TIME OPERATED WIRELESS ELECTRICAL DEVICE CONTROL

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

in

Electronics and Communication Engineering

BY

AJAY SINGH DHAKA-061013

AKASH AGARWAL-061014

RAKESH NITHARWAL-051046

Under the Supervision of

DR.RAJIV KUMAR

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

MAY- 2010

Jaypee University of Information Technology

Waknaghat, Solan - 173 234, Himachal Pradesh

CERTIFICATE

This is to certify that project report entitled "TIME OPEARATED WIRELESS ELECTRICAL DEVICE CONTROL", submitted by Akash Agarwal, Ajay Singh and Rakesh Nitharwal in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

Date: 20/05/10

Dr. Rajiv Kumar

(Dept.of Electronics and Communication)

Certified that this work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma

1. Ajay Singh Dhaka, 061013

2. Akash Agarwal, 061014

3. Rakesh Nitharwal, 051046

ACKNOWLEDGEMENT

It gives us great pleasure in acknowledging the invaluable assistance extended to us by various personalities in the successful completion of this project. This project is the result of the pieces of advice of many people.

We are highly thankful to our mentor Dr. Rajiv Kumar for his guidance and constructive criticism all during the making of this project.He pacified our intense times with his creative and innovative ideas, reinforcing our interest in the work.

Finally we would like to thank all our friends for the support they gave us from time to time.

DATE: 20/05/10

Ajay Singh
(061013)

Akash Agarwal
(061014)

Rakesh Nitharwal
(051046)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

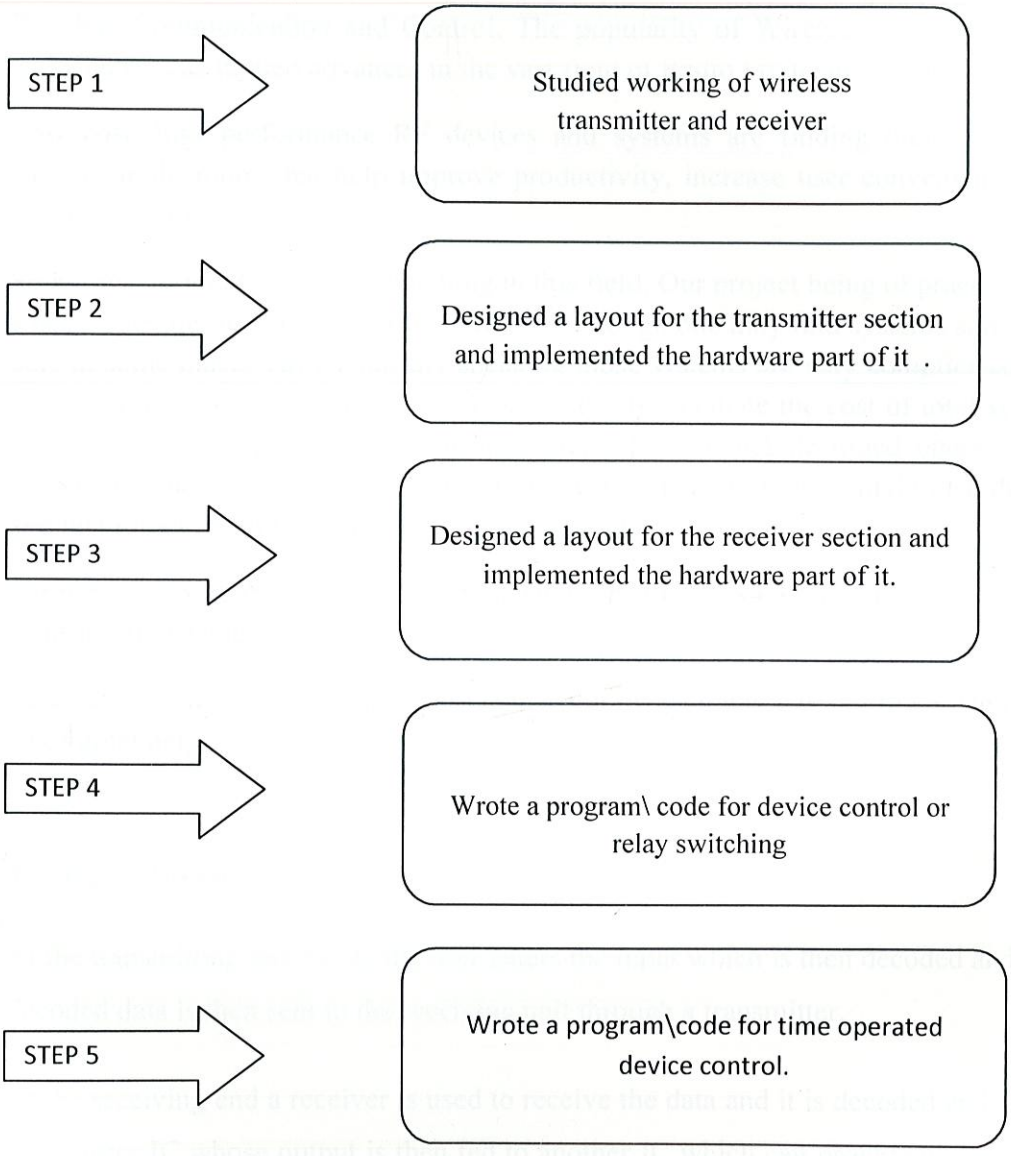
"Time Operated Wireless Electrical Device Control" is a small step in the direction of automation of our surroundings i.e. the place of our physical presence be it our home, office, laboratory etc. which is a very vast and specialized field whose constituents vary from place to place and from user to user .One category of such systems provides safety features like smoke and fire detection ,intrusion prevention etc. the other being totally focused on luxury i.e. automatic door control , automatic climate control etc. and the last been control of electrical appliances.

In our project we have taken the last case i.e. controlling the electrical devices surrounding us from one convenient place so as to make our lives more comfortable. These devices can range from small loads like bulbs etc. to higher current requiring devices .To provide safety features we have designed a thermostat in our system . It also helps in reducing electricity consumption.

We have included timed operation feature in the system using which one can easily set the time at which he wants the device to be on and off thus one need not to keep a watch on the clock every time .

PROJECT FLOW

The project idea has evolved over a period of time and can be divided into a number of steps or phases .These steps denotes the direction in which our team has worked and reached our goal.



CHAPTER-1

INTRODUCTION

With the advancement in technology and with each and every device becoming cheaper, smaller and faster one can easily say that the present age is the age of Wireless Communication and Control. The popularity of Wireless Communication and Control has fuelled advances in the vast field of Radio Frequency technology.

Low cost, high-performance RF devices and systems are finding their way into various applications that help improve productivity, increase user convenience and enhance security.

So we also thought of doing something in this field. Our project being of practical has a large potential in various sectors in India. Although one may find systems similar to ours in some places but technically speaking those systems are very complicated and also are very costly. But in our case one can easily calculate the cost of total system just on his fingertips .We have further extended it to include timed operation of devices i.e. automatic control ,in which one can enter the time domain of a device and just forget about the device.

Further our system can also be upgraded to include another feature like Energy Consumption Meter.

At a time when our energy needs are increasing by each day passing this system can be of great help.

1.1 Description

At the transmitting end firstly the user enters the input which is then decoded and this decoded data is then sent to the receiving unit through a transmitter.

At the receiving end a receiver is used to receive the data and it is decoded and fed to an inverter IC whose output is then fed to another IC which can provide higher level of currents at the output. The outputs from this IC drives the relays and hence control the devices we want to control.

To make the whole system user friendly a code in C language is used and this code enables the user to control whole of the system just by click of a button. Finally another code written in C language will make the system time operated.

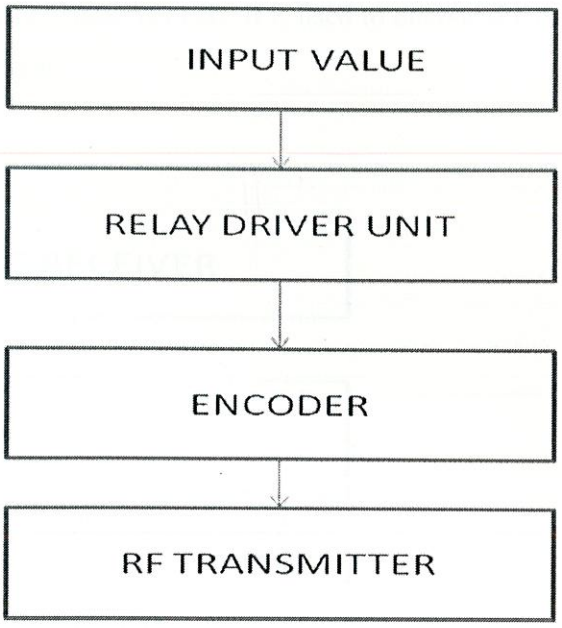


FIGURE-1.1 Block Diagram of Transmitter

1.1.1 Transmitter

Transformer

We have used a step down transformer so as to step down the input voltage from supply socket i.e. 220V AC down to the required value.

Rectifier

We need to convert our AC supply into DC because our system is working on that we are using a Full Wave Rectifier to perform this operation.

RC Filter

To reduce the ripples in our output from Rectifier we are using this filter.

Relay Driver Unit

We will be giving input at transmitter through our parallel port. This port however cannot supply large voltages so as to drive heavier loads we are using this.

Encoder

This is a 2^{12} series encoder HT12E. It is used to encode the data input by the user and is sent to the receiver.

1.1.2 Receiver

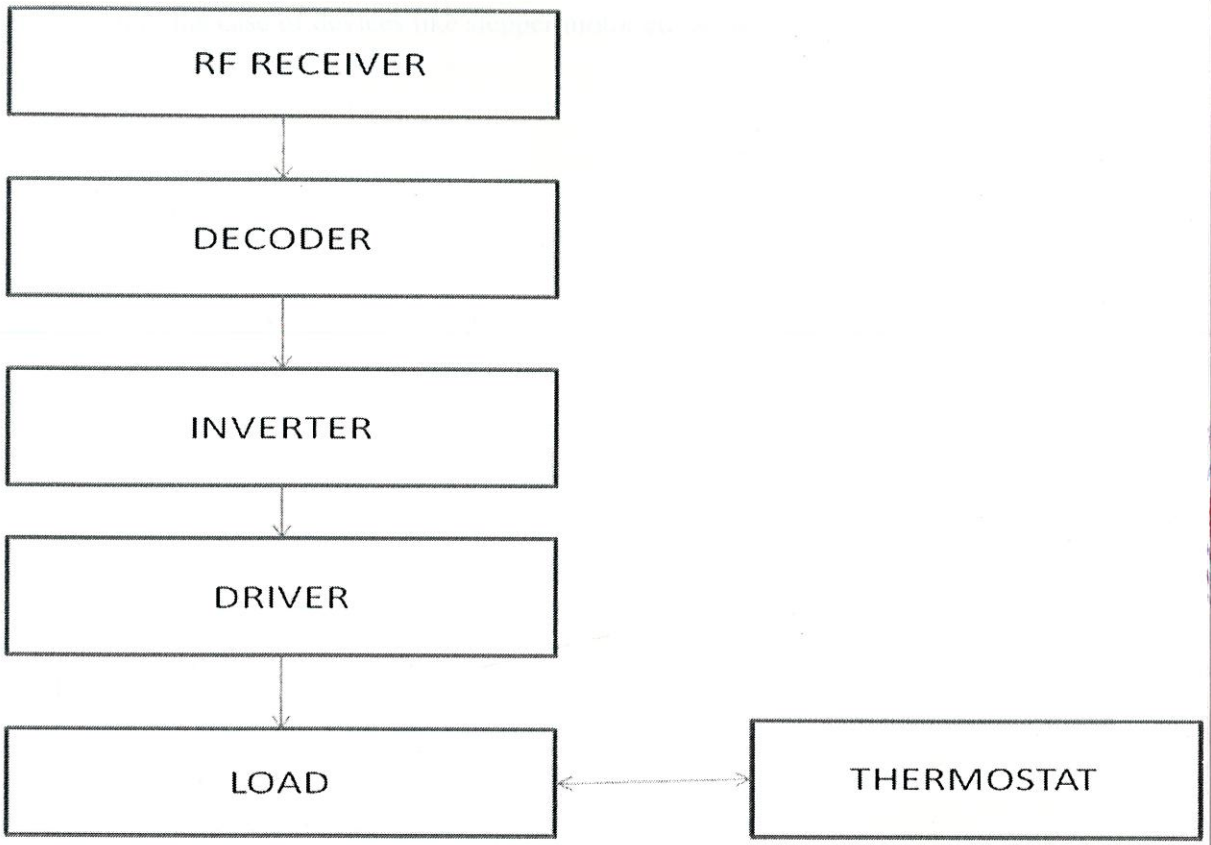


FIGURE-1.2 Block Diagram Of Receiver

Decoder

This is a 2^{12} series decoder HT12D. It is used to decode the data which is being received by the RF receiver.

Inverter

Another IC which acts as an inverter is used and is fed by the outputs from the HT12D.

