The Design & Construction

Of

A Home Automation

(Domotics)

By

AKINBOBOLA BLESSING

2003/15312EE

A PROJECT REPORT SUBMITTED TO THE

DEPARTMENT OF ELECTRICAL & COMPUTER

ENGINEERING, FEDERAL UNIVERSITY OF

TECHNOLOGY, MINNA,

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF BACHELOR OF ENGINEERING

IN ELECTRICAL & COMPUTER ENGINEERING.

NOVEMBER 2008

HOME AUTOMATION

(DOMOTICS)

AKINBOBOLA BLESSING

2003/15312EE

A Thesis submitted to the Department of Electrical and

Computer Engineering, Federal University of

Technology Minna, Niger State.

November 2008

DEDICATION

This project is dedicated to my late parents, my father, Eng Akinbobola Johnson and my dear mother, Mrs. Akinbobola Maltinda, May your gentle souls rest in peace. To my now only parents; my dear uncle, Eng Ajisegiri A. Benson and his dear wife (My mother), Dcn. Ajisegiri Moronke, they are the inspiration and motivation behind my education and well-being. Only God can reward your parenthoods over me.

To my siblings: Abisola, Omolayo, Ajibola, Gbenga, Aderemi, Omolayo, Ibukun, Sunkanmi, Jide, Ayo, Damilola, kole, Sade, Bose , Femi, Sis Abisola, Sis Titi, Olawole (ABOVE), your supports both in prayer and good wishes counts to this success.

DECLARATION

This is to certify that this project titled "Design and Construction of A Home Automation" was carried out by AKINBOBOLA BLESSING under the supervision of ENGR M.S. AHMED and submitted to the Department of Electrical and Computer Engineering of Federal University of Technology, Minna, for the award of Bachelor of Engineering (B.ENG) degree in Electrical and Computer Engineering.

AKINBOBOLA BLESSING	[illegible] 23/09/08
....	....
Name of student	Signature and date
ENGR M.S AHMED	[illegible] 23/05/08
....	....
Supervisor	Signature and date
ENGR. DR Y.A. ADEDIRAN	
....	....
Head of Department	Signature and date
....	....
External Examiner	Signature and date

ACKNOWLEDGEMENT

All thanks goes to the Almighty GOD for time and life that HE gives abundantly.

To my lecturers, friend, relatives and well wishers; the offspring of your prayers and good wishes which I cherish so much has given birth to these achievements, thank you so much.

I would like to thank ENGR M.S. AHMED, my project Supervisor, for providing me enough interactive time for counseling, question and answer sessions and general project supervision details as instructing me on Home Automation construction.

I would also like to thank the Head of Departments ENGR DR Y.A. ADEDIRAN for his detailed interactive instruction in ensuring proper project write-up.

ABSTRACT

My research sought to simulate a computer-controlled environment on a smaller scale to assess the positive and negative aspects of an automated home.

Through this technology, homeowners could lead a more efficient lifestyle. With the integration of computer and electrical engineering, I designed a system of computer-controlled appliances that react to various stimuli.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

1.1 General Introduction

With the computational power and speed of today's computers, nearly everything can be automated to eliminate the hassle of manual work. Switching the lights on and off is too difficult for most contemporary Nigerians. Therefore, I designed and built the components for a computer-controlled room to facilitate this process.

In the 1950s, the "home of the future" was inclined to rapid manufacturing. The vision of a futuristic home during this period consisted of a home that was easily transported, rapidly constructed, and produced in large numbers; inside, the home would have built-in furniture and a small kitchen, and the entire structure would be made of plastic. Hence, "Living in the home of the future was like living in a Swiss army knife" [1]. However, a few decades later, the theory of a smart room has developed and the new vision of the "home of the future" is no different from the 1950s caricature, except that the innovative home would function through artificial intelligence.

An automated household or a Smart Room also called **smart homes** or **domotics** "is a physical space for living or working in that is agent controlled and can bring computational power embedded within it to bear in a manner that helps users of the environment perform their daily tasks" [2]

Automated rooms, in theory, are designed to sense what is occurring within its walls and respond by commanding lights, doors, etc in order to demonstrate intelligent behavior and help the inhabitants of the home. Javier Zamora, the general manager of Eneo Labs, [3] an organization that is trying to create a smart home, explains that a smart home would be able to predict the inhabitants' routine and adapt consequently, Zamora also commented that smart houses have two focal constituents: an "information network," which was "like a human body's nervous system in that all devices inside the house would be connected to it; and a "brain," which coordinated what was inside the home and connected it to what went on outside."[4] A smart room has an intelligent agent that is able to attain and apply knowledge about the environment and its residents in order to improve their lives.

Other applications of computer controlled electrical devices are not as complex but still just as popular. With a computer sequencing software, an elaborate electrical circuitry, and with brilliant craftsmanship, several advocates have managed to create stunning displays with computer controlled lights. Enthusiasts, like Mark Obermiller-a renowned celebrity for his computer controlled Christmas lights [5], have made it their hobby to create these amazing spectacles with lights and electrical control boxes that turn the lights on and off, and the software that run the control boxes and program the animated sequences. These small groups of hobbyists usually use Dasher, a software program that configures specific settings for the lights used in the display. The control hardware, used by these small groups, takes a computer's "low voltage digital signals and translates them to the standard electrical current powering the lights." [6] This process can be computed by using digital input-output cards that are connected to electromagnetic relays that in turn are connected to light circuits, or by implementing a control box to a computer's parallel ports as well as to light circuits. [7]

The use of automated devices on various levels whether it be the home or for entertainment, has proven that there is a market for computer controlled appliances.

1.2 Aims and Objectives

Working with basic electric components, my goal was to design a system that would control 8 different circuits (one for each output pin of the parallel port). Specifically, I needed to:

1. Build electromagnetic relay that connect to the parallel port of a computer and control lighting in an electric circuit.
2. Construct a box of outlets that are controlled by the relays
3. Design a main box with openings for the outlets and parallel port.
4. Develop software, with a graphical user interface, to control the electromagnetic relays, taking input from the user and sending output to the parallel port. Unlike in the X10 system, the software will allow for full appliance control and customizability.

1.3 Motivation

In the world today, most electrical hazards resulting to burnt, damages and inferno are traceable to improper electrical installation. According to Smarthouse Magazine "most electricians don't have a clue what to do...

when it comes to installation". But when it come to home automation installation and installation, they hardly make mistakes as it is called DOMOTICS "its full baked home automation".

