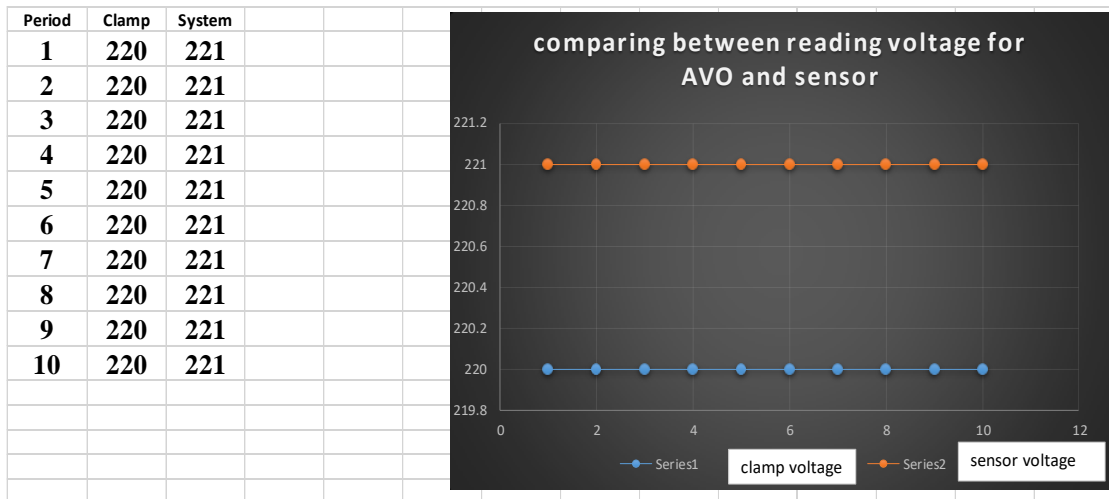


(a)

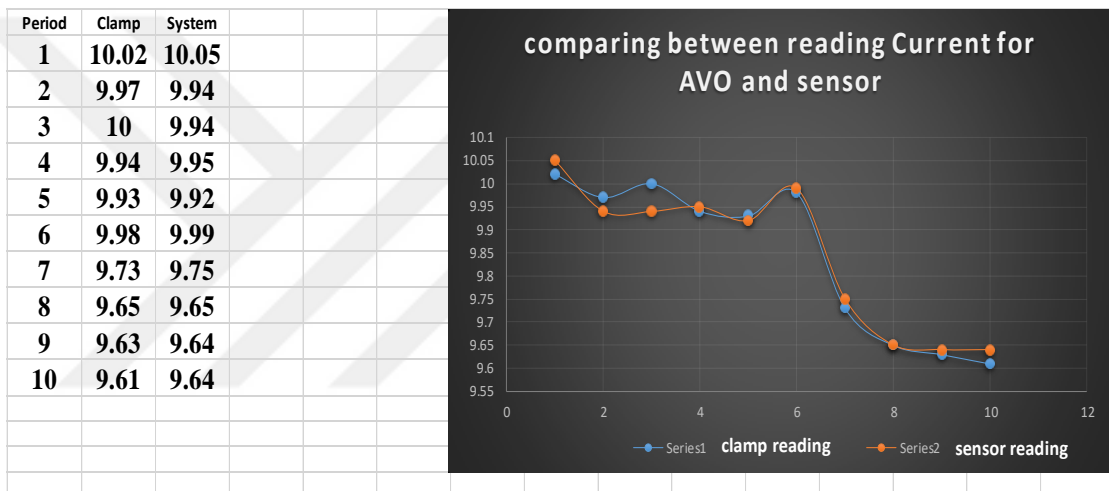


(b)

Figure 4.2: Iron reading records and its chart for (a): voltage, (b) Current

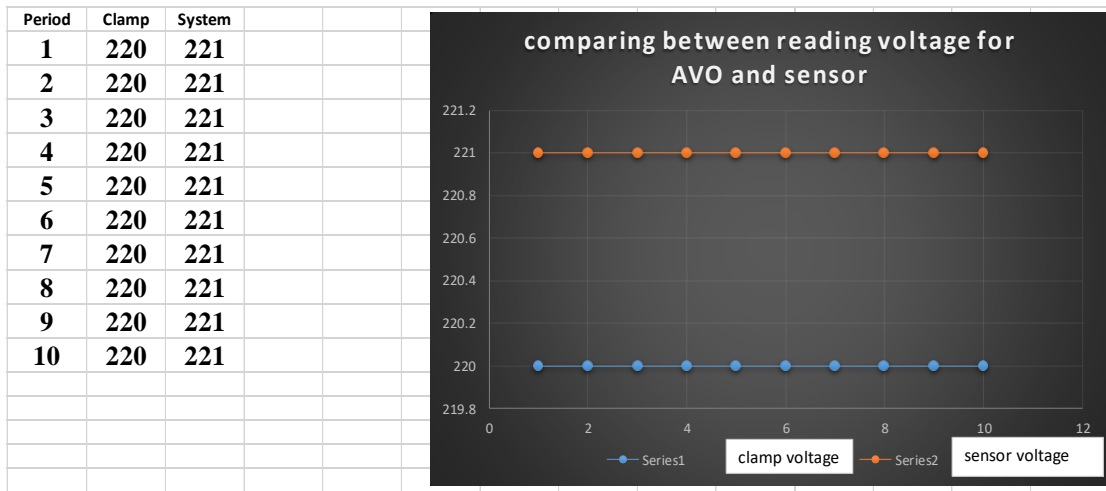


(a)