Driver

In order to provide higher levels of current at the output and to enable our system to work even in the case of devices like stepper motor etc we are using this IC

CHAPTER 2

HARDWARE IMPLEMENTATION

Project implementation has been done in two parts:
1. Transmitter Section.
2. Receiver Section.

2.1 TRANSMITTER SECTION

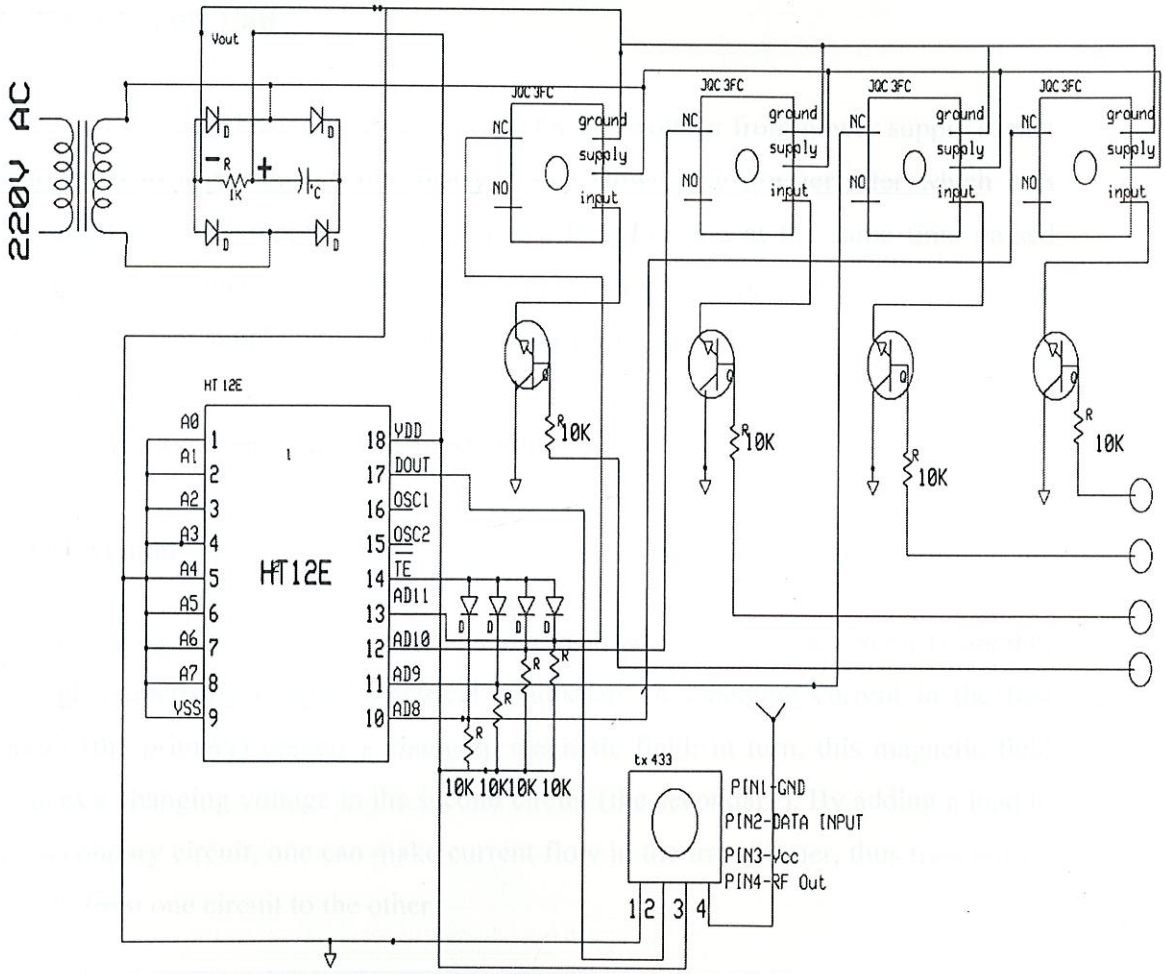


FIGURE-2.1 Transmitter Section

In our transmitter section we have basically following units:

1. Power Supply Unit
2. Relay Driver Unit
3. Encoder Unit
4. RF Transmitter Unit

We will now discuss these components in detail.

1. Power Supply Unit

In this unit what is basically done is firstly the AC voltage from power supply socket is lowered to a particular value using a Step Down Transformer after which it is converted into DC supply via a Full Wave Rectifier and at the same time passed through an RC Filter so as to remove the ripples from the supply.

Thus, our power supply unit consists of two parts

1. Transformer
2. Full wave Rectifier along with an RC Filter

1. Transfomer

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled electrical conductors. A changing current in the first circuit (the primary) creates a changing magnetic field; in turn, this magnetic field induces a changing voltage in the second circuit (the secondary). By adding a load to the secondary circuit, one can make current flow in the transformer, thus transferring energy from one circuit to the other.

Transformers are some of the most efficient electrical 'machines' with some large units able to transfer 99.75% of their input power to their output. Transformers come in a range of sizes from a thumbnail-sized coupling transformer hidden inside a stage microphone to huge units weighing hundreds of tons used to interconnect portions of national power grids. All operate with the same basic principles, though a variety of

designs exist to perform specialized roles throughout home and industry. In our case we are basically using a step down transformer with turns ratio 15:1

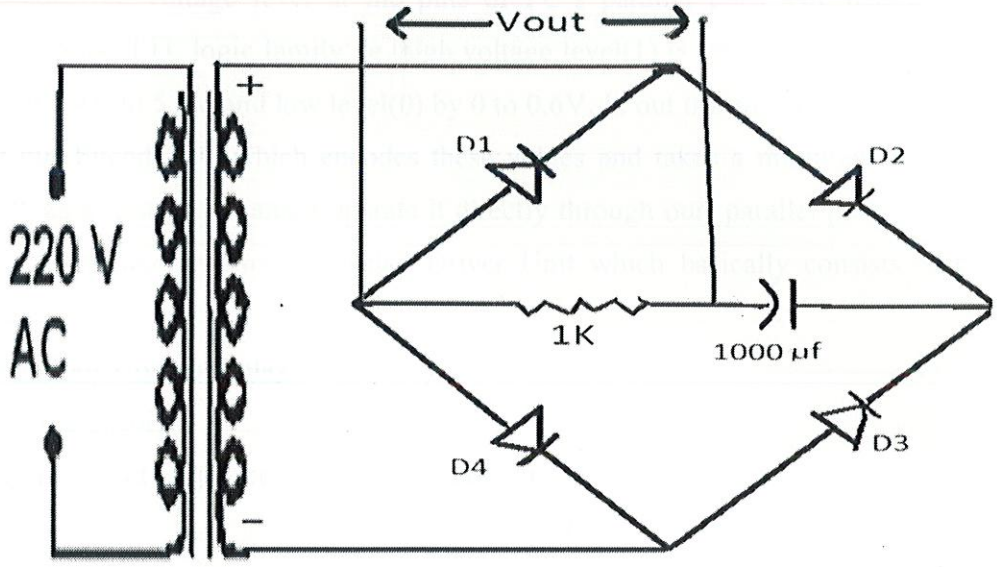


FIGURE 2.2 Power Supply Unit

2. Full Wave Rectifier along with an RC Filter

Firstly when the positive cycle comes diodes D2 and D4 are conducting. Similarly during the negative cycle diodes D3 and D1 are conducting .Thus as a result the input supply which was AC is now converted into a DC supply. The RC filter in between the bridge will reduce the ripples to a considerable level.

2. Relay Driver Unit

We will be giving inputs i.e. whether we wish to give the command to On or Off a particular device will be done through push of a button. This button is basically controlling the voltage level at the pins of PC's parallel port. The parallel port belongs to the TTL logic family i.e. high voltage level(1) is represented by a voltage range of 2.4V to 5.0V and low level(0) by 0 to 0.6V. In out transmitter section we are using our Encoder IC which encodes these values and takes a minimum of 9V as logic 0, as a result we cannot operate it directly through our parallel port. To solve this problem we have used a Relay Driver Unit which basically consists of two devices

1. Solid State Contact Relay
2. NPN transistor

In this we would only focus on discussion of contact relays

1. Relay

Firstly we will look upon some basic features of relay.

A relay is an electrical switch that opens and closes under the control of another electric circuit. Because a relay is able to control an output circuit of higher power than the input circuit, it can be considered in broad sense to be a form of electrical amplifier.

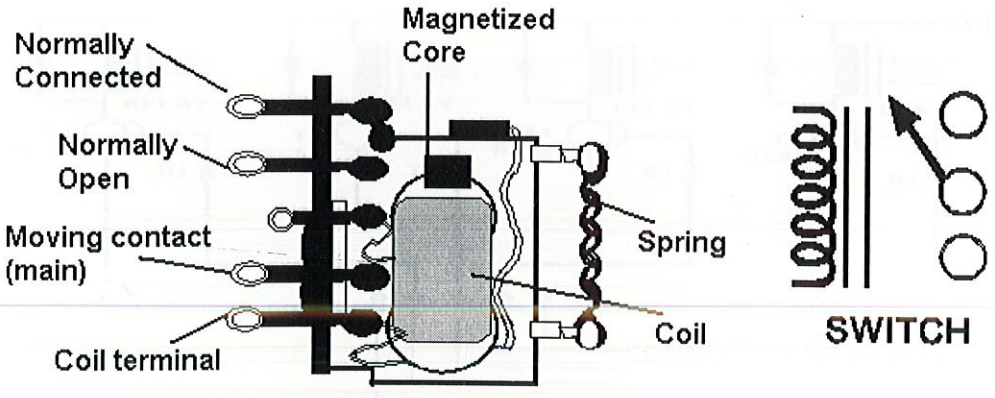


FIGURE 2.3 CONTACT RELAY

These contacts can be either Normally Open (NO), Normally Closed (NC), or change-over contacts.

Normally-open contacts connect the circuit when the relay is activated; the circuit is disconnected when the relay is inactive. It is also called Form A contact or "make" contact. Form A contact is ideal for applications that require to switch a high-current power source from a remote device.

Normally-closed contacts disconnect the circuit when the relay is activated; the circuit is connected when the relay is inactive. It is also called Form B contact or "break" contact. Form B contact is ideal for applications that require the circuit to remain closed until the relay is activated.

Change-over contacts control two circuits: one normally-open contact and one normally-closed contact with a common terminal. It is also called Form C contact

Now we will look upon the working of this unit.When the input from the parallel port is low then input voltage in that case will be 0 to 0.6V in that case the transistor will be in cutoff stage hence relay will not be energized.

When the input from port driver is 1 then at that time the value of the voltage will range from 2.4V to 5V.We want our transistor to go into saturation at the time when input is high for proper working of this unit as a switch.

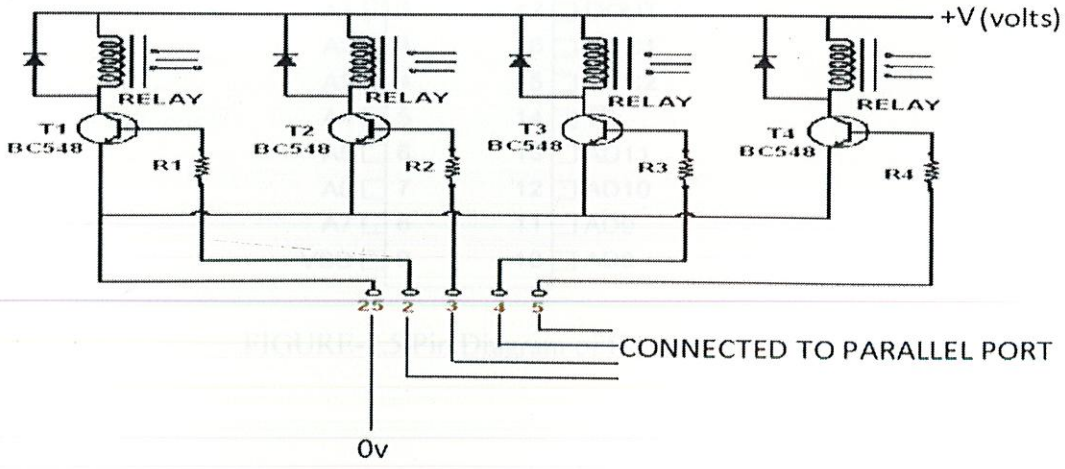


FIGURE 2.4 Relay Driver Unit

3. Encoder Unit (HT12E IC)

Features

1. Operating voltage 2.4V~12V.
2. Low power and high noise immunity CMOS technology.
3. Minimum 4 transmission words.
4. Built-in oscillator needs only 5% resistor.
5. Data code has positive polarity.
6. Minimal external components.

General Description

The 2_{12} encoders are a series of CMOS LSIs for remote control system applications. They are capable of encoding information which consists of N address bits and 12_N data bits. Each address/data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header bits via an RF or an infrared transmission medium upon receipt of a trigger signal. The capability to select a TE trigger on the HT12E further enhances the application flexibility of the 2_{12} series of encoders.