1.4 Scope

This project is centered at using the controlled 5v from the computer through the printer port with designed software to control home appliances that are on 220v and then test its functionality.

1.5 Sources of Materials

Materials for this project were gotten from the Internet, past lecture notes, various text books and consultation with my supervisor, other lecturers and colleagues.

1.6 Outline

This chapter introduces the project (Design and Construction of a Home Automation). It reveals the motive, aim, methodology and scope of the project and also gives a brief historical background on Home Automation.

CHAPTER TWO

LITERATURE REVIEW / THEORETICAL BACKGROUND

2.1 Early History and Previous work

Products such as X10 controllers and IBM's Home Director allow individuals to automate their homes with little to no adjustments to the electrical wiring of the home.[10] In particular, X10 is a communications language that allows electrical devices to send digital information to each other through brief radio frequency bursts over a power line.[11] Thus when used in the home, X10 controllers allow compatible products to communicate through a structure's existing electrical wiring. Installation of X10 is relatively simple considering that it only requires a transmitter and receivers plugged into the household appliances and an outlet.

Inexpensive and widely available, X10 is user-friendly and requires little work to setup. Despite its benefits, however, X10 falls short of fulfilling the possibilities of a truly automated room. The use of a transmitter as opposed to a central computer to control electrical devices can lead to many problems and errors. For example, since X10 communicates through sending radio signals, interference between signals is likely to occur whereas an appliance following a computer program would not face such an issue. In addition, X10 does not have a sophisticated sensor system. Though X10 controllers enable time sensors, the system is not yet sophisticated enough to accommodate sound, touch, or infrared as a device stimulus.

In comparison to X10 controllers. IBM's Home Director is a home networking system that goes beyond simple appliances and connects the home's subsystems such as lighting. heating. air conditioning as Ill as security systems. [12] Home Director also links devices such as PCs and DVD players. but unlike X10. the system is limited when it comes to simple appliances such as coffeemakers and the toaster. It is also a much more complicated installation process than that of X10 as it requires professional installation. Indeed. X10 and Home Director appear to present a similar product that operates on two different levels. While X10 is best for electrical devices belonging to a smaller scale. Home Director is specialized to coordinate the larger subsystems of the home. However. both are ineffective systems if a homeowner wishes to automate every electrical activity in the home. With our design. I hope to make complete automation of one's home a possibility.

Smart houses are a reality and they are the gateway to a future that was once only possible in imagination. In my research. I examined in detail the basic principles of a computer-controlled room with a specific focus on lighting activated by sensors: in particular. RFID tags. Furthermore. I explore its possible application to everyday life by assessing the results of my small-scale simulation.

2.20 Major Components Used

2.21 Opto-coupler (4N35)

Description

The 4N35 is an opto-coupler for general purpose applications. It contains a light emitting diode optically coupled to a phototransistor. It is packaged in a 6-pin DIP package and available in wide lead spacing option and lead bend SMD option. Response time, tr, is typically 3 μs and minimum CTR is 100% at input current of 10 mA.

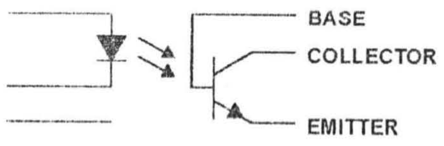


Fig 2.1 Circuit Diagram

Features

1. High Current Transfer Ratio (CTR: min. 100% at IF = 10 mA, VCE = 10 V)
2. Response time (tr: typ., 3 μs at VCE = 10 V, IC = 2 mA, RL = 100 Ω)
3. Input-output isolation voltage (Viso = 3550 Vrms)
4. Dual-in-line package

Applications

1. I/O interfaces for computers
2. System appliances, measuring instruments
3. Signal transmission between circuits of different potentials and impedance

2.22 Electromechanical Relays

The **electromechanical relay** (EMR) is a device that uses an electromagnet to provide the force to close (or open) switch contacts, in other words, an electrically powered switch.

When the electromagnet, called the *coil*, is energized, it pulls down on the spring-loaded *armature*. Relay contacts are described as being one of two kinds: normally open contacts (NO), which are open in the unenergized state, and normally closed contacts (NC), which are closed in the unenergized state.

The electrical specifications for the contacts are different than for the coil. For the contacts, the maximum current and voltage for DC and AC operation is specified. For the coil, the intended voltage and coil resistance are usually specified. The coil voltage and resistance can be used to calculate the steady-state coil current. Actually, it takes more voltage and current to pull in the relay contacts than it does to hold them there because the armature must be pulled in across an air gap. Hence, these quantities are called, respectively, **pull-in voltage** and **pull-in current.** For example, the contacts of a particular 6-V relay actually close at 2.1 V and stay closed until the voltage is decreased to 1 V. The values of voltage and current needed to keep the relay energized are called the **minimum holding voltage** and **sealed current.** Notice that the actual pull-in voltage is much less than the rated coil voltage.

This is to guarantee that the relay will pull quickly and reliably when operated at the rated voltage. Coil voltages are specified to be AC or DC. The difference is that AC coils are constructed with **shaded poles** to prevent "buzzing" with 60-cycle power. A shaded-pole relay has a metal ring around the pole face of the electromagnet. Magnetic flux induced into this ring keeps the relay closed when the AC cycles through 0 V.

2.23 The Parallel Port

A port contains a set of signal lines that the CPU sends or receives data with other components. We use ports to communicate via modem, printer, keyboard, mouse and other such devices. In signaling, open signals are "1" and close signals are "0" so it is like binary system. A parallel port sends 8 bits and receives 5 bits at a time.