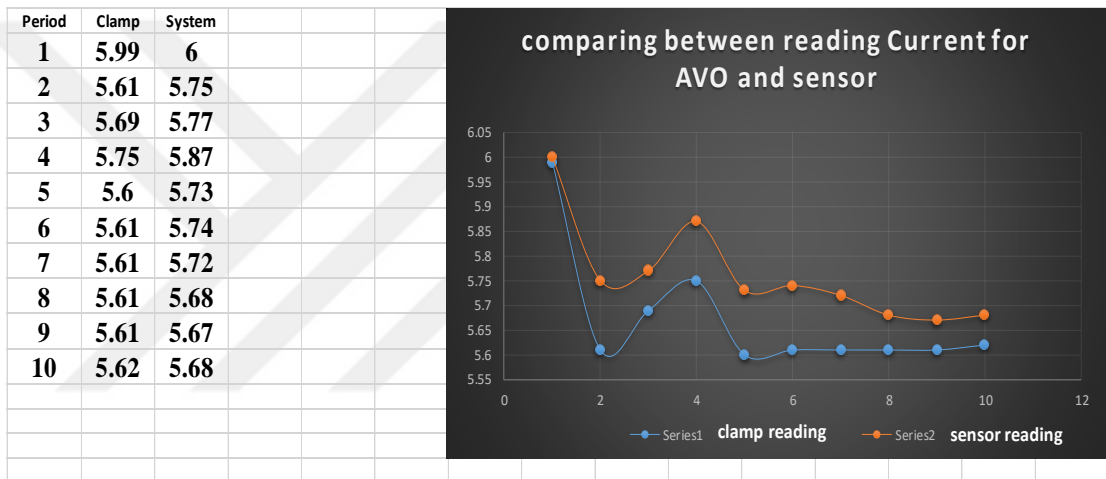


(b)

Figure 4.3: Oven reading records and its chart for (a): voltage, (b) Current



(a)



(b)

Figure 4.4: Microwave reading records and its chart for (a): voltage, (b) Current

As shown in Figures 4.2 (a), 4.3(a), and 4.4(a), the voltage readings are very stable, where the difference between the readings of clamp meter and smart switch readings with time are low and it is about +1 volt. For current test shown in Figures 4.2 (b), 4.3(b), and 4.4(b), it is appeared that the current readings changed with time according to the absorbed power of the devices. The readings of both clamp meter and smart switch are very close to each other. These results prove that the smart switch can be considered as a dependent system for power management and power cost calculations.

4.3 System Analyses

4.3.1 Reliability

For the proposed smart switch, the real-time analysis shows that the system is reliable for doing power management tasks without any problem. The tests of the system are done by connecting some electric devices to the smart switch for everyday jobs. The results show that all system components worked efficiently with no recorded problem. Therefore, the smart switch can be considered as an authorized system for working in real life tasks.

4.3.2 Ease of Installation and management

The system is very easy to install and it does not need any experts. It can be installed by plugging the input power to the feeder and connect power outlet to the desired devices. For user interface management page, it is very simple to run on the smartphone devices and it is easy to understand. It includes a switching buttons and power reading monitors, so the overall system can be considered as an easy system in both installation and management.

4.3.3 Communication

In order to test the communication between the smart switch and the user, the smart switch is tested in both the system installation and system control. For system installation, the smart switch is connected to a building and setup its IP address, which is easy steps that any user can do. The users can communicate with the smart switch easily through the internet by just installing a router in the building and then they can use the system interface program to monitor and control overall process.

4.3.4 Cost

The cost of the smart switch system is very important to the customers. The system is low cost because of it is made from low cost components such as Arduino, electric boards, transformers, power supply, an electronic components which costs less than 100\$ in overall, so it can be considered as a frugal system.

CHAPTER 5

CONCLUSION, SUGGESTION FOR FUTURE WORK

5.1. CONCLUSION

The benefit from using a smart switch is to allow users to fully control and monitor the electrical power of their buildings by controlling the smart switches' outlets. The smart switch offered a graphical user interface that recently most of the people used it and helped them to be familiar with the device. The switch provides scheduling options that can determine the operation time of each electric terminal. The switch can be accessed by numerous platforms including phones, PC desktops, tablets, etc. which permits the user to modify any event related to the switch outlet. The Smart switch can calculate the real time voltage and current as well as the power consumptions. Therefore, it can work as a power meter with a benefit of calculating the electricity costs too. The test results show the smart switch working efficiently and the user can control every output terminal easily. The system monitoring gives high accuracy readings for the voltage, current, and the power consumption, so it can be considered as a trustworthy system. This smart switch can be integrated with more sensors to offer a lifestyle requirement such as turn lights on/off depending on light conditions, controlling air conditioner depending on the environment temperature, switch devices like light, TV's depending on human activity and other lifestyle features.

5.2. SUGGESTION FOR FUTURE WORK

The suggestions can be summarized as following;

1. Adding another sensor such as light sensors, IR sensors in order to improve the power management system.
2. Improving the GUI system to be more stylish and user-friendly and including all needed features, and to be supporting all common OS of smartphone and PC.
3. Improving the system to work with high power consumption loads and connecting it with National Electricity Network to be authorized smart switch with integrating meter.