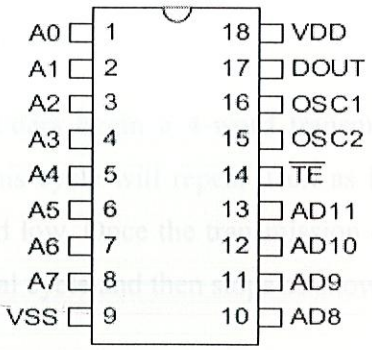


FIGURE-2.5 Pin Diagram of HT12E

TABLE-2.1 Pin Description of HT12E

Pin Name	I/O	Internal Connection	Description
A0~A7	I	CMOS IN Pull-high (HT12A)	Input pins for address A0~A7 setting These pins can be externally set to VSS or left open
		NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	
AD8~AD11	I	NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address/data AD8~AD11 setting These pins can be externally set to VSS or left open
D8~D11	I	CMOS IN Pull-high	Input pins for data D8~D11 setting and transmission enable, active low These pins should be externally set to VSS or left open (see Note)
DOUT	O	CMOS OUT	Encoder data serial transmission output
L/MB	I	CMOS IN Pull-high	Latch/Momentary transmission format selection pin: Latch: Floating or VDD Momentary: VSS

Functional Description

The 212 series of encoders begin a 4-word transmission cycle upon receipt of a transmission enable. This cycle will repeat itself as long as the transmission enable (TE or D8~D11) is held low. Once the transmission enable returns high the encoder output completes its final cycle and then stops as shown below.

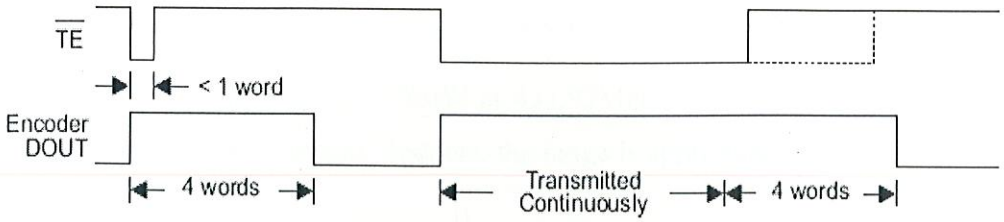


FIGURE-2.6 Functional Cycle

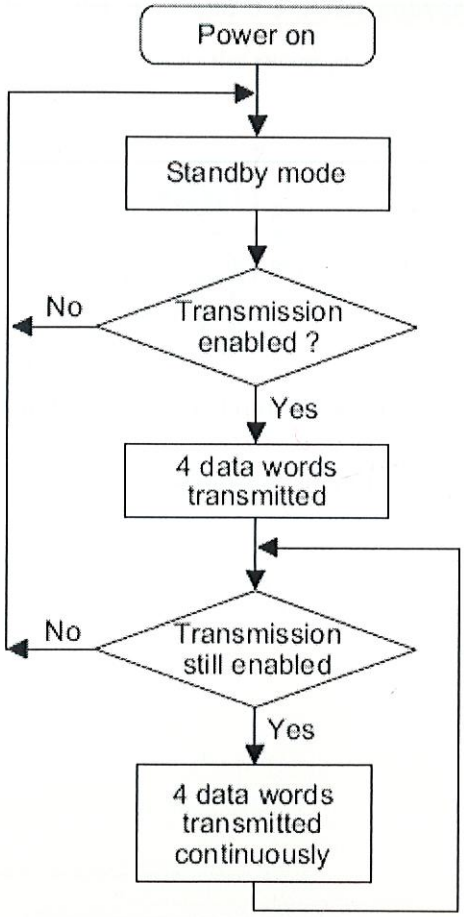


FIGURE-2.7 Flowchart for working of HT12E

4. RF Transmitter Unit

This unit comprises of TX-433 which is basically a Radio Frequency Transmitter

The transmitter output is up to 8mW at 433.92MHz with a range of approximately 400 foot (open area) outdoors. Indoors, the range is approximately 200 foot.

The TX-433 transmitter accepts both linear and digital inputs, can operate from 1.5 to 12 Volts-DC, and makes building a miniature hand-held RF transmitter very easy. The TX-433 is approximately the size of a standard postage stamp. It uses Amplitude Shift Keying method to denote 1 and 0's .After converting the incoming serial data into this form it transmit the data to the receiver .

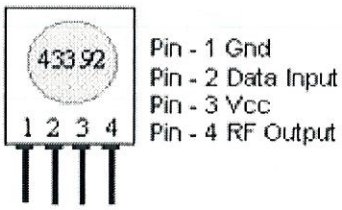


FIGURE-2.8 RF Transmitter pin diagram

So this ends the description of Transmitter section of our project.

2.2 RECEIVER SECTION

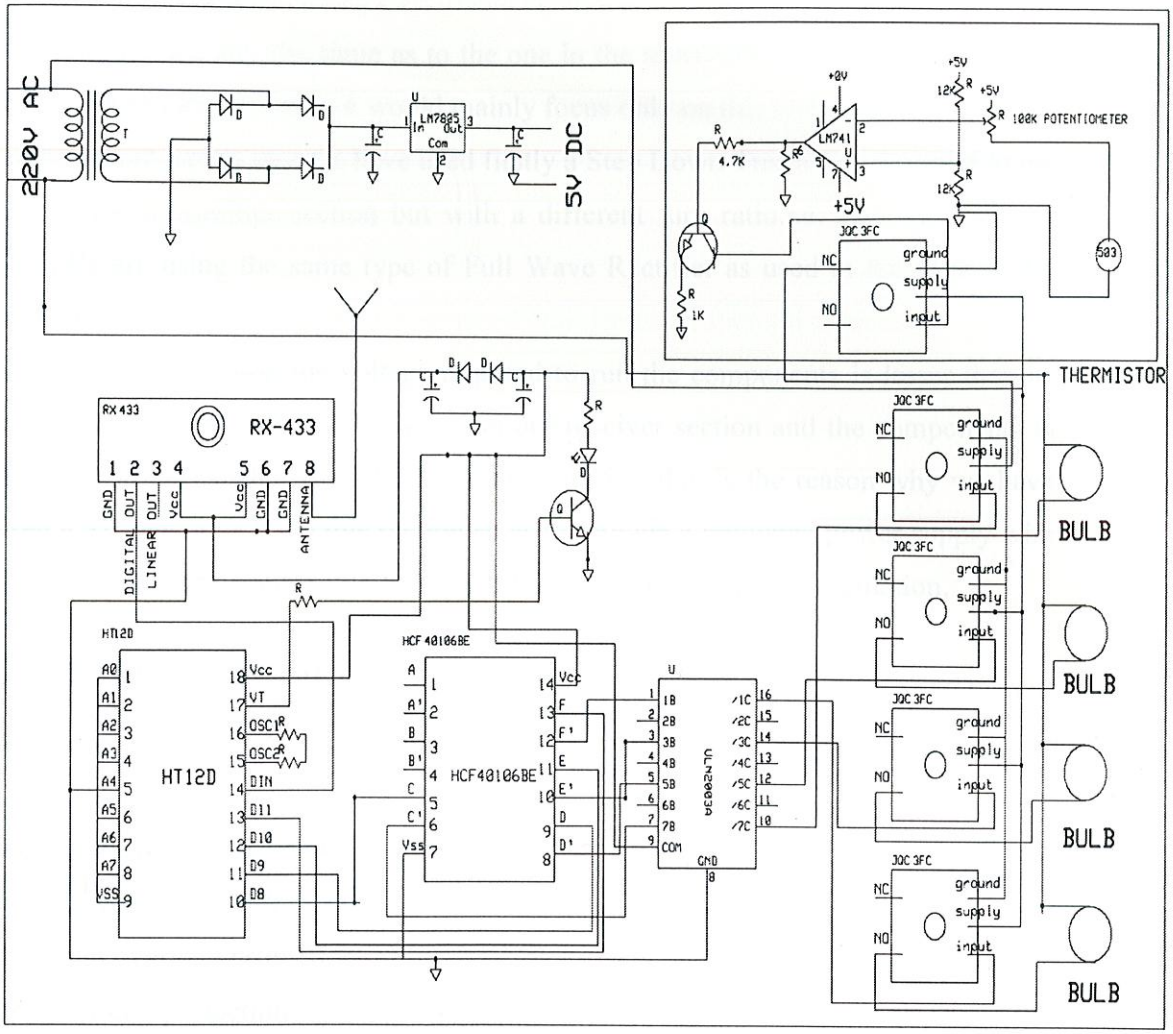


FIGURE-2.9 Receiver Schematic

In our receiver section we have basically following components:

1. Power Supply Unit (U7805 IC)
2. Decoder Unit (HT12D IC)
3. Inverting Unit (HCF40106BE IC)
4. Driver Unit (ULN2003 IC)
5. RF Receiver Unit
6. Thermostat

We will now discuss these components in detail

1. Power Supply Unit

This unit is basically the same as to the one in the transmitter except one device i.e. Voltage Regulator IC and we would mainly focus only on this under this unit.

In the power supply unit we have used firstly a Step Down Transformer similar to the one in the transmitter section but with a different turn ratio.i.e. approximately an 18:1.We are using the same type of Full Wave Rectifier as used in the Transmitter section.

In the receiver section the voltage required to run the components is lesser than its counterpart i.e. we just need +5V to run our receiver section and the components in this section are costlier than the transmitter section that is the reason why we have used a 5V regulator IC in this section so as to provide a regulated power supply with approximately 0% ripples .We have used U7805 IC as for voltage regulation,

U7805 (+5V regulator IC)

Features

Output current up to 1.5 A
Output voltages of 5 V
Thermal overload protection
Short circuit protection
Output transition SOA protection
2 % output voltage tolerance
Guaranteed in extended temperature range

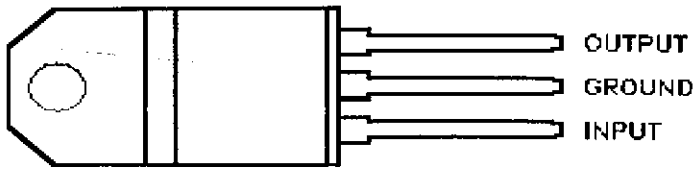


FIGURE 2.10 PIN DIAGRAM OF 7805

General Description

The L78xx series of three-terminal positive regulators is available in TO-220, TO-220FP, TO-3, D²PAK and DPAK packages and several fixed output voltages, making it useful in a wide range of applications. These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltage and currents.

TABLE 2.2 Absolute Maximum Ratings of U7805

Parameter	Value
Input voltage	7Vto 35V
Output current	internally limited
Power dissipation	internally limited
Storage temperature range	208 to 423k
Operating Junction temperature range	210 to 423k
DC output voltage	4.8 to 5.2V

Application information

The 7805 series of fixed voltage regulators are designed with thermal overload protection that shuts down the circuit when subjected to an excessive power overload

condition. internal short-circuit protection that limits the maximum current the circuit will pass, and output transistor safe-area compensation that reduces the output short-circuit current as the voltage across the pass transistor is increased. In many low current applications, compensation capacitors are not required. However, it is recommended that the regulator input be bypassed with capacitor connected to the if the output load capacitance is large. An input bypass capacitor should be selected to provide good high frequency characteristics to insure stable operation under all load conditions. A 0.33 μF or larger tantalum, mylar or other capacitor having low internal impedance at high frequencies should be chosen. The bypass capacitor should be mounted with the shortest possible leads directly across the regulators input terminals. Normally good construction techniques should be used to minimize ground loops and lead resistance drops since the regulator has no external sense lead.

The addition of an operational amplifier allows adjustment to higher or intermediate values while retaining regulation characteristics. The minimum voltage obtained with the arrangement is 2 V greater than the regulator voltage.

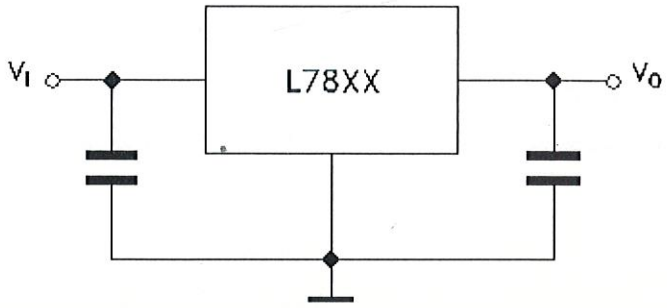


FIGURE 2.11 Application/test circuit for 7805

2. Decoder Unit (HT12D Decoder IC)

Features

1. Operating voltage: 2.4V~12V
2. Low power and high noise immunity CMOS technology
3. Low standby current

4. Capable of decoding 12 bits of information
5. Binary address setting
6. Received codes are checked 3 times
7. Built-in oscillator needs only 5% resistor
8. Easy interface with an RF or an infrared transmission medium
9. Minimal external components
10. Pairs with Holtek 212 series of encoders

General Description

The 212 decoders are a series of CMOS LSIs for remote control system applications. They are paired with Holtek's 212 series of encoders .For proper operation, a pair of encoder/decoder with the same number of addresses and data format should be chosen. The decoders receive serial addresses and data from a programmed 212 series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. They compare the serial input data three times continuously with their local addresses. If no error or unmatched codes are found, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission. The 212 series of decoders are capable of decoding information that consists of N bits of address and 12_N bits of data. Of this series, the HT12D is arranged to provide 8 address bits and 4 data bits, and HT12F is used to decode 12 bits of address information.