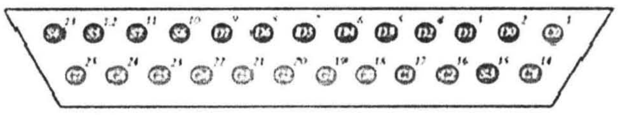
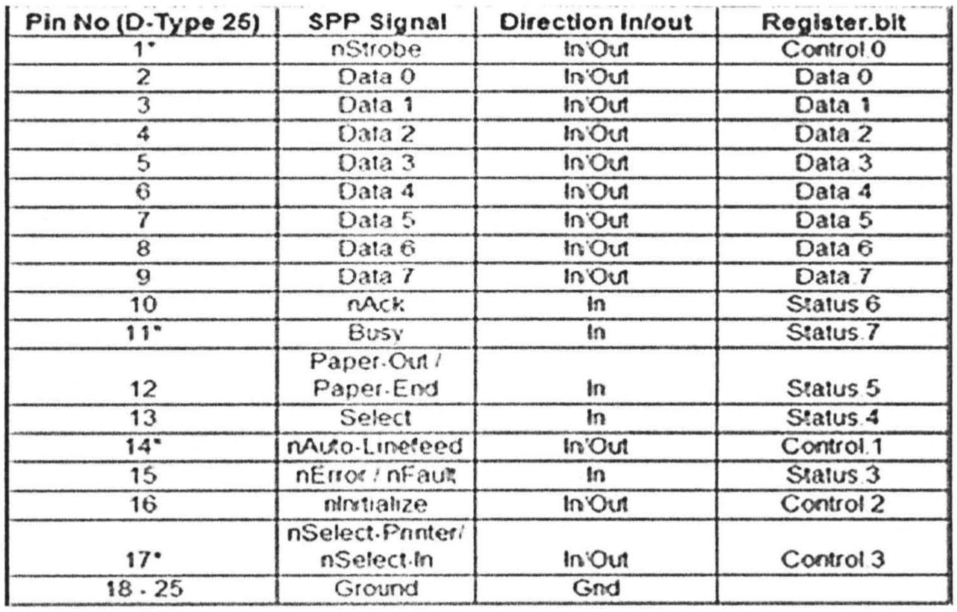


Fig 2.2 The Parallel Port

The parallel port comprises of 3 different ports the Data Port, the Status Port and the Control Port. These ports are distributed among the 25 PIN of the port. The data port ranges from PIN 2 to PIN 9 i.e. 8 PIN indicating a byte. The Status port ranges from PIN 10 to PIN 13 and the PIN 15 is also a part of the status port. The Control port comprises of PINs 1, 14, 16 and 17. The detail of these PINs and their functions is given below:



Pin No (D-Type 25)	SPP Signal	Direction In/out	Register.bit
1*	nStrobe	In/Out	Control 0
2	Data 0	In/Out	Data 0
3	Data 1	In/Out	Data 1
4	Data 2	In/Out	Data 2
5	Data 3	In/Out	Data 3
6	Data 4	In/Out	Data 4
7	Data 5	In/Out	Data 5
8	Data 6	In/Out	Data 6
9	Data 7	In/Out	Data 7
10	nAck	In	Status 6
11*	Busy	In	Status 7
12	Paper-Out / Paper-End	In	Status 5
13	Select	In	Status 4
14*	nAuto-Linefeed	In/Out	Control 1
15	nError / nFault	In	Status 3
16	nInitialize	In/Out	Control 2
17*	nSelect-Printer / nSelect-In	In/Out	Control 3
18 - 25	Ground	Gnd	

Fig 2.3 Parallel port details and functions

* Pins with * symbol in this table are hardware inverted. That means, If a pin has a 'low' ie. 0V, Corresponding bit in the register will have value 1. Signals with prefix 'n' are active low. Normally these pins will have low value. When it needs to send some indication, it will become high. For example, normally nStrobe will be high, when the data is placed in the port, computer makes that pin low.

Normally, data, control and status registers will have following addresses. We need these addresses in programming later.

Register	LPT1	LPT2
Data register (Base Address + 0)	0x378	0x278
Status register (Base Address + 1)	0x379	0x279
Control register (Base Address + 2)	0x37a	0x27a

Fig2.4 Parallel port address lines

By default, data port is output port. To enable the bidirectional property of the port, we need to set the bit 5 of control register.

To know the details of parallel ports available in computer, follow this procedure:

1. Right click on My Computer, go to "Properties".
2. Select the tab Hardware, Click Device manager.
3. You will get a tree structure of devices; in that Expand "Ports (Com1 &LPT)".
4. Double Click on the ECP Printer Port (LPT1) or any other LPT port if available.
5. You will get details of LPT port. Make sure that "Use this Port (enable)" is selected.
6. Select tab recourses. In that you will get the address range of port.

CHAPTER THREE

DESIGN AND IMPLEMENTATION

3.1 The Control Relay Switch Circuit

The heart of a computer-controlled room is a small circuit switch known as: **The Control Relay Switch Circuit**. The Relay uses a 5V DC control current to switch a much stronger current (240V AC) on and off. This is accomplished by using a Relay, an opto-coupler, and three resistors to control the flow of electricity over the circuit, a diode and LED as functionality indicator. This is a type of relay (Normally Open / Normally Close) that cuts all current flow across the output socket to appliances. In the circuit I developed, the relay serve as the gate mechanism for the wall current by alternatively restricting and opening the current to reach the outlet that the controlled appliance is plugged in to.

The opto-isolator is a 6-pin transistor and Light emitting diode that has complete electrical isolation from one side to the other. When certain voltages are applied to the pins on one side, a LED comes on, triggering a phototransistor to let current flow between the ports on the opposite side. In the relay control circuit, the opto-isolator acts as an intermediary between the parallel port input and the relay. When the parallel port sends voltage into the opto-isolator, the component allows voltage to reach the 100 ohm coiled pins on the electromechanical relay, which consequently close the circuit and allows wall current to flow.

The resistors in the control switch are devices that allow the voltage across the sensitive opto-isolator to prevent it from being destroyed. Without these resistors, the LED in the opto-isolator would immediately be burned out, rendering the circuit useless.

3.2 Implementation

I constructed 8 relay switches for this project. For each relay, opto-isolator, the transistor and two resistors were laid out on a circuit board. They were then soldered together at the appropriate pins, as demonstrated in Figure #1. Each switch circuit was connected to four pieces of stripped copper wire, two outputs and two inputs. One of the input wires was connected to a pin on the parallel port and carried a current to the opto-isolator. A 470-ohm resistor was used between the parallel port and the opto-isolator to decrease the voltage. To complete the circuit, a wire was connected to another pin on the opto-isolator, serving as a ground; I also put a Light Emitting Diode (LED) between the Parallel port, the ground and the opto-isolator for functionality test of the Computer Parallel port.

The other end of the elctromechanical relay was soldered to one input wire and one output wire. This formed a 220 volt AC circuit that was controlled by the opto-isolator.