REFERENCES

- [1] Liu, Y., (2014). Design and implementation of the ELIQ Smart-plug system device, Master's Thesis in the Master Degree Programme Embedded Electronic System Design, Chalmers University of Technology, Sweden.
- [2] Singh, R. S. S., Ibrahim, A. F. T., Salim, S. I. M., Chiew, W. Y., (2009). Door Sensors for Automatic Light Switching System, Third UKSim Eur. Symp. Comput. Model. Simul., pp. 574–578,
- [3] Chobot, E., Newby, D; Chandler, R., Abu-Mulaweh, N., Chen, C., Pomalaza-Ráez, C., (2013), Design and Implementation of A Wireless Sensor and Actuator Network for Energy Measurement and Control at Home, *International Journal of Embedded Systems and Applications* (IJESA) Vol.3, No.1, March.
- [4] PARAB, J. S., SHELAKE, V. G., KAMAT, R. K.; NAIK, G. M., (2007). Exploring C for Microcontrollers: A Hands-on Approach, Book, Springer, ISBN 978-1-4020-6067-0.
- [5] Mowad, M. L., Fathy, A., Hafez, A., (2014). Smart Home Automated Control System Using Android Application and Microcontroller, *International Journal of Scientific & Engineering Research*, Vol 5, Issue 5, pp.935-939, May.
- [6] Boulos, M. N., Steve, W., Tavares, C., Jones, R., (2011). How smartphones are changing the face of mobile and participatory healthcare: an overview, with an example from eCAALYX, *Biomed Eng Online.*; 10: 24, doi: 10.1186/1475-925X-10-24, Apr.
- [7] C.-H. Lien., C.-H. Lin, Y.-W. Bai, M.-F. Liu, and M.-B. Lin., (2006). Remotely controllable outlet system for home power management, in Proceedings of the *IEEE 10th International Symposium on Consumer Electronics* (ISCE '06), pp. 1–6, Saint Petersburg, Russia, July.
- [8] C.-H. Lien, Y.-W. Bai, and M.-B. Lin., (2007). Remote-controllable power outlet system for home power management, *IEEE Transactions on Consumer Electronics*, vol. 53, no. 4, pp. 1634– 1641.

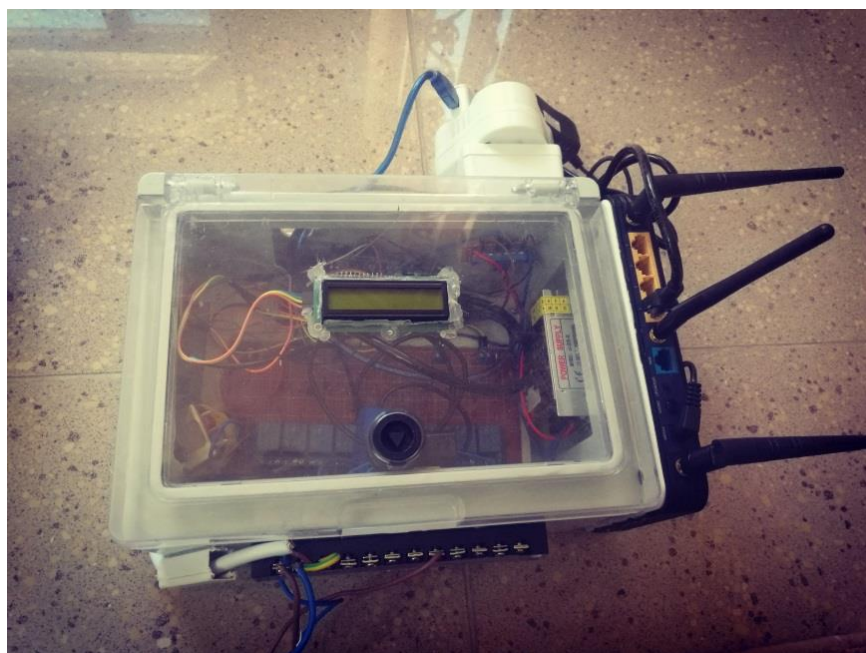
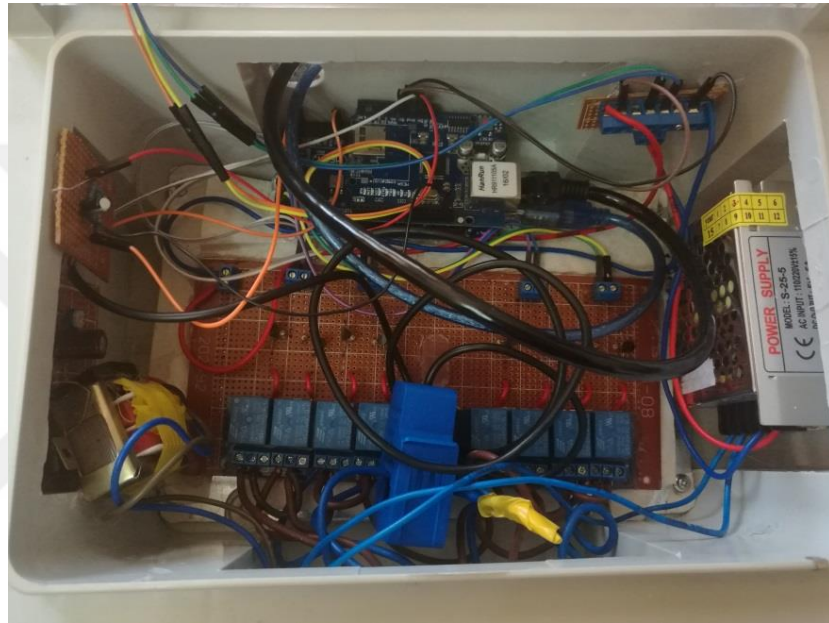
- [9] C.-H. Lien, H.-C. Chen, Y.-W. Bai, and M.-B. Lin., (2008). Power monitoring and control for electric home appliances based on power line communication,” in Proceedings of the *IEEE International Instrumentation and Measurement Technology Conference (IMTC '08)*, pp. 2179–2184, *IEEE, Vancouver, Canada*, May.
- [10] C.-H. Lien, J.-C. Lin, and Y.-W. Bai., (2008). design and implementation of a remotely controllable UPS outlet system, in Proceedings of the 12th *IEEE International Symposium on Consumer Electronics (ISCE '08)*, pp. 1–4, *IEEE, Vilamoura, Portugal*, April.
- [11] C. H. Lien, Y. W. Bai, H. C. Chen, and C. H. Hung., (2009). Home appliance energy monitoring and control based on power line communication, in Proceedings of the *IEEE International Symposium on Consumer Electronics (ISCE '09)*, *IEEE Consumer Electronics Society*, Las Vegas, Nev, USA, January.
- [12] M. Lee, Y. Uhm, Y. Kim, G. Kim, and S. Park., (2009). Intelligent power management device with middleware based living pattern learning for power reduction, *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 2081–2089.
- [13] Y.-W. Bai and C.-H. Hung., (2008). Remote power on/off control and current measurement for home electric outlets based on a low-power embedded board and ZigBee communication, in *Proceedings of the 12th IEEE International Symposium on Consumer Electronics (ISCE '08)*, pp. 1–4, Vilamoura, Portugal, April.
- [14] A. L. Lentine, J. R. Ford, J. R. Finn et al., (2012). An intelligent electrical outlet for autonomous load control for electric power grids with a large percentage of renewable resources, in *Proceedings of the IEEE Power and Energy Society General Meeting (PES '12)*, pp. 1–8, *IEEE Power & Energy Society*, San Diego, Calif, USA, July.
- [15] Shajahan, A. H.; Anand A., (2013). Data acquisition and control using Arduino-Android Platform: Smart Plug, *IEEE, Madras Institute of Technology Campus*, India, ISBN:978-1-4673-6150-7/13.
- [16] G. Horvat, D. Vinko, and D. Zagar., (2013). Household power outlet overload protection and monitoring using cost effective embedded solution, in *Proceedings of*