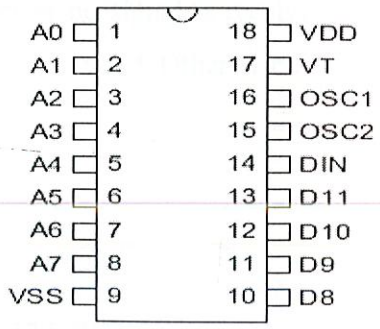


FIGURE 2.12 Pin Diagram of HT12D

TABLE-2.3 Pin Description of HT12D

Pin Name	I/O	Internal Connection	Description
A0~A11 (HT12F)	I	NMOS Transmission Gate	Input pins for address A0~A11 setting These pins can be externally set to VSS or left open.
A0~A7 (HT12D)			Input pins for address A0~A7 setting These pins can be externally set to VSS or left open.
D8~D11 (HT12D)	O	CMOS OUT	Output data pins, power-on state is low.
DIN	I	CMOS IN	Serial data input pin
VT	O	CMOS OUT	Valid transmission, active high
OSC1	I	Oscillator	Oscillator input pin
OSC2	O	Oscillator	Oscillator output pin
VSS	----	----	Negative power supply, ground
VDD	----	----	Positive power supply

Functional Description

The 212 series of decoders provides various combinations of addresses and data pins in different packages so as to pair with the 212 series of encoders. The decoders receive data that are transmitted by an encoder and interpret the first N bits of code period as addresses and the last 12_N bits as data, where N is the address code number. A signal on the DIN pin activates the oscillator which in turn decodes the incoming address and data. The decoders will then check the received address three times continuously. If the received address codes all match the contents of the decoder's local address, the 12_N bits of data are decoded to activate the output pins and the VT pin is set high to indicate a valid transmission. This will last unless the address code is incorrect or no signal is received. The output of the VT pin is high only when the transmission is valid. Otherwise it is always low.

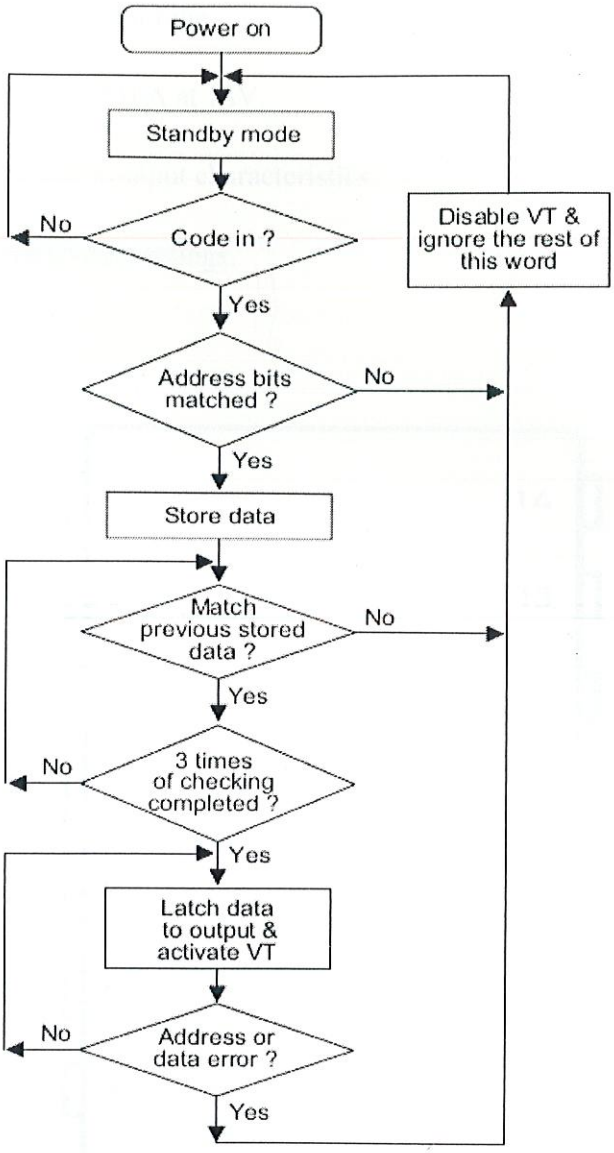


FIGURE-2.13 Flowchart for working of HT12D

3. Inverting Unit (HCF40106BE)

General Description

Schmitt Trigger action with no external component.

Noise immunity greater than 50%.

No limit on input rise and fall time.

Maximum input current of 1uA at 18V.

Standardized symmetrical output characteristics.

5V, 10V and 15V parametric ratings.

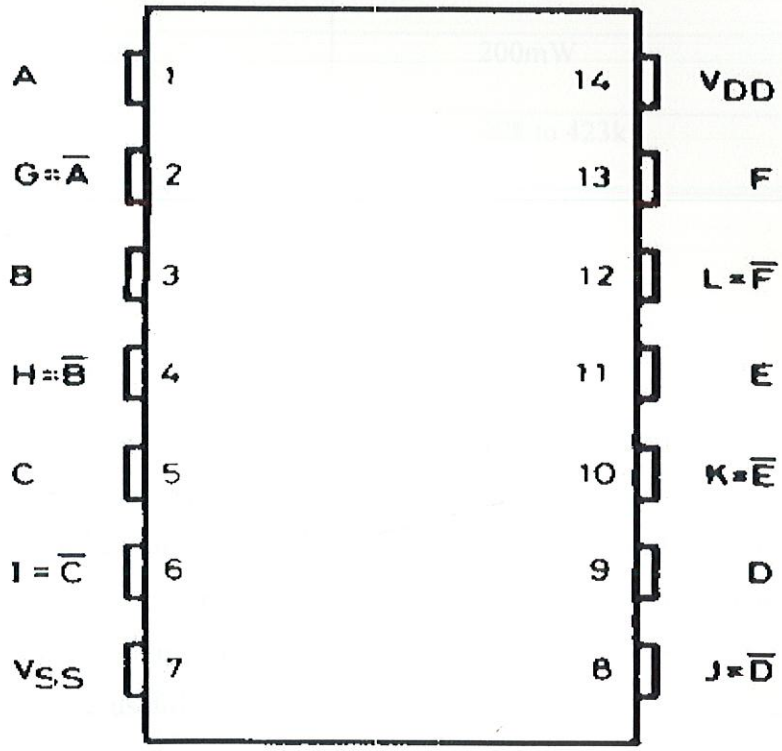


FIGURE 2.14 Pin diagram of HCF40106BE

General Description

The HCF40106BE consists of six Schmitt-trigger circuits. Each circuit functions as an inverter with Schmitt -trigger action on the input. The trigger switches at different points for positive and negative going signals. The difference between positive -going voltage (Vp) and the negative going voltage(Vn) is defined as hysteresis voltage(Vh)

TABLE 2.4 Absolute Maximum Ratings of HCF40106BE

Parameter	Value
Supply Voltage	-0.5 to 20v
Input Voltage	-0.5 to (supply voltage+0.5)V DC
Input Current	± 10Ma
Total Power dissipation	200mW
Storage Temperature	208 to 423k

4. Driver Unit (ULN2003 IC)

General Description

The ULN2003 is a high voltage, high current darlington array containing seven open collector darlington pairs with common emitters. Each channel rated at 500mA and can withstand peak currents of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout. It can interface to all common logic families like TTL/CMOS/PMOS/DTL etc. These versatile devices are useful for driving a wide range of loads including solenoids, relays DC motors, LED displays filament lamps, thermal printheads and high power buffers.

TABLE 2.5 Absolute Maximum Ratings of ULN 2003

Symbol	Parameter	Value
Vo	Output Voltage	50V
Vin	Input Voltage	30V
Ic	Maximum Output Current	600mA
Tamb	Operating Temperature Range	– 20 to 85°C
Tstg	Storage Temperature Range	– 55 to 150°C
Tj	Junction Temperature	150 °C

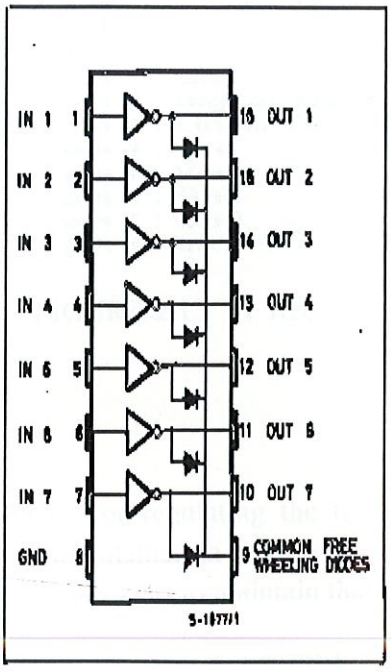


FIGURE 2.15– Circuit Diagram of HCF40106BE

5. RF Receiver Unit

The receiver also operates at 433.92MHz, and has a sensitivity of 3uV. The RX-434 receiver operates from 4.5 to 5.5 volts-DC, and has both linear and digital outputs. For maximum range, the recommended antenna should be approximately 35cm long. To convert from centimeters to inches -- multiply by 0.3937. For 35cm, the length in inches will be approximately 35cm x 0.3937 = 13.7795 inches long.

FIGURE-2.16 RF Receiver

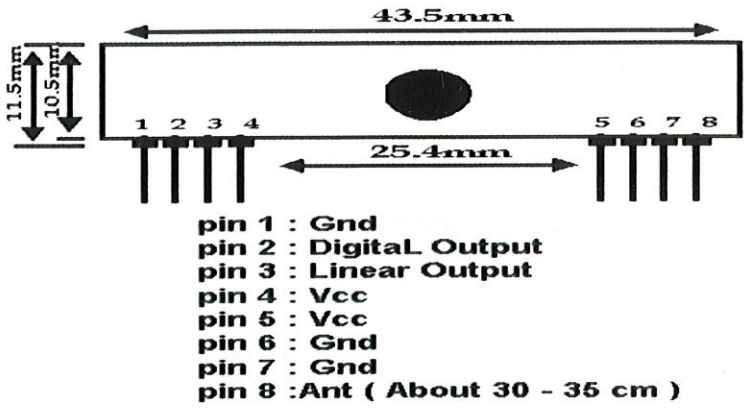


FIGURE-2.17 RF Receiver Pin out

6. Thermostat

A thermostat is a device for regulating the temperature of a system so that the system's temperature is maintained in a particular range. Thus a thermostat switches On or Off a particular device so as to maintain that in the desired range.

In our case our thermostat has a thermistor whose resistance changes with temperature as a result voltage drop also changes .This change in voltage drop causes the device to be On or Off

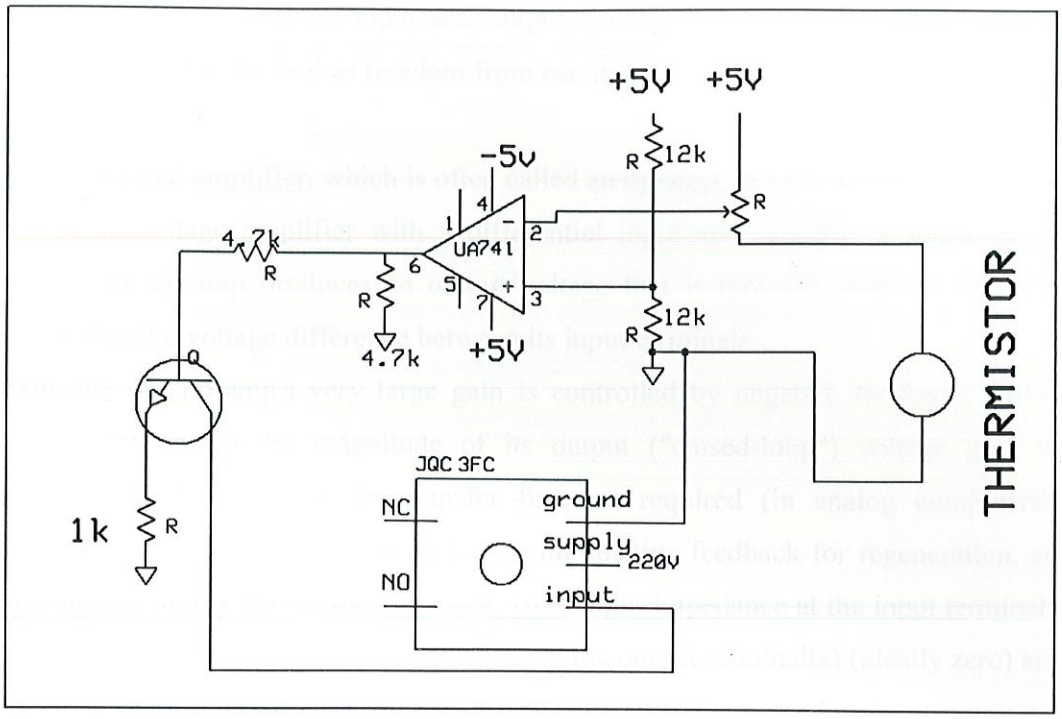


FIGURE 2.18 Schematic of Thermostat Unit

UA 741

TABLE 2.6 Absolute Maximum Ratings of UA741

Parameter	Value
Maximum Input Voltage Range	± 18v
Opearating Temperature Range	(273k to 343k)
Power Dissipation	500 mw
Differential Input Voltage	± 30v
Large Signal Voltage Gain	2,000 to 20,000

General Description

The U741 series are general purpose operational amplifiers which feature improved performance over industry standards like the LM709. They are direct, plug-in replacements for the 709C, LM201, MC1439 and 748 in most applications. The amplifiers offer many features which make their application nearly foolproof:

overload protection on the input and output, no latch-up when the common mode range is exceeded, as well as freedom from oscillation

An operational amplifier, which is often called an op-amp, is a DC-coupled high-gain electronic voltage amplifier with a differential input and, usually, a single-ended output. An op-amp produces an output voltage that is typically millions of times larger than the voltage difference between its input terminals.