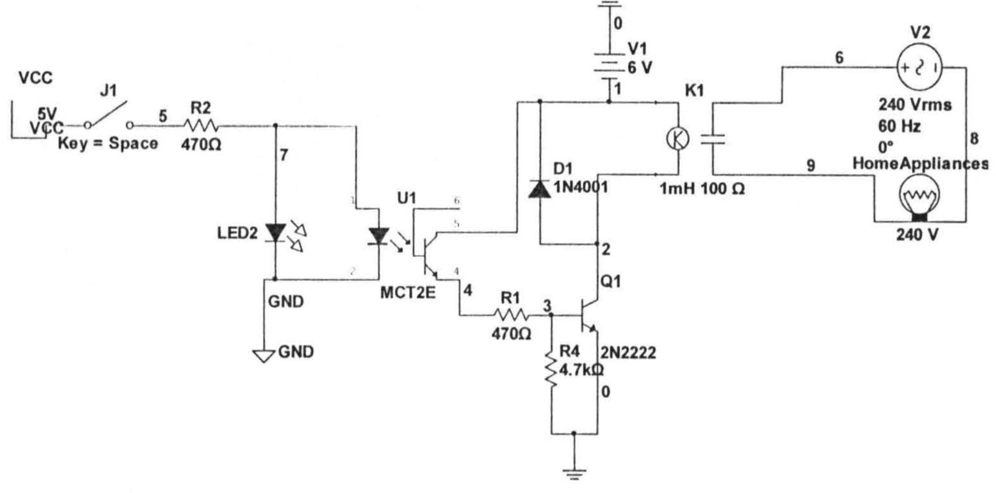


Fig 3.1 The Circuit Diagram

NOTE: 8 of the above circuit was built and used to control 8 home electrical appliances (TV, Home Theater, Computer, Bulbs etc). The complete Circuit Diagram is on Appendix C and the block diagram on appendix B.

3.30 Parallel Port Control Circuit

3.31 Connecting circuits to parallel port

PC parallel port is 25-pin D-shaped female connector in the back of the computer. It is normally used for connecting computer to printer, but many other types of hardware for that port

Not all 25 are needed always. Usually you can easily do with only 8 output pins (data lines) and signal ground. I have presented those pins in the table below. Those output pins are adequate for many purposes.

3.32 Pin Mapping and function

2	D0
3	D1
4	D2
5	D3
6	D4
7	D5
8	D6
9	D7

Pins 18,19,20,21,22,23,24 and 25 are all ground pins.

Those data pins are TTL level output pins. This means that they put out ideally 0V when they are in low logic level (0) and +5V when they are in high logic level (1). In real world the voltages can be something different from ideal when the circuit is loaded. The output current capacity of the parallel port is limited to only few milli-amperes.

The outputs are designed so that they give at least 2.4V at 2.6 mA load...

When taking current from PC parallel port, keep the load low, only up to few milli-amperes. Trying to toke too much current (for example shorting pins to ground) can fry the parallel port. I have not killed any parallel port (yet) in this method, but I have had in cases where too much load has made the parallel port IC very hot. Extra careful is required.

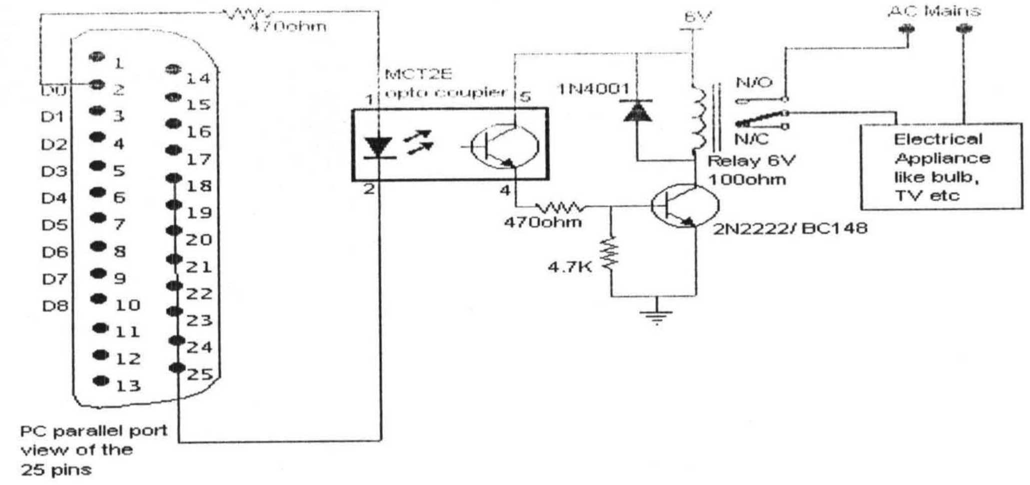


Fig 3.2 The Circuit Diagram showing the Parallel Connection

3.40 Parallel Port Control Programming

One of the key components of our computer-controlled room project is the software responsible for controlling the room itself. The program is written in C, although any language which can acquire parallel port access would have been suitable.

Although the room controlling software, RController, was written mainly as a proof-of-concept, it meets many of the criteria for a program deployed publicly for the same task. RController was designed to allow the end user of this product to configure when changes in the state of attached devices would occur. This program supports both time and RFID “sensors.” The decision to unify the factors which control the room under the class of sensors was made to facilitate setup and allow for changes to be quickly made. RController is assigned events and which changes to invoke upon this event. Events are prioritized to create complex chains of actions. For example, an event/change pair such as "the toaster is on between 7:00 and 9:00am when George is present" can be created. RController is capable of using any number of parallel ports to control individual devices. These devices can be named and referred to by name once created.

Control is done by means of sensors. All sensors implement a common interface, allowing for the addition of future sensors easily. Current sensors include both an RFID interface, and a time module. Through chains of events, complex events can be created to control individual devices with multiple criteria. A priority-based system was used to allow for complex events.

RController is written so as to allow for easy future expansion. The modular design of the program makes the addition of new sensors easy. Control by new ports can be accomplished by modifying the port controller to allow for additional nonparallel port addresses. The program can be readily modified to include any conventional port, although changes in the hardware would need to take place for the system to be compatible with additional ports.

The main library controlling the communications between the computer and the devices, such as the lamps, is a native DLL named InpOut32. This allows the user to specify where the port is, which pins are connected to which devices, and whether or not the devices are active. The user interface (UI) will handle the rest, including setting the state of the port and registering the devices. Each device is given a unique name which can be specified by either the user or the computer; the UI will make sure that every parameter of each device is available so that no conflicts arise. The Phidgets library is also implemented in the program. This library provides functions for controlling the USB RFID device used for input in this project.