- the IEEE Mediterranean Conference on Embedded Computing (MECO '13)*, Budva, Montenegro.
- [17] Yan, M., Shi, H., (2013). Smart Living Using Bluetooth based Android Smartphone, *International Journal of Wireless & Mobile Networks (IJWMN)*, Vol. **5**, No. **1**, February.
- [18] Calcutt, D., Cowan, F. Parchizadeh, H., (2003). 8051 Microcontroller: An Applications Based Introduction, Book, Newnes, ISBN: 9780080470221, Dec.
- [19] Gowthaman, A., Mohanraj, R., Anadhan, A., Mohan, S., (2016). Implementation of School Children Tracking System And Transportation Safety Enhancement By Using Rfid Technique, *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)*, ISSN: 0976-1353 Vol.**21** Issue 2, Apr.
- [20] Khadse, R., Gawai, N., Faruk, B. M., (2014) Overview and Comparative Study of Different Microcontrollers, *International Journal of Research in Applied Science & Engineering Technology (IJRASET)*, Volume **2** Issue XII, ISSN: 2321-9653, December.
- [21] Denton, T., (2013). Automobile Electrical and Electronic Systems, Book, 4th edition, Routledge, ISBN: 9781136073816, 2013.
- [22] Gridling, G., Weiss, B., (2007). Introduction to Microcontrollers”, Courses 182.064 & 182.074, Embedded Computing Systems Group, *Institute of Computer Engineering, Vienna University of Technology*, Feb.
- [23] Shibu, K. V., (2009). Intro To Embedded Systems, Book, 1st, Tata McGraw-Hill Education, ISBN:9781259081514.
- [24] ARM Developer., (2015). ELF for the ARM Architecture, ARM IHI 0044F, Current Through ABI Release 2.10, ARM Limited, Nov.
- [25] Hughes, J. M., (2016). Arduino: A Technical Reference, Book, O'Reilly Media, Inc., ISBN: 9781491934319, May.
- [26] Zhen, Y., (2011) Current Sensing Circuit Concepts and Fundamentals, *Microchip Technology Inc.*, AN1332.
- [27] Lumisense Technologies., (2010) AC Current Sensor, *Lumisense Technologies Ltd.*, Chennai.
- [28] Halvorsen H. P., Sensors and Actuators with Arduino, white paper.