Typically the op-amp's very large gain is controlled by negative feedback, which largely determines the magnitude of its output ("closed-loop") voltage gain in amplifier applications, or the transfer function required (in analog computers). Without negative feedback, and perhaps with positive feedback for regeneration, an op-amp essentially acts as a comparator. High input impedance at the input terminals (ideally infinite) and low output impedance at the output terminal(s) (ideally zero) are important typical characteristics

The op-amp is one type of differential amplifier. Other types of differential amplifier include the fully differential amplifier(similar to the op-amp, but with two outputs), the instrumentation amplifier (usually built from three op-amps), the isolation amplifier (similar to the instrumentation amplifier, but with tolerance to common-mode voltages that would destroy an ordinary op-amp) and negative feedback amplifier (usually built from one or more op-amps and a resistive feedback network).

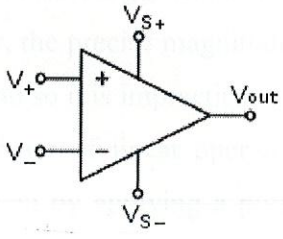


Figure 2.19 Circuit diagram symbol for an op-amp

The circuit symbol for an op-amp is shown to the right, where:

- V_+: non-inverting input
- V_-: inverting input
- V_{out}: output
- V_{S+}: positive power supply
- V_{S-}: negative power supply

The power supply pins (V_{S+} and V_{S-}) can be labelled in different ways (IC power supply_pins). Despite different labelling, the function remains the same — to provide additional power for amplification of signal. Often these pins are left out of the diagram for clarity, and the power configuration is described or assumed from the circuit.

The amplifier's differential inputs consist of V_+input and V_-input, and ideally the op-amp amplifies only the difference in voltage between the two, which is called the differential input voltage. The output voltage of the op-amp is given by the equation,

$$V_{out} = (V_+ - V_-) A_{OL}$$

where $V_+$is the voltage at the non-inverting terminal, V_- is the voltage at the inverting terminal and A_{OL} is the open-loop gain of the amplifier. (The term "open-loop" refers to the absence of a feedback loop from the output to the input.)

The magnitude of A_{OL} is typically very large—seldom less than a million—and therefore even a quite small difference between $V_+$and V_-(a few microvolts or less) will result in amplifier saturation, where the output voltage goes to either the extreme maximum or minimum end of its range, which is set approximately by the power supply voltages. Additionally, the precise magnitude of A_{OL} is not well controlled by the manufacturing process, and so it is impractical to use an operational amplifier as a stand-alone differential amplifier. If linear operation is desired, negative feedback must be used, usually achieved by applying a portion of the output voltage to the inverting input. The feedback enables the output of the amplifier to keep the inputs at or near the same voltage so that saturation does not occur. Another benefit is that if much negative feedback is used, the circuit's overall gain and other parameters become determined more by the feedback network than by the op-amp itself. If the feedback network is made of components with relatively constant, predictable, values

such as resistors, capacitors and inductors, the unpredictability and inconstancy of the op-amp's parameters (typical of semiconductor devices) do not seriously affect the circuit's performance.

If no negative feedback is used, the op-amp functions as a switch or comparator.Here we want our opamp to act as a comparator so we are using this in open loop mode.

Choosing variable resistance value to set the thermostat at approx 310k. We need to set variable resistance value for deciding our operation range.

We have a formula:

B= ln(R'-R)/(1/T' – 1/T)

R= resistance at room temp.(298.15K) = 50k

R'= resistance at temp. T'

T= room temp.

B= constant and in our case = 4200

So if I need to set my cutoff temp. to be 310k

R' will come out to be 30k.

Now from the circuit voltage at pin3 = 2.5V

Current through thermistor(at 310k temp.)

= 5V/(30k+100k)

= 0.0384615mA

Whenever the voltage at pin2 will drop below 2.5V output will be high and relay will be

energized and bulb will be switched off. Let us suppose potentiometer provides 35k

Then voltage at pin2 will drop down to less then 2.5V at approx. 37degrees hence thermostat will cut the supply. So this ends the description of Receiver section of our project.

CHAPTER-3
TESTING AND SOFTWARE IMPLEMENTATION

3.1 Transmitter

The data sent to microcontroller is processed and sent to encoder and the HT12E send this data to the RF transmitter. We have checked manually the working of the transmitter by the following circuit diagram

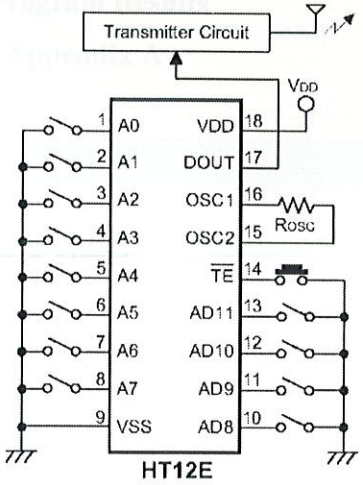


FIGURE-3.1 Testing of Encoder circuit

3.2 Receiver

The data received by the RF receiver is transmitted to the decoder HT12D. So we have checked the working of the encoder and decoder by directly connecting the wires as shown in the diagram

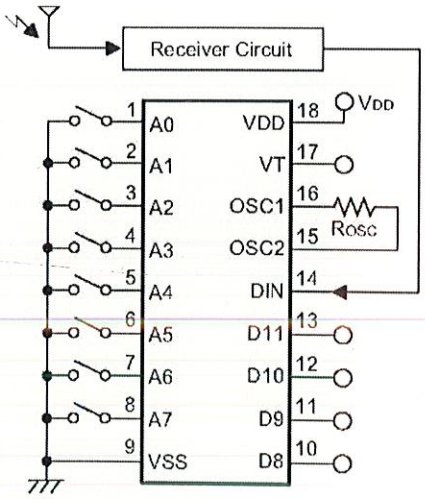


FIGURE- 3.2 Testing of Decoder circuit

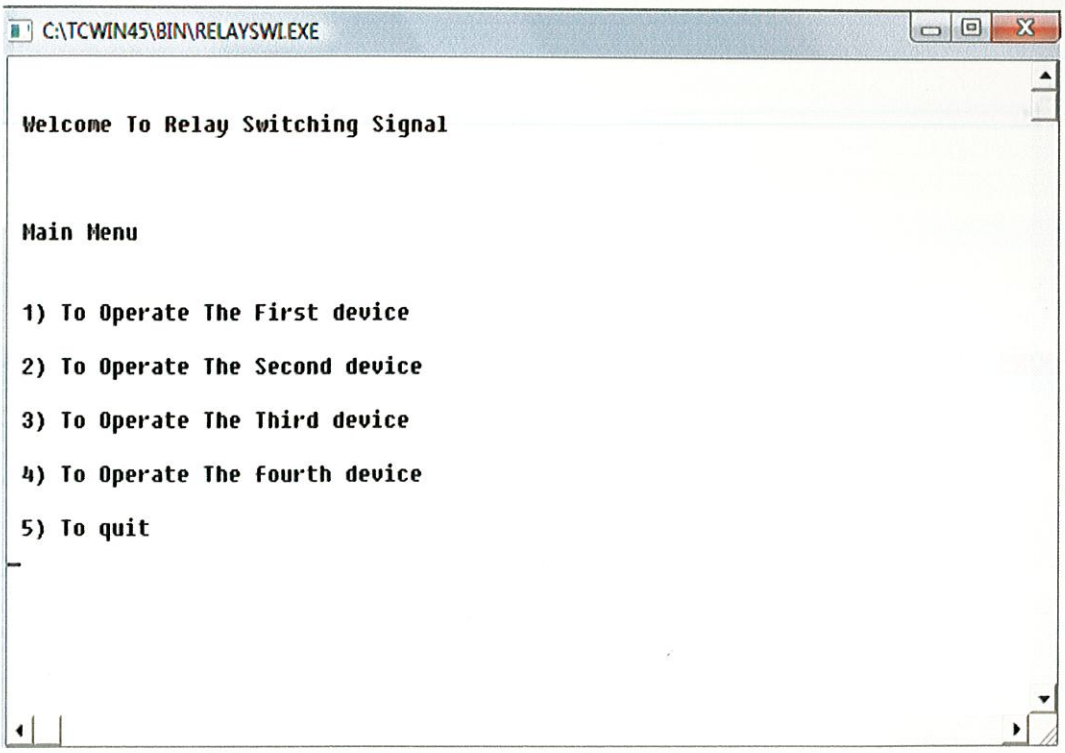
3.3 Software

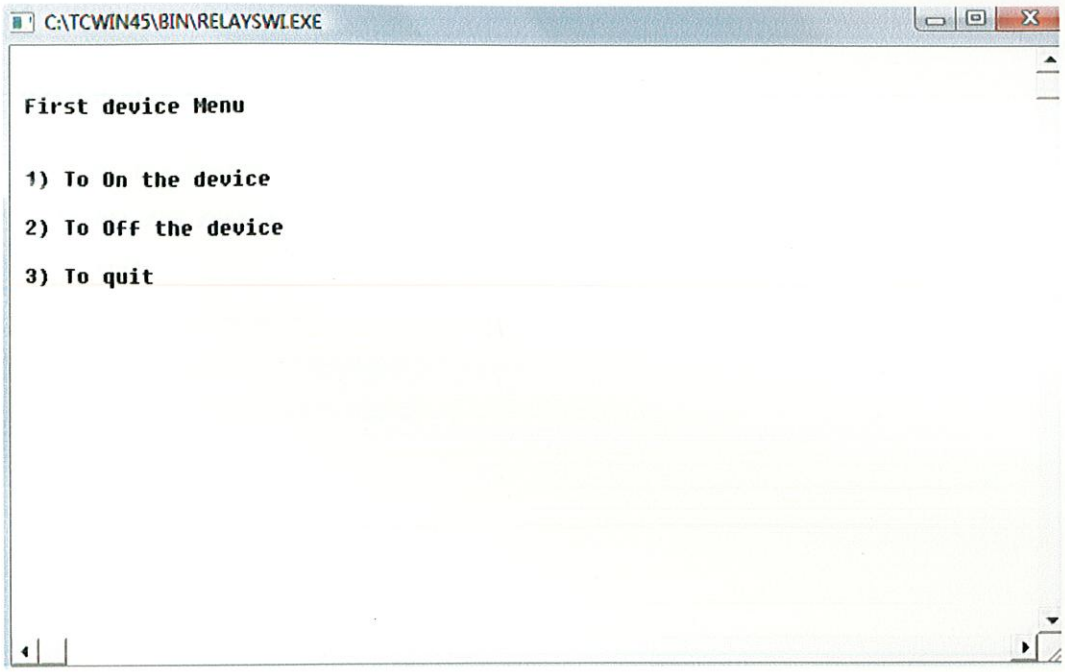
The software design is a key element in the development of a project. For controlling the voltage at the parallel port pins we have designed the first code. This code acts as an interface between the user and the hardware.

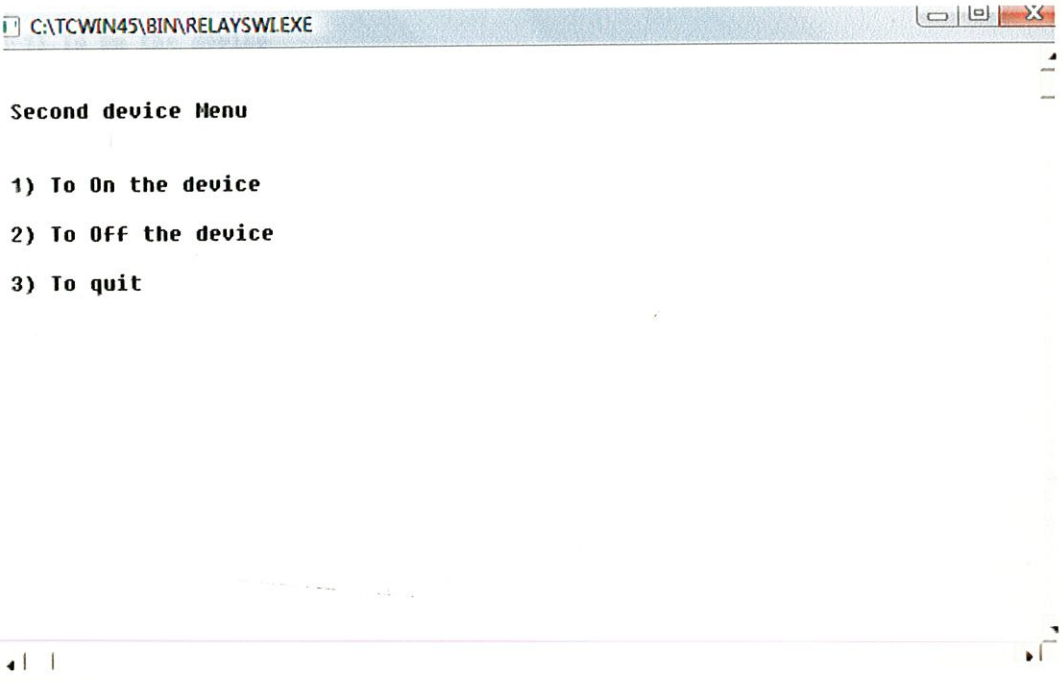
The second code provides timed operation feature in the system i.e. one can input the time at which he wants the device to on and off.