3.41 The C-Program

The program to control devices using PC parallel port is written in C language, the devices are controlled by pressing the keys 1-8 that corresponds to each of the 8 possible devices. The program written for this work is in Appendix 1

3.42 The Controlling Software

The above code is the compiled with Borland C++ builder, as a simple executable (.EXE) file for win32 application, which can be launched independently or via command prompt. Using the keyboard keys 1-8, the various controlling switch can then be controlled on or off (0/1) which resulted in triggering the relay switch and control the flow of 240v into the outlet socket.

3.50 Parallel port modes

The IEEE 1284 Standard which has been published in 1994 defines five modes of data transfer for parallel port. They are:

1. Compatibility Mode
2. Nibble Mode
3. Byte Mode
4. EPP
5. ECP

The programs, circuits, and other information found in this tutorial are compatible to almost all types of parallel ports and can be used without any problems

3.51 Hardware

The pin outs of DB25 connector is shown in the picture below:

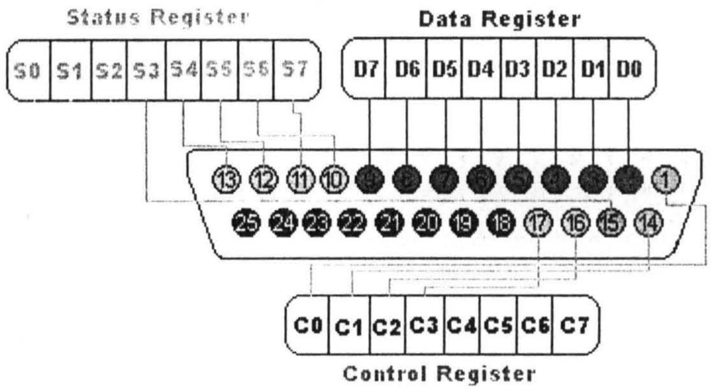


Fig 3.3 The Pin outs of DB25 Connector

The lines in DB25 connector are divided into three groups, they are:

1. Data lines (data bus)
2. Control lines
3. Status lines

As the name refers, data is transferred over data lines. Control lines are used to control the peripheral, and of course, the peripheral returns status signals back to the computer through Status lines. These lines are connected to Data, Control and Status registers internally. The details of parallel port signal lines are given below:

Pin No (DB25)	Signal name	Direction	Register - bit	Inverted
1	nStrobe	Out	Control-	Yes

			0	
2	Data0	In/Out	Data-0	No
3	Data1	In/Out	Data-1	No
4	Data2	In/Out	Data-2	No
5	Data3	In/Out	Data-3	No
6	Data4	In/Out	Data-4	No
7	Data5	In/Out	Data-5	No
8	Data6	In/Out	Data-6	No
9	Data7	In/Out	Data-7	No
10	nAck	In	Status-6	No
11	Busy	In	Status-7	Yes
12	Paper-Out	In	Status-5	No
13	Select	In	Status-4	No
14	Linefeed	Out	Control-1	Yes
15	nError	In	Status-3	No
16	nInitialize	Out	Control-2	No
17	nSelect-Printer	Out	Control-3	Yes
18-25	Ground	-	-	-

Fig 3.4 parallel port signal lines

3.52 Parallel port registers

The Data, Control and Status lines are connected to there corresponding registers inside the computer. So, by manipulating these registers in program, one can easily read or write to parallel port with programming languages like 'C' and BASIC.

The registers found in a standard parallel port are:

1. Data register
2. Status register
3. Control register

As their names specify, Data register is connected to Data lines, Control register is connected to Control lines and Status register is connected to Status lines. (Here the word connection does not mean that there is some physical connection between data/control/status lines. The registers are virtually connected to the corresponding lines.) So, whatever you write to these registers will appear in the corresponding lines as voltages. Of course, you can measure it with a multimeter. And whatever you give to Parallel port as voltages can be read from these registers (with some restrictions). For example, if we write '1' to Data register, the line Data0 will be driven to +5v. Just like this, we can programmatically turn on and off any of the Data lines and Control lines.

3.53 Location of the registers

In an IBM PC, these registers are IO mapped and will have an unique address. We have to find these addresses to work with the parallel port. For a typical PC, the base address of LPT1 is 0x378 and of LPT2 is 0x278. The Data register resides at this base address, Status register at base address + 1 and the control register is at base address + 2. So, once we have the base address, we can calculate the address of each register in this manner. The table below shows the register addresses of LPT1 and LPT2.

Register	**LPT1**	**LPT2**
Data register (baseaddress + 0)	0x378	0x278
Status register (baseaddress + 1)	0x379	0x279
Control register (baseaddress + 2)	0x37a	0x27a

Fig 3.5 Register addresses of LPT1 and LPT2

Using the code

I used *Inpout32.dll* (or *hwinterface.ocx*) for Win 98/NT/2000/XP. This DLL has the following features:

1. Works seamless with all versions of Windows (Win 98, NT, 2000 and XP)
2. Using a kernel mode driver embedded in the DLL
3. No special software or driver installation required
4. Driver will be automatically installed and configured automatically when the DLL is loaded
5. No special APIs required, only two functions Inp32 and Out32.
6. Can be easily used with VC++ and VB or C#.

```
val=axHwinterface1.InPort(888);
axHwinterface1.OutPort(888,(short)(val|2));
```

The above code reads the value from port 0x378 (888 decimal) and then OR with value 2. This code will send value 1 to D1.

3.6 The 6V Sourcing Circuit

To trigger the relay, 6v is required as shown in the circuit diagram, the design for this contain 4-diodes 1N4001 arranged in a rectifying mode to convert the 6V AC from the transformer into 6V DC with a 10V, 2200MicroFarad Capacitor to filter any remained AC in the output. The ouput reads on the XMM1 (multi-meter) is +6V.