- [29] Robotshop., Link:<http://www.robotshop.com/en/current-voltage-sensors.html>.
- [30] Lawrey, S., Scott, D., (2016). Cambridge IGCSE® Computer Science Coursebook, *Cambridge University Press*, 1st Edition, ISBN:978-1107518698.
- [31] Rao, K. V., Rao, P. K., Ravi, B., (2016). PIR Sensor-Based Robot With HMH Applications, *International Journal of Engineering Research and Development*, Vol. **12**, Issue 5, PP.23-28, May.
- [32] Mathas, C., (2012). Sensing Motion with Passive Infrared (PIR) Sensors, Contributed By Electronic Products, Jun.
- [33] Sindhu, A. M., George, J., Roy, S., Chandra. J., (2016). Smart Streetlight Using IR Sensors, *IOSR Journal of Mobile Computing & Application (IOSR-JMCA)*, e-Vol. **3**, Issue 2, PP 39-44 ISSN: 2394-0050, Mar. - Apr.
- [34] Robotshop., Link: <http://www.robotshop.com/en/infrared-light-sensors.html>
- [35] Abdurrahman, B. S., (2014). Android Smart Switch For Smart Home, *Faculty of Electrical Engineering, Universiti Teknologi Malaysia*.
- [36] Robotshop., Link: <http://www.robotshop.com/ca/en/relays-led-controllers.html>
- [37] Gadgil, S. S., Lobo, R. R., (2016). Arduino Applications for Smart Cities, *International Journal of Computer Sciences and Engineering (IJCSE)*, Vol.-**4**(4), E-ISSN: 2347-2693, Jun.
- [38] Kavitate, S., Jallapelli, O., More, H., Nirve, S. A., (2017). E-JACKET, *International Journal of Advance Research and Innovative Ideas in Education (IJARIIE)*, ISSN:2395-4396, Vol. **3** Issue 2.
- [39] Ram, S., Tripathi, A., (2017). Android Controlled Air Conditioner Remote, *International Research Journal of Engineering and Technology (IRJET)*, e-ISSN: 2395 -0056, Vol. **04**, Issue: 03, Mar.

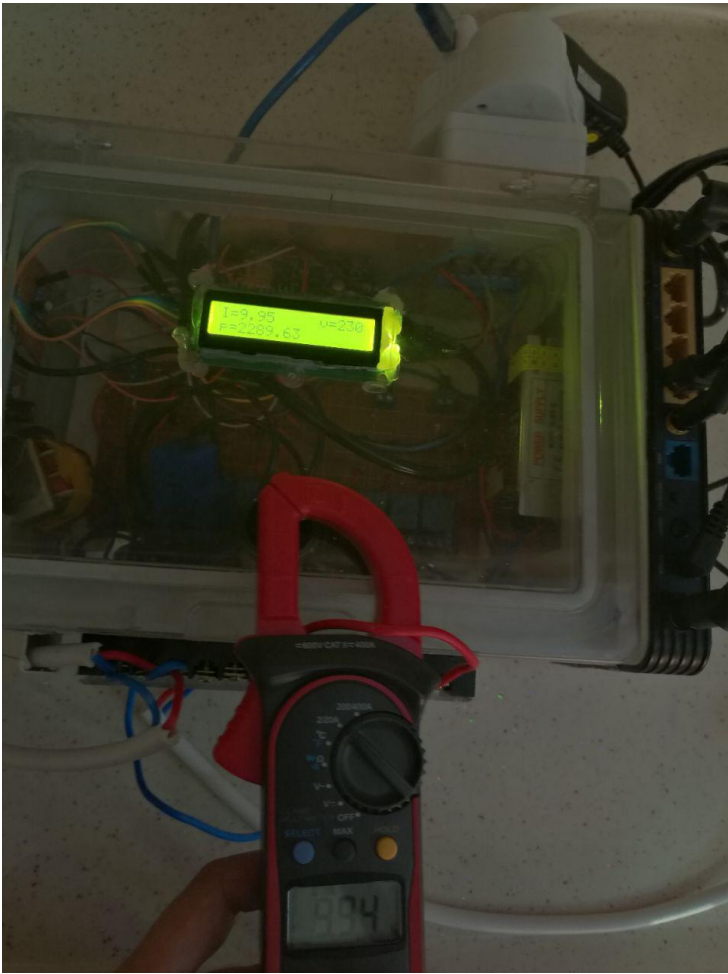
APPENDIX 1

Smart Switch System Components



APPENDIX 2

Voltage and current Readings





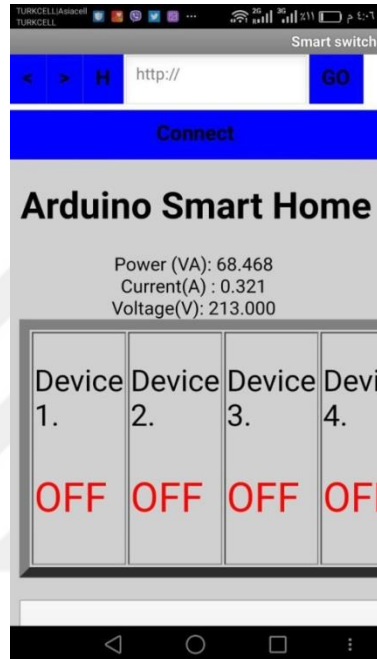






APPENDIX 3

Smartphone application GUI



APPENDIX 4

Smart Switch System Arduino C Code

Library definition for Ethernet shield and Liquid Crystal Display

```
#include <SPI.h>
```

```
#include <Ethernet.h>
```

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27,16,2);
```

Define gateway,mac address,subnet,IPAddress for Ethernet shield

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0x2D };
```

```
byte gateway[] = { 192, 168, 0, 1 };
```

```
byte subnet[] = { 255, 255, 255, 0 };
```

```
IPAddress ip(192,168,0,199);
```

```
EthernetServer server(80);
```

Define the output pins in Arduino mega microcontroller as the output signals to control with relays