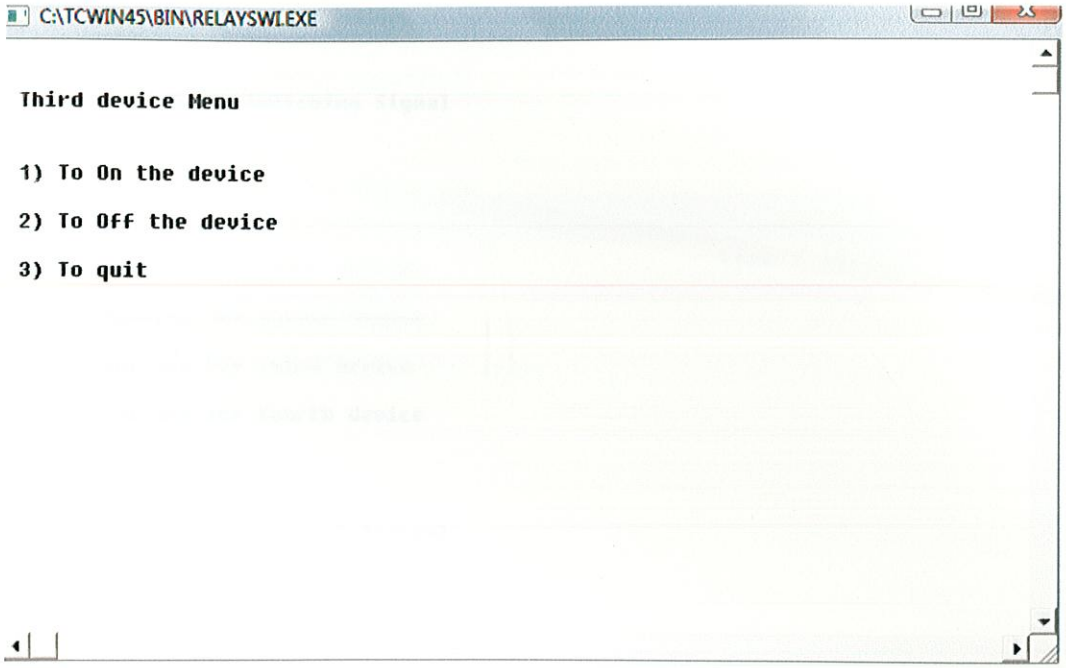
3.3.1 Relay Switching Program Results

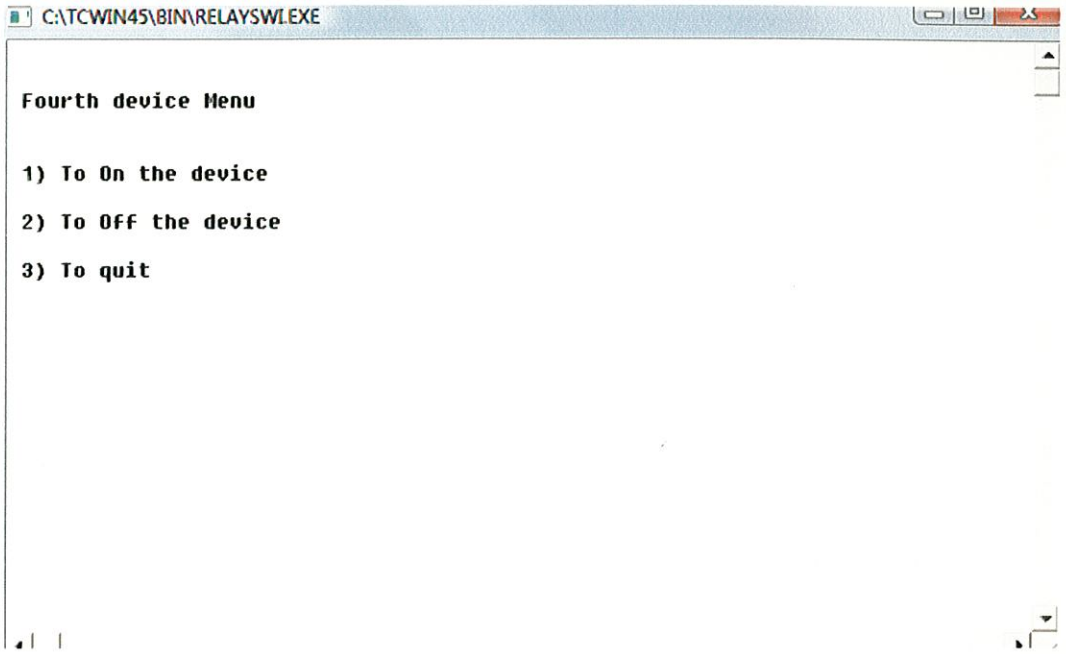
For detailed code Refer Appendix A

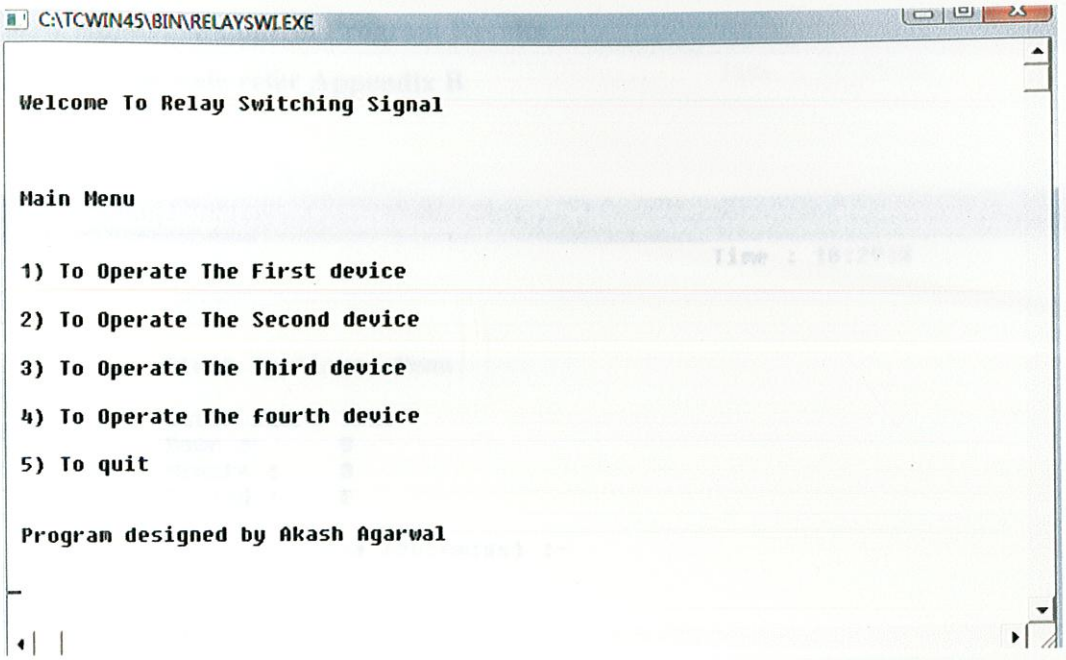



C:\TCWIN45\BIN\RELAYSWI.EXE
First device Menu
1) To On the device
2) To Off the device
3) To quit


C:\TCWIN45\BIN\RELAYSWI.EXE
Second device Menu
1) To On the device
2) To Off the device
3) To quit

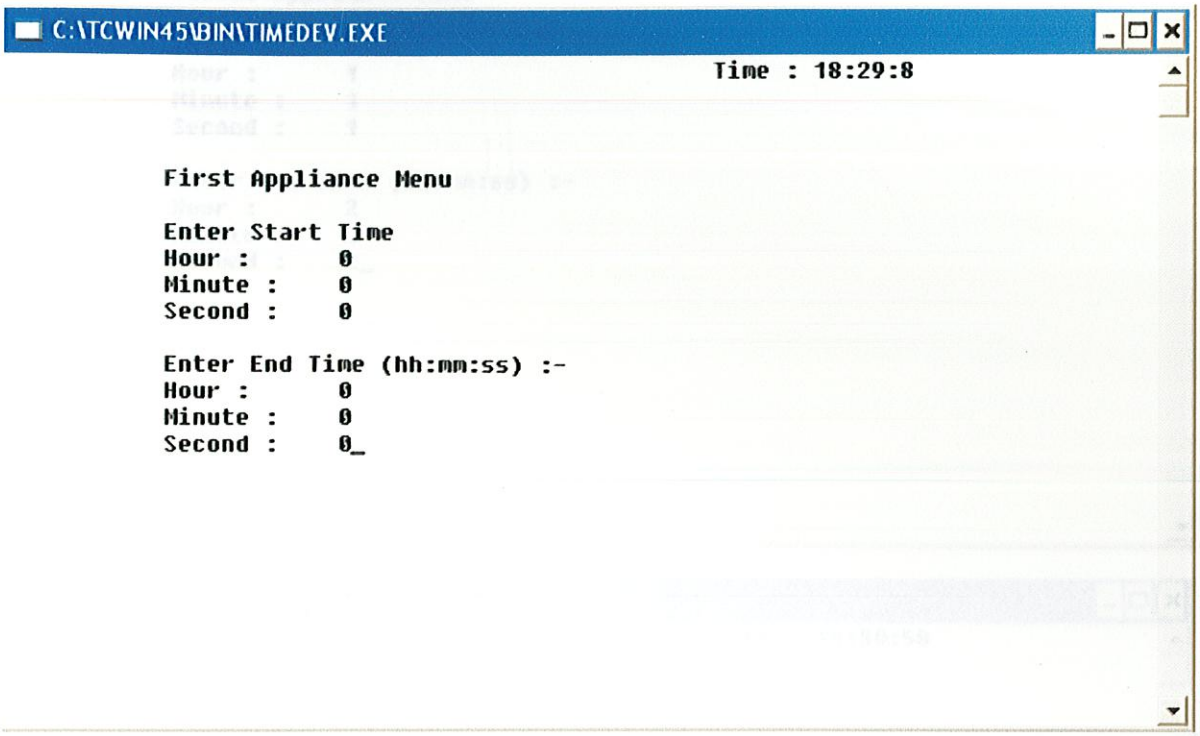

C:\TCWIN45\BIN\RELAYSWI.EXE
Third device Menu
1) To On the device
2) To Off the device
3) To quit

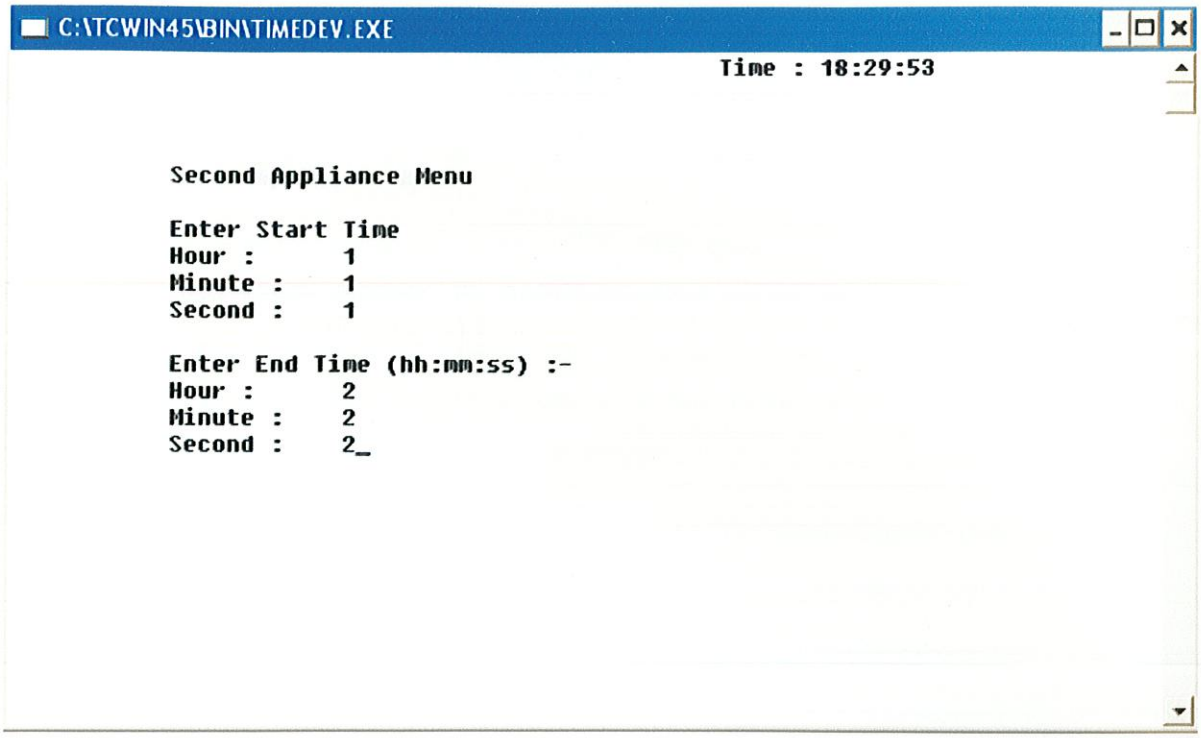

C:\TCWIN45\BIN\RELAYSWI.EXE
Fourth device Menu
1) To On the device
2) To Off the device
3) To quit