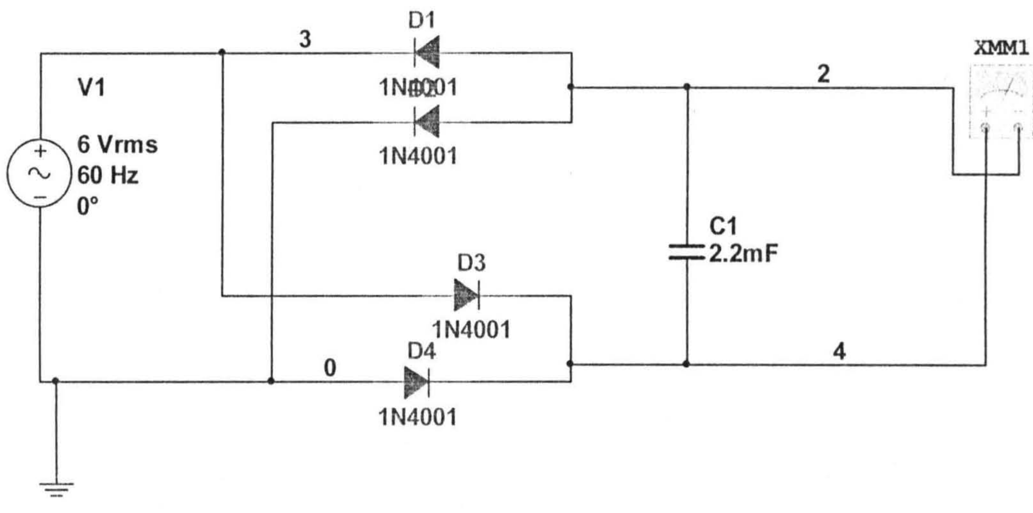


Fig 3.6 The 6v Relay Source Circuit

3.70 Outlet Box

In a typical room, electric devices are plugged directly into the wall outlet. To control these devices with a computer, I wired the relays between the wall outlet and two outlet boxes. The entire circuit is demonstrated in Figure #2. The outlet box consists of 8 outlets socket fix on the front panel of the case. The box is powered by a computer plug at the back of the box. Each relay was attached to one of the output pins on the parallel port and to one pair of outlets.

The switch controlling circuits were then grounded to an external object to dissipate any electrical buildup that could destroy the opto-isolator or the transistor. As a further precaution, rubber tubing was used to shield the wire coming out of the parallel port, and the rubber was shrunk with a heat gun. All the 8 controlling circuits were placed into a 6 by 6 by 4 plastic boxes, with an opening for the parallel port.

3.80 Main Box

My last physical challenge was the construction of a container for the outlet boxes and SSR box. I chose transparent plastic as my building material due to its transparency nature which enables visibility of the components used.

CHAPTER FOUR

TESTS, RESULTS AND DISCUSSION

4.1 Testing Steps and Results Obtained

The details of parallel port signal lines are given below:

Pin No (DB25)	Signal name	Direction	Register - bit	Inverted
1	nStrobe	Out	Control-0	Yes
2	Data0	In/Out	Data-0	No
3	Data1	In/Out	Data-1	No
4	Data2	In/Out	Data-2	No
5	Data3	In/Out	Data-3	No
6	Data4	In/Out	Data-4	No
7	Data5	In/Out	Data-5	No
8	Data6	In/Out	Data-6	No
9	Data7	In/Out	Data-7	No
10	nAck	In	Status-6	No
11	Busy	In	Status-7	Yes
12	Paper-Out	In	Status-5	No
13	Select	In	Status-4	No
14	Linefeed	Out	Control-1	Yes
15	nError	In	Status-3	No
16	nInitialize	Out	Control-2	No
17	nSelect-Printer	Out	Control-3	Yes
18-25	Ground	-	-	-

Fig 4.1 Parallel port signal lines detailed results

4.2 Discussion of Results

Using,

```
        val=axHwinterface1.InPort(888);
        axHwinterface1.OutPort(888,(short)(val|2));
```

The above code reads the value from port 0x378 (888 decimal) and then OR with value 2. This code will send value 1 to D1.

Now in the Program.cs which is the entry point for the Console application, the following code will go in the **Main** method:

```
int address = 888;

int value = 24;

PortAccessAPI.Output(adress, value);
```

Here the address **888** as int is actually **0x378** as Hex, which is the data port of the parallel port.

To reset the data that is sent on the data port, you need to invoke the Output method with a value 0x00 i.e.0 as shown below:

```
int address = 888;

PortAccessAPI.Output(adress, 0);
```

This was all about writing data to the parallel port; now to read data from the parallel port. In the file PortAccessAPI.cs I have declared a function "Input", this will be used to read the parallel port. This function takes a parameter "address", this is the address of the parallel port that we want to read. This method will return an integer as the data that is read from the requested port. The code will look like:

```
int address = 888;

int value;

value = PortAccessAPI.Input(adress);
```

The variable "**value**" will contain the data that is read from the parallel port **0x378**.

4.3 Relevance, Limitation and Suggested Remedy

At the conclusion of this project, I had spent an estimated N16,000.00 on materials. Furthermore, through mistakes in building the design, many materials were wasted and parts of my final project were damaged such as parallel port pins and relay. However, the design worked despite the various shortcomings.

My project proves the effectiveness of the electromagnetic relays as a circuit controller device; it is possible to use a low, DC current from a computer to toggle a high AC current on and off using an SSR. A Solid State Relay was used as opposed to an electromechanical relay due to the break in signal caused by the opto-isolator; this allows for fast, reliable switching capability.

Furthermore, I have ascertained that the parallel port is in fact the optimum computer port to use for controlling appliances. The parallel port sends low, harmless signals of 5 volts to the optoisolator of the Solid State Relay; this means there is a low risk of injury. Also, the parallel port has 8 output pins and is easy to control with inpout32.dll, a freeware parallel port driver.

The use of RFID as a sensor also proved to be successful. This is an indication that RFIDs could be practically utilized as a sensory tool in computer-controlled lighting and also automated homes.

On a software level, a program written in any high level language can be used to read input from the USB RFID and output to the parallel port with inpout32.dll. In comparison to existing home automation systems, my current design is still crude, and without refinement, it is an impractical alternative to a system such as X10. However, my designs show advantages over the X10 system in that devices would be powered by a central computer as Ill as have more sensory abilities, thus allowing more sophisticated options for user preferences.

CHAPTER FIVE

Conclusion

5.1 General Summary

The success of the implementation of this project is as a result of the combination of the components used with calculated values and regulated, thus future expansion is also welcome with specifications of any additional components or devices used.

5.2 Open Issues

The parallel port is become less common; therefore, "smart" houses will likely use another interface, such as USB or Wifi. Computer-controlled devices can extend past the home; cars, planes, and factory robots are already controlled, to a certain extent, by computers. In years to come, the extent of computer control will almost certainly increase, making life simpler for everybody.