(turn on or turn off)

```
#define RELAY_CH1 2
```

```
#define RELAY_CH2 3
```

```
#define RELAY_CH3 4
```

```
#define RELAY_CH4 5
```

```
#define RELAY_CH5 6
```

```
#define RELAY_CH6 7
```

```
#define RELAY_CH7 8
```

```
#define RELAY_CH8 9
```

```
String readString;
```

```
int reading;
```

```
float x;
```

```
float voltage;
```

```
float current;
```

```
const int currentPin = A1;
```

```
const unsigned long sampleTime = 100000UL;
```

sample over 100ms, it is an exact number of cycles for both 50Hz and 60Hz mains

```
const unsigned long numSamples = 250UL;
```

choose the number of samples to divide sampleTime exactly, but low enough for the ADC to keep up

```
const unsigned long sampleInterval = sampleTime/numSamples;
```

the sampling interval, must be longer than then ADC conversion time

```
const int adc_zero = 510;
```

```
void setup()
```

```
{
```

```
  lcd.init();
```

```
  Print a message to the LCD.
```

```
  lcd.backlight();
```

```
  lcd.print("smart house");
```

```
  delay(1500);
```

```
  lcd.clear();
```

```
  Serial.begin(9600);
```

```
  while (!Serial) {
```

```
  }
```

```
  pinMode(RELAY_CH1, OUTPUT);
```

```
  digitalWrite(RELAY_CH1, LOW); // switch on Device1
```

```
  pinMode(RELAY_CH2, OUTPUT);
```

```
  digitalWrite(RELAY_CH2, LOW); // switch on Device2
```

```
  pinMode(RELAY_CH3, OUTPUT);
```

```
  digitalWrite(RELAY_CH3, LOW); // switch on Device3
```

```
  pinMode(RELAY_CH4, OUTPUT);
```

```
  digitalWrite(RELAY_CH4, LOW); // switch on Device4
```

```
  pinMode(RELAY_CH5, OUTPUT);
```

```
  digitalWrite(RELAY_CH5, LOW); // switch on Device5
```

```
  pinMode(RELAY_CH6, OUTPUT);
```

```
  digitalWrite(RELAY_CH6, LOW); // switch on Device6
```

```
  pinMode(RELAY_CH7, OUTPUT);
```

```
  digitalWrite(RELAY_CH7, LOW); // switch on Device7
```

```
  pinMode(RELAY_CH8, OUTPUT);
```

```
  digitalWrite(RELAY_CH8, LOW); // switch on Device8
```

```
// Open serial communications and wait for port to open:
```

```
Serial.begin(9600);
```

```
// start the Ethernet connection and the server:
```

```
Ethernet.begin(mac, ip);
```



```

server.begin();
Serial.print("server is at ");
Serial.println(Ethernet.localIP());
}
void loop()
{
// the code used to read data from current sensor
int a=analogRead(A1);
unsigned long currentAcc = 0;
unsigned int count = 0;
unsigned long prevMicros = micros() - sampleInterval ;
while (count < numSamples)
{
if (micros() - prevMicros >= sampleInterval)
{
int adc_raw = analogRead(a) - adc_zero;
currentAcc += (unsigned long)(adc_raw * adc_raw);
++count;
prevMicros += sampleInterval;
}
}
float rms = sqrt((float)currentAcc/(float)numSamples) * (50 / 1024.0);
rms=rms-0.10;
if (rms<0.20)
{
rms=0;
}
current=2*rms;

//write the load current value on the LCD
lcd.backlight();
lcd.print("current=");
lcd.print(current);
// the code used to read data from voltage sensor
reading = analogRead(A0);
x= reading*5/1023;
voltage =x*75;
//write the voltage value on the LCD

```

```

lcd.setCursor(0, 1);
lcd.print("voltage=");
lcd.backlight();
lcd.print(voltage);
delay(500);
lcd.clear();
//The code used to create the HTML page to control with 8 devices and read the values of AC current
and voltage sensors
EthernetClient client = server.available();
if (client) {
Serial.println("new client");
boolean currentLineIsBlank = true;
while (client.connected())
if (client.available()) {
char c = client.read();
Serial.write(c);
if (readString.length() < 100) {
readString += c;
Serial.write(c);
if (c == '\n' && currentLineIsBlank) {
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("Connection: close");
client.println("Refresh: 5"); // refresh the page automatically every 5 sec
client.println();
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<HEAD>");
client.println("<body bgcolor=\"#D0D0D0\">");
client.print("<center> <p> <h1> Arduino Smart Home </h1></p> ");
reading = analogRead(current);
Serial.print("Read sensor: ");
switch (a)
{
case 0:
Serial.println("OK");
break;
case -1:

```

```

Serial.println("Checksum error");
break;
case -2:
Serial.println("Time out error");
break;
default:
Serial.println("Unknown error");
break;
}
client.print("Current (mA): ");
client.println((float)current,3);
client.println("<br />");
switch (y)
{
case 0:
Serial.println("OK");
break;
case -1:
Serial.println("Checksum error");
break;
case -2:
Serial.println("Time out error");
break;
default:
Serial.println("Unknown error");
break;
}
client.print("volte(v): ");
client.println((float)y, 3);
client.println("<br />");
// Relay Status Display
client.println("<center>");
client.println("<table border=\"9\">");
client.println("<tr>");
if (digitalRead(RELAY_CH1))
{
client.print("<td>  <p style=\"font-family:arial;color:black;font-size:26px;\">Device  1.</p><p
style=\"font-family:arial;color:green;font-size:35px;\">ON</p> </td>");