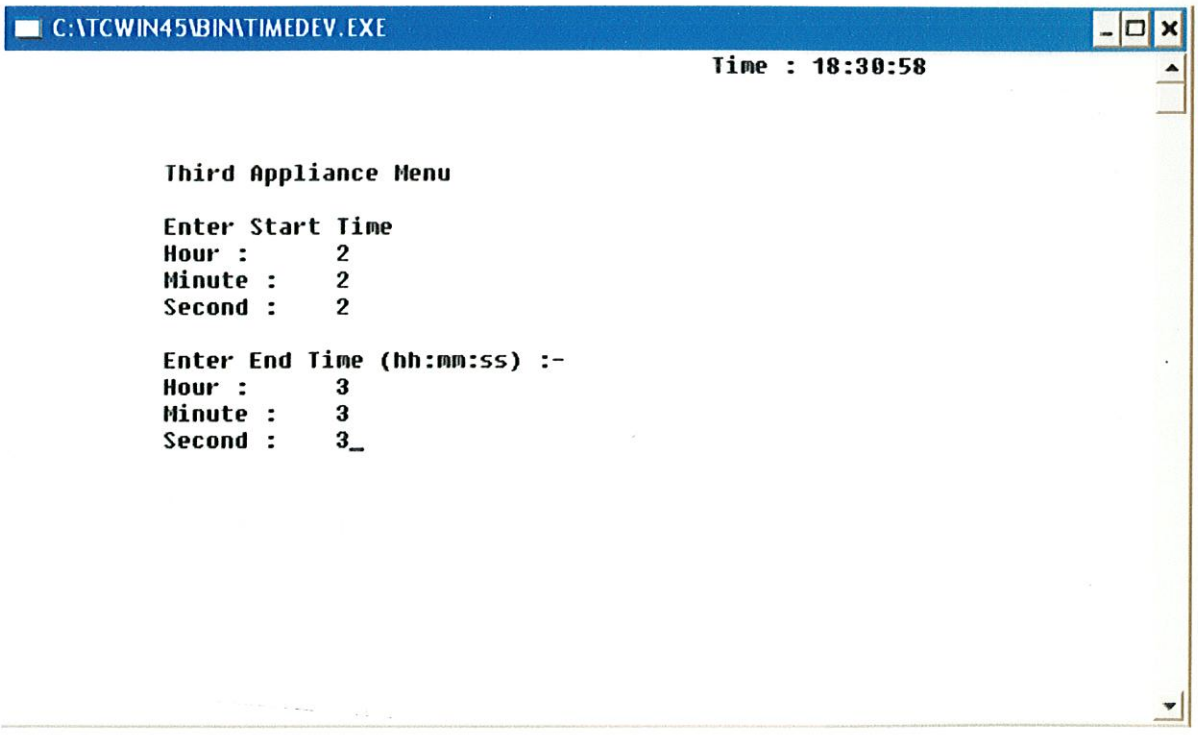

C:\TCWIN45\BIN\RELAYSW.EXE
Welcome To Relay Switching Signal
Main Menu
1) To Operate The First device
2) To Operate The Second device
3) To Operate The Third device
4) To Operate The fourth device
5) To quit
Program designed by Akash Agarwal

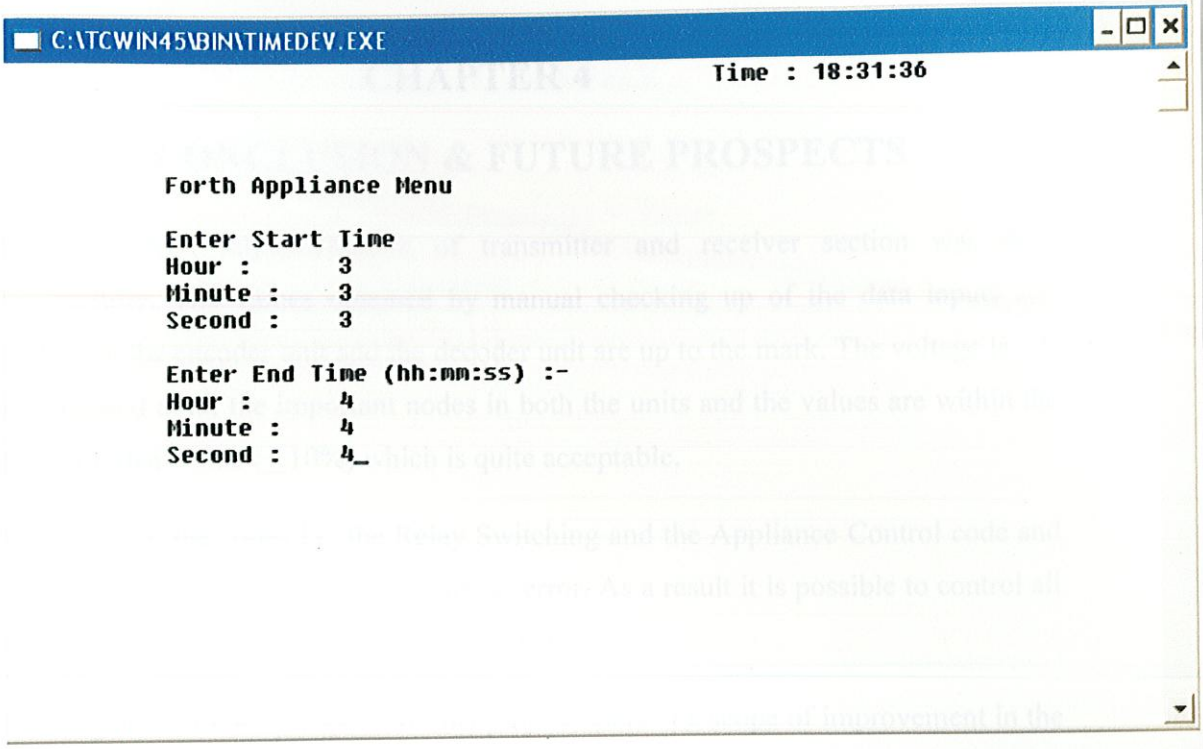
3.3.2 Appliance Control Program Results

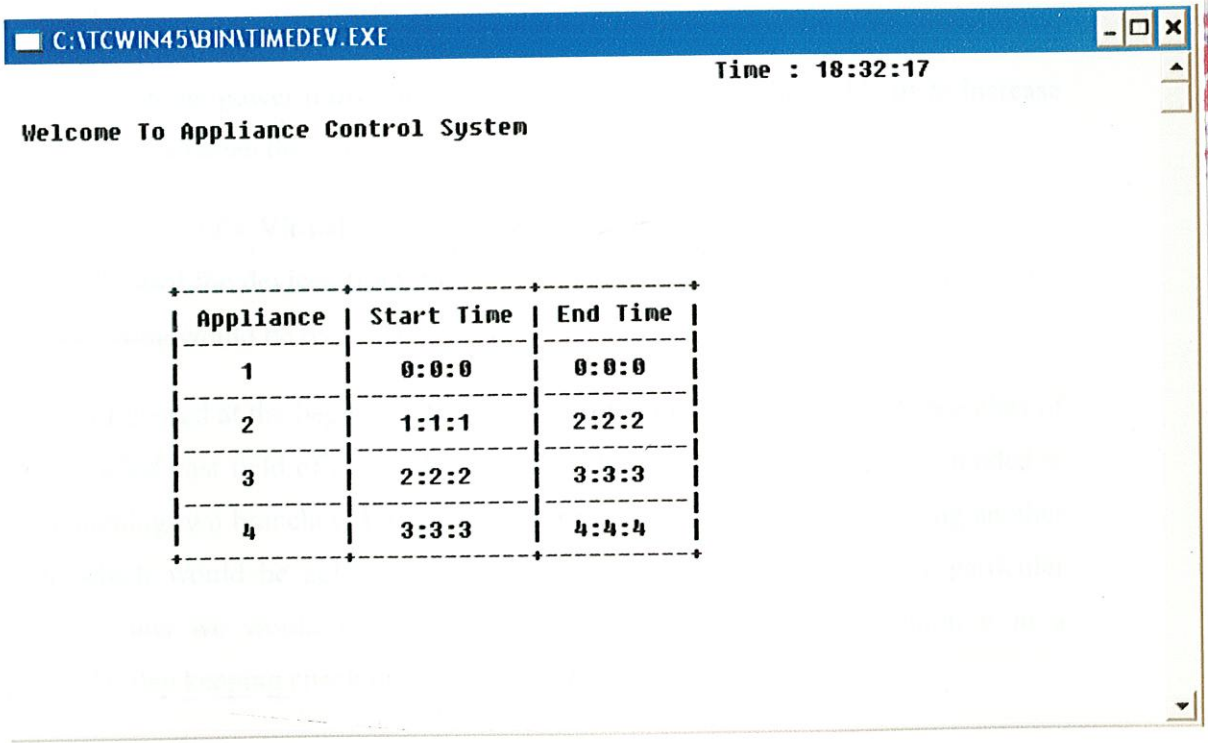
For detailed code refer Appendix B




C:\TCWIN45\BIN\TIMEDEV.EXE
Time : 18:29:53
Second Appliance Menu
Enter Start Time
Hour : 1
Minute : 1
Second : 1
Enter End Time (hh:mm:ss) :-
Hour : 2
Minute : 2
Second : 2_


C:\TCWIN45\BIN\TIMEDEV.EXE
Time : 18:30:58
Third Appliance Menu
Enter Start Time
Hour : 2
Minute : 2
Second : 2
Enter End Time (hh:mm:ss) :-
Hour : 3
Minute : 3
Second : 3_





Appliance	Start Time	End Time
1	0:0:0	0:0:0
2	1:1:1	2:2:2
3	2:2:2	3:3:3
4	3:3:3	4:4:4

CHAPTER 4

CONCLUSION & FUTURE PROSPECTS

In the project, implementation of transmitter and receiver section was done successfully. The values obtained by manual checking up of the data inputs and outputs at the encoder unit and the decoder unit are up to the mark. The voltage levels are checked at all the important nodes in both the units and the values are within the range of ideal value ($\pm$ 10%) which is quite acceptable.

Further, both the codes i.e. the Relay Switching and the Appliance Control code and Thermostat are working properly with no error. As a result it is possible to control all four of the devices from a personal computer.

Looking at the future prospects of this project, there is a scope of improvement in the system by

1. Increasing the power transmitted by the antenna. This would enable us to increase the distance between the transmitter and the receiver unit.

2. Introduction of a Virtual Private Network into the system. This would enable the user to control the devices from more than one computer using LAN. For this .NET programming would be required.

As we discussed at the beginning that our project is only one of the three branches of the so called vast field of Home Automation. This project can further be extended to its remaining two branches. We can even enhance this project by developing another code which would be able to calculate electricity consumption by any particular device. Thus we would be able to calculate total electricity consumption in a particular day keeping check on energy bills.

BIBLIOGRAPHY

BOOKS:

1. Programming The Parallel Port by Dhananjay V. Gadre

2. C How To Program by Deitel & Deitel

3. Home automation and wiring by James Gerhart

WEB PAGES:

4. www.google.com

5. www.wikipedia.com

6. www.homeauto.com

7. www.hyper-sense.com.tw

RESEARCH PAPERS:

8. Myers, Brad A. et al (2004) "Taking handheld devices to the next level". IEEE Computer Society, December 2004. pp. 36-45.

9. D. Greaves. "Control Software for Home Automation, Design Aspects and Position Paper". Icdcsw. Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops (ICDCSW'02), 2002.

10. Simon Aurell "Remote Controlling devices using Instant Messaging". Bachelor Thesis in Software Engineering, June 2005 (University of Gothenburg).