It is also important to remember that for two reasons, relays have a finite life. First, because the relay is a mechanical device, the moving parts eventually worn out, and second, the electrical contacts can become pitted because of arcing. The contact part is very dependent on the electric current that is being switched. For example, a certain relay is rated for 9 million operations at 1.5 amps but only 2 million operations at 3 amps. Two million operations sounds like a lot, but if this relay Ire in a 24-hour factory being used in a repetitive operation every 10 seconds, it would have to be replaced every 8 months. Contact life also depends on the type of load being controlled. For example, inductive loads such as motors cause much more arcing and pitting than resistive loads such as lights and heating elements.

5.3 Recommendations

Upon completion of the project, it became clear that a system of computer controlled outlets is relatively simple to design and implement. These outlets can be used to control lighting in a room, as Ill as any other electric appliances. It is probable that computer-controlled rooms will develop into computer-controlled houses, or "smart" houses and become omnipresent in the near future.

A "smart" house will implement a similar design to the one I created, except every wall outlet will be an integral part of the system. Also, the entirety of the house will be controlled by one central computer. Yet, the priority of a smart home should not focus only on providing a domestic bliss but on other humanitarian causes. One of the most promising uses of a smart home is to help elderly people to live in a secure and independent environment.

There has been an increment of individuals who are physically impaired due to aging. In1985, five and a half million disabled elders were living at home. But it is estimated that by 2030, seventy million Americans will be over the age of 65 [5] and over ten million of the elderly will be living at home with disabilities[6].

Even with these statistics, surveys indicate that most elders want to remain in their homes as they grow old despite their disabilities which may compromise their safety. The development of a smart room for seniors "can tremendously impact and facilitate the desire of adults to age in place" [7].

References

[1].http://www.popularmechanics.com/blogs /technology_news/4216434.html

[2].http://people.arch.usyd.edu.au/~mary/Pubs/2005pdf/Ubiq_Comptg_Macindoe.pdf

[3].http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=27
2806

[4].http://www.cnn.com/2005/TECH/05/27/ vision.home/index.html

[5].S. Lanspery, J.J. Callahan Jr., J.R. Miller, and J. Hyde. Introduction: Staying
put. In S. Lanspery and J. Hyde, editors, *Staying Put: Adapting the Places Instead of the People*, pages 1-22, Baywood Publishing Company, 1997.

[6].R. Elliott. Assistive technology for the frail elderly: An overview. U.S. Department of Health and Human Services, aspe.hhs.gov, 1991.

[7].http://www.todaysengineer.org/2007/May/smart_homes.asp

[8]. http://en.wikipedia.org/wiki/RFID

[9].http://www.computerchristmaslights.com/articles.htm

[10].http://www.smarthome.com/about_x10.html

[11].http://en.wikipedia.org/wiki/X10_(industry_standard)

[12].http://www.hometoys.com/htinews/feb99/articles/ibm/director.htm

Appendix A

THE C PROGRAM

```
/*program to control devices using PC parallel port

The devices are controlled by pressing the keys 1-8

that corresponds to each of the 8 possible devices

*/



#include<dos.h>

#include<stdio.h>

#include<conio.h>

#define PORT 0x378 /* This is the parallel port address */



main()

{

char val=0, key=0;

char str1[] ="ON ";

char str2[] ="OFF";

char *str;

clrscr();

printf ("Press the appropriate number key to turn on/off devices:\n\n");

printf ("Here Device1 is connected to D0 of parallel port and so on\n\n");

printf ("Press \"x\" to quit\n\n");

gotoxy(1,8);

printf ("Device1:OFF Device2:OFF Device3:OFF Device4:OFF\n");

printf ("Device5:OFF Device6:OFF Device7:OFF Device8:OFF");
```

```
while(key!='x' && key!='X')

{

gotoxy(1,12);

printf ("Value in hex sent to the port:");

key=getch();

switch(key){


case '1':


        gotoxy(9,8);

        val=(val&0x01)?(val&(~0x01)):val|0x01;

        str=(val&0x01)?str1:str2;

        printf("%s",str);

        outportb(PORT,val);

        gotoxy(1,13);

        printf("%x",val);

        break;


case '2':


        gotoxy(21,8);

        val=(val&0x02)?(val&(~0x02)):val|0x02;

        str=(val&0x02)?str1:str2;

        printf("%s",str);

        outportb(PORT,val);
        gotoxy(1,13);

        printf("%x",val);

        break;


    case '3':


        gotoxy(33,8);

        val=(val&0x04)?(val&(~0x04)):val|0x04;

        str=(val&0x04)?str1:str2;

        printf("%s",str);

        outportb(PORT,val);

        gotoxy(1,13);

        printf("%x",val);

        break;


    case '4':


        gotoxy(45,8);

        val=(val&0x08)?(val&(~0x08)):val|0x08;

        str=(val&0x08)?str1:str2;

        printf("%s",str);

        outportb(PORT,val);

        gotoxy(1,13);

        printf("%x",val);

        break;
```

```
case '5':

        gotoxy(9,9);
        val=(val&0x10)?(val&(~0x10)):val|0x10;
        str=(val&0x10)?str1:str2;
        printf("%s",str);
        outportb(PORT,val);
        gotoxy(1,13);
        printf("%x",val);
        break;

case '6':

        gotoxy(21,9);
        val=(val&0x20)?(val&(~0x20)):val|0x20;
        str=(val&0x20)?str1:str2;
        printf("%s",str);
        outportb(PORT,val);
        gotoxy(1,13);
        printf("%x",val);
        break;

case '7':
```

```
            gotoxy(33,9);

            val=(val&0x40)?(val&(~0x40)):val|0x40;

            str=(val&0x40)?str1:str2;

            printf("%s",str);

            outportb(PORT,val);

            gotoxy(1,13);

            printf("%x",val);

            break;


    case '8':

            gotoxy(45,9);

            val=(val&0x80)?(val&(~0x80)):val|0x80;

            str=(val&0x80)?str1:str2;

            printf("%s",str);

            outportb(PORT,val);

            gotoxy(1,13);

            printf("%x",(unsigned char)val);

            break;


            }


}



}
```