```

```

}
else
{
client.print("<td> <p style=\"font-family:arial;color:black;font-size:26px;\">Device 1.</p><p
style=\"font-family:arial;color:red;font-size:35px;\">OFF</p></td>");
}
if (digitalRead(RELAY_CH2))
{
client.print("<td> <p style=\"font-family:arial;color:black;font-size:26px;\">Device 2.</p><p
style=\"font-family:arial;color:green;font-size:35px;\">ON</p></td>");
}
else
{
client.print("<td> <p style=\"font-family:arial;color:black;font-size:26px;\">Device 2.</p><p
style=\"font-family:arial;color:red;font-size:35px;\">OFF</p></td>");
}
if (digitalRead(RELAY_CH3))
{
client.print("<td><p style=\"font-family:arial;color:black;font-size:26px;\">Device 3.</p><p
style=\"font-family:arial;color:green;font-size:35px;\">ON</p></td>");
}
else
{
client.print("<td><p style=\"font-family:arial;color:black;font-size:26px;\">Device 3.</p><p
style=\"font-family:arial;color:red;font-size:35px;\">OFF</p></td>");
}
if (digitalRead(RELAY_CH4))
{
client.print("<td><p style=\"font-family:arial;color:black;font-size:26px;\">Device 4.</p><p
style=\"font-family:arial;color:green;font-size:35px;\">ON</p></td>");
}
else
{
client.print("<td><p style=\"font-family:arial;color:black;font-size:26px;\">Device 4.</p><p
style=\"font-family:arial;color:red;font-size:35px;\">OFF</p></td>");
if (digitalRead(RELAY_CH5))
{

```

```

client.print("<td><p      style=\"font-family:arial;color:black;font-size:26px;\">Device      5.</p><p
style=\"font-family:arial;color:green;font-size:35px;\">ON</p></td>");
}
else
{
client.print("<td><p      style=\"font-family:arial;color:black;font-size:26px;\">Device      5.</p><p
style=\"font-family:arial;color:red;font-size:35px;\">OFF</p></td>");
}
if (digitalRead(RELAY_CH6))
{
client.print("<td><p      style=\"font-family:arial;color:black;font-size:26px;\">Device      6.</p><p
style=\"font-family:arial;color:green;font-size:35px;\">ON</p></td>");
}
else
{
client.print("<td><p      style=\"font-family:arial;color:black;font-size:26px;\">Device      6.</p><p
style=\"font-family:arial;color:red;font-size:35px;\">OFF</p></td>");
}
if (digitalRead(RELAY_CH7))
{
client.print("<td><p      style=\"font-family:arial;color:black;font-size:26px;\">Device      7.</p><p
style=\"font-family:arial;color:green;font-size:35px;\">ON</p></td>");
}
else
{
client.print("<td><p      style=\"font-family:arial;color:black;font-size:26px;\">Device      7.</p><p
style=\"font-family:arial;color:red;font-size:35px;\">OFF</p></td>");
}
if (digitalRead(RELAY_CH8))
{
client.print("<td><p      style=\"font-family:arial;color:black;font-size:26px;\">Device      8.</p><p
style=\"font-family:arial;color:green;font-size:35px;\">ON</p></td>");
}
else
{
client.print("<td><p      style=\"font-family:arial;color:black;font-size:26px;\">Device      8.</p><p
style=\"font-family:arial;color:red;font-size:35px;\">OFF</p></td>");
}

```

```

client.println("</tr>");
client.println("</table>");
client.println("</center>");
client.println("<br />");
// Relay Control Code
client.println("<a href='\"/?relay1on\"'> <button style='\"width:360px;height:120px\"'> <font
size='\"7\"'; color:red ;>Device 1 ON </font> </button> </a> ");
client.println("<a href='\"/?relay1off\"'> <button style='\"width:360px;height:120px\"'> <font
size='\"7\">Device 1 OFF </font> </button> </a> <br />");
client.println("<br />");
client.println("<a href='\"/?relay2on\"'> <button style='\"width:360px;height:120px\"'> <font
size='\"7\">Device 2 ON </font> </button> </a> ");
client.println("<a href='\"/?relay2off\"'> <button style='\"width:360px;height:120px\"'> <font
size='\"7\">Device 2 OFF </font> </button> </a> <br />");
client.println("<br />");
client.println("<a href='\"/?relay3on\"'> <button style='\"width:360px;height:120px\"'> <font
size='\"7\">Device 3 ON </font> </button> </a> ");
client.println("<a href='\"/?relay3off\"'> <button style='\"width:360px;height:120px\"'> <font
size='\"7\">Device 3 OFF </font> </button> </a> <br />");
client.println("<br />");
client.println("<a href='\"/?relay4on\"'> <button style='\"width:360px;height:120px\"'> <font
size='\"7\">Device 4 ON </font> </button> </a> ");
client.println("<a href='\"/?relay4off\"'> <button style='\"width:360px;height:120px\"'> <font
size='\"7\">Device 4 OFF </font> </button> </a> <br />");
client.println("<br />");
client.println("<a href='\"/?relay5on\"'> <button style='\"width:360px;height:120px\"'> <font
size='\"7\">Device 5 ON </font> </button> </a> ");
client.println("<a href='\"/?relay5off\"'><button style='\"width:360px;height:120px\"'> <font
size='\"7\">Device 5 OFF </font> </button> </a> <br />");
client.println("<br />");
client.println("<a href='\"/?relay6on\"'><button style='\"width:360px;height:120px\"'> <font
size='\"7\">Device 6 ON </font> </button> </a> ");
client.println("<a href='\"/?relay6off\"'><button style='\"width:360px;height:120px\"'> <font
size='\"7\">Device 6 OFF </font> </button> </a> <br />");
client.println("<br />");
client.println("<a href='\"/?relay7on\"'><button style='\"width:360px;height:120px\"'> <font
size='\"7\">Device 7 ON </font> </button> </a> ");