11. Armando Roy Delgado, Rich Picking and Vic Grout "Remote-Controlled Home Automation Systems with Different Network Technologies,2008

APPENDIX A

Code for Relay Switching

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>

int d1,d2,d3,d4,val;
void go()
{
val=0;
if(d1==1)
val+=1;
if(d2==1)
val+=2;
if(d3==1)
val+=4;
if(d4==1)
val+=8;
outportb(0x378,(val));
outportb(0x37a,0x00);
}
void device1()
{
char ans1,ch1;
ans1=1;
do
{
clrscr();
printf("\n\n First Device Menu\n\n");
printf("\n 1) To Switch On The Device \n");
printf("\n 2) To Switch Off The Device \n");
printf("\n 3) To Quit \n");
ch1=getch();
switch(ch1)
{
case '3':
ans1=0;
break;
case '1':
d1=1;
go();
break;
case '2':
d1=0;
```

```
go();
break;
default:
printf("\n Invalid choice\n");
getch();
break;
}
}
while(ans1==1);
clrscr();
}
void device2()
{
char ans1,ch1;
ans1=1;
do
{
clrscr();
printf("\n\n Second device Menu\n\n");
printf("\n 1) To Switch On The Device \n");
printf("\n 2) To Switch Off The Device \n");
printf("\n 3) To Quit \n");
ch1=getch();
switch(ch1)
{
case '3':
ans1=0;
break;
case '1':
d2=1;
go();
break;
case '2':
d2=0;
go();
break;
default:
printf("\n Invalid choice\n");
getch();
break;
}
}
while(ans1==1);
clrscr();
}
void device3()
{
char ans1,ch1;
```

```
ans1=1;
do
{
clrscr();
printf("\n\n Third Device Menu\n\n");
printf("\n 1) To Switch On The Device \n");
printf("\n 2) To Switch Off The device \n");
printf("\n 3) To Quit \n");
ch1=getch();
switch(ch1)
{
case '3':
ans1=0;
break;
case '1':
d3=1;
go();
break;
case '2':
d3=0;
go();
break;
default:
printf("\n Invalid choice\n");
getch();
break;
}
}
while(ans1==1);
clrscr();
}
void device4()
{
char ans1,ch1;
ans1=1;
do
{
clrscr();
printf("\n\n Fourth Device Menu\n\n");
printf("\n 1) To Switch On The Device \n");
printf("\n 2) To Switch Off The Device \n");
printf("\n 3) To Quit \n");
ch1=getch();
switch(ch1)
{
case '3':
ans1=0;
break;
```

```
case '1':
d4=1;
go();
break;
case '2':
d4=0;
go();
break;
default:
printf("\n Invalid choice\n");
getch();
break;
}
}
while(ans1==1);
clrscr();
}
void main()
{
clrscr();
char ans,ch;
d1=d2=d3=d4=0;
printf("\n\n Welcome To Relay Switching Signal \n\n");
ans=1;
do
{
printf("\n\n Main Menu\n\n");
printf("\n 1) To Operate The First device \n");
printf("\n 2) To Operate The Second device \n");
printf("\n 3) To Operate The Third device \n");
printf("\n 4) To Operate The fourth device \n");
printf("\n 5) To quit \n");
ch=getch();
switch(ch)
{
case '5':printf("\n\n Program designed by Akash Agarwal \n\n");
outportb(0x378,(0));
outportb(0x37a,0x00);
getch();
ans=0;
break;
case '1':
device1();
break;
case '2':
device2();
break;
case '3':
```

```
device3();
break;
case '4':
device4();
break;
default:
printf("\n\n Invalid choice\n\n");
getch();
break;
}
clrscr();
}
while(ans==1);
clrscr();
}
```

APPENDIX B

Code for Appliance Control

```
#include<stdio.h>
#include<stdio.h>
#include<conio.h>
#include<dos.h>

int d1,d2,d3,d4,val;
int hh=0,mm=0,ss=0;
int hh1=0,mm1=0,ss1=0;
int hh2=0,mm2=0,ss2=0;
int hh3=0,mm3=0,ss3=0;
int hh4=0,mm4=0,ss4=0;
int hh11=0,mm11=0,ss11=0;
int hh22=0,mm22=0,ss22=0;
int hh33=0,mm33=0,ss33=0;
int hh44=0,mm44=0,ss44=0;
struct  time tt;


void showtime()
{
        gettime(&tt);
        hh=0;
        mm=0;
        ss=0;
        mm= tt.ti_min;
        hh= tt.ti_hour;
        ss= tt.ti_sec;
        gotoxy(50,1);
        printf("Time : %d:%d:%d  ",hh,mm,ss);
}
void go()
{
     val=0;
     if(d1==1)
     val=val+1;
     if(d2==1)
     val=val+2;
     if(d3==1)
     val=val+4;
     if(d4==1)
```

```
        val=val+8;
        outportb(0x378,(val));
        outportb(0x37a,0x00);
}

void device1()
{
        clrscr();
        showtime();
        gotoxy(12,5);
        printf("First Appliance Menu");
        gotoxy(12,7);
        printf("Enter Start Time  ");
        gotoxy(12,8);
        printf("Hour : ");
        gotoxy(12,9);
        printf("Minute : ");
        gotoxy(12,10);
        printf("Second : ");
        gotoxy(24,8);
        scanf("%d",&hh1);
        gotoxy(24,9);
        scanf("%d",&mm1);
        gotoxy(24,10);
        scanf("%d",&ss1);
        gotoxy(12,12);
        printf("Enter End Time (hh:mm:ss) :- ");
        gotoxy(12,13);
        printf("Hour : ");
        gotoxy(12,14);
        printf("Minute : ");
        gotoxy(12,15);
        printf("Second : ");
        gotoxy(24,13);
        scanf("%d",&hh11);
        gotoxy(24,14);
        scanf("%d",&mm11);
        gotoxy(24,15);
        scanf("%d",&ss11);
}

void device2()
{
        clrscr();
        showtime();
        gotoxy(12,5);
        printf("Second Appliance Menu");
        gotoxy(12,7);
```

```
        printf("Enter Start Time  ");
        gotoxy(12,8);
        printf("Hour : ");
        gotoxy(12,9);
        printf("Minute : ");
        gotoxy(12,10);
        printf("Second : ");
        gotoxy(24,8);
        scanf("%d",&hh2);
        gotoxy(24,9);
        scanf("%d",&mm2);
        gotoxy(24,10);
        scanf("%d",&ss2);
        gotoxy(12,12);
        printf("Enter End Time (hh:mm:ss) :- ");
        gotoxy(12,13);
        printf("Hour : ");
        gotoxy(12,14);
        printf("Minute : ");
        gotoxy(12,15);
        printf("Second : ");
        gotoxy(24,13);
        scanf("%d",&hh22);
        gotoxy(24,14);
        scanf("%d",&mm22);
        gotoxy(24,15);
        scanf("%d",&ss22);
}
void device3()
{
        clrscr();
        showtime();
        gotoxy(12,5);
        printf("Third Appliance Menu");
        gotoxy(12,7);
        printf("Enter Start Time  ");
        gotoxy(12,8);
        printf("Hour : ");
        gotoxy(12,9);
        printf("Minute : ");
        gotoxy(12,10);
        printf("Second : ");
        gotoxy(24,8);
        scanf("%d",&hh3);
        gotoxy(24,9);
        scanf("%d",&mm3);
        gotoxy(24,10);
        scanf("%d",&ss3);
```

```
        gotoxy(12,12);
        printf("Enter End Time (hh:mm:ss) :- ");
        gotoxy(12,13);
        printf("Hour : ");
        gotoxy(12,14);
        printf("Minute : ");
        gotoxy(12,15);
        printf("Second : ");
        gotoxy(24,13);
        scanf("%d",&hh33);
        gotoxy(24,14);
        scanf("%d",&mm33);
        gotoxy(24,15);
        scanf("%d",&ss33);
}

void device4()
{
        clrscr();
        showtime();
        gotoxy(12,5);
        printf("Forth Appliance Menu");
        gotoxy(12,7);
        printf("Enter Start Time  ");
        gotoxy(12,8);
        printf("Hour : ");
        gotoxy(12,9);
        printf("Minute : ");
        gotoxy(12,10);
        printf("Second : ");
        gotoxy(24,8);
        scanf("%d",&hh4);
        gotoxy(24,9);
        scanf("%d",&mm4);
        gotoxy(24,10);
        scanf("%d",&ss4);
        gotoxy(12,12);
        printf("Enter End Time (hh:mm:ss) :- ");
        gotoxy(12,13);
        printf("Hour : ");
        gotoxy(12,14);
        printf("Minute : ");
        gotoxy(12,15);
        printf("Second : ");
        gotoxy(24,13);
        scanf("%d",&hh44);
        gotoxy(24,14);
        scanf("%d",&mm44);
```

```
        gotoxy(24,15);
        scanf("%d",&ss44);
}

void operate()
{
        clrscr();
        showtime();
        printf("\n\n Welcome To Appliance Control System \n\n");
       gotoxy(12,9);
        printf("+-----------+-----------+----------+");
        gotoxy(12,10);
        printf("| Appliance | Start Time | End Time |");
        gotoxy(12,11);
        printf("|-----------|-----------|----------|");
        gotoxy(12,12);
        printf("|   1     |          |         |");
        gotoxy(12,13);
        printf("|-----------|-----------|----------|");
        gotoxy(12,14);
        printf("|   2     |          |         |");
        gotoxy(12,15);
        printf("|-----------|-----------|----------|");
        gotoxy(12,16);
        printf("|   3     |          |         |");
        gotoxy(12,17);
        printf("|-----------|-----------|----------|");
        gotoxy(12,18);
        printf("|   4     |          |         |");
        gotoxy(12,19);
        printf("+-----------+-----------+----------+");
       gotoxy(27,12);
        printf(" %d:%d:%d",hh1,mm1,ss1);
        gotoxy(39,12);
        printf(" %d:%d:%d",hh11,mm11,ss11);

        gotoxy(27,14);
        printf(" %d:%d:%d",hh2,mm2,ss2);
        gotoxy(39,14);
        printf(" %d:%d:%d",hh22,mm22,ss22);

        gotoxy(27,16);
        printf(" %d:%d:%d",hh3,mm3,ss3);
        gotoxy(39,16);
        printf(" %d:%d:%d",hh33,mm33,ss33);

        gotoxy(27,18);
        printf(" %d:%d:%d",hh4,mm4,ss4);
```

```
gotoxy(39,18);
printf(" %d:%d:%d",hh44,mm44,ss44);


while (!kbhit())
{
        showtime();

        //1
        if((hh-hh1)>0 || (hh==hh1 && (mm-mm1)>0) || (hh==hh1 &&
mm==mm1 && (ss-ss1)>0))
        {
                d1=1;
                if((hh-hh11)>0 || (hh==hh11 && (mm-mm11)>0) || (hh==hh11
&& mm==mm11 && (ss-ss11)>0))
                        d1=0;
                else
                        d1=1;
        }
        else
                d1=0;

        //2
        if((hh-hh2)>0 || (hh==hh2 && (mm-mm2)>0) || (hh==hh2 &&
mm==mm2 && (ss-ss2)>0))
        {
                d2=1;
                if((hh-hh22)>0 || (hh==hh22 && (mm-mm22)>0) || (hh==hh22
&& mm==mm22 && (ss-ss22)>0))
                        d2=0;
                else
                        d2=1;
        }
        else
                d2=0;

        //3
        if((hh-hh3)>0 || (hh==hh3 && (mm-mm3)>0) || (hh==hh3 &&
mm==mm3 && (ss-ss3)>0))
        {
                d3=1;
                if((hh-hh33)>0 || (hh==hh33 && (mm-mm33)>0) || (hh==hh33
&& mm==mm33 && (ss-ss33)>0))
                        d3=0;
                else
                        d3=1;
        }
        else
```

```
                d3=0;


\               //4
                if((hh-hh4)>0 || (hh==hh4 && (mm-mm4)>0) || (hh==hh4 &&
mm==mm4 && (ss-ss4)>0))
                {
                        d4=1;
                        if((hh-hh44)>0 || (hh==hh44 && (mm-mm44)>0) || (hh==hh44
&& mm==mm44 && (ss-ss44)>0))
                                d4=0;
                        else
                                d4=1;
                }
                else
                        d4=0;


                go();
                delay(200);

        }
}

void main()
{
      clrscr();

      char ans,ch;
d1=d2=d3=d4=0;
      showtime();
      printf("\n\n Welcome To Appliance Control System \n\n");
      ans=1;
      do
{
      clrscr();
      showtime();
      printf("\n\n Welcome To Appliance Control System \n\n");
      printf("\n\n Main Menu\n\n");
      printf("\n 1) To Operate First Appliance \n");
      printf("\n 2) To Operate Second Appliance \n");
      printf("\n 3) To Operate Third Appliance \n");
      printf("\n 4) To Operate Fourth Appliance \n");
      printf("\n 5) To Start Operation \n");
      printf("\n 6) To quit \n");
      ch=getch();
      switch(ch)
{
```

```
case '6':printf("\n\n Program Developed By:- AKASH \n\n");
outportb(0x378,(0));
outportb(0x37a,0x00);
getch();
ans=0;
break;
case '1':
device1();
break;
case '2':
device2();
break;
case '3':
device3();
break;
case '4':
device4();
```