Appendix B

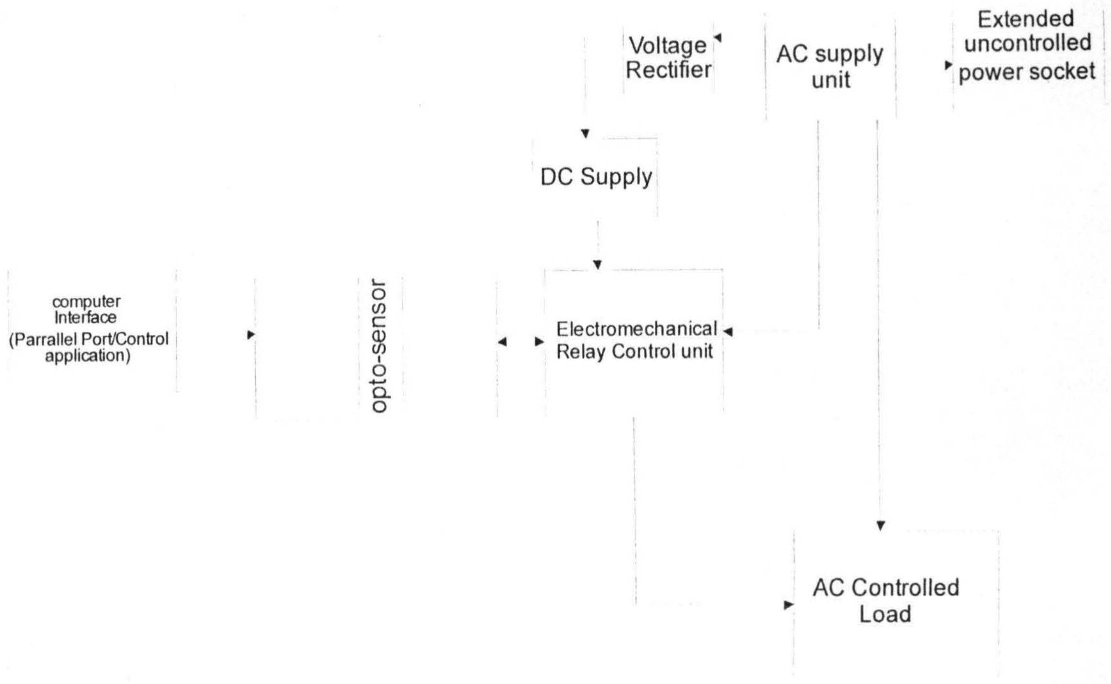

Voltage Rectifier
AC supply unit
Extended uncontrolled power socket
DC Supply
computer Interface (Parrallel Port/Control application)
opto-sensor
Electromechanical Relay Control unit
AC Controlled Load

Fig 5.1 The 8-way Home Automation Block Diagram

Appendix C

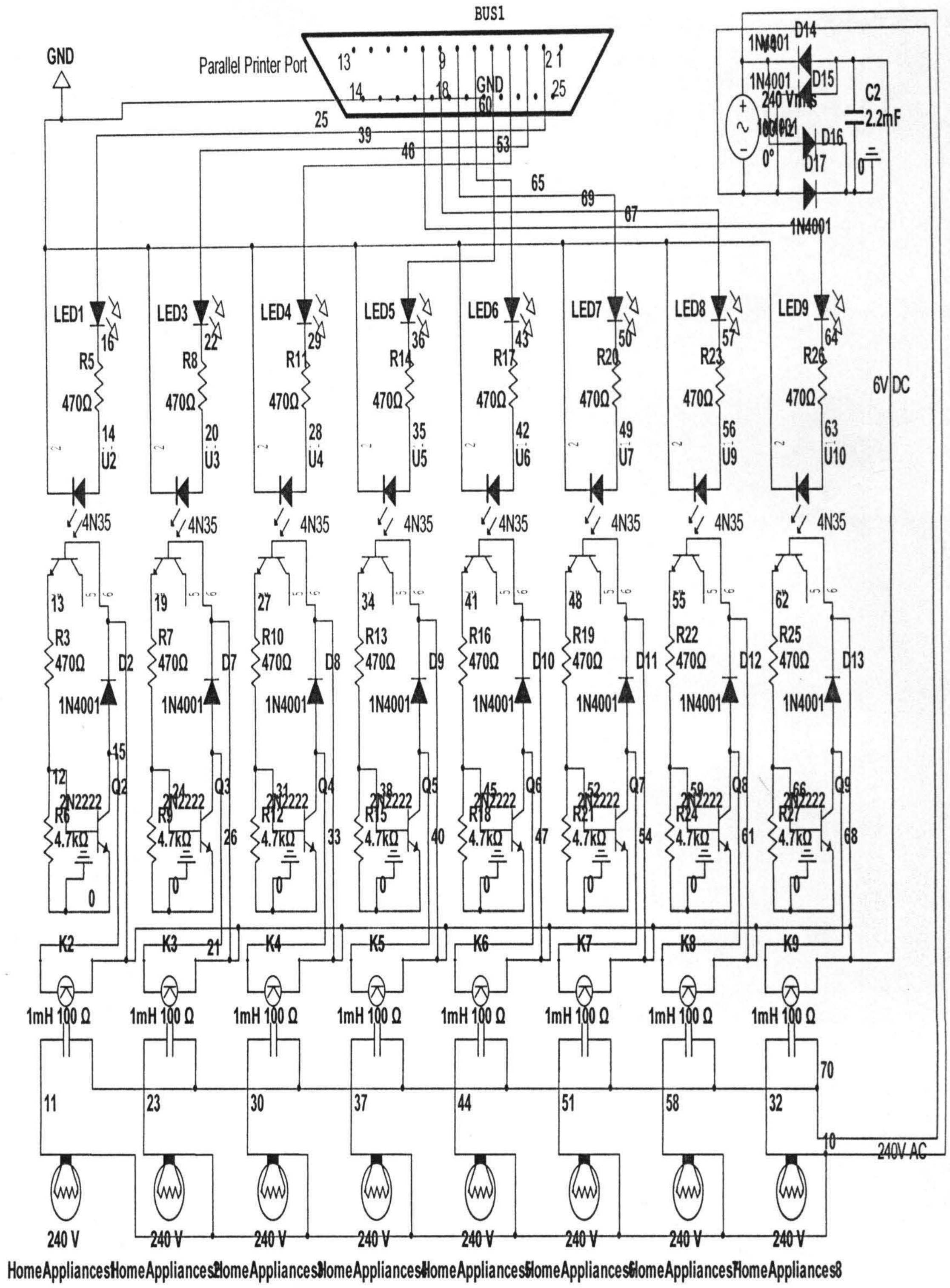


Fig 5.2 The Complete 8-ways Home Automation Circuit