```

```

client.println("<a href='\"/?relay7off\"'><button style='\"width:360px;height:120px\"'> <font
size='\"7\"'>Device 7 OFF </font> </button> </a> <br />");
client.println("<br />");
client.println("<a href='\"/?relay8on\"'><button style='\"width:360px;height:120px\"'> <font
size='\"7\"'>Device 8 ON </font> </button> </a> ";
client.println("<a href='\"/?relay8off\"'><button style='\"width:360px;height:120px\"'> <font
size='\"7\"'>Device 8 OFF </font> </button> </a> <br />");
client.println("<br />");
// control arduino pin via ethernet Start
if(readString.indexOf("<?relay1on\">0)//checks for on
{
digitalWrite(RELAY_CH1, HIGH);
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/on.png' />");
//client.println("Light 1 Is On");
client.println("<br />");
}
else{
if(readString.indexOf("<?relay1off\">0)
{
digitalWrite(RELAY_CH1, LOW); // set pin 4 low
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/off.png' />");
//client.println("Light 1 Is Off");
client.println("<br />");
}
}
if(readString.indexOf("<?relay2on\">0)
{
digitalWrite(RELAY_CH2, HIGH);
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/on.png' />");
//client.println("Light 1 Is On");
client.println("<br />");
}
else{
if(readString.indexOf("<?relay2off\">0)//checks for off
{
digitalWrite(RELAY_CH2, LOW);
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/off.png' />");

```

```

//client.println("Light 1 Is Off");
client.println("<br />");
}
}
if(readString.indexOf("?relay3on") >0)//checks for on
{
digitalWrite(RELAY_CH3, HIGH); // set pin 4 high
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/on.png' />");
//client.println("Light 1 Is On");
client.println("<br />");
}
else{
if(readString.indexOf("?relay3off") >0)
{
digitalWrite(RELAY_CH3, LOW);
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/off.png' />");
//client.println("Light 1 Is Off");
client.println("<br />");
}
}
if(readString.indexOf("?relay4on") >0)
{
digitalWrite(RELAY_CH4, HIGH);

client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/on.png' />");
//client.println("Light 1 Is On");
client.println("<br />");
}
else{
if(readString.indexOf("?relay4off") >0)
{
digitalWrite(RELAY_CH4, LOW);
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/off.png' />");
//client.println("Light 1 Is Off");
client.println("<br />");
}
}
if(readString.indexOf("?relay5on") >0)

```



```

{
digitalWrite(RELAY_CH5, HIGH);
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/on.png' />");
//client.println("Light 1 Is On");
client.println("<br />");
}
else
{
if(readString.indexOf("?relay5off") >0)//checks for off
{
digitalWrite(RELAY_CH5, LOW);
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/off.png' />");
//client.println("Light 1 Is Off");
client.println("<br />");
}
}
if(readString.indexOf("?relay6on") >0)
{
digitalWrite(RELAY_CH6, HIGH);
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/on.png' />");
//client.println("Light 1 Is On");
client.println("<br />");
}
else{
if(readString.indexOf("?relay6off") >0)
{
digitalWrite(RELAY_CH6, LOW);
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/off.png' />");
//client.println("Light 1 Is Off");
client.println("<br />");
}
}
if(readString.indexOf("?relay7on") >0)
{
digitalWrite(RELAY_CH7, HIGH);
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/on.png' />");
//client.println("Light 1 Is On");
client.println("<br />");
}
}

```

```

}
else{
if(readString.indexOf("?relay7off") >0)
{
digitalWrite(RELAY_CH7, LOW);
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/off.png' />");
//client.println("Light 1 Is Off");
client.println("<br />");
}
}
if(readString.indexOf("?relay8on") >0)
{
digitalWrite(RELAY_CH8, HIGH);
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/on.png' />");
//client.println("Light 1 Is On");
client.println("<br />");
}
else{
if(readString.indexOf("?relay8off") >0) {
digitalWrite(RELAY_CH8, LOW);
client.println("<link rel='apple-touch-icon' href='http://chriscosma.co.cc/off.png' />");
//client.println("Light 1 Is Off");
client.println("<br />");
}
}
// client.println("<hr> <p> By <a href='\"http://androidcontrol.blogspot.com\"'></p><p style='\"font-
family:arial;color:blue;font-size:20px;\">Android Control Blogspot</p></a>");
readString="";
client.println("</body>");
client.println("</html>");
break;
}
if ( c == \n )
if (c == '\n') {
currentLineIsBlank = true;
}
else if (c != '\r') {
currentLineIsBlank = false;
}

```

```
} } }
```

```
// give the web browser time to receive the data
```

```
delay(1);
```

```
//client.println("<meta http-equiv=\"refresh\" content=\"10;url=http://192.168.0.199/\"/>");
```

```
// close the connection:
```

```
client.stop();
```

```
Serial.println("client disconnected");
```

```
}//if(Client)
```

```
}//Loop
